

Р.К. Абаев, Х.К. Албегов, В.А. Дзюев
(Северо-Кавказский горно-металлургический институт (ГТУ);
e-mail: rus_06_06@mail.ru)

ОБ ОПТИМИЗАЦИИ ПЛАН-ТРАСС КАНАЛОВ СВЯЗИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

Проведён сравнительный анализ и выбран один из методов нахождения оптимальной структуры каналов связи АСУ, разработан соответствующий алгоритм. Результаты могут быть полезны при решении проблем техносферной безопасности.

Ключевые слова: алгоритм, кратчайший путь, оптимизация, динамическое программирование, обход.

R.K. Abaev, Kh.K. Albegov, V.A. Dziov ABOUT OPTIMIZATION OF PLAN OF SLOPES OF CHANNELS OF AUTOMATED CONTROL SYSTEMS

Comparative analysis and selected one of the methods for finding the optimal structure of the communication channels of automated control systems, developed appropriate algorithm. The results may be useful in solving the problems of technosphere safety.

Key words: algorithm, shortest path, optimization, dynamic programming, bypass.

На сегодняшний день существует несколько методов нахождения кратчайшего пути прокладки план-трасс каналов связи. Задача авторов в том, чтобы определить кратчайший путь для выбора оптимальной структуры каналов связи АСУ на схеме автоматизации, а также для дальнейшей реализации в программном модуле.

Для построения кратчайших соединений между устройствами необходимо решить две основные задачи:

1. Разработать алгоритм поиска кратчайшего соединения.
2. Разработать программу решения задачи в AutodeskAutoCAD.

В первую очередь, необходимо найти способ представить схему автоматизации в памяти таким образом, чтобы можно было с ней работать в среде программирования. Для данной работы была использована среда Borland Delphi 7.

Представим схему автоматизации в форме лабиринта точек, лабиринт – в форме графа.

Деревом графа будем называть конечное множество точек, входящих в этот граф.

Многие реальные объекты могут быть представлены с помощью графов, а также целый ряд задач можно решить, используя теорию графов, в том числе и поиск кратчайшего пути. Для представления лабиринта в памяти ЭВМ создадим двумерный массив. Все пространство обозначим фиктивными значениями.

Введём обозначения (рис. 1):

-1 – пустое пространство (которое не нужно обходить);

-2 – непустое пространство (которое нужно обойти).

-2	-2	-2	-2	-2
-2	-1	-1	-1	-2
-2	-2	-2	-1	-2
-2	-1	-1	-1	-2
-2	-1	-1	-1	-2
-2	-2	-2	-2	-2

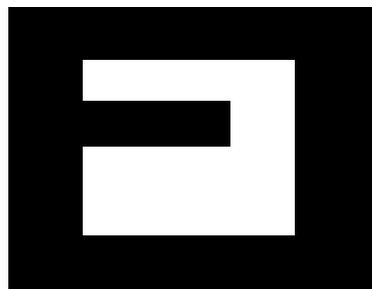


Рис. 1. Представление лабиринта

При высоком разрешении такой способ позволяет представить любой чертеж AutoCAD в виде двумерной матрицы, которая будет использоваться для построения кратчайшего пути.

После анализа, сделанного в предыдущих работах [1, 2], нами были детально рассмотрены наиболее подходящие из распространённых методов – это метод рекурсивного обхода и волновой метод.

Прохождение дерева рекурсивным обходом состоит в исследовании карты (в нашем случае – чертежа) до тех пор, пока не будет найдено решение. Путь проходит по прямой до тех пор, пока не встретится с пересекающимся путём (путями). При нахождении пересекающегося пути необходимо определить направление дальнейшего движения из возможных, если выбранный путь оказался неверным, возвращаемся и выбираем другой. Если на пути встречается финишная локация – решение найдено. При этом сохраняется весь путь для определения финишной локации. Чаще всего приходится сталкиваться не с финишной локацией, а с тупиком. Тупик бывает двух видов: либо некуда идти, либо там, куда можно идти, мы уже были. В этом случае следует сделать шаг назад и выбрать другой путь. Если шаг назад ничего не дает, то нужно сделать еще один шаг назад и так до тех пор, пока не появится альтернатива. Если же все варианты исчерпаны, а решение не найдено, то решения не существует. Такое может случиться, если финишная локация окружена сплошной стеной.

К преимуществам метода стоит отнести то, что он прост в реализации. Недостатки метода:

- обход происходит нерационально и может пройти много времени, прежде чем решение будет найдено;
- полученное решение может быть неоптимальным.

Рассмотрим ещё один пример для рекурсивного обхода: проход, ведущий вниз от стартовой локации, заканчивается финишной локацией. Если пойти прямо, то мы попадём в "зал" – группу локаций без ограничений, а затем дойдем до тупика. Каждая клетка "зала" – это перекрёсток из четырёх дорожек. На каждом перекрестке есть три варианта пути ($3 \times 3 = 9$). Далее опять три варианта (уже $3 \times 3 \times 3 = 27$) и т.д. Количество вариантов растёт в геометрической прогрессии, поэтому при реализации данного алгоритма в программном коде программа может просто зависнуть, что приведёт к потере данных.

Для поиска решения поставленной задачи нами был выбран волновой метод, который гарантированно позволяет построить кратчайший путь от точки до точки, а также информирует о том, когда невозможно их соединить.

Суть метода удобнее будет описать на следующем примере: пусть в стартовой локации опрокидывают бочку с водой. Вода будет растекаться по сторонам, постепенно добираясь даже до самых отдаленных мест. Рано или поздно достигается финишная локация. При этом необходимо вести наблюдение за тем, как она туда попадает. Это и будет кратчайший путь. Если двигаться дальше некуда, а цель не достигнута, то решения нет.

После применения волнового метода к решению нашей задачи был выявлен существенный недостаток – он работает крайне медленно для средних и больших чертежей (от $150 \times 150 \text{ dpi}$ и более). Например, расчёт пути на чертеже с разрешением 800×800 при средних характеристиках ЭВМ занимает около 20 мин. Это происходит потому, что на каждой итерации цикла просматривается весь чертеж полностью.

Поэтому авторами предлагается усовершенствовать алгоритм с использованием динамического программирования. Суть предложения заключается в том, что те данные, которые были использованы нами при прохождении пути, запоминаются и не проверяются вновь. Тогда при решении задачи нужно просматривать только те ячейки, которые имеют значение "текущая итерация" + 1. Для этого необходимо иметь список всех итераций на текущем шаге. В дополнение к этому, нужно улучшить алгоритм таким образом, чтобы он корректно работал в AutoCAD. Если мы будем выводить множество полученных точек и соединять их короткими линиями, то получившаяся полилиния будет содержать слишком много элементов. Это приведет к тому, что AutoCAD перестанет отвечать на запросы. Для предотвращения этой проблемы нужны координаты начала и конца каждой линии. Пока некое множество точек имеет одну общую координату, мы имеем дело с прямой линией. Если общие координаты не обнаружены – записываем координаты получившейся линии и ведём слежение за новыми точками.

Блок-схема модифицированного алгоритма представлен на рис. 2.

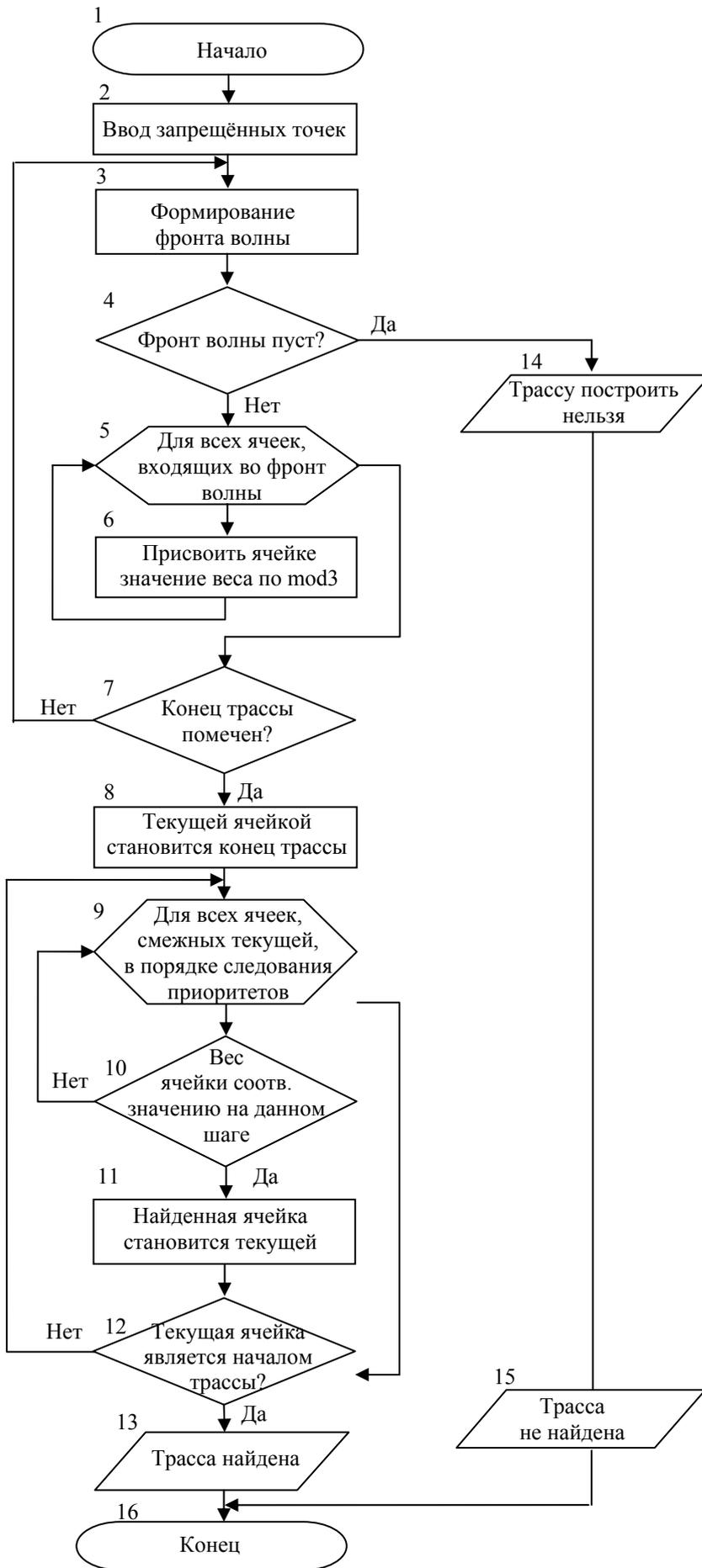


Рис. 2. Блок-схема модифицированного волнового алгоритма

Работа алгоритма делится на два этапа:

Этап 1 – поиск пути прокладки кабельной план-трассы каналов связи.

В блоке 2 происходит загрузка матрицы-схемы распознавания запрещённых ячеек, то есть определение препятствий, которые встретятся на пути прокладки кабеля.

Блок 3 определяет возможные пути по методу волнового алгоритма.

В блоке 4 проверяется возможность прохода волны по выбранному направлению движения волны. Если определено, что фронт волны является непустым, переходим к блоку 5. Если же фронт волны является пустым, выводится сообщение – "трассу построить нельзя" (в блоке 14), после чего алгоритм завершает свою работу.

Блок 5 определяет ячейки, которые участвуют в распространении волны на текущем ходе.

В блоке 6 присваиваются значения весовых коэффициентов на текущем ходе.

В блоках 7 и 8 обрабатывается цикл до тех пор, пока не найдёт конца пути.

Этап 2 – построение пути.

В блоках 9 и 10 определяются смежные ячейки с текущими значениями веса.

В блоке 11 найденная ячейка становится текущей, пока не доходит до начала трассы.

В результате проведённой работы были проанализированы методы оптимизации план-трасс каналов связи АСУ, выявлены преимущества и недостатки каждого из методов, определен наиболее подходящий метод, на основании выбранного метода разработан алгоритм. Разработанный алгоритм был реализован на одном из языков программирования. До внедрения алгоритма в программный код, то есть в классическом его представлении, обработка всего пространства в графическом представлении 800×800 пикселей для оптимальной прокладки занимала около 20 минут на ЭВМ со средними характеристиками (AMD Athlon 64×2, 2,21 GHz, ОЗУ 2 Гб). После внедрения данного алгоритма в программу результат для волнового метода значительно улучшился. На той же ЭВМ при разрешении 1600×1600 программа строит путь трассы кабеля в среднем за 2 секунды.

Поставленная задача для оптимальной прокладки план-трассы была решена, произведена апробация на примере проектирования реального объекта. Это позволило судить о том, что предлагаемое изменение алгоритма дает преимущества в соотношении время/качество. Иными словами, применение динамического программирования к волновому методу позволило ускорить процесс обработки информации с более высоким качеством в предлагаемой области, где необходимо проложить кабельную трассу.

Литература

1. **Абаев Р.К.** Анализ методик оптимизации маршрутов прокладки кабельных трасс при проектировании распределённой автоматизированной системы управления // VIII Всероссийская научно-техническая конференция: сборник статей. Пенза, 2008. 277 с.
2. **Струченков В.И.** Методы оптимизации. Основы теории, задачи, обучающие компьютерные программы: учеб. пособие. М.: Изд-во "Экзамен", 2005. 256 с.
3. **Полещук Н.Н.** AutoCAD: разработка приложений, настройка и адаптация. С.-Пб.: БХВ-Петербург, 2006. 992 с.
4. **Архангельский А.Я.** Программирование в Delphi 7. М.: Изд. ООО "Бином-Пресс", 2004. 1152 с.
5. **Фаронов В.В.** Программирование баз данных в Delphi 7: Учеб. курс. С.-Пб.: Питер, 2006. 458с.