УДК 519.178

Оптимальный алгоритм перемещения объектов в квадратной решетке

В. В. Морозенко, А. В. Москалев
Пермский Государственный Университет, 614990, Пермь, ул. Букирева, 15

Для произвольного натурального n рассматривается задача оптимального перемещения объектов в графе, который является декартовым произведением n-вершинной цепи на себя. Предполагается, что из каждой вершины в каждую должен быть перемещен ровно один объект, причем за один такт по любому ребру может быть перемещено не более одного объекта. Описывается алгоритм, который осуществляет перемещение всех объектов за минимальное число тактов, и доказывается его оптимальность.

Постановка задачи

Проблема оптимального синхронного перемещения объектов внутри некоторой структуры часто возникает на практике. Например, она возникает при поиске распределения транспортных потоков внутри транспортной сети, обеспечивающего максимально возможную скорость перемещения. Другим примером может служить оптимизация по времени процесса перераспределения заданий между узлами имитационной модели. Эта задача возникает из-за необходимости в кратчайшие сроки переместить часть заданий с более загруженных на менее загруженные узлы вычислительной сети [1].

Во всех подобных задачах под структурой, внутри которой происходит перемещение объектов, понимается граф. При этом предполагается, что перемещение объекта из одной вершины в другую происходит за один такт вдоль ребра, соединяющего эти вершины, и за один такт вдоль любого ребра может быть перемещено не более одного объекта. Если число перемещаемых объектов велико, то между ними возникает конфликт из-за ребер, по которым перемещаются объекты. Для разрешения конфликтов можно назначить каждому объекту некоторый приоритет и каждый раз, когда будет возникать очередь из нескольких объектов, претендующих на перемещение по одному и тому же ребру, предпочтение отдавать объекту с наивысшим приоритетом.

Очевидно, существует множество правил распределения приоритетов между перемещаемыми объектами. Однако, как только правило будет зафиксировано, соответствующий алгоритм перемещения объектов становится полностью детерминированным. С практической же точки зрения наибольший интерес вызывает правило, обеспечивающее перемещение всех объектов в кратчайшие сроки. В [2] такое правило назначения приоритетов и соответствующий ему оптимальный алгоритм перемещения объектов найдены для произвольного дерева при условии, что из каждой его вершины в любую другую должно быть перемещено ровно по одному объекту. Поиск правила осуществлен с помощью алгоритма, имеющего полиномиальную сложность относительно числа вершин в дереве.

В настоящей работе в качестве графа выбрана квадратная решетка $n \times n$ с числом вершин, равным n^2. Её можно рассматривать как декартово произведение $G \times G$, где граф G – это n-вершинная цепь. Вершины решетки обозначим через $v_{i,j}$, где $i,j = \overline{0, n-1}$, так чтобы вершины $v_{i,j}$ и $v_{r,t}$ были смежными тогда и только тогда, когда $|i-r|+|j-t|=1$ (см. рис. 1).

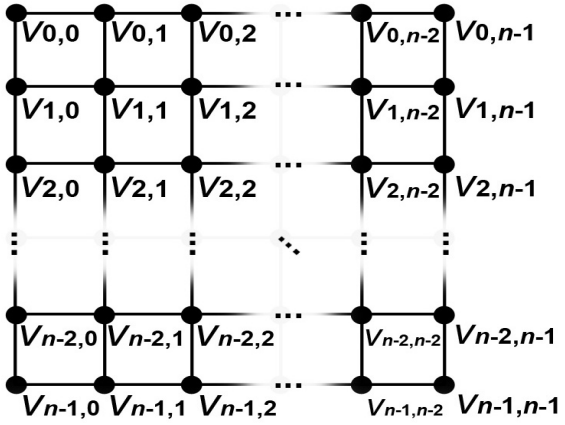


Рис. 1. *Квадратная решетка n×n*

Будем предполагать, что из каждой вершины решетки в любую другую вершину должен быть перемещен ровно один объект. При этом считается, что за один такт работы алгоритма по каждому ребру может быть перемещено не более одного объекта, и допускается одновременное перемещение объектов по всем ребрам. Задача состоит в том, чтобы найти такую последовательность перемещений объектов, при которой время (т.е. число тактов), требуемое для перемещения всех объектов, было минимальным.

Нижние оценки сложности алгоритма перемещения

Поскольку решетка содержит $2n(n-1)$ ребер и за один такт по любому ребру допускается перемещение максимум одного объекта, то всего за один такт по всем ребрам может быть перемещено не более, чем $2n(n-1)$ объектов. С другой стороны, всего в решетке должно быть перемещено ровно $n^2(n^2-1)$ объектов. Следовательно, получаем нижнюю оценку на минимальное число тактов $t(n)$, требуемое для перемещения всех объектов

$$t(n) \geq n(n+1)/2 .$$

Однако полученную оценку можно улучшить. Для этого цепь, образованную вершинами $v_{i,0}$, $v_{i,1},\ldots,v_{i,n-1}$, обозначим через P_i и назовем *i-ой горизонталью*, а цепь, образованную вершинами $v_{0,j}$, $v_{1,j},\ldots,v_{n-1,j}$, обозначим через Q_j и назовем *j-ой вертикалью*, где $i,j = \overline{0,n-1}$. Пусть $m = \lfloor n/2 \rfloor$. Тогда между вершинами, входящими в горизонтали $P_0,P_1,\ldots,P_{m-1}$, с одной стороны, и остальными вершинами решетки, с другой стороны, должно быть перемещено $2mn^2(n-m)$ объектов. Очевидно, каждый из этих объектов должен переместиться хотя бы по одному «вертикальному» ребру, соединяющему вершину из горизонтали P_{m-1} со смежной вершиной из горизонтали P_m, т.е. по какому-либо ребру вида $[v_{m-1,j}, v_{m,j}]$, где $j = \overline{0,n-1}$. А поскольку число таких ребер в решетке равно n, то число тактов, требуемое для перемещения всех указанных объектов через эти ребра не может быть меньше, чем $2mn(n-m)$, где $m = \lfloor n/2 \rfloor$. Таким образом мы доказали следующую теорему.

Теорема. *Сложность t(n) оптимального алгоритма перемещения всех объектов в квадратной решетке n×n удовлетворяет неравенству*

$$t(n) \geq \begin{cases} n^3/2, & \text{если } n \text{ четно}, \\ n(n^2-1)/2, & \text{иначе}. \end{cases} \qquad (1)$$

Полученная оценка является неулучшаемой, поскольку, как будет показано далее, оптимальный алгоритм, выполняющий перемещение всех объектов в кратчайшие сроки, требует число тактов, указанное в правой части неравенства (1).

Оптимальный алгоритм перемещения

Опишем алгоритм, решающий задачу перемещения объектов в решетке $n \times n$, и докажем, что он выполняет все перемещения за минимально возможное число тактов, т.е. является оптимальным. В работе этого алгоритма будем использовать оптимальный алгоритм перемещения объектов в n-вершинной цепи, описанный в [2]. Там же доказано, что его сложность $c(n)$ удовлетворяет равенству

$$c(n) = \begin{cases} n^2/2, & \text{если } n \text{ четно}, \\ (n^2-1)/2, & \text{иначе}. \end{cases} \qquad (2)$$

Через $[i, j; r, t]$, где $i, j, r, t = \overline{0,n-1}$, причем $|i-r|+|j-t| \neq 0$, обозначим объект, который в результате работы алгоритма должен быть перемещен из начальной вершины $v_{i,j}$ в финальную для этого объекта вершину $v_{r,t}$. Для каждой фиксированной пары i и j количество таких объектов равно n^2-1, и для каждого из них найдется единственная пара k и l, где $k,l = \overline{0,n-1}$, причем $k+l \neq 0$, для которой сам объект будет иметь вид

$$[i, j; i+k \ (mod\ n), j+l \ (mod\ n)].$$

Оптимальный алгоритм перемещения объектов в решетке $n \times n$ будет состоять из n этапов. Для каждой четверки $i, j, r, t = \overline{0,n-1}$

, где $|i-r|+|j-t|\neq 0$, объект $[i, j; r, j]$ перемещается из $v_{i,j}$ в $v_{r,j}$ вдоль j-ой вертикали на первом этапе алгоритма, а объект $[i, j; i, t]$ перемещается из $v_{i,j}$ в $v_{i,t}$ вдоль i-ой горизонтали на n-ом этапе. Что касается объектов $[i, j; r, t]$, где $i \neq r, j \neq t$, то каждый из них перемещается в течение двух последовательных этапов: сначала вдоль i-ой горизонтали из $v_{i,j}$ в $v_{i,t}$, а затем вдоль t-ой вертикали из $v_{i,t}$ в $v_{r,t}$. Перемещение объектов и разрешение возникающих при этом конфликтов осуществляется с помощью оптимального алгоритма перемещений в n-вершинной цепи из [2].

Опишем оптимальный алгоритм перемещения в квадратной решетке $n \times n$ с помощью математической индукции по параметру $k = \overline{1, n}$ – номеру этапа.

Базис индукции: $k = 1$. Очевидно, что исходная задача для решетки $n \times n$ содержит в себе (помимо прочего) n подзадач перемещения объектов $[i, j; r, j]$, при $i, r = \overline{0, n-1}$, $i \neq r$, внутри j-ой вертикали, где $j = \overline{0, n-1}$. На первом этапе алгоритма n указанных подзадач решаются одновременно и независимо друг от друга внутри каждой вертикали с помощью оптимального алгоритма для n-вершинной цепи из [2]. Независимость в данном случае означает, что перемещение объектов и разрешение конфликтов происходит только внутри каждой вертикали. По окончании этого этапа из каждой вершины $v_{i,j}$ в любую другую вершину $v_{r,j}$ той же вертикали Q_j будет перемещено ровно по одному объекту. Таким образом в свои финальные вершины будут перемещены все объекты вида $[i, j; r, j]$, где $i, j, r = \overline{0, n-1}$, $i \neq r$.

Кроме того, на первом этапе из каждой вершины $v_{i,j}$ вдоль i-ой горизонтали перемещается $n - 1$ объектов

$$[i, j; i + 1 \ (mod\ n), j + 1 \ (mod\ n)],$$

$$[i, j; i + 1 \ (mod\ n), j + 2 \ (mod\ n)],$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$[i, j; i + 1 \ (mod\ n), j + n - 1 \ (mod\ n)],$$

причем для каждого $l = \overline{1, n-1}$ в вершину $v_{i, j + l \ (mod\ n)}$ перемещается только объект

$$[i, j; i + 1 \ (mod\ n), j + l \ (mod\ n)].$$

Перемещение всех указанных объектов внутри каждой горизонтали выполняется согласно алгоритму для n-вершинной цепи из [2]. При этом каждый из перечисленных объектов перемещается на первом этапе лишь в соответствующую промежуточную вершину $v_{i, j + l \ (mod\ n)}$, а в свою финальную вершину $v_{i + 1 \ (mod\ n), j + l \ (mod\ n)}$ он переместится на втором этапе, двигаясь вдоль вертикали $Q_{j + l \ (mod\ n)}$. Таким образом в силу (2) на первом этапе будет выполнено $n^2/2$ тактов, если n четно, либо $(n^2 - 1)/2$ в противном случае. В итоге по окончании первого этапа в каждую вершину переместится $n - 1$ объектов по вертикали, для которых эта вершина является финальной, и столько же объектов по горизонтали, для которых эта вершина является лишь промежуточной.

Индуктивный переход: предположим, что на k-ом этапе, где $k < n$, из каждой вершины $v_{i,j}$ вдоль i-ой горизонтали согласно алгоритму из [2] были перемещены $n - 1$ объектов

$$[i, j; i + k \ (mod\ n), j + 1 \ (mod\ n)],$$

$$[i, j; i + k \ (mod\ n), j + 2 \ (mod\ n)],$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

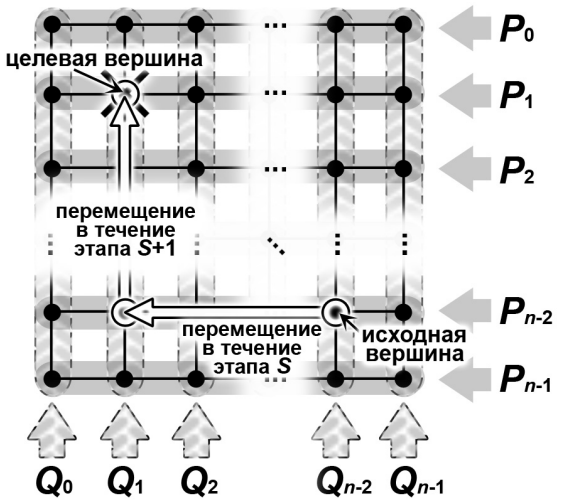
$$[i, j; i + k \ (mod\ n), j + n - 1 \ (mod\ n)]$$

причем для каждого $l = \overline{1, n-1}$ в промежуточную вершину $v_{i, j + l \ (mod\ n)}$ был перемещен только объект

$$[i, j; i + k \ (mod\ n), j + l \ (mod\ n)].$$

Тогда на $(k + 1)$-ом этапе переместим далее каждый такой объект из $v_{i, j + l \ (mod\ n)}$ в соответствующую ему финальную вершину $v_{i + k \ (mod\ n), j + l \ (mod\ n)}$ вдоль вертикали $Q_{j + l \ (mod\ n)}$ (см. рис. 2).

Рис. 2. *Перемещение объектов в два этапа*



Кроме того, с помощью алгоритма из [2] на $(k + 1)$-ом этапе из каждой вершины $v_{i,j}$

вдоль i-ой горизонтали выполним перемещение $n-1$ объектов

$$[i, j;\ i + k + 1\ (mod\ n), j + 1\ (mod\ n)],$$
$$[i, j;\ i + k + 1\ (mod\ n), j + 2\ (mod\ n)],$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$[i, j;\ i + k + 1\ (mod\ n), j + n - 1\ (mod\ n)],$$

причем для каждого $l = \overline{1, n-1}$ в промежуточную вершину $v_{i, j + l\ (mod\ n)}$ переместим только объект

$$[i, j;\ i + k + 1\ (mod\ n), j + l\ (mod\ n)].$$

Если $k + 1 < n$, то каждый такой объект будет перемещен в свою финальную вершину вдоль соответствующей вертикали на $(k + 2)$-ом этапе. Если же $k + 1 = n$, то $(k + 1)$-ый этап является последним, и после его окончания все указанные объекты окажутся в своих финальных вершинах. Индуктивное описание алгоритма завершено.

Из приведенного описания видно, что в своих финальных вершинах по окончании очередного этапа алгоритма оказываются все объекты, перемещаемые на этом этапе алгоритма вдоль любой вертикали, а также все объекты, перемещаемые по горизонтали только на последнем этапе алгоритма. Все остальные объекты, т.е. перемещаемые вдоль любой горизонтали на всех этапах, кроме последнего, по окончании этих этапов оказываются лишь в промежуточных вершинах.

Докажем корректность данного алгоритма. Для этого достаточно показать, что в результате его работы для любой фиксированной четверки i, j, r, t, где $i, j, r, t = \overline{0, n-1}$, $|i - r| + |j - t| \neq 0$, из начальной вершины $v_{i,j}$ в финальную вершину $v_{r,t}$ будет перемещен ровно один объект $[i, j; r, t]$. В самом деле, если $i \neq r$, $t = j$, то согласно базису индукции объект $[i, j; r, j]$ будет перемещен из $v_{i,j}$ в $v_{r,j}$ на первом этапе алгоритма. Если $i = r$, $j \neq t$, то объект $[i, j; i, t]$ будет перемещен из $v_{i,j}$ в $v_{i,t}$ на n-ом этапе алгоритма. Если же $i \neq r$, $j \neq t$, то объект $[i, j; r, t]$ перемещается из начальной вершины $v_{i,j}$ в свою финальную вершину $v_{r,t}$ в течение двух последовательных этапов: сначала вдоль i-ой горизонтали из $v_{i,j}$ в $v_{i,t}$, а затем вдоль t-ой вертикали из $v_{i,t}$ в $v_{r,t}$. Более точно, пусть $r = i + k\ (mod\ n)$, $t = j + l\ (mod\ n)$, где $k, l = \overline{1, n-1}$. Тогда объект $[i, j; r, t]$ будет перемещен на k-ом этапе из $v_{i,j}$ в $v_{i,t}$, а затем на $(k + 1)$-ом этапе – из $v_{i,t}$ в $v_{r,t}$. Все конфликты, возникающие при перемещении объектов внутри каждой вертикали и горизонтали, разрешаются независимо в соответствии с алгоритмом для перемещений в n-вершинной цепи из [2]. Таким образом описанный алгоритм корректно решает задачу перемещения объектов в квадратной решетке $n \times n$.

Вычислим сложность алгоритма. На каждом из n его этапов независимо и одновременно решаются n подзадач перемещения объектов для каждой горизонтали и столько же подзадач для каждой вертикали. Поскольку эти подзадачи решаются одновременно и независимо друг от друга с помощью оптимального алгоритма для перемещений в n-вершинной цепи из [2], то согласно (2) каждый этап состоит из $n^2/2$ тактов, если n четно, либо из $(n^2 - 1)/2$ тактов в противном случае. Следовательно, сложность $s(n)$ описанного алгоритма удовлетворяет равенству

$$s(n) = \begin{cases} n^3/2, & \textit{если } n \textit{ четно}, \\ n(n^2 - 1)/2, & \textit{иначе}. \end{cases}$$

Согласно оценке (1) это означает, что данный алгоритм является оптимальным по времени, т.е. выполняет все требуемые перемещения объектов за минимально возможное число тактов.

Заключение

В данной работе найден алгоритм, который для произвольного натурального n решает задачу перемещения объектов в квадратной решетке $n \times n$ за минимальное число тактов, и доказана его оптимальность. Ранее в [2,3] аналогичные результаты были получены для n-вершинной цепи и n-мерного единичного куба. Обращение к конкретным топологиям объясняется тем, что задача нахождения алгоритма, обеспечивающего оптимальное (т.е. в кратчайшие сроки) перемещение объектов в произвольном графе, является, по-видимому, *NP*-трудной, поскольку она близка к *NP*-полной задаче составления оптимального расписания для выполнения несколькими однотипными параллельно работающими исполнителями системы зависимых заданий с частичным порядком – отношением предшествования [4]. Поэтому в описываемой проблематике прослеживаются два направления исследований.

Первое направление связано с попытками решать задачу перемещения объектов в общем виде с помощью приближенных или ге-

нетических алгоритмов [5,6]. Другое направление, в рамках которого выполнена настоящая работа, состоит в обнаружении широких классов топологий, для которых удается с помощью полиномиальных по сложности процедур получить оптимальные алгоритмы перемещения всех объектов за минимальное число тактов.

Очевидно, что минимальное количество тактов для графа любой структуры всегда выражается полиномом от числа его вершин. Однако остается открытым вопрос, для всех ли графов поиск правил перемещения объектов и разрешения конфликтов, приводящих к минимальному числу тактов, может быть осуществлен с помощью полиномиального алгоритма. Как следует из [2,3] и данной работы, такие алгоритмы существуют для классов n-вершинных цепей, n-мерных единичных кубов и квадратных решеток $n \times n$.

Список литературы

1. *Zheng G*. Achieving High Performance on Extremely Large Parallel Machines: Performance Prediction and Load Balancing; in Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.
2. *Морозенко В.В.* Оптимальный алгоритм перемещения объектов в имитационной модели с древовидной топологией // Математика программных систем: Межвуз. сб. науч. тр./ Перм. ун-т. Пермь, 2007, с.16–27.
3. *Москалев А.В.* Рекурсивный оптимальный алгоритм перемещения объектов в n-мерном единичном кубе // Сборник статей «Современная математика и математическое образование, проблемы истории и философии математики» / Международная конференция, Россия, Тамбов, 22–25 апреля, 2008, с. 52–55.
4. *Коффман Э.Г.* Теория расписаний и вычислительные машины. М.: Наука, 1984.
5. *Mikov A.I.* Simulation and Design of Hardware and Software with Triad// Proc.2nd Intl.Conf. on Electronic Hardware Description Languages, Las Vegas, USA, 1995, pp. 15–20.
6. *Миков А.И.*, *Замятина Е.Б.*, *Осмехин К.А.* Метод динамической балансировки процессов имитационного моделирования. В кн. «Материалы Всероссийской научно-технической конференции «Методы и средства обработки информации МСО-2005». М.: Изд-во МГУ, 2005, с. 472–478.

The optimal algorithm for movement of objects in quadratic lattice

V.V. Morozenko, A.V. Moskalev
Perm State University, 614990, Perm, Bukireva st., 15

For natural N we have considered a task about optimal movement of objects in graph, which is Cartesian product of chain with N vertexes by itself. Let only one object has to be moved from each vertex to each other vertex. Let each movement of any object through any edge takes one time tick, and no more than one object can be moved through any edge in the graph during one time tick. We have described an algorithm of objects' movement, which requires minimal quantity of time ticks, and we have proved that such algorithm is optimal.