УДК 519.68

А.В. СВИРИДЕНКО, О.А. ЩЕРБИНА

БЛОЧНЫЕ ЛОКАЛЬНЫЕ ЭЛИМИНАЦИОННЫЕ АЛГОРИТМЫ ДЛЯ РАЗРЕЖЕННЫХ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

Ключевые слова: *дискретная оптимизация, локальные элиминационные алгоритмы, декомпозиция, вычислительный эксперимент.*

ВВЕДЕНИЕ

Использование моделей и алгоритмов дискретной оптимизации (ДО) позволяет решать многие практические задачи, которые обычно имеют специальную структуру. При этом матрицы ограничений в задачах большой размерности, как правило, содержат большое количество нулевых элементов (сильно «разрежены»), а ненулевые матричные элементы часто группируются в блоки. Блочность многих прикладных задач ДО обусловлена слабой связностью подсистем моделируемых реальных сложных систем.

В настоящее время особенно актуальна разработка в ДО декомпозиционных подходов, позволяющих решать задачи большой размерности [1, 2]. Перспективными декомпозиционными методами, использующими разреженность матрицы ограничений задач ДО, являются локальные элиминационные алгоритмы (ЛЭА) [3], а для выделения специальных блочных структур задач ДО — такие графовые декомпозиционные подходы, как методы древовидной декомпозиции (ДД) [4–6]. Среди блочных структур особо выделим блочно-древовидную структуру, частным случаем которой является квазиблочная структура. Задачи с данной структурой использованы для проведения описанного ниже вычислительного эксперимента.

Целью настоящей работы является выявление реальных вычислительных возможностей локального алгоритма в сочетании с современными решателями задач ДО.

ЛОКАЛЬНЫЕ ЭЛИМИНАЦИОННЫЕ АЛГОРИТМЫ В ЗАДАЧАХ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

В работах [3, 7] предложен общий класс локальных элиминационных алгоритмов вычисления информации [8], имеющих декомпозиционный характер и позволяющих на основе вычисления локальной информации получать глобальную информацию о решении всей задачи. Локальный элиминационный алгоритм вычисляет информацию о локальных элементах структуры задачи, которая задается структурным графом, записывая локальную информацию об этих элементах в виде новых зависимостей, добавляемых к задаче, затем элиминируя просмотренные элементы и использованные зависимости.

Процедура ЛЭА, таким образом, разбивается на две части: прямую (элиминация элементов, вычисление и запоминание информации в виде локальных решений и получение в конце значения критерия) и обратную (нахождение глобального решения всей задачи по найденным в прямой части таблицам с локальными решениями, обеспечивающего достижение критерия в прямой части).

Процедуру локальной элиминации можно применить для элиминации не только отдельных переменных, но и множеств переменных в виде «блочной элиминации переменных» [7, 9], позволяющей элиминировать несколько переменных одновременно.

Использование метода сжатия подмножеств переменных в суперпеременные позволяет получить конденсированные или суперзадачи ДО, имеющие более простую структуру (например, древовидную), которые можно решить эффективнее.

ИССЛЕДОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ВОЗМОЖНОСТЕЙ ЛОКАЛЬНЫХ АЛГОРИТМОВ

Среди чрезвычайно важных вопросов при исследовании эффективности локального алгоритма (ЛА)[1] выделяется следующий: «Всегда ли применение ЛА в сочетании с некоторым алгоритмом ДО (для решения задач в блоках) эффективнее использования упомянутого алгоритма?».

Наряду с теоретическим анализом оценок эффективности значительный интерес представляет проведение соответствующего сравнительного вычислительного эксперимента ЛА с другими алгоритмами ДО. Понятно, что проведение исчерпывающего вычислительного эксперимента для ЛА в сочетании со всеми существующими алгоритмами ДО (или, по крайней мере, наиболее известными из них) чрезвычайно трудоемкий процесс.

О фреймворке SYMPHONY для решения задач частично-целочисленного линейного программирования. Для проведения вычислительного эксперимента по установлению вычислительных возможностей ЛЭА в сочетании с современными решателями реализован ЛЭА совместно с решателем фреймворка SYMPHONY [10], являющегося частью проекта COIN-OR [11] и использующегося для последовательного или параллельного решения задач частично-целочисленного линейного программирования (ЧЦЛП). Данный фреймворк был выбран из-за открытого исходного кода, переносимости, а также наличия технологии «теплого» старта (warm start), реализующей постоптимальный анализ (ПА) задач ЦЛП.

Отметим, что SYMPHOMY не имеет собственного решателя задач ЛП, но позволяет работать с такими решателями ЛП, как Clp, Cplex, Xpress, используя Osi-интерфейс. Кроме того, SYMPHONY поддерживает специализированные решатели для задачи коммивояжера, задачи маршрутизации, разбиения и др.

Технология «теплого» старта для реализации постоптимального анализа. Описываемая технология используется для реализации процедуры ПА такими современными решателями, как Gurobi, SCIP, CBC, SYMPHONY и др. В [12] исследованы возможности технологии теплого старта из фреймворка SYMPHONY для уменьшения перебора при решении подзадач, генерируемых вычислительной схемой ЛЭА.

В фреймворке SYMPHONY теплый старт реализован на основе компактного описания дерева поиска в момент приостановки вычислений. Используя его, пользователь может сохранять промежуточные вычисления на диске, а в дальнейшем применять их, перезапуская процесс вычислений после изменения данных задачи.

Отметим, что использование этой процедуры ПА не всегда гарантирует положительный результат. В случае SYMPHONY было замечено, что, как правило, процедура теплого старта работает лучше всего при незначительном изменении условий задачи.

Анализ результатов вычислительного эксперимента. Все вычисления проводились на базе процессора Intel Core 2 Duo @ 2.66 GHz, 2 GB ОЗУ и операционной системы Linux, версия ядра 2.6.35-24-generic. В качестве рабочего фреймворка для реализации ЛЭА использовалась библиотека SYMPHONY версии 5.4.1.

---

[1] Здесь под ЛА понимается локальный алгоритм декомпозиции, являющийся частной реализацией ЛЭА.

Результаты вычислительного эксперимента приведены в табл. 1, в которой n, m и k — соответственно число переменных, ограничений и блоков, s — размер перемычки, минимальное время решения задачи при использовании соответствующего алгоритма выделено жирным шрифтом.

Таблица 1

Номер задачи	Параметры задач				Время решения искусственно сгенерированных задач с квазиблочной структурой с использованием решателей, мин		
	n	m	k	s	SYMPHONY	SYMPHONY + ЛЭА	SYMPHONY + ЛЭА + ПА
1	180	12	6	1	9.503	0.028	**0.026**
2	180	12	6	2	3.019	**0.046**	0.047
3	180	12	6	4	1.164	0.493	**0.485**
4	180	12	6	5	**0.084**	5.667	5.295
5	180	12	6	6	**0.186**	12.4289	14.444
6	180	12	6	6	**2.7167**	5.321	5.057
7	240	24	12	3	3.831	0.16	**0.158**
8	240	24	12	3	6.958	0.290	**0.289**
9	240	24	12	3	1.057	0.133	**0.132**
10	240	24	12	3	2.751	**0.157**	0.162
11	240	24	12	4	1.45	0.477	**0.475**
12	240	20	10	4	22.143	**0.513**	0.517
13	240	6	3	6	**0.172**	8.361	7.488
14	240	6	3	6	**1.07**	8.447	7.688
15	240	16	8	6	**5.463**	11.422	11.681
16	240	16	8	6	**3.667**	12.046	11.572
17	320	12	6	6	31.065	24.313	**24.249**
18	320	12	6	6	44.556	**21.534**	21.636
19	320	16	8	1	9.605	0.025	**0.024**
20	320	20	10	1	24.435	0.029	**0.026**
21	320	20	10	2	18.464	**0.092**	0.093
22	320	20	10	3	31	0.289	**0.289**
23	320	20	10	6	**13.178**	15.519	15.146
24	320	20	10	6	23.853	14.327	**14.043**
25	320	40	20	1	3.943	0.016	**0.015**
26	320	40	20	2	2.519	**0.048**	0.049
27	320	40	20	3	3.382	0.158	**0.157**
28	320	40	20	3	1.807	0.182	**0.181**
29	420	6	3	1	2.965	**0.04**	0.041
30	420	6	3	1	4.686	0.057	**0.056**
31	420	6	3	2	5.69	**0.156**	0.157
32	420	6	3	2	15.116	0.251	**0.25**
33	420	6	3	3	4.201	0.682	**0.678**
34	420	6	3	3	10.211	0.899	**0.893**
35	420	6	3	4	3.703	**1.922**	1.933
36	420	6	3	4	4.101	**1.865**	1.878
37	420	6	3	5	17.812	**16.154**	16.201
38	420	6	3	5	19.145	2.696	**2.66**
39	420	6	3	5	6.185	**2.571**	2.584
40	420	6	3	6	**11.544**	12.111	12.112
41	420	6	3	6	**9.7**	23.254	23.234
42	420	60	30	1	5.51	0.019	**0.018**
43	420	60	30	2	14.414	0.066	**0.065**
44	500	50	25	1	90.02	**0.021**	0.022

Описание тестовых задач. В ходе эксперимента все тестовые задачи ЦЛП с бинарными переменными имели искусственно сгенерированную квазиблочную структуру, а все блоки в отдельно взятой задаче — одинаковое количество переменных, а также одинаковое число переменных в перемычках между ними. Величина перемычек варьировалась для одних и тех же параметров задачи, что необходимо для оценки влияния ПА на время решения задачи при увеличении размера перемычек.

Тестовые задачи ЦЛП генерировались по заданному числу переменных, ограничений и размеру перемычек между блоками. Исходя из количества переменных и ограничений, вычислялись размеры и число блоков. Далее с помощью датчика случайных чисел генерировались коэффициенты целевой функции, а также коэффициенты матрицы ограничений и правых частей для каждого блока.

Каждая тестовая задача решалась с помощью трех алгоритмов: базового решателя SYMPHONY для задач ЧЦЛП, использующего интерфейс OsiSym; ЛЭА в сочетании с решателем SYMPHONY, с помощью которого решались выделенные подзадачи, соответствующие блокам; ЛЭА в сочетании с решателем SYMPHONY с поддержкой ПА (применялась описанная технология теплого старта). Во всех трех случаях решатель SYMPHONY использовал препроцессинг (preprocessing).

В ходе проведенного вычислительного эксперимента было установлено явное преимущество ЛЭА в сочетании с решателем SYMPHONY при решении квазиблочных задач с небольшими перемычками по сравнению с решателем SYMPHONY (см. табл. 1), а также обнаружено, что при увеличении размера перемычек в задачах с одним и тем же количеством переменных и размером блоков ЛЭА становился менее эффективным вследствие увеличения объема перебора при решении подзадач в блоках. Для повышения эффективности ЛЭА при решении таких задач использовалась технология теплого старта. Поскольку задачи ЦЛП, соответствующие одному и тому же блоку, для разных значений переменных в перемычках отличаются одна от другой лишь правыми частями, предполагалось, что использование технологии теплого старта поможет не решать полностью каждую задачу при переборе значений переменных из перемычек, а лишь частично, используя информацию, полученную при решении других задач, что должно было в принципе увеличить производительность ЛЭА. Однако результат эксперимента оказался неоднозначным. Время решения большинства задач было практически сопоставимо как с использованием технологии теплого старта, так и без него. Но для некоторых тестовых задач данная технология была эффективной (см. табл. 1, задачи 7, 17, 24), а для других — оказалась неэффективной (см. табл. 1, задачи 5, 15, 40), что выражалось в повышении и соответственно снижении производительности ЛЭА и измерялось временем, затрачиваемым на решение задачи.

ЗАКЛЮЧЕНИЕ

Выявлены реальные вычислительные возможности блочных локальных элиминационных алгоритмов (ЛЭА) в сочетании с решателем SYMPHONY при решении квазиблочных задач целочисленного линейного программирования (ЦЛП). В ходе проведенного вычислительного эксперимента установлено явное преимущество ЛЭА в сочетании с решателем SYMPHONY при решении задач с небольшими перемычками по сравнению с решателем SYMPHONY, а также обнаружено, что при увеличении размера перемычек в задачах с одним и тем же количеством переменных и размером блоков ЛЭА становился менее эффективным вследствие увеличения объема перебора при решении подзадач в блоках.

Для повышения эффективности ЛЭА при решении таких задач использовалась технология теплого старта, реализующая постоптимальный анализ.

Перспективно не только продолжение данного исследования, но и анализ возможности технологии постоптимального анализа, предоставляемой другими современными решателями. Представляет также интерес изучение вычислительных возможностей локальных алгоритмов при решении разреженных задач ЦЛП из реальных приложений.

СПИСОК ЛИТЕРАТУРЫ

1. Xu J., Jiao F., Berger B. A tree-decomposition approach to protein structure prediction // Proc. of Comput. Syst. Bioinform. Conf. — 2005. — P. 247–256.
2. Khanafer A., Clautiaux F., Talbi El-G. Tree-decomposition based heuristics for the two-dimensional bin packing problem with conflicts // Comput. & OR. — 2012. — **39**, N 1. — P. 54–63.
3. Щербина О.А. Локальные элиминационные алгоритмы решения разреженных дискретных задач // Журн. вычисл. математики и мат. физики. — 2008. — **48**, № 1. — С. 161–177.
4. Щербина О.А. Древовидная декомпозиция и задачи дискретной оптимизации (обзор) // Кибернетика и системный анализ. — 2007. — № 4. — С. 102–118.
5. Bodlaender H.L., Koster A.M.C.A. Treewidth computations I: Upper bounds // Inform. and Comput. — 2010. — **208**. — P. 259–275.
6. Быкова В В. Вычислительные аспекты древовидной ширины графа // Прикл. дискрет. математика. — 2011. — № 3(13). — С. 65–79.
7. Shcherbina O. Graph-based local elimination algorithms in discrete optimization // Foundations of Computational Intelligence, Vol. 3. Global Optimization Series: Studies in Computational Intelligence, Vol. 203 / A. Abraham, A.-E. Hassanien, P. Siarry, A. Engelbrecht (eds.). — Berlin; Heidelberg: Springer, 2009. — P. 235–266.
8. Журавлев Ю.И. Избранные научные труды. — М.: Магистр, 1998. — 420 с.
9. Bertele U., Brioschi F. Nonserial dynamic programming. — New York: Acad. press, 1972. — 235 p.
10. https://projects.coin-or.org/SYMPHONY.
11. http://www.coin-or.org.
12. http://coin-or.org/download/source/SYMPHONY/.

Поступила 01.11.2011
После доработки 03.12.2012