Случайный поиск в задаче о назначениях

Ширинкина Дарья Андреевна, гр. 422

Санкт-Петербургский государственный университет
Прикладная математика и информатика
Вычислительная стохастика и статистические модели

Научный руководитель: д.ф.-м.н., профессор Ю. А. Сушков
Рецензент: д.ф.-м.н., профессор Н. К. Кривулин

Санкт-Петербург
2016г.

Постановка задачи

Задано:

- Набор функций $f_1(x[1:d]), f_2(x[1:d]) \dots f_n(x[1:d])$, где $f_i : D \subset \mathbb{R}^d \to \mathbb{R}$ и $i \in I = \{1, 2, \dots, n\}$
- $\{g[1], g[2], \dots, g[m]\}$ — набор фиксированных значений, $g[j] \in \mathbb{R}$ и $j \in J = \{1, 2, \dots, m\}$
- $n \geqslant m$
- $\varphi : J \to I$ — инъективное отображение
- Целевая функция $C(x[1:d], \varphi)$, отвечающая за точность приближения $f_i(x[1:d])$ к значениям $g[j]$

Задача синтеза системы:

$$C(x[1:d], \varphi) \to \min_{x, \varphi}$$

Постановка задачи

Часто используемые представления целевой функции:

- $C(x[1:d],\varphi) = F_1(x,\varphi) = \sum_{j\in J}(f_{\varphi(j)}(x[1:d]) - g[j])^2$
- $C(x[1:d],\varphi) = F_2(x,\varphi) = \max_{j\in J}(f_{\varphi(j)}(x[1:d]) - g[j])^2$
- $C(x[1:d],\varphi) = F_3(x,\varphi) = \max_{j\in J}\left|\frac{f_{\varphi(j)}(x[1:d])}{g[j]} - 1\right|$

Цель: изучение способа решения задачи, при котором отображение φ заменяется на дополнительное отображение для дальнейшего применения к целевой функции случайного поиска.

Случайный поиск

Пусть нужно найти минимум целевой функции $C(x[1:d])$. Случайный поиск имеет n_{stage} этапов, каждый этап состоит из m_{step} шагов.

Схема работы алгоритма:

- На шаге j выбираются случайные значения $x_j[1:d]$
- Вычисляются $C_{\min}^{j} = \min\{C(x_j[1:d]); C_{\min}^{j-1}\}$
- После выполнения m_{step} шагов изменяется закон выбора значений $x_j[1:d]$ и происходит переход к следующему этапу

Детерминированный поиск оптимального отображения

Рассмотрим изученный способ решения (Сушков, 2000).
Перепишем целевую функцию:

$$C(x[1:d]) = \min_{\varphi}(C(x[1:d], \varphi)).$$

Идея подхода:

- Применяем случайный поиск к функции $C(x[1:d])$ по аргументу $x[1:d]$
- Для вычисления функции $C(x[1:d])$ в фиксированной точке $\tilde{x}[1:d]$ используем алгоритм, находящий $\min_{\varphi}(C(\tilde{x}[1:d], \varphi))$

Недостатки:

- Время работы
- Алгоритм нахождения $\min_{\varphi}(C(\tilde{x}[1:d], \varphi))$ не является универсальным

Случайный поиск оптимального отображения

Представим целевую функцию $C(x[1:d], \varphi)$ в виде

$$C\left(x[1:d], \psi(y[1:r])\right)$$

- Множество значений отображения ψ — все возможные отображения φ
- $y \in [0;1]^r$, где $r \in \mathbb{N}$
- Пусть $\psi(\tilde{y}, j) = \tilde{\varphi}(j)$, если $\psi(\tilde{y}) = \tilde{\varphi}$

Применяем случайный поиск к функции $C(x[1:d], \psi(y[1:r]))$ по аргументам $x[1:d]$ и $y[1:r]$.

В работе предложены 3 варианта задания функции $\psi(y, j)$.

Метод 1

Рассматриваем $r = m$ и $j \in J = \{1, 2, \dots, m\}$, $I = \{1, 2, \dots, n\}$.
Вспомогательная функция:

$$\hat{\psi}(y[1:m], j) = \lceil (n - j + 1) \cdot y[j] \rceil.$$

Пусть $\mathfrak{N} = \mathfrak{N}_1 = \{1, 2, \dots, n\}$ — упорядоченный набор.
Вычисление $\psi(y[1:m], j)$:

1. $\psi(y[1:m], j)$ равно числу из $\mathfrak{N}_j$ с номером $\hat{\psi}(y[1:m], j)$, обозначим его за α_j
2. Если $\psi(y[1:m], j) = \alpha_j$ — выбранное число, тогда $\mathfrak{N}_{\mathfrak{j}+1} = \mathfrak{N}_j \setminus \alpha_j$

Метод 2

Предполагаем, что $r = 1$ и $I = J = \{1, 2, \ldots, n\}$.
Вспомогательная функция:

$$\hat{\psi}(y_j, j) = \lceil (n - j + 1) \cdot y_j \rceil + 1$$

При этом $y_1 = y$, а $y_{j+1} = \{(n - j + 1) \cdot y_j\}$.
Положим $\mathfrak{N} = \mathfrak{N}_1 = \{1, 2, \ldots, n\}$ — упорядоченный набор.
Вычисление $\psi(y, j)$:

1. $\psi(y, j)$ равно числу из $\mathfrak{N}_j$ с номером $\hat{\psi}(y_j, j)$, обозначим его за α_j
2. Если $\psi(y, j) = \alpha_j$ — выбранное число, тогда $\mathfrak{N}_{j+1} = \mathfrak{N}_j \setminus \alpha_j$

Метод 3

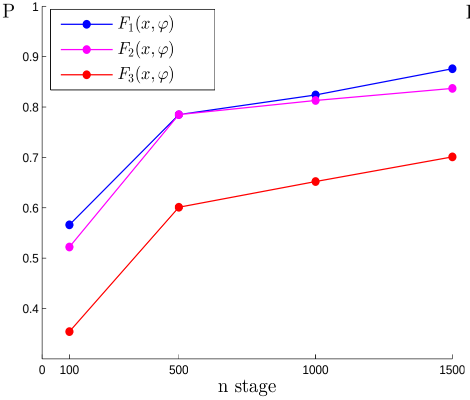
Метод с использованием факториальной системы счисления.

Предполагаем, что $r = 1$ и $I = J = \{1, 2, \ldots, n\}$. Вычисление $\psi(y, j)$:

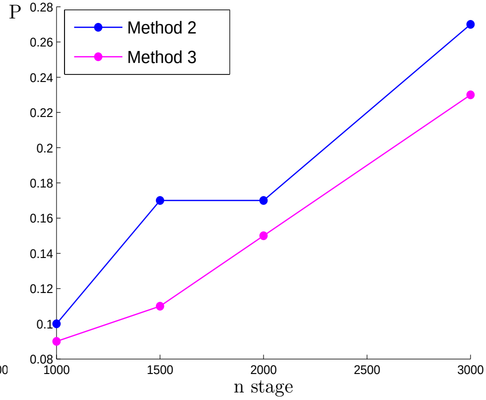
1. $\nu = \lfloor y \cdot n! \rfloor$ — это номер перестановки порядка n
2. ν в факториальной системе счисления: $\nu = \sum_{k=1}^{n-1} \nu_k \cdot k!$
3. По множеству $\{\nu_1, \nu_2, \ldots, \nu_{n-1}\}$ строится перестановка, j-ый элемент которой — $\psi(y, j)$: коэффициент ν_k обозначает число инверсий для элемента $k + 1$

Для получения численных результатов использовались:

- $f_i(x[1:d]) = \frac{1}{2}\langle A_i x, x\rangle + \langle b_i, x\rangle + c_i$, где A_i — матрица порядка $d \times d$, симметричная и неотрицательно определенная, b_i — вектор размерности d, $c_i \in \mathbb{R}$, $i \in I = \{1, 2, \dots, n\}$
- $g[j] = f_{i_j}(x^*[1:d])$, где $j \in J = \{1, 2, \dots, m\}$ и $\{i_1, i_2, \dots, i_m\}$ — случайный набор индексов из I без повторений, $x^*[1:d]$ — известная точка минимума
- $P = \frac{\hat{N}}{N}$, где P — вероятность нахождения глобального минимума, $\hat{N}$ — число найденных значений целевой функции в окрестности $0,02$ глобального минимума, N — общее число запусков алгоритма поиска минимума



а) метод 1



б) метод 2 и метод 3 (F_1)

Рис. 1: Зависимость вероятности нахождения минимума целевой функции от количества этапов случайного поиска. Параметры: $m_{step} = 100$, $n = m = 9$.

Результаты

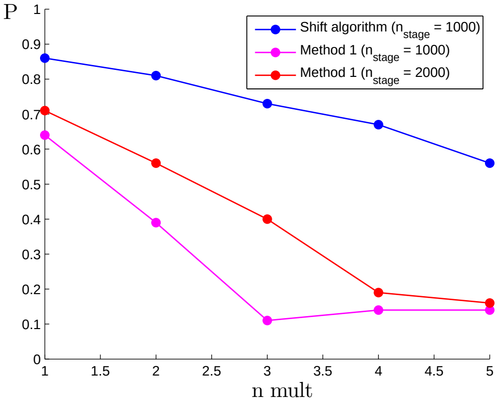


Рис. 2: Зависимость вероятности нахождения минимума целевой функции F_1 от количества многоэкстремальных функций (n mult) среди набора $\{f_1, f_2, \ldots, f_n\}$ (остальные f_i квадратичного вида) для метода 1 и метода детерминированного поиска отображения (Shift algorithm). Параметры: $n = m = 9$, $m_{step} = 100$.

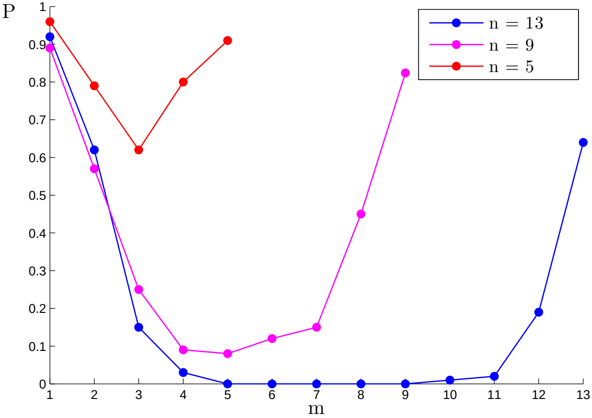


Рис. 3: Зависимость вероятности нахождения минимума целевой функции F_1 от m для метода 1.

Заключение

Результаты:

- Рассмотрены 3 различных способа задания отображения $\psi(y)$
- Выявлен один более эффективный метод из рассмотренных трёх способов задания отображения $\psi(y)$
- Получено, что в случае, когда m близко к $\frac{n}{2}$, для нахождения минимума с помощью метода 1 нужно увеличивать количество общих шагов случайного поиска по сравнению с $m \approx 1$ и $m \approx n$
- Показано, что количество общих шагов случайного поиска стоит увеличивать в случае, когда функции f_i имеют сложный многоэкстремальный характер