

3. Asratian A. S. and Casselgren C. J. Some results on interval edge colorings of (α, β) -biregular bipartite graphs. Linköping, Sweden: Linköpingsuniversitet, 2006.

УДК 519.1

СХЕМА ВЫДЕЛЕНИЯ ПАРОСОЧЕТАНИЙ¹

Т. А. Магомедов

В [1, с. 165] приведены следующие достаточные условия существования в двудольном графе $G = (X, Y, E)$ полного паросочетания множества X с множеством Y :

$$\min_{x \in X} d_G x \geq \max_{y \in Y} d_G y. \quad (1)$$

Необходимые и достаточные условия сформулированы в известной теореме Холла [1, с. 164].

Теорема 1. Для существования в двудольном графе $G = (X, Y, E)$ полного паросочетания множества X с множеством Y достаточно выполнение условий

$$\forall (x, y) \in E \quad (d_G x \geq d_G y),$$

которые в дальнейшем будем называть «условиями доминирования».

Определение 1. Пусть граф $G = (X, Y, E)$ удовлетворяет условиям доминирования. Если после удаления из E некоторого паросочетания условия доминирования выполняются, то данное удаление назовём *сохраняющим*.

Определение 2. Разбиение множества рёбер E графа $G = (X, Y, E)$ на последовательность $A, B, C, \dots$ из Δ паросочетаний называется *непрерывным*, если любое ребро, инцидентное вершине $x \in X$, включено в одно из первых $d_G x$ паросочетаний данной последовательности.

Теорема 2. Пусть граф $G_1 = (X_1, Y_1, E_1)$ удовлетворяет условиям доминирования; $G_i = (X_1, Y_i, E_i)$ — граф, полученный удалением из графа $G_{i-1} = (X_1, Y_{i-1}, E_{i-1})$ минимального паросочетания M_{i-1} , насыщающего все вершины наибольшей степени в G_{i-1} , $i = 2, \dots, \Delta$. Тогда

- 1) каждое из этих удалений является сохраняющим;
- 2) $M_\Delta \equiv E_\Delta$ является полным паросочетанием множества X_1 с множеством Y_1 в G_1 ;
- 3) последовательность $M_\Delta, \dots, M_1$ представляет непрерывное разбиение множества E_1 на Δ паросочетаний.

ЛИТЕРАТУРА

1. Свами М., Тхуласираман К. Графы, сети и алгоритмы. М.: Мир, 1984.

УДК 519.17

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ПОИСКА ОСТОВНОГО ДЕРЕВА МИНИМАЛЬНОГО ВЕСА НА ПРЕДФРАКТАЛЬНОМ ГРАФЕ

Л. И. Сенникова А., А. Кочкаров

В работе предлагается описание *параллельного алгоритма* $\mathcal{R}$ поиска остовного дерева минимального веса (ОДМВ) [1] на предфрактальном графе [2, 3].

¹Работа поддержана гос. заданием, проект №01.1923.2011.

В основе определения фрактальных графов лежит операция *замены вершины затравкой* (ЗВЗ). Термином «затравка» условимся называть какой-либо связный граф $H = (W, Q)$. Суть операции ЗВЗ заключается в следующем. В данном графе $G = (V, E)$ у намеченной для замещения вершины $\tilde{v} \in V$ выделяется множество $\tilde{V} \subseteq V$ смежных ей вершин. Далее из графа G удаляется вершина $\tilde{v}$ и все инцидентные ей рёбра. Затем каждая вершина $\tilde{v}_j \in \tilde{V}$, $j = 1, 2, \dots, |\tilde{V}|$, соединяется ребром с одной из вершин затравки H . Вершины соединяются произвольно (случайным образом) или по определённом правилу при необходимости.

Предфрактальный граф будем обозначать через $G_L = (V_L, E_L)$, где V_L — множество вершин графа, а E_L — множество его рёбер. Определим его рекуррентно, поэтапно заменяя каждый раз в построенном на предыдущем этапе $l = 1, 2, \dots, L - 1$ графе $G_l = (V_l, E_l)$ каждую его вершину затравкой H . На этапе $l = 1$ предфрактальному графу соответствует затравка $G_1 = H$. Об описанном процессе говорят, что *предфрактальный граф G_L порожден затравкой H* . Процесс порождения предфрактального графа G_L по существу есть процесс построения последовательности предфрактальных графов $G_1, G_2, \dots, G_l, \dots, G_L$, называемой *траекторией*. Фрактальный граф $G = (V, E)$, порожденный затравкой H , определяется бесконечной траекторией. Предфрактальный граф G_L условимся называть (n, q, L) -графом, если он порожден n -вершинной q -рёберной связной затравкой H .

Для предфрактального графа G_L рёбра, появившиеся на l -м, $l \in \{1, 2, \dots, L\}$, этапе порождения, будем называть *рёбрами ранга l* . *Новыми* рёбрами предфрактального графа G_L назовём рёбра ранга L , а все остальные рёбра назовём *старыми*.

При удалении из предфрактального графа G_L всех рёбер рангов $l = 1, 2, \dots, L - r$ получим множество $\{B_{L,i}^{(r)} : i = 1, 2, \dots, n^{L-r}\}$ *блоков r -го ранга*, где i — порядковый номер блока; $r \in \{1, 2, \dots, L - 1\}$. Термином *подграф-затравка $z_s^{(l)}$* будем называть блок $B_{l,s}^{(l)}$ первого ранга предфрактального графа G_l из траектории. Мощность множества $Z(G_L)$ всех подграф-затравок из траектории графа G_L равна $(n^L - 1)/(n - 1)$.

Будем говорить, что *предфрактальный граф G_L взвешен*, если каждому его ребру $e^{(l)} \in E_L$ приписано действительное число $\omega(e^{(l)}) \in (\theta^{l-1}a, \theta^{l-1}b)$, где l — ранг ребра; $a > 0$ и $\theta < a/b$.

Алгоритм $\mathcal{R}$ осуществляет поиск ОДМВ $T = (V_L, E_T)$ на взвешенном предфрактальном графе G_L . Алгоритм использует k процессоров $p_1, p_2, \dots, p_k$ на многопроцессорной вычислительной машине с распределённой памятью. Назначим каждый процессор одной из подграф-затравок $z_s^{(l)}$, $l = 1, \dots, L$, $s = 1, \dots, n^{l-1}$, предфрактального графа G_L , тогда число используемых процессоров равно $k = (n^L - 1)/(n - 1)$. Суть работы алгоритма заключается в следующем. Каждая подграф-затравка $z_s^{(l)}$ рассматривается как отдельно взятый граф. При этом каждый из k процессоров p_i , $i = 1, \dots, k$, независимо от других находит ОДМВ T_i на своей подграф-затравке $z_s^{(l)}$. Поиск ОДМВ отдельно взятой подграф-затравки осуществляется алгоритмом Прима. Нахождение ОДМВ $T_1, T_2, \dots, T_k$ всех подграф-затравок $z_s^{(l)}$ определяет ОДМВ T предфрактального графа G_L . Каждое ребро предфрактального графа имеет свой уникальный номер, однозначно определяющий ребро во всей траектории. Таким образом, выделение ОДМВ на подграф-затравке $z_s^{(l)}$ соответствует выделению множества рёбер на предфрактальном графе G_L .

Кроме принципиальной возможности эффективного распараллеливая задачи о поиске ОДМВ на предфрактальном графе важен еще и следующий факт.

Теорема 1. Вычислительная сложность алгоритма $\mathcal{R}$ для предфрактального (n, q, L) -графа (G_L) с числом вершин $|V_L| = N$ равна $O(Nn^2)$.

Вычислительная сложность алгоритма Прима равна $O(N^2)$. Сравнив её с вычислительной сложностью алгоритма $\mathcal{R}$, получаем, что при реализации алгоритма $\mathcal{R}$ на одном процессоре поиск ОДМВ на предфрактальном графе будет осуществлен быстрее, чем широко известным алгоритмом Прима.

ЛИТЕРАТУРА

1. Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. Лекции по теории графов. М.: Наука, 1990.
2. Кочкаров А. А., Кочкаров Р. А. Параллельный алгоритм поиска кратчайшего пути на предфрактальном графе // Журнал вычисл. матем. и матем. физики. 2004. Т. 44. № 6. С. 1157–1162.
3. Кочкаров А. А., Сенникова Л. И. Количественные оценки некоторых связностных характеристик предфрактальных графов // Прикладная дискретная математика. 2011. № 4(14). С. 56–61.

УДК 519.5

ВОССТАНОВЛЕНИЕ ГРАФА КОЛЛЕКТИВОМ АВТОМАТОВ

Е. А. Татаринов

Рассматривается задача [1] восстановления конечного связного неориентированного графа G без петель и кратных ребер при помощи агента, который перемещается по рёбрам графа G , считывает и изменяет метки на его вершинах и инциденторах. На основе собранной информации агент строит граф H , изоморфный графу G с точностью до меток на элементах графов. Требуется найти алгоритм обхода и разметки графа G для решения этой задачи.

Известен ряд алгоритмов, реализующих восстановление графа при помощи построения на его вершинах неявной нумерации [2] при помощи агента A , они подробно описаны в [3, 4]. Наиболее простым в реализации является Базовый Алгоритм [3], однако он имеет кубическую, от числа вершин в графе, верхнюю оценку временной сложности. Предлагается модификация Базового Алгоритма, понижающая эту оценку. При этом верхняя оценка временной сложности зависит от количества агентов, которые проводят восстановление графа.

В [4] показано, что верхняя оценка временной сложности зависит от длины максимального простого цикла t в графе, цикломатического числа q [5] и количества вершин n исследуемого графа и равна $O(n + qt)$. В процессе восстановления агент разбивает рёбра графа на два множества: древесные и обратные [5], а все пройденные вершины, у которых не все рёбра восстановлены, образуют красный путь [3].

Наибольшего времени требуют обратные ребра, для восстановления которых агент выполняет проход по вершинам красного пути, длина которого соизмерима с длиной наибольшего простого цикла. Для сокращения этого прохода используются агенты A_i , $i = 1, \dots, j$. Они двигаются вдоль красного пути, сохраняя между собой равное расстояние. Для этого они обмениваются сообщениями A с A_j , A_i с A_{i-1} для $i = 2, \dots, j$. Для каждого агента A_i , $i = 1, \dots, j$, фиксируется длина красного пути от его начала (конца) до вершины, в которой находится этот агент.