Литература

1. *Корбут, А. А.* Дискретное программирование /А. А. Корбут, Ю. Ю. Финкельштейн, Д. Б. Юдин. - М. : Наука, 1969. - 368 с.

2. *Скобелев, В. Г.* Дискретная оптимизация / В. Г. Скобелев, В. Г. Христиановский В. В. - Киев : УМК ВО, 1988. – 63 с.

3. *Финкельштейн, Ю. Ю.* Приближенные методы и прикладные задачи дискретного программирования / Ю. Ю. Финкельштейн. – М. : Наука, 1976. – 264 с.

Цехан Ольга Борисовна, доцент кафедры математического и информационного обеспечения экономических систем Гродненского государственного университета имени Янки Купалы, кандидат физико-математических наук, tsekhan@grsu.by

Кизер Руслан Владимирович, студент 4-го курса факультета математики и информатики Гродненского государственного университета имени Янки Купалы, ruskiz@gmail.com

Севко Андрей Юрьевич, студент 4-го курса факультета математики и информатики Гродненского государственного университета имени Янки Купалы, sevoch4a1@mail.ru

Гуринович Анастасия Сергеевна, студентка 4-го курса факультета математики и информатики Гродненского государственного университета имени Янки Купалы, kehri@mail.ru

УДК 004

О. Б. Цехан, Д. С. Шпак, Т. Г. Мазан

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ИЗУЧЕНИЯ АЛГОРИТМОВ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ НА БАЗЕ ПАКЕТА MATHEMATICA

Описана разработка и применение электронного документа с демонстрационно-обучающими элементами для изучения и реализации методов решения задач целочисленного линейного программирования. Программный модуль реализован с использованием средств пакета Mathematica.

Введение

Задачи дискретной оптимизации образуют класс математических моделей, появляющихся при исследовании разнообразных вопросов организации сложных систем и управления в них. Интерес к таким задачам возник в 50-е гг. и постоянно возрастает благодаря широким приложениям, а также развитию методов их решения и совершенствованию вычислительных средств. Задача дискретной оптимизации является задачей оптимизации функции

$$f = f(x) \tag{1}$$

на дискретном множестве X.

Среди задач дискретной оптимизации выделяют класс комбинаторных задач, в которых множество X конечно, и более узкий класс целочисленной оптимизации.

В работе рассматриваются задачи целочисленного линейного программирования (ЗЦЛП), в которых как целевая функция, так и функции в системе ограничений являются линейными. В настоящее время существует несколько специальных методов решения ЗЦЛП, из которых наиболее применяемыми являются метод ветвей и границ и метод Гомори.

Дискретные оптимизационные задачи находят широкое применение в различных областях, где используются математические методы. Необходимость решения таких задач приводит к тому, что дискретная оптимизация становится важным элементом образования специалистов. Поэтому технология решения задач дискретного программирования должна стать одной из важных составных частей современного математического образования для специалистов по прикладной математике.

В специальном курсе «Математические модели и методы решения задач оптимального планирования и управления» изучаются различные методы, в частности, метод ветвей и границ. Студенты должны знать: постановку общей задачи дискретного программирования и ее особенности: приложения методов в экономике и других областях и уметь использовать эти методы, рассмотренные в процессе обучения, к решению новых задач дискретного программирования.

Пример 2. Найти глобальный минимум функции двух переменных $f = 2x - y$ в области, ограниченной неравенствами $x - y \geq -4, x + y \leq 8, x - 2y \leq 3$.

ConstrainedMin[2x+y, {x-y≥-4,x+y≤8,x-2y≤3}, {x,y}]

ConstrainedMin :: deprec :

ConstrainedMin is deprecated and will not be supported in future versions of Mathematica . Use NMinimize or Minimize instead . More.

{0,{x→0,y→0}}

Прокомментируем отличия данных результатов. В примере 2 использовать процедуру ConstrainedMin не является правильным вариантом, потому что координаты *x* и *y* при этом должны быть неотрицательны, иначе получили неверный результат.

Впрочем, для решения задачи оптимизации можно воспользоваться и функциями Nmaximize (Nminimize).

Формат команд:

Nmaximize[{f, cons}, {x, y, ...}], Nmaximize[{f, cons}, {x, y, ...}].

Процедуры Nmaximize(Nminimize)[{*f*, cons}, {*x*, *y*, ...}] ищут приближенный максимум (минимум) функции *f* с ограничениями cons.

Функции minimize и maximize входят в одну из самых многочисленных и важных групп функций – алгебраические и символьные вычисления(algebraic computation) - в подгруппу символьное дифференцирование и интегрирование. Функции Nminimize и Nmaximize относятся к группе приближенных функций (approximate functions) для численной оптимизации. Такие функции в отличие от числовых функций заведомо используют приближенные алгоритмы и даже для точных данных никогда не пытаются вычислить точные значения. На это указывает N перед функцией.

На одном из этапов решения ЗЦЛП методом ветвей и границ необходимо выяснить, являются ли найденные значения переменных целыми. Для данной проверки можно использовать функцию Head, которая определяет, к какому типу относится переменная x.

Формат команды: Head[*x*].

Пример 3. Определить тип заданных чисел.

{Head[3], Head[1/3], Head[0.3], Head[0.3+i]}

{Integer, Rational, Real, Complex}.

При определении дополнительного ограничения нужно найти целую часть вещественной нецелочисленной координаты. Для этого служит функция Floor.

Формат команды: Floor[*x*].

Пример 4. Найти наибольшее целое число, не превышающее данного.

In. **Floor[3.9]**

Out: 3

Для формирования множества планов задачи линейного программирования и ее графического отображения в пакете Mathematica можно применять следующие команды: ImplicitPlot и InequalityPlot.

Формат команд:

InequalityPlot[ineqs, {x, xmin, xmax},{y, ymin, ymax}] – изображает и закрашивает множество, заданное неравенствами ineqn, зависящими от двух переменных *x* и *y*, причем могут задаваться максимальные и минимальные значения *x* и *y*. Перед использованием данной функции необходимо подключить соответствующую библиотеку с помощью команды **<<Graphics`InequalityGraphics`.**

ImplicitPlot[eqn, { x, xmin, xmax},{y, ymin, ymax }] – чертит прямые, заданные равенствами eqn, зависящими от двух переменных *x* и *y*, причем задаются максимальные и минимальные значения *x* и *y*. Перед использованием этой функции необходимо подключить соответствующую библиотеку с помощью команды **<<Graphics` ImplicitPlot `.**

Приведем примеры использования данных функций.

Пример 5. Построить область, задаваемую неравенствами, зависящими от двух переменных *x* и *y*: $x - y \geq -5, x + y \leq 7, x - 2y \leq 2$. Используем процедуру InequalityPlot при условии неотрицательности обеих переменных

<<Graphics` InequalityGraphics `

InequalityPlot[x-y≥-5^x+y≤7^x-2y≤2^x≥0^y≥0, {x}, {y},

PlotRange→ {{-1,6}, {-1, 6}}]

Результат построения на рис. 1.

Используем процедуру ImplicitPlot.

<<Graphics`ImplicitPlot`
p1:=ImplicitPlot[x-y==-5, {x,-1,6}, {y,-1,6}]
p2:=ImplicitPlot[x+y==7, {x,-1,6}, {y,-1,6}]
p3:=ImplicitPlot[x-2y==2, {x,-1,6}, {y,-1,6}]
Show[p1, p2, p3]

Результат построения на рис. 2.

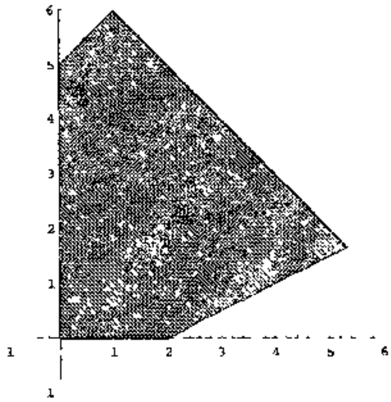

Рис. 1

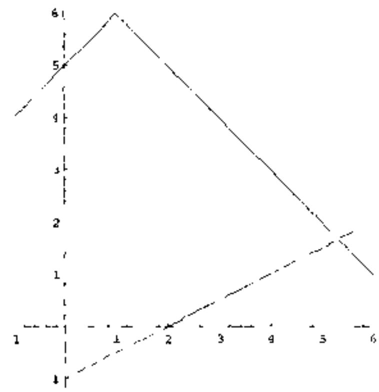

Рис. 2

Можно наблюдать явные отличия в использовании данных функций. Это касается как синтаксиса команд, так и вывода результата. Для наглядного изображения области при применении команды ImplicitPlot необходимо сформировать объекты $p1$, $p2$ и $p3$, которые содержат по отдельности каждую прямую, причем записываются ограничения в виде равенств. Затем командой Show[$p1$, $p2$, $p3$] все прямые изображаем на одном графике, при этом остается непонятным, какая область соответствует заданным неравенствам. В отличие от ImplicitPlot функция InequalityPlot одновременно рисует и закрашивает область, заданную неравенствами, перечисленными через символ ^.

Пример решения задачи целочисленного линейного программирования методом ветвей и границ по шагам

Условие. Решить ЗЦЛП на минимум с двумя переменными, используя алгоритм Лэнд и Дойг.

Р е ш е н и е .

$$f = -2x - y \to \min, X = \{x - y \geq -5, x + y \leq 7, x - 2y \leq 2x, y \geq 0, x, y \in Z\} \qquad (2)$$

Шаг 1. Строим множество планов ЗЦЛП (2).

<<Graphics`InequalityGraphics`
InequalityPlot[x-y≥-5^x+y≤7^x-2y≤2^x≥0^y≥0, {x}, {y},
PlotRange→ {{-1,6}, {-1, 6}}]

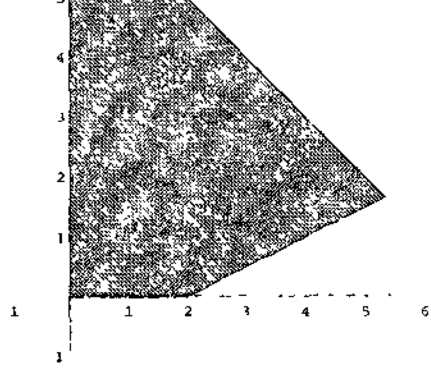

Рис. 3

Шаг 2. Решаем задачу целочисленного линейного программирования (2). Значение целевой функции присваиваем переменной f, а координаты точки, в которой достигается минимум функции f, переменным x1 и y1:

{f, {x1, y1}}=Minimize[{-2x + y, x-y≥-5, x + y≤7, x-2y≤2}, {x,y}]

$$\left\{-9,\left\{x\to\frac{16}{3},y\to\frac{5}{3}\right\}\right\}.$$

Шаг 3. Для дальнейшего обращения к значениям координат запишем их в десятичном виде в переменные x2 и y2:

x2=N[x/.x1]

5.33333

y2=N[y/.y1]

1.66667

Шаг 4. Находим тип переменных x2 и y2 и записываем эти значения в temp1 и temp 2. Проверяем, есть ли среди найденных значений тип Real. Если да, то определяем ближайшее целое число любой из координат, не превосходящее ее, например x2, иначе выводим минимальное значение целевой функции f:

temp=Real

t0=Necelie

{temp1,temp2}={Head[x2],Head[y2]}

t = If[Or[temp = temp1,temp = temp2],t0,f]

If[t0 = t,Floor[x2],0]

{Real, Real}

Necelie

5

Получили нецелевые значения x2 и y2. Следовательно, рассматриваем ограничения x≤5 и x≥6, т. к. координата x точки (16/3; 5/3) попадает в промежуток [5; 6].

Шаг 5. Добавляем ограничение x ≤ 5 к исходным и решаем новую задачу:

{f1,{x3,y3}}=Minimize[{-2x+y, x-y≥-5, x+y≤7, x-2y≤2, x≤Round[x2]} , {x,y}]

$$\left\{-\frac{17}{2},\left\{x\to 5,y\to\frac{3}{2}\right\}\right\}.$$

Шаг 6. Добавляем ограничение x≥6 к исходным и решаем еще одну новую задачу:

{f2,{x4,y4}}=Minimize[{-2x+y, x-y≥-5, x+y≤7, x-2y≤2 , x≤Floor[x2]+1} , {x,y}]

{∞, **{x→Indeterminate, y→Indeterminate}**}

Анализируем полученный результат и делаем вывод, что, так как исходная задача с добавленным ограничением x≥6 не имеет общего множества, то нас он не удовлетворяет.

Шаг 7. Так как минимальное значение целевой функции на шаге 5 принадлежит отрезку [1; 2], то рассматриваем два ограничения y≤1 и y≥2. Решая исходную задачу с ограничением y≤1, получаем

Minimize[{-2x+y, x-y≥-5, x+y≤7, x-2y≤2, y≤1}, {x,y}]

{-7, {x→4,y→1}}

Шаг 8. Решая исходную задачу с ограничением y≥2, получаем

Minimize[{-2x+y, x-y≥-5, x+y≤7, x-2y≤2, y≥2}, {x,y}]

{-8, {x→5, y→2}}.

Шаг 9. Делаем вывод: координаты x и y – целые значения; т. к. мы решаем задачу на минимум, то из двух последних значений целевой функции выбираем минимальное, то есть {–8, {x→5, y→2}} – ответ задачи.

Заключение

Разработанный программный модуль представляет собой электронный документ с демонстрационно-обучающими элементами. Данный проект интересен тем, что является наглядным пособием для закрепления изученного теоретического материала, причем пошаговая реализация решения алгоритма в представленной интерпретации усваивается учащимися гораздо быстрее и лучше. Модуль дает возможность самостоятельно изучать представленные методы, а в настоящее время становится все более актуальным и распространенным компьютеризированная подача обучающих материалов и их самостоятельное изучение. В дальнейшем возможно усовершенствование реализованной программы за счет добавления новых возможностей и функций.

Литература

1. *Кофман, А.* Методы и модели исследования операций. Целочисленное программирование / А. Кофман, А. Анри-Лабордер. – М. : Мир, 1977 – 432 с.
2. *Муравьев, В. А.* Практическое введение в пакет Mathematica / В. А. Муравьев, Д. К. Бурланков. – Нижний Новгород : Изд-во Нижегородского ун-та, 2000. – 124 с.
3. *Дьяконов, В.* Компьютерная математика. Теория и практика / В. Дьяконов. – СПб. : Питер, 2001. – 820 с.
4. *Дьяконов, В.* Mathematica 4.0 с пакетами расширений / В. Дьяконов. – М. : Нолидж, 2000. – 656 с.

Цехан Ольга Борисовна, доцент кафедры математического и информационного обеспечения экономических систем Гродненского государственного университета имени Янки Купалы, кандидат физико-математических наук, tsekhan@grsu.by

Шпак Дарья Сергеевна, студентка 4-го курса кафедры теории функций, функционального анализа и прикладной математики факультета математики и информатики Гродненского государственного университета имени Янки Купалы,kamellia07@mail.ru

Мазан Татьяна Георгиевна, студентка 4-го курса кафедры теории функций, функционального анализа и прикладной математики факультета математики и информатики Гродненского государственного университета имени. Янки Купалы, mahatonia@mail.ru

УДК 005.963.1

**Р. М. Шидловский, А. В. Шейбут, Е. А. Левчук,
Н. А. Шаповалова, В. Н. Леванцов**

ИННОВАЦИОННЫЕ ФОРМЫ ОРГАНИЗАЦИИ ТЕСТИРОВАНИЯ В УЧЕБНОМ ПРОЦЕССЕ

Рассматриваются вопросы создания распределенной компьютерной системы тестирования знания студентов высших учебных заведений, обучающихся на стационаре. Предлагаемые решения позволяют значительно повысить эффективность работы преподавателей, обеспечивая эффективный инновационный дополнительный механизм усвоения теоретических сведений по различным дисциплинам учебного плана. Рассматриваемая система прошла апробацию на ряде кафедр Учреждения образования «ГГУ им. Ф. Скорины», обеспечивающих подготовку специалистов в области информационных технологий.

Введение

Системы компьютерного контроля знаний – это системы тестирования, позволяющие проводить анализ знаний учащихся при помощи современных информационных технологий. Одно из преимуществ автоматизированных систем контроля знаний в том, что они могут использовать сложные методики представления заданий учащимся, называемые стратегиями тестирования.

В настоящее время существует много различных систем, которые реализуют большую часть необходимых функций. Проблема состоит в том, что все эти системы ориентированы главным образом на использование в дистанционном режиме. В данной статье описывается система тестирования, разработанная на кафедре автоматизированных систем обработки информации и предназначенная прежде всего на применение тестирования в учебном процессе вуза на стационаре.