$\mu_k = 1/n\sum_{j=1}^{n}\left(\vartheta_j - \widehat{m}_n\right)^k$, $k = 2,3,4$ – несмещенные и состоятельные оценки математического ожидания, дисперсии, асимметрии и эксцесса соответственно [2].

В работе по наблюдениям за дисперсионным гамма-процессом строятся оценки математического ожидания, дисперсии, асимметрии, эксцесса и оценки параметров σ, ν, θ; используя теорему 1 строятся оценки параметров a, b_1, b_2.

Литература

1. *Cont R., Tankov P.* Financial modeling with jump processes. Chapman and Hall CRC Press. 2003. P. 192.
2. *Крамер Г.* Математические методы статистики. М.: Мир. 1975. С. 375–386.
3. *Madan D. B.*, Carr P.P., E.C. Chang E.C. The variance gamma process and option pricing // European Finance Preview 2. 1998. P. 79–105.
4. *Schoutens W.* Levy processes in finance. Williams. 2003. P. 108–109.

ЗАДАЧИ МИНИМИЗАЦИИ ВРЕМЕНИ ЗАВЕРШЕНИЯ ПРОЕКТА НА МНОГОПРОЦЕССОРНОЙ СИСТЕМЕ С РАЗНЫМИ СКОРОСТЯМИ ПРОЦЕССОРОВ

Ю. С. Мазаник, В. М. Котов

ВВЕДЕНИЕ

В классической задаче минимизации времени завершения проекта на многопроцессорной системе рассматривается система из $m > 1$ машин $M_1,..,M_m$ с соответствующими им скоростями $s_1,...,s_m$ и N работ, с неотрицательными временами обработки $a_1,..,a_n > 0$. Требуется распределить все работы таким образом, чтобы время их обработки было минимальным.

В on-line версии данной задачи, все работы поступают последовательно, и каждую работу необходимо назначить на выполнение одной из машин сразу после поступления и, не имея никакой информации о последующих работах.

Поскольку последовательность работ заранее не известна, данная задаче не имеет оптимального решения. Существует несколько способов оценки эффективности построенного решения. Стандартной является оценка качества предложенного алгоритма, относительно offline задачи.

Такой коэффициент принято называть асимптотическим коэффициентом эффективности, а метод оценивания сравнительным анализом.

Обозначим через $opt(n)$ время работы оптимального алгоритма для решения offline задачи, а через $A(n)$ – время работы предлагаемого алгоритма решения online задачи, тогда формула для вычисления асимптотического коэффициента эффективности имеет вид $R(A) = \lim\limits_{n\to\infty} \frac{A(n)}{opt(n)}$. Гарантированная оценка времени работы вычисляется по формуле $G(A) = \max\limits_{\forall n} \frac{A(n)}{opt(n)}$.

Впервые данная проблема была исследована Грэмом (Graham) [1] , который показал, что LS (list scheduling),алгоритм имеет асимптотическую оценку не менее $3 - \frac{4}{m+1}$.

В работе [2] была рассмотрена классическая задача, с дополнительным условием на время выпуска работ с машин.

Получена зависящая от количества машин оценка эффективности $\frac{1 + z + \sqrt{4m - 4 + (1-z)^2}}{2}$, где $z = \frac{r_{\max}}{\max(r_{\max}, p_{\max})}$. ($r_{\max}$ - максимальное время выпуска, $p_{\max}$ – самая трудная работа).

В работе [3] был предложен алгоритм с оценкой 2.45 для рассматриваемой ниже задачи.

ПОСТАНОВКА ЗАДАЧИ, ОБОЗНАЧЕНИЯ И БАЗОВЫЕ ФОРМУЛЫ

В данной работе рассмотрен частный случай классической on-line задачи, когда $s_1 = ... = s_{m-1} = 1,\ 1 < s_m = s < 2$.

Введем некоторые стандартные обозначения:

LB_n– нижняя оценка времени работы оптимального алгоритма на шаге n,

m – количество машин,

a_i –ремя выполнения i-ой работы,

$a_{\max}$ – время выполнения максимальной из поступивших данном шаге работ,

$a_{\max 2}$ – время выполнения второй по длительности среди поступивших работ,

$L(X_i, j)$ – суммарная загрузка i-ой машины на шаге j (сокращенно $L(X_i)$),

L_{jcc} – средняя загрузка машин на шаге j,

LB_j – оценку можно вычислить по формуле:

$$LB_j = \left\{ \sum_{i=1}^{j} \frac{a_i}{m+s-1}, \frac{a_{\max}}{s}, a_{\max 2} \right\}.$$

Назовем работу обычной, если ее вес не превосходит $1.2LB_j$, в противном случае назовем работу большой.

АЛГОРИТМ НАЗНАЧЕНИЯ

Разобьем машины на группы по 6 в каждой. Быструю машину будем рассматривать отдельно. В случае, если количество машин не представимо в виде $6p$ или $6p+1$ оставшиеся машины будут образовывать неполную группу (без машины типа F)

$$\begin{pmatrix} A_1 & B_1 & C_1 & D_1 & E_1 & F_1 \\ \\ A_{\frac{M}{k}} & B_{\frac{M}{k}} & C_{\frac{M}{k}} & D_{\frac{M}{k}} & E_{\frac{M}{k}} & F_{\frac{M}{k}} \end{pmatrix}.$$

В каждой из групп на первые 5 машин будем назначать обычные работы пока загрузка машин не превышает $2.4LB_j$

$$L(X_i, j) + a_j < 2.4LB_j, \forall X \in \{A, B, C, D, E\}.$$

Быструю машину будем загружать большими работами пока ее загрузка не превышает $2.4LB_j$ – с нее будем начинать итерацию загрузки машин типа F. Как только загрузка быстрой машины на с текущей работой превысит $2.4LB_j$, будем загружать машину F, на каждом шаге выбирая менше всего загруженную машину типа F. Когда каждая машина типа F будет загружена хотя бы по одному разу, будем говорить о том, что итерация завершена.

Теорема об асимптотической оценке. При указанной процедуре загрузки алгоритм дает гарантированную оценку 2.4.

Лемма 1. Любую обычную работу можно выполнить на машинах из групп ($A - E$), чтобы время завершения не превышало $2.4LB_j$.

Доказательство. Предположим, что на текущем шаге какая-то обычная работа не может выполняться на машинах типа (A-E) в пределах времени $2.4LB_j$. Тогда для каждых 5-и машин имеем:

$L_j(X) + a_j > 2.4LB_j$, откуда следует, что $L_j > 1.2LB_j, \forall X \in \{A,B,C,D,E\}$. Поэтому при любой загрузке машины F имеем: $L_j(A) + L_j(B) + L_j(C) + L_j(D) + L_j(E) > 1.2LB_j \cdot 5 = 6LB_j$. Тогда средняя загрузка больше LB_j, чего быть не может т.к. LB_j больше средней загрузки по построению. (Оставшиеся машины загружены больше $1.2LB_j$ каждая т.е. не влияют на оценку алгоритма).

Лемма 2. При H не меньше 7, где H это количество групп машин, загружая машины типа F в порядке увеличения их нагрузки, после завершения каждой из итераций загрузки больших машин, загрузка минимально загруженной машины типа F всегда будет меньше $0.4LB_j$.

Доказательство. Не нарушая общности, предположим, что машины упорядочены в порядке возрастания загрузки (т.е. первой идет наименее загруженная машина). В процессе нагрузки не будем изменять порядок следования машин до конца текущей итерации. Для начала загрузки предположение верно, т.к. загрузка машины $F_1 = 0$. Рассмотрим поток больших работ, загружаемых на машины типа F. Обозначим данные работы как $p_1,...,p_H$, где H это количество наборов. Соответствующие им LB_j обозначим $LB_1,...,LB_H$.

Заметим, что если мы будем класть большую работу на машину загрузка, которой меньше $0.4LB_j$, то полученная загрузка не превысит $sLB_j + 0.4LB_j < (2 + 0.4)LB_j = 2.4LB_j$, а в случае быстрой машины ее загрузка при тех же начальных условиях будет $LB_j + 0.4LB_j < (1 + 0.4)LB_j = 1.4LB_j$. Т.е. в начале любой итерации мы будем 2 раза класть большие работы на машину со скоростью s.

Рассмотрим несколько первых шагов после загрузки быстрой машины: $p_3 > 1.2LB_3$. Предположим, что $p_3 \neq p_{\max}$. В этом случае получаем $p_3 \le p_{\max 2} \le LB_3$, что неверно, откуда сделует, что $p_3 = p_{\max}$ (аналогично можно показать, что каждая последующая работа является максимальной для своего шага). Откуда $p_3 > 1.2LB_3 > 1.2LB_2$ и $p_4 > 1.2LB_4 > 1.2p_3 > 1.2^2 LB_3$.

Таким образом, получим $p_H > 1.2^{H-1} LB_1$. Для того, чтобы существовало нужное нам H, необходимо, чтобы выполнялось соотношение:

$$p_1 + 0.4LB_1 < 0.4LB_{H+1}$$

$$0.4LB_1 \ge 0.4p_H > 0.4 \cdot 1.2^{H-1} LB_1$$

$$p_1 + 0.4LB_1 \le LB_1 + 0.4LB_1, \{p_1 = p_{\max} > sLB_1\}$$

Таким образом, H можно найти из формулы $1.4LB_1 < 0.4 \cdot 1.2^{H-1} LB_1$.

$$H \ge \left\lceil \frac{\ln\left(\frac{7}{2}\right)}{\ln(1.2)} \right\rceil + 1,$$

т.е. если $H \ge 7$ то машин больше чем 43 и предложенный алгоритм дает оценку 2.4.

ЗАКЛЮЧЕНИЕ

В работе построен алгоритм, для задачи минимизации времени завершения проекта на многопроцессорной системе со скоростями процессоров 1 и s, и доказано, что представленный алгоритм дает асимптотическую оценку 2.4 для количества машин, большего, чем 43.

Литература

1. *R. L. Graham*. Bounds on multiprocessing timing anomalies. // SIAM J. Appl. Math. 17 1969. P. 263–269.
2. *Christos Loulamas, George J. Kyparisis*. «Makespan minimization on uniform parallel machines with release times». // European Journal of Operational Research. 2004.
3. *T. C. E. Cheng , C. T. Ng , Vladimir Kotov*. «A new algorithm for online uniform-machine scheduling to minimize the makespan». // Information Processing Letters. V 99. 2006.

ВЕРОЯТНОСТНО-СТАТИСТИЧЕСКИЙ АНАЛИЗ ЦЕПЕЙ МАРКОВА ПЕРЕМЕННОГО ПОРЯДКА

М. В. Мальцев

1. ВВЕДЕНИЕ

Цепи Маркова [1] используются для решения многих задач статистического анализа данных в генетике [2], экономике и других областях научной и практической деятельности. Общей моделью в таких исследованиях является цепь Маркова s-го порядка, $s \ge 1$. Однако число параметров данной модели возрастает экспоненциально при увеличении порядка s. Для статистического оценивания параметров требуется иметь реализацию последовательности далеко не всегда доступной на практике длительности. Поэтому актуальной является задача построения малопараметрических моделей цепи Маркова высокого порядка.