

УДК 658.7

А.В. ПОПОВ, М.В. ИВАНОВ, Я.В. ИЛЮШКО

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***ОПТИМИЗАЦИЯ ПОТОКОВ В РАСПРЕДЕЛЕННОЙ ТРАНСПОРТНОЙ СЕТИ**

Поставлена и решена задача оптимизации транспортных логистических потоков промышленного предприятия для нахождения маршрута грузоперевозок между заранее заданными пунктами внутри города без преобразования карты в граф. Предложен алгоритм распространения волны для нахождения кратчайшего маршрута грузоперевозок. Приведен и представлен в формализованном виде алгоритм преобразования исходного изображения в маршрутную матрицу. Представлен пример использования алгоритма. Представленный подход можно использовать для оптимизации транспортных потоков в логистике.

Ключевые слова: логистика, транспортные потоки, алгоритм Ли, волновой алгоритм, маршрутная матрица, формализация.

Введение

Современное развитие информационных технологий, рост предприятий, сотрудничество с внешними организациями, удорожания энергоносителей создают необходимость в разносторонней организации и оптимизации потоков в логистике.

Программные продукты, на данный момент времени решают многие задачи транспортной логистикой (LogisticsMaster, ANTOR MonitorMaster). Но учитывая их высокую стоимость, а также необходимость использования вспомогательного программного обеспечения для их работы, в следствии чего они фактически выступают расширяющими плагинами других приложений. Задачей данной публикации является формирование подхода для создания программного продукта, который будет предоставлять аналогичные возможности, основываясь на простых, не требующих больших затрат методах.

Рассмотрев существующие алгоритмы [1 – 3] (поиск в ширину, поиск в глубину, алгоритм Дейкстры, задачу коммивояжера, двунаправленный поиск, алгоритм Флойда — Уоршелла и другие), их достоинства и недостатки, можно сделать вывод, что наиболее подходящим для реализации является алгоритм Ли, основанный на распространении волны.

Данный алгоритм позволяет не только найти кратчайшее расстояние между маркерами (точки, определяющие начало маршрута и конечные элементы распространения), но и учесть все непроходимые места на карте. Использование этого алгоритма позволяет найти маршрут и просчитать среднее время транспортировки, затраты на перевозку и т.д.

Постановка задачи исследования

На пути движения материального потока от начальной точки – первичного источника сырья, до конечной – конечный потребитель, находятся логистические операции, осуществляемые при помощи различных транспортных средств. Затраты на выполнение всех транспортных операций составляют не менее 50% от общей суммы затрат отводимой логистике [4 – 6].

В том случае, когда объемы транспортной работы выделяются в большой самостоятельный массив (например, при функционировании транспорта общего пользования, а также в ряде случаев транспорта необщего пользования), возникает ряд специфических задач, которые относят к задачам транспортной логистики. Для обеспечения эффективной деятельности транспортной системы предприятия необходимо учесть следующие моменты:

- создание транспортных систем, в том числе создание транспортных коридоров и транспортных цепей;
- совместное планирование транспортных процессов с учетом различных видов (в случае смешанных перевозок);
- обеспечение технологического единства транспортно-складского процесса;
- совместное планирование транспортного процесса со складским и производственным;
- выбор вида транспортного средства;
- выбор типа транспортного средства;
- определение рациональных маршрутов доставки.

Транспортный коридор - это часть национальной или международной транспортной системы,

которая обеспечивает значительные грузовые перевозки между отдельными географическими районами. Включает в себя: подвижные транспортные средства и стационарные устройства всех видов транспорта, работающих на данном направлении, а также совокупность правовых условий осуществления этих перевозок.

Транспортная цепь состоит из этапов перевозки груза на определенные расстояния, в течение определенного периода времени, с использованием транспортных средств одного или нескольких видов транспорта. Все это время грузы остаются в неизменном виде (например, грузовой пакет или контейнер).

Проводя разносторонний анализ по всем этапам транспортировки грузов, можно снизить затраты, а также время, затраченное на транспортировку.

В работе используется волновой алгоритм [7] для решения подобного класса задач, который позволит достаточно просто решить задачу маршрутизации.

Большинство приложений используют алгоритмы, которые опираются на теорию графов и нуждаются в создании маршрутной базы, где будет, как минимум, информация о дорогах. Для наполнения данными такой базы, приходится вручную формировать граф по изображению транспортной сети, что требует больших затрат сил и времени.

Решение задачи исследования

Кратко рассмотрим алгоритм Ли [7] — волновой алгоритм поиска пути на карте (алгоритм трассировки). С его помощью можно построить путь, или трассу, между двумя любыми элементами в лабиринте.

Из начального элемента распространяется в четырёх направлениях волна. Тот элемент, в который она пришла, образует фронт волны.

Элементы первого фронта волны являются источниками вторичных волн.

Элементы второго фронта генерируют волну третьего фронта и так далее. Процесс заканчивается при достижении конечного элемента. На втором этапе строится трасса. Построение производится в соответствии с некоторыми правилами:

- при построении трассы движение проходит в зависимости от выбранных приоритетов,
- путевые координаты уменьшаются при переходе от начального элемента к конечному.

Приоритеты выбираются в процессе разработки. В зависимости от выбора тех или иных приоритетов получаются различные трассы.

Используя волновой алгоритм, можно найти трассу в лабиринте с любым количеством стен. В этом

и заключается преимущество его использования.

Недостаток алгоритма Ли в том, что при построении трассы требуется большой объем памяти, но возможности современных компьютеров позволяют этот минус преодолеть.

Нахождение кратчайшего пути можно разбить на два подэтапа: анализ изображения, с созданием анализируемой матрицы и анализ матрицы путем нахождения кратчайшего маршрута между точками.

Для более подробного и понятного описания алгоритма воспользуемся системой расширенных алгебр алгоритмов [8, 9] (она позволяет представить алгоритм в виде краткого псевдокода на k-значной логике).

Первый этап включает в себя четыре *правила преобразования изображения* в маршрутную матрицу, основываясь на множествах проходимых и не проходимых точек, используя сравнение и занесение в матрицу соответствующих значений:

Правило 1

ЕСЛИ «элемент массива» РАВЕН «красный»
ТО «фиксируем начальную точку»

Правило 2

ЕСЛИ «элемент массива» РАВЕН «зеленый»
ТО «фиксируем конечную точку»

Правило 3

ЕСЛИ «элемент массива» РАВЕН «белый»
ИЛИ «синий» ИЛИ ... ТО «фиксируем проходимую точку»

Правило 4

ЕСЛИ «элемент массива» РАВЕН «остальные»
ТО «фиксируем точку препятствия»

МАССИВ= \langle ВВОД(М) *
[сост_c](b⁰n v b^kn v b¹n v b^mn)
*ПРАВИЛА \rangle

ВВОД(М) – ввод элемента карты для анализа;

сост_c = 0 – цвет «красный», k – цвет «зеленый», 1 – цвет принадлежащий множеству проходимых точек, m – цвет принадлежащий множеству не проходимых точек, фиксирующее условие n;

Пусть СТ{a₁,a₂,...,a_t} – множество проходимых точек, где a_i – цвет принадлежащий этому множеству. CF{b₁,b₂,...,b_u} – множество проходимых точек, где b_i – цвет принадлежащий этому множеству.

Детализируем оператор ПРАВИЛА, для этого введем следующие операторы:

ФНТ – фиксация координат начальной точки;

ФКТ – фиксация конечной точки;

ФПТ – фиксация проходимой точки;

ФНПТ – фиксация препятствия;

крас – «красный» цвет, начальная точка;

прох – элемент множества цветов проходимых точек;

непр – элемент множества цветов не проходимых точек;

зел – зеленый цвет, конечная точка.

С учетом операторов регулярная схема будет выглядеть следующим образом:

ПРАВИЛА = { [п= крас](ФНТ)*
[прох](ФПТ)
[непр](ФНПТ)
*[зел](ФКТ) }.

Данный алгоритм можно представить в виде блок-схемы (рис. 1).

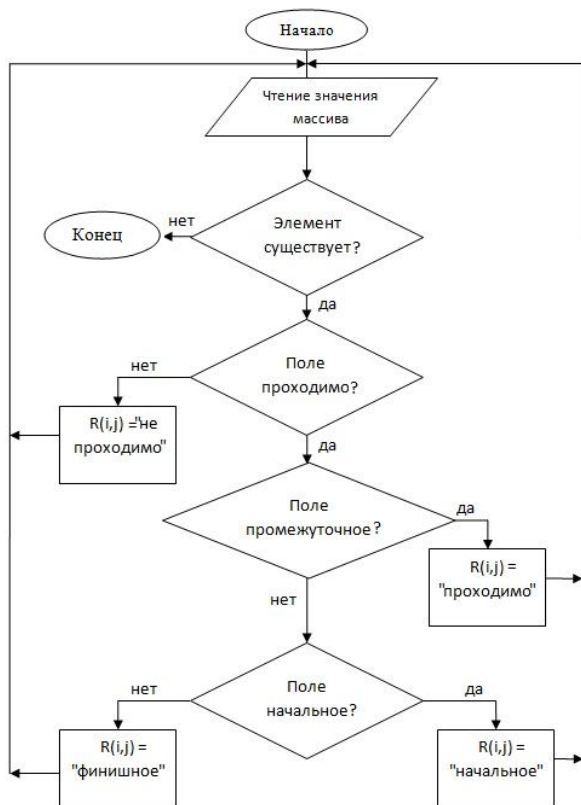


Рис. 1. Преобразование элемента карты в элемент матрицы

Рассмотрим правила описывающие поведения алгоритма при построение маршрута.

Правила для просмотра элемента (ПР):

ПРАВИЛА_ПР

ЕСЛИ «элемент массива» РАВЕН «счётчик итераций» ТО ПР_ЭЛ

Правила используемые при поиске необходимых элементов:

ПРАВИЛА_ПСК

Правило 1

ЕСЛИ «элемент массива» РАВЕН «начальная точка» ТО ПСК_ОКР

Правило 2

ЕСЛИ «элемент массива» РАВЕН «проходимая точка» ТО «элемент массива» ПРИСВАИВАНИЕ «счётчик итераций»

Правило 3

ЕСЛИ «элемент массива» РАВЕН «остальные» ТО «оставить без изменений»

Правила наращивания счетчика итераций (СИ):

ПРАВИЛА_СИ

Правило 1

ЕСЛИ «окончание просмотра» РАВЕН единица ТО «счётчик итераций» РАВЕН «увеличенный счетчик итераций»

Правило 2

ЕСЛИ «окончание просмотра» РАВЕН ноль ТО «продолжение просмотра»

Регулярная схема просмотр элементов

ЭЛ=< ВВОД(МИ)*
[сост_м](b⁰n v b^kn v b¹n)
*ПРАВИЛА_ПСК>
ПРАВИЛА_ПСК={ [нт](ПСК_ОКР)*
[кт](ЭСУ)
*[ост](ОБИ) }.

Регулярная схема для счетчика итераций

СИ=< ВВОД(ССИ) *
[п_прос](i⁰n v i¹n)
*ПРАВИЛА_СИ>
ПРАВИЛА_СИ={ [0](ПРП)*
[1](УСИ)[пред_счет](ОСТАНОВ) }

ВВОД(МИ) – ввод рассматриваемого элемента;
ВВОД(ССИ) – ввод текущего состояния счетчика итераций;

п_прос — принимает значение единицы при завершении по.строчного просмотра массива;

сост_м – состояние рассматриваемого элемента, где 0 – точка поиска, переход к поиску точек в окрестности, 1 – оставить без изменений и k – увеличить фронт волны;

пред_счет – предельное состояние счетчика (если достигнуто, то алгоритм завершен);

нт – начальная точка для распространения волны (на каждой итерации разная);

кт – конечная точка;

ост – оставшиеся точки, входящие во множество не рассматриваемых;

ЭСУ – задание значения фронта волны в точки;

ОБИ – оставления значения без изменения;

УСИ – увеличение счетчика итераций;

ПРП – продолжение просмотра массива;

ПСК_ОКР — функция поиска минимального значения среди клеток (выполняется для клеток в окрестности рассматриваемой точки любым подходящим способом);

ПР_ЭЛ — функция вызываемая для просмотра окрестных элементов рассматриваемой точки;

ОСТАНОВ — завершение алгоритма.

Цвета, используемые для нахождения маршрутной матрицы, основываются на множестве цветов, относящихся к гамме анализируемой карты и

соответственно могут изменяться в зависимости от используемой карты.

Использование предложенного подхода сводится всего лишь к хранению карты разбитой на участки и к базе данным, которая содержит контрольные точки с привязкой по участкам и координатами (по необходимости).

На рис. 2 приведен пример тестового лабиринта с нанесенными на него маркерами, где «красная» точка обозначает начало, «зеленая» - конец.

Применив волновой алгоритм к заданному лабиринту, получим кратчайшее расстояние между точками (рис. 3).

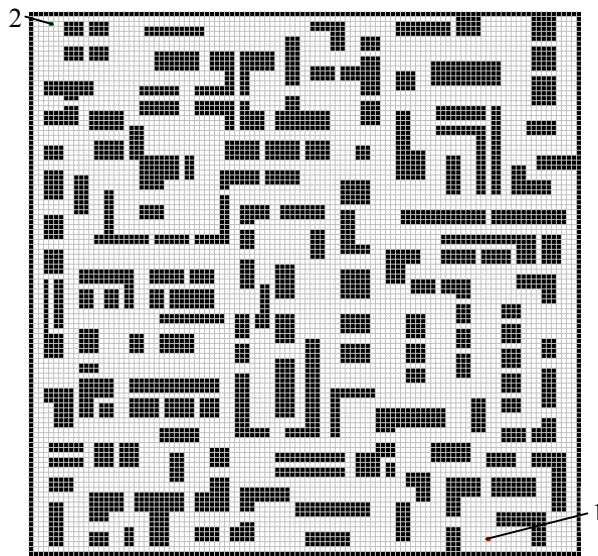


Рис. 2. Карта с отмеченными маркерами:
1 – начальная точка (точка «красного» цвета);
2 – конечная точка («зеленый» цвет)

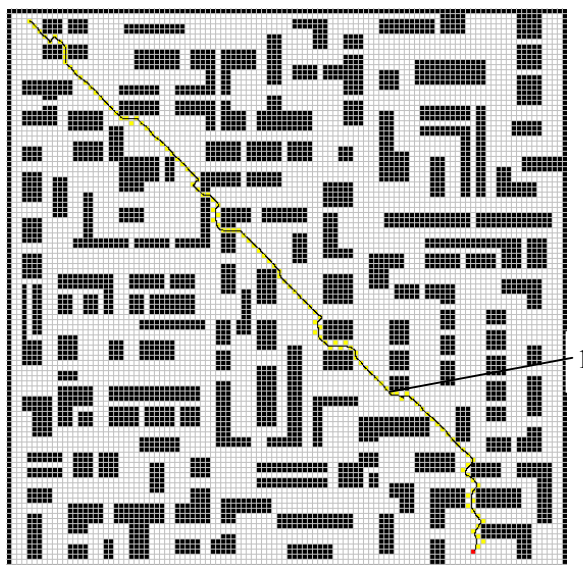


Рис. 3. Карта с полученным маршрутом:
1 – найденный маршрут

Выше было сказано, что данный алгоритм преобразует исходное изображение в матрицу и в последствии работает с ней. Поэтому представим её после преобразования и после работы алгоритма.

Табл. 1 демонстрирует участок матрицы, после преобразования изображения, на котором находится начальная точка. Значение, которое она принимает, на этапе инициализации, равно 1998. Значения проходимых и непроходимых точек выбираются опытно-теоретическим путем операясь на размер изображения и число итераций нахождения маршрута (непроходимая точка предполагается равной бесконечности, проходимая – максимуму числу итераций, начальная – на единицу меньше от максимального числа итераций, альтернативный вариант – значения превышающие количество итераций, либо количество проходимых точек).

Таблица 1
Участок матрицы с начальной точкой

2000	2000	1999	1999	2000	2000	2000
1999	1999	1999	1999	1999	1999	1999
1999	1999	1999	1999	1999	1999	1999
1999	1999	1999	1999	1999	1999	1999
1999	1999	1999	1999	1999	1999	1999
1999	1999	1999	1999	1999	1999	1998
1999	1999	1999	1999	1999	1999	1999
1999	2000	2000	2000	1999	1999	1999
1999	2000	2000	2000	1999	1999	1999
1999	2000	2000	2000	1999	1999	1999

Табл. 2 в свою очередь демонстрирует участок матрицы с конечной точкой, со значением равным 0 (эта точка является инициатором распространения фронтов волны).

Таблица 2
Участок матрицы с конечной точкой

2000	2000	2000	2000	2000	2000	2000
2000	1999	1999	1999	1999	1999	1999
2000	1999	1999	1999	1999	1999	1999
2000	1999	1999	1999	1999	1999	1999
2000	1999	0	1999	1999	1999	1999
2000	1999	1999	1999	1999	1999	1999
2000	1999	1999	1999	1999	1999	1999
2000	1999	2000	2000	2000	1999	1999
2000	1999	2000	2000	2000	1999	1999
2000	1999	2000	2000	2000	1999	1999

В результате выполнения волнового алгоритма получим матрицу на которой отображаются волны первого, второго, третьего и т.д фронтов, относительно которых мы будем формировать маршрут, выбирая значения в порядке их возрастания/убывания.

Табл. 3, 4 демонстрируют участки матрицы распространения волны с уже полученными значениями каждого фронта.

Таблица 3

Участок матрицы
с сформированными начальными фронтами волн

3	3	3	3	3	4	5	6
2	2	2	2	3	4	5	6
1	1	1	2	3	4	5	6
1	0	1	2	3	4	5	6
1	1	1	2	3	4	5	6
2	2	2	2	3	4	5	6
3	2000	2000	2000	3	4	2000	2000
4	2000	2000	2000	4	4	2000	2000
5	2000	2000	2000	5	5	2000	2000
6	2000	2000	2000	6	6	2000	2000
7	7	8	7	7	7	7	8
8	2000	2000	2000	8	8	2000	2000
9	2000	2000	2000	9	9	2000	2000
10	2000	2000	2000	10	10	2000	2000
11	2000	2000	2000	11	11	2000	2000
12	12	13	12	12	12	12	13
13	13	13	13	13	13	13	13

Таблица 4

Участок матрицы
с сформированными конечными фронтами волн

112	112	113	114	115	116	117
112	113	113	114	115	116	117
112	113	114	114	115	116	117
2000	113	114	2000	2000	2000	2000
2000	114	114	2000	2000	2000	2000
2000	115	115	2000	2000	2000	2000
115	116	116	116	117	118	119
115	116	117	117	117	118	119
115	116	117	118	118	118	119
115	116	117	118	119	119	119
115	116	117	118	119	120	120
115	116	117	118	119	1998	122
115	116	117	118	119	120	121
2000	2000	2000	118	119	120	121
2000	2000	2000	119	119	120	121

Из таблиц видно что фронты волны формируются для всех элементов матрицы, что увеличивает время построение маршрута, для избавления от этого недостатка необходимо вводить поправочные коэффициенты.

Заключение

Рассмотрен один из возможных вариантов оптимизации транспортных логистических потоков промышленного территориально-распределенного предприятия, позволяющий найти оптимальный маршрут грузоперевозок между заранее заданными пунктами внутри города, без преобразования карты в граф.

Волновой алгоритм нахождения кратчайшего маршрута представлен в формализованном виде с использованием расширенной алгебры алгоритмов. Описан алгоритм преобразования исходного изображения в маршрутную матрицу. Представлены пример задания исходных данных и результаты работы алгоритма с использованием случайного лабиринта.

Недостатком предложенного подход является отсутствие возможности определения направленности дороги, но достоинством является быстрая скорость реализации приложения.

Предложенный подход может быть использовано в целях оптимизации транспортных потоков в логистике, получения сметных данных маршрута.

Алгоритм можно применить в задаче автоматического построения графов, и других схожих задачах.

Литература

1. *Алгоритмы: построение и анализ [Текст] : учеб. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – 2-е изд. – М. : Вильямс, 2005. – 1292 с.*
2. *Алгоритмы: построение и анализ [Текст] : учеб. / Х. Томас, Т. Кормен, Ч. Лейзерсон [и др.]. – М. : Вильямс, 2004. – 1152 с.*
3. *Левитин, А.В. Алгоритмы: введение в разработку и анализ [Текст] : учеб. / А.В. Левитин. – М. : «Вильямс», 2006. – 576 с.*
4. *Миротин, Л.Б. Транспортная логистика [Текст] : учеб. пособие / Л.Б. Миротин, Б.Э. Ташбаев. – М. : Брандес, 1996. – 511 с.*
5. *Гаджинский, А.М. Логистика: Учебник для высших и средних специальных учебных заведений [Текст] / А.М. Гаджинский. – М. : Информационно-внедренческий центр "Маркетинг", 1999. – 228 с.*
6. *Неруш, Ю.М. Логистика [Текст] : учеб. / Ю.М. Неруш. – М. : Проспект, 2006. – 520 с.*
7. *Lee, C.Y. An algorithm for Path Connections and Its Applications [Текст] / C.Y. Lee // IRE Transactions on Electronic Computers. Science journal. – 1961. – Vol. EC-10, № 2. – P. 364 – 365.*

8. Акуловский, В.Г. К – значная логика в расширенной алгебре алгоритмов [Текст] / В.Г. Акуловский // Проблемы программирования. – 2008. – № 2. – С. 50 – 56.

9. Акуловский, В.Г. Расширенная алгебра алгоритмов [Текст] / В.Г. Акуловский // Проблемы программирования. – 2007. – № 3. – С. 3 – 15.

Поступила в редакцию 6.02.2012

Рецензент: д-р техн. наук, проф., проф. каф. АСУ И.П. Гамаюн, Национальный технический университет «ХПИ», Харьков.

ОПТИМІЗАЦІЯ ТРАНСПОРТНИХ ПОТОКІВ ПРОМИСЛОВОГО ПІДПРИЄМСТВА

А.В. Попов, М.В. Иванов, Я.В. Ілюшко

Поставлена на вирішена задача оптимізації транспортних логістичних потоків промислового підприємства, для знаходження маршруту вантажоперевезень між задалегідь заданими пунктами в середині міста, без перетворення карти в граф. Запропоновано алгоритм знаходження найкоротшого маршруту вантажоперевезень. Наведено та представлено у формалізованому виді алгоритм перетворення початкового зображення в маршрутну матрицю. Наведено приклад використання алгоритму. Наведений підхід може використовувати для оптимізації транспортних потоків в логістиці.

Ключові слова: логістика, транспортні потоки, алгоритм Лі, хвильовий алгоритм, маршрутна матриця, формалізація.

OPTIMIZATION OF THE TRANSPORT FLOW OF THE INDUSTRIAL ENTERPRISE

A. V. Popov, M. V. Ivanov, Ya. V. Ilushko

The task of the industrial enterprise transport flow optimization that allows to find an optimal route of a cargo transportation between in advance set points, without map conversion to graph is set and solved. The shortest route finding algorithm is proposed. The algorithm of the image conversion in a rout matrix is fetched and formalized. An example of algorithm usage is shown. The solution can be used for transport flows optimization in logistics.

Keywords: logistics, transport flows, Li's algorithm, wave algorithm, way matrix, formalization.

Попов Андрей Вячеславович – канд. техн. наук, доцент, доцент кафедры Информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Иванов Михаил Валерьевич – магистрант кафедры Информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Ілюшко Ярослав Викторович – канд. техн. наук, доцент кафедры основ компьютерного конструирования, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.