УДК 519.853

В.М. ЯХНО, Л.Л. ФЕДОТОВА

Київський національний університет технологій та дизайну

МЕТОД РОЗВ'ЯЗАННЯ ЗАДАЧІ НЕЛІНІЙНОГО ПРОГРАМУВАННЯ З ОБМЕЖЕННЯМИ У ВИГЛЯДІ НЕРІВНОСТЕЙ

У статті описано метод розв'язання задачі нелінійного програмування з обмеженнями у вигляді нерівностей. Процес пошуку оптимального рішення суттєво відрізняється від інших методів спуску при розв'язанні аналогічних задач – крок обирається та виконується одночасно вздовж двох напрямків.

Ключові слова: *нелінійного програмування, крок, методів спуску*

Значна кількість задач пов'язаних з проектуванням механізмів легкої промисловості формулюється у вигляді задач нелінійного програмування з обмеженнями у вигляді нерівностей. Такою, наприклад, є задача розрахунку місць розташування направляючих пристроїв з метою мінімізації натягу нитки [1].

В загальному списку задач нелінійного програмування задачі з обмеженнями у вигляді нерівностей займають значне місце.

Для пошуку рішень таких задач використовуються методи, в яких відштовхуючись від деякого початкового рішення, послідовно одержують значення, що знаходяться все ближче до шуканого оптимального значення.

Серед пошукових методів особливо важливими є градієнтні методи.

У методах можливих напрямків після вибору підходящого напрямку рух у ньому відбувається доти, поки функція цілі буде спадати (у випадку пошуку її мінімуму) або поки не буде досягнута границя допустимої області.

Обмеження у виді рівностей дозволяють будувати зручні штрафні функції, що є такими які добре диференціюються [3].

Для задач з обмеженнями у вигляді нерівностей необхідно розробляти спеціальні алгоритми або використовувати недиференційовані штрафні функції (у випадку невідомих наперед допустимих планів).

Об'єкти та методи дослідження

Метод, що пропонується, подібно до алгоритмів можливих напрямків, використовує строго допустимі плани задачі. На відміну від більшості реалізацій алгоритмів можливих напрямків [2] алгоритм реалізації методу не потребує розв'язання задачі лінійного програмування на кожній ітерації.

На кожній ітерації алгоритму необхідно або розв'язати задачу бінарного пошуку, або знайти мінімум опуклої униз функції одного аргументу (залежно від стратегії вибору кроку – алгоритм з дробленням кроку або алгоритм, що обирає крок з мінімальним значення функції цілі).

Постановка завдання

Задача формулюється наступним чином: знайти мінімальне значення нелінійної цільової функції аргументів в області, що визначається нерівностями, які містять нелінійні функції аргументів:

$$\min f_0(x_1, x_2, ..., x_n),$$

$$f_j(x_1, x_2, ..., x_n) \le 0, \quad j = 1,2,...,m,$$

$$X = (x_1, x_2, ..., x_n), \; X \subset E^n$$

Функції обмежень f_j є неперервно диференційованими і множина допустимих планів є неперервною множиною (окіл кожної допустимої точки містить внутрішні точки множини).

Результати та їх обговорення

Нехай X^0– план задачі, що є допустимим, ε_0 - додатне число та J_0– множина обмежень, що виконуються в точці X^0 із похибкою меншою ніж ε_0 :

$$-\varepsilon_0 < f_j(x_1, x_2, ..., x_n) \le 0, \quad j \in J_0 .$$

Вірогідність порушення обмежень множини J_0 під час переміщень в околі точки X^0 є найбільш значною.

Позначимо через $grad^n(f_i(X^0))$ нормалізовані (довжина дорівнює одиниці) вектори-градієнти функцій обмежень та функції мети в точці X^0.

Визначимо вектор $P(X^0)$наступним чином

$$P^*(X^0) = -grad^n(f_0(X^0)),$$

$$P^-(X^0) = \sum_j \left(-grad^n(f_j(X^0))\right), \quad j \in J_0$$

$$P(X^0) = P^*(X^0) + P^-(X^0)$$

Графічна ілюстрація того, що вектор $P^-(X^0)$ знаходиться всередині множини допустимих напрямків, для випадку $n=2$ наведена на рис. 1 з такими позначеннями: f_1 – обмеження $f_1(x_1,x_2)\le 0$, f_2 – обмеження $f_2(x_1,x_2)\le 0$, $-f_1'$ – $-gradf_1(X^0)$, $-f_2'$ – $-gradf_2(X^0)$, P^- – вектор $P^-(X^0)$.

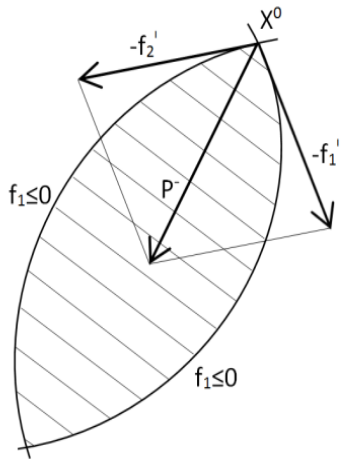

Рис.1

Якщо кількість елементів у множині J_0 менше ніж розмірність простору *n* та всі вектори, що визначають вектор $P(X^0)$, є лінійно незалежними, вектор $P(X^0)$, за визначенням, належить до перетину конусів можливих напрямків та конусу спадання функції цілі. В цьому випадку вектор $P(X^0)$ може бути використаний для покращення функції цілі.

Вектор $P(X^0)$є допустимим напрямком спуску і існує таке додатне число α, що вектор $X^* = X^0 + \alpha P(X^0)$ зменшує функцію мети і є допустимим планом задачі, або

$$f_0(X^0) > f_0(X^0 + \alpha \cdot P(X^0))$$

$$f_j(X^0 + \alpha \cdot P(X^0)) \le 0, \quad j = 1,2,...,m$$

Якщо кількість елементів в множині J_0 дорівнює *n* (ми будемо розглядати лише невироджений випадок, у виродженому випадку зайві обмеження можна відкинути за допомогою відомої обчислювальної процедури), вектор $P(X^0)$ може і не бути допустимим (тобто таким, що належить конусу можливих напрямків) напрямком спуску.

Якщо план X^0 не є розв’язком задачі, то в цьому випадку існують додатні числа $\beta<1$ і μ такі, що вектор $u=\mu\cdot(\beta\ \cdot P^*(X^0)+(1-\beta)\cdot P^-(X^0))$ належить до перетину конусів можливих напрямків та конусу спадання функції цілі. Тут β=1, якщо не виходимо при цьому за межі конуса допустимих планів, і $0<\beta<1$, якщо вектор *u*виводить функцію цілі за межі конуса допустимих планів.

Крок μ, за допомогою процедури одновимірного пошуку, що виконується одночасно вздовж двох напрямків $P^*(X^0)$ *і* $\cdot P^-(X^0)$, можна вибрати таким чином, щоб точка $X^1=X^0+\mu\cdot(\beta\ \cdot P^*(X^0)+(1\text{-}\beta)\cdot P^-(X^0)\ )$або була просто кращою ніж X^0, або найкращою точкою з тих, що належать перетину границі допустимих планів для обмеження$f_1(x_1,x_2)$та гіперповерхні, яка натягнена на вектори $P^*(X^0)$ *і* $\cdot P^-(X^0)$.

Ілюстрація цих обчислень для випадку *n=2* наведена на рис.2 з наступними позначеннями:еліптичні лінії визначають лінії рівня функції, що мінімізується, P^* – нормалізований векторантиградієнту функції, що мінімізується, $P^*(X^0)$, заштрихована ділянка – множина допустимих планів задачі, що утворена двома обмеженнями $f_1(X)$та$f_2(X)$, а $-f'_1$ і $-f'_2$ – антиградієнти цих обмежень.Точка X^1 входить у множину допустимих планів і визначає напрямок подальшого пошуку оптимального рішення.

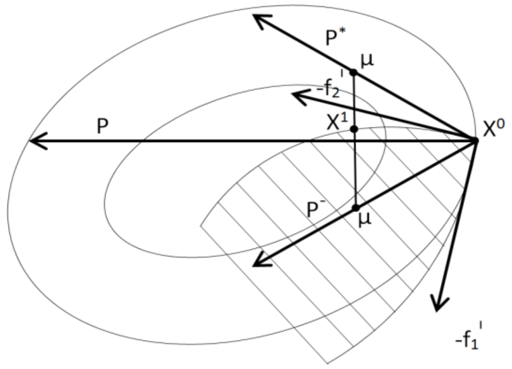

Рис. 2

Вектор $\bar{u} = \beta\cdot(\mu\cdot P^*(X^0) + (1\text{-}\mu)\cdot P^-(X^0))$ може бути використаний для покращення функції цілі. Якщо це не так, в точці X^0виконуються умови Куна – Таккера.

Слід зазначити наступні характерні особливості методу:

– використовується основна ідея методу спуску,

– на відміну від інших методів спуску для визначення наступної точки рекурсивної послідовності використовується не один вектор, а два, вздовж яких виконується пошук необхідного кроку.

Висновки

Описаний метод розв’язання нелінійної оптимізаційної задачі дозволяє по-перше, звузити поле пошуку рішення, обмеживши його так званим конусом допустимих планів, і по-друге, дає можливість знайти розв’язок задачі нелінійного програмування з обмеженнями у вигляді нерівностей, не виходячи при цьому з множини допустимих рішень.

Список використаної літератури

1. Щербань В.Ю., Хомяк О.Н., Щербань Ю.Ю. Механика нити. – Типографія Верховної Ради України, 2000. – 180 с.
2. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике: Кн. 1. – М.: Мир. – 1986.
3. Зайченко Ю.П. Исследование операций. – Вища школа. – 1975.

Стаття надійшла до редакції 12.12.2012

Метод решения задачи нелинейного программирования с ограничениями в виде неравенств

Яхно В.М., Федотова Л.Л.
Киевский национальный университет технологий и дизайна

В статье описан метод решения задачи нелинейного программирования с ограничениями в виде неравенств. Процесс поиска оптимального решения существенно отличается от других методов спуска при решении аналогичных задач –шаг выбирается и выполняется одновременно вдоль двух направлений.

Ключевые слова: нелинейного программирования, шаг, методов спуска.

Method for solving the problem of nonlinear programming with restrictions in the form of inequalities.

V. Yakhno, L. Fedotova
Kyiv National University of Technologies and Design

The article describes the method for solving the problem of nonlinear programming with restrictions in the form of inequalities. The process of the search for an optimal solution essentially differs from other slope methods for solving the similar tasks – the step is choosed and executed along of two directions at the same time.

Keywords: nonlinear programming step descent methods.