

**Параллельная реализация алгоритма выделения
оптимальной совместной подсистемы системы
линейных неравенств**

Катериночкина Н. Н., Шмаков А. С.

nkater@ccas.ru, ashmak@mail.ru

В процессе оптимизации некоторых моделей алгоритмов распознавания требуется как можно точнее удовлетворить определенной системе условий, которая описывается достаточно большим числом линейных неравенств и в целом может быть противоречивой. В общем случае решение такой проблемы сводится к выделению максимальной по мощности совместной подсистемы из заданной системы линейных неравенств.

Постановка задачи

Пусть задана система линейных неравенств:

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = 1, \dots, m, \quad (1)$$

разбитая на блоки одинаковой длины l . Система (1) несовместна. Требуется найти совместную подсистему, содержащую максимальное число полных блоков заданной длины l . В частности, при $l = 1$ осуществляется поиск максимальной совместной подсистемы системы (1).

Разработан алгоритм, который осуществляет поиск совместной подсистемы, содержащей максимальное число полных блоков. В работах [1] и [2] доказывается, что для выделения всех нерасширяемых совместных подсистем системы (1) достаточно перебрать все ее подсистемы мощности r и ранга r . Именно на этом утверждении основана работа алгоритма, состоящая в переборе всех подсистем мощности r и ранга r системы (1). Для каждой из таких подсистем требуется решить несложную задачу: найти одно узловое решение. Для этого надо заменить все знаки неравенств в подсистеме на равенства и найти одно решение полученной системы линейных уравнений. Найденное узловое решение подсистемы подставляется во все неравенства системы (1) и выделяется из нее те неравенства, которым это решение удовлетворяет. Выделенные неравенства образуют совместную подсистему. Перебрав таким образом все подсистемы, получаем некоторое множество совместных подсистем, среди которых будут все нерасширяемые совместные подсистемы. Среди них можно выбрать оптимальные подсистемы с интересующими свойствами.

Однако, трудоемкость поставленной задачи равна перебору $C_m^r = \frac{m!}{(m-r)!r!}$ r -подсистем. При больших значениях m эта задача требует

больших вычислительных ресурсов и времени. Для ее решения предлагается построить параллельный алгоритм.

Параллельный алгоритм

Данную задачу удается достаточно легко и эффективно распараллелить на уровне данных, поскольку её решение состоит в переборе всех возможных r -подсистем системы (1). При этом задача рассмотрения очередной подсистемы является по сути самостоятельной задачей, включающей в себя следующие этапы:

- 1) поиск ранга подсистемы;
- 2) сравнение полученного ранга с рангом всей системы;
- 3) если ранг меньше, то происходит переход к следующей подсистеме;
- 4) если ранг равен, то подсистема является r -подсистемой, и далее ищется узловое решение и соответствующая ему подсистема неравенств.

Предлагается разбить множество всех подсистем на непересекающиеся подмножества и параллельно решать поставленную задачу для каждого из подмножеств.

Для эффективного распараллеливания необходимо добиться максимальной загрузки всех доступных процессоров. Для этого разделим множество всех возможных r -подсистем следующим образом. Введем булевский вектор $\gamma = (\gamma_1, \dots, \gamma_m) \in E^m$, который будет описывать конкретную r -подсистему системы (1), где $\gamma_i = 1$, если неравенство с номером i входит в подсистему, и $\gamma_i = 0$ иначе. Будем перебирать все возможные векторы γ , такие что $|\gamma| = r$. Пусть $\Lambda = \{\gamma \in E^m : |\gamma| = r\}$, тогда $|\Lambda| = C_m^r$. Используя известную формулу для биномиальных коэффициентов:

$$C_m^r = C_{m-k}^r + C_{m-k}^{r-1} + \dots + C_{m-k}^{r-k}, \quad (2)$$

будем делить множество всех векторов Λ на подмножества. Так, из (2) при $k = 1$ следует $\Lambda = \Lambda_1 \cup \Lambda_2$, где

$$\begin{aligned} \Lambda_1 &= \{\gamma \in \Lambda \mid \gamma_1 = 1, \sum_{i=2}^m \gamma_i = r - 1\}; \\ \Lambda_2 &= \{\gamma \in \Lambda \mid \gamma_1 = 0, \sum_{i=2}^m \gamma_i = r\}. \end{aligned}$$

Аналогично образом можно разделить множество Λ на $l = 2^q$ подмножеств. В результате получится упорядоченный набор булевских векторов, который представляет собой булевский куб размерности q .

Пусть у нас имеется p доступных процессоров. Выделяем один из них для основного или родительского процесса. Остальные $p - 1$ будут использоваться для вычислений. Назовем их *пулом свободных процессоров*. Основной процесс начинает работу с того, что определяет

Кол-во проц.	Время (1 проц.)	Время (p проц.)	Ускорение	Эффективность
2	5646, 12	3029, 15	1, 86	0, 932
4	5646, 12	1718, 26	3, 29	0, 821
8	5646, 12	930, 88	6, 07	0, 758
9	5646, 12	794, 40	7, 11	0, 790
10	5646, 12	739, 18	7, 64	0, 764
16	5646, 12	504, 07	11, 20	0, 700
20	5646, 12	405, 92	13, 91	0, 695
25	5646, 12	496, 85	11, 36	0, 455
30	5646, 12	305, 04	18, 51	0, 617
31	5646, 12	275, 56	20, 49	0, 661
32	5646, 12	284, 50	19, 85	0, 620

Таблица 1. Показатели эффективности распараллеливания.

$q = \lceil \log_2(p - 1) \rceil + 1$ и делит множество Λ на 2^q подмножеств. Не ограничивая общности, будем считать, что задача задана корректно, т. е. выполнены следующие ограничения: $r \geq q - 1$, $m \gg r$, иначе задача имеет достаточное простое решение и не требует больших вычислительных мощностей.

Итак, главный процесс формирует нетривиальные задания (т. е. непустые подмножества) на решения задачи поиска максимальной совместной подсистемы и узлового решения для задачи меньшей размерности, затем отдает эти задания процессорам, причем они переходят в *пул занятых процессоров*. В начальный момент все процессы получают задание, поскольку задача разбивается на большое число подзадач, чем количество процессов. Каждый из них решает самостоятельную задачу меньшей размерности либо по количеству неравенств, либо по рангу системы, либо и по тому, и по другому. Завершив свое выполнение, каждый из процессов возвращает результаты в основной процесс и переходит в пул свободных процессов. Если остались еще задания, то главный процесс выдает последовательно произвольным процессам, попадающим в пул свободных процессов, новое задание на обработку. Если заданий больше нет, главный процесс посылает сигнал на уничтожение процессу.

Главный процесс собирает результаты от всех процессов по каждому из заданий и последовательно выбирает максимальное размер совместной подсистемы, а также номера соответствующих неравенств и узловое решение. После обработки всех заданий главный процесс выдает результат задачи.

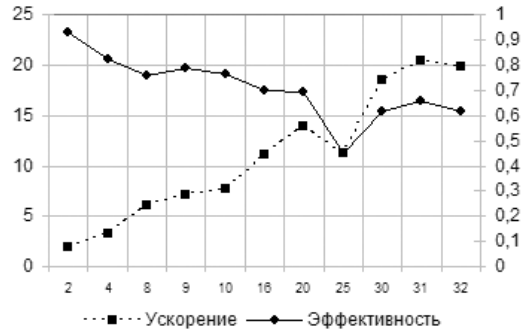


Рис. 1. Зависимость ускорения и эффективности работы алгоритма от количества процессоров.

Результаты

Данная задача достаточно просто и эффективно распараллеливается на уровне данных, т. е. для решения всей задачи надо перебрать некоторое множество, решив для каждой выборки некоторую самостоятельную задачу, тогда как при параллельном подходе мы можем разделить исходное множество на подмножества и параллельно решать ту же задачу для каждого из них. В идеальном случае при делении множества на абсолютно одинаковые подмножества (с точки зрения их сложности, как вычислительных задач, а не с точки зрения их мощности, как множеств), можно получить идеальный случай распараллеливания. Однако, заранее невозможно определить вычислительную сложность каждого из подмножеств. В предложенном подходе исходное множество заранее делится на количество подмножеств, превышающих количество свободных процессоров. В начальный момент каждый из процессоров получает задачу на обработку подмножества. Закончив свою работу, он либо завершается, либо получает новое задание. Процессоры, получившие «большие» задания, будут работать только над ними; процессоры, получившие «небольшие» задания, смогут обработать несколько из них.

Для оценки эффективности распараллеливания будем применять следующие показатели (см. Таблицу 1, Рис. 1):

- Ускорение $A_p = \frac{T_1}{T_p}$, где T_p — время исполнения распараллеленной программы на p процессорах, T_1 — время исполнения исходной программы.
- Эффективность $S_p = \frac{T_1}{pT_p}$, показывающая долю использования процессоров.

Вычислительные эксперименты проводились на многопроцессорном вычислительном комплексе MVS-15000BM, установленном в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН).

Работа выполнена при поддержке проектов РФФИ № 05-07-90333, Целевой программы № 14 Президиума РАН, Целевой программы № 2 Отделения математических наук РАН.

Литература

- [1] Катериночкина Н. Н. Методы выделения максимальной совместной подсистемы системы линейных неравенств. Сообщение по прикладной математике. — Москва: Вычислительный центр РАН, 1997.
- [2] Черников С. Н. Линейные неравенства. — Москва: Наука, 1968.
- [3] Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002.