УДК. 681.324

С.В.ЛИСТРОВОЙ[1], Е.С.ЛИСТРОВАЯ[2]

[1]*Украинская государственная академия железнодорожного транспорта, Харьков*
[2]*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина*

ОБЩИЙ ПОДХОД К ОРГАНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ПРИ РЕШЕНИИ ЗАДАЧ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ И ТЕОРИИ ГРАФОВ

В работе предложена универсальная процедура решения произвольных задач дискретной оптимизации, которая может быть эффективно распараллелена для решения на основе циклических параллельных вычислительных структур. Решение задач основано на представление пространства возможных решений задач дискретной оптимизации и теории графов в виде стянутого древа всех путей, что позволяет сводить решение рассматриваемых задач к определению экстремальных путей в стянутом дереве путей, формирование которых на ярусе дерева может осуществляться одновременно, что и позволяет организовать эффективное распараллеливание вычислительного процесса.

Ключевые слова: *параллельные циклические вычисления, стянутое дерево путей графа.*

Введение

Покажем, что для определения свойств произвольной сложной системы можно использовать стянутое дерево всех путей графа позволяющее с единых позиций решать произвольные задачи теории графов и комбинаторной оптимизации, к которым сводятся модели функционирования широкого круга систем управления. Использование стянутого дерева путей в свою очередь позволит эффективно распараллелить процесс их решения на одном типе архитектур параллельных вычислительных систем (ПВС), что в некотором смысле снимает проблему, связанную с тем, что для различных типов задач необходимо разрабатывать различные архитектуры ПВС [1, 2].

1. Постановка задачи на исследование

Рассмотрим конечное множество произвольных объектов теории графов или конфигураций комбинаторной оптимизации. В общем случае объект может быть произвольным, но должно быть определено конечное множество элементов $\Omega = \{\omega_l\}$ или подмножества $L_i \in \Omega$ и правила R позволяющие формировать объекты из исходных элементов или подмножеств L_i принадлежащие множеству Ω. Пусть задано некоторое разделение множества Ω на семейства подмножеств $\{L_i\}$ такое, что $\bigcup_i L_i = \Omega$ и L_i описывают интересующие нас объекты состоящие из базовых элементов {li} таких, что $\bigcup_i l_i = \Omega$ и правило R позволяющее из базовых элементов определять весовые характеристики произвольных объединений $L_к \cup L_p \in \Omega$ характеризующие свойства $\{\nu\}$ и требуется определить объект с интересующим нас свойством $\nu^* \in \{\nu\}$.

Представим множество всех возможных объединений подмножеств L_i, в виде графа $D_{□}$ с параллельно ярусной структурой, состоящей из n горизонтальных линеек с вершинами 1, 2, … , n и n ярусами, каждый из которых содержит все вершины графа $D_{□}$, при этом каждой вершине графа $D_{□}$ поставим в соответствие базовый элемент l_i.

В графе $D_{□}$ произвольная вершина i может быть достигнута путями рангов r = 1, r = 2,….., r = n – 1, а произвольному пути μ_{st}, удовлетворяющему правилам построения R и проходящему через вершины (j, p, … k, t) соответствует объединение базовых элементов $(l_j \cup l_p \cup \ldots \cup l_k \cup l_t)$, определяющих некоторый объект $L_i \in \Omega$.

Длина этого пути $d(\mu_{st})$ определяется по правилам, принадлежащим множеству R. Следовательно, множество всех путей $m_{si}(r)$ в графе $D_{□}$, удовлетворяющих правилам R определяет область допустимых решений исходной задачи по выделению объекта с интересующим нас свойством $\nu^* \in \{\nu\}$.

В качестве исходной вершины в графе $D_{□}$ будем использовать фиктивную вершину S, которую в некоторых случаях удобно отождествлять с нулевым или исходным состоянием системы, это приводит к тому, что максимальный ранг пути в графе $D_{□}$ становится равным n, а добавление вершины S к базовым элементам системы не изменяет их свойств, определяемых правилами R.

Множество сложных систем, обладающих различными свойствами $\{F_i\}$, может быть отображено с помощью некоторого подмножества графов $\{G_i\}$. Рассмотрим произвольный n вершинный граф $G(V,E) \in \{G_i\}$, который описывает состояние системы $F \in \{F_i\}$ с конечным числом состояний n. Вершины $\{i\} \in V$ графа G(V,E) соответствуют возможным состояниям системы, пути в графе G(V,E) определяемые последовательностью прохождения вершин {vi} и ребер{(i, j)} характеризуют возможный порядок достижения состояния i = p из некоторого исходного состояния s.

Важной характеристикой пути является ранг пути r – число ребер, образующих путь.

В графе G(V,E) максимальное значение ранга r = n – 1, и в общем случае ранг произвольного пути μsp характеризует сумму начального состояния, конечного состояния и числа состояний предшествования, через которые может быть достигнуто состояние p, из некоторого исходного состояния s.

Тогда множества путей m^r_{sj}, $j = (\overline{1,n})$, определяют способы достижения состояния j.

В качестве базовых элементов {li} выберем подмножество $\{v_i\} \in V$ вершин графа G (V,E), тогда объектам $\{L_j\}$ будут соответствовать все множество объектов Ω, которое можно породить на множестве V, используя правила R например, клики в произвольных графах, независимые множества, циклы графов, вектора и матрицы, задающие некоторый объект в G (V, E).

Каждый объект будем характеризовать m + 1 весовой характеристикой, где m – это некоторые второстепенные характеристики объекта, на которые в общем случае могут быть наложены ограничения на то, что они не должны превышать некоторых величин $\{b_i\}$ i = (1, 2, …, m) и имеется один определяющий показатель качества объекта, построив из исходных элементов или их подмножеств объект, принадлежащий множеству Ω и обладающий определенным свойством ν.

Правила P определения весовых характеристик объектов естественно должны определяться в соответствии с правилами формирования самих объектов и $P \in R$.

Таким образом, пути μ^r_{sj} в графе D□ соответствует объект L_j, который может быть построен из r базовых элементов $\{v_j\}$, включая элемент j, на основе правил R, а множества путей m^r_{sj}; $j = (\overline{1,n})$ определяют множество объектов Lj, которые можно построить из r базовых элементов {vj}, включая элемент j.

2. Решение задачи

Введем обобщенные процедуры A_0 – для формирования путей в графе D□, позволяющую перечислять все объекты множества $\{L_j\}$ и $A_0^{/}$ – для определения объектов $\{L_j\}$ с интересующим нас свойством, при этом рассмотрение начнем со случая когда определяющей характеристикой объектов $\{L_j\}$ является одна весовая характеристика объекта.

2.1. Процедура A_0

Шаг 1. Формируем в графе D□ и з вершины S в множества путей $m^{r=1}_{sj}$ все возможные пути ранга r = 1 , удовлетворяющие правилу R.

Шаг 2. На основе путей текущего ранга r, формируем все возможные пути ранга r: = r + 1 в подмножествах $m^{r:=r+1}_{sj}$, удовлетворяющие правилу R.

Шаг 3. Проверяем подмножества $m^{r:=r}_{sj}$ текущего ранга r, пусты или нет, если нет, то переходим к выполнению шага 2, иначе процедура заканчивает работу, поскольку все объекты $L_j \in \Omega$ перечислены.

Процедура A_0 при переходе от произвольного ранга r к рангу r + 1 позволяет строить из объектов L_j, содержащих r базовых элементов v_j, объекты L_j, содержащие r + 1 базовый элемент v_j.

2.2. Обобщенная процедура $A_0^{/}$

Шаг 1. Из вершины S строятся пути ранга r = 1 удовлетворяющие правилам R и в соответствии с правилами R определяются их весовые характеристики.

Шаг 2. На основе путей текущего ранга r строятся все возможные пути следующего ранга r = r + 1, удовлетворяющие правилам R на основе следующего рекуррентного соотношения

$$\mu^{r+1}_{SP} = \min_{d_j(m^r_{sj} \cup (j,p))} (\max)\{m^r_{sj} \cup (j,p)\};$$

$$j = (\overline{1,n});\ p = (\overline{1,n});\ j \neq p,$$

где весовые характеристики d_l объектов определяются в соответствии с правилами $P \in R$.

Шаг 3. Проверяем подмножества m^r_{sj} текущего ранга r, пусты или нет, если нет, то переходим к выполнению шага 2, иначе процедура заканчивает работу, и из полученных локальных экстремумов на всех рангах выбирается глобальный экстремум.

В зависимости от интересующих свойств объектов и правил R может возникнуть необходимость в работе процедуры $A_0^{/}$ до достижения некоторого конкретного значения ранга r = k, например, если объектом поиска является клика максимального веса, мощность которой не превышает k. Следует также отметить, что в общем случае в подмножествах $m_{sj}^r; j = \overline{(1,n)}$ возможно появление одинаковых путей и дублирующие пути в этих подмножествах можно удалять и, кроме того, удалять их на ярусах.

Представляет интерес выяснить, в каких случаях процедура позволяет получать точное или приближенное решение задачи определения объекта L_j с требуемым свойством ν, связанным с минимизацией или максимизацией весовой характеристики объекта L_j. Для этого более подробно проанализируем процесс перехода от исходного графа G (V,E) с множеством вершин, соответствующих базовым элементам, из которых по правилам R в графе G(V,E) могут быть построены интересующие нас объекты. Особенностью отображения графа G(V,E) в D□ является то, что если анализируется конкретный объект Lj в G(V,E), то ему в графе D□ соответствует некоторый путь μ_{sj}^r, построенный на том же множестве вершин, что и объект в G(V,E), и весовая характеристика объекта Lj в G(V,E) используется для оценки длины пути μ_{sj}^r в D□, но при этом множества путей в графах D□ и G (V,E) совпадают.

Предположим, что объектами в G (V,E) являются сами пути графа G (V,E), тогда отображение графа G (V,E) есть отображением на самого себя и граф D□, в этом случае, как показано в работах [3, 4], представляет собой стянутое дерево всех путей данного графа для прочтения которых на каждой горизонтальной линейке разрешается бывать только один раз. И, следовательно, если мы в этом случае будем решать задачу определения, например, кратчайших гамильтоновых путей или кратчайших гамильтоновых циклов на основе процедуры $A_0^{/}$, то переход к графу D□ только упорядочивает процесс поиска кратчайших гамильтоновых путей в графе за счет формирования на основе рекуррентного соотношения (1), нетрудно показать, что в этом случае оптимальное решение будет точным, если процесс работы процедуры непрерывен, т.е. в процесс ее работы либо не возникает ситуаций, когда множества $m_{sj}^r = \varnothing$, либо если такая ситуация возникла, то все пути, ведущие в j в графе G(V,E) рангов r + 1, … , r = n длиннее, чем пути, полученные в j на ранге r – 1 (доказательство справедливости данного утверждения элементарно и поэтому не приводится). Однако легко видеть, что если множество m_{sj}^r оказалось пустым на некотором ранге r = q, то может оказаться, что в вершину j существует более длинный, чем построенный процедурой $A_0^{/}$ путь $\mu_{sj}^{r=q}$, но продление которого, в дальнейшем, может позволить получить более короткий гамильтонов путь. Данная ситуация фактически является камнем преткновения при решении всех NP-полных задач и более чем трехсотлетний опыт показывает, что в этом случае гарантировано получить точное решение можно получить только полным перебором или неявным полным перебором на основе метода ветвей и границ, что при достаточно большой размерности задачи одно и тоже. И так, если мы рассматриваем отображение графа G(V,E) в D□, и объектами L_j в G(V,E) являются сами пути графа, то переход к анализу на графе D□ не изменяет данную ситуацию и мы сталкиваемся с той же проблемой, что и решение данной задачи на графе G(V,E) без использования преобразований. Введем процедуру В преобразования объектов $\{L_j\}$ в графе D□ , являющуюся частным случае процедуры $A_0^{/}$, когда мы выбираем минимальные по длине пути.

2.3. Процедура В

Шаг 1. Используя правило R, формируем множество всех объектов $L_j^{r=2}$, состоящих из r = 2 базовых элементов с весами $d_j^{r=2}$, и выделяем объект L_j^{*r} с минимальным весом $d_j^{*r=2}$, если таких несколько, то выделяем их все.

Шаг 2. На основе выделенного объекта L_j^{*r} и правила R формируем все возможные объекты $L_j^{r:=r+1}$, состоящие из r: = r + 1 базовых элементов с весами $d_j^{r=r+1}$, и выделяем объект $L_j^{*r=r+1}$ с минимальным весом $d_j^{*r=r+1}$, если таких несколько то выделяем их все.

Шаг 3. Проверяем на основе L_j^{*r} и правила R можно построить объекты из r: = r + 1 базовых элементов, если нет, то процедура заканчивает работу, иначе переходим к выполнению шага 2.

В результате работы процедуры В получим объекты $L_j^{*r=2}$ $L_j^{*r=3}$ … $L_j^{*r=к}$ содержащие соответственно по 2 ,по 3 и т.д. по k базовых элементов. Нетрудно увидеть, что справедливо следующее утверждение.

Утверждение 1. Если объекты $\{L_j\}$ удовлетворяют свойству В, заключающемуся в том, что применение процедуры B к множеству $\{L_j\}$ позволяет построить объекты $L_j^{*r=2}$, $L_j^{*r=3}, \ldots, L_j^{*r=к}$, содержащие соответственно по 2, по 3 и т.д. по k базовых элементов с минимальными весами d_j^{*r1}, то процедура $A_0^{/}$ дает точное решение задачи. Утверждение справедливо, поскольку если предположить, что свойство B выполняется и на основе процедуры $A_0^{/}$ получен глобальный экстремум который не является точным решением, то это возможно если существуют объекты $L_j^{**r=2}, L_j^{**r=3}, \ldots, L_j^{**r=к}$ длины которых меньше чем у объектов $L_j^{*r=2}, L_j^{*r=3}, \ldots, L_j^{*r=к}$, что противоречит первоначальному предположению, о том, что свойство B выполняется.

Таким образом если свойство B выполняется предложенная процедура $A_0^{/}$ будет давать точное решение, в условиях непрерывности работы процедуры, а если не выполняется то приближенное, поскольку в этом случае мы фактически сталкиваемся с проблематикой NP-полных задач.

Следует также отметить, что в случае выполнения свойства B, наличие пустых множеств m_{sj}^{r} при работе процедуры $A_0^{/}$ означает, что на основе правил R построить объект $L_j^{r:=r}$, содержащий базовый элемент j или нельзя в принципе (последнее нужно обосновывать для каждой задачи в отдельности), или, что строить данный элемент не имеет смысла, поскольку его весовая характеристика будет существенно хуже тех, что построены на ярусе.

Таким образом непрерывность реализации процедуры B, является необходимым и достаточным условием для получения точного решения. Для случая, когда объекты $\{L_j\}$ характеризуются одной весовой характеристикой можно упростить процедуру $A_0^{/}$ за счет выделения на ярусах глобальных экстремумов и последующем формировании всех возможных путей следующего ранга в графе D□. Обозначим эту процедуру $A_0^{//}$.

2.4. Процедура $A_0^{//}$

Шаг 1. Из вершины S строятся пути ранга r = 1, удовлетворяющие правилам R, и в соответствии с правилами R определяются их весовые характеристики.

Шаг 2. На основе путей текущего ранга r строятся все возможные пути следующего ранга r = r + 1, удовлетворяющие правилам R, на основе следующего рекуррентного соотношения:

$$\mu_{SP}^{r:=r+1} = \min_{d_j(m_{sj}^{r} \cup (j,p))} (\max)\{m_{sj}^{r} \cup (j,p)\};$$

$$j = (\overline{1,n});\ p = (\overline{1,n});\ j \neq p,$$

где весовые характеристики dl объектов определяются в соответствии с правилами P ∈ R, среди них на ярусе выделяется путь μ_{sj}^{*r}, максимальный или минимальный на ярусе по длине, и на его основе формируем всевозможные пути ранга r: = r + 1 в графе D□.

Шаг 3. Проверяем подмножества m_{sj}^{r} текущего ранга r, пусты или нет, если нет, то переходим к выполнению шага 2, иначе процедура заканчивает работу, и из полученных на последнем ранге путей выбирается лучший (максимальный или минимальный) по длине путь.

Заключение

Начнем с процедуры A_0, представляющей процедуру перечисления всех возможных путей, которые можно породить в графе D□, используя правила R; при этом, если само правило R предполагает наличие экспоненциального числа формируемых объектов, то число путей в множествах m_{sj}^{r} будет от ранга r=1 до ранга r=n-1 возрастать экспоненциально. В случае использования процедур $A_0^{/}$ и $A_0^{//}$ возможно два варианта: первый, когда число путей во множествах растет экспоненциально, т.е. процедура $A_0^{/}$ реализуется в виде процедуры B, в которой в каждом подмножестве может выделяться несколько одинаковых экстремальных путей. Если в каждом подмножестве m_{sj}^{r} в процессе работы процедуры $A_0^{/}$ выделяется только один произвольный экстремальный путь, то число путей, которое строится процедурой на каждом ярусе, и общее их число является полиномиальным. Это вызвано тем, что на каждом ярусе в любом подмножестве m_{sj}^{r} процедура $A_0^{/}$ в случае выделения только одного экстремального пути строит не более (n-1) – пути, а на ярусе соответственно не более (n-1) n ≈ n_2 путей. Так как, число ярусов в графе D□ не может превосходить (n–1), то общее число путей, которое построит процедура $A_0^{/}$, дойдя до последнего яруса, не превысит (n-1)n (n-1)≈ n_3 путей. В случае использования процедуры $A_0^{//}$, число формируемых путей еще больше сокращается, поскольку в каждом подмножестве m_{sj}^{r} в этом случае формируется только по одному пути, на ярусе формируется по (n-1) – пути и общее число построенных путей процедурой $A_0^{//}$ по достижению яруса (n-1) не превысит числа $(n – 1)n \approx n_2$. Следует отметить, что при использовании процедур $A_0^{/}$ и $A_0^{//}$ в конкретных алгоритмах для определения временной сложности их реализации необходимо учитывать время, затрачиваемое алгоритмом на определения весовых характеристик путей и удовлетворению правилам R, определяемых свойствами формируемых объектов. Обозначим число операций, необходимых для вычисления длин путей и проверки правил R, через (p), тогда временная сложность алгоритмов на основе данных процедур не превысит соответственно $O(pn_3)$ и $O(pn_2)$. Следует отметить, что в соответствии с утверждением (1), алгоритмы, построенные на основе данных процедур, будут давать приближенное решение. Поэтому для каждой конкретной задачи должна быть дана оценка погрешности в среднем. Поскольку формирование путей на ярусах множества m_{sj}^{r} можно осуществлять одновременно, то если иметь n процессорных элементов для их формирования, то временная сложность алгоритмов на основе рассматриваемых процедур не превысит соответственно $O(pn_2)$ и O(pn).

Литература

1. *Андреев А. Кластеры и суперкомпьютеры: близнецы или братья? / А. Андреев, В. Воеводин, С. Жуматий // Открытые системы. – 2000. – № 5, 6. – С. 9-14.*
2. *Валях Е. Последовательно-параллельные вычисления. / Е. Валях. – М.: Мир, 1985. – 456 с.*
3. *Методы и модели планирования ресурсов в Grid – системах / В.С. Пономаренко, С.В. Листровой, С.В. Минухин, С.В. Знахур: Монография. – Х.: ВД «ІНЖЕК», 2008. – 408 с.*
4. *Листровой С.В. Архитектура параллельных вычислительных систем циклического типа / С.В. Листровой // Электронное моделирование. – 1992. – Т. 14, № 2. – С. 28-36.*

Поступила в редакцию 14.01.2009

Рецензент: д-р техн. наук, проф., проф. кафедры А.В. Алешин, Украинская государственная академия железнодорожного транспорта, Харьков, Украина.

ЗАГАЛЬНИЙ ПІДХІД ДО ОРГАНІЗАЦІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ПРИ РІШЕННІ ЗАДАЧ КОМБИНАТОРНОЇ ОПТИМІЗАЦІЇ ТА ТЕОРІЇ ГРАФІВ

С.В. Лістровий, О.С. Лістрова

У роботі запропонована універсальна процедура рішення довільних задач дискретної оптимізації, які можливо ефективно розпаралелити для рішення на основі циклічних паралельних обчислювальних структур. Рішення задач основані на визначенні простору можливих рішень задач дискретної оптимізації і теорії графів у вигляді стягнутого дерева всіх шляхів, що дозволяє зводити рішення задач, що розглядаємо, до визначення екстремальних шляхів в стягнутому дереві шляхів, формування яких на ярусі дерева може відбувається одночасно, що і дозволяє організувати ефективне розпаралелювання обчислювального процесу.

Ключові слова: паралельні циклічні обчислювання, стягнуте дерево шляхів графа.

GENERAL APPROACH TO ORGANIZATIONS OF THE PARALLEL CALCULATIONS AT DECISION OF THE PROBLEMS TO COMBINATORIAL OPTIMIZATION AND GRAPH THEORY

S.V. Listrjvoy , E.S. Listrovaya

In work is offered universal procedure of the decision of the free problems to discrete optimization, which can be effectively распараллелена for decision on base of the round-robin parallel computing structures. The Decision of the problems is founded on presentation space possible decisions of the problems to discrete optimization and graph theory in the manner of joined tree all ways that allows to reduce the decision of the considered problems to determination of the extreme ways in joined tree of the ways, shaping which on tier tree can be realized simultaneously, as allows to organize the efficient multisequencing of the computing process.

Key words: parallel round-robin calculations, joined tree of the ways column.

Листровой Сергей Владимирович – д-р техн. наук, проф. проф. кафедры СКС, Украинская государственная академия железнодорожного транспорта, Харьков, Украина, e-mail: om1@yandex.ru.

Листровая Елена Сергеевна – канд. техн. наук, доцент, доцент кафедры экономико-математического моделирования, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: Listravkina@gmail.com.