DOI: 10.18137/RNU.V9187.24.03.P.3 УДК 004.021

Л.М. Божко, А.И. Дергачев, С.А. Дергачев

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ СИМПЛЕКС-МЕТОДА

Аннотация. Использование математического аппарата для выбора наиболее эффективного решения проблемы, поиска оптимального варианта актуально и востребовано с учетом ограниченности ресурсов. Методы линейного программирования находят широкое применение при решении экономических задач, планировании производства. Исследование направлено на разработку методики открытых проектов решения специальных задач. Предлагаемый подход позволяет оценить эффективность управленческой деятельности с позиций системного подхода к решению задач на ЭВМ, рассчитать и обосновать предлагаемое решение поставленной задачи. В качестве примера подготовки к решению на ЭВМ выбрана задача линейного программирования. Решение такой задачи позволяет находить оптимальный вариант использования отведенных ресурсов.

Ключевые слова: математическая модель, задача линейного программирования, симплекс-метод, математическое моделирование.

L.M. Bozhko, A.I. Dergachev, S.A. Dergachev

MATHEMATICAL MODEL FOR SOLVING THE LINEAR PROGRAMMING PROBLEM USING THE SIMPLEX METHOD

Abstract. The use of a mathematical apparatus to choose the most effective solution of the problem and find the best option is relevant and in demand taking into account the limited resources. Linear programming methods are widely used in solving economic problems, in production planning. The study is aimed at developing a methodology for developing open projects for solving special problems that can be used in organizations. The proposed approach makes it possible to assess the effectiveness of management activities from the standpoint of a systematic approach to preparing for solving tasks on a computer, to calculate and justify the proposed solution of the task. As an example of preparation for a computer solution, a linear programming problem was chosen. Solving this problem allows you to find the best option for using the allocated resources.

Keywords: mathematical model, linear programming problem, simplex method, mathematical modeling.

Введение

При решении управленческих задач с применением современных компьютерных технологий зачастую необходима выдача результата решения в строгом соответствии с задаваемым макетом выходного документа. Такой порядок оформления результата решения задачи позволяет сократить время, затрачиваемое на принятие решения. Однако встает вопрос о понимании программных текстов, что позволяет оценить результаты в практике программирования для ЭВМ. Действительно, наличие исходных текстов программ может быть полезным, но этого бывает недостаточно. Чтобы понять нетривиальную программу, в большинстве случаев требуется дополнительная информация. Решение этой проблемы

Божко Леся Михайловна
доктор экономических наук, доцент, профессор кафедры информационных и вычислительных систем, Петербургский университет путей сообщения Александра I, Санкт-Петербург. Сфера научных интересов: компьютерное и имитационное моделирование, алгоритмы и методы имитационного моделирования в экономических системах, цифровые технологии в менеджменте. Автор более 160 опубликованных научных работ. ORCID: 0000-0002-3329-7977, AuthorID: 648758, SPIN-код: 4076-2375.
Электронный адрес: lemib@rambler.ru

Дергачев Алексей Иванович
кандидат военных наук, доцент, доцент кафедры информационных и вычислительных систем, Петербургский университет путей сообщения Александра I, Санкт-Петербург. Сфера научных интересов: компьютерное и имитационное моделирование, алгоритмы и методы имитационного моделирования при строительстве и восстановлении транспортных объектов. Автор более 100 опубликованных научных работ. ORCID: 0000-0001-9061-8530, AuthorID: 747502, SPIN-код: 6282-7442,
Электронный адрес: d_ader@mail.ru

Дергачев Сергей Алексеевич
кандидат экономических наук, доцент кафедры информационных и вычислительных систем, Петербургский университет путей сообщения Александра I, Санкт-Петербург. Сфера научных интересов: компьютерное и имитационное моделирование, алгоритмы и методы имитационного моделирования при строительстве и восстановлении транспортных объектов. Автор 30 опубликованных научных работ. ORCID: 0009-0000-9513-6530, AuthorID: 750463, SPIN-код: 2513-5950.
Электронный адрес: deburg@mail.ru

состоит в разработке подробной и аккуратно оформленной проектной документации, в которую программная документация входит как одна из составляющих.

По мнению профессора А.А. Шалыто, причины отсутствия или формального наличия проектной документации для программ связаны с тем, что чем более закрыт (не документирован) проект, тем более незаменим его автор, и технический прогресс привел к менее ответственному отношению к программированию [1]. Без качественной проектной документации повторное использование кода (одного из достоинств объектно ориентированного программирования) может привести к сбоям. Вместе с тем наличие проектной документации информационной системы позволяет обеспечить разработку защищенного программного обеспечения. Для создания программной документации может быть использована технология Doc Book, дополненная скриптами преобразования документов в соответствии с ГОСТами разработки программной документации [2].

В данной статье в качестве образца подробной открытой проектной документации (подготовки к решению на ЭВМ) выбрана задача линейного программирования, так как проблема экономии ресурсов не теряет актуальности. Решение такой задачи позволяет находить оптимальный (наилучший) вариант использования выделенных ограниченных ресурсов в результате их обслуживания имеющимися средствами.

Решение задач линейного программирования

Задачи линейного программирования решаются с помощью целого ряда методов: апекс-метода, ставшего новой версией итерационного метода [3], нейросетевого метода [4], смешанного целочисленного линейного программирования [5] и др.

Симплекс-метод находит применение для решения широкого круга экономических задач [6–9]. Линейное программирование находит применение во многих сферах.

Для решения задачи линейного программирования разработано два открытых проекта:

• для решения симплекс-методом (задача 1);

• для решения М-методом (задача 2).

Необходимость рассмотрения задачи 2 совместно с задачей 1 вызвана двумя обстоятельствами. Во-первых, некоторые задачи линейного программирования не удается решить симплекс-методом, однако после преобразования они могут быть решены М-методом; во-вторых, укрупненные схемы алгоритмов симплекс-метода и М-метода в своей основе идентичны (основное отличие алгоритмов состоит в размерах заполняемых таблиц и правилах заполнения).

После анализа соответствующих госстандартов[1], документов по стандартизации и требований, предъявляемых к открытому проекту, можно определить следующую структуру проекта.

1. Описание постановки задачи (функциональная часть).

1.1 Характеристика задачи.

1.2 Выходные данные.

1.3 Входные данные (постоянные и переменные).

2. Описание алгоритма (математическое обеспечение).

2.1 Математическое описание (в том числе модель задачи).

2.2 Выбор метода решения.

2.3 Алгоритм решения (в виде схемы алгоритма).

3. Отладка программы (программное обеспечение).

3.1 Исходный текст программы.

3.2 Исходные данные (их значения).

3.3 Выявление и исправление синтаксических и семантических ошибок.

3.4 Результаты решения задачи другими средствами.

3.5 Результаты решения задачи на ЭВМ по составленной программе.

4. Текст программы (программное обеспечение).

4.1 Описание программных элементов.

4.2 Текст ...-программы (вместо многоточия указывается название языка программирования).

5. Вывод о работоспособности программы.

Сутью процесса подготовки задачи к решению на ЭВМ является разработка трех последовательно выполняемых частей (триады программирования): математическая мо-

---

[1] ГОСТ 34.003–90. Межгосударственный стандарт. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. М. : Изд-во стандартов, 2002. 9 с. ОКСТУ 0034. Группа П00; ГОСТ 24.104–85. Межгосударственный стандарт. Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Общие требования. М. : Изд-во стандартов, 2002. 13 с. ОКСТУ 0024. Группа П87.

дель – схема алгоритма – текст программы. Рассмотрим первые два этапа подготовки задач к решению на ЭВМ (второй этап – только математическое описание).

Основная часть постановки задачи

Подготовить предложение по определению количества каждого из двух возможных типов схем погрузок комплексов спецгруза для максимальной загрузки (по весу) железнодорожного подвижного состава *F*, т.

Для погрузки выделен железнодорожный подвижной состав четырех видов ($i = \overline{1,4}$) в количестве:

- четырехосных крытых вагонов $(b(1))$ – 12 ед.*);
- двухосных крытых вагонов $(b(2))$ – 8 ед.;
- двухосных платформ $(b(3))$ – 12 ед.;
- четырехосных платформ $(b(4))$ – 16 ед.

Погрузка комплексов спецгруза может производиться с использованием схем погрузок двух типов ($j = \overline{1,2}$), включающих следующее количество и вид железнодорожного подвижного состава:

а) для одной схемы погрузки 1-го типа (j = 1):
1) четырехосных крытых вагонов ($a_{(1,1)}$) – 2 ед./схему;
2) двухосных крытых вагонов ($a_{(2,1)}$) – 1 ед./схему;
3) четырехосных платформ ($a_{(4,1)}$) – 4 ед./схему;
б) для одной схемы погрузки 2-го типа (j = 2):
1) четырехосных крытых вагонов ($a_{(1,2)}$) – 2 ед./схему;
2) двухосных крытых вагонов ($a_{(2,2)}$) – 2 ед./схему;
3) двухосных платформ ($a_{(3,2)}$) – 4 ед./схему.

Расчетная загрузка железнодорожного подвижного состава в зависимости от типа схемы погрузки составляет:

- для 1-го типа (j = 1) $C_{(1)} = 200$ т/схему;
- для 2-го типа (j = 2) $C_{(2)} = 300$ т/схему.

Количество единиц подвижного состава по каждому *i*-му виду $h_{(i)}$ непосредственно зависит от выбранного количества схем погрузки всех типов, то есть в данном случае – от количества схем погрузки $X_{(1)}$ и $X_{(2)}$, являющихся неизвестными (их значения следует определить в процессе решения задачи).

Вычисления производятся по формулам:

- для четырехосных крытых вагонов (i = 1)

$$h_{(1)} = a_{(1,1)}X_{(1)} + a_{(1,2)}X_{(2)}, \quad (1)$$

- для двухосных крытых вагонов (i = 2)

$$h_{(2)} = a_{(2,1)}X_{(1)} + a_{(2,2)}X_{(2)}, \quad (2)$$

- для двухосных платформ (i = 3)

$$h_{(3)} = a_{(3,1)}X_{(1)} + a_{(3,2)}X_{(2)}, \quad (3)$$

- для четырехосных платформ (i = 4)

$$h_{(4)} = a_{(4,1)}X_{(1)} + a_{(4,2)}X_{(2)}, \quad (4)$$

где $X_{(1)}$, $X_{(2)}$ – неизвестные, представляющие собой количество схем погрузки, соответственно, 1-го (j = 1) и 2-го (j = 2) типов, схем; $a_{(i,j)}$ – норма расхода подвижного состава *i*-го вида на одну схему погрузки *j*-го типа, ед./схему.

Требуется определить:

а) максимально возможную загрузку (по весу) F, т, выделенного железнодорожного подвижного состава при перевозке комплексов спецгруза;

б) количество схем погрузки 1-го типа ($j = 1$) $X_{(1)}$ и 2-го типа ($j = 2$) $X_{(2)}$, обеспечивающее максимальную загрузку (по весу) выделенного железнодорожного подвижного состава при перевозке комплексов спецгруза, т;

в) количество единиц железнодорожного подвижного состава четырех видов $h_{(1)}$, $h_{(2)}$, $h_{(3)}$ и $h_{(4)}$, используемое для перевозки комплексов спецгруза;

г) количество единиц неиспользуемого железнодорожного подвижного состава (резерв) по каждому i-му виду ($i = \{1, 2, 3, 4\}$) $R_{(1)}$, $R_{(2)}$, $R_{(3)}$ и $R_{(4)}$.

Выходные данные сгруппированы в макете выходного документа, представленном на Рисунке 1. Выходные данные расположены так, чтобы на ознакомление с документом было затрачено минимальное время.

ОПТИМАЛЬНЫЙ ПЛАН ИСПОЛЬЗОВАНИЯ ЖЕЛЕЗНОДОРОЖНОГО ПОДВИЖНОГО СОСТАВА
ПРИ ПОГРУЗКЕ КОМПЛЕКСОВ СПЕЦГРУЗА

Максимальная загрузка ж.-д. подвижного состава – ________, т

Расчетная загрузка (по весу) одной схемы погрузки:

1-го типа – __________, т/схему;

2-го типа – __________, т/схему.

3. Количество схем погрузки, выбираемое для перевозки спецгруза:

1-го типа – __________, схем;

2-го типа – __________, схем.

4. Использование ж.-д. подвижного состава при погрузке спецгруза показано в Таблице 1.

Таблица 1

Использование железнодорожного подвижного состава

Наименование показателей	Виды ж.-д. подвижного состава			
	Крытые вагоны		Платформы	
	4-осные	2-осные	4-осные	2-осные
1. Потребность в подвижном составе при использовании схемы погрузки 1-го типа, ед./схему	–	–	–	–
2. То же, 2-го типа, ед./схему	–	–	–	–
3. Выделено подвижного состава, ед.	✓ –	✓ –	✓ –	✓ –
4. Используется для перевозки, ед.	✓ –	✓ –	✓ –	✓ –
5. Остается в резерве, ед.	✓ –	✓ –	✓ –	✓ –

______________________ (Инициалы и фамилия) ______________ (Дата)

Рисунок 1. Макет выходного документа

Источник: здесь и далее рисунки выполнены авторами.

Обозначение переменных содержится в макете выходного документа (см. Рисунок 2).

ОПТИМАЛЬНЫЙ ПЛАН ИСПОЛЬЗОВАНИЯ ЖЕЛЕЗНОДОРОЖНОГО ПОДВИЖНОГО СОСТАВА ПРИ ПОГРУЗКЕ КОМПЛЕКСОВ СПЕЦГРУЗА

Максимальная загрузка ж.-д. подвижного состава – F, т
Расчетная загрузка (по весу) одной схемы погрузки:
1-го типа – $C_{(1)}$, т/схему;
2-го типа – $C_{(2)}$, т/схему.
3. Количество схем погрузки, выбираемое для перевозки комплексов спецгруза:
1-го типа – $X_{(1)}$, схем;
2-го типа – $X_{(2)}$, схем.
3. Использование ж.-д. подвижного состава при погрузке комплексов спецгруза показано в Таблице 2.

Таблица 2

Использование железнодорожного подвижного состава

Наименование показателей	Виды ж.-д. подвижного состава			
	Крытые вагоны		Платформы	
	4-осные	2-осные	4-осные	2-осные
1. Потребность в подвижном составе при использовании схемы погрузки 1-го типа, ед./схему	$a_{(1,1)}$	$a_{(2,1)}$	$a_{(3,1)}$	$a_{(4,1)}$
2. То же, 2-го типа, ед./схему	$a_{(1,2)}$	$a_{(2,2)}$	$a_{(3,2)}$	$a_{(4,2)}$
3. Выделено подвижного состава, ед.	$b_{(1)}$	$b_{(2)}$	$b_{(3)}$	$b_{(4)}$
4. Используется для перевозки, ед.	$h_{(1)}$	$h_{(2)}$	$h_{(3)}$	$h_{(4)}$
5. Остается в резерве, ед.	$R_{(1)}$	$R_{(2)}$	$R_{(3)}$	$R_{(4)}$

Две составляющие реквизита «Подпись», расположенные в предпоследней строке, обозначить W_1 и W_2, а реквизит последней строки «Дата» – D_{ata}

Рисунок 2. Обозначение переменных в макете выходного документа

Математическое описание

Для достижения максимальной загрузки (по весу) F, т, железнодорожного подвижного состава необходимо реализовать $X_{(1)}$ схем погрузки 1-го типа и $X_{(2)}$ схем погрузки 2-го типа. Загрузка (по весу) железнодорожного подвижного состава при этом составит

$$F = 200X_{(1)} + 300X_{(2)}. \tag{5}$$

Количество железнодорожного подвижного состава определенного вида, задействованное в схемах погрузок, не должно превышать количество выделенного для погрузки подвижного состава этого вида. Исходя из этой предпосылки значения неизвестных $X_{(1)}$ и $X_{(2)}$ должны удовлетворять системе ограничений:

$$\left.\begin{array}{ll} \text{– для четырехосных крытых вагонов} & 2X_{(1)} + 2X_{(2)} \le 12 \\ \text{– для двухосных крытых вагонов} & X_{(1)} + 2X_{(2)} \le 8 \\ \text{– для двухосных платформ} & 4X_{(2)} \le 12 \\ \text{– для четырехосных платформ} & 4X_{(1)} \le 16 \end{array}\right\} \tag{6}$$

Кроме того, количество используемых схем погрузок каждого типа не может иметь значение меньше нуля, поэтому следует определить условия для принятия допустимых значений:

$$X_{(1)} \ge 0; X_{(2)} \ge 0\ . \tag{7}$$

Решение задачи об использовании ресурсов состоит в отыскании значений неизвестных $X_{(1)}$ и $X_{(2)}$, удовлетворяющих системе ограничений (6), условиям (7) и обеспечивающих для линейной функции (5) наибольшее значение.

Сгруппировав рассмотренные три математические зависимости в единую систему, получим математическую модель задачи линейного программирования в стандартной форме:

$$\left.\begin{array}{l} F = 200\mathrm{X}_{(1)} + 300X_{(2)} \to max \\ \text{Система ограничений} \quad \begin{cases} 2X_{(1)} + 2X_{(2)} \le 12 \\ X_{(1)} + 2X_{(2)} \le 8 \\ 4X_{(2)} \le 12 \\ 4X_{(1)} \le 16 \end{cases} \\ X_{(1)} \ge 0; X_{(2)} \ge 0 \end{array}\right\}, \tag{8}$$

где «→max» – требование получения максимального значения линейной функции *F*.

Выбор метода решения

Полученная система (8) является стандартной формой записи задачи линейного программирования и предположительно может быть решена с использованием симплекс-метода, являющегося аналитическим методом. Выбор этого итерационного метода обосновывается тем, что в данном варианте вычисления, во-первых, сравнительно просты для выполнения и, во-вторых, сводятся к заполнению и пересчету симплексных таблиц, что позволяет без особых затруднений разработать программу для ЭВМ.

Система ограничений, указанная в системе зависимостей (9), включает только линейно-независимые уравнения, то есть такие уравнения, которые нельзя получить из остальных путем сложения или умножения на постоянные величины (если бы были уравнения, полученные из остальных уравнений, то их следовало бы отбросить). После того как задача линейного программирования представлена в канонической форме записи, может быть заполнена симплексная таблица:

$$\left.\begin{array}{l} F = 200\mathrm{X}_{(1)} + 300\mathrm{X}_{(2)} + 0\mathrm{X}_{(3)} + 0\mathrm{X}_{(4)} + 0\mathrm{X}_{(5)} + 0\mathrm{X}_{(6)} \to \max \\ \text{Система ограничений} \quad \begin{cases} 2X_1 + 2X_2 + 1X_3 + 0X_4 + 0X_5 + 0X_6 = 12 \\ 1X_1 + 2X_2 + 0X_3 + 1X_4 + 0X_5 + 0X_6 = 8 \\ 0X_1 + 4X_2 + 0X_3 + 0X_4 + 1X_5 + 0X_6 = 12 \\ 4X_1 + 0X_2 + 0X_3 + 0X_4 + 0X_5 + 1X_6 = 16 \end{cases} \\ \mathrm{X}_{(1)} \ge 0; \mathrm{X}_{(2)} \ge 0; \mathrm{X}_{(3)} \ge 0; \mathrm{X}_{(4)} \ge 0; \mathrm{X}_{(5)} \ge 0; \mathrm{X}_{(6)} \ge 0 \end{array}\right\} \tag{9}$$

В представляемом исследовании рассматривается только одна задача (*задача 1*) – нахождение оптимального решения по максимуму. Это ограничение позволяет без излишнего усложнения материала и увеличения его объёма выполнить одну из поставленных целей, предусматривающую оказание активной помощи в усвоении методики решения задачи линейного программирования. Фактом, подтверждающим широкие возможности применения симплекс-метода, является использование его в качестве составной части М-метода при решении *задачи 2*, в которой осуществлен поиск минимума.

Симплексная таблица приведена на Рисунке 3.

Симплексная таблица, заполненная входными данными рассматриваемой задачи, приведена на Рисунке 4 и представляет собой начальный базисный допустимый план (далее – БДП).

j\i	0	1	2	…	n	n+1	n+2	…	n+m	n+m+1
0		$C_{(1)}$	$C_{(2)}$	…	$C_{(n)}$	0	0	…	0	0
1	n+1	$a_{(1,1)}$	$a_{(1,2)}$	…	$a_{(1,n)}$	1	0	…	0	$b_{(n)}$
2	n+2	$a_{(2,1)}$	$a_{(2,2)}$	…	$a_{(2,n)}$	0	1	…	0	$b_{(2)}$
…	…	…	…	…	…	…	…	…	…	…
m	n+m	$a_{(m,1)}$	$a_{(m,2)}$	…	$a_{(m,n)}$	0	0	…	1	$b_{(m)}$

Значение целевой функции

Нулевая строка

Индексы базисных неизвестных

Столбец свободных членов системы ограничений (значение базисных неизвестных)

Рисунок 3. Симплексная таблица, заполненная начальным допустимым планом

Примечания

1. Часть нулевой строки в диапазоне столбцов с 1-го по (n + m) называется индексной строкой.

2. Клетка нулевой строки и нулевого столбца не заполняется.

j\i	0	1	2	3	4	5	6	7
0		200	300	0	0	0	0	0
1	3	2	2	1	0	0	0	12
2	4	1	2	0	1	0	0	8
3	5	0	4	0	0	1	0	12
4	6	4	0	0	0	0	1	16

Рисунок 4. Симплексная таблица, заполненная начальным допустимым планом

Симплексный метод предусматривает решение задачи с использованием пяти подзадач:

– подзадача 1 «Ввод данных и заполнение симплексной таблицы»;

– подзадача 2 «Выбор ключевого столбца»;

– подзадача 3 «Поиск ведущей строки»;

– подзадача 4 «Пересчет симплексной таблицы»;

– подзадача 5 «Формирование и выдача на печать выходного документа».

Кроме того, симплекс-метод предусматривает проверку *двух условий*:

1) после подзадачи 2 при положительном ответе на вопрос «БДП оптимален?» выполняется подзадача 5, а при отрицательном – подзадача 3;

2) после подзадачи 3 при отрицательном ответе на вопрос «Задача неразрешима?» выполняется подзадача 4, а при положительном ответе выдается сообщение «Задача неразрешима», и решение задачи прекращается.

Симплекс-метод предусматривает для каждой подзадачи методику ее решения с использованием конкретных математических зависимостей в виде формул, а для каждого условия – его математический аналог.

Математическая модель ввода входных данных и заполнения симплексной таблицы представляет собой следующую систему формул:

$$S_{(i,j)} = \begin{cases} n+i \, при \, 1 \le i \le m \, и \, j=0, \\ C_{(j)} при \, i=0 \, и \, 1 \le j \le n, \\ a_{(i,j)} при \, 1 \le i \le m \, и \, 1 \le j \le n, \\ b_{(i)} при \, 1 \le i \le m \, и \, j = n+m+1, \\ 1 \, при \, 1 \le i \le m \, и \, j = n+i, \\ 0 - в \quad остальных \quad случаях \end{cases}, \tag{10}$$

где $C_{(j)}$ – коэффициент целевой функции при *j*-м неизвестном;

$a_{(i,j)}$ – коэффициент при *j*-м неизвестном, расположенный в i -й строке;

$b_{(i,j)}$ – i -й свободный член системы ограничений.

Заполненная симплексная таблица и представляет собой начальный БДП.

Выбор ключевого столбца. Проверка плана на оптимальность

Предназначение ключевого столбца состоит в том, чтобы определить свободную неизвестную $X_{(j)}$ (при $j = \overline{1, n+m}$), которую следует ввести в базис для БДП.

Для улучшения БДП, записанного в симплексной таблице, необходимо последовательно вводить в базис свободные неизвестные $X_{(j)}$, у которых находящиеся в индексной строке коэффициенты целевой функции $C_{(j)}$ являются положительными числами; при поиске максимума происходит последовательное увеличение абсолютного значения целевой функции *F* (в конечном счете находится максимум ее абсолютного значения). Если в индексной строке очередной таблицы несколько положительных чисел, то целесообразнее вводить в базис ту свободную неизвестную, которой соответствует большее положительное число индексной строки, – абсолютное значение целевой функции *F* будет увеличиваться быстрее. Если в индексной строке очередной таблицы нет ни одного положительного числа, то БДП улучшить нельзя, ибо он уже оптимален.

Столбец, в котором в строке находится наибольшее положительное число, называется *ключевым*. Номер ключевого столбца таблицы совпадает с индексом свободной неизвестной, вводимой в базис (в нулевой столбец) для получения улучшенного БДП.

Математическая модель выбора ключевого столбца представлена формулой

$$r_{max} = S_{(0,k)} = \max\{S_{(0,j)}\}, \, если 1 \le j \le (n+m) \, и \, S(0,j) > 0, \tag{11}$$

где r_{max} – максимальное значение числа, находящегося в индексной строке в диапазоне столбцов с 1-го по $(n + m)$ и указывающего тем самым на то, что этот столбец является ключевым; k – номер ключевого столбца.

Формула (11) обеспечивает выбор максимального значения положительного числа в индексной строке (то есть в нулевой строке в диапазоне столбцов с 1-го по $(n + m)$.

Математическая запись условия «БДП оптимален?» позволяет оценить полученный план с позиции его оптимальности. Математическая форма записи рассматриваемого условия

$$r_{\max} \le 0. \tag{12}$$

Проверка плана на оптимальность выполняется после выбора ключевого столбца (подзадача 2). План считается оптимальным при условии, если в индексной строке нет ни одного положительного числа, то есть при $r_{\max} \le 0$.

Поиск ведущей строки. Проверка задачи на разрешимость

Поиск ведущей строки осуществляется в следующей последовательности:

1) исключаются из рассмотрения строки, у которых в ключевом столбце нулевые или отрицательные элементы;

2) определяются частные от деления для остальных строк. В числителе указываются элементы, находящиеся в последнем столбце таблицы (столбец $(n+m+1)$), а в знаменателе – положительные элементы ключевого столбца этих же строк;

3) выбирается наименьшее число из полученных частных от деления;

4) объявляется ведущей строка, у которой полученное значение частного от деления является наименьшим (строке присваивается индекс v).

Элемент таблицы, расположенный на пересечении ключевого столбца и ведущей строки, называют *генеральным* (разрешающим) элементом с обозначением $S_{(v,k)}$.

Математическая модель поиска ведущей строки представлена формулами:

$$Z_{(i)} = \frac{S_{(i,n+m+1)}}{S_{(i,k)}} \, при \, S(i,k) > 0 \, и \, 1 \le i \le m, \tag{13}$$

$$Z_{min} = \frac{S_{(v,n+m+1)}}{S_{(v,k)}} = \min\left\{z_{(i)}\right\} при \, 1 \le i \le m, \tag{14}$$

где v – номер ведущей строки; k – номер ключевого столбца.

В формуле (13) в числителе указывается элемент, находящийся в столбце $(n+m+1)$, а в знаменателе – положительный элемент, находящийся в ключевом столбце той же строки. В формуле (14) осуществляется процесс поиска минимального значения частного от деления $z_{\min}$, а соответствующие ему числитель $S_{(v,n+m+1)}$ и знаменатель $S_{(v,k)}$ содержат индекс ведущей строки v.

Математическая запись условия «Задача неразрешима?». Проверка задачи на разрешимость выполняется после поиска ведущей строки (*подзадача 3*). Если получено значение хотя бы одного $z_{(i)}$ (частного от деления), то задача является разрешимой и выполняется пересчет симплексной таблицы (*подзадача 4*). В противном случае – задача неразрешима, и тогда выдается сообщение «Задача неразрешима», и решение задачи прекращается.

Пересчет симплексной таблицы

Предназначение пересчета симплексной таблицы заключается в замене в базисе базисной неизвестной на свободную неизвестную и во внесении изменений в симплексную таблицу, то есть в пересчете всех ее элементов.

Для улучшения БДП необходимо производить пересчет полученных симплексных таблиц до тех пор, пока не будет получен оптимальный план. Пересчитываемые значения элементов таблицы помещаются в ту же таблицу на место старых.

Математическая модель пересчета симплексной таблицы представляет собой следующую систему формул:

$$s_{(i_9 j)} = \begin{cases} k \text{ при } i = v \text{ и } j = 0 & (15) \\ \dfrac{s_{(i,j)}}{s_{(v,k)}} \text{ при } i = v,\ 1 \le i \le (n+m+1),\ j \ne k & (16) \\ s_{(i_9 j)} \dfrac{s_{(v,j)} * s_{(i,k)}}{s_{(v,k)}} \text{ при } 0 \le i > m,\ i \ne v \text{ и } 1 \le i \le (n+m+1),\ j \ne k, & (17) \end{cases}$$

где v – номер ведущей строки; k – номер ключевого столбца.

Анализ математической модели показывает, что необходимо учитывать особенность пересчета пяти видов данных, к которым относятся:

- индекс ключевого столбца k;
- элементы ведущей строки;
- элементы ключевого столбца;
- генеральный элемент;
- остальные элементы таблицы, не вошедшие в перечень четырех предыдущих перечислений.

Поэтому алгоритм пересчета состоит из пяти этапов:

1) внесение изменений в нулевой столбец таблицы по формуле (15);

2) пересчет всех остальных элементов таблицы за исключением элементов ключевого столбца и ведущей строки, по формуле (17);

3) пересчет по формуле (16) элементов ведущей строки, кроме генерального элемента $S_{(v,k)}$;

4) пересчет по формуле (17) элементов ключевого столбца за исключением генерального элемента (в результате пересчитанные элементы получают нулевые значения, то есть $S_{(i,k)} = 0$);

5) пересчет генерального элемента $S_{(v,k)}$ по формуле (16), в результате которого $S_{(v,k)} = 1$.

Формирование и выдача на печать выходного документа

Для вывода на печать значений всех неизвестных (базисных и свободных) используется вектор $\left\{X_{[b]}\right\}_{b=\overline{1,n+m}}$ (его краткая запись X). Как видно из полной записи вектора X, для всех элементов, имеющих конкретные индексы, обобщающим является индекс b.

Последовательность действий следующая:

а) получить итоговую симплексную таблицу с оптимальным планом;

б) обнулить вектор $\left\{X_{[b]}\right\}_{b=\overline{1,n+m}}$;

в) присвоить элементам вектора X положительные значения, полученные в столбце $(n+m+1)$ итоговой таблицы;

г) выдать на печать значения всех неизвестных.

Математическая модель, предназначенная для создания вектора X (его полная запись $\left\{X_{[b]}\right\}_{b=\overline{1,n+m}}$), представлена формулами

$$x_{(b)} = \begin{cases} S_{(i,n+m+1)}, если\, b = S_{(i,0)}\; и\;\; 1 \le i \le m \\ 0 - в\;\; остальных\;\; случаях \end{cases}, \qquad (18)$$

где $b = 1, 2, \ldots, (n + m)$.

Математическая модель, предназначенная для создания вектора H:

$$S_{(i)} = \sum\nolimits_{j=1}^{n} \left(a_{(i,j)} \cdot X_{(j)} \right) \text{при} 1 \le i \le m \;\; \text{и} 1 \le j \le n. \qquad (19)$$

Заключение

Таким образом, программное обеспечение с подробной открытой проектной документацией, в которой программная документация является составной частью, может расцениваться в качестве разновидности образцов проектирования, позволяющих достаточно просто выявить как достоинства, так и недостатки выполненных проектов.

Математическая модель задачи, как правило, не является точным и безотказным предписанием действий, которое необходимо для ЭВМ. Таким предписанием для ЭВМ является программа, составленная на основе математических моделей и разработанной схемы алгоритма решения задачи. Необходимость использования схемы алгоритма объясняется тем, что в ней, наряду с записью выполняемых действий, четко определены все логические связи (наиболее слабое звено при составлении программы для ЭВМ). При практическом использовании математической модели необходимо проводить ее тестирование.

Литература

1. *Шалыто А.А.* Новая инициатива в программировании. Движение за открытую проектную документацию // Информационно-управляющие системы. 2003. № 4 (5). С. 52–56. EDN KYOJYX.

2. *Лившиц М.В., Шестопалов М.Ю.* Цикл безопасной разработки при проектировании программного обеспечения информационных систем // Проектирование и обеспечение качества информационных процессов и систем : Сборник докладов Международной конференции. Санкт-Петербург, 15–17 марта 2022 г. СПб. : ЛЭТИ, 2022. С. 128–131. EDN KIGEOL.

3. *Соколинский Л.Б., Соколинская И.М.* О новой версии апекс-метода для решения задач линейного программирования // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2023. Т. 12. № 2. С. 5–46. EDN QKTRMC. DOI: 10.14529/cmse230201

4. *Ольховский Н.А.* Исследование нейросетевого метода решения задач линейного программирования // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2023. Т. 12. № 4. С. 55–75. EDN AQKNZA. DOI: 10.14529/cmse230402

5. *Усатюк В.С., Егоров С.И.* Поиск треппин-сетов методом смешанного целочисленного линейного программирования с использованием априорного списка кодовых вершин // Известия Юго-Западного государственного университета. 2023. Т. 27. № 4. С. 79–97. EDN RFTGRE. DOI: https://doi.org/10.21869/2223-1560-2023-27-4-79-97

6. *Полежаев С.В.* Симплекс-метод: основные идеи // Современные информационно-коммуникационные технологии. 2022. № 12. С. 44–46. EDN ESCQMQ.

7. *Вихарев Н.А.* Использование симплекс-метода для оптимизации переработки никеля с помощью цифрового двойника // Ceteris Paribus. 2022. № 12. С. 9–11. EDN UPTZPD.

8. *Султанов А.Т.* Применение симплекс-метода для решения задач инженерной оптимизации // Уральский научный вестник. 2023. Т. 10. № 7. С. 8–15. EDN YXTKNL.

9. *Галкин В.А., Кузина Е.Л., Кузина М.А., Василенко Е.А.* Применение симплекс-метода в повышении экологичности производства на транспортных предприятиях // Качество. Инновации. Образование. 2024. № 1 (189). С. 26–33. EDN AFVYFG. DOI: 10.31145/1999-513x-2024-1-26-33

References

1. Shalyto A.A. (2003) A New Initiative in Programming. Movement for Open Design Documentation. *Information and Control Systems*. No. 4 (5). Pp. 52–56. (In Russian).
2. Livshits M.V., Shestopalov M.Yu. (2022) The Secure Development Cycle in the Design of Information Systems Software. In: *Proektirovanie i obespechenie kachestva informatsionnykh protsessov i sistem* [Design and Quality Assurance of Information Processes and Systems] : Proceedings of the International Conference. Saint Petersburg, March 15–17, 2022 St. Petersburg : Saint-Petersburg Electrotechnical University Publ. Pp. 128–131. (In Russian).
3. Sokolinsky L.B., Sokolinskaya I.M. (2023) On New Version of the Apex Method for Solving Linear Programming Problems. *Bulletin of South Ural State University. Series: Computational Mathematics and Computer Science*. Vol. 12. No. 2. Pp. 5–46. DOI: 10.14529/cmse230201 (In Russian).
4. Olkhovsky N.A. (2023) Study of Neural Network Nodels for Linear Programming. *Bulletin of South Ural State University. Series: Computational Mathematics and Computer Science.* Vol. 12. No. 4. Pp. 55–75. DOI: 10.14529/cmse230402 (In Russian).
5. Usatyuk V.S., Egorov S.I. (2003) Trapping Sets Search Using the Method of Mixed Integer Linear Programming with a Priori List of Variable Nodes. *News of Southwestern State University.* Vol. 27. No. 4. Pp. 79–97. (In Russian).
6. Polezhaev S.V. (2022) Simplex-Method: Basic Ideas. *Sovremennye informatsionno-kommunikatsionnye tekhnologii* [Modern Information and Communication Technologies]. No. 12. Pp. 44–46. (In Russian).
7. Vikharev N.A. (2022) Using the Simplex Method to Optimize Nickel Processing Using a Digital Twin. *Ceteris Paribus.* No. 12. Pp. 9–11. (In Russian).
8. Sultanov A.T. (2023) Application of the Simplex Method for Solving Engineering Optimization Problems. *Ural'skii nauchnyi vestnik* [Ural Scientific Bulletin]. Vol. 10. No. 7. Pp. 8–15. (In Russian).
9. Galkin V.A., Kuzina E.L., Kuzina M.A., Vasilenko E.A. (2024) The Simplex Method Application in Increasing the Production Environmental Friendliness at Transport Enterprises. *Kachestvo. Innovatsii. Obrazovanie* [Quality. Innovation. Education]. No. 1 (189). Pp. 26–33. DOI: 10.31145/1999-513x-2024-1-26-33 (In Russian).