

Кусочно-непрерывные пути в задачах построения и оптимизации расписаний¹

А.М. Магомедов

Аннотация. Исходные данные к расписанию заданы в виде двудольного графа, где вершинам двух долей графа соотнесены специализированные процессоры и задания соответственно, ребрам – предписанные операции единичной длительности по обработке заданий процессорами. Расписание рассматривается как отображение множества ребер графа в множество положительных целых чисел – «цветов» ребер (множество дискретных временных промежутков единичной длительности, назначенных для выполнения той или иной операции). В качестве теоретико-графовой модели расписания для мультипроцессорной системы без простоев процессоров и прерываний заданий рассматривается реберная интервальная раскраска двудольного графа. Определены необходимые для анализа конструкции графа, с их использованием установлен ряд свойств интервальной раскраски. На основе понятия кусочно-непрерывного пути разработан эвристический алгоритм интервальной раскраски, эффективный для случаев, представляющих интерес как для теории, так и для прикладных задач оптимизации расписаний. Обсуждаются результаты применения программного обеспечения, разработанного на основе эвристического алгоритма, и перспективы работы.

Ключевые слова: расписание, граф, алгоритм, раскраска, сложность.

Введение

При исследовании задач теории расписаний часто прибегают к методам теории графов. В качестве примеров укажем на формулировку задачи составления расписания учебных занятий в [1, с. 403], исследование приводимости «школьного» расписания к «беззаконному» виду [2] и на обсуждение расписаний обслуживания множества заданий в мультипроцессорной системе без простоев процессоров и прерываний обработки каждого задания [3], а также [4-7].

Последний случай рассмотрим подробнее. Пусть предписано обработать множество заданий $X = \{x_0, \dots, x_{n-1}\}$ в системе специализированных процессоров $Y = \{y_0, \dots, y_{m-1}\}$. Для каждого задания x_i определен набор процессоров, каждый из которых должен выполнить од-

ну операцию по обработке x_i ; длительность операции равна единице, запрещено одновременное участие объекта (задания, процессора) в двух и более операциях, условия предшествования отсутствуют. Требуется выяснить, существует ли расписание без простоев объектов.

Рассмотрим двудольный граф $G = (X, Y, E)$, где $(x_i, y_j) \in E$ если и только если процессору y_j предписано выполнить операцию по обработке задания x_i . Составить расписание означает указать для каждого $(x_i, y_j) \in E$ некоторое целое положительное число $c(x_i, y_j)$ – номер дискретного момента времени, в течение которого y_j выполняет операцию над x_i . Если рассматривать $c(x_i, y_j)$ в качестве цвета ребра (x_i, y_j) , получим эквивалентную задачу ребер-

¹ Работа выполнена при финансовой поддержке гос. задания 0202-2016-0007, выполняемого Отделом математики и информатики ДНЦ РАН.

ной раскраски двудольного графа. Запрет на одновременное участие объекта в двух и более операциях в терминологии теории графов означает, что раскраска должна быть *правильной*, т.е. в каждой вершине все *представленные* в ней цвета (цвета всех инцидентных ребер) должны быть различны. А требование отсутствия простоев означает, что раскраска должна быть не только *правильной*, но и *интервальной*, т.е. в каждой вершине графа все представленные в ней цвета должны образовать целочисленный интервал.

Следующее определение взято из [2]. *Интервальной t -раскраской* графа $G = (V, E)$ называется правильная раскраска ребер G в цвета $1, 2, \dots, t$, при которой в каждый цвет окрашено хотя бы одно ребро, а для каждой вершины $v \in V$ ребра, инцидентные v , окрашены в $d(v)$ последовательных цветов. Когда значение t безразлично, в приведенном определении упоминание t опускается. В разделе 1 (Утверждение 2) мы покажем, что приведенное определение интервальной раскраски избыточно.

В [2] доказано, что задача об интервальной раскраске заданного графа $G = (V, E)$ NP-полна; свойство -полноты сохраняется и для класса двудольных графов [8]. Однако для регулярных графов аналогичная преемственность не наблюдается: задача об интервальной раскрашиваемости обыкновенного регулярного графа -полна [2], при этом, как мы увидим в разделе 1 (Утверждение 4), любой регулярный двудольный граф допускает интервальную раскраску.

В утверждениях раздела 1 используется понятие цепи. Напомним [9], что цепью в связном графе $G = (V, E)$ называется последовательность вершин a_i и попарно различных ребер b_i :

$$P_0 = (a_0, b_0, a_1, b_1, \dots, b_{n-1}, a_n), \quad (1)$$

где каждые два соседних ребра b_{i-1} и b_i имеют общую вершину a_i . Если все вершины в (1) попарно различны, то цепь называется *простой*. Если концевые вершины a_0 и a_n простой цепи (1) совпадают, то (1) называется простым циклом. Цепь (1) в графе G назовем *непродолжимой*, если P_0 содержит все ребра, инцидентные вершине a_n . Для дальнейших наших целей естественным представляется вместо «просто» непродолжимой цепи рассматривать самую длинную цепь в графе. Однако задача вычисления самой длинной

цепи в заданном графе $G = (V, E)$ является NP-полной [10, с. 104]; как показано в [10, с. 269], к этой задаче сводится задача «Гамильтонов путь между двумя вершинами».

В основе изложенного в разделе 2 алгоритма интервальной раскраски лежит понятие *кусочно-непрерывного пути*. Приведем рекурсивное определение кусочно-непрерывного пути в связном графе G . Пусть P – произвольная непродолжимая цепь. Пока не все ребра графа включены в P , выполним действия: 1) обозначим через a_k последнюю из вершин цепи P , имеющих инцидентное ребро, не входящее в P (ввиду свойства связности такая вершина всегда найдется); 2) построим непродолжимую цепь P_1 , начинающуюся с вершины a_k и содержащую лишь ребра, не включенные в P ; 3) результат конкатенации P и P_1 и обозначим снова через P . Последовательность ребер графа G , перечисленных в том порядке, в каком они следуют в итоговом P , будем называть кусочно-непрерывным путем в графе G . Очевидно, что кусочно-непрерывных путей в графе G может быть несколько.

В разделе 1 доказаны утверждения об интервальной раскраске и расписаниях. В разделе 2 приведен алгоритм проверки существования интервальной раскраски, представляющий результат глубокой переработки алгоритма, приведенного в [7]; достаточно сказать, что количество попыток, необходимых для раскраски очередного ребра без перекрашивания остальных ребер, удалось уменьшить с восьми [7] до двух.

1. Интервальная раскраска связного графа

В случае несвязного графа задача об интервальной раскраске графа сводится к рассмотрению каждой связной компоненты графа, поэтому ограничимся случаем связного графа.

Утверждение 1. Пусть задана интервальная раскраска связного графа и

$$P_0 = (a_0, b_0, a_1, b_1, \dots, b_{n-1}, a_n)$$

– цепь, соединяющая два ребра: $b_0 = (a_0, a_1)$ и $b_{n-1} = (a_{n-1}, a_n)$ с цветами t_0 и t_{n-1} соответственно, $t_0 < t_{n-1}$. Тогда для любого τ , $t_0 < \tau < t_{n-1}$ найдется ребро цвета τ , инцидентное некоторой вершине из P_0 .

Доказательство. Обозначим интервал цветов, представленных в вершине a_i , через $[m_i, M_i]$; $m' = \min_i m_i$, $M' = \max_i M_i$. Т.к. каждые два соседних интервала последовательности

$$[m_0, M_0], [m_1, M_1], \dots, [m_{n-1}, M_{n-1}]$$

имеют непустое пересечение, то каждый цвет из интервала $[m', M']$ представлен в некоторой вершине цепи P_0 . Поскольку $t_0 \in [m_0, M_0]$, $t_{n-1} \in [m_{n-1}, M_{n-1}]$, то $t_0, t_{n-1} \in [m', M']$; следовательно, и $\tau \in [m', M']$, поэтому цвет τ представлен в некоторой вершине цепи P_0 .

Утверждение 2. Определение интервальной t -раскраски, приведенное во Введении, равносильно следующему: «Интервальной t -раскраской связного графа G называется раскраска ребер G в цвета $1, 2, \dots, t$, при которой найдутся ребра, окрашенные в цвета 1 и t , и для каждой вершины v цвета, представленные в v , образуют множество последовательных чисел».

Доказательство. Утверждение немедленно следует из Утверждения 1.

Утверждение 3. Простой цикл четной длины всегда обладает интервальной раскраской, а простой цикл нечетной длины не имеет интервальной раскраски.

Доказательство. Пусть для простого цикла с N вершинами (и N ребрами) интервальная раскраска существует. Тогда каждой из его вершин инцидентно одно ребро с цветом $1 \pmod{2}$ и одно ребро с цветом $0 \pmod{2}$; отсюда следует, что число ребер N четно. С другой стороны, в случае четной длины простого цикла для получения интервальной раскраски достаточно выполнить обход цикла, попеременно раскрашивая ребра в цвета $1, 2, 1, 2, \dots$

Утверждение 4. Если некоторый граф $G = (V, E)$ обладает интервальной раскраской, то граф, полученный добавлением к G висячего ребра, также обладает интервальной раскраской.

Доказательство. В самом деле, пусть граф G обладает интервальной раскраской, $[m, M]$ – интервал цветов, представленных в вершине $u \in V$, а G' – граф, полученный добавлением к G висячего ребра (u, v) . Окрасив ребро (u, v) в цвет $M + 1$, получим искомую интервальную раскраску графа G' .

Из теоремы Эйлера [9, с. 192] следует, что связный регулярный граф G четной степени Δ обладает эйлеровым циклом (под степенью ре-

гулярного графа будем понимать степень вершины графа); по теореме Ю. Петерсена [11] связный регулярный граф четной степени $2k$ допускает разложение на k 2-факторов (каждая связная компонента 2-фактора является гамильтоновым циклом). Если каждый из этих 2-факторов является набором циклов четной длины в графе G , то, помечивая последовательные ребра каждого цикла i -го набора метками $2i, 2i + 1, 2i, 2i + 1, \dots$ ($i = 0, 1, \dots, k - 1$), получим интервальную раскраску.

Из перечисленных свойств графа сохраним лишь требование четности длины любого цикла. Но последнее, согласно теореме Кенига (1936 г.), означает двудольный характер графа, а как уже отмечалось ранее, не всякий двудольный граф интервально раскрашиваем. Более того, проверка интервальной раскрашиваемости заданного двудольного графа является NP -полной задачей, следовательно, не существует эффективно проверяемых необходимых и достаточных условий интервальной раскрашиваемости (в предположении истинности гипотезы $P \neq NP$). Поэтому для обеспечения свойства интервальной раскрашиваемости следует добавить некоторое дополнительное условие (разумеется, слабее регулярности с четной степенью). Потребуем выполнение свойства регулярности без условия четности степени графа Δ . В случае нечетного Δ достаточно найти в заданном регулярном двудольном графе $G = (X, Y, E)$ полное паросочетание X с Y (существование такого паросочетания следует из [1, следствие 8.16.1, с. 165]), раскрасить его ребра цветом $\Delta - 1$ и удалить его, затем воспользоваться теоремой Ю. Петерсена для интервальной раскраски оставшегося графа $\Delta - 1$ цветами: $0, 1, \dots, \Delta - 2$. Таким образом, мы получили доказательство следующего известного утверждения [12].

Утверждение 5. Любой регулярный двудольный граф допускает интервальную раскраску.

Проверка существования расписания обработки n объектов $x_0, x_1, \dots, x_{n-1}$ с заданными условиями частичного предшествования (для некоторых пар объектов указано, что один из объектов пары должен быть обработан раньше другого) в однопроцессорной системе сводится

к проверке существования сортировки объектов в соответствии с условиями частичного предшествования (назовем искомое расписание «тривиальным»).

Сопоставим множеству объектов $\{x_0, x_1, \dots, x_{n-1}\}$ множество вершин $V = \{v_0, v_1, \dots, v_{n-1}\}$ и рассмотрим орграф $G = (V, E)$, где дуги соответствуют условиям частичного предшествования: из вершины v_i в вершину v_j дуга проведена тогда и только тогда, когда объект x_i предшествует объекту x_j .

Утверждение 6. Тривиальное расписание существует тогда и только тогда, когда орграф G является ациклическим.

Доказательство. Если в орграфе G имеется цикл, то тривиальное расписание, очевидно, не существует (ориентированный граф имеет цикл в том и только в том случае, когда алгоритм поиска в глубину находит обратную дугу [13]). Если G – ациклический орграф, то [1, с. 95] его вершины можно пометить числами из множества $\{0, 1, \dots, n-1\}$ таким образом, что если $(i, j) \in E$, то $i < j$.

Замечание. Определенное в доказательстве Утверждения 6 упорядочение вершин называется *топологической сортировкой*; процедура топологической сортировки приведена в [1, с. 96]. Задача топологической сортировки «текстуально» близка к задаче поиска ориентированного гамильтонова пути в каждой компоненте ориентированного графа. Как уже отмечалось, в общем случае задача о гамильтоновом пути NP-полна, но для ациклических ориентированных графов задача разрешима за полиномиальное время [14].

2. Жадный алгоритм интервальной раскраски

Требуется проверить, существует ли интервальная раскраска заданного связного двудольного графа $G = (V, E)$, $|V| = n$, $|E| = m$, не более чем t цветами (и выполнить раскраску в случае существования). Известно [11], что t не может быть более $n-1$; положим t равным $n-1$ (заметим, что это единственный пункт алгоритма, учитывающий двудольную природу графа G).

Далее использованы некоторые обозначения языков программирования, понятные без дополнительных объяснений (например, $i++$ означает увеличение значения переменной i на единицу).

Построим кусочно-непрерывный путь $P = \{e_0, \dots, e_{m-1}\}$ и будем последовательно окрашивать ориентированные ребра $e_i = (v_i, w_i)$ цветами из целочисленного интервала $[1..t]$ так, что в любой момент времени в каждой вершине v цвета окрашенных инцидентных ребер образуют некоторый список L_v , составленный из последовательных целых чисел. Алгоритм раскраски, обладающий таким свойством, будем называть *жадным алгоритмом*. Массив текущих списков цветов в вершинах множества V будем обозначать через L , $L = \{L_{v_0}, \dots, L_{v_{n-1}}\}$. *Приграничным цветом* для пустого интервала назовем любой цвет из интервала $[1..t]$. Для непустого интервала $[a..b]$ *нижним приграничным цветом* назовем цвет $a-1$, если $1 \leq a-1 \leq t$, *верхним приграничным цветом* – цвет $b+1$, если $1 \leq b+1 \leq t$; в случае существования, нижний и верхний приграничные цвета для списка L_{v_i} будем обозначать через $low(v_i)$ и $high(v_i)$ соответственно.

Рассмотрим ситуацию, когда ребра $e_0, \dots, e_{i-1}$ уже окрашены и требуется окрасить ребро e_i , $i > 0$. Список цветов в начальной вершине v_i ориентированного ребра $e_i = (v_i, w_i)$ не пуст, т.к. содержит цвет ребра e_{i-1} . Очевидно, что окраска ребра e_i без изменений цветов ранее окрашенных i ребер возможна только в следующих двух случаях: 1) $low(v_i)$ существует и является приграничным цветом для w_i ; 2) $high(v_i)$ существует и является приграничным цветом для w_i . Соответственно допустимы две попытки окрасить ребро e_i без изменений цветов ранее окрашенных ребер; номера 1 или 2 предпринятых попыток будем сохранять в ячейке q_i (удобно использовать и третье значение $q_i = 0$ в качестве признака, что ни одна из двух попыток пока не предпринята). Заметим, что элементы массива q индексируются, начиная с единицы, в отличие от всех других массивов, используемых в нашем алгоритме.

Для записи и хранения цвета ребра e_i будем использовать ячейку C_i . В качестве признака успешного (т.е. с генерацией интервальной

раскраски) или неудачного (т.е. без получения интервальной раскраски) завершения жадного алгоритма используется логическая переменная *Success*.

Перейдем к изложению жадного алгоритма, который осуществляет перебор с возвратами и включает стадию инициализации и итерации с двумя пунктами: «Прямой шаг» и «Возвратный шаг».

2.1. Инициализация

$Success := false$ (интервальная раскраска отсутствует); всем v_i (соответственно w_i) присвоить номера начальных (конечных) вершин ориентированных ребер e_i из кусочно-непрерывного пути P , $i = 0, \dots, m-1$; для всех вершин выбрать пустые списки цветов: $L_{v_0} := \emptyset$; ...; $L_{v_{n-1}} := \emptyset$; всем элементам массивов q и C присвоить нулевые значения: $q_1 := 0$; ...; $q_{m-1} := 0$; $C_0 := 0$; ...; $C_{m-1} := 0$; начальное ребро e_0 окрасить в наименьший допустимый цвет: $C_0 := 1$; $L_{v_0} := \{1\}$; $L_{w_0} := \{1\}$; в качестве очередного ребра для окрашивания выбрать ребро с номером 1: $i := 1$.

2.2. Прямой шаг

FWRD: данный пункт выполняется сразу после инициализации или после возвратного шага и заключается в окрашивании (более точно, в попытке окрашивания) ребра, номер которого записан в ячейке i .

2.2.1. Рассмотрим сначала случай $i = 0$:

если $C_0 < t$, то $\{C_0++$; добавить C_0 в списки L_{v_i} и L_{w_i} ; $i++$; перейти к метке FWRD}

иначе $\{Success := false$; завершить алгоритм}.

Замечание. При первом выполнении прямого шага сразу после инициализации i равно единице, однако переход к прямому шагу после возвратных шагов может быть осуществлен и при $i = 0$.

2.2.2. Рассмотрим случай $i > 0$.

Увеличить на единицу текущий номер попытки раскрасить ребро e_i : q_i++ .

В зависимости от равенств: $q_i=1$ или $q_i=2$ перейти к пунктам а) или б) соответственно.

а) Данный пункт выполняется при $q_i = 1$.

Если $low(v_i)$ для начальной вершины v_i рассматриваемого ребра e_i существует и является приграничным цветом для конечной вершины w_i этого ребра,

то

{
 $C_i := low(v_i)$; добавить $low(v_i)$ в списки L_{v_i} и L_{w_i} ;
если $i < m-1$, то $\{i++$; перейти к метке FWRD},
иначе $\{Success := true$; завершить алгоритм}

иначе $\{q_i := 2$; перейти к следующему пункту б).

б) Данный пункт выполняется при $q_i = 2$.

Если $high(v_i)$ для начальной вершины v_i рассматриваемого ребра e_i существует и является приграничным цветом для конечной вершины w_i этого ребра,

то

{
 $C_i := high(v_i)$; добавить $high(v_i)$ в списки L_{v_i} и L_{w_i} ;
если $i < m-1$, то $\{i++$; перейти к метке FWRD},
иначе $\{Success := true$; завершить алгоритм}

иначе перейти к метке BACK.

2.3. Обратный шаг

BACK:

данный пункт выполняется, если в прямом шаге не удалось окрасить ребро e_i , $0 < i$, $q_i = 2$.

Пока $(i > 0)$ и $(q_i = 2)$

{
 $q_i = 0$; $i--$; удалить C_i из списков L_{v_i} и L_{w_i} ;
если $i = 0$, перейти к метке FWRD;
если $q_i < 2$, перейти к метке FWRD;
}

Конец алгоритма.

Заключение

Известно [15], что все двудольные графы с числом вершин $n < 15$ допускают интервальную раскраску, для $n = 15, 16, 17, 18$ вопрос остается открытым. Программное обеспечение [16; 17], разработанное на основе жадного алгоритма, было протестировано на значительном количестве (> 366 тыс.) двудольных графов $G = (X, Y, E)$, $|X| = 4$, $|Y| = 11$. В 97% случаях компьютерной программе удалось найти интервальную раскраску, что можно считать признаком высокой надежности жадного алгоритма. Пример интервальной раскраски приведен на Рис. 1.

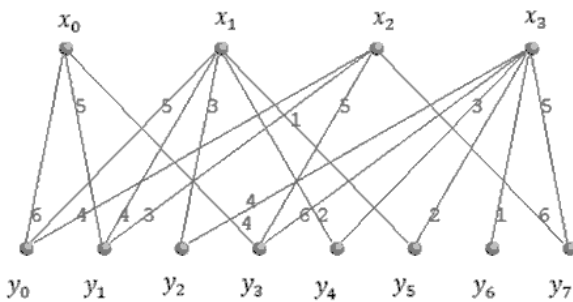


Рис. 1. Интервальная раскраска, сгенерированная программным обеспечением [17]

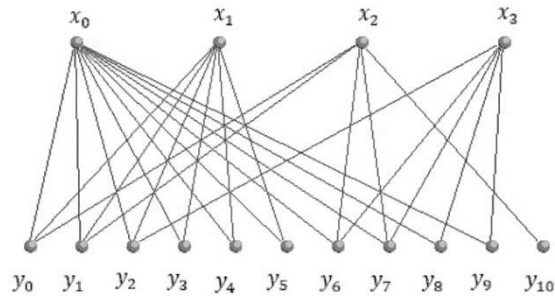


Рис. 2. Пример графа, для которого жадный алгоритм не смог найти интервальной раскраски

Ближайшей перспективой является разработка алгоритма (и соответствующего программного обеспечения), способного с полной определенностью ответить на вопрос об интервальной раскрашиваемости **всех** двудольных графов вида $G = (X, Y, E)$, $|X| = 4, |Y| = 11$. Однако полное решение даже для случая $n = |X| + |Y| = 15$ представляется автору нетривиальной задачей. При этом, с учетом результатов статьи [15] и Утверждения 4 из раздела 1 данной статьи, достаточно ограничиться связными двудольными графами без висячих ребер. В этой связи заметим, что жадному алгоритму (более точно, программному обеспечению [17]) не удалось найти интервальной раскраски для графа, приведенного на Рис. 2, в то время как из сказанного следует, что такая раскраска существует: (x_2, y_{10}) – висячее ребро, $n = 15$.

Литература

1. Свами М., Тхуласираман К. Графы, сети и алгоритмы / Пер. с англ. – М.: Мир, 1984. 455 с. (Swamy, M.N.S., and K. Thulasiraman. Graphs, Networks and Algorithms. — N.Y.: Wiley, 1980. 612 p.)
2. Асратян А. С., Камалян Р. Р. Интервальные раскраски ребер мультиграфа // Прикладная математика, вып. 5. Ереван: Изд-во Ереванского ун-та. 1987. С. 25-34.
3. Танаев В.С., Сотсков Ю.Н., Струсевиц В.А. Теория расписаний. Многостадийные системы. М.: Наука, 1989. – 328 с.
4. Магомедов А.М. К вопросу об интервальной Δ - раскраске двудольных графов // Автоматика и телемеханика. 2015. № 1. С. 101–109.
5. Магомедов А.М., Магомедов Т.А. Реберно-вершинные инцидентные паросочетания в задачах расписаний // Прикладная дискретная математика. 2015. №1 (27). С. 92-95.
6. Магомедов А.М., Магомедов Т.А. Последовательное разбиение ребер двудольного графа на паросочетания // Дискретная математика. 2016. Т. 28. Вып. 1. С. 78-86.
7. Магомедов А.М. Цепочечные структуры в задачах о расписаниях // Прикладная дискретная математика. 2016. №3 (33). С. 67-77.
8. Севастьянов С.В. Об интервальной раскрашиваемости ребер двудольного графа // Методы дискретного анализа. 1990. Т. 50. С. 61-72.
9. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М.: Книжный дом «Либроком», 2009. – 392 с.
10. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. 416 с. (Garey, M. R., and D.S. Johnson. Computers and Intractability. W.H. Freeman, 1979. 347 p.)
11. Petersen, Julius (1891), Die Theorie der regulären graphs, Acta Mathematica 15: p. 193–220.
12. Камалян Р.Р. Интервальные раскраски полных двудольных графов и деревьев // Препринт ВЦ АН Армянской ССР, Ереван. 1989. 11 с.
13. Tucker, Alan. Chapter 2: Covering Circuits and Graph Colorings // Applied Combinatorics. — 5th. — Hoboken: John Wiley & sons, 2006. — P. 49.
14. Lawler, E.L. Combinatorial Optimization: Networks and Matroids. – New York: Holt, Rinehart and Winston, 1976.
15. Giaro, K. Compact task scheduling on dedicated processors with no waiting period (in Polish) // PhD thesis, Technical University of Gdansk, IETI Faculty, Gdansk. 1999.
16. Магомедов А.М. Реберная раскраска двудольных графов // Свидетельство № 2016660830 о государственной регистрации программы для ЭВМ от 22.09.2016.
17. Магомедов А.М. Генерация графов и их раскраска методом жадного алгоритма // Свидетельство № 2017611516 о государственной регистрации программы для ЭВМ от 6.02.2017.

Магомедов Абдулкарим Магомедович. Заведующий кафедрой ФГБОУ высшего образования «Дагестанский государственный университет» (ДГУ), г. Махачкала. Окончил ДГУ в 1972 году. Доктор физико-математических наук, профессор. Количество печатных работ: 190. Область научных интересов: информационные технологии и дискретная математика. E-mail: magomedtagirl@yandex.ru

Piecewise continuous paths in the task of building and optimizing schedules

A.M. Magomedov

Data for the schedule is given in a form of a bipartite graph where the vertices of the two parts of the graph are associated with specialized processors and tasks, and the edges are associated with prescribed operations of single-unit duration for processing by the processors. The schedule is viewed as a mapping from a set of graph edges to the set of positive integers – "colors" of edges (the set of discrete time intervals of single-unit duration assigned to perform a particular operation). Problem of edge interval coloring of a bipartite graph is considered as a theoretical-graph model for schedule of a multiprocessor system without downtime of the processors and job interruptions. Graph structures necessary for subsequent sections are defined in the Introduction. Using those in section 1 a number of properties of interval coloring are defined. Based on the concept of piecewise continuous path in section 2, we developed a heuristic algorithm for interval coloring, effective for both theoretical and practical problems of schedule optimizations. In the Conclusion we discuss the results of the application of software developed on the basis of that heuristic algorithm and future prospects.

Keywords: schedule, graph, algorithm, colors, complexity

References

1. Swamy, M.N.S. and K. Thulasiraman. Graphs, Networks and Algorithms. — N.Y.: Wiley, 1980. 612 p.
2. Asratjan, A.S., and R.R. Kamaljan. 1987. Interval'nye raskraski reber mul'tigrafa [Interval coloring of the edges of the multi-graph]. Prikladnaja matematika. Erevan: Izd-vo Erevanskogo un-ta [Applied mathematics. Yerevan: Publishing house of the Yerevan University] 5: 25-34.
3. Tanaev, V.S., Ju.N. Sotskov, and V.A. Strusevich, eds 1989. Teorija raspisanij. Mnogostadijnye sistemy [The scheduling theory. Multistage system]. Moscow: Science. 328 p.
4. Magomedov, A.M. 2015. K voprosu ob interval'noj Δ -raskraske dvudol'nyh grafov [To the question about the interval Δ -coloring of bipartite graphs]. Avtomatika i telemekhanika [Automatics and telemechanics] 1:101-109.
5. Magomedov, A.M., and T.A. Magomedov. 2015. Reberno-vershinnye incidentnye parosochetaniya v zadachah raspisanij [The edge-vertex incident matching in the task schedules]. Prikladnaja diskretnaja matematika [Applied discrete mathematics] 1 (27): 92-95.
6. Magomedov, A.M., and T.A. Magomedov. 2016. Posledovatel'noe razbienie reber dvudol'nogo grafa na parosochetaniya [Successive splitting of edges on a bipartite graph matching]. Diskretnaja matematika [Discrete mathematics] 1(28): 78-86.
7. Magomedov, A.M. 2016. Cepochechnye struktury v zadachah o raspisanijah [Chain structure in schedule's tasks]. Prikladnaja diskretnaja matematika [Applied discrete mathematics] 3 (33): 67-77.
8. Sevast'janov, S.V. 1990. Ob interval'noj raskrashivaemosti reber dvudol'nogo grafa [About the interval-coloring of edges of the bipartite graph]. Metody diskretnogo analiza [Discrete analysis methods] 50: 61-72.
9. Emelichev, V.A., O.I. Mel'nikov, V.I. Sarvanov, and R.I. Tyshkevich, eds. 2009. Lekcii po teorii grafov [Lectures on graph theory]. Moscow: Book house «Librokom». 392 p.
10. Swamy, M.N.S. and K. Thulasiraman. Graphs, Networks and Algorithms. — N.Y.: Wiley, 1980. 612 p.
11. Petersen, Julius (1891), Die Theorie der regulären graphs, Acta Mathematica 15: p. 193–220.
12. Kamaljan, R.R. 1989. Interval'nye raskraski polnyh dvudol'nyh grafov i derev'ev [Interval coloring of complete bipartite graphs and trees]. Preprint VC AN Armjanskoy SSR, Erevan [Preprint VTS an Armenian SSR, Yerevan]. 11 p.
13. Tucker, Alan. Chapter 2: Covering Circuits and Graph Colorings // Applied Combinatorics. — 5th. — Hoboken: John Wiley & sons, 2006. — P. 49.
14. Lawler, E. L. Combinatorial Optimization: Networks and Matroids. — New York: Holt, Rinehart and Winston, 1976.
15. Giaro, K. Compact task scheduling on dedicated processors with no waiting period (in Polish) // PhD thesis, Technical University of Gdansk, IETI Faculty, Gdansk. — 1999.
16. Magomedov A.M. 2016. Rebernaja raskraska dvudol'nyh grafov [Edge-coloring of bipartite graphs]. Svidetel'stvo № 2016660830 o gosudarstvennoj registracii programmy dlja JeVM [Certificate No. 2016660830 on the state registration of the computer program].
17. Magomedov A.M. 2017. Generacija grafov i ih raskraska metodom zhadnogo algoritma [Generation of graphs and their coloring method, greedy algorithm]. Svidetel'stvo № 2017611516 o gosudarstvennoj registracii programmy dlja JeVM [Certificate No. 2017611516 on the state registration of the computer program].

Abdulkarim Magomedov Magomedovich. Federal state budgetary educational institution of higher professional education "Dagestan state University" (DGU) in Makhachkala, head. chair of discrete mathematics and Informatics, doctor of physical-mathematics sciences, professor. Education (diploma): Mathematics, graduated from Dagestan state University (DGU) in 1972. The number of publications, including evidence of the patent office to software – 190 (monographs no). Research interests: information technology and discrete mathematics. E-mail: magomedtagirl@yandex.ru