

СТИСЛИЙ ОПИС ПАКЕТІВ РОЗШИРЕННЯ СКМ MAPLE ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ ЛІНІЙНОГО ПРОГРАМУВАННЯ

Вінницький національний технічний університет;

Анотація

Розглядається механізм імплементації основних теоретичних положень з принципів впровадження СКМ Maple для розв'язування задач лінійного програмування.

Ключові слова: система комп'ютерної математики Maple, пакет розширення Simplex, пакет Optimization, задача лінійного програмування.

Abstract

The mechanism of illumination of basic theoretical propositions on the principles of implementation of Maple computer mathematics system for solving linear programming problems is considered.

Keywords: Maple computer math system, Simplex extension package, Optimization package, linear programming problem

Вступ

Останні десятиліття система Maple набуває все більшого поширення серед науковців для обґрунтованого використання під час викладання студентам математично спрямованих дисциплін [3;4; 5]. Найбільш цінними є матеріали, які стосуються покращення наочності, а також реалізації концепції адаптації сучасних СКМ для навчання математики студентів через створення навчальних тренажерів для автоматизованого відтворення покрокового розв'язання типових математичних задач.

У працях [6;7; 8] висвітлено використання СКМ у процесі навчання лінійного програмування, зокрема, розроблено методику використання СКМ, відповідні Maple-тренажери, а також шаблони робочих листів та динамічну керовану модель для унаочнення математичних понять та об'єктів.

Результати дослідження

Система Maple має потужний набір інструментів для розв'язування задач оптимізації: пакет розширення **Simplex** для розв'язування задач лінійного програмування симплекс-методом та новий пакет **Optimization**, що надає можливість розв'язувати не тільки задачі лінійного, але й квадратичного та нелінійного програмування з підвищеним ступенем візуалізації [1; 2].

Команда, яка надає доступ до всіх команд пакета, має вигляд **with(simplex)**. Після виконання вказаної команди в оперативну пам'ять комп'ютера завантажуються усі процедури пакета. Для здійснення доступу тільки до однієї або декількох команд пакета можна скористатися тією самою командою з додатковими опціями **with(simplex, nam1, nam2,...)**; де **nam1,...** – назви команд пакета: *basis, convexhull, cterm, define_zero, display, dual, feasible, maximize, minimize, pivot, pivoteqn, pivotvar, ratio, setup, standardize*, що викликають відповідні процедури, які надають можливість користувачеві поетапно розв'язувати задачу лінійного програмування за симплекс-методом.

Розглянемо найбільш важливі для розв'язування задачі лінійного програмування функції *maximize* і *minimize*. Синтаксис звернення до вказаних команд має доволі простий вигляд **maximize(f, consts)**, **minimize(f, consts)**, де *f* – лінійний вираз, який описує цільову функцію задачі; *consts* – множина або список лінійних обмежень задачі.

Процедури *maximize* та *minimize* повертають або множину рівностей, які описують оптимальний план

```
>with(simplex):
maximize(-x[1]+2*x[2]+3*x[3],{x[1]+2*x[2]-3*x[3]<=4,
5*x[1]-6*x[2]+7*x[3]<=8,9*x[1]+10*x[3]<=11},NONNEGATIVE);
```

$$\left\{ x_3 = \frac{11}{10}, x_1 = 0, x_2 = \frac{73}{20} \right\},$$

```
>minimize(-x[1]+2*x[2]+3*x[3],{x[1]+2*x[2]-3*x[3]<=4,
5*x[1]-6*x[2]+7*x[3]<=8,9*x[1]+10*x[3]<=11},NONNEGATIVE);
```

$$\{x_1 = \frac{11}{9}, x_2 = 0, x_3 = 0\},$$

або порожню множину (рис.1), якщо система обмежень *consts* є несумісною.

>**maximize(3*y[1]-y[2],{-y[1]-y[2]>=-1, y[1]+y[2]>=2},NONNEGATIVE);**

{ }

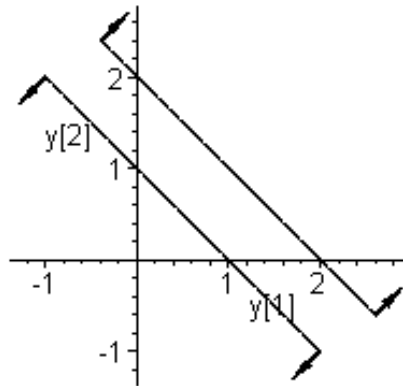


Рисунок 1 – Оптимальний план або порожня множина, або *NULL*, якщо цільова функція на допустимій множині необмежена

>**maximize(47*y[1]+43*y[2],{-3*y[1]-y[2]<=-9, 17*y[1]-64*y[2]<=51,6*y[1]+18*y[2]>=56},NONNEGATIVE);**

Звернемо увагу, що у випадку відсутності розв'язку задачі лінійного програмування через необмеженість області допустимих значень (рис.2) команда *maximize* повертає результат *NULL*, який не супроводжується жодним записом в області виведення результатів виконання команди на екрані монітора. Очевидно, якщо переформулювати указану задачу на знаходження найменшого значення, то розв'язок буде знайдено

>**minimize(47*y[1]+43*y[2],{-3*y[1]-y[2]<=-9, 17*y[1]-64*y[2]<=51,6*y[1]+18*y[2]>=56},NONNEGATIVE);**

$$\{y_1 = \frac{53}{24}, y_2 = \frac{19}{8}\}.$$

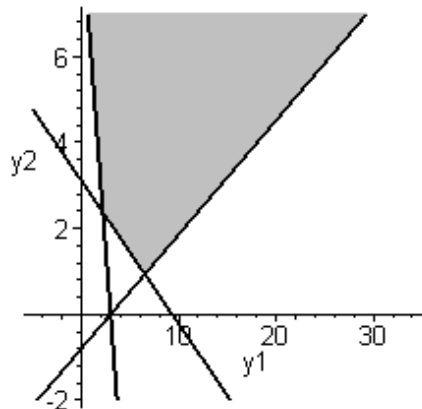


Рисунок 2 – Область допустимих значень

Оптимальний розв'язок може бути підставлений у цільову функцію для одержання її екстремального значення. Це можна зробити за допомогою команд **eval** або **subs**

>**'z'[min] = subs ({y[1]=53/24, y[2]=19/8}, 47*y[1]+43*y[2]);**

$$z_{min} = \frac{2471}{12}.$$

Новий пакет оптимізації підключається за допомогою стандартної команди **with (Optimization):Maximize, Minimize, LPSolve, LSSolve, QPSolve, NLPsolve, ImportMPC, Interactive.**

Для розв'язання задач лінійного програмування призначено функцію **LPSolve**(*obj*, [*constr*,*bd*,*opts*]), де *obj* – цільова функція; *constr* – лінійні співвідношення системи обмежень; *bd* – послідовність, що задає межі можливих значень однієї або кількох змінних; *opts* – рівність, що задає одну із опцій команди LPSolve.

```
>with (Optimization):
LPSolve(-7*x+2*y, {4*x-12*y<=20, -x+3*y<=3},
x=-5..5, y=0..infinity, maximize);
```

[21., [x = -3., y = 0.]]

Як уже зазначалось, новий пакет **Optimization** надає можливість розв'язувати не тільки задачі лінійного програмування, що продемонстровано вище, але й квадратичного та нелінійного програмування з підвищеним ступенем візуалізації.

Для розв'язання задач нелінійного програмування призначена функція **NLPSolve**(*obj*, [*constr*,*bd*,*opts*]):

```
with(Optimization):
NLPSolve(x^3+2*x*y-2*y^2,x=-10..10,y=-10..10,
initialpoint={x=3,y=4},maximize);
```

[1050., [x = 10., y = 5.]]

Але для розв'язання задач квадратичного програмування рекомендується використовувати команду **QPSolve** (*obj*, [*constr*,*bd*,*opts*]):

```
>QPSolve(2*x+5*y+3*x^2+3*x*y+2*y^2,
{x-y>=2}, assume=nonnegative);
```

[16., [x = 2., y = 0.]]

Розв'язати задачу цілочислового програмування можна, використавши функцію **LPSolve** (*obj*, [*constr*,*bd*,*opts*]):

```
> restart;
with(Optimization):
z:=41*x[1]+33*x[2];
```

$41 x_1 + 33 x_2$

```
> linear_constraints := [133835*x[1]-4529360*x[2] <= -561587,
- 55167*x[1]-2182400*x[2] <= -1084571,
-103761*x[1]-378664*x[2] <= - 515601,
- 917135*x[1]-340041*x[2] <= -2475628,
55775*x[1]+557469*x[2] <= 2655572,
0 <= x[1], 0 <=x[2]];
```

$$[133835 x_1 - 4529360 x_2 \leq -561587, -55167 x_1 - 2182400 x_2 \leq -1084571, -103761 x_1 - 378664 x_2 \leq -515601, -917135 x_1 - 340041 x_2 \leq -2475628, 55775 x_1 + 557469 x_2 \leq 2655572, 0 \leq x_1, 0 \leq x_2]$$

```
LPSolve(-z, linear_constraints, integervariables = [(x[k] $ k=1..2)]);
```

[-1222, [x₁ = 29, x₂ = 1]]

Висновки

Отже, з вище викладеного та проілюстрованого випливає, що систему комп'ютерної математики Maple або подібні до неї, доцільно вивчати та використовувати під час викладання дисциплін математичного спрямування, особливо на практичних заняттях або під час самостійної підготовки студентів, що суттєво зменшує час на непродуктивні дії, зокрема, занадто громіздкі обчислення або графічні побудови.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дьяконов В. П. Maple 7: учебный курс/ Дьяконов В. П. — СПб.: Питер, 2002. — 672 с.
2. Дьяконов В.П. Maple 9.5/10 в математике, физике и образовании / Дьяконов В. П. — М.: СОЛОН-Пресс, 2005. — 720с.
3. Михалевич В. М. Математичне програмування в Maple. Частина II. Двоїсті та цілочислові задачі лінійного програмування: навчальний посібник / Михалевич В. М., Тютюнник О. І. — Вінниця: ВНТУ, 2013. — 78с.
4. Михалевич В. М. Математичне програмування разом з Maple. Частина I. Методи розв'язування задач лінійного програмування : навч. посіб. / В. М. Михалевич. — Вінниця : ВНТУ, 2008. — 158 с.
5. Крупський Я. В. Розвиток системи Maple шляхом створення навчальних тренажерів з покрокового розв'язання типових задач вищої математики / Крупський Я. В., Михалевич В. М. // New information technologies in education for all: learning environment. — 2011. — С. 159–165.
6. Триус Ю. В. Використання систем комп'ютерної математики при вивченні і розв'язуванні задач оптимізації // Проблеми сучасного підручника : [зб. наук. праць / ред. кол.]. — К. : Педагогічна думка, 2004. — Вип. 5. — Ч. II. — С. 191–200.
7. Тютюнник О. І. Новый тип учебных задач по линейному программированию в условиях использования СКМ [Электронный ресурс] / О. І. Тютюнник, В. М. Михалевич // Информационно-технологическое обеспечение образовательного процесса современного университета : сб. докл. междунар. интернет-конф. (Минск, 1–30 нояб. 2013 г.). — Минск, 2014. — С. 119–135.
8. Тютюнник О. І. Реалізація принципу наочності за допомогою засобів СКМ у процесі навчання лінійного програмування / О. І. Тютюнник, В. М. Михалевич // Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми : зб. наук. пр. — Випуск 36 / редкол. : І. А. Зязюн (голова) та ін. — Київ-Вінниця : ТОВ фірма «Планер», 2013. — С. 434–440.

Дем'яник Ірина Валеріївна — студентка групи 1 ПІ-19б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: ira.demianyk10@gmail.com.

Науковий керівник: **Оксана Іванівна Тютюнник** — кандидат педагогічних наук, доцент кафедри вищої математики, Вінницький національний технічний університет, м. Вінниця, e-mail: tutunnik.oksana@gmail.com.

Demianyk Iryna Valeriivna— student of 1 PI-19b group, Faculty of Information Technology and Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: bazhanvitalik78@gmail.com.

Supervisor: **Tyutyunnyk Oksana I.** — Candidate of Pedagogical Sciences (Eng.), Docent of the Chair for Higher Mathematics, Vinnitsia National Technical University, Vinnitsia, tutunnik.oksana@gmail.com.