побудованої для функції $\varphi_k(x, x^*_{k-2}, x^*_{k-1})$, порівняно з тунельною функцією, побудованою для $f(x)$. Формулу (6) можна узагальнити таким чином, щоб досягти локального збільшення в околах декількох знайдених точок локальних мінімумів цільової функції, що спричинить посилення нерівності (7) і відповідне покращення характеристих побудованих тунельних алгоритмів глобальної мінімізації.

1. Голуб Б.М., Оліярник Ю.П. Тунельний алгоритм пошуку глобального мінімуму неперервної функції // Вісн. Львів. ун-ту. - 1995. – Вип.45 – С.35-39. 2. Голуб Б.М., Оліярник Ю.П. Функції наповнення і їх застосування в глобальній оптимізації// Тези доп. міжнар. конф. "Моделювання і дослідження стійкості процесів. Дослідження систем." Київ, 19-23 травня 1997 р. – К.: 1997.– С.26.

УДК 519.6

Б.М. Голуб, Ю.М. Щербина

КВАЗІНЬЮТОНІВСЬКА МОДИФІКАЦІЯ МЕТОДУ ЛІНЕАРИЗАЦІЇ ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ МАТЕМАТИЧНОГО ПРОГРАМУВАННЯ

Одним з найбільш ефективних методів розв'язування задач математичного програмування з нелінійними функціями є метод лінеаризації Пшеничного [1] та його модифікації. У загальному випадку метод лінеаризації володіє лінійною швидкістю збіжності. В [2] запропонована квазіньютонівська модифікація з надлінійною швидкістю збіжності в околі точки розв'язку. Для розв'язування поширених на практиці задач математичного програмування на простій множині типу паралелепіпеда в [3] досліджено метод лінеаризації із явним врахуванням паралелепіпедних обмежень.

У даній статті сформульований алгоритм методу лінеаризації для розв'язування задачі математичного програмування

$$\min_{u \in U}\{f_0(u) : f_i(u) \le 0, \quad i \in J = \{1,2,\dots,l\}\}, \tag{1}$$

де $U = \{u \in R^n : -\infty < a^i \le u^i \le b^i < \infty,\ i \in I = \{1,2,\dots,n\}\}$, R^n - n-вимірний евклідів простір, $f_i(u), i \in \{0\} \cup J$ - неперервно диференційовні на U функції. З метою спрощення формул у задачі

(1) відсутні обмеження типу рівності. Відзначимо, що ефективність наведеного нижче алгоритму в порівнянні з іншими методами оптимізації була найбільш істотною при наявності нелінійних обмежень типу рівності.

На кожній ітерації методу в поточній точці u потрібно розв'язувати допоміжну задачу квадратичного програмування

$$\min_{p\in Q(u,\mu)}\{\langle f_0(u),p\rangle+\frac{1}{2}\langle p,Ap\rangle:\ \langle f_i'(u),p\rangle+f_i(u)\le 0,\ i\in J_\delta(u)\},\qquad(2)$$

де A – симетрична додатно означена матриця, $Q(u,\mu)$ - означене нижче,

$$J_\delta(u)=\{i\in J: f_i(u)\ge F(u)-\delta\},\quad \delta>0,$$
$$F(u)=\max\{0,f_1(u),...,f_l(u)\}.$$

Ефективні алгоритми розв'язування задачі (2) наведено, наприклад, в [1]. Множники Лагранжа цієї задачі позначимо через $\lambda^i, i\in J_\delta(u)$. Приймемо $\lambda^i=0,\ i\in J\setminus J_\delta(u)$. Позначимо

$$\mathrm{L}(u,\lambda)=f_0(u)+\sum_{i\in J}\lambda^i f_i(u),\quad \Phi_N(u)=f_0(u)+NF(u),\quad N>0.$$

Матриця A вирішальним чином впливає на швидкість збіжності алгоритму. Якщо $A=\mathrm{L}''_{uu}(u,\lambda)$, то теоретично можна досягти квадратичної збіжності. Оскільки обчислення других похідних не завжди є можливим, пропонується для розрахунку елементів матриці A використовувати квазіньютонівські формули. Доцільно (з точки зору кількості обчислень) вести перерахунок не самої матриці A, а її факторів Холеського L (нижня трикутна одинична матриця) та D (додатна діагональна матриця): $A=LDL^*$. Обчислювальні схеми квазіньютонівського перерахунку модифікованих факторів Холеського наведені в [4]. Позначимо $a^{\max}=\max_{1\le i\le n}a^i$, $d^{\min}=\min_{1\le i\le n}d^i$.

Означимо множини:

$$I(x)=\{i\in I: a^i+r<u^i<b^i-r\},$$
$$I_\Gamma(u,\lambda)=\{i\in I: \partial\mathrm{L}(u,\lambda)/\partial u^i>0, u^i=a^i;\ \partial\mathrm{L}(u,\lambda)/\partial u^i<0, u^i=b^i\}$$
$$I_a(u,\lambda)=\{i\in I\setminus I_\Gamma(u,\lambda): u^i\le a^i+r\},$$
$$I_b(u,\lambda)=\{i\in I\setminus I_\Gamma(u,\lambda): u^i\le b^i-r\},$$
$$Q(u,\lambda)=\{p\in R^n: p^i\ge-\theta^i(u),\ i\in I_a(u,\lambda),$$
$$p^i\le\theta^i(u),\ i\in I_b(u,\lambda),\quad p^i=0,\ i\in I_\Gamma(u,\lambda)\},$$

де $\theta^i(u) = u^i - a^i$, $i \in I_a(u,\lambda)$, $\qquad \theta^i(u) = b^i - u^i$, $i \in I_b(u,\lambda)$, $0 < r < \min\{1, \min_i(b^i - a^i)/2\}$, а $I_\Gamma(u,\lambda)$ - множина індексів з I, яка перераховується під час виконання алгоритму.

Нехай обрано числа $N > 0$, $\delta > 0$, $S > 1$, $\sigma > 0$, $r > 0, 0 < \varepsilon < 0.5, c_0 > 0$, $0 < \gamma < 1$ і початкове наближення $u_0 \in U$. Приймемо $L_0 = I_n, D_0 = I_n$ (I_n - n-вимірна одинична матриця), а також $\lambda_{-1} = 0, \lambda_{-1} \in R^l$ і $I_\Gamma(u_0,\lambda_{-1}) = I_{\bar{\Gamma}}(u_0,\lambda_{-1})$.

Опишемо загальний крок алгоритму. Нехай точка u_k, число c_k, множина $I_\Gamma(u_k,\lambda_{k-1})$ та матриці L_k та D_k вже побудовані.

Алгоритм.

1. Розв'язати задачу (2) при $u = u_k, \lambda = \lambda_{k-1}, A = L_k D_k L_k^*$. Отримаємо $p_k = p(u_k)$, $\lambda_k^i = \lambda^i(u_k), i \in J_\delta(u_k)$. Приймемо $u_k^i = 0$ для $i \notin J_\delta(u_k)$.

2. Якщо $\|p_k\| = 0$, то перейти до кроку 9.

3. Обчислити $\beta_k = \max\{\beta : \beta \le 1, u_k + \beta p_k \in U\}$.

4. Якщо $\|p_k\| \le c_k$, $\beta_k = 1$, $\Phi_N(u_k + p_k) \le \Phi_N(u_0)$, то прийняти $u_{k+1} = u_k + p_k$, $c_{k+1} = \gamma\|p_k\|$ і перейти до кроку 6. Інакше прийняти $c_{k+1} = c_k$ і перейти до наступного кроку.

5. Починаючи з $\alpha = \beta_k$, дробити α діленням навпіл до першого виконання нерівності $\Phi_N(u_k + \alpha p_k) \le \Phi_N(u_k) + \alpha\varepsilon\langle p_k, \mathrm{L}'_u(u_k,\lambda_k)\rangle$. Прийняти $u_{k+1} = u_k + \alpha_k p_k$.

6. Прийняти $I_\Gamma(u_{k+1},\lambda_k) = I_{\bar{\Gamma}}(u_{k+1},\lambda_k)$. Якщо $I_\Gamma(u_{k+1},\lambda_k) \ne$ $\ne I_\Gamma(u_k,\lambda_{k-1})$, то перейти до кроку 8.

7. Перерахувати матриці L_{k+1} та $D_{k=1}$ [4]. Якщо $a_{k+1}^{\max} / d_{k+1}^{\min} \le S$, то перейти до кроку 10.

8. Прийняти $L_{k+1} = I_n, D_{k+1} = I_n$ і перейти до кроку 10.

9. Якщо $\lambda_k^i \ge 0,\ \lambda_k^i f_i(u_k) = 0,\ i \in J$ і

$$\frac{\partial \mathrm{L}(u,\lambda)}{\partial u^i} \begin{cases} \ge 0, \text{ якщо } u^i = a^i \\ = 0, \text{ якщо } a^i < u^i < b^i \\ \le 0, \text{ якщо } u^i = b^i \end{cases},$$

то процес обчислень завершуємо. У протилежному випадку приймемо $u_{k+1} = u_k, I_\Gamma(u_{k+1}, \lambda_k) = \varnothing$.

10. Приймемо $k = k + 1$ та перейдемо до кроку 1.

Теорема. *Нехай існують такі константи* $N > 0, \delta > 0$ та $r > 0$, *що*

1) множина $\Omega = \{u \in U : \Phi_N(u) \le \Phi_N(u_0)$ *компактна;*

2) $f_i'(u),\ i = 0,...l$ *задовольняють в* Ω *умові Ліпшиця;*

3) задача (2) має розв'язок при довільному $u \in \Omega$, *причому* $\sum_{i \in J_\delta(u)} \lambda^i(u) \le N$.

Тоді алгоритм будує послідовність $\{u_k\}$ *таку, що* $u_k \in U,\ p(u_k) \to 0,\ F(u_k) \to 0$ *при* $k \to \infty$ *і в довільній граничній точці цієї послідовності задовольняються необхідні умови екстремума для задачі (1).*

При деяких додаткових умовах щодо гладкості функцій $f_i(u)$ і регулярності точки розв'язку для квазіньютонівських формул Бройдена-Флетчера-Гольдфарба-Шенно та Девідона-Флетчера-Пауелла [4] можна довести надлінійну швидкість збіжності в околі точки розв'язку.

Сформульований алгоритм ефективно розв'язував [5] і задачі оптимального керування у наступній постановці:

$$\min_{u,\xi} \int_0^T f_0(x(t), u(t), t, \xi)dt + \Psi(x(T), T, \xi), \tag{4}$$

де $$dx/dt = f(x(t), u(t), t, \xi),\ 0 \le t \le T, \qquad x(0) = v(\xi),$$

$$\begin{gathered} h_0(u(0), \xi) = 0, \quad g_0(u(0), \xi) \le 0, \\ h(x(T), u(T), T, \xi) = 0, \quad g(x(T), u(T), T, \xi) \le 0 \\ h(x(t), u(t), t, \xi) = 0, \quad g(x(t), u(t), t, \xi) \le 0 \\ \int_0^T \bar{h}(x(t), u(t), t, \xi)dt + H(x(T), T, \xi) = 0 \\ \int_0^T \bar{g}(x(t), u(t), t, \xi)dt + G(x(T), T, \xi) \le 0, \end{gathered} \tag{5}$$

$x(t) \in R^n,\ u(t) \in R^m,\ \xi \in R^q,\ t$ – час, а всі вектор-функції неперервно диференційовані за сукупністю аргументів.

Наведена задача оптимального керування шляхом дискретизації [6] зводиться до задачі (1).

Для розв'язування задач (1) або (4)-(5) наведеним алгоритмом розроблено пакет прикладних програм.

1. Пшеничный Б.Н. Метод линеаризации. – М.: Наука, 1983. 2. Щербина Ю.Н., Голуб Б.М. Квазиньютоновская модификация метода линеаризации // Кибернетика. – 1988. -№6. – С.66-71. 3. Щербина Ю.Н., Голуб Б.М. Модификация метода линеаризации для решения задачи математического программирования на простом множестве типа параллелепипеда // Математ. методы и физ.-мех. поля. – 1989. - №30. – С.24-28. 4. Голуб Б.М. Одна схема побудови квазіньютонівських алгоритмів для безумовної мінімізації функцій // Вісн. Львів. ун-ту. –1989. – Вип.31. – С.22-24. 5. Голуб Б.М., Щербина Ю.М. Чисельне розв'язування задач оптимального керування // Вісн. Львів. ун-ту. –1993. – Вип.39. – С.8-14. 6. Евтушенко Ю.Г. Методы решения экстремальных задач и их применение в системах оптимизации. - М.: Наука, 1982.

УДК 681.3:06:51

Мирослава Дзіковська

ПРЕДСТАВЛЕННЯ ДАНИХ ДЛЯ СТІЙКОГО РОЗУМІННЯ МОВИ У ДІАЛОГОВІЙ СИСТЕМІ TRIPS-98

1. Вступ

На практиці все більшого застосування набувають системи, призначені для розуміння людської мови і розумного спілкування комп'ютера з користувачем. Хоча сьогодні найширшого застосування набули системи автоматичного диктування, які майже не проводять аналізу отриманих речень, вже існують системи, здатні розуміти зміст певної множини фраз і виконувати певні дії як результат розмови з користувачем, як наприклад, автоматизовані системи резервування авіаквитків.

Розроблена на факультеті комп'ютерних наук Рочестерського університету діалогова система TRIPS призначена для планування перевезення людей і вантажів в екстремальних ситуаціях. Її попередником була система TRAINS, призначена для планування руху на залізниці [1]. Більша частина модулів написана на мові Common LISP, за винятком системи вводу-виводу графічної інформації, написаної на мові Java, і системи обробки звуку, написаної на C++.