МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Реферат №\_\_3\_\_**

по дисциплине«Корпоративные системы управления»

Тема: «Маршрутизация одного транспортного средства от пункта отправления до пункта назначения»

ИСПОЛНИТЕЛЬ: \_\_Журавлев Н. В.\_\_

ФИО

группа ИУ5-34М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

ИСПОЛНИТЕЛЬ: \_\_Клюкин \_Н. А.\_\_

ФИО

группа ИУ5-35М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"07"\_\_декабря\_\_\_2024 г.

ПРЕПОДАВАТЕЛЬ: \_\_\_\_Сухобоков А.В.\_\_\_\_\_

ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

""\_декабря\_\_\_2024 г.

Москва - 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Оглавление**

[Введение 3](#_Toc184475365)

[Приложения кратчайшего пути 3](#_Toc184475366)

[Инструкция по использованию 3](#_Toc184475367)

[Замена оборудования 6](#_Toc184475368)

[Задача нахождения кратчайшего пути (SPP) 7](#_Toc184475369)

[Формулировка кратчайшего пути 8](#_Toc184475370)

[Алгоритм кратчайшего пути Дейкстры 9](#_Toc184475371)

[Описание алгоритма 9](#_Toc184475372)

[Иллюстрация алгоритма кратчайшего пути 10](#_Toc184475373)

[Характеристики алгоритма SPP Дейкстры 11](#_Toc184475374)

[Пример алгоритма SPP2 13](#_Toc184475375)

[Варианты задачи поиска кратчайшего пути (SPP) 15](#_Toc184475376)

[Формулировка минимальной стоимости сетевого потока SPP 20](#_Toc184475377)

[Решение с помощью Excel Shortest Path Sheets и Solver 22](#_Toc184475378)

[Решение с помощью математического языка программирования AMPL 27](#_Toc184475379)

[Решение SPP-формулировки в формате LP 32](#_Toc184475380)

# Введение

Эту задачу чаще называют задачей кратчайшего пути (SPP). Следует отметить, что задача разбиения множества — это еще одна часто используемая задача в транспортных моделях, имеющая идентичную аббревиатуру, поэтому аббревиатуру SPP следует интерпретировать осторожно. Задача разбиения множества обсуждается в рамках маршрутизации туда и обратно с одним и несколькими транспортными средствами.

# Приложения кратчайшего пути

Инструкция по использованию

Вы можете запросить маршрут проезда между любыми двумя адресами в континентальной части Соединенных Штатов с различных веб-сайтов. Чтобы ответить на ваш запрос, программное обеспечение должно найти кратчайший путь между двумя точками на базовой уличной сети в Соединенных Штатах. Уличная сеть основана на файлах TIGER, которые публикуются Бюро переписи населения США и полностью пересматриваются каждые 10 лет на ротационной основе для различных областей страны. Маршруты проезда, созданные популярным приложением для персональных компьютеров для поездки от государственной дороги Georgia 400 до аэропорта Атланты, показаны в графическом и текстовом формате на рис. 1 и 2.

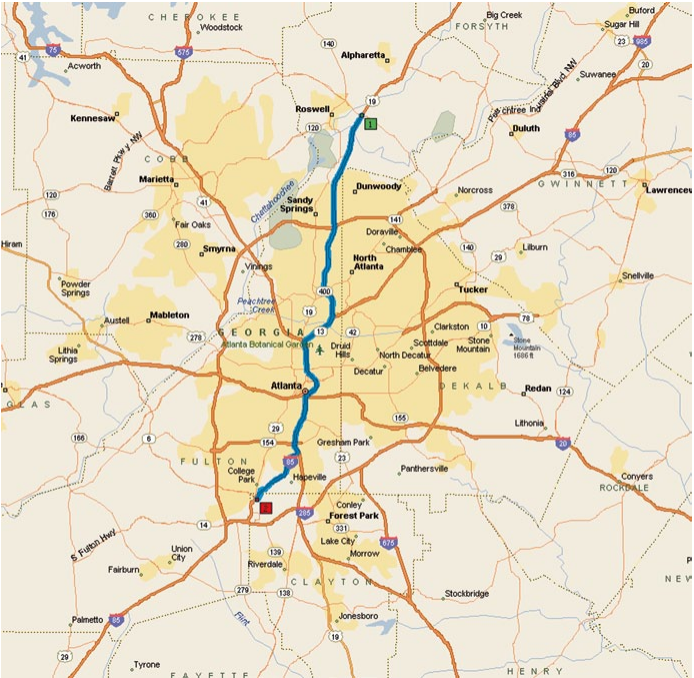


Рисунок 1. Иллюстрация программного обеспечения для планирования. (Карта создана Microsoft MapPoint 2010)

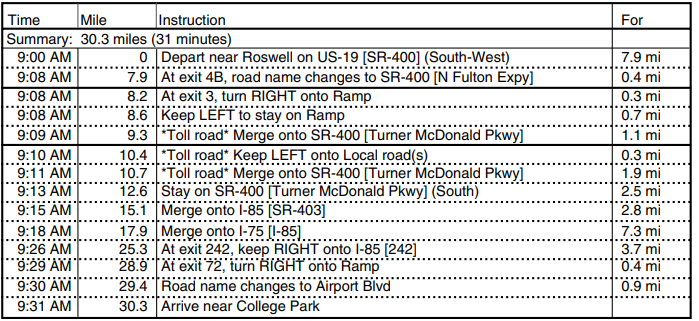


Рисунок 2. Инструкции по планированию маршрута вождения

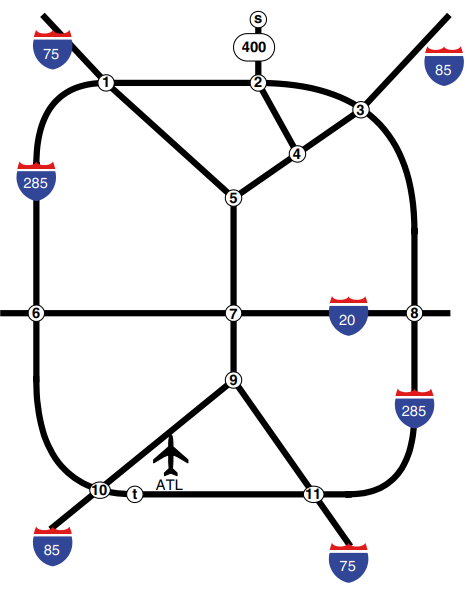


Рисунок 3. Преобразование сети основных дорог в абстрактную сеть

Концептуально физическая дорожная сеть абстрагируется до математической сети, где перекрестки дорог соответствуют узлам в сети, а сегменты дорог между перекрестками соответствуют ребрам в сети. На рис. 3 показана сеть, основанная на межштатных автомагистралях и некоторых основных дорогах штата для столичного региона Атланты. Ее можно использовать для планирования поездки из начальной точки в северных пригородах Атланты, обозначенной начальным узлом s, в конечный узел, соответствующий аэропорту Атланты, обозначенный конечным узлом t. Пример сети состоит из 13 узлов и 25 ребер, но реальная сеть будет содержать гораздо больше узлов и ребер.

Поиск кратчайшего пути или направлений между двумя местоположениями является типичным примером подхода к инженерному моделированию. Карта и начальное и конечное местоположения соответствуют данным. Абстрактная дорожная сеть с ее ребрами, перекрестками и длинами проезда формирует информацию. Наконец, после выполнения алгоритма решения становится известен кратчайший путь между пунктами отправления и назначения, что соответствует знанию о том, как проехать кратчайшее расстояние от пункта отправления до пункта назначения.

Замена оборудования

Типичное решение, которое необходимо принять при планировании замены оборудования, — когда заменить конкретное транспортное средство или машину на более новую модель. Это решение обычно основывается на компромиссе между стоимостью лизинга и стоимостью обслуживания, где предполагается, что более новое оборудование будет иметь более высокую стоимость лизинга, но более низкую стоимость обслуживания, а бывшее в употреблении оборудование будет иметь противоположные стоимостные характеристики. Также предполагается, что транспортное средство должно быть доступно в течение всего горизонта планирования.

Соответствующая сеть имеет узел для каждого периода времени в течение горизонта планирования. Каждое решение о развертывании машины можно представить в виде дуги от начального периода до конечного периода использования этой машины. Длина дуги равна сумме расходов на лизинг и обслуживание за соответствующий срок службы транспортного средства, дисконтированных до настоящего времени. Затем можно найти график замены оборудования с наименьшей стоимостью как кратчайший путь от начального узла до конечного узла горизонта планирования.

# Задача нахождения кратчайшего пути (SPP)

Во всех приведенных выше примерах и приложениях реальная задача планирования сводится к абстрактной сети, состоящей из узлов и ненаправленных ребер или направленных дуг. Транспортировка может осуществляться только по сетевым связям между сетевыми узлами, и путешествие начинается в начальном узле s и заканчивается в конечном узле t. Сетевые связи имеют длину, соответствующую реальной длине путешествия, длительности путешествия или стоимости. Цель состоит в том, чтобы найти кратчайший путь от начального узла s до конечного узла t.

При решении реальной задачи с помощью SPP необходимо выполнить две основные задачи. Первая задача — абстрагирование реальной задачи до SPP. Эта задача требует идентификации реальных сущностей, соответствующих узлам и ребрам сети, и определения метода вычисления стоимости для стоимости на каждом ребре. Вторая задача — решение абстрактной SPP. Эта задача требует идентификации надлежащего варианта SPP и алгоритма решения, наиболее подходящего для этого варианта. Если реализация алгоритма недоступна, алгоритм необходимо реализовать и протестировать. Наконец, необходимо решить экземпляр проблемы и проверить полученное решение. Первая задача специфична для каждой задачи и отличается для каждой задачи, а также требует понимания проблемы и методологии инженерного проектирования. Вторая задача является общей, поскольку она выполняется для стандартизированной формулировки проблемы. Для нее требуются методы из компьютерной науки и разработки программного обеспечения. Алгоритмы решения описаны в литературе, и доступны стандартные реализации решателя.

Формулировка кратчайшего пути

SPP можно рассматривать как поиск пути с минимальной стоимостью для отправки одной единицы потока из исходного узла в узел назначения. Таким образом, SPP является подклассом задачи потока сети с минимальной стоимостью, или MCNFP. Следующая нотация была введена для описания задач потока сети с минимальной стоимостью. Цель состоит в том, чтобы найти набор потоков с минимальной стоимостью (x), которые удовлетворяют требованиям внешнего потока (b). Также существует одно ограничение для каждого узла, которое обеспечивает сохранение потока в этом узле.

Поток на направленной дуге от узла i к узлу j

Стоимость единицы для одной единицы потока, транспортируемой из узла i в узел j

Внешний поток для узла i (1 для исходного узла, −1 для узла назначения, ноль для промежуточных узлов)

**Формула 1.** **Формулировка кратчайшего пути как формулировка потока сети с минимальной стоимостью**

Поскольку SPP является подзадачей MCNFP, ее можно решить с помощью стандартных алгоритмов потока сети. Это будет подробно обсуждаться ниже. Однако из-за своей простой структуры SPP можно очень эффективно решить с помощью жадного алгоритма при условии, что ни одна из стоимостей ребер не является отрицательной. Первоначально этот алгоритм был предложен Дейкстрой (1959).

# Алгоритм кратчайшего пути Дейкстры

Описание алгоритма

Алгоритмы определения кратчайшего пути основаны на понятии одной или нескольких меток, связанных с каждым узлом. В начале алгоритма все метки называются временными. Временные метки являются верхними границами длины кратчайшего пути от начального узла до соответствующего узла. Алгоритмы установки меток преобразуют одну временную метку в постоянную метку во время каждой итерации. Постоянная метка является точной длины кратчайшего пути от начального узла до соответствующего узла. Алгоритм Дейкстры относится к этому классу алгоритмов установки меток. В описании алгоритма Дейкстры используются следующие обозначения:

N, D Набор всех узлов и узлов назначения соответственно

T, P Набор узлов с временными и постоянными метками соответственно

l(k) Метка узла k

Длина ребра между узлами k и j

Г(k) Набор узлов-последователей узла k, также называемый прямой звездой узла k, который является набором узлов, для которых существует ребро или дуга от узла k до этого узла

pred(k) Узел-предшественник на текущем кратчайшем пути к узлу k

Алгоритм имеет фазу инициализации, а затем основную итеративную фазу. На фазе инициализации все метки узлов устанавливаются на бесконечность и устанавливаются как временные. Метка исходного узла s устанавливается на ноль и устанавливается как временные. Основная фаза итеративно выполняет следующие четыре шага.

* + - 1. Найдите узел p с минимальной временной меткой
      2. Для всех узлов-последователей узла p с временными метками обновите их метки, если необходимо, чтобы указать сокращенную длину кратчайшего пути к этим узлам. Формула обновления для временных меток:

Если временная метка была сокращена, то узел-предшественник на текущем кратчайшем пути к узлу x устанавливается на узел p, т. е. pred(x) = p

* + - 1. Поскольку узел p является узлом с наименьшей временной меткой, а все стоимости ребер неотрицательны, более короткого пути к узлу p существовать не может. Таким образом, l(p) является точной длиной кратчайшего пути к узлу p, а метка узла p может быть помечена как постоянная
      2. Если все узлы назначения имеют постоянные метки, остановитесь, иначе перейдите к шагу 1

Алгоритм Дейкстры можно очень компактно описать в псевдокоде. Такое описание полезно как механизм коммуникации между аналитиком по исследованию операций и программистом по информатике, поскольку оно описывает все шаги алгоритма однозначно и полностью, но без описания низкоуровневых деталей, шагов выполнения и структур данных.

**Алгоритм 1. Алгоритм кратчайшего пути Дейкстры для плотных графов в псевдокоде**

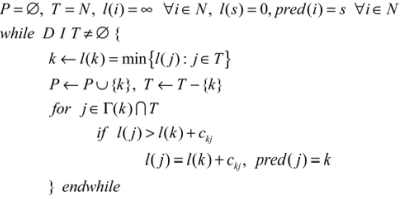


Иллюстрация алгоритма кратчайшего пути

Алгоритм SPP проиллюстрирован в следующей небольшой сети, показанной на рис. 4 и 5. Число на каждом ребре указывает длину или стоимость этого ребра. Выноска для каждого узла дает метку этого узла и его предшествующего узла, если он был определен. Первый рисунок показывает сеть после фазы инициализации, второй рисунок показывает сеть после завершения алгоритма. Длина кратчайшего пути к узлу t равна 12, и он достигает узла t из узла 3. Затем полный кратчайший путь можно найти путем обратной трассировки через предшествующие узлы. Например, кратчайший путь к узлу 3 имеет длину 9, и он достигает узла 3 из узла 1. Кратчайший путь показан жирными ребрами во второй сети.

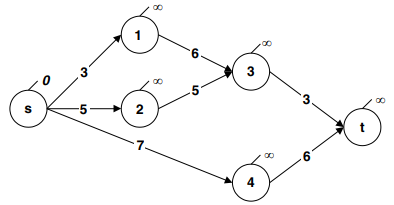


Рисунок 4. Задача о кратчайшем пути SPP1

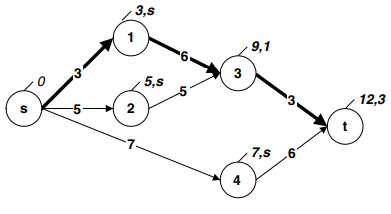


Рисунок 5. Решение задачи о кратчайшем пути SPP1

Характеристики алгоритма SPP Дейкстры

Алгоритм Дейкстры — классический пример прямого динамического программирования. Он основывает оптимальность своего конечного пути на принципе оптимальности и свойстве неотрицательных длин дуг или ребер. Результатом алгоритма является направленное дерево с корнем в исходном узле s.

С точки зрения вычислений самым дорогим шагом является идентификация узла с наименьшей временной меткой. Нахождение наименьшей временной метки включает проверку максимум n узлов, и этот процесс должен быть повторен для n итераций. Следовательно, общая сложность алгоритма составляет O() для этой реализации.

Максимальный размер задачи, который может быть решен за разумное время, зависит от скорости компьютера, но размеры задач от 100 000 до 1 000 000 узлов осуществимы на современных персональных компьютерах.

Плотность графа — это количество ребер или дуг, присутствующих в графе, деленное на общее количество возможных ребер или дуг, которое равно n(n−1) для ребер и 2n(n−1) для дуг. Если каждый узел соединен с каждым другим узлом, граф называется полным графом, а его плотность равна единице. Таким образом, плотность графа — это число от 0 до 1. Описанная выше реализация эффективна только для графов с высокой плотностью. Граф называется разреженным, если он имеет низкую плотность. Для разреженных графов приведенная выше реализация неэффективна. Предположим, что временные метки хранятся в упорядоченной последовательности по возрастанию значения. Поскольку во время каждой итерации могут измениться только несколько временных меток, упорядоченная последовательность временных меток может измениться только в нескольких местах. Было бы неэффективно проверять всю последовательность во время каждой итерации, чтобы найти минимальную временную метку. Для разреженных графов эффективнее хранить и поддерживать временные метки в структуре кучи. Затем поиск временной метки с наименьшим значением включает в себя простое удаление метки в верхней части кучи. Существует несколько реализаций кучи, таких как двоичная, Фибоначчи и радиксная куча. Вычислительная сложность для алгоритма SPP составляет O(m + n · log n) для кучи Фибоначчи и O((m + n) · log n) для двоичной кучи. Сокращенное время выполнения должно быть сбалансировано с существенно более длительным временем реализации. Два основных преимущества алгоритма Дейкстры — это его простые структуры данных и выполнения, в то время как реализации кучи значительно более сложны. Даже при использовании стандартной библиотеки типов (STL) для реализации кучи программы, основанные на имплантации кучи, будут в сотни раз больше и сложнее, чем алгоритм Дейкстры. Соответствующие усилия по реализации оправданы только в том случае, если SPP будет решаться неоднократно или должен решаться для чрезвычайно больших экземпляров задач. Алгоритм Дейкстры не будет хорошим выбором для поиска кратчайшего пути по дорожной сети для определения направлений движения между двумя адресами или местоположениями.

Пример алгоритма SPP2

Далее приведен несколько более крупный пример SPP. Таблица 1 показывает асимметричные расстояния между узлами, а рис. 6 показывает сеть с данными. Рисунок 7 показывает решение SPP2 от узла 1 до узла 4.

Таблица 1. Пример кратчайшего пути. Матрица расстояний SPP2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| From/to | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 |  | 10 |  |  |  |  | 3 | 6 | 12 |
| 2 | 10 |  | 18 |  |  |  | 2 |  | 13 |
| 3 |  | 18 |  | 25 |  | 20 |  |  |  |
| 4 |  |  | 25 |  | 5 | 16 | 4 |  |  |
| 5 |  |  |  | 5 |  | 10 |  |  |  |
| 6 |  |  | 20 |  | 10 |  | 14 | 15 | 9 |
| 7 |  | 2 |  | 4 |  | 14 |  |  | 24 |
| 8 | 6 |  |  |  | 23 | 15 |  |  | 5 |
| 9 | 12 | 13 |  |  |  | 9 | 24 | 5 |  |

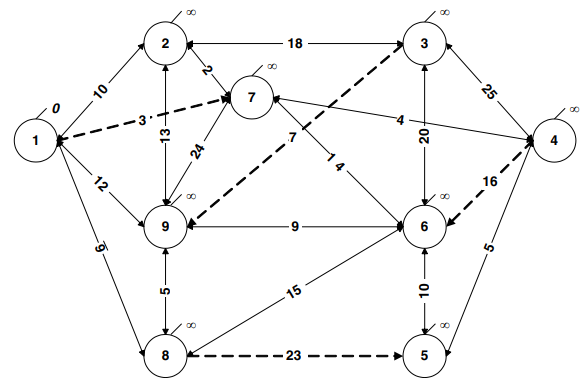


Рисунок 6. Пример кратчайшего пути в сети SPP2

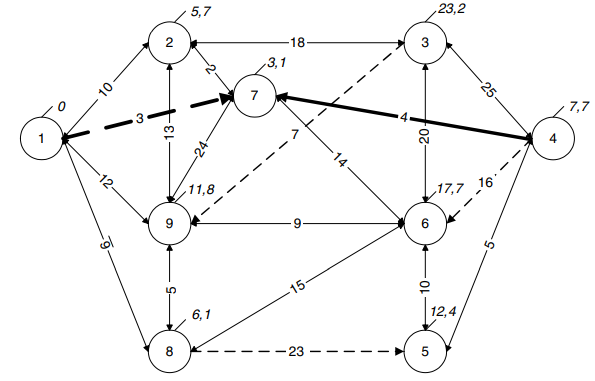


Рисунок 7. Пример решения SPP2 с кратчайшим путем

Сплошные ребра обозначают двунаправленные соединители с одинаковыми симметричными расстояниями в обоих направлениях, пунктирные ребра обозначают асимметричные однонаправленные соединители.

Из-за теоретического интереса и простоты SPP и алгоритма Дейкстры для него, он широко обсуждался в справочных публикациях и в Интернете. Среди прочего, Атанасиос (2006) создал JAVA-апплет, реализующий алгоритм кратчайшего пути Дейкстры, который прекрасно иллюстрирует прогрессию динамического программирования. Этот апплет находится на личной домашней странице этого студента по информатике, поэтому апплет может быть недоступен в будущем.

Варианты задачи поиска кратчайшего пути (SPP)

Большое количество приложений можно абстрагировать либо к классическому SPP, либо к одному из его вариантов. Некоторые из наиболее часто используемых вариантов описаны далее.

Один источник к одному приемнику (s к t)

Это классический и оригинальный вариант, описанный выше. Дейкстра (1959) был одним из первых, кто предложил оптимальный алгоритм для решения этой проблемы, основанный на прямом динамическом программировании. Каждый узел имеет одну метку. Все узлы изначально имеют временную метку. Алгоритм преобразует во время каждой итерации метку одного узла из временной в постоянную. Этот классический алгоритм прост в реализации и относительно эффективен, так что задачи, которые имеют до 1 000 000 узлов, могут быть решены за разумное время на современных компьютерах.

Один источник ко всем приемникам (s к всем)

Этот вариант определяет кратчайший путь от одного источника ко всем возможным узлам назначения. При решении с помощью алгоритма Дейкстры эта задача требует очень мало дополнительных вычислений. По сути, вычисления продолжаются после того, как метка первого узла назначения станет постоянной, пока метки всех узлов назначения не станут постоянными.

Все пары

Этот вариант определяет кратчайший путь от всех исходных узлов до всех возможных узлов назначения.

k кратчайших путей

Этот вариант определяет не только кратчайший путь между исходным и конечным узлом, но и следующие k−1 кратчайших путей. Он в основном используется, когда кто-то заинтересован в выполнении анализа чувствительности на кратчайшем пути, поскольку инкрементная стоимость или сожаление между первым и вторым кратчайшим путем к узлу будут известны по завершении алгоритма.

Общие затраты

Все описанные выше варианты основаны на свойстве, что не существует ребра или дуги с отрицательной стоимостью. Это подразумевает, что кумулятивная стоимость вдоль ребер пути не уменьшается. Алгоритмы, используемые для решения этих задач, относятся к классу алгоритмов установки меток. Если существуют ребра с отрицательной стоимостью, говорят, что проблема имеет общие затраты. Алгоритмы, используемые для решения проблемы общей стоимости, относятся к классу алгоритмов коррекции меток. Эти алгоритмы намного менее эффективны, чем алгоритмы установки меток, поэтому гораздо меньшие задачи могут быть решены с оптимальностью. Все метки преобразуются из временных в постоянные на последней итерации. Наконец, если существует монотонно неубывающая переменная статуса вдоль пути, то допускаются отрицательные дуговые издержки. Каждый узел больше не будет иметь одну метку, но может иметь несколько меток в зависимости от неубывающей переменной статуса. Во время каждой итерации одна метка узла, но не сам узел, преобразуется из временной в постоянную. Во время выполнения алгоритма узел будет иметь несколько меток, некоторые из которых могут быть постоянными, а некоторые — временными. Примером этого варианта с общими издержками является подзадача ценообразования формулировки разбиения множеств и алгоритма для маршрутизации транспортных средств в оба конца. В этом примере неубывающая переменная статуса — это совокупное количество товаров, которые должно доставить транспортное средство по его маршруту. Когда транспортное средство посещает узел, количество товаров, доставленных транспортным средством, всегда увеличивается, а общее количество ограничено вместимостью транспортного средства, поэтому количество узлов, посещенных на маршруте, ограничено. Этот вариант подробно обсуждается далее в алгоритме задачи разбиения множеств (SPP) для задач маршрутизации транспортных средств (VRP).

Самый длинный путь в ациклических графах

Во всех вариантах, описанных до сих пор, целью было найти кратчайший путь между исходным и конечным узлами. В управлении проектами часто интересуются поиском минимальной общей продолжительности проекта. Проектная сеть — это тип сети, используемый для представления действий и событий в планировании и управлении проектами. Узел в сети соответствует событию, а дуга — действию или задаче. Возможными событиями являются (1) начало проекта, (2) завершение проекта и (3) завершение одной или нескольких задач или действий, которые должны быть выполнены в проекте 6 Single Flow Routing Through a Network. Действия могут иметь отношения предшествования, т. е. одно действие не может быть начато, пока не будут завершены одно или несколько других действий. Существование дуги от узла i до узла j подразумевает, что задача i должна быть завершена до того, как может быть начата задача j, а длина дуги — это длительность задачи. Путь — это последовательность действий, удовлетворяющая ограничениям предшествования, т. е. допустимая последовательность. Задачи на самом длинном пути от начального до конечного узла называются критическими, поскольку они определяют минимальную общую длину проекта. Задачи, которые не находятся на критическом пути, могут иметь некоторое резервное время, и сокращение продолжительности этих задач не сократит общую продолжительность проекта. Два исторических метода управления проектами, которые используют алгоритм самого длинного пути, — это метод критического пути (CPM) и метод оценки и обзора программы (PERT). PERT был впервые использован в 1950-х годах для управления процессом строительства Polaris, первой атомной баллистической подводной лодки ВМС США. PERT используется для управления крупномасштабными и сложными проектами. CPM использовался компанией Dupont для управления проектами промышленного строительства. CPM широко использовался как в бумажно-карандашной реализации, так и в современном программном обеспечении для управления проектами, таком как Microsoft Project, для управления проектами в строительстве, разработке продуктов и разработке программного обеспечения.

Обратите внимание, что стандартный алгоритм кратчайшего пути Дейкстры необходимо модифицировать двумя способами. Во-первых, оператор максимума используется для определения новой метки последующих узлов. Во-вторых, если узел-последователь имел постоянную метку и значение этой метки увеличивается, т. е. происходит чистое изменение, то метка этого узла-последователя должна быть снова сделана временной. Модифицированный алгоритм Дейкстры для поиска самого длинного пути показан ниже в псевдокоде. Были разработаны другие алгоритмы для определения самого длинного пути, но указанная выше модификация очень проста в реализации, хотя вычислительная сложность увеличивается по сравнению со стандартным алгоритмом Дейкстры. Обратите внимание, что если в графе существует направленный цикл, алгоритм никогда не прекратит выполнение. Наличие направленного цикла указывает на то, что нет приемлемого решения проблемы управления проектами, поскольку существуют по крайней мере две задачи, которые являются взаимозависимыми, т. е. каждая задача должна быть завершена до того, как сможет начаться другая задача.

**Алгоритм 6.2. Алгоритм нахождения самого длинного пути для плотных графов в псевдокоде**

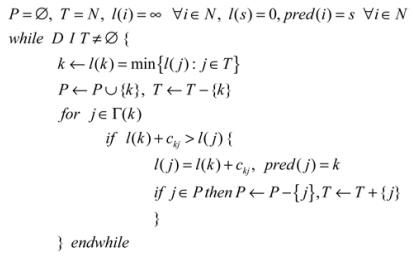


Диаграмма перехода состояний

Часто проблемы можно свести к поиску оптимального пути через диаграмму перехода состояний. Диаграмма перехода состояний — это сеть, где каждый узел соответствует допустимому состоянию системы, а дуги соответствуют допустимым переходам из одного состояния в другое. Оптимальный путь может быть кратчайшим или длиннейшим путем от начального состояния до конечного состояния. Сведение проблемы включает определение краткой и полной характеристики состояния системы и определение всех допустимых состояний и допустимых переходов.

Рассмотрим классическую головоломку с тремя кувшинами для воды. 8-галлонный кувшин наполнен водой. Кроме того, даны один пустой 5-галлонный и один пустой 3-галлонный кувшины. Ваша задача — найти последовательность переливаний, чтобы разделить 8 галлонов воды на две равные части по 4 галлона, используя только три контейнера. Никакие другие измерительные приборы не допускаются. Какое наименьшее количество переливаний или переливаний необходимо для достижения этого разделения.

Один из методов решения состоит в построении диаграммы переходов состояний для этой задачи и нахождении кратчайшего пути через сеть, где длина каждой дуги равна единице. Узлы сети соответствуют всем возможным состояниям воды в контейнерах. Дуги соответствуют возможным переходам между этими состояниями. Состояние системы кодируется в 3-кортеже; соответствующем количеству воды в 8-, 5- и 3-галлонном кувшине соответственно. Начальный узел обозначен как (8,0,0), а конечный узел обозначен как (4,4,0), поскольку только два самых больших контейнера могут вместить 4 галлона. Из начального узла можно достичь только двух возможных узлов, а именно (3,5,0) и (5,0,3).

Соответствующая сеть со всеми возможными состояниями и возможными переходами показана на рис. 8. Дуги изображены пунктирными линиями, а их направление обозначено стрелкой. Двунаправленные ребра изображены сплошными линиями. Выполнение ребра более одного раза только увеличит общее количество заливок без достижения нового состояния, поэтому в оптимальном плане заливки ни одно ребро не будет выполнено более одного раза. Индекс каждого возможного состояния показан в круге рядом с узлом. Исходя из этой сети, наименьшее количество заливок, необходимое для разделения воды пополам, равно семи.

# Формулировка минимальной стоимости сетевого потока SPP

Ранее SPP была сформулирована как задача минимальной стоимости сетевого потока (MCNFP) в (1). SPP эквивалентна нахождению пути потока с минимальной стоимостью от исходного узла s до конечного узла t для одной единицы потока. Внешний приток (предложение) в узле s, таким образом, равен единице, а внешний отток (спрос) в узле t отрицателен, и все остальные узлы не имеют внешних потоков. Переменные потока никогда не будут больше единицы, поскольку единственный требуемый поток наружу равен единице. Следовательно, нет никаких верхних или нижних границ, кроме ограничений неотрицательности.

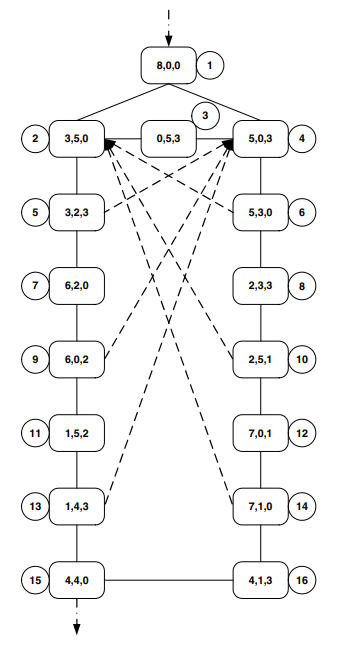
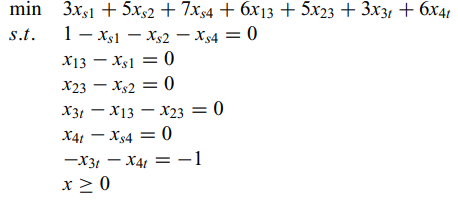


Рисунок 8. Сеть головоломок по заливке воды

Полная математическая формулировка для задачи кратчайшего пути, показанной на рис. 4, приведена ниже. В SPP есть шесть узлов и семь дуг. Соответствующая MCNFP имеет, таким образом, семь переменных потока и шесть ограничений сохранения потока.



Формулировки MCNFP с одним продуктом могут быть очень эффективно решены с помощью различных решателей линейного программирования (ЛП), поскольку базис может храниться очень компактно, а обновление базиса не требует обращения матрицы.

Решение с помощью Excel Shortest Path Sheets и Solver

Решение примера задачи, изображенного на рис. 6, с решателем, включенным в Microsoft Excel, проиллюстрировано на следующих восьми рисунках. Первые четыре рисунка показывают данные и статус решения до выполнения решателя. Далее показаны два диалоговых окна для выполнения решателя. Наконец, решение показано на последних двух рисунках.

Рисунок 9 показывает структуру сети через двумерную матрицу смежности. Матрица имеет строку для каждого исходного узла и столбец для каждого конечного узла. Если в сети существует дуга от исходного до конечного узла, пропускная способность этой дуги указывается в соответствующей ячейке. Поскольку в формулировке MCNFP для SPP транспортируется только единица с одним потоком, все пропускные способности в этом случае можно установить равными единице. Следует отметить, что эта матрица смежности асимметрична, чтобы соответствовать асимметричной структуре сети. Например, имеется пропускная способность от узла 4 до узла 6, но соответствующая пропускная способность от узла 6 до узла 4 отсутствует.

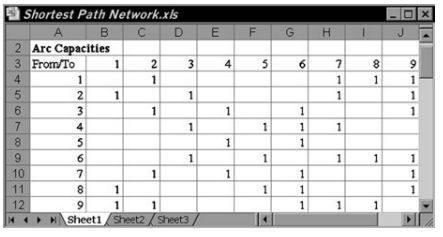


Рисунок 9. Электронная таблица SPP2 Excel, дуговые емкости

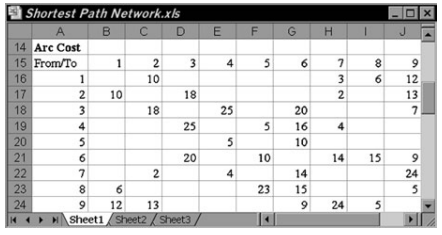


Рисунок 10. SPP2 Excel таблица дуговых расходов

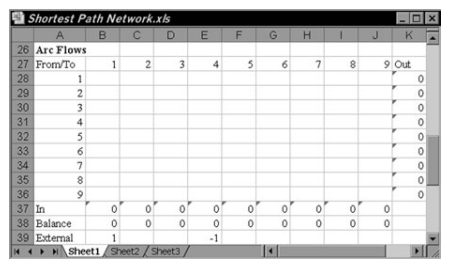


Рисунок 11. SPP2 таблица Excel начальный баланс потока

Также следует отметить, что представление матрицы смежности сети является очень простым и интуитивно понятным методом представления, который хорошо совместим с программным обеспечением для электронных таблиц. Однако он крайне неэффективен с точки зрения объема памяти и времени выполнения для больших экземпляров задач и разреженных сетей.

Рисунок 10 показывает стоимость дуг в той же двумерной структуре матрицы смежности.

Рисунок 6.11 показывает раздел электронной таблицы, соответствующий переменным потока. Он имеет ту же двумерную структуру матрицы смежности, дополненную строками и столбцами для проверки сохранения потока. Для каждого исходного узла сумма исходящих потоков вычисляется в столбце K, а для каждого узла назначения сумма всех входящих потоков вычисляется в строке 37. Разница между исходящими и входящими потоками вычисляется в строке 38. Учитывая соглашение о знаках для входящих и исходящих потоков, разница в строке 38 вычисляется как сумма исходящих потоков за вычетом суммы входящих потоков. Эта разница должна равняться внешним потокам для каждого узла, которые показаны в строке 39.

Рисунок 12 показывает вычисление целевой функции в той же двумерной структуре смежности. Каждая ячейка вычисляет произведение количества потока на стоимость единицы потока. Поскольку в это время все потоки все еще равны нулю, все элементы в матрице равны нулю. Сумма стоимостей потока для исходных узлов и узлов назначения вычисляется в столбце K и строке 52 соответственно. Затем в ячейке K52 вычисляется общая стоимость потока. Эту величину можно было бы вычислить напрямую с помощью функции Excel SUMPRODUCT, но структура матрицы смежности предоставляет дополнительную информацию о том, какие отдельные потоки и узлы вносят вклад в общую стоимость.

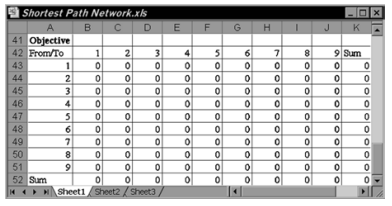


Рисунок 12. Начальная целевая функция таблицы Excel SPP2

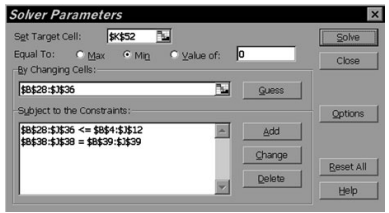


Рисунок 13. Параметры решателя таблиц Excel SPP2

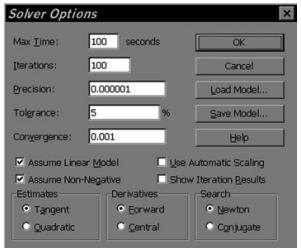


Рисунок 14. Параметры решателя таблиц Excel SPP2

На рисунках 13 и 14 показаны диалоговые окна, определяющие задачу и указывающие параметры управления решателя для этой проблемы сетевого потока. Цель состоит в том, чтобы минимизировать общую стоимость потока, вычисленную в ячейке K52 как сумму произведений значений потока и удельных затрат потока. Переменными решения являются переменные потока, хранящиеся в ячейках матрицы смежности B28:J36. В этой формулировке присутствуют два типа ограничений. Первый набор — это ограничения верхней границы для отдельных значений потока. Каждое значение потока должно быть меньше или равно соответствующей пропускной способности дуги. Это дает ограничение B28:J36≤B4:J12. Второй набор ограничений моделирует сохранение потока для каждого узла. Внутренний баланс потока должен быть равен внешним величинам потока. Это дает ограничение B38:J38=B39:J39.

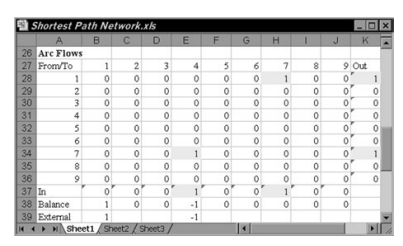


Рисунок 15. Потоки решения для электронных таблиц SPP2 Excel

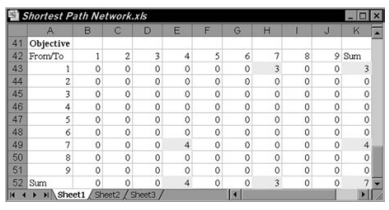


Рисунок 16. Целевая функция решения электронной таблицы SPP2 Excel

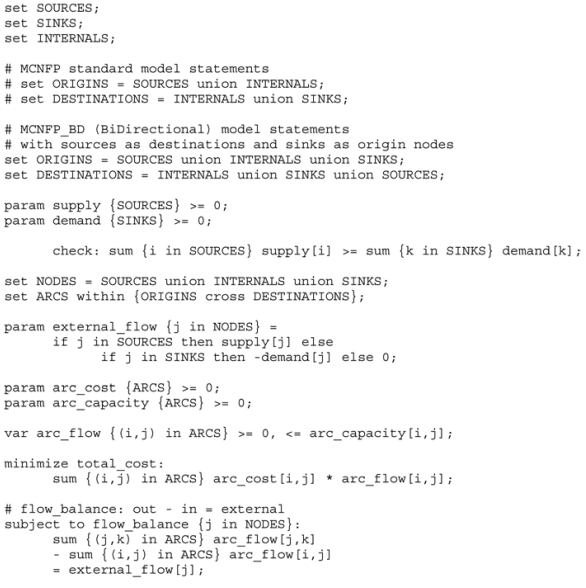
MCNFP имеет только неотрицательные переменные и линейные целевые и ограничения. Таким образом, ее можно решить с помощью алгоритма линейного программирования решателя. Два основных параметра управления в диалоговом окне «Параметры», которые необходимо указать, — это «Предположить линейную модель» и «Предположить неотрицательное». Решатель находит оптимальное решение для этого конкретного примера за долю секунды.

На рис. 15 и 16 показаны переменные потока решения и значения целевой функции. Ячейкам в электронной таблице назначено условное форматирование, поэтому они затеняются, если в них хранится ненулевое значение. Кратчайший путь от узла 1 до узла 4 проходит через узел 7, а длина кратчайшего пути равна 7. На втором рисунке подробно описаны стоимостные вклады отдельных транспортных перемещений.

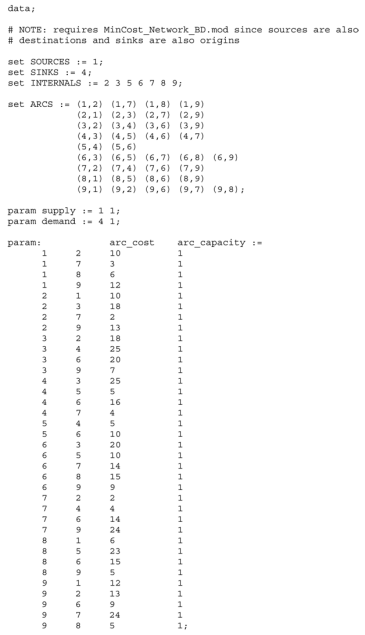
Решение с помощью математического языка программирования AMPL

Решение примера задачи, изображенного на рис. 6, с помощью математического языка программирования AMPL (A Math Programming Language) проиллюстрировано в следующих четырех листингах текстовых файлов. Листинги соответствуют файлам модели, данных, выполнения и выходных файлов соответственно. Первый список показывает модель, которая является общей формулировкой потока сети с минимальной стоимостью для одного товара (MCNFP). Эта модель также будет использоваться в следующей главе о маршрутизации нескольких потоков, но поскольку задача о кратчайшем пути является подклассом задачи потока сети с минимальной стоимостью, модель может быть использована здесь без каких-либо изменений. Существует множество альтернативных моделей AMPL для задачи потока сети с минимальной стоимостью, и некоторые из них можно найти на веб-сайте AMPL.

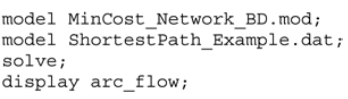
Листинг 1. Файл модели AMPL MCNFP



Листинг 2. Файл данных AMPL SPP2



Листинг 3. Файл исполнения AMPL SPP2



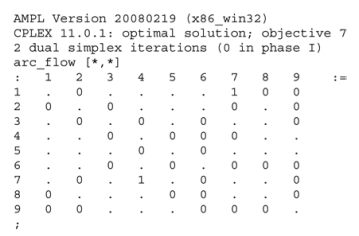
Модель содержит узлы и дуги. Узлы являются источниками, стоками или внутренними узлами в зависимости от того, есть ли у них внешний поток, входящий как предложение, внешний поток, выходящий как спрос, или нет внешнего потока соответственно. AMPL выполняет проверку осуществимости перед выполнением решателя, чтобы гарантировать, что общий объем предложения во всех исходных узлах больше общего спроса. Дуги определяются их исходными и конечными узлами. Исходные узлы дуг являются либо источниками, либо внутренними узлами. Если у стоковых узлов также есть исходящие дуги, то исходные узлы являются объединением всех трех типов узлов. Конечные узлы дуг являются либо стоковыми узлами, либо внутренними узлами. Если у исходных узлов также есть входящие дуги, то конечные узлы являются объединением всех трех типов узлов. Каждая дуга имеет стоимость, пропускную способность и поток дуги, все из которых должны быть неотрицательными. Поток дуги не должен превышать пропускную способность дуги. Цель этой формулировки — минимизировать общую стоимость транспортировки, которая является суммой произведений потока дуги и стоимости дуги по всем дугам. Наконец, для каждого узла существует сохранение ограничений потока, что гарантирует, что внешний поток равен разнице между исходящими и входящими потоками. Поставка моделируется с положительным внешним потоком, а спрос моделируется как отрицательный внешний поток.

Данные для примера кратчайшего пути показаны в листинге 6.2. Есть девять узлов, один исходный узел (1), один узел стока (4) и девять внутренних узлов. Поскольку многие дуги являются двунаправленными, исходный узел 1 и узел стока 4 являются одновременно внутренними узлами и исходным и стоковым узлом соответственно. Есть 36 однонаправленных дуг. Данные дуг приведены в конце листинга с исходным узлом, узлом назначения, стоимостью дуги и пропускной способностью дуги, перечисленными в одной строке для каждой дуги.

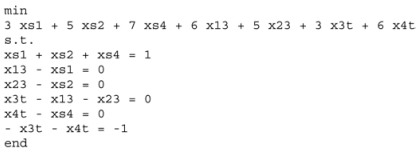
В листинге 3 показана последовательность команд в файле выполнения, который управляет выполнением программы AMPL. Этот файл выполнения или выполнения по соглашению имеет расширение «run». Имя файла выполнения является единственным аргументом командной строки для программы AMPL. Первая строка определяет файл модели, который по соглашению имеет расширение «mod». Для задачи поиска кратчайшего пути модель представляет собой стандартную модель сети с минимальной стоимостью. Для этого конкретного экземпляра задачи исходный узел также является узлом назначения, а узел стока также является узлом начала для нескольких двунаправленных дуг, поэтому необходимо использовать расширенный файл модели «MinCost\_Network\_BD.mod». Вторая строка определяет файл данных, который по соглашению имеет расширение «dat». Третья строка содержит команду «solve», которая заставляет модель решаться решателем по умолчанию. Наконец, четвертая строка отображает потоки в решении. Дополнительную информацию о языке AMPL можно найти в Fourer et al. (2002) и на веб-сайте AMPL Optimization LLC.

Наконец, в листинге 4 показан выходной журнал, сгенерированный при выполнении файла запуска с помощью программы AMPL. В этом случае решателем был CPLEX. Оптимальное значение найденного решения было равно 7, и на дуге (1,7) и (7,1) было два ненулевых потока. Это решение идентично решению, найденному решателем Excel и ручным выполнением алгоритма Дейкстры. В листинге показаны потоки дуг в формате вывода по умолчанию. Формат вывода можно настроить с помощью языка AMPL.

Листинг 4. Журнал выходных данных AMPL SPP2



Листинг 5.



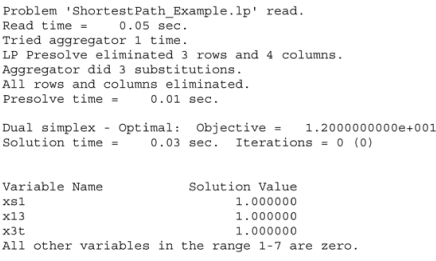
Программа AMPL использует текстовые файлы ввода и вывода ASCII в качестве своего пользовательского интерфейса и, как следствие, не имеет такого же интуитивно понятного графического пользовательского интерфейса, как программа Excel. Однако размер файла модели не изменяется и не увеличивается с размером данных экземпляра, а размер файла данных увеличивается пропорционально количеству узлов и дуг в экземпляре. Как следствие, программа AMPL может обрабатывать гораздо более крупные экземпляры задач. Обычно AMPL взаимодействует либо с Excel, либо с программой базы данных, такой как Access, которая содержит данные дуг и в которой хранится решение потока дуги.

Существуют другие языки моделирования математического программирования и интерфейсы моделирования для решателей смешанно-целочисленного программирования, как в виде приложений с открытым исходным кодом, так и в виде фирменных приложений.

Решение SPP-формулировки в формате LP

Наконец, задача MCNFP может быть сформулирована в формате LP и сохранена в файле. Формат LP соответствует явной алгебраической формулировке задачи. Многие решатели LP имеют возможность читать формулировку задачи в формате LP, а затем решать ее. Формат LP определен в Руководстве по форматам файлов CPLEX (ILOG 2003). Пример формулировки SPP1 в формате LP приведен на рис. 5. Файл формулировки задачи по соглашению имеет расширение «lp» и представляет собой текстовый файл ASCII. Обратите внимание, что этот файл содержит как модель, так и данные в интегрированном формате. Формат LP почти идентичен алгебраической формулировке задачи.

Листинг 6. Файл журнала решения CPLEX примера SPP1



Файл журнала решения этой задачи в формате LP с помощью решателя CPLEX показан далее в листинге 6. Сначала задача считывается из входного файла, затем решается, и, наконец, оптимальные неотрицательные значения переменных решения записываются в файл журнала.

Формулировка в формате LP использует текстовые файлы ввода и вывода ASCII в качестве своего пользовательского интерфейса и, как следствие, не имеет того же интуитивно понятного графического пользовательского интерфейса, что и программа Excel. Однако размер файла LP растет пропорционально количеству узлов и дуг в экземпляре. Как следствие, файл формулировки LP может обрабатывать гораздо большие экземпляры задач, чем Excel. Формат формулировки LP смешивает модель и данные, поэтому, если данные изменяются или требуется решить другую проблему, необходимо сгенерировать новый файл формулировки. Формулировка LP также требует дополнительного программирования, если она должна быть сопряжена с базой данных, реализованной в Excel или Access. Кроме того, пользователь несет ответственность за определение нотации переменных, а использование нотации множеств не допускается. Таким образом, формулировка LP находится между очень интуитивной, но менее мощной методологией решения с использованием Excel и мощной и ориентированной на программирование методологией решения математических языков программирования, таких как AMPL.