ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ ЗАДАЧ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ БЕЗОШИБОЧНЫХ ДРОБНО-РАЦИОНАЛЬНЫХ ВЫЧИСЛЕНИЙ

А.Н. Половинкин

В статье рассматривается параллельная реализация метода ветвей и границ для задачи целочисленного линейного программирования с использованием безошибочных дробно-рациональных вычислений. Важность использования точных алгоритмов для решения оптимизационных и геометрических задач обсуждается, например, в [4]. Исследуется время работы алгоритма в в зависимости от выбранной стратегии ветвления и выбора переменной для ветвления, а также эффективность распараллеливания для вычислительных систем с общей памятью.

РАССМАТРИВАЕМАЯ ЗАДАЧА И МЕТОД РЕШЕНИЯ

В работе рассматривается задача целочисленного линейного программирования (ЦЛП) в стандартной форме: $\max\,(c, x)\;\; x \in \{x \mid Ax \leqslant b, x \geqslant 0, x \in \mathbb{Z}^n\}$, где $A \in \mathbb{Z}^{m \times n},\ b \in \mathbb{Z}^m,\ c \in \mathbb{Z}^n$. К данному виду можно свести многие задачи комбинаторной оптимизации [2]: задачу о рюкзаке, задачу о раскрое, задачу коммивояжера, задачу о выполнимости КНФ и т.д. ЗЦЛП относится к классу NP-трудных, в настоящее время для нее неизвестны эффективные алгоритмы решения. Обычно в различных методах решения задач подобного типа используют эвристические соображения, основанные на интуиции и вычислительном эксперименте.

Одним из основных способов решения задач дискретной оптимизации является метод ветвей и границ [5]. Ключевая идея алгоритмов из данного класса заключается в использовании конечности множества вариантов (допустимых решений) и в переходе к сокращенному (направленному) перебору. Кроме того, важную роль играет оценка и отбрасывание неперспективных множеств вариантов.

В основе методов ветвей и границ лежит несколько принципов, позволяющих в широком классе случаев существенно уменьшить объем перебора: вычисление нижней границы (оценки) значения целевой функции на множестве допустимых решений (называемой *текущим рекордом*) или некотором его подмножестве, последовательное разбиение множества допустимых решений на постепенно уменьшающиеся подмножества (*ветвление*), пересчёт оценок и вычисление допустимых решений.

В дальнейшем изложении будут использоваться следующие обозначения: P^Z - исходная задача ЦЛП, P - ЛП-релаксация [1] задачи P^Z, P' - ЛП-релаксация текущей подзадачи, $\wp$ - список активных подзадач, l - текущий рекорд, $\hat{x}$ - текущее оптимальное решение.

Ниже приведено формальное описание метода ветвей и границ для решения задачи ЦЛП [3]:

1. Положить $\wp = \langle P \rangle,\ l = -\infty$.
2. Если $\wp = \emptyset$, то СТОП: если $l = -\infty$, то условия задачи P^Z несовместны; если $l \neq -\infty$, то $\hat{x}$ - оптимальный вектор, а l - оптимальное значение целевой функции.
3. Выбрать и удалить из $\wp$ задачу P'. Решить её.
4. Если условия задачи P' совместны, и $\tilde{x}$ - её решение, а $\tilde{x}_0$ - оптимальное значение целевой функции, причем $\tilde{x}$ целочислен и $\tilde{x}_0 > l$, то в качестве l взять $\tilde{x}_0$, а в качестве $\hat{x}$ взять $\tilde{x}$. Перейти на шаг 2.
5. Если $\tilde{x}$ нецелочислен, причем $\tilde{x}_0 > l$, то выбрать дробную компоненту $\tilde{x}_i$ и, добавив к ограничениям задачи P' неравенства $x_i \leqslant \lfloor \tilde{x}_i \rfloor$ и $x_i \geqslant \lceil \tilde{x}_i \rceil$ получить задачи P'_0 и P'_1 соответсвенно. Добавить P'_0 и P'_1 к $\wp$. Перейти на шаг 1.

Описанный процесс можно представить в виде корневого бинарного дерева. Корню этого дерева соответствует задача линейного программирования P. Если какой-либо вершине соответствует задача линейного программирования P' с оптимальным нецелочисленным вектором $\tilde{x}$, то её потомкам соответствуют задачи P'_0 и P'_1, полученные из P' приписыванием к ограничениям дополнительных неравенств $x_i \leqslant \lfloor \tilde{x}_i \rfloor$ и $x_i \geqslant \lceil \tilde{x}_i \rceil$ соответсвенно. Вершина является листом, если соответствующая ей задача имеет целочисленный оптимальный вектор либо условия задачи несовместны.

Одной из ключевых проблем в методе ветвей и границ является выбор *стратегии ветвления*: в каком порядке выбирать подзадачи из списка $\wp$ и какую из нецелочисленных переменных в оптимальном векторе $\tilde{x}$ выбирать для построения новых неравенств. В этой области мало теоретических результатов: для выбора хорошей стратегии обычно используют эвристические соображения, основанные на интуиции и вычислительном эксперименте.

В данной работе были рассмотрены следующие стратегии для выбора подзадачи из списка $\wp$:

1. *поиск в глубину*. В процессе работы алгоритма происходит спуск вниз до первого листа: на шаге 3 задача P' извлекается из начала списка, на шаге 5 задачи P'_0 и P'_1 добавляются в начало списка.
2. *поиск в ширину*. В процессе работы алгоритма происходит генерирование новых вершин и решение соответствующих им подзадач по ярусам поискового дерева: на шаге 3 задача P' извлекается из начала списка, на шаге 5 задачи P'_0 и P'_1 добавляются в конец списка.
3. *поиск по лучшей верхней оценке*. Выбирается задача из списка с наибольшей верхней оценкой.
4. *метод «лучшей проекции»* [6]. Основная идея данной эвристики заключается в попытке «предсказания», какая из задач может привести к наилучшему целочисленному решению, которое основано на оценке уменьшения значения целевой функции, необходимого для сведения к нулю условий нецелочисленности.

Для выбора дробной переменной при построении ветвления использовались следующие подходы:

1. ветвление по первой по порядку нецелочисленной координате оптимального решения задачи P'.
2. ветвление по последней по порядку нецелочисленной координате оптимального решения задачи P'.
3. ветвление по нецелочисленной координате оптимального решения задачи P', которой соответствует максимальная дробная часть: $\tilde{x}_i = \arg\max\{\{\tilde{x}_j\},\ 1-\{\tilde{x}_j\} \mid \tilde{x}_j \notin \mathbb{Z}\}$.
4. ветвление по способу Дрейбека-Томлина [1]. Основная идея данной эвристики заключается в том, чтобы оценить уменьшение значения целевой функции при выборе той или иной переменной в качестве базисной («заглядывание вперед» на один шаг симплекс-метода).

ИСПОЛЬЗОВАНИЕ БЕЗОШИБОЧНЫХ ДРОБНО-РАЦИОНАЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Как было сказано выше, на каждом шаге метода ветвей и границ происходит решение задачи линейного программирования. Большинство свободно распространяемых решателей ЦЛП (например, GLPK [12], LP_SOLVE [8]) используют в своей работе симплекс-метод либо метод внутренней точки, которые оперируют с вещественной арифметикой. Наряду с преимуществами (скорость работы) данный подход имеет существенный недостаток: из-за неизбежных ошибок округления, а также в случае, если возникающие подматрицы ограничений плохо обусловлены, найденные решения ЛП-релаксаций могут существенно отличаться от истинных решений, а это, в свою очередь, приводит к некорректной работе самого метода ветвей и границ. В качестве примера предлагается рассмотреть следующую задачу: $\max\ (s, x)\ \ x \in \{x \mid \tilde{H}x \leqslant \tilde{b}, x \geqslant 0, x \in \mathbb{Z}^n\}$, где $H=\{h_{ij}\}$ $h_{ij}=1/(i+j-1), i=1,\dots,n;\ j=1,\dots,n$ (матрица Гильберта), $k_i = \text{НОК}(i+j-1),\ j=1,\dots,n$ (наименьшее общее кратное знаменателей дробей *i*-ой строки матрицы Гильберта), h_i - *i*-я строка матрицы Гильберта, $\tilde{h}_i = h_i \cdot k_i$, $\tilde{H} = (\tilde{h}_1, \dots, \tilde{h}_n)$, $s=(1,\dots,1)^T$, $\tilde{b} = \tilde{H}s + s$. При решении данной задачи с использованием библиотек GLPK и LP_SOLVE метод выдает сообщения об ошибках после нескольких итераций.

Для того чтобы гарантированно избежать подобных эффектов, можно использовать безошибочные дробно-рациональные вычисления, которые реализованы в ряде библиотек (Arageli [9], GMP [7], LiDIA [10], NTL [11]). Обзор программного обеспечения для решения ряда оптимизационных и геометрических задач с использованием безошибочной арифметики см. в [4]. На базе библиотеки Arageli автором разработана реализация прямого и двойственного вариантов симплекс-метода, оперирующего с типом rational<big_int> (рациональные числа, числитель и знаменатель которых представляют собой целые числа произвольной длины), которая используется при решении задач линейного программирования в методе ветвей и границ.

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ

В работе рассматривается параллельная реализация метода ветвей и ганиц для решения задач ЦЛП на основе безошибочных дробно-рациональных вычислений для систем с общей памятью. Другие программные реализации алгоритмов решения задач ЦЛП на основе точных вычислений автору не известны.

Как видно из описания алгоритма, шаги 3-5 можно (за некоторым исключением) выполнять параллельно. При этом каждый поток (независимо от других) выбирает подзадачу из списка $\wp$ в соответствии с выбранной стратегией и решает её. Очевидно, что при данной схеме распараллеливания отсутствует необходимость в управляющем потоке: вычислительная загрузка будет распределена равномерно между вычислительными потоками. Проведённый эксперимент показал, что накладные расходы на организацию параллелизма, связанные с необходимостью блокировки списка $\wp$ при добавлении и удалении задач, несравнимо малы по сравнению с расходами на решение возникающих задач линейного программирования.

ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Рассматривались случайно сгенерированные задачи ЦЛП с числом переменных n=8,12,16,20 и числом ограничений $m=1/4\cdot n$. Коэффициенты целевой функции c_j, значения элементов a_{ij} и b_i генерировались случайно равномерно: $a_{ij}\in[10;200]$, $b_i\in[150\cdot n;250\cdot n]$, $c_j\in[-100,100]$. Для каждой размерности n генерировалось 100 задач. Вычисления проводились на системе следующей конфигурации: Core2 Quad Q6600 2.4 ГГц, RAM 2 Гб.

По результатам эксперимента с последовательной версией алгоритма выяснилось, что для данного класса задач оптимальным является следующая стратегия: выбирать из списка $\wp$ задачу с наилучшей верхней оценкой («жадная» стратегия), переменную для построения ветвления выбирать по способу Дрейбека-Томлина.

Ниже приведены данные по ускорению алгоритма в зависимости от числа используемых потоков (при n=20).

Таблица1. Ускорение параллельного алгоритма ветвей и границ при использовании оптимальной стратегии (поиск по лучшей верхней оценке, ветвление по способу Дрейбека-Томлина)

	Число потоков			
	1	2	3	4
Время вычислений, с	7,64	4,17	3,12	2,69
Ускорение		1,83	2,45	2,85
Среднее число решенных подзадач	76,49	81,64	90,45	99,87
Среднее число решенных подзадач на поток	76,49	40,82	30,15	24,96

Таблица 2. Ускорение параллельного алгоритма ветвей и границ при использовании неоптимальной стратегии (поиск в ширину, ветвление по способу Дрейбека-Томлина)

	Число потоков			
	1	2	3	4
Время вычислений, с	59,74	33,46	22,65	17,36
Ускорение		1,79	2,64	3,44
Среднее число решенных подзадач	540,36	594,2	597,67	604,43
Среднее число решенных подзадач на поток	540,36	291,1	199,22	151,11

Как видно из таблиц 1, 2, полученное ускорение меньше линейного. Это связано с ростом среднего числа подзадач, решенных каждым потоком в отдельности. Рост среднего числа решенных подзадач с ростом числа вычислительных потоков можно объяснить тем, что вычислительные потоки порождают некоторое число дополнительных «неперспективных» подзадач: для оптимальной стратегии ветвления доля таких новых «неперспективных» подзадач больше, чем для стратегии, при использовании которой процент таких подзадач велик сам по себе.

ЛИТЕРАТУРА:

1. Муртаф Б. Современное линейное программирование. М.: Мир, 1984
2. Ху Т. Целочисленное программирование и потоки в сетях. М.: Мир, 1974.
3. Шевченко В.Н., Золотых Н.Ю. Линейное и целочисленное линейное программирование. Нижний Новгород, Изд-во Нижегородского Госуниверситета, 2005.
4. Fukuda K. Exact algorithms and software in optimization and polyhedral computation // Proceedings of the twenty-first international symposium on Symbolic and algebraic computation. ACM, 2008. P. 333-334.
5. Land A., Doig A. An automatic method for solving discrete programming models // Econometrica. Vol. 28 No 3, 1960. P. 497-520.
6. Lee E.K., Mitchell J.E. Branch-and-bound methods for integer programming // Encyclopedia of Optimization. Volume II. Kluwer Academic Publishers, 2001. P. 509-519.
7. http://gmplib.org
8. http://lpsolve.sourceforge.net/5.5/
9. http://www.arageli.org
10. http://www.cdc.informatik.tu-darmstadt.de/TI/LiDIA
11. http://www.shoup.net/ntl
12. http://www.gnu.org/software/glpk/