О масштабируемости и эффективности одного метода решения задачи о ранце в распределенной вычислительной среде*

Р. М. Колпаков[1], М. А. Посыпкин[2]

[1] *МГУ им. М. В. Ломоносова*
[2] *Центр Грид-технологий и распределенных вычислений Института системного анализа РАН*

Работа посвящена теоретическому исследованию эффективности решения задачи о ранце в распределенной вычислительной среде. Обобщено традиционное понятие масштабируемости параллельных систем. Исследована асимптотика поведения параллельной сложности одного варианта реализации метода ветвей и границ решения задачи о ранце. Показано, что масштабируемость отсутствует для всего множества задач о ранце, но может иметь место для различных семейств.

Введение

Задачи дискретной оптимизации, также называемые задачами целочисленного программирования, состоят в нахождении максимального (минимального) значения функции $f(x)$ на конечном множестве G. К этому классу относятся классические задачи коммивояжера, задача о ранце, покрытии графа и многие другие [1]. В настоящее время разработано большое число алгоритмов, которые позволяют эффективно решать такие задачи: методы динамического программирования, ветвей и границ, отсечений и др. Все перечисленные алгоритмы основаны на идее сокращенного перебора элементов множества G, отличающегося от полного тем, что в процессе работы алгоритма различные подмножества G исключаются и их элементы не рассматриваются. Известно, что число шагов перечислен-

* Работа выполнена при поддержке гранта РФФИ № 09-07-00352-а и Совета по грантам Президента Российской Федерации (грант НШ-5511.2008.9).

ных алгоритмов, необходимых для нахождения точного решения задачи и доказательства его оптимальности, сильно зависит от исходных данных задачи. При этом, сложность работы алгоритмов колеблется в очень широких пределах: от одного до экспоненциального по числу переменных задачи числа шагов.

В связи с высокой вычислительной сложностью для решения задач оптимизации часто применяют методы параллельных [2] и распределенных [3] вычислений. К классу распределенных систем относятся получившие в последнее время широкое распространение *Грид-системы*, состоящие из множества разнородных по архитектуре и производительности вычислительных узлов, связанных между собой посредством Интернета.

Грид-системы главным образом предназначены для решения задач, которые можно представить в виде большой совокупности слабосвязанных подзадач, которые можно запускать независимым образом. При этом, каждая подзадача получает некоторые данные на вход, обрабатывает их и сохраняет результат. Результаты расчетов можно затем подвергнуть дальнейшей обработке. Передача данных в основном осуществляется через хранилища файлов, поэтому связанные с ней накладные расходы данных могут существенно снизить скорость работы алгоритма. В связи с этим, при реализации предпочтение следует отдавать вычислительным схемам, не предполагающим интенсивный обмен данными между подзадачами. Также, поскольку запуск заданий в Грид связан со значительными задержками, целесообразно минимизировать количество запускаемых подзадач.

В данной работе исследуется реализация *метода ветвей и границ* (МВГ) [1] в Грид-системах. МВГ состоит в пошаговой декомпозиции исходной задачи на подзадачи с исключением из рассмотрения подзадач, заведомо не содержащих оптимальных решений. Наиболее простой и подходящий для Грид-систем способ организации вычислений состоит в следующем: первоначально управляющий узел выполняет некоторое число ветвлений и формирует некоторый список подзадач. Затем полученные подзадачи рассылаются остальным (рабочим) вычислительным узлам. На рабочих узлах полученные подзадачи решаются полностью, после чего наилучшие найденные на рабочих узлах решения передаются управляющему процессу. Управляющий процесс обрабатывает полученные решения, выбирая оптимальное. Далее подробно рассматривается указанная схема организации вычислений, исследуются характеристики ее производительности на примере *задачи о ранце* [4,5]. Эта задача является одной из наиболее изученных задач дискретной оптимизации и, как правило, используется в качестве теста для различных систем решения задач оптимизации в распределенной вычислительной среде [6–9].

1. Анализ эффективности параллельных приложений

Анализ эффективности использования высокопроизводительных вычислительных систем является неотъемлемой частью разработки и тестирования параллельных приложений, используется владельцами кластерных систем коллективного доступа для формирования рекомендаций пользователям, оценки целесообразности предоставления ресурсов[10]. Для проведения подобных исследований был разработан теоретический аппарат [2], базовыми понятиями которого являются *ускорение* и *эффективность*. Приведем определения этих понятий. Рассмотрим некоторую вычислительную задачу A, время решения которой на одном процессоре эталонным алгоритмом составляет $T_s(A)$. Ускорение $S_m(\xi, A)$ параллельного алгоритма ξ при решении задачи A на системе из m процессоров задается отношением времени работы эталонного последовательного алгоритма на одном процессоре ко времени решения $T_m(\xi, A)$ задачи A параллельным алгоритмом ξ на данной системе:

$$S_m(\xi, A) = \frac{T_s(A)}{T_m(\xi, A)} \ . \qquad (1)$$

Эффективность работы параллельного алгоритма ξ при решении задачи A определяется как отношение ускорения к числу задействованных процессоров:

$$E_m(\xi, A) = \frac{S_m(\xi, A)}{m} \,. \qquad (2)$$

Важнейшую роль при анализе параллельных алгоритмов играет изучение асимптотического поведения эффективности при росте числа процессоров и размерности задачи. Обычно наблюдается снижение эффективности при росте числа процессоров m из-за возрастающих потерь производительности, вызванных передачей данных. Также, как правило, увеличение размерности задачи $|A|$ приводит к снижению доли коммуникаций по отношению к вычислениям и, как следствие, к увеличению эффективности. Под размерностью задачи понимается некоторый параметр, характеризующий число операций последовательного алгоритма. Например, в случае матричных вычислений таким параметром может служить число строк и столбцов матрицы.

В работе [2] вводится *функция изоэффективности* $I(m)$, определяющая соотношение между числом процессоров и размерностью решаемой задачи, при котором эффективность остается постоянной. Задача считается *маситабируемой*, если существует функция изоэффективности.

В классических алгоритмах решения систем линейных уравнений, которые главным образом используются для иллюстрации моделей производительности в литературе, информационная структура алгоритма однозначно определяется числом неизвестных n. Поэтому при заданном n времена $T_s(A)$, $T_m(\xi, A)$ и производные величины (1, 2) определяются однозначным образом. В отличие от задач линейной алгебры, число операций и структура дерева ветвления метода ветвей и границ существенно зависят не только от числа переменных задачи, но и от значений коэффициентов [11]. Для анализа эффективности подобных алгоритмов требуется обобщить определение масштабируемости следующим образом. Пусть имеется класс задач $\mathbf{A}$. Введем понятие *предельной эффективности* $E^*(\xi, \mathbf{A})$ алгоритма ξ для семейства задач $\mathbf{A}$ как предел:

$$E^*(\xi, \mathbf{A}) = \lim_{M\to\infty} \inf_{m\geq M} \left(\lim_{s\to\infty} \inf_{A\in\mathbf{A}, |A|\geq s} E_m(\xi, A) \right). \tag{3}$$

Будем говорить, что *параллельный алгоритм* ξ *масштабируем*, если $E^*(\xi, \mathbf{A}) > 0$. Заметим, что предел (3) всегда существует, так как функции

$$\inf_{A\in\mathbf{A}, |A|\geq s} E_m(\xi, A) \text{ и } \inf_{m\geq M} \left(\lim_{s\to\infty} \inf_{A\in\mathbf{A}, |A|\geq s} E_m(\xi, A) \right)$$

ограничены сверху и не убывают. Далее в работе исследуется предельная эффективность одного параллельного варианта реализации МВГ для задачи о ранце.

2. Метод ветвей и границ для задачи о ранце

Задача о ранце формально записывается следующим образом:

$$\begin{cases} f(x) = \sum_{i=1}^{n} p_i x_i \to \max, \\ \sum\limits_{i=1}^{n} w_i x_i \leq C, \\ x_i \in \{0,1\}, 1 \leq i \leq n. \end{cases} \tag{4}$$

В данной постановке через p_i и w_i обозначены стоимость и вес i-го предмета, помещаемого в ранец грузоподъемностью C. Задача состоит в определении набора предметов максимальной суммарной стоимости, который можно разместить в ранце. Дадим общее описание метода ветвей и границ

[1, 4, 5] для решения задачи о ранце. В процессе работы алгоритма поддерживаются следующие данные: *рекорд*, т. е. наибольшее найденное на данный момент времени значение целевой функции $f(x)$, *рекордное решение* на котором достигается рекорд и текущий список *подзадач*, на которые разбита исходная задача. Подзадачи получаются из задачи (4) некоторым переменным фиксированных значений. Приведем более формальное описание МВГ.

Данные: рекорд f^0, рекордное решение x^0, список подзадач.

Шаг 1. В список подзадач помещается исходная задача. Рекорд полагается равным 0.

Шаг 2. Если список подзадач пуст, то алгоритм завершается. В противном случае выбирается подзадача *P* из списка подзадач. Подзадача *P* удаляется из списка.

Шаг 3. Проверяется, выполнены ли для выбранной подзадачи *P* указанные далее в работе условия отсева. Если хотя бы одно из условий отсева выполнено, то осуществляется переход к шагу 2. При необходимости на этом шаге также обновляются рекорд и соответствующее рекордное решение.

Шаг 4. Выбранная подзадача подвергается декомпозиции. Для этого выбирается переменная x_b, называемая *переменной ветвления*. Подзадача *P* разбивается на две подзадачи P_0 и P_1, получаемые присваиванием переменной x_b значений *0* и *1* соответственно. Построенные подзадачи P_0 и P_1 помещаются в список подзадач и осуществляется переход к шагу 2.

Результатом работы алгоритма является окончательное рекордное решение. Заметим, что работа описанного нами алгоритма существенным образом зависит от процедуры выбора очередной подзадачи из списка подзадач и процедуры выбора переменной ветвления для декомпозиции выбранной подзадачи. Алгоритм, для которого данные процедуры строго определены, будем называть *вариантом* метода ветвей и границ.

Будем далее рассматривать стандартный вариант МВГ для задачи о ранце, где применяются следующие правила отсева:

- суммарный вес предметов, положенных в ранец, превосходит ограничение;
- вес оставшихся предметов не больше ограничения;
- решение оценочной задачи линейной релаксации не больше чем рекорд.

3. Характеристики производительности решения задачи о ранце последовательным и параллельным вариантом МВГ

Процесс решения задачи о ранце методом ветвей и границ можно представить в виде *дерева ветвления*, вершинам которого соответствуют рассматриваемые в процессе решения подзадачи. Корнем дерева ветвления полагается вершина, соответствующая исходной задаче. Вершина, соответствующая подзадаче P, соединяется дугами с вершинами, соответствующими подзадачам P_0 и P_1, полученными из P в результате декомпозиции.

Пусть A — некоторая задача. Обозначим через $|A|$ количество вершин в дереве ветвления при решении этой задачи последовательным вариантом МВГ. Это число также будем называть *сложностью решения задачи* A. Далее ограничимся рассмотрением таких задач, сложность решения которых не зависит от порядка выбора вершин для ветвления. К этому классу относится, в частности, *задача о сумме подмножеств* [4,5]. Если t_b — время обработки одной вершины, то время решения всей задачи на однопроцессорной машине примерно составляет $T_s(A) = |A| t_b$.

Для определения понятия *параллельной сложности* рассмотрим параллельную систему из m одинаковых процессоров. Будем считать, что процессоры работают по шагам. За один шаг процессор делает одно из двух действий:

- обрабатывает одну подзадачу (отбрасывает, либо производит декомпозицию на подзадачи);
- простаивает.

Будем считать, что шаги на разных процессорах выполняются синхронно, все процессоры одновременно начинают и заканчивают решение задачи в момент окончания обработки последней подзадачи. Таким образом, все процессоры выполняют одинаковое количество шагов, которое будем называть *параллельной сложностью решения задачи* A заданным параллельным алгоритмом ξ и обозначать через $P_m(\xi, A)$. Также определим понятие *коммуникационной сложности* $R_m(\xi, A)$ решения задачи A параллельным алгоритмом ξ, как число подзадач, переданных с одного процессора на другой.

Пусть время передачи одной подзадачи по сети t_c и пусть коммуникационное оборудование не позволяет передавать больше B подзадач одновременно. Тогда для времени параллельной обработки $T_m(\xi, A)$ справедливы следующие наравенства:

$$P_m(\xi, A)t_b + \frac{R_m(\xi, A)}{B} t_c \le T_m(\xi, A) \le P_m(\xi, A)t_b + R_m(\xi, A)t_c. \qquad (5)$$

В работе ограничимся рассмотрением следующего класса параллельных алгоритмов. Из всего множества процессоров системы выделим *управляющий процессор*, остальные процессоры будут *рабочими*. Управляющий процессор выполняет t шагов МВГ, генерируя при этом m подзадач

$A_i = \{A_1, \ldots, A_m\}$. После этого полученные подзадачи рассылаются рабочим процессорам, кроме одной, которая остается на управляющем процессоре. Рабочие процессоры, получив подзадачу, полностью ее решают. Управляющий процессор полностью решает оставленную у себя подзадачу. Алгоритм заканчивает свою работу, когда все процессоры закончат решение полученных подзадач. Будем считать, что для ветвления на управляющем процессе каждый раз выбирается задача, соответствующая наиболее удаленной от корня вершине построенной части дерева ветвления. Такой вариант реализации МВГ назовем *фронтальным*. Очевидно справедливы следующие соотношения для параллельной и коммуникационной сложностей:

$$\begin{aligned} P_m(\xi, A) &= t + \max_{i=1,\ldots,m} |A_i|, \\ R_m(A) &= m-1. \end{aligned} \tag{6}$$

Подставляя выражения из (6) в (5), получим:

$$\left(t + \max_{i=1,\ldots,m} |A_i|\right) t_b + \frac{m-1}{B} t_c \le T_m(\xi, A) \le \left(t + \max_{i=1,\ldots,m} |A_i|\right) t_b + (m-1) t_c \,. \tag{7}$$

Так как $\max |A_i| \ge \frac{|A|}{m}$, то $T_m(\xi, A) \approx \left(t + \max_{i=1,..,m} |A_i|\right) t_b$ при $|A| \to \infty$. Следовательно, справедливо соотношение:

$$\varphi_m = \lim_{s\to\infty} \inf_{A\in\mathbf{A},|A|\ge s} E_m(\xi, A) = \lim_{s\to\infty} \inf_{A\in\mathbf{A},|A|\ge s} \frac{|A|}{\left(t + \max_{i=1,..,m} |A_i|\right) m}. \tag{8}$$

Покажем, что значение эффективности определяется выбором множества рассматриваемых задач $\mathbf{A}$. Рассмотрим известное семейство задач $\{\Phi(n,k)\}$, введенное в работе[12]. Задача $\Phi(n,k)$ формулируется следующим образом:

$$\begin{cases} \sum_{i=1}^{n} a x_i \to \max, \\ \sum_{i=1}^{n} a x_i \le ka + 1, \\ x_i \in \{0,1\}, i = 1,..,n, \\ \qquad a > 1. \end{cases} \tag{9}$$

Легко показать [12], что последовательная сложность решения этой задачи составляет $|\Phi(n,k)| = 2\binom{n}{k+1} - 1$. В результате применения ветвления к этой задаче получатся две новые задачи $\Phi(n-1,k)$ и $\Phi(n-1,k-1)$. Пусть $\lim\limits_{n\to\infty} \frac{k}{n} = \alpha$, тогда

$$\lim_{n\to\infty} \frac{\left|\Phi(n-1,k)\right|}{\left|\Phi(n,k)\right|} = \lim_{n\to\infty} \binom{n}{k+1} \Big/ \binom{n+1}{k+1} = \lim_{n\to\infty} \frac{n-k}{n+1} = 1-\alpha;$$

$$\lim_{n\to\infty} \frac{\left|\Phi(n-1,k-1)\right|}{\left|\Phi(n,k)\right|} = \lim_{n\to\infty} \binom{n}{k} \Big/ \binom{n+1}{k+1} = \lim_{n\to\infty} \frac{k+1}{n+1} = \alpha.$$

Таким образом, в результате ветвления задача $\Phi(n,k)$ разбивается на две подзадачи со сложностями, отношение которых асимптотически стремится к $\alpha/(1-\alpha)$. Рассмотрим систему из m процессоров и исследуем асимптотическое поведение параллельной сложности решения задачи $\Phi(n,k)$ при $n\to\infty$, $\frac{k}{n}\to\infty$. Так как исследуется асимптотическое поведение сложности, то можно без ограничения общности считать, что $m < \min(k, n-k)$. Поэтому, для получения m подзадач потребуется ровно $t = m-1$ шагов МВГ. При этом концевым вершинам соответствуют подзадачи вида $\Phi(n-p, k-q)$, где $0 \le q \le p \le m-1$. Так как величины p и q ограничены при росте n, то можно считать, что каждая подзадача в начальном дереве ветвлений разбивается на две, асимптотические сложности решения которых находятся в отношении $\alpha/(1-\alpha)$.

Каждой вершине u дерева ветвления, за исключением корневой, припишем асимптотику $r(u)$ отношения сложности решения соответствующей подзадачи ко сложности решения исходной задачи. Таким образом, корневой вершине будет приписано значение 1, а каждым двум дочерним вершинам любой внутренней вершины u будут приписаны величины $\alpha r(u)$ и $(1-\alpha)r(u)$. Тогда если $u_1,..,u_m$ — концевые вершины дерева ветвления, которое было получено на управляющем процессе, то из соотношения (8) получим:

$$\varphi_m = \lim_{s\to\infty} \inf_{A\in\mathbf{A},|A|\ge s} \frac{1}{\left(m-1+\max\limits_{i=1,..,m} r(u_i)\right)m} = \lim_{s\to\infty} \inf_{A\in\mathbf{A},|A|\ge s} \frac{1}{m\cdot \max\limits_{i=1,..,m} r(u_i)}.$$

Пусть для решения задачи на управляющем процессе применялась фронтальное ветвление, т. е. для разбиения каждый раз выбиралась наиболее удаленная от корня дерева вершина. Рассмотрим случай $\alpha \geq \frac{1}{2}$. Тогда

$$\alpha^{\log_2 m+1} \leq \max_{i=1,..,m} r(u_i) \leq \alpha^{\log_2 m-1},$$

что равносильно

$$\alpha m^{\log_2 \alpha} \leq \max_{i=1,..,m} r(u_i) \leq \frac{m^{\log_2 \alpha}}{\alpha}.$$

Следовательно, справедливо неравенство

$$\alpha m^{-(\log_2 \alpha+1)} \leq \varphi_m \leq \frac{1}{\alpha} m^{-(\log_2 \alpha+1)}.$$

Асимптотическое поведение φ_m определяется множителем $m^{-(\log_2 \alpha+1)}$. Заметим, что масштабируемость имеет место только при $\alpha = \frac{1}{2}$. В этом случае, предельная эффективность не менее $\frac{1}{2}$. Если же $\alpha > \frac{1}{2}$, то

$$E^*\left(\xi, \left\{\Phi(n,k\right\}\right) = 0.$$

Таким образом, предельная эффективность рассмотренного алгоритма распределения нагрузки для всего множества задач о ранце также составляет 0. В то же время можно указать такое семейство задач, для которого эффективность будет не менее $\frac{1}{2}$.

Заключение

В работе введено понятие параллельной и коммуникационной сложности для метода ветвей и границ. Также рассмотрено традиционное понятие изоэффективности и масштабируемости параллельных систем. Показано, что традиционные определения не могут быть применены в случае, когда число операций алгоритма зависит от значений коэффициентов задачи. Для такого случая введено понятие предельной эффективности.

Исследована предельная эффективность фронтальной параллельной реализации метода ветвей и границ для решения задачи о ранце. Показано, что для всего множества задач о ранце предельная эффективность равняется 0. В то же время существуют семейства задач, для которых предельная эффективность отлична от нуля и, следовательно, имеет место масштабируемость. Полученные результаты показывают, что для методов типа ветвей и границ, исследование асимптотических характеристик производительности параллельной реализации являются достаточно сложной областью, требующей модификации традиционного аппарата.

В дальнейшем планируется исследовать параллельную сложность решения задачи о ранце другими вариантами параллельной реализации МВГ, в частности, с применением динамической балансировки нагрузки. Также планируется рассмотреть другие семейства задач о ранце и получить оценки предельной эффективности для таких семейств.

Литература

1. *Сигал И. Х., Иванова А. П.* Введение в прикладное дискретное программирование.М.: Физматлит, 2007.
2. *Grama A., Gupta A., Karypis G., Kumar V.* Introduction to Parallel Computing, Addison-Wesley, 2003.
3. *Emelyanov S. V., Afanasiev A. P., Grinberg Y. R., Krivtsov V. Y., Peltsverger B. V., Sukhoroslov O. V., Taylor R. G., Voloshinov V. V.* Distributed Computing and Its Applications. Editor: E. P. Velikhov, Bristol, ME, USA: Felicity Press, 2005.
4. *Kellerer H., Pfershy U., Pisinger D.* Knapsack Problems. Springer Verlag, 2004. 546 p.
5. *Martello S., Toth P.* Knapsack Problems. John Wiley & Sons Ltd., 1990.
6. *Alba E., Almeida F., Blesa M. et al.* Efficient Parallel LAN/WAN Algorithms for Optimization. The MALLBA Project. Parallel Computing 32(5–6). 2006. P. 415–440.
7. *Eckstein J., Philips C., Hart W.* PEBBL 1.0 User Guide. RUTCOR Research Report RRR 19–2006.
8. *Ralphs T.* Parallel branch and cut. In: Parallel Combinatorial Optimization. N. J.: JohnWiley & Sons, 2006. P. 53–103.
9. *Афанасьев А. П., Посыпкин М. А., Сигал И. Х.* Проект BNB-Grid: решение задач глобальной оптимизации в распределенной среде // Труды второй международной конференции «Системный анализ и информационные технологии» САИТ-2007. Т. 2. С. 177–181.
10. *Воеводин Вл. В.* НИВЦ МГУ: широким фронтом совместных дел // Численные методы, параллельные вычисления и информационные технологии: Сборник научных трудов под ред. Вл. В. Воеводина и Е. Е. Тыртышникова, М.: Издательство МГУ, 2008.
11. *Колпаков Р. М., Посыпкин М. А.* Верхняя и нижняя оценки трудоемкости метода ветвей и границ для задачи о ранце // Труды ИСА РАН. 2008. Т. 32. С. 137–158.
12. *Колпаков Р. М., Посыпкин М. А.* Асимптотическая оценка сложности метода ветвей и границ с ветвлением по дробной переменной для задачи о ранце // Дискретн. анализ и исслед. опер. 2008. 15:1.[I] С. 58–81