DOI 10.37882/2223–2966.2022.01–2.21

РЕШЕНИЕ ЗАДАЧ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ СЕТЕВЫМИ МЕТОДАМИ

SOLUTION OF DYNAMIC PROGRAMMING PROBLEMS BY NETWORK METHODS

K. Chaadaev

Summary. The possibility of increasing the efficiency of the application of the dynamic programming method by transforming discrete optimization problems into network models is considered. The development of the corresponding algorithm and its software implementation in relation to the solution of multicriteria problems. The developed algorithm and the corresponding machine program can serve as a universal basis for constructing any algorithms and programs for solving discrete dynamic programming problems.

Keywords: algorithm, computational complexity, decomposition, discrete optimization, dynamic programming, exact methods, network model.

Чаадаев Кирилл Витальевич
Московский государственный технический университет им. Н.Э. Баумана
vkchaadaev@gmail.com

Аннотация. Рассматривается возможность повышения эффективности применения метода динамического программирования посредством трансформации задач дискретной оптимизации в сетевые модели. Проведена разработка соответствующего алгоритма и его программная реализация применительно к решению многокритериальных задач. Разработанный алгоритм и соответствующая машинная программа могут служить универсальной основой для построения любых алгоритмов и программ решения дискретных задач динамического программирования.

Ключевые слова: алгоритм, вычислительная сложность, декомпозиция, динамическое программирование, дискретная оптимизация, сетевая модель, точные методы.

Введение

При выполнении многих технических и экономических прикладных задач в различных сферах деятельности нередко возникают проблемы, связанные с необходимостью принятия оптимальных проектных решений, поиск которых целесообразен методами дискретной оптимизации. Отличительной особенностью таких задач является наличие ограниченного множества возможных решений, перебрав которые можно получить оптимальное в виде максимума или минимума целевой функции. Классическим способом исследования задач дискретной оптимизации является метод динамического программирования хотя и относительно редко использующийся для нахождения оптимального решения, поскольку для решения большинства задач или используются более эффективные методы математического программирования или достаточными оказываются различные эмпирические алгоритмы. Оправданным использование метода динамического программирования представляется в случаях, когда, например, при линейном программировании, возникают трудности в формализации условий задачи [8]. Одной таких задач является определение кратчайшего или критического пути на ориентированном графе, где применение динамического программирования существенно снижает объем вычислений, обеспечивая эффективность метода в целом, повышая его практическую значимость, в частности.

Исследованиям в этой области посвящены многие работы как зарубежных, так и отечественных ученых. В частности, Будаевой А.А. показывается, что подавляющее большинство задач планирования и управления являются достаточно сложными с большим количеством условий, поэтому для их эффективного решения наиболее подходит методы дискретной многокритериальной оптимизации [4]. Теоретические вопросы, связанные с математическими моделями технических систем с дискретным характером функционирования и применением аналитических методов расчёта и решения задач проектирования с использованием методов оптимизации подробно рассмотрены Алиевым Т.И. [2]. В работе Мещерякова Г.А. и Дроженко В.М. [11] обосновывается, что в процессе постановки и решения оптимизационных экономических задач возникает потребность использования точных, математически формализованных методов, таких как теория графов и сетей (метод ветвей и границ) с распределением исполнителей по работам (задачи назначений), выбором кратчайшего пути на графе при условии минимизации затрат. Метод сетевого программирования для получения точных решений или верхних (нижних) оценок задач многоэкстремальной дискретной оптимизации дает хорошие результаты при использовании для управления проектами [5]. Примеры прикладных задач из различных сфер деятельности, их математические модели и методы решения на основе современной теории оптимизации, алгоритмы, основанные на комплексном применении динамического программирования и метода ветвей и границ, доведённые до практических реализаций рассмотрены Струченковым В.И. [14]. Однако, поиск решений в задачах дискретной оптимизации связан с принципиальными вычислительными трудностями, в то время как современные комбинаторные методы для практического решения задач дискретной оптимизации ориентированы на разработку алгоритмов, которые позволяют получать приближенное решение с гарантированной оценкой отклонения от оптимума. В целом, алгоритмы упрощения является эффективным приемом поиска решений оптимизационной задачи [16]. Так, например, в работе [6] предлагается итерационный эвристический метод решения задач дискретной комбинаторной оптимизации, основанный на выборе направления движения в пределах комбинаторного дерева с использованием стохастического эвристического критерия, а в работе [7] предложена модификация эволюционных методов оптимизации, позволяющая решать задачи оптимизации с дискретными переменными без предварительного преобразования исходной задачи в задачу псевдобулевой оптимизации.

Тем не менее, при решении задач большой размерности, например, проектирование сетей связи национального масштаба, то есть там, где наличествует многократно повторяющийся цикл расчётов, время поиска, даже с учетом возможностей современных средств вычислительной техники, может оказаться существенно большим. То есть, проблема повышения производительности метода динамического программирования остается актуальной. Исходя из изложенного, цель настоящей работы состоит в разработке алгоритма решения общей задачи динамического программирования при векторном представлении данных, позволяющем трансформировать дискретные задачи в сетевые модели и тем самым существенно повысить эффективность метода.

Методология

Многие задачи оптимального планирования и организации систем являются многошаговыми и решаются методом динамического программирования [3], позволяющий отбраковку вариантов путей, приводящих в конкретное состояние и допускающей дополнительно и отбраковку бесперспективных состояний в процессе счёта [10]. При этом считается, что отсутствует универсальный метод решения задач динамического программирования [9]. С другой стороны, практически любая дискретная задача динамического программирования может быть сведена к задаче определения кратчайшего или критического пути в ориентированном графе [1].

Действительно, практически любая задача динамического программирования путем соответствующего выбора переменных может быть представив в виде сепарабельной модели:

$$\min \quad (\max) Z = \sum_{i=1}^{n} f_i(y_i) \tag{1}$$

где: n — число этапов (шагов) решения,

или $$min(max)Z = \prod_{i=1}^{n} f_i(y_i) \tag{2}$$

причем путем логарифмирования легко перейти от модели (2) к модели (1).

Так как всегда возможна дискретизация переменных y_j, то условие (1) может быть представлено в виде:

$$min(max)Z = \sum_i \sum_i C_{ii} X_{ii} \tag{3}$$

где: C_{ij} — дискретные значения функций $f_j(y_j)$;
X_{ij} — принимает значения «0» или «1».

Тем самым задача динамического программирования трансформируется в целочисленную задачу линейного программирования и, в принципе, может решаться методами целочисленного линейного программирования.

С другой стороны, как показано в [15] любая целочисленная задача линейного программирования мо-нет быть решена сетевыми методами. При этом задача представляется ориентированным графом с начальной и конечной вершинами, как показано на рис. 1.

Для указанного на рис. 1 представления математической модели (3) значения C_{ij} соответствуют ребрам графа, а переменная X_{ij} означает наличие или отсутствие соединений между вершинами.

Тем самым решение целочисленной задачи линейного программирования указанного вида трансформируется в задачу определения кратчайшего или критического пути на ориентированном графе. Наиболее распространенным алгоритмом решения таких задач является алгоритм Дейкстры [12].

Решение подобных задач усложняется при наличии ограничений и в многокритериальных задачах.

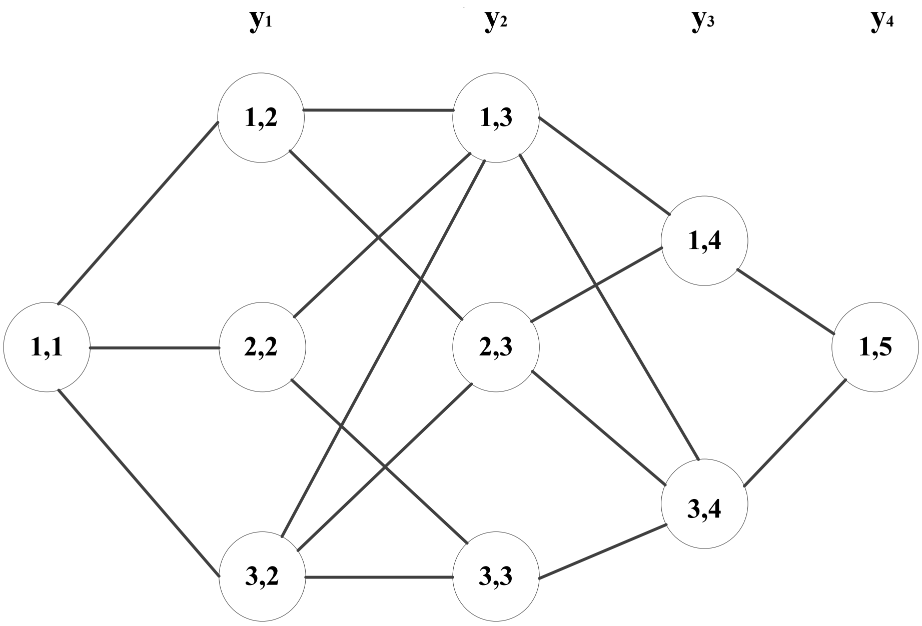

Рис. 1. Представление целочисленной задачи линейного программирования ориентированным графом

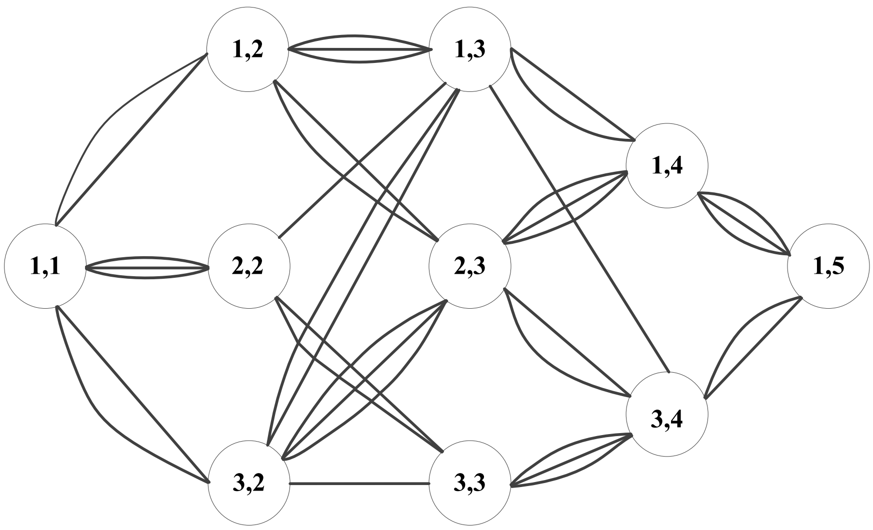

Рис. 2. Пример ориентированного графа в многокритериальных задачах

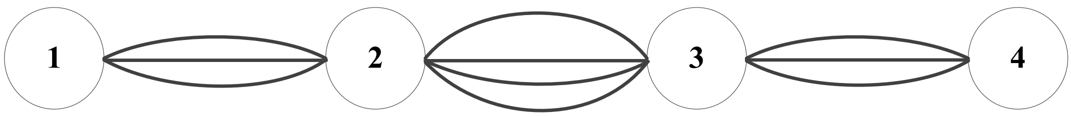

Рис. 3. Трансформация ориентированного графа при наличии ограничений

Рис. 4. Алгоритм решения общей задачи динамического программирования

Так как другие критерии и ограничения также представляют собой сепарабельные функции, то их учет сводится к увеличению числа соединений между вершинами графа. Пример такого графа приведен на рис. 2.

Ряд задач динамического программирования имеет смысл лишь при наличии ограничений или/и нескольких критериев и трансформируются в модели, представляемые графом, указанном на рис. 3.

Таким образом, наиболее сложными среди рассмотренных задач являются такие, которые трансформируются в сетевую модель, показанную на рис. 2. Следовательно, алгоритм решения таких задач является наиболее общим и универсальным.

В качестве основы для построения универсального алгоритма решения дискретных задач динамического программирования используем векторное представление данных, то есть любое решение (выбранный вариант построения системы) может быть охарактеризовано вектором:

$$\bar{R} = [r_1,\ r_2,\ \ldots,\ r_j,\ \ldots,\ r_m]$$

где: r_j — номер варианта построения j-го элемента на соответствующем этапе (шаге);

$$j = \overline{1,m};\ r_j = \overline{1,n}$$

Поиск решения распадаентся на m этапов (шагов), на каждом из которых определяются наборы векторов частных решений, показатели, соответствующие этим решениям и производится отбор доминирующих решений по условиям предпочтения.

Структурная схема алгоритма решения общей задачи динамического программирования при векторном представлении данных (векторным методом) представлена на рис. 4.)

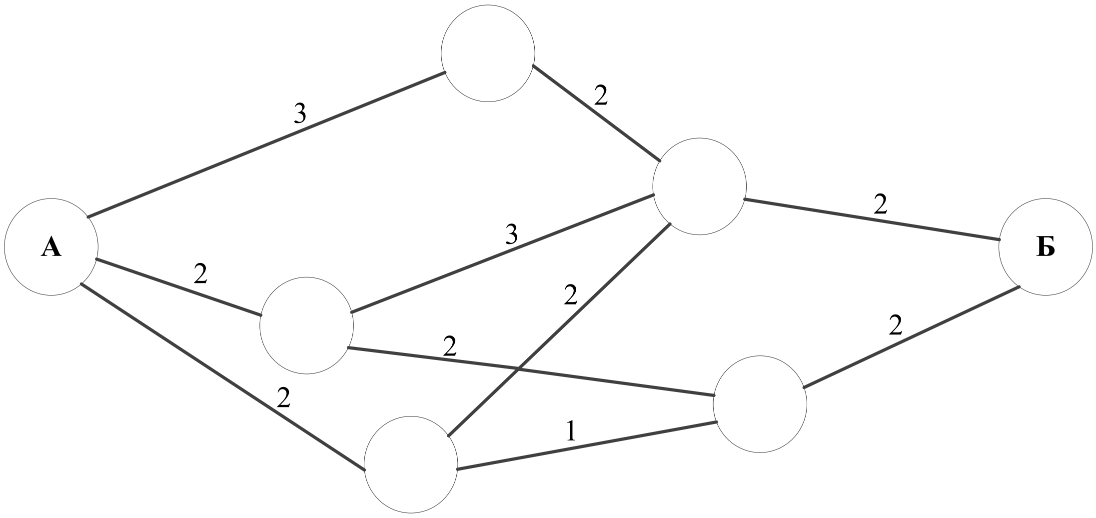

Рис. 5. Варианты построения сети связи между двумя пунктами

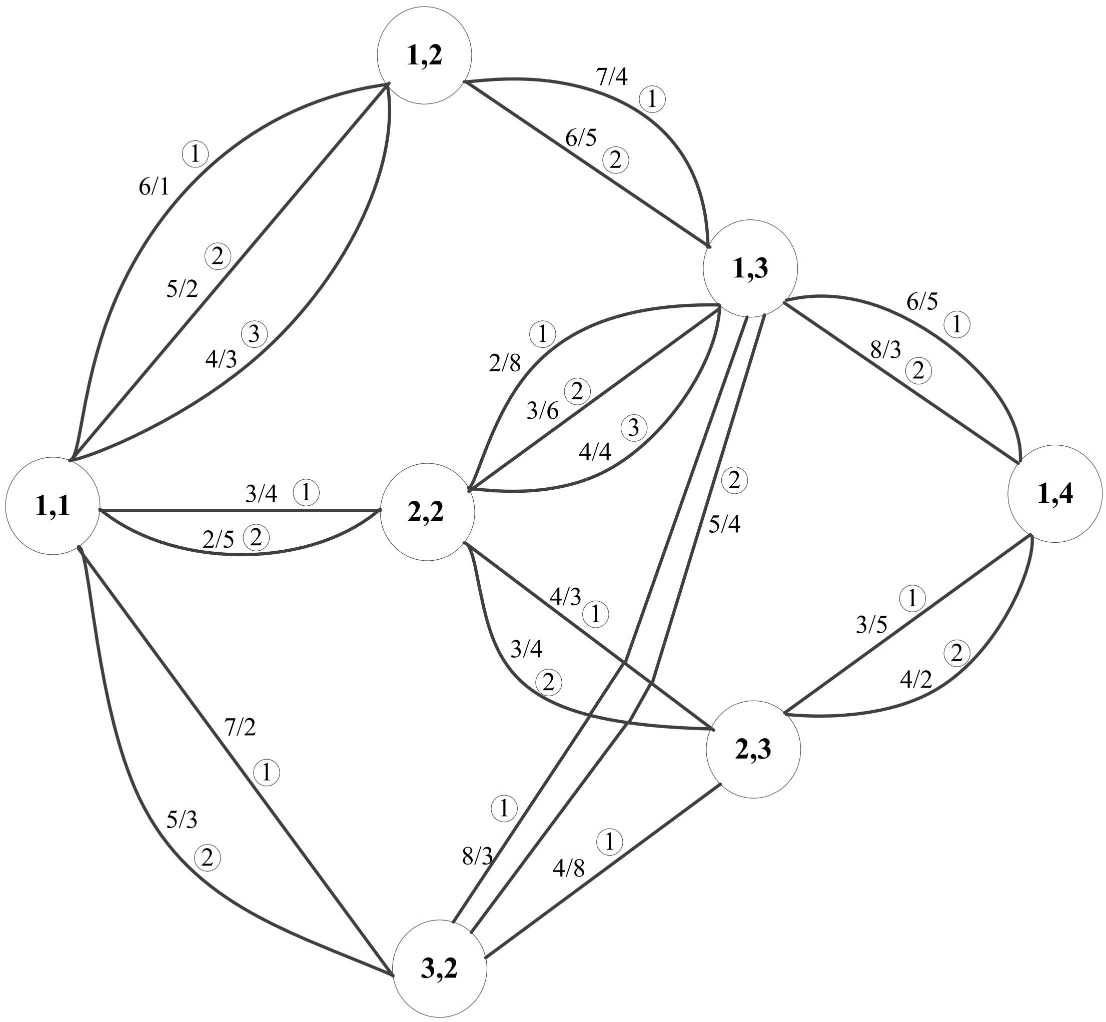

Рис. 6. Графическое представление двухкритериальной задачи динамического программирования в виде сетевой модели

Таблица 1. Исходные данные для решения задачи

<table>
<tr><th rowspan="3">Варианты построения элементов</th><th colspan="15">Показатели элементов</th></tr>
<tr><th rowspan="2" colspan="3">1</th><th colspan="7">2</th><th colspan="5">3</th></tr>
<tr><th colspan="2">1,2</th><th colspan="3">2,2</th><th colspan="2">3,2</th><th colspan="3">1,3</th><th colspan="2">2,3</th></tr>
<tr><td rowspan="2">1</td><td colspan="3">1,2</td><td colspan="2">1,3</td><td colspan="3">1,3</td><td colspan="2">1,3</td><td colspan="3">11,4</td><td colspan="2">1,4</td></tr>
<tr><td>6/1</td><td>5/2</td><td>4/3</td><td>7/4</td><td>6/5</td><td>2/8</td><td>3/6</td><td>4/4</td><td>8/3</td><td>5/4</td><td>6/5</td><td>8/3</td><td>9/1</td><td>3/5</td><td>4/2</td></tr>
<tr><td rowspan="2">2</td><td colspan="3">2,2</td><td colspan="2">-</td><td colspan="3">2,3</td><td colspan="2">2,3</td><td colspan="3">-</td><td colspan="2">-</td></tr>
<tr><td colspan="2">3/4</td><td>2/5</td><td colspan="2">-</td><td colspan="2">4/3</td><td>3/4</td><td colspan="2">4/8</td><td colspan="3">-</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">3</td><td colspan="3">3,2</td><td colspan="2">-</td><td colspan="3">-</td><td colspan="2">-</td><td colspan="3">-</td><td colspan="2">-</td></tr>
<tr><td colspan="2">7/2</td><td>5/3</td><td colspan="2">-</td><td colspan="3">-</td><td colspan="2">-</td><td colspan="3">-</td><td colspan="2">-</td></tr>
</table>

Пояснения к алгоритму:

- последний набор доминирующих решений на последнем шаге является окончательным решением задачи и выводится в качестве результата, при необходимости получения единственного решения нужно использовать дополнительные условия (ограничения, свертку критериев);
- общие положения о применении условий предпочтения описаны в [13], конкретные критерии рассмотрены ниже на примере.

Пример реализации алгоритма

Пусть необходимо спроектировать систему связи между двумя пунктами (А и Б), причем участки этой системы могут быть географически расположены и построены различным образом (см. рис. 5). Число вариантов построения каждого участка показано на рис. 5 на дугах графа.

Критериями отбора вариантов построения системы связи являются затраты (З) и надежность (К – коэффициент простоя оборудования), то есть оптимизационная модель может быть записана в виде:

$$\begin{cases} \sum_j З_j = min \\ \sum_j К_j = max \end{cases}$$

Предполагаем при этом, что выбор направлений и вариантов построения участков связи учитывает соблюдение всех ограничений. Таким образом, мы имеем двухкритериальную задачу динамического программирования, которая непосредственно может быть представлена в виде сетевой модели (рис. 6).

На рис. 6 приняты следующие обозначения:

- над ребрами в кружках — номера ребер в каждой группе (значение для ребер);
- над ребрами через дробную черту — значение затрат (в числителе млн. руб.) и коэффициентов простоя оборудования (в знаменателе *10–6);
- внутри узлов — полные обозначения узлов;
- над узлами — значения q для узлов.

Решения записываются в виде:

$$\overbrace{[(q_1, q_2 \dots q_n)}^{\bar{Q}}\overbrace{(r_1, r_2 \dots r_m)]}^{\bar{R}}$$

где: q — элементы вектора узлов,
a — элементы вектора ребер.

Условия предпочтения могут быть представлены следующим образом:

- для ребер:
  - если $((З_{r+1} > З_r) \wedge (К_{r+1} < К_r)) \vee ((З_{r+1} > З_r) \wedge (К_{r+1} < К_r))$, то оба решения являются доминирующими;
  - если $(З_{r+1} > З_r) \wedge (К_{r+1} > К_r)$, то решение $(r+1)$ отбрасывается (не является доминирующим);
  - если $(З_{r+1} < З_r) \wedge (К_{r+1} < К_r)$, то решение r отбрасывается (не является доминирующим);
  - если $(З_{r+1} = З_r) \wedge (К_{r+1} = К_r)$, то решение являются эквивалентными и отбрасывается любое решение;
  - если $((З_{r+1} = З_r) \wedge (К_{r+1} > К_r)) \vee ()З_{r+1} > З_r) \wedge (К_{r+1} = К_r))$, то отбрасывается $(r+1)$-е решение.
- для узлов:
  - если $(N_{q+1} \neq N_g)$, то оба решения доминирующие;
  - если $(N_{q+1} = N_g)$, то решения эквивален*тные и* выбор решений производится в соответствии с условиями предпочтения для ребер.

Таблица 2

1. Начальный (базисный) набор вариантов решений

$\overline{Q}$	100			200		300	
Узлы	1,2			2,2		3,2	
$\overline{R}$	100	200	300	100	200	100	200
З	6	5	4	3	2	7	5
К	1	2	3	4	5	2	3

2. Набор векторов решений и значения критериев на 2-м шаге

$\overline{Q}$	110						210					
Узлы	1,3						1,3					
$\overline{R}$	110	120	210	220	310	320	110	120	130	210	220	230
З	13	12	12	11	11	10	5	6	7	4	5	6
К	5	6	6	7	7	8	12	10	8	12	11	9

$\overline{Q}$	220				310				320	
Узлы	2,3				1,3				2,3	
$\overline{R}$	110	120	210	220	110	120	210	220	110	210
З	7	6	6	5	15	12	13	10	11	9
К	7	8	8	9	5	6	6	7	10	11

3. Доминирующий набор векторов решений на 2-м шаге

$\overline{Q}$	110		210				220			310
Узлы	1,3		1,3				2,3			1,3
$\overline{R}$	110	120	210	220	230	130	110	210	220	110
З	13	12	4	5	6	7	7	6	5	10
К	5	6	13	11	9	8	7	8	9	7

5. Набор векторов решений и значения критериев на 3-м шаге

$\overline{Q}$	111				211							
Узлы	1,4				1,4							
$\overline{R}$	111	112	121	122	211	212	221	222	231	232	131	132
З	19	21	18	20	10	12	11	13	12	14	13	15
К	11	8	11	9	18	16	16	14	14	12	12	11

$\overline{Q}$	221						311			
Узлы	1,4						1,4			
$\overline{R}$	111	112	211	212	221	222	221	222	211	212
З	10	11	9	10	8	9	16	18	12	13
К	12	9	13	10	14	11	13	11	16	13

7. Доминирующий набор векторов решений на 3-м шаге решения задачи

$\overline{Q}$	221				111
$\overline{R}$	221	222	212	112	112
З	8	9	10	11	21
К	14	11	10	9	8

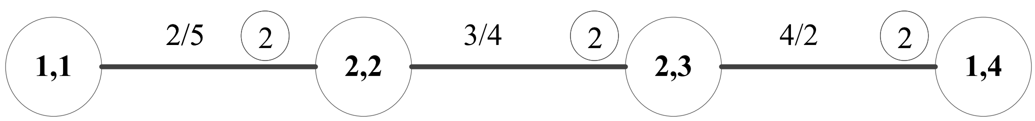

Рис. 7

По графу, представленному на рис. 6, составим таблицу исходных данных (табл. 1).

Последовательность решения описывается в виде векторных преобразований (таблица 2).

Пусть в качестве ограничений задачи задано, что $K_{доп} \leq 11 * 10^{-6}$, тогда решение задачи имеет вид: 221(222) при $З$ = 9 млн. руб. и $K = 11 * 10^{-6}$. Решение означает, что трасса проходит через узлы и ребра (рис. 7).

Для сокращения объемов расчета примем меры по сокращению промежуточных вычислений, которые сводятся к следующим.

Так как исходные данные имеют определенную погрешность, то с учетом этой погрешности эквивалентными могут считаться все решения, отличающиеся по всем критериям на величину $\overline{\Delta}$, не более $\overline{\Delta}_{доп}$, где:

$$\overline{\Delta} = [|S_{1,r+1} - S_{1,r}|, |S_{2,r+1} - S_{2,r}| \dots |S_{\omega,r+1} - S_{\omega,r}| \dots]$$

где: ω — номер критерия оптимальности.

Еще большего сокращения набора доминирующих решений можно достичь, используя «жесткие» условия предпочтения в виде:

$\sum_{\omega}\beta_{\omega,r+1} * \delta_{S_{\omega},r+1} < 0$ — решение (r + 1) отбрасывается;

$\sum_{\omega}\beta_{\omega,r+1} * \delta_{S\omega,r+1} \geq 0$ — решение (r + 1) сохраняется;

где: β_{ω} — весовые коэффициенты, $\Sigma\beta_{\omega} = 1$ (при одинаковых значениях β_{ω} можно принять $\beta_{\omega} = 1$).

$$\delta_{S_{\omega}} = \frac{\Delta S_{\omega}}{S_{\omega} - S_{щ_{min}}}$$

$$ДS_{щ} = \begin{cases} S_{щ,r} - S_{щ,r+1} \text{ при определении минимума критерия} \\ S_{щ,r+1} - S_{щ,r} \text{ при определении максимума критерия} \end{cases}$$

$$S_{щ_{max}} = \max_{i}\{S_1, S_2, \dots, S_e\}$$

$$S_{щ_{min}} = \min_{i}\{S_1, S_2, \dots, S_e\}$$

где: e — число критериев оптимальности.

Введение коэффициентов β позволяет выделить наиболее важные критерии.

На основании рассмотренного алгоритма разработана соответствующее программное обеспечение, являющееся основой для построения программ решения любых дискретных задач динамического программирования.

Заключение

1. Дискретные задачи динамического программирования трансформируются в сетевые модели. Тем самым может быть построен универсальный алгоритм решения таких задач.
2. Разработанный алгоритм и соответствующая машинная программа могут служить универсальной основой для построения любых алгоритмов и программ решения дискретных задач динамического программирования.
3. Элементом новизны предлагаемого решения является высокопроизводительный алгоритм расчета, позволяющий оптимизировать трудозатраты при программировании.
4. Направления дальнейших исследований будут связаны с доработкой программного обеспечения для построения цифрового двойника, позволяющего осуществлять перспективное проектирование сетей связи, оптимизируемых по заданным критериям.

ЛИТЕРАТУРА

1. Алексеев О.Г. Комплексное применение методов дискретной оптимизации.— М.: Наука, 1987.— 248 с.
2. Алиев Т.И. Задачи и методы проектирования дискретных систем.— СПб: Университет ИТМО, 2015.— 127 с.
3. Беллман Р. Динамическое программирование.— М.: Изд-во иностр. лит., 1960.— 400 с.
4. Будаева А.А. Использование метода эталонов для решения задач дискретной многокритериальной оптимизации // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика.— 2015.— Том 15.— № 1.— С. 22–27.— DOI 10.18500/1816–9791–2015–15–1–22–27.
5. Буркова И.В. Метод сетевого программирования в задачах дискретной оптимизации // Вестник Воронежского государственного технического университета.— 2010.— Том 6.— № 8.— С. 154–159.
6. Ватутин Э.И., Дремов Е.Н., Мартынов И.А., Титов В.С. Метод взвешенного случайного перебора для решения задач дискретной комбинаторной оптимизации // Известия Волгоградского государственного технического университета.— 2014.— № 10 (137).— С. 59–64.
7. Галушин П.В., Семенкина О.Э. Разработка и исследование эволюционных алгоритмов дискретной оптимизации // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева.— 2011.— № 5 (38).— С. 25–29.
8. Иванко Е.Е. Метод динамического программирования в минимаксной задаче распределения заданий с равноценными исполнителями // Вестник Южно-Уральского государственного университета. Серия: математическое моделирование и программирование.— 2013.— Том 6.— № 1.— С. 124–133.

9. Карманов В.Г. Математическое программирование.— М.: Физматлит, 2001.— 263 с.
10. Карпов Д.А., Струченков В.И. Динамическое программирование в прикладных задачах специального вида // Прикладная информатика.— 2020.— Том 15.— № 3 (87).— С. 46–59.— DOI 10.37791/2687–0649–2020–15–3–46–59.
11. Мещеряков Г.А., Дроженко В.М. Применение методов дискретной оптимизации для синтеза структур систем менеджмента // РИСК: Ресурсы, Информация, Снабжение, Конкуренция.— 2012.— № 1.— С. 39–41.
12. Нечепуренко М.И., Попков В.К., Майнагашев С.М., Кауль С.Б., Проскуряков В.А., Кохов В.А., Грызунов А.Б. Алгоритмы и программы решения задач на графах и сетях.— Новосибирск: Наука. Сиб. отд-ние, 1990.— 515 с.
13. Растригин Л.А. Современные принципы управления сложными объектами.— М.: Сов. радио, 1980.— 232 с.
14. Струченков В.И. Дискретная оптимизация: модели, методы, алгоритмы решения прикладных задач.— М.: СОЛОН-Пресс, 2016.— 192 с.
15. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей.— М.: Мир, 1984.— 496 с.
16. Чернов С., Титов С., Чернова Л., Гогунский В., Чернова Л., Колесникова К. Алгоритм упрощения решения в задачах дискретной оптимизации // Восточно-Европейский журнал передовых технологий.— 2018.— Том 3.— № 4 (93).— С. 34–43.— DOI 10.15587/1729–4061.2018.133405.

© Чаадаев Кирилл Витальевич (vkchaadaev@gmail.com).

Журнал «Современная наука: актуальные проблемы теории и практики»

Московский Государственный Технический Университет им. Н.Э. Баумана