Решатель задач поиска глобального минимума и максимума функций

Н. В. Панов
КТИ ВТ СО РАН, Новосибирск, Россия

Работа посвящена особенностям реализации эффективного решателя задач поиска глобального экстремума функций, позволяющего получать глобальный оптимум с гарантированной (доказательной) точностью, а равно и аргументы, его доставляющие. Делается обзор следующих техник, приёмов и методов: автоматическое и символьное дифференцирование, упрощение выражений, способы вычисления интервальных оценок и их точность, методы распространения ограничений, способы определения подобластей, гарантированно не содержащих оптимум, методы адаптивного дробления области определения, особенности реализации параллельных алгоритмов поиска и их взаимодействия, интервальные генетические алгоритмы.

Введение

Множество задач, возникающих в различных сферах человеческой деятельности, могут быть сведены к задаче поиска глобального оптимума. Многомерная оптимизация является неотъемлемой частью важнейших этапов моделирования различных (инженерных, экономических, и т.д.) систем. При этом в ряде задач, возникающих в практике оптимизации, требуется не просто приближённое численное решение, но ещё и гарантия того, что найденный оптимум действительно глобальный, а не один из локальных. Подобные постановки задач обычно характеризуют термином «гарантирующая или доказательная[1] глобальная оптимизация», и они являются чрезвычайно трудными. Кроме того, зачастую требуется не только указать величину глобального оптимума, но и определить значение аргументов, при котором он достигается.

Своей целью мы ставим создание решателя задач поиска глобального оптимума функций. Целевые функции при этом могут быть вещественно или интервальнозначные, невыпуклые, многоэкстремальные, недифференцируемые. Единственное условие, накладываемое на целевую функцию, это то, что она должна задаваться в явном виде: в виде математического выражения или подпрограммы, состоящей из комбинации переменных, арифметических операций и математических функций. Это необходимо для вычисления соответствующей интервальнозначной функции, подробнее см. [1].

При разработке особое внимание мы уделяем надёжности и эффективности работы создаваемого решателя. Для повышения надёжности несколько лет назад мы сознательно отказались от C++ в качестве языка разработки и постепенно перешли на Java. Благодаря постоянному развитию этой платформы, в частности, усовершенствованию технологии динамической компиляции во время выполнения программы разница в производительности по сравнению со статически компилируемыми языками уже не столь

[1] По принятой терминологии. См. например К.И. Бабенко «Основы численного анализа» М.:, Наука, 1986

значительна, как во времена, когда Java была создана. В то же время несколько иная парадигма программирования, а так же обилие высокоуровневых библиотек и компонент позволило существенно увеличить скорость разработки и стабильность продукта. Под эффективностью работы мы понимаем как разработку и реализацию наиболее вычислительно эффективных алгоритмов поиска, так и наиболее эффективную утилизацию доступных аппаратных ресурсов, в том числе обеспечение эффективной параллелизации вычислений. Основные инструменты разработки, используемые нами в этом проекте, это jdk7, JUnit, git, Intel VTune, Eclipse с рядом дополнительных компонентов, предназначенных для статического и динамического анализа кода, таких как FindBug.

Основные компоненты решателя

На данный момент основными инструментами поиска являются логика распространения ограничений и различные стратегии адаптивного дробления области поиска в сочетании с оцениванием целевой функции по получающимся подобластям. В дальнейшем планируется добавить ещё и явное решение нелинейных уравнений.

Алгоритмы адаптивного дробления используют тот факт, что интервальные оценки ассимптотически точные. Это означает, что при уменьшении размеров области определения точность интервального расширения функции увеличивается. Отмеченный факт может быть положен в основу процедуры уточнения интервальной оценки области значений функции.

В самом деле, если разбить исходный брус $\boldsymbol{x}$ на два подбруса $\boldsymbol{x}'$ и $\boldsymbol{x}''$, дающие в объединении весь $\boldsymbol{x}$, то есть такие, что $\boldsymbol{x}' \cup \boldsymbol{x}'' = \boldsymbol{x}$, то

$$\{\, f(x) \mid x \in \boldsymbol{x} \,\} = \{\, f(x) \mid x \in \boldsymbol{x}' \,\} \cup \{\, f(x) \mid x \in \boldsymbol{x}'' \,\}.$$

Соответственно, можно вычислить интервальные расширение на каждом подбрусе и в качестве новой оценки минимума целевой функции на $\boldsymbol{x}$ взять

$$\min\{\, \underline{\boldsymbol{f}(\boldsymbol{x}')}\,,\ \underline{\boldsymbol{f}(\boldsymbol{x}'')}\,\}$$

и она будет, вообще говоря, более точна, чем исходная оценка $\boldsymbol{f}(\boldsymbol{x})$, так как у брусов $\boldsymbol{x}'$ и $\boldsymbol{x}''$ размеры меньше, чем у исходного $\boldsymbol{x}$. Брусы-потомки $\boldsymbol{x}'$ и $\boldsymbol{x}''$ можно, в свою очередь, опять разбить на более мелкие части, найти для них интервальные расширения и далее уточнить оценку для минимума, потом снова повторить процедуру и так далее.

Ключевым моментом в подобном подходе является даже не то, как именно дробить каждый конкретный брус, а какой именно брус выбирать для уточнения интервальной оценки на каждой итерации алгоритма.

В решателе реализовано шесть различных алгоритмов адаптивного дробления, лишь один из них детерминистский. Остальные, такие как случайный поиск, поиски с приоритетом, интервальный метод имитации отжига, интервальные генетические алгоритмы, используют стохастические алгоритмы для выбора очередного бруса для дробления. Описание этих алгоритмов, доказательство их сходимости и результаты вычислительных экспериментов можно найти в работах [2–4]. Вкратце, общая идея этих методов в том, что бы за счёт допущения стохастических переходов при выборе очередного бруса для дробления и уточнения интервальной оценки преодолеть эффекты застаивания и избыточности интервальной оценки и повысить вычислительную эффективность методов [5].

Существенное влияние на эффективность алгоритмов адаптивного дробления оказывают используемые типы интервальных расширений. Наши исследования показали, что в зависимости от характерного размера интервала и особенностей целевой функции предпочтительно использовать разные интервальные расширения. Подробнее об этом см. [6].

В настоящее время основной режим работы блока адаптивного дробления — параллельное исполнение нескольких стратегий дробления с применением разных типов интервальных расширений. Для наиболее эффективной параллельной работы методов адаптивного дробления им необходимо обмениваться текущими значениями оптимума.

Интервальное расширение функции гарантирует, что значения функции не выйдут за обозначенные границы. Следовательно, если при поиске глобального минимума на каком-либо подбрусе нижняя граница интервального расширения целевой функции больше, чем найденный на данный момент минимум, то такой брус можно исключать из рассмотрения, так как оптимум на нём гарантированно недостижим. Эта методика получила название «отбраковка по значению».

При параллельной работе нескольких алгоритмов для сохранения эффективной масштабируемости на большее количество потоков такое взаимодействие требуется организовать неблокирующим способом. При такой параллельной работе нескольких алгоритмов неизбежны ситуации, когда какой-то из алгоритмов закончил работу на своей области. Такое возможно как при завершении итераций по достижению критерия остановки, например, локализация оптимума на интервале заданного размера, так и потому, что другой алгоритм обновил критерий отбраковки по значению и это позволило отбросить из рассмотрения всю область, над которой работал данный алгоритм. В этом случае требуется, заблокировав лишь ещё один вычислительный поток, разделить его работу с закончившим.

Другими критериями отбраковки могут выступать интервальные оценки области значений первой и второй производных целевой функции. Действительно, если интервальные расширения частных производных на некой области не содержат ноль, значит, на этой области функция не достигает экстремума, следовательно, если она не содержит границы области поиска, её можно исключить из рассмотрения. Интервальная оценка второй производной позволяет определять выпуклость-вогнутость функции.

Целевая функция задаётся пользователем. Для получения производных целевой функции изначально использовался алгоритм автоматического дифференцирования. Впоследствии, когда для нужд распространения ограничений, а также для нелинейного решателя потребовалась возможность символьного манипулирования с выражениями, автоматическое дифференцирование было заменено символьным, так как последнее позволяет на своей основе реализовать символьное упрощение и преобразование выражений. Подробнее об автоматическом и символьном дифференцировании см. например [7].

Распространение ограничений (иногда также называемое удовлетворением ограничениям или программированием в ограничениях) является одной из интенсивно развивающихся областей искусственного интеллекта и применяется для решения разнообразных задач. Фактически, это один из способов уменьшения пространства поиска за счёт проверки совместности конечного набора ограничений и отсечения недопустимых значений переменных. Подробнее о методе см. например [8].

В нашем случае ограничения берутся из условия равенства нулю производной в точке экстремума и условию выпуклости функции, а также из условия, что целевая функция должна принимать значения не больше тех, которые к этому моменту найдены методами адаптивного дробления и точечными (неинтервальными) алгоритмами локального поиска. В случае, если целевая функция недифференцируема, то последний способ становится единственным источником наполнения множества ограничений.

Блок распространения ограничений работает в параллель с блоком адаптивного дробления. Такая организация работы способствует более эффективному поиску, так как оба блока в процессе работы обмениваются уточненными координатами области поиска и оценками оптимума.

Список литературы

[1] Шарый С.П. *Конечномерный интервальный анализ.*
– Электронная книга, доступная на
http://www.nsc.ru/interval/index.php?j=Library/InteBooks/index.

[2] Панов Н.В., Шарый С.П. Стохастические подходы в интервальных методах глобальной оптимизации // *Всероссийское (с международным участием) совещание по интервальному анализу и его приложениям ИНТЕРВАЛ-06, 1–4 июля 2006 года, Петергоф, Россия. Расширенные тезисы докладов.* — Санкт-Петербург: ВВМ, 2006. – С. 101–105.

[3] Панов Н.В. Адаптивный мета-алгоритм глобальной оптимизации // *Естественные и технические науки* (ISSN 1684-2626). – 2009. №1 (39). — Москва: Спутник+, С. 315–318.

[4] Панов Н.В. Объединение стохастических и интервальных подходов для решения задач глобальной оптимизации функций // *Вычислительные технологии.* – 2009. – Т. 14, №5. – С. 49–65.

[5] Шарый С.П. Рандомизированные алгоритмы в интервальной глобальной оптимизации // *Сиб. Журнал Вычисл. Матем.* – 2008. – Т. 11, №4. – С. 457–474.

[6] Панов Н.В. Оценка области значений функций методами интервального анализа // *Вопросы современной науки и практики (Тамбов, Университет им. Вернадского).* – 2009. – №3 (17). – С. 78–86.

[7] Christodoulos A. Floudas, Panos M. Pardalos *Encyclopedia of Optimization* – Springer, 2009.

[8] Семенов А.Л. Методы распространения ограничений: основные концепции // *Труды Пятой международной конференции памяти академика А.П. Ершова «Перспективы систем информатики».* — Новосибирск, ИПО Эмари, 2003. Т. «Международное совещание по интервальной математике и методам распространения ограничений». – С. 20–31.