

О некоторых вопросах численной реализации решения оптимизационных задач сетевого планирования

Кузеванов И.И., Саженкова Т.В.

Алтайский государственный университет, г. Барнаул

kuzevanovii@inbox.ru, t.sazhenkova@gmail.com

Аннотация

В работе рассматривается задача оптимизации выполнения проекта в плане согласования его стоимости и временных характеристик. Стоимостные и временные характеристики выражаются в стоимости дня реализации проекта в зависимости от затрат на его оснащённость. На основе модельной задачи разработан программный продукт для численных расчётов оптимальных параметров экономического проекта.

Ключевые слова: оптимизация сетевой модели, сетевой граф, метод критического пути, критическая работа.

1. Постановка задачи

Эффективность управления экономическими проектами требует тщательного планирования их реализации и умения обосновать решения расчетами, убедительными для потенциальных инвесторов. В работе методами сетевого планирования и управления (СПУ) осуществляется исследование модельного экономического проекта. Теоретические основы и совокупность используемых расчетных методов представлены в [1–3].

В исследовании используются следующие характеристики экономического процесса:

$t_{\text{опт}}$ — оптимистическая оценка – минимальный срок выполнения работы, в течение которого работа выполняется в наиболее благоприятных условиях ($t_{\text{опт}} = t_{\text{min}}$);

$t_{\text{пес}}$ — пессимистическая оценка – максимальный срок, необходимый для выполнения работы при наиболее неблагоприятных условиях ($t_{\text{пес}} = t_{\text{max}}$).

Разработка программного продукта проведена в работе с использованием исходных данных модельной задачи, список которых с указанием взаимной последовательности и продолжительности работ представлен в следующей таблице:

Работа	t_{min}	t_{max}	Опирается на работы	s_k
b_1	3	8	–	6
b_2	4	10	–	8
b_3	1	6	–	4
b_4	1	9	b_1	6
b_5	1	4	b_1	3
b_6	1	2	b_3	2
b_7	1	4	b_2, b_5, b_6	3
b_8	4	13	b_2, b_5, b_6	9
b_9	1	8	b_4, b_7	5
b_{10}	6	17	b_3	10
b_{11}	2	10	b_2, b_5, b_6, b_{10}	7

s_k – стоимость сокращения времени k -ой работы на 1 день, то есть величина вложений необходимых для такого сокращения. Если не тратить ресурсы, довольствуясь минимальной оснащённостью, то работы выполняются за t_{\max} дней.

В работе рассматривается следующая оптимизационная задача: построение графика зависимости минимального количества вложенных ресурсов от времени реализации проекта. При этом приближение окончания проекта будет давать потенциальную прибыль в I единиц в день.

2. Общая схема процесса оптимизации

I. Построение графа при t_{\max} .

II. Нахождение критического пути, используя алгоритм Дейкстры.

III. Стоимостная оптимизация посредством уменьшения времени критической дуги самым дешевым способом (определяется по наименьшему s_k среди работ, входящих в критическую дугу), пока не появится новый критический путь. Далее уменьшается время L на дугах всех критических путей одновременно.

Останавливаемся тогда, когда самое дешевое ускорение работы будет дороже, чем I [4–6].

3. Алгоритм построения графа

1. Обработка списка работ для получения начального списка событий.

Сначала к списку работ из ячейки столбца “Опирается на работы” (список необходимых работ) добавляются все уже выполненные работы на момент начала соответствующей по строке работы. Например, если требуются работы b_2, b_5, b_6 , то после обработки список будет выглядеть так: b_1, b_2, b_3, b_5, b_6 . Далее производится составления множества всех возможных пересечений работ из столбца “Опирается на работы” и его соединение с множеством работ из того же столбца. Также добавляется элемент, в которых входят все работы проекта, это завершающее событие. Элементы данного множества становятся событиями.

2. Составляется начальное множество ребер. В него попадают все работы проекта, при этом в качестве события-источника для ребра-работы i ставится такое событие a , множество работ которого совпадает с требуемыми работами для работы i . В качестве события-места назначения ставится такое событие, множество работ которого является наименьшим среди тех, что включают в свое множество работ работу i .

3. Добавляем фиктивные работы. Для каждого события I получаем его список работ Works, затем проходим по списку входящих ребер b_j и удаляем из списка Works работы, соответствующие ребру b_j и событию, из которого исходит ребро b_j . Для каждой работы b_k из списка Works ищем событие J , такое, что:

- 1) включает работу b_k ;
- 2) размер его множества работ меньше, чем у события i ;
- 3) размер его множества работ максимален среди тех, что удовлетворяют пунктам 1 и 2.

Добавляем работу, исходящую из j и входящую в i , ее длительность ставим равной 0.

4. Обрабатываем кратные ребра – такие ребра, которые представляют разные работы, но их источники и места назначения равны. Для этого добавляем фиктивные события по следующему алгоритму. Для каждой пары событий находим ребра, которые их соединяют. Если их количество равно 1 или 0, то никаких действий не требуется. Если 2 или больше, то:

- 1) находим A – разность множеств события-места назначения и события-источника;
- 2) добавляем событие, содержащее все работы события источника и одну работу b из A ;
- 3) удаляем работу b из A ;
- 4) находим ребро, которое представляет работу b , и меняем его место назначения на добавленное событие;
- 5) повторяем шаги 2-4 до тех пор, пока множество A не будет содержать только 1 элемент.

5. Пронумеруем вершины:

- 1) присваиваем переменной i значение 0;
- 2) ищем событие, для которого все предыдущие события (связанные с ним ребром в качестве события-источника) уже пронумерованы. Для начального события это считается верным, так как предыдущих событий нет;
- 3) присваиваем ему номер i , увеличиваем i на единицу;
- 4) повторяем шаги 2-3 пока есть непрономерованные события.

6. Граф построен.

4. Алгоритм Дейкстры нахождения критического пути

1. Расчет раннего срока наступления событий:

- 1) ранний срок наступления начального события равен 0;
- 2) ищем событие, для предыдущих событий которого посчитаны все ранние сроки;
- 3) рассчитываем его ранний срок, как максимальное число среди (раннее время предыдущего события плюс длительность работы, соединяющей предыдущее событие с текущим);
- 4) повторяем шаги 2-3 до тех пор, пока не посчитано раннее время для всех событий.

2. Расчет позднего срока наступления событий:

- 1) поздний срок наступления конечного события равен его раннему сроку;
- 2) ищем событие, для последующих событий которого посчитаны все ранние сроки;
- 3) рассчитываем его ранний срок, как максимальное число среди (позднее время последующего события минус длительность работы, соединяющей предыдущее событие с текущим);
- 4) повторяем шаги 2-3 до тех пор, пока не посчитано позднее время для всех событий.

3. Расчеты резервов времени всех событий.

4. Нахождение критического пути:

- 1) ищем все события, для которых резерв времени равен 0;
- 2) так как события уже пронумерованы, то критический путь получается в виде $[a_1, a_2, \dots, a_n]$, где a_i – номера событий.

5. Алгоритм оптимизации

1. Определяется подграф нашего изначального графа, состоящий только из критических путей. Вес ребра будет устанавливаться по формуле:

$$w_k = \begin{cases} s_k, & \text{если } t_{\max} - \frac{Inv_k}{s_k} > t_{\min} \\ \infty, & \text{иначе} \end{cases}$$

Для реализации этого шага необходимо таблицу исходных данных пополнить столбцом начальных значений Inv_k :

Работа	t_{\min}	t_{\max}	Опирается на работы	Inv_k	s_k
b_1	3	8	–	0	6
b_2	4	10	–	0	8
b_3	1	6	–	0	4
b_4	1	9	b_1	0	6
b_5	1	4	b_1	0	3
b_6	1	2	b_3	0	2
b_7	1	4	b_2, b_5, b_6	0	3
b_8	4	13	b_2, b_5, b_6	0	9
b_9	1	8	b_4, b_7	0	5
b_{10}	6	17	b_3	0	10
b_{11}	2	10	b_2, b_5, b_6, b_{10}	0	7

2. Устанавливается список всех путей из начального события в конечное (как минимум 1)

3. Находится множество всех ребер Edges, вес которых не равен ∞ , проходим по ним рекурсивно Recursion(Edges):

1) создаем пустой список списков ребер, которые мы будем удешевлять EdgeList;

2) проходим в цикле по ребрам Edge из Edges:

i Создаем копию Edges, называем Edges2

ii Удаляем из Edges2 все ребра путей, которые содержат Edge

iii Добавляем в EdgeList список Recursion(Edges2)+Edge

3) возвращаем минимальный путь из EdgeList или пустой список, если такой есть.

4. Если рекурсия вернула пустой список или путь, длина которого ∞ , то получена минимальная длительность проекта.

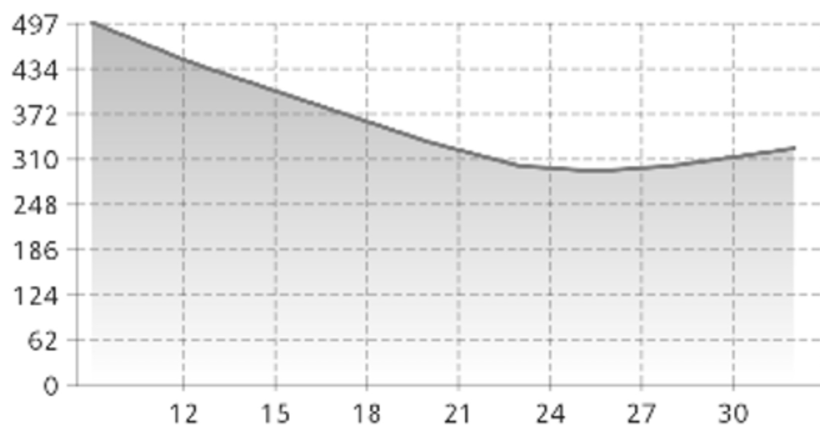
5. Иначе увеличиваем Inv_k на соответствующие s_k для каждого ребра, полученного из рекурсии. Для каждого шага оптимизации сохраняется результат и строится график.

6. Реализация алгоритмов осуществлена на языке Kotlin

Общий код программы расположен по ссылке <https://github.com/ReptileStyle/ProjectManager/releases/tag/demo>.

Данное программное обеспечение позволило осуществить для модельной задачи по описанным выше алгоритмам построение исходного сетевого графа, найти критический путь и вычислить его длину.

При реализации алгоритма оптимизации потенциальная прибыль за 1 день, которую проект будет приносить после завершения, для данной модельной задачи задана равной 10 условным единицам. В результате получен следующий оптимизационный график:



Он демонстрирует, что для модельной задачи наиболее выгодным является такой вариант вложений, при котором проект завершается за 24 дня. Итоговая стоимость проекта тогда равняется 297 условным единицам.

Список литературы

1. Кремер Н.Ш., Путко Б.А., Тришин И.М., Фридман М.Н. Исследование операций в экономике / Под ред. проф. Н.Ш. Кремера. — М. : ЮНИТИ, 2005. — 407 с.
2. Плескунов М.А. Задачи сетевого планирования. — Екатеринбург : Издательство Уральского университета, 2014. — 92 с.
3. Таха Х. Введение в исследование операций. — М., СПб., Киев : Вильямс, 2001. — 912 с.
4. Лебеденко Е.В., Куцакин М.А., Дунаев В.А. Модель случайного времени наступления событий в задачах сетевого планирования и управления проектами, выполняемыми коллективом автономных исполнителей // Интернет-журнал «НАУКОВЕДЕНИЕ». — 2016. — Т. 8, № 3. — URL: <http://naukovedenie.ru/PDF/107TVN316.pdf>.
5. Голенко-Гинзбург Д.И. Стохастические сетевые модели планирования и управления разработками: Монография. — Воронеж : Научная книга, 2010. — 284 с.
6. Катаргин Н.В. Оптимизация сетевого графика комплекса работ в Excel // Корпоративное управление и инновационное развитие экономики Севера: Вестник научно-исследовательского центра корпоративного права, управления и венчурного инвестирования Сыктывкарского государственного университета. — 2018. — № 2. — С. 129–136.