

рых характеризует состояние объекта: $x^0 = (x_1^0, x_2^0, \dots, x_m^0)$, $x_i^0 \in [0,1]$, $i = \overline{1, m}$. Пусть задано некоторое разбиение множества X на классы $\{X_1, X_2, \dots, X_n\}$, где $X_i \cap X_j = \emptyset$ при $i \neq j$ и $X = \bigcup_{i=1}^n X_i$. Каждому классу X_j , $j \in \{1, \dots, n\}$ соответствует эталонный вектор $(x_1^j, x_2^j, \dots, x_m^j)$, $x_i^j \in [0,1]$, $i = \overline{1, m}$ и управляющее воздействие y_j .

Пусть на вход поступает объекта $z = (z_1, \dots, z_m)$. Далее отображаем значение каждого параметра на отрезок $[0,1]$, тем самым получаем вектор x^0 , затем находим эталонный вектора x^j , ближайший к x^0 . Полученный результат интерпретируется в терминах принадлежности объекта z к классу X_j , $j \in \{1, \dots, n\}$. Далее к объекту z применяем управляющее воздействие y_j (например, сделать операцию или направить на томографию).

Данный метод можно улучшить, если введем понятие *бифуркации* – некоторое пороговое значения параметра, которое однозначно определяет класс, которым описывается текущее состояние объекта. Например, температура тела человека выше 37°C однозначно может определять наличие инфекции в организме.

Литература

1. Интернет-адрес:
<http://elib.bsu.by/bitstream/123456789/7664/1/Методы%20пол%20и%20обработ%20изображений%20в%20мед%20сист.pdf>.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М., 2005.

ВОПРОСЫ РЕАЛИЗАЦИИ ОПОРНЫХ МЕТОДОВ РЕШЕНИЯ ДИНАМИЧЕСКИХ ТРАНСПОРТНЫХ СЕТЕВЫХ ЗАДАЧ

С. Н. Костюкович

ВВЕДЕНИЕ

Транспортные задачи линейного программирования представляют собой математические модели различных важных практических задач. Актуальным является умение решать задачи в сетевой постановке, так как именно транспортные сети наиболее приближены к реальным коммуникационным процессам.

В реальной жизни часто возникают динамические транспортные сетевые задачи больших размеров. По этой причине для их решения необходимо разрабатывать эффективные программные реализации, использующие последние достижения в области программирования и вычислительной техники.

В данной работе разработана программная реализация решения динамической транспортной сетевой задачи адаптивным методом на объектно-ориентированном языке Java.

ПОСТАНОВКА ЗАДАЧИ

Рассмотрим постановку динамической транспортной сетевой задачи [1].

На нестационарной сети $S(t) = \{I, U(t)\}$, изменяющей свои параметры в дискретные моменты времени $t \in T = \{0, 1, 2, \dots, t_1 - 1\}$, рассмотрим следующую динамическую транспортную задачу:

$$\sum_{t \in T} \left(\sum_{(i,j) \in U(t)} c_{ij}(t) x_{ij}(t) + \sum_{i \in I} c_i(t+1) x_i(t+1) \right) \rightarrow \min,$$

$$x_i(t+1) = x_i(t) - \sum_{j \in I_i^+(U(t))} x_{ij}(t) + \sum_{j \in I_i^-(U(t))} x_{ij}(t) + a_i(t)$$

$$0 \leq x_i(t+1) \leq d_i(t+1), x_i(0) = 0, i \in I;$$

$$0 \leq x_{ij}(t) \leq d_{ij}(t), (i,j) \in U(t), t \in T.$$

Здесь $x_i(t)$ – запас некоторого продукта в узле i в момент t ; $a_i(t)$ – интенсивность узла i в момент t ($a_i(t)$ – объем производства, если $a_i(t) > 0$, $a_i(t)$ – объем потребления, если $a_i(t) < 0$). $x_{ij}(t)$ – объем перевозки из узла i в узел j ; $c_{ij}(t)$ – стоимость перевозки единицы продукта из узла i в узел j ; $c_i(t)$ – стоимость хранения единицы продукта в узле i ; $d_{ij}(t)$ – пропускная способность коммуникации $i \rightarrow j$; $d_i(t)$ – объем склада в узле i ; $I_i^+(U) = \{j \in I : (i,j) \in U\}$, $I_i^-(U) = \{j \in I : (j,i) \in U\}$.

Динамическую транспортную сетевую задачу представим в виде задачи оптимального управления [2]:

$$J(u) = \sum_{t \in T} (c'_u(t) u(t) + c'_x(t+1) x(t+1)) \rightarrow \min,$$

$$x(t+1) = x(t) - B(t)u(t) + a(t),$$

$$x(0) = 0, u(t) \in V(t) = \{u : 0 \leq u \leq d_u(t)\},$$

$$0 \leq x(t+1) \leq d_x(t+1), t \in T,$$

где $B(t) = \{b^{ij}(t), (i, j) \in U(t)\}$ есть $|I| \times |U(t)|$ -матрица, задаваемая сетью $S(t)$, $b^{ij}(t)$ – $|I|$ -вектор с компонентами $b^{ij}(t) = 0$, если $k \neq i, k \neq j, b^{ij}(t) = 1, b^{ij}(t) = -1$; $x(t) = \{x_i(t), i \in I\}$ – вектор состояния системы, $u(t) = \{u_{ij}(t), (i, j) \in U(t)\}$ – вектор управления; $d_x(t) = \{d_i(t), i \in I\}, c_u(t) = \{c_{ij}(t), (i, j) \in U(t)\}, c_x(t) = \{c_i(t), i \in I\}, d_u(t) = \{d_{ij}(t), (i, j) \in U(t)\}$

ВОПРОСЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ АДАПТИВНОГО МЕТОДА РЕШЕНИЯ ДИНАМИЧЕСКИХ ТРАНСПОРТНЫХ СЕТЕВЫХ ЗАДАЧ

Описание динамической транспортной сетевой задачи в терминах объектной модели

Объектная модель описывает структуру объектов, составляющих систему, их атрибуты, операции, взаимосвязи с другими объектами [3].

Будем называть объектом понятие, абстракцию или любую вещь с четко очерченными границами, имеющую смысл в контексте рассматриваемой прикладной проблемы.

Цель разработки объектной модели – описать объекты, составляющие в совокупности проектируемую систему, а также выявить и указать различные зависимости между объектами [3].

Для транспортной сетевой задачи в качестве основных объектов рассмотрим:

- Вершина
- Дуга
- Дерево
- Поток
- Сеть

Для программной реализации воспользуемся языком Java. Таким образом, основные объекты будут реализованы в специальных пользовательских типах данных – классах. В классе дается обобщенное описание некоторого набора родственных, реально существующих объектов. Объект – конкретный экземпляр класса [4].

Иерархия классов

Рассмотрим иерархию классов программной реализации адаптивного метода решения динамических транспортных сетевых задач.

Программа состоит из следующих блоков:

- InputOutput
- Main
 - Model
 - Net
 - Util
 - NetworkFlowProblem
- Exception
- Test

InputOutput – модули ввода данных и вывода полученного решения. Они представляют интерфейсы, которые реализованы в классах, предназначенных для конкретных форматов входной и выходной информации (txt, csv и т.п.).

Main – модуль, реализующий логику приложения, то есть модуль, в котором происходит непосредственно решение задачи.

Exception – модуль обработки исключительных ситуаций завершения работы программы (некорректно заданные начальные условия и т.п.).

Test – модуль тестирования (запуска) приложения. Он может быть заменен, к примеру, на вызов с сайта или из другой программы.

Численные эксперименты

Для проведения численных экспериментов в первую очередь необходимо решить вопрос генерации динамических сетей.

Генерация входных данных для сетевых транспортных задач представляет собой более сложную задачу в сравнении с генерацией входных данных для обычной задачи линейного программирования.

В данной работе для генерации входных данных была использована реализация программы NETGEN, разработанной и описанной D.Klingman, A. Napier, J. Stutz в 1973 году [5].

Приведем таблицу результатов проведенных численных экспериментов:

Результаты численных экспериментов

Эксперимент	Количество периодов времени	Количество узлов	Количество дуг	Параметры расширенной сети $ I \times U $	Количество итераций
1	3	5	8	15x34	11
2	3	8	10	24x46	6
3	3	10	30	39x110	53

Полученные результаты

Таким образом, динамическая транспортная сетевая задача может быть сведена к динамической задаче оптимального управления и решена адаптивным методом [1]. Достоинством данного подхода является то, что для каждого момента времени мы имеем отдельный опорный поток, а значит, при программной реализации метода, можем уменьшить количество обращений к внешней памяти и хранить все необходимые данные в оперативной памяти, в силу того, что решаем задачу меньшей размерности. Такой подход должен быть особенно эффективен для задач с большим числом периодов времени. Недостаток этого метода состоит в значительном увеличении количества накладных расходов.

В данной работе разработана программная реализация решения динамических транспортных сетевых задач на языке Java. Проведено описание исходной задачи в терминах объектной модели, проектирование программы, построение программного кода и тестирование разработанного приложения.

Литература

3. Габасов Р.Ф. Методы линейного программирования. Часть 2. Транспортные задачи / Р.Ф. Габасов, Ф.М. Кириллова. Минск: изд-во БГУ им.Ленина, 1984. 214 с.
4. Габасов Р. Конструктивные методы оптимизации / Р. Габасов, Ф.М. Кириллова, А.И. Тятюшкин. Минск: изд-во «Университетское», 1978.
5. Миста // Объектная модель системы [Электронный ресурс] / Режим доступа: http://www.mista.ru/oop_book/glava2_1.htm. Дата доступа: 22.05.2013.
6. Блинов И. Н. Java2. Практическое руководство / И.Н. Блинов, В.С. Романчик. Минск: «УниверсалПресс», 2005.
7. Klingman D. NETGEN: A Program for Generating Large Scale (UN) Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems / D. Klingman, A. Napier, J. Stutz. // Management Science. 1974, №20. С. 814–822.