ния благодаря добавлению дополнительных критериев оценки свободы и качества расположения групп судов в расписании.

Корректность данного алгоритма была проверена с помощью метода верификации на модели. Были составлены модель алгоритма и набор спецификаций, которым он дожен соответствовать. И с помощью системы NuSMV была произведена проверка предложенного алгоритма на корректность.

Список литературы

1. *Чебатуркин А. А.* Методы верификации конечных автоматов, взаимодействующих по акторной модели / А. А. Чебатуркин, М. А. Мазин. — СПб.: СПбИТМО, 2010.

2. NuSMV User Manual [Электронный ресурс]. — Электрон. дан. — Режим доступа: http://nusmv.irst.itc.it/NuSMV/userman/index-v2.html

3. Правила пропуска судов и составов через шлюзы внутренних водных путей Российской Федерации: утв. приказом Минтранса РФ от 24 июля 2002 г. № 100.

4. *Кононов В. В.* Гидротехнические сооружения водных путей, портов и континентального шельфа (Судоходный канал и бетонный шлюз) / В. В. Кононов. — СПб.: СПбГУВК, 2009.

УДК 621.396

А. А. Чертков,
канд. техн. наук, доцент
ГУМРФ имени адмирала С. О. Макарова;

Д. А. Загрединов,
аспирант,
ГУМРФ имени адмирала С. О. Макарова;

Ю. Б. Михайлов,
аспирант,
ГУМРФ имени адмирала С. О. Макарова

МОДЕЛЬ НЕЛИНЕЙНОЙ ЛОГИСТИЧЕСКОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ ПЕРЕГРУЗОЧНОГО ПРОЦЕССА

MODEL OF NONLINEAR LOGISTICS SYSTEM OF AUTOMATION OF HANDLING PROCESS

В статье рассматривается модель и оптимизация решения нелинейной многомерной транспортной задачи, базирующейся на численных методах квадратичного программирования, реализуемого в вычислительной среде MATLAB. Достоверность результатов численного моделирования подтверждается экспериментом. По алгоритму произведен синтез логистической системы автоматизации перегрузочного процесса, обеспечивающей минимум транспортных расходов на управление ресурсами. Приведен пример расчета системы.

The article is devoted to the model and optimization of the solution of the nonlinear multidimensional transport problem which is based on numerical quadratic programming methods, which is implemented in the MatLAB computing codes. Numerical modeling high quality is confirmed by experiment. The automation logistic system synthesis is devoted to reloading process providing transportation minimum for resource costs. The example of system calculation is given.

Ключевые слова: алгоритм, модель, моделирование, численные методы, квадратичное программирование, транспортная задача, оптимальное решение, достоверность, логистическая система, автоматизация перегрузочного процесса.

Key words: algorithm, model, modeling, numerical methods, the quadratic programming, the transport task, optimal solution, reliability, logistics system, automation of the reloading process.

В ПОСЛЕДНЕЕ время в исследованиях транспортных потоков наряду с линейными методами математического программирования все шире стали применять модели и алгоритмы нелинейной динамики. Целесообразность их применения обоснована наличием в транспортном потоке устойчивых и неустойчивых режимов движения, потерь устойчивости при изменении условий движения, нелинейных обратных связей, необходимости в большом числе переменных для адекватного описания системы.

Наибольшее применение в таком классе задач нашли методы линейного программирования, которые имеют много эффективных алгоритмов решения с помощью вычислительных средств. Но эти методы неприменимы, если функции стоимости нелинейны, за исключением тех случаев, когда они обладают определенными простыми структурными свойствами, дающими возможность тем или иным способом эффективно использовать линейную аппроксимацию.

Другими известными численными методами решения такого класса задач являются динамическое программирование, градиентные методы, методы, базирующиеся на генетических алгоритмах и др.

Динамическое программирование представляет собой особый математический метод оптимизации решений, специально приспособленный к многошаговым (или многоэтапным) операциям. Динамическое программирование (ДП) начало развиваться в 1950-х гг. благодаря работам Р. Беллмана. В основе метода ДП лежит принцип оптимальности, сформулированный Р. Беллманом. Этот принцип и идея трансформации сложной задачи оптимизации в семейство аналогичных многошаговых простых задач приводят к рекуррентным соотношениям и функциональным уравнениям для оптимизации целевой функции. Рекуррентные соотношения позволяют последовательно получить решение для исходной задачи оптимизации.

Вместе с тем динамическому программированию свойственны и недостатки. Прежде всего в нем нет единого универсального метода решения. Практически каждая задача, решаемая этим методом, характеризуется своими особенностями и требует проведения поиска наиболее приемлемой совокупности алгоритмов для ее решения. Кроме того, большие объемы и трудоемкость решения многошаговых задач, имеющих множество состояний, приводят к необходимости отбора задач малой размерности либо использования сжатой информации. А это достигается с помощью методов анализа вариантов и перебора списка состояний. Последний характерный для этого метода недостаток чаще упоминается в литературе как «проклятие размерности».

Используя градиентные методы, можно найти решение широкого спектра задач нелинейного программирования. Применение этих методов в общем случае позволяет найти точку локального экстремума. Поэтому более целесообразно использовать их для нахождения решения задач выпуклого программирования. Процесс нахождения решения задачи с помощью градиентных методов состоит в том, что, начиная с некоторой точки x_k, осуществляется последовательный переход к некоторым другим точкам до тех пор, пока не будет найдено приемлемое решение исходной задачи. Градиентные методы могут быть подразделены на две группы.

К первой группе относятся методы, при использовании которых исследуемые точки не выходят за пределы области допустимых решений задачи. В данном случае наиболее распространенным является метод Франка–Вульфа. Ко второй — методы, при использовании которых исследуемые точки могут как принадлежать, так и не принадлежать области допустимых решений. Однако в результате реализации итерационного процесса находится точка области допустимых решений, определяющая приемлемое решение. Наиболее часто используются метод штрафных функций и метод Эрроу–Гурвица. При нахождении решения задачи градиентными методами итерационный процесс продолжается до тех пор, пока градиент функции

$$\nabla f(x_k) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n} \right)$$

в очередной точке x_{k+1} не станет равным нулю или же пока не выполнится неравенство

$$\left| f(x_{k+1}) - f(x_k) \right| < \varepsilon,$$

где $\varepsilon > 0$ (точность полученного решения).

Градиентные методы являются основными численными методами поиска оптимальных решений в многомерных задачах. На практике наиболее востребованными являются задачи поиска минимума квадратичной целевой функции, частным случаем которых являются многомерные транспортные задачи с квадратичной функцией стоимости перевозок. Рассмотрим решение многомерной и нелинейной транспортной задачи с использованием квадратичного программирования, реализуемого в среде MATLAB с применением градиентных методов.

Постановка задачи квадратичного программирования предусматривает поиск минимума квадратичной целевой функции общего вида, зависимость которой от своих аргументов определяется следующим выражением:

$$f(x) + a + \sum_{i=1}^{n} b_i x_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j \,. \tag{1}$$

Коэффициенты c_{ij} удобно считать элементами симметричной матрицы:

$$\begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{pmatrix}. \tag{2}$$

Можно показать, что у квадратичной целевой функции (1) имеется минимум, причем единственный, в том и только в том случае, когда матрица (2) является положительно определенной (это условие в дальнейшем мы будем предполагать выполненным).

В случае квадратичной целевой функции (1) легко вычисляются компоненты градиента:

$$\frac{\partial f}{\partial x} = b_i + \sum_{j=1}^{n} c_{ij} x_j, \quad i = 1, 2, \ldots, n, \tag{3}$$

а матрица Гессе при этом не зависит от выбора точки x и совпадает с матрицей коэффициентов (2):

$$H_{ij}(x) = \frac{\partial^2 f}{\partial x_i \partial x_j} = c_{ij}, \quad i, j = 1, 2, \ldots, n. \tag{4}$$

Формулами (3) и (4) можно воспользоваться при определении направления поиска как в методе наискорейшего спуска, так и в методе Ньютона.

Приведем теперь без доказательства общую формулу, позволяющую для произвольной квадратичной целевой функции (1) найти величину шага, обеспечивающую минимум функции при любом выборе направления поиска:

$$h = -\sum_{i=1}^{n} u_i \left(b_i + \sum_{j=1}^{n} c_{ij} x_j \right) / \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} u_i u_j. \tag{5}$$

В методе наискорейшего спуска общая формула (5) для величины шага заметно упрощается:

$$h = \sum_{i=1}^{n} u_i^2 \Big/ \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} u_i u_j. \qquad (6)$$

В методе Ньютона вычисленный в соответствии с (5) шаг оказывается равным единице:

$$h = 1.$$

Это означает, что в случае квадратичных целевых функций метод Ньютона сразу же за один шаг приводит к цели.

В вычислительной среде MATLAB задача квадратичного программирования решается с помощью функции *quadprog*, которая имеет синтаксис

$$x = quadprog\ (H, f, A, b, Aeq, beq, lb, ub),$$

где H, A и Aeq есть матрицы, а f, b, beq, lb, ub и x есть векторы.

В соответствии с указанным синтаксисом функция *quadprog* имеет следующее описание:

— вычисляет вектор x, который минимизирует целевую функцию $f(x) = \min_x \frac{1}{2} x^T Hx + f^T x$, при условии выполнения ограничения неравенства $A \cdot x \le b$;

— решает указанную выше задачу с дополнительным выполнением ограничений типа равенства $Aeq \cdot x = beq$;

— определяет набор нижних и верхних границ для переменной x, так чтобы решение находилось в диапазоне $lb \le x \le ub$.

Первые пять параметров функции — входные (их значения должны задаваться), вектор x — выходной параметр (должен вычисляться при решении задачи). В матричном виде квадратичная часть целевой $f(x)$ функции запишется так:

$$f(x) = 0{,}5 \cdot x' \cdot H \cdot x,$$

где H есть матрица Гессе. Элементами матрицы H являются коэффициенты при переменной x второго порядка. Линейная часть функции $f(x)$ равна $f'' \cdot x$, где координатами вектора f являются коэффициенты при переменной x первого порядка. Свободные члены целевой функции не включаются в обращение к целевой функции *quadprog*, поэтому они учитываются отдельно.

Покажем возможность использования функции *quadprog* для решения многомерной транспортной задачи с квадратичной целевой функцией и сравним полученные результаты с результатами решения той же задачи методом множителей Лагранжа [1], которые приведены в табл. 1.

Таблица 1

Оптимальное решение многомерной транспортной задачи из [1]

Потребители	Склад 1	Склад 2	Склад 3	Издержки	Накопленные издержки
1	25	0	0	25,00	25,00
2	40	0	0	81,00	106,00
3	5	55	0	130,75	236,75
4	0	0	30	30,00	266,75
5	0	0	20	20,00	286,75
6	0	0	30	65,00	351,75
7	30	0	5	110,00	461,75
8	0	0	30	96,00	557,75
9	0	25	0	50,00	607,75
10	0	0	40	240,00	847,75

Как следует из [1], поиск решения производился для транспортной задачи с тремя складами и десятью пунктами потребления, в которой функция стоимости перевозок наряду с постоянной и линейной составляющими включала и квадратичную составляющую.

Исходные данные (табл. 2) те же, что использовались при решении этой задачи методом множителей Лагранжа. Приведем пояснения к задаче [1].

Таблица 2

Показатели стоимости перевозок

Потребители	Из склада 1			Из склада 2			Из склада 3			Спрос r_i
	a_{1j}	b_{1j}	c_{1j}	a_{2j}	b_{2j}	c_{2j}	a_{3j}	b_{3j}	c_{3j}	
1	1,0			3,1		2,0	7,0			25
2	2,0		1,0	4,1			3,0			40
3	3,0	0,01		2,1			9,0			60
4	1,5			1,1	0,1		1,0			30
5	2,5			2,6			1,0			20
6	5,0	–0,01	10,0	3,0			2,0		5,0	30
7	3,0			1,0	0,2	5,0	4,0			35
8	6,0			2,0			3,0		6,0	30
9	6,0	–0,05	8,0	2,0			5,0			25
10	6,0			5,0	0,01		6,0			40

Будем считать, что стоимость перевозки представляет собой квадратичную функцию:

$$g_{ij}(x)=a_{ij}x + b_{ij}x^2 + c_{ij}(x) \qquad (7)$$

от перевозимого количества x плюс «организационные» расходы.

Под «организационными» расходами $c_{ij}(x)$ понимаются затраты, не зависящие от перевозимого количества и равные нулю, когда перевозка не производится.

Предполагается, что надо перевезти 100 единиц со склада 1, 80 единиц со склада 2 и 155 единиц со склада 3. Таким образом, суммарные запасы на складах, равные 335 единиц, равны суммарному спросу потребителей, равному также 335 единиц, то есть

$$x_1 + x_2 + x_3 = \sum_{i=1}^{N=10} r_i .$$

Тогда задача поиска оптимального решения состоит в определении величин x_{ij}, удовлетворяющих условиям:

$$\begin{cases} x_{ij} \ge 0, \\ \sum_{j=1}^{N} x_{ij} = x_i, \quad i = 1, 2, ..., M, \\ \sum_{i=1}^{M} x_{ij} = r_j, \quad j = 1, 2, ..., N, \end{cases}$$

так, чтобы минимизировать общую стоимость перемещения ресурсов:

$$C_{MN} = \sum_{i=1}^{M}\sum_{j=1}^{N} g_{ij}(x_{ij}) .$$

Пример программы решения транспортной задачи с тремя складами и десятью пунктами потребления с применением квадратичного программирования представлен файлом sah799.m, содержание которого поясняется комментариями.

```
% sah799.m
% Распределение ресурсов.
% Три пункта отправления (склада), десять пунктов потребления.
% Матрица коэффициентов aij линейной части f'x целевой функции
AA=[1.0 3.1 7.0;2.0 4.1 3.0;3.0 2.1 9.0;1.5 1.1 1.0;2.5 2.6 1.0;
   5.0 3.0 2.0; 3.0 1.0 4.0;6.0 2.0 3.0;6.0 2.0 5.0; 6.0 5.0 6.0];
% Вектор организационных расходов (из коэффициентов cij):
M=[0 1 0 0 0 10 0 0 8 0 2 0 0 0 0 0 5 0 0 0 0 0 0 0 0 5 0 6 0 0];
% Формирование вектора f линейной части целевой функции
f=[AA(:,1);AA(:,2);AA(:,3)]';
% Вектор-строка коэффициентов bij, пропорциональных квадрату переменных x
P=[0 0 0.01 0 0 -0.01 0 0 -0.05 0 0 0 0 0.1 0 0 0.2 0 0 0.01 zeros(1,10)];
% Диагональная матрица Гессе
H=2*diag(P);
% Ограничения - равенства:
Aeq=[eye(10) eye(10) eye(10)];
beq=[25 40 60 30 20 30 35 30 25 40]';
% Ограничения - неравенства:
A=[ones(1,10) zeros(1,10) zeros(1,10);zeros(1,10) ones(1,10) zeros(1,10)];
b=[90 100]';
% Граничные условия:
lb=zeros(30,1); ub=[];
%opt=optimoptions('quadprog');
% Расчет экономичного плана:
[x,fval]=quadprog(H,f,A,b,Aeq,beq,lb,ub);
% Учет аддитивной составляющей организационных расходов:
Y1=(x>0);
J1=fval+M*Y1;
% Расчетное распределение товара по пунктам потребления:
C=[x(1:10) x(11:20) x(21:30)];
Y=(C>0);
Qv=[P(1:10)' P(11:20)' P(21:30)'];
Qv1=Qv.*[x(1:10).^2 x(11:20).^2 x(21:30).^2];
L=AA.*C+[M(1:10)' M(11:20)' M(21:30)'].*Y+Qv1;
JJ=[L(:,1) L(:,2) L(:,3)]
J2=L(:,1)+L(:,2)+L(:,3);
J3=cumsum(J2);
Cost=[JJ J3]
J4=sum(J2)
% Проверка результатов расчета по J1 и J4:
Rez=[J1 J4]
```

Результатами вычислительного эксперимента являются следующие:

```
x =
   25.0000           0.0000 0.0000
   35.0000           0.0000 5.0000
   -0.0000           60.0000      0.0000
   -0.0000           0.0000 30.0000
    0.0000           -0.0000      20.0000
   -0.0000           0.0000 30.0000
```

```
    30.0000           5.0000 0.0000
     0.0000           10.0000      20.0000
     0.0000           25.0000      0.0000
     0.0000           0            40.0000
Cost =
    25.0000    2.0000    0.0000   27.0000
    71.0000    0.0000   15.0000  113.0000
    -0.0000  126.0000    0.0000  239.0000
    -0.0000    0.0000   30.0000  269.0000
     0.0000   -0.0000   20.0000  289.0000
    -0.0000    0.0000   65.0000  354.0000
    90.0000   15.0000    0.0000  459.0000
     0.0000   20.0000   66.0000  545.0000
     8.0000   50.0000    0.0000  603.0000
     0.0000         0  240.0000  843.0000
Rez =
  843.0000
```

На основе представленного вычислительного алгоритма разработана программа в кодах MATLAB, позволяющая оперативно решать нелинейные логистические задачи с изменяющимися ограничениями при соблюдении логических условий функционирования технологических объектов.

Список литературы

1. *Беллман Р.* Прикладные задачи динамического программирования / Р. Беллман, С. Дрейфус. — М.: Наука, 1965.

2. Математическое программирование в примерах и задачах: учеб. пособие / под ред. И. Л. Акулич. — М.: Высш. шк., 2003. — 320 с.

3. *Ашманов А. Ф.* Численные методы оптимизации / А. Ф. Ашманов, М. В. Соловьев. — М.: Физматгиз, 2008. — 320 с.

4. *Венцель Е. С.* Исследование операций / Е. С. Венцель. — М.: Сов. радио, 2004. — 550 с.

5. *Солодовников А. С.* Задача квадратичного программирования / А. С. Солодовников. — М.: Финансовая академия, 2004. — 397 с.