

УДК 004.021

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО ПЕРЕДВИЖЕНИЯ С ДОСТАВКОЙ ГРУЗОВ

Н.А. ГУРЕЦКИЙ

(Представлено: С.А. ОЛЬШАНИКОВ)

Представлен практический способ отыскания оптимального пути передвижения с доставкой грузов. Цель работы – написание программы для расчета наиболее выгодной с точки зрения времени и затрат на выполнение работ очередности посещения адресов. Задача решалась путем рекурсивного перебора всех возможных вариантов посещения адресов и нахождения оптимального маршрута передвижения. Программа написана на языке программирования C++ с использованием Win32 API.

В транспортной логистике, особенно для курьерских компаний, распространена задача определения оптимальной последовательности посещения адресов. Строительные компании также заинтересованы в определении наименьшего пути, который нужно пройти на стройке (перевезти груз, закопать ямы и т.п.). Причем, зачастую, переместить груз из одной точки в другую за один раз не получается и приходится возвращаться. Тогда любой путь и конечные пункты можно представить в виде матрицы чисел, где положительные элементы – это откуда нужно что-то взять и сколько, отрицательные – куда положить и сколько, нулевые – те, которым «ничего не нужно».

В реализации данной программы заложен алгоритм построения кратчайшего по расстоянию маршрута, проходящего через заданные пункты. Алгоритм основан на полном переборе возможных вариантов маршрутов.

Для выявления оптимального пути необходимо изначально собрать растровые данные. Растровые данные – двумерная сетка числовых значений, представляющих некоторую измеренную характеристику, где пространственные данные поверхности спроектированы на плоскость сетки. Значения в каждой дискретной точке сетки, называемой пикселями, обычно интерпретируются как среднее над областью, спроектированной на каждую точку сетки (рис. 1). Другие интерпретации, конечно, возможны в зависимости от источника поступления данных о поверхности (сенсор, процесс ассимиляции или другие). Процесс сбора растровых данных – процедура простая, интуитивно понятная. Поскольку, исключая дискретизацию, такое представление данных аналогично построению географических карт с использованием линий уровня.

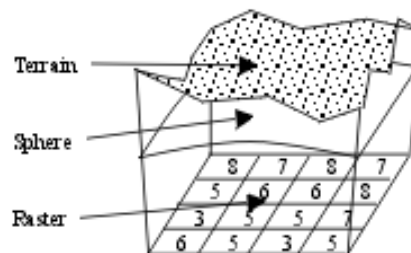


Рис. 4. Пример высот, представленных в виде растровых данных

На сегодняшний день известны несколько различных подходов к проблеме нахождения оптимальных путей через пространство. Мы попробуем составить дерево всевозможных решений. Подход, принятый в этой работе, должен упростить проблему поиска по дереву. Подобные, часто более простые версии этого подхода, используются разработчиками компьютерных игр для перемещения модулей по небольшим картам.

В самом начале перемещения для землеперемещающей машины область поверхности не имеет границ, то есть может перемещаться фактически бесконечно во всех направлениях. Ясно, что мы должны ограничить себя несколько меньшей областью, чтобы выполнить поиск пути на компьютере. Кроме того, мы хотели бы, чтобы эта область была как можно меньше без того, чтобы подвергать сомнению оптимальность пути. К тому же мы хотим уменьшить использование памяти. Ограничение области для движения машины будет иметь положительный эффект, связанный с уменьшением числа возможных путей, которые мы потенциально должны исследовать при поиске оптимального.

Дискретизируем бесконечное число позиций в непрерывном пространстве на конечное число точек. Чтобы облегчить представление, при дискретизации будем назначать эти точки в регулярной сетке. Область пространства, называемая ячейкой, закрывает каждую точку. Все ячейки не пересекаются и объединение всех ячеек должно закрыть все пространство. Движение нашего транспортного средства ограничено перемещением по «прямым линиям» между вершинами. Сетка может иметь любой вид, подходящий к заданному пространству и приложению при анализе. Два примера плоских сеток даны на рисунках 2 и 3.

Для нашего приложения использована регулярная «квадратная» плоская сетка. Это удобно, так как затем можно выбирать ячейки, которые точно совпадают с пикселями наших растровых данных.

Следующий шаг должен ограничить число возможных перемещений из данной ячейки. Если ячейка положительная, то мы ищем отрицательную ячейку, чтобы переместить груз. Иначе, если ячейка отрицательная, либо равна нулю, то мы просто перемещаемся в положительную ячейку. При этом каждый раз вычисляется расстояние до конечной клетки, которое помещается в результат.

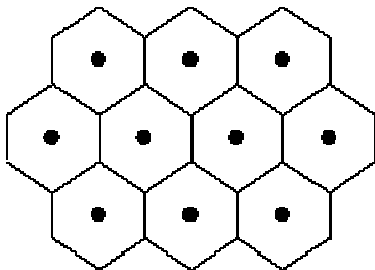


Рис. 5. Регулярная гексагональная плоская сетка

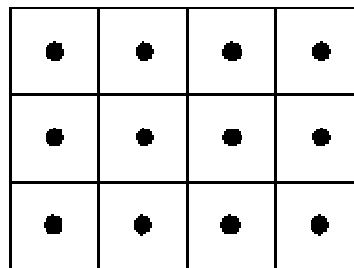


Рис. 6. Регулярная квадратная плоская сетка

Для определения оптимальности маршрута будем использовать следующие критерии:

1. Общая длина пройденного пути.
2. Длина, пройденная пустым (из отрицательной ячейки, либо ячейки равной нулю к положительной).
3. Длина груженым (из положительной ячейки в отрицательную).

В качестве языка программирования выбран язык C++, т.к. он является одним из самых быстрых, в плане вычислительных операций, языков.

Для составления дерева решений использована рекурсия. Конечно, все дерево решений не хранится, т.к. это огромные затраты оперативной памяти. Сохраняем лишь 20 наиболее оптимальных маршрутов. Стоит отметить, что при рекурсии идет огромное потребление оперативной памяти, но сокращается время работы. Поскольку всевозможных вариантов очень много (зависит от растровых данных), то целесообразно пустить эту функцию в отдельный поток, чтобы избежать зависания основного потока приложения.

На данный момент программа работает следующим образом: вводится размер матрицы и загружается текстовый файл с исходными растровыми данными. Растровые данные являются целыми числами. В результате работы программа выдает отчет, в котором сформирована наиболее оптимальная последовательность посещения адресов-ячеек. Оптимальный путь может быть и не один, поэтому пользователь может просмотреть отчет о 20 наиболее оптимальных маршрутах. Перемещения через точки можно наглядно отследить через интерфейс, написанный на Windows API [1],[2].

В результате данной работы решена задача определения оптимального маршрута при перевозке грузов и разработано актуальное программное обеспечение для ее решения, имеющее перспективы коммерческого использования. На данный момент ведутся работы по увеличению функционала программы.

ЛИТЕРАТУРА

1. Windows API. [Электронный ресурс] – Режим доступа https://ru.wikipedia.org/wiki/Windows_API . Дата обращения 10.04.2015.
2. Windows API. [Электронный ресурс] – Режим доступа <https://msdn.microsoft.com/en-us/library/cc433218>. Дата обращения 11.04.2015.

УДК 004.021

ОПТИМИЗАЦИЯ ПРОИЗВОДСТВЕННОЙ ЛОГИСТИКИ В СТРОИТЕЛЬСТВЕ НА ПРИМЕРЕ РАБОТЫ ЗЕМЛЕПЕРЕМЕЩАЮЩИХ МАШИН

Н.А. ГУРЕЦКИЙ

(Представлено: С.А. ОЛЬШАНИКОВ)

Затронута важная задача оптимизации при проектировании организационной части строительного производства. Понятно, что решение сложной оптимизационной задачи с множеством переменных невозможно без использования программных средств. Задача решена путем рекурсивного перебора всех возможных вариантов перемещения и нахождения оптимального маршрута. Если опти-