ВЫЧИСЛИТЕЛЬНАЯ РЕАЛИЗАЦИЯ ОБОБЩЕННОГО МЕТОДА DIRECT ДЛЯ ЗАДАЧИ С ФУНКЦИОНАЛЬНЫМИ ОГРАНИЧЕНИЯМИ

А.Ю. Жунин

Нижегородский госуниверситет им. Н.И. Лобачевского

Рассмотрен вариант вычислительной реализации обобщения метода Direct [1] на задачи с липшицевыми функциональными ограничениями при неограниченном диапазоне значений констант Липшица. Данное обобщение предложено в работе [2], в которой приведено теоретическое описание и обоснование двух возможных подходов. Здесь реализован второй подход. В докладе будут приведены результаты вычислительных экспериментов.

Введение

Метод Direct, предложенный в работе [1], разработан для задач безусловной липшицевой многоэкстремальной оптимизации с неограниченным диапазоном изменения константы Липшица. Описание метода, а также некоторые его модификации можно также найти в [3]. Метод Direct не рассчитан на более широкий класс задач с функциональными ограничениями.

В работе [2] предложены два специальных подхода к распространению метода на эти более сложные задачи. Второй из этих подходов непосредственно обобщает концепции метода Direct на случай наличия функциональных ограничений. При этом используется предположение о том, что неизвестная константа L липшицевой минимизируемой функции f, где $L \in [0, \infty)$, связана с константой L_g липшицевой функции g обобщенного ограничения соотношением $L_g \in [0, \gamma L],\ \gamma > 0$.

Постановка задачи

Задача условной многоэкстремальной оптимизации рассматривается в следующей постановке:

$$f^* = f(x^*) = min f(x),\ \ x \in X, \tag{1}$$

$$X = \{x \in D = [a, b] \subset R^N:\ g_i(x) \le 0,\ (i = \overline{1, m})\}. \tag{2}$$

Набор функциональных ограничений сводится к одному обобщенному ограничению

$$g(x) \le 0, \tag{3}$$

где

$$g(x) = \max\{g_1(x), \dots, g_m(x)\}. \tag{4}$$

Предполагается, что функция f является липшицевой с константой L, значение L неизвестно и $L \in [0, \infty)$. Функция g липшицева с константой L_g, значение константы неизвестно, но взаимосвязано с L соотношением $L_g \in [0, \gamma L]$, где $\gamma > 0$.

Также как и метод Direct, рассматриваемое обобщение реализовано как компонентный метод, основанный на динамическом разбиении исходного множества-гиперинтервала $D = [a, b]$ на гиперинтервалы $D_i = [a_i, b_i]$ меньшего размера по центральной схеме деления на три. На каждой итерации совокупность гиперинтервалов $\{D_i\}$ полностью покрывает D не пересекаясь при этом по внутренним точкам. В центре каждого гиперинтервала D_i вычисляются как значения целевой функции так и обобщенного ограничения, обозначаемые далее как f_i и g_i.

В [2] предложен специальный принцип выбора подмножества «лучших» (*потенциально оптимальных*) гиперинтервалов D_t, которые должны подвергаться очередному делению на три (по большему ребру) на текущей итерации. Такие гиперинтервалы выделяются из множества недоминируемых. *Недоминируемыми* в [2] названы такие гиперинтервалы D_t, что:

$$\exists\, L \in [0,\infty):\ f_t - L\frac{d_t}{2} = \min\left\{f_i - L\frac{d_i}{2} : \exists\, L_g \in [0, L*\gamma]: g_i - L_g\frac{d_i}{2} \le 0\right\}, \quad (5)$$

$$\exists\, L_g \in [0, L\gamma]: g_t - L_g\frac{d_t}{2} \le 0. \quad (6)$$

Потенциально оптимальные это такие недоминируемые гиперинтервалы D_t, для которых выполнено дополнительное условие, препятствующее чрезмерному дроблению гиперинтервалов малого диаметра

$$f_t - L\frac{d_t}{2} \le f_{min} - \varepsilon|f_{min}|. \quad (7)$$

Здесь под f_{min} понимается текущая найденная оценка условного глобально-оптимального значения целевой функции.

В [2] изложены теоретические основы реализации правила отбора (6)-(7) с помощью трех вспомогательных этапов. В следующих разделах описаны принципы их программной реализации, приводящие к выделению подмножества потенциально оптимальных гиперинтервалов.

Описание алгоритма

Алгоритм состоит из трех основных частей: обработка «слоев», создание общей таблицы недоминируемы в «слоях» гиперинтервалов, обработка таблицы с выделением потенциально оптимальных.

Выполнение метода начинается с того, что начальный гиперинтервал D делится на 3 равные части (по большему ребру). Это задает стартовый набор гиперинтервалов, образующих первый слой. *Слоем* будем называть подмножество гиперинтервалов с одинаковыми значениями d_i.

Далее описаны действия, выполняемые на некоторой промежуточной итерации.

Обработка слоя

Гиперинтервалы слоя с $d_i = c$ ассоциируются с набором точек (g_i, f_i). Те из них, у которых $g_i < 0$, будем называть *допустимыми*. Обработка допустимого подмножества гиперинтервалов заключается в отыскании среди них минимальной по значению целевой функции. Обозначим это минимальное значение через f^c_{min}.

Подмножество гиперинтервалов, соответствующих точкам с $g_i \ge 0$,будем называть *потенциально допустимыми*, множество их номеров обозначим через P.

Обработка множества потенциально допустимых точек заключается в отыскании в нем подмножества точек с номерами $i \in B$, каждая точка которого удовлетворяет следующему условию.

Если $\mathrm{i} \in \mathrm{B}$, то $\mathrm{f_i} \le \mathrm{f^c_{min}}$ и $\nexists (j \in P)\&(j \ne i)$: $(f_i < f_j)\&(g_i \le g_j)$.

Как видим, одним из условий обработки является то, что отобранные потенциально допустимые точки должны иметь значения целевой функции меньше, чем минимальное значение среди допустимых гиперинтервалов слоя. С допустимыми отобранными точками связывается некоторое «минимально допустимое» значение константы Липшица L_i, равное нулю, а с отобранными потенциально допустимыми – «минимально допустимое» значение константы Липшица, равное

$$L_i = \frac{2g_i}{d_i\gamma}. \quad (8)$$

Отобранные допустимые и потенциально допустимые гиперинтервалы записываются в таблицу вместе со своим ключом L_i, являющимся «минимально допустимым» для них значением константы Липшица. Таблица упорядочивается по ключу. Значением в таблице по данному ключу является список гиперинтервалов с одинаковым значением ключа.

Такие таблицы строятся для каждого из существующих слоев.

Образование общей таблицы

Готовые таблицы слоев обрабатываются поочередно. Обход каждой из таблиц в отдельности начинается с последней строки. Запоминается значение ключа текущей строки L_k и значение ключа уже обработанной строки L_{k-1}. В случае когда обработка только началась (т.е. текущая строка – последняя в таблице), предполагается, что ранее обрабатывалась строка с бесконечным ключом $L_{k-1} = \infty$. По текущему ключу в общей таблице определяется строка, если она существует, в которую копируется список гиперинтервалов из строки обрабатываемой таблицы. Этот список также копируется в последующие строки общей таблицы с большими ключами до строки (не включительно) с ключом L_{k-1}, и если $L_{k-1} = \infty$, то копирование происходит до конца общей таблицы, включая последнюю строку.

Если в общей таблице не существует строки с ключом L_k, то такая строка создается. Затем происходит копирование списка гиперинтервалов по выше приведенному алгоритму для случая, когда строка существует, и копируется список из строки общей таблицы с ключом меньшим L_k в строку с ключом L_k.

Обработка общей таблицы

Структура построенной общей таблицы такова, что все интервалы из одной ее строки, в общем случае, относятся к разным слоям, и каждый из них является недоминируемым в своем слое для некоторого промежутка значений L. Причем в таблице для набора гиперинтервалов данной строки хранятся граничные значения L_{min} (ключ текущей строки в таблице) и L_{max} (ключ следующей строки таблицы), обладающие тем свойством, что вне этого промежутка хотя бы один из гиперинтервалов строки перестает быть недоминируемым в своем слое. Заметим, что когда обрабатывается последняя строка таблицы, L_{max} считается равной ∞.

Рассмотрим попарно гиперинтервалы из одной строки таблицы как точки на плоскости (f_e, g_e).

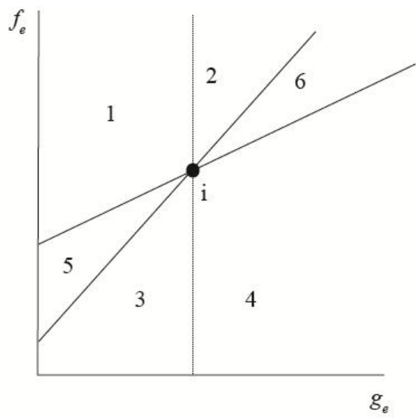

Рис. 1. Точка i с секторами

На рис. 1 показана схема секторов точки, выделяемых для принятия решения о недоминируемости точки из пары.

Если точка j попадает в сектора 1, 2 по отношению к точке i, то она уже не может быть недоминируемой в рассматриваемой группе, и точку j следует исключить из рассмотрения на данном этапе. Если j попала в сектора 3, 4, то исключить из рассмотрения на данном этапе следует точку i. В случае попадания j в сектор 5 вычисляется тангенс угла наклона прямой, проходящей через i и j, и он устанавливается в качестве нового значения L_{max} для j и в качестве нового значения L_{min} для i. При попадании в сектор 6 также вычисляется наклон аналогичной прямой, и тангенс угла наклона устанавливается как L_{max} для i и как L_{min} для j. Далее процесс продолжается до рассмотрения всех оставшихся пар точек в этой группе.

Дополнительная выбраковка оставшихся гиперинтервалов происходит при проверке условия (7), которое для них должно выполняться при соответствующем гиперинтервалу значении $L = L_{max}$.

При выполнении указанной процедуры для всех строк общей таблицы некоторые гиперинтервалы могут быть отобраны вторично. Повторное их включение в формируемый список потенциально оптимальных гиперинтервалов блокируется.

Все отобранные потенциально оптимальные гиперинтервалы разделяются на три равные части по своему большему ребру и в новых центрах проводятся дополнительные измерения функций задачи. Далее процесс обработки повторяется. Останов вычислений выполняется по исчерпанию отведенного вычислительного ресурса.

Во время доклада будут приведены результаты вычислительных экспериментов.

Работа выполнена при финансовой поддержке Минобрнауки России, государственное соглашение о предоставлении гранта № 14.B37.21.0878.

Литература

1. Jones D.R., Perttunen C.D., Stuckman B.E. Lipschitzian optimization without the Lipschitz constant // J. Optim. Theory and Appl. V. 79. № 1. 1993. P. 157–181.
2. Городецкий С.Ю. Обобщения метода Direct на задачи с функциональными ограничениями // Труды XII Всероссийской конференции «Высокопроизводительные параллельные вычисления на кластерных системах». 26-28 ноября 2012, Н.Новгород. – Н.Новгород: Изд-во ННГУ, 2012 (настоящий сборник).
3. Сергеев Я.Д., Квасов Д.Е. Диагональные методы глобальной оптимизации. – М.: Физматлит, 2008. – 352 с.