

Архитектура и программная организация библиотеки для решения задач оптимизации методом ветвей и границ на многопроцессорных вычислительных комплексах*

М. А. Посыпкин

Введение

Задача оптимизации в самой общей постановке формулируется следующим образом: требуется найти элемент из заданного множества, на котором функция принимает наибольшее или наименьшее на данном множестве значение. Задачи оптимизации являются математическими моделями многих проблем в физике, экономике, военном деле и других областях. Известно, что зачастую задача поиска глобального минимума функции является исключительно сложной в вычислительном отношении. В частности, установлено, что многие задачи дискретной оптимизации [1] являются NP-трудными, требуют для своего решения количества операций, близкого к полному перебору всех элементов из пространства решений.

Одним из путей ускорения решения таких задач является применение методов параллельных и распределенных вычислений, которые позволяют вовлекать в процесс решения задач сразу несколько вычислительных устройств. Одним из основных методов, применяемых при решении задач оптимизации, является *метод ветвей и границ* (МВГ), который подробно рассматривается в следующем разделе. Суть метода заключается в последовательном разбиении множества допустимых решений на подмножества с последующим отсеком подмножеств, не содержащих решения. Декомпозиционный характер метода делает его перспективным с точки зрения применения параллельных и распределенных вычислений.

Общая схема метода ветвей и границ не зависит от конкретной задачи. Особенности конкретной задачи определяется то, каким образом производятся различные операции МВГ, в частности, разбиение пространства решений и вычисление оценок. Универсальность метода делает целесообразным разработку общего инструментария для решения различных задач этим методом, в котором бы присутствовала как проблемно-независимая часть, так и части, специфические для конкретных задач. Данная статья посвящена описанию библиотеки BNB-Solver, которая предоставляет пользователю средства разработки приложений, основанных на методе ветвей

* Работа выполнена при поддержке РФФИ, проекты 05–01–00495-а, 06–07–89079-а.

и границ. В состав библиотеки входят проблемно-независимые модули, ответственные за последовательный и параллельный вариант реализации МВГ, и модули, специфичные для конкретных задач. Таким образом, библиотека может применяться как для решения конкретных задач, для которых написаны соответствующие модули, так и для разработки вариантов МВГ, предназначенных для решения новых задач.

Метод ветвей и границ

Метод ветвей и границ является одним из основных методов, применяемых при численном решении задач оптимизации, в частности, задач дискретной [1] и глобальной [2] оптимизации. Суть метода заключается в последовательном разбиении множества допустимых решений на подмножества с последующим отсеком подмножеств, не содержащих решения.

Рассмотрим задачу дискретного программирования $f(x) \rightarrow \max, x \in G$. Будем говорить, что функция $\phi : 2^G \rightarrow \mathbb{R}$, является *верхней оценкой* для функции f , если для любого подмножества G' множества G справедливо следующее соотношение: $\forall x \in G' : \phi(G') \geq f(x)$. *Правило отсева* подмножеств, не содержащих оптимальных решений, состоит в том, что если для некоторого $x_0 \in G$ и $G' \subseteq G$ известно, что $\phi(G') < f(x_0)$, то подмножество G' может быть исключено из дальнейшего поиска, так как согласно определению верхней оценки, оно не содержит оптимальных решений.

Для получения оценки решается *оценочная задача*. На основании результатов решения оценочной задачи подмножество S исключается из дальнейшего процесса поиска, если для него выполнено хотя бы одно из следующих свойств:

1. S не содержит допустимых решений.
2. При решении оценочной задачи на подмножестве S получено допустимое решение исходной задачи.
3. S может быть отсеяно по правилу отсева.

В процессе решения поддерживается *список задач-кандидатов*, который в начальный момент содержит только один элемент, соответствующий всему множеству допустимых решений, определяемому исходной задачей. На каждом шаге в методе ветвей и границ производится разбиение (ветвление) одного из элементов списка задач-кандидатов на несколько новых задач, для каждой из которых решается оценочная задача. В этом списке задача, подвергнутая ветвлению, заменяется задачами, полученными в результате этого ветвления.

Графически процесс решения задачи методом ветвей и границ можно представить в виде дерева, называемого *деревом ветвлений*. Вершинам дерева соответствуют задачи, получаемые в результате ветвления, а дуги соединяют данную задачу с задачами, полученными из нее в результате ветвления. Ветвлению подвергаются задачи, соответствующие конечным вершинам дерева. Будем обозначать затемненным кружком задачи, исключенные из дальнейшего процесса поиска по правилам 1–3, а белым

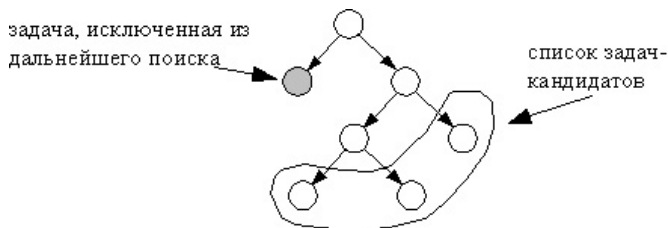


Рис. 1. Дерево ветвления в методе ветвей и границ

кружком — все прочие задачи. Тогда текущий список задач-кандидатов соответствует всем белым конечным вершинам дерева ветвления (рис. 1).

Параллельная реализация метода ветвей и границ

Многопроцессорные вычислительные комплексы могут обладать различной архитектурой. В настоящей работе рассматривается модель параллельного программирования на основе передачи сообщений, которая, как правило, применяется на компьютерах с распределенной памятью. В этой модели программа представляет собой совокупность параллельно выполняющихся процессов, взаимодействующих с помощью сообщений.

Каждый процесс поддерживает собственный список задач-кандидатов, который в начальный момент пуст. Решение задачи начинается на одном из процессов. По мере генерации часть построенных задач-кандидатов может передаваться для обработки от одного процессора другому. Будем считать, что задача-кандидат может быть передана другому процессору для обработки только после того, как для нее вычислена оценка. Такая схема является достаточно общей и большинство подходов к параллельной реализации метода ветвей и границ ей соответствуют. Библиотека BNB-Solver проектировалась с таким расчетом, чтобы любой подход, соответствующий указанной схеме можно было бы реализовать с ее помощью.

Организация библиотеки BNB-Solver

BNB-Solver является модульной объектно-ориентированной библиотекой, реализованной на языке C++. Модули, входящие в библиотеку, можно разделить на несколько групп:

1. Модули коммуникаций, отвечающие за примитивные взаимодействия.
2. Модули стратегии ветвления, реализующие алгоритм ветвления.
3. Модули распараллеливания, ответственные за логику распределения работы между процессами.
4. Проблемно-ориентированные модули, инкапсулирующие особенности конкретного приложения.

Приложение, написанное с использованием библиотеки (в дальнейшем такие приложения будем называть сокращенно BNB-приложениями) совмещает различные модули с помощью механизма шаблонов [3]. Например, класс, соответствующий параллельному решателю для задачи о ранце, записывается следующим образом:

```
BNBClmsSolver < tKSSetFactory, WTraverse < tKSSetFactory > >,
где BNBClmsSolver — класс, реализующий один из вариантов стратегии «управляющий-рабочие», tKSSetFactory — «фабрика классов» для задачи о ранце, а WTraverse — класс, реализующий алгоритм ветвления. Далее в разделе подробно рассматриваются перечисленные модули.
```

Модули коммуникаций

Библиотека BNB-Solver предназначена для разработки параллельных приложений, в которых взаимодействие параллельных процессов осуществляется с помощью передачи сообщений. Модули коммуникаций обеспечивают простую, независимую от программно-аппаратной платформы поддержку передачи сообщений.

Взаимодействие организуется с помощью класса `AbstractCommunicator`, который предоставляет набор методов для передачи сообщений. Предполагается, что параллельное приложение состоит из нескольких процессов, идентифицируемых с помощью номера (ранга). Этот номер возвращается методом `getMyId()`. Перед отправкой сообщения на процесс-отправителе данные запаковываются во внутренний буфер с помощью перегруженной операции `<<`. После чего процесс-отправитель вызывает метод `send`, указывая в качестве параметра номер процесса-приемника, который, в свою очередь, вызывает метод `recv`.

На листинге 1 приведен фрагмент кода, демонстрирующий простейшее взаимодействие. Процесс с номером 0 пересылает процессу с номером 1 сообщение, содержащее целое число 111. Процесс с номером 1 принимает сообщение и распаковывает его содержимое в переменную `i`, затем печатает ее значение.

Листинг 1

```
1:  main(int argc, char* argv[]) {
2:      AbstractCommunicator ac(&argc, &argv);
3:      if(ac.getMyId() == 0) {
4:          ac << 111;
5:          ac.send(1);
6:      } else if(ac.getMyId() == 1) {
7:          int i;
8:          ac.recv(0);
9:          ac >> i;
10:         printf("%d\n", i);
11:     }
12: }
```

Модули стратегии ветвления

В методе ветвей и границ возможно применение различных стратегий ветвления. Стратегия ветвления определяет, какую из задач-кандидатов из списка выбирать для декомпозиции. Известны следующие основные стратегии:

1. Выбирается задача из числа сгенерированных на предыдущем шаге (одностороннее ветвление или ветвление в глубину).
2. Выбирается задача с наименьшим значением нижней оценки (в задачах на минимум) или наибольшим значением верхней (в задачах на максимум).
3. Выбирается задача, наименее удаленная от корня дерева ветвления.

Существуют также и другие стратегии. В библиотеке BNB-Solver предусмотрена возможность указания различных стратегий ветвления с помощью задания различных контейнеров для хранения вершин в качестве шаблонного параметра модуля, управляющего процессом ветвления. В частности, если в качестве контейнера указать стек, то ветвление будет односторонним, так как для ветвления всегда берется вершина стека, которая была сформирована на предыдущем шаге. Если требуется более сложная стратегия, то в качестве контейнера указывается очередь с приоритетом. При этом отношение порядка, задаваемое на конечных вершинах, определяет порядок выбора этих вершин из контейнера. Например, если нужно выбирать всегда задачу с наименьшей верхней оценкой, то вершины следует упорядочить в соответствии со значением этой оценки.

Модули распараллеливания

Распараллеливание метода ветвей и границ может осуществляться различными способами. Подробный обзор вариантов приводится в работах [4–6]. В рамках библиотеки BNB-Solver может быть реализован любой вариант, в котором каждый процесс параллельной программы управляет своим списком задач-кандидатов. Задачи-кандидаты передаются между процессами.

В настоящий момент реализована обычная схема управляющий-рабочие, в которой один из процессов является управляющим и занят распределением работы между остальными (рабочими) процессами. Распределение вычислений управляется *пороговыми значениями*, с помощью которых задается число ветвлений, которое требуется произвести до начала обмена данными. На управляющем процессе используются следующие пороговые значения:

- Master vertex threshold — максимальное число вершин в дереве ветвления;
- Master candidate problems threshold — максимальное число задач кандидатов.

При достижении одного из этих значений, управляющий процесс переключается с решения задачи на распределение задач по рабочим процессам. Поведение рабочего процесса управляется следующими параметрами:

- Slave vertex threshold — максимальное число вершин в дереве ветвления;
- Slave candidate problems threshold — максимальное число задач-кандидатов.

Рабочий процесс, получив очередную задачу-кандидата, выполняет ветвления до момента достижения одного из этих пороговых значений. После чего задачи-кандидаты пересылаются на управляющий процесс.

Проблемно-ориентированные модули

Проблемно-ориентированные модули инкапсулируют специфику конкретной задачи. Для включения задачи в библиотеку требуется реализовать класс, который предоставляет ряд проблемно-зависимых операций для организации процесса вычислений. Наиболее существенными из этих операций являются следующие:

- операция ветвления, реализующая декомпозицию задачи-кандидата на подзадачи;
- получение начальной задачи-кандидата (корня дерева ветвления);
- правила отсева.

На данный момент в библиотеке реализованы модули для задач о ранце с одним и несколькими ограничениями [7, 8], и для задачи коммивояжера [9].

Заключение

В статье приводится описание библиотеки BNB-Solver, которая позволяет разрабатывать последовательные и параллельные приложения, основанные на методе ветвей и границ. Библиотека была использована при создании программ, предназначенных для решения задач о ранце с одним и несколькими ограничениями и задачи коммивояжера.

В дальнейшем планируется расширить список приложений библиотеки, добавить в нее новые модули для балансировки нагрузки, ветвления и усовершенствовать существующие компоненты.

Автор выражает благодарность профессору И. Х. Сигалу, который познакомил автора с задачами оптимизации, сформулировал задачу и неоднократно обсуждал с автором алгоритмы, используемые в библиотеке.

Литература

1. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование. М.: ФИЗМАТЛИТ, 2002.
2. Евтушенко Ю. Г., Ратькин В. А. Метод половинных делений для глобальной оптимизации функции многих переменных // Техническая кибернетика. № 1. 1987. С. 119–126.
3. Трауструп Б. Язык программирования C++. М.: БИНОМ, 1999.

4. *Ananth G. Y., Kumar V., Pardalos P.* Parallel processing of discrete optimization problems // Encyclopedia of Microcomputers. 1993. Vol. 13. P. 129–147.
5. *Trienekens H., Bruin A.* Towards a taxonomy of parallel branch and bound algorithms. Report EUR-CS-92–01. Erasmus University, Rotterdam, 1992.
6. *Gendron B., Grainic T. G.* Parallel branch-and-bound survey and synthesis // Operations Research. 1994. Vol. 42. № 6. P. 1042–1066.
7. *Посыпкин М. А., Сигал И. Х., Галимянова Н. Н.* Алгоритмы параллельных вычислений для решения некоторых классов задач дискретной оптимизации. М.: ВЦ РАН, 2005.
8. *Посыпкин М. А., Сигал И. Х.* Исследование алгоритмов параллельных вычислений в задачах дискретной оптимизации ранцевого типа // ЖВМ и МФ. 2005. Т. 45. № 10. С. 1801–1809.
9. *Martello S., Toth P.* Knapsack Problems. John Wiley & Sons Ltd., England, 1990.
10. *Kellerer H., Pfershy U., Pisinger D.* Knapsack Problems. Springer-Verlag, 2004.

Приложение

Применение библиотеки BNB-Solver при генерации исходных данных для задачи об одномерном булевом ранце

Для проверки качества разработанного программного обеспечения требуется проводить большое число испытаний на различных наборах исходных данных. Для этих целей могут быть использованы наборы, взятые из постановок, встречавшихся в промышленных приложениях. Как правило, количество таких наборов невелико, поэтому для проведения сравнения на большом количестве примеров применяют генераторы исходных данных.

В работах [9, 10] предложены различные способы генерации исходных данных для задачи о ранце. Полученные наборы применялись для сравнения производительности последовательных алгоритмов. В случае параллельных алгоритмов появляется дополнительное требование к наборам исходных данных, состоящее в том, что сгенерированные задачи должны решаться достаточно долго, чтобы распараллеливание имело смысл и давало ощутимый эффект, который можно измерить. С другой стороны, сгенерированные примеры должны решаться за приемлемое время, чтобы вычислительный эксперимент можно было осуществить.

Для генерации исходных данных для задачи о булевом ранце было разработано параллельное приложение, работающее по следующей схеме. На управляющем процессе генерируются исходные данные очередной задачи. Сгенерированная задача пересылается одному из свободных рабочих процессов, который решает ее. Процесс решения на рабочем процессе завершается, если превышена верхняя граница для числа итераций, либо если задача полностью решена. Если число итераций не превысило верхней границы и не меньше заданной нижней границы, то исходные данные задачи пересылаются на управляющий процесс, сохраняет их в файле

на диске. Таким образом, генерируются задачи со сложностью решения, принадлежащей заданному диапазону.

Разработанное приложение позволяет подключать различные процедуры генерации исходных данных. На данный момент реализованы процедуры, описанные в [9]. Для решения задач на рабочих процессах используется модуль, предназначенный для решения задачи об одномерном булевом ранце с помощью метода ветвей и границ из библиотеки BNB-Solver. Обмены данными между рабочими и управляющим процессами реализованы с помощью коммуникационных модулей этой библиотеки. Сгенерировано большое число примеров размерности от 1 000 до 20 000 со сложностью решения до 50 000 000 ветвлений.