Институт динамики систем и теории управления СО РАН
Вычислительный центр им. А.А. Дородницына РАН
Институт систем энергетики им. Л.А. Мелентьева СО РАН
Институт проблем управления им. В.А. Трапезникова РАН
Иркутский государственный университет
Российская ассоциация математического программирования
Иркутская государственная сельскохозяйственная академия

XV Байкальская международная
школа-семинар

МЕТОДЫ ОПТИМИЗАЦИИ И ИХ ПРИЛОЖЕНИЯ

Байкал, пос. Листвянка
23 – 29 июня 2011 г.

Том 5
Прикладные задачи

Иркутск — 2011

Нижние оценки для задачи выбора порядка презентаций медиа объектов[6]

П.А. Кононова, Ю.А. Кочетов

Институт математики им. С.Л. Соболева СО РАН, Новосибирск

polinusik@gorodok.net, jkochet@math.nsc.ru

Аннотация. Рассматривается задача теории расписаний потокового типа для двух машин с буфером на второй машине. Она является обобщением известной задачи Джонсона и *NP*-трудна в сильном смысле. Вводится понятие ограниченной задачи, и показана эквивалентность исходной и ограниченной задачи. Исследуются нижние оценки оптимума. Установлено, что переход к ограниченной задаче приводит к существенному улучшению нижних оценок.

Введение

Рассматривается следующая задача, возникающая при создании электронно-цифровых библиотек и музеев [2]. Медиа объекты (файлы) загружаются из удаленной базы данных и затем воспроизводятся. Для каждого файла задано время загрузки и воспроизведения. При загрузке файл поступает в буфер транслирующего устройства и покидает его сразу после завершения воспроизведения. Известен размер буфера. Размеры файлов известны и пропорциональны времени загрузки. Воспроизведение файла не может начаться раньше окончания его загрузки. Если файл начал загрузку или воспроизведение, то этот процесс не прерывается. Требуется так задать порядок загрузки и обработки файлов, чтобы минимизировать время окончания воспроизведения последнего файла.

Известно [1], что эта задача является *NP*-трудной в сильном смысле даже для случая, когда время загрузки каждого файла меньше времени его воспроизведения. Там же выделены полиномиально разрешимые случаи, а для общего случая разработан метод локального поиска с чередующимися окрестностями. В данной работе приводятся нижние оценки оптимума и результаты численных экспериментов для их сравнения.

[6]Работа выполнена при финансовой поддержке РФФИ (грант 11-07-00474) и АВЦП Рособразования (проект 2.1.1/3235).

1 Математическая модель

Представим задачу в терминах теории расписаний. Рассмотрим две машины: A и B. Каждому файлу сопоставим работу. Загрузке файла будет соответствовать выполнение работы на машине A. Длительность работы i на машине A обозначим через a_i и положим равной времени загрузки соответствующего файла. Воспроизведению файла соответствует выполнение работы на машине B. Длительность работы i на машине B обозначим b_i и положим равной длительности воспроизведения файла. Так как воспроизведение файла не может начинаться до окончания загрузки файла, т.е. каждая работа сначала выполняется на машине A, а затем на машине B, то полученная задача является задачей потокового типа для двух машин. Отличием данной задачи от известной задачи Джонсона является наличие буфера. Во время выполнения работы на машине A она загружается в буфер и освобождает его сразу после окончания выполнения работы на машине B. Обозначим через Ω размер буфера в секундах, т.е. время, необходимое для заполнения всего буфера работами. Тогда работа i занимает в буфере a_i секунд. Эту задачу будем называть задачей Джонсона с буфером и обозначать ДБ.

Пусть последовательность $\sigma(i)$, $i = 1, \ldots, n$, задает порядок выполнения работ. В общем случае порядок на машине A может отличаться от порядка на машине B. Тем не менее, можно показать, что общий случай сводится к частному случаю, когда порядки на обеих машинах совпадают. Введем переменные $x_{ij} \in \{0, 1\}$, которые, так же как и σ, задают порядок выполнения работ: $x_{ij} = 1$, если работа i выполняется j-ой по порядку, т.е. $i = \sigma(j)$, и $x_{ij} = 0$ в противном случае. Тогда

$$\sum_{i=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1, \quad i,\, j = 1, \ldots, n. \tag{1}$$

Переменные x_{ij} позволяют вычислить длительность работы на машине A для j–й по порядку работы в σ. Обозначим эту величину $p^a_{\sigma(j)}$. Аналогичную величину для машины B обозначим $p^b_{\sigma(j)}$:

$$p^a_{\sigma(j)} = \sum_{i=1}^{n} a_i x_{ij}, \quad j = 1, \ldots, n, \tag{2}$$

$$p^b_{\sigma(j)} = \sum_{i=1}^{n} b_i x_{ij}, \quad j = 1. \ldots, n. \tag{3}$$

Введем дополнительные переменные $s^a_{\sigma(j)}$ и $s^b_{\sigma(j)}$, которые будут определять начало выполнения j-й по порядку работы в σ на машинах A и B соответственно:

$$s^a_{\sigma(1)} = 0, \tag{4}$$

$$s^a_{\sigma(j+1)} \geq s^a_{\sigma(j)} + p^a_{\sigma(j)}, \quad j = 1, \ldots, n-1, \tag{5}$$

$$s^b_{\sigma(j+1)} \geq s^b_{\sigma(j)} + p^b_{\sigma(j)}, \quad j = 1, \ldots, n-1, \tag{6}$$

$$s^b_{\sigma(j)} \geq s^a_{\sigma(j)} + p^a_{\sigma(j)}, \quad j = 1, \ldots, n. \tag{7}$$

Условие ограниченности буфера влечет [1]:

$$s^a_{\sigma(j+1)} \geq s^b_{\sigma(j)} + p^b_{\sigma(j)} + p^a_{\sigma(j)} - \Omega, \quad j = 1, \ldots, n-1. \tag{8}$$

Сформулированные условия задают множество допустимых расписаний. Задача состоит в нахождении допустимого расписания, доставляющего минимум следующей целевой функции:

$$F(\sigma) = s^b_{\sigma(n)} + p^b_{\sigma(n)}. \tag{9}$$

Задача (1)–(9) является задачей целочисленного линейного программирования. Такие задачи можно решать с помощью коммерческих пакетов. Одним из лучших является пакет CPLEX [4]. К сожалению численные эксперименты с этим пакетом уже для 20 работ требуют больших затрат машинного времени. Для случайно сгенерированных исходных данных на этой размерности требуется больше суток машинного времени. В связи с этим актуальным является разработка приближенных методов, в частности метаэвристик [1,3], а также нижних оценок оптимума.

1.1 Ограниченная задача

Рассмотрим частный случай задачи ДБ, в котором для каждой работы i выполнено неравенство $a_i + b_i \leq \Omega$. Будем называть такую задачу ограниченной задачей Джонсона с буфером и обозначать ОДБ.

Теорема 1 ([1]). *Задачи ДБ и ОДБ эквивалентны.*

Доказательство теоремы 1 основано на следующем сведении задачи ДБ к ОДБ. Пусть I произвольный пример задачи ДБ. Построим пример $\overline{I}$ задачи ОДБ с тем же числом работ и длительностями, равными $\overline{a}_i = a_i$ и $\overline{b}_i = \min(b_i, \Omega - a_i)$. Положим $\delta_i = b_i - \overline{b}_i$. Размер буфера оставим прежним. Тогда для целевой функции $\overline{F}(\sigma)$ задачи ОДБ выполнено следующее равенство: $F(\sigma) = \overline{F}(\sigma) + \sum_{i=1}^{n} \delta_i$ для любой перестановки σ.

2 Нижние оценки

Рассмотрим нижние оценки оптимума задачи (1)–(9) и покажем связь между ними. Заменим условие целочисленности переменных x_{ij} на условие принадлежности их отрезку $[0, 1]$. Получим задачу линейного программирования. Ее

оптимальное значение обозначим через LB_{lp}. Наряду с этой нижней оценкой рассмотрим оценку линейного программирования для задачи ОДБ. Обозначим ее $\overline{LB}_{lp}$.

Теорема 2. $LB_{lp} \leq \overline{LB}_{lp} + \sum_{i=1}^{n} \delta_i$.

Задача без буферного ограничения совпадает с классической задачей Джонсона. Известно, что для задачи Джонсона существует оптимальное расписание, в котором порядок выполнения работ на машинах A и B совпадает и машина A работает без простоев. Величина оптимального решения будет нижней оценкой для исходной задачи с буфером. Обозначим LB_J нижнюю оценку, полученную алгоритмом Джонсона для задачи ДБ, и $\overline{LB}_J$ — нижнюю оценку для задачи ОДБ.

Теорема 3. $LB_J \leq \overline{LB}_J + \sum_{i=1}^{n} \delta_i$.

Оценки $\overline{LB}_J$ и $\overline{LB}_{lp}$ полиномиально вычислимы, однако для первой оценки имеется строго полиномиальный алгоритм Джонсона, а для второй оценки требуется решение задачи линейного программирования. В этом смысле первая оценка предпочтительнее. Тем не менее, нельзя утверждать, что одна из этих оценок лучше другой. В ходе численных экспериментов было установлено, что они несравнимы между собой, но разница между ними относительно мала.

3 Численные эксперименты

Теоремы 2 и 3 показывают, что переход к ограниченной задаче приводит к росту нижних оценок. Чтобы понять, насколько сильно это влияние, проведен следующий эксперимент. Для трех классов задач $n = 20, 30, 100$ случайным образом с равномерным распределением выбирались величины a_i и b_i из интервала от 20 до 40. В каждом классе генерировалось 50 примеров задачи ДБ. Размер буфера в каждом классе менялся от 40 до 80. Исследовалась зависимость величины

$$\varepsilon = \frac{\overline{LB}_{lp} + \sum_{i=1}^{n} \delta_i - LB_{lp}}{LB_{lp}} 100\%$$

от размера буфера в каждом классе. Результаты расчетов представлены на рис. 1. При $\Omega > 80$ размер буфера не играет роли и задача становится полиномиально разрешимой. При $\Omega < 40$ возникают работы, размером больше буфера. Наиболее интересным является случай $\Omega = 60$. При таком буфере различие в оценках оказывается максимальным и достигает 9%. Таким образом, переход

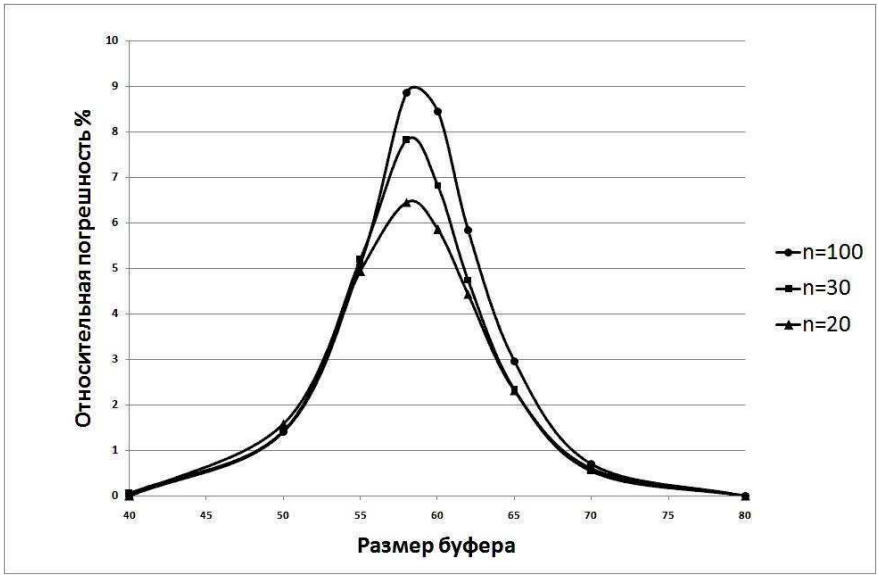

Рис. 1: Зависимость ε от размера буфера

к ограниченной задаче действительно приводит к уточнению нижних оценок оптимума.

Следующий эксперимент посвящен сравнению нижних оценок с верхними оценками, получаемыми алгоритмом локального поиска с чередующимися окрестностями (VNS). В табл. 1 представлено число несовпадений верхних и нижних оценок для 50 тестовых примеров в каждом классе при изменении размера буфера от 40 до 80. При $n = 20$ и размере буфера более 62 оценки $\overline{LB}_J$ и $\overline{LB}_{lp}$ не совпадают. На этом классе всегда получалось, что $\overline{LB}_J \geq \overline{LB}_{lp}$. На других исходных данных наблюдалось и обратное соотношение. Разница между оценками не превышала 0.5%.

Таблица 1: Различие между верхними и нижними оценками

n		40	50	55	58	60	62	65	70	80
20	$\overline{LB}_{lp} \neq \overline{LB}_J$	0	0	0	0	0	3	4	3	4
	$VNS \neq \overline{LB}_J$	0	0	0	2	5	7	2	0	0
	$VNS \neq \overline{LB}_{lp}$	0	0	0	2	5	8	6	3	4
30	$\overline{LB}_{lp} \neq \overline{LB}_J$	0	0	0	0	0	0	0	0	0
	$VNS \neq \overline{LB}_{lp}$	0	0	0	0	2	2	1	0	0
100	$\overline{LB}_{lp} \neq \overline{LB}_J$	0	0	0	0	0	0	0	0	0
	$VNS \neq \overline{LB}_{lp}$	0	0	0	0	0	0	0	0	0

Легко заметить, что с ростом размерности число несовпадений падает, а при $n = 100$ исчезает вовсе. Получается удивительный эффект: с ростом размерности задачи она становится проще и при больших n легко решается точно.

Алгоритм VNS быстро находит решение, значение которого совпадает с нижней оценкой. Такой эффект связан со способом генерации исходных данных. При большом разнообразии длительностей работ их легко компоновать под размер буфера. Изменив способ генерации исходных данных, можно получить трудные примеры, которые даже при отсутсвии разрыва целочисленности будут представлять "головоломку" [3].

Список литературы

1. *Kononov A.V., Kononova P.A., Hong J.-S.*, Two-stage multimedia scheduling problem with an active prefetch model // Preprints of the 13th IFAC Symp. on Information Control Problems in Manufacturing. Moscow (Russia), June 3–5, 2009. Moscow, 2009. P. 1997–2002.

2. *Lin F.-C., Hong J.-S., Lin B.M.T.* A two-machine flow shop problem with processing time-dependent buffer constraints — An application in multimedia problem // Computers and Operations Research. 2009. Vol. 36 (4). P. 1158–1175.

3. *Кононова П.А.* Алгоритм локального поиска для задачи выбора порядка презентаций медиа объектов // Тр. ИВМиМГ. Информатика. Новосибирск: Изд-во ИВМиМГ СО РАН, 2009. Т. 9. С. 177–182.

4. *CPLEX.* URL: `http://www-142.ibm.com/software/products/ru/ru/ilogcple/`.