

УДК 519.179.2
ББК 22.176 + 65.23

АЛГОРИТМЫ УПРАВЛЕНИЯ РЕСУРСАМИ В ОБОБЩЕННЫХ СТОХАСТИЧЕСКИХ СЕТЕВЫХ ГРАФИКАХ

Иванов Н. Н.¹

(ФГБУН Институт проблем управления
им. В.А. Трапезникова РАН, Москва)

Для обобщенного стохастического сетевого графика рассматриваются три метода управления свободными ресурсами, направленных на ускорение выполнения в реальном времени моделируемого процесса управления. Оценка целесообразности и проверка эффективности предложенных методов проводится на основе дерева состояний сетевого графика, моделирующего процесс управления. Обсуждаются средства имитационного моделирования, позволяющие по дереву состояний получить сравнительные данные по каждому из предлагаемых методов.

Ключевые слова: обобщенный стохастический сетевой график, дерево состояний сетевого графика, время выполнения сетевого графика, имитационное моделирование.

1. Введение

Сетевые графики, созданные первоначально в предположении о том, что времена прохождения отдельных дуг являются детерминированными величинами, в настоящее время рассматриваются в предположениях, связанных со случайными временами прохождения дуг с известными распределениями [1]. Такие графы получили название стохастических сетевых графиков (ССГ).

В работах [2, 3] рассматривается обобщенный вариант ССГ, названный обобщенным стохастическим сетевым графиком

¹ Николай Николаевич Иванов, доктор технических наук, доцент (ivanov.nni@yandex.ru).

(ОССГ) и имеющий своим отличием допущение в качестве дисциплины, помимо стандартной, при которой событие наступает после прохождения всех дуг, входящих в соответствующую данному событию вершину (дисциплина «И»), альтернативной дисциплины «(исключенное) ИЛИ», при которой свершение события происходит в результате прохождения первой (по времени) из дуг, входящих в соответствующую ему вершину.

Применительно к процессам в управляющих параллельных вычислительных системах (ВС), структурированным с помощью ОССГ, последние могут рассматриваться как модели, на которых осуществляется контроль и обеспечение временной надежности управляющих программных комплексов. Это предполагает, что для заданных распределений времени выполнения отдельных работ в каждом таком комплексе предварительно на основе поведения моделирующего ОССГ может быть сделан вывод о возможности выполнения управляющей программы не более чем за заданное директивное время [4].

Для выполнения требования временной надежности ВС могут быть привлечены различные алгоритмы использования свободных вычислительных ресурсов для ускорения выполнения управляющей программы в случае, если эта совокупность выбрана исходя из понятия степени параллелизма ОССГ. Напомним, степень параллелизма определяется как верхняя оценка такого числа исполнителей, при котором отсутствие очередей сочетается с минимизацией простоев исполнителей [3].

Необходимость использования свободных вычислительных ресурсов определяется пользователем, а их целесообразность и эффективность может быть подтверждена обсуждаемыми в работе способами и соответствующими этим способам инструментами имитационного моделирования.

2. Алгоритмы управления временем выполнения ОССГ, основанные на использовании свободных ресурсов

В процессе выполнения ОССГ могут возникать моменты, когда исполнительные устройства (например, каналы ВС), выбранные исходя из степени параллелизма ОССГ [3] или по ка-

ким-либо иным критериям, не будут задействованы полностью. При этом возникает возможность использования свободных ресурсов в целях ускорения процесса выполнения ОССГ. Ресурс, использованный при прохождении некоторой дуги, считается свободным в момент окончания ее прохождения.

Алгоритмы управления, нацеленные на ускорение выполнения ОССГ, могут быть построены с учетом того, могут ли свободные ресурсы использоваться с разделением на части или их использование не предполагает такой возможности, а в последнем случае включение в работу свободного ресурса допускается только в момент начала прохождения дуги и не предусмотрена возможность его изъятия до окончания ее прохождения, либо неделимый свободный ресурс может быть включен в работу независимо от того, какая доля времени уже затрачена на прохождение дуги.

2.1. АЛГОРИТМЫ С ДЕЛЕНИЕМ РЕСУРСОВ В МОМЕНТЫ ОКОНЧАНИЯ ПРОХОЖДЕНИЯ ДУГ

Эти алгоритмы предусматривают неизменное число исполнителей для каждой дуги в любой момент выполнения ОССГ до момента самого раннего окончания прохождения какой-либо дуги. При этом для прохождения любой дуги используется не менее одного исполнителя, а общее их число определяется либо по степени параллелизма ОССГ [3], либо выбор числа исполнителей проведен в результате предварительного имитационного моделирования дерева состояний [3] при заданных распределениях времени прохождения дуг. Ниже для этого числа принято обозначение N .

Алгоритм I, распределяющий ресурсы в равных долях по всем одновременно проходимым дугам. Алгоритм предполагает, что для каждой дуги при ее прохождении может быть использовано разное (возможно, нецелое) число (но не менее одного) исполнителей, при этом в любой момент времени сумма этих чисел, одинаковых для любой дуги в любой момент времени, по всем одновременно проходимым дугам строго равна степени параллелизма N . После завершения прохождения некоторой дуги производится пересчет чисел исполнителей для всех проходимых дуг в соответствии с равенством $k = N/n$, где

k – число исполнителей, назначаемых для каждой дуги; n – число одновременно проходимых в рассматриваемый момент времени дуг.

Алгоритм II, распределяющий ресурсы в равных целых долях по всем одновременно проходимым дугам. Алгоритм предполагает, что для каждой дуги при ее прохождении может быть использовано разное целое число исполнителей, но при этом в любой момент времени сумма этих чисел, одинаковых для любой дуги в любой момент времени, по всем одновременно проходимым дугам не превосходит N . После завершения прохождения некоторой дуги производится пересчет чисел исполнителей для всех проходимых дуг в соответствии с равенством $k = \lfloor N/n \rfloor$, где k – целое число исполнителей, назначаемых для каждой дуги; n – число одновременно проходимых в рассматриваемый момент времени дуг; N – степень параллелизма ОССГ.

2.2. АЛГОРИТМ III С ДЕЛЕНИЕМ РЕСУРСОВ В МОМЕНТЫ СВЕРШЕНИЯ СОБЫТИЙ

При использовании этого алгоритма пересчет числа подключаемых резервных ресурсов происходит в более редкие моменты времени, а именно, только в моменты свершения событий в ОССГ. При этом дуги, которые еще не закончили свое прохождение в момент свершения события, продолжают прохождение с тем числом исполнителей, которое было на момент свершения события. После свершения события накапливается некоторое количество свободных ресурсов, которые распределяются в равных долях (но не менее одного исполнителя для каждой дуги) между теми дугами, которые инициализировались в результате свершения события. При таком распределении ресурсов каждая дуга с момента своей инициализации и до момента окончания прохождения проходит с фиксированным числом исполнителей.

В рассматриваемом алгоритме должны соблюдаться следующие правила: 1) число назначаемых исполнителей для каждой дуги должно быть не менее единицы; 2) все свободные исполнители распределяются по инициализируемым дугам поровну в целых числах, в максимальном количестве в соответствии с

накопленным ресурсом; 3) минимальное общее число исполнителей ОССГ выбирается так, чтобы ни для какого набора случайных времен прохождения дуг не существовало событий, для которых недостаточно числа свободных исполнителей для назначения по одному исполнителю для каждой инициализируемой дуги.

В этом алгоритме максимальное число исполнителей может превзойти степень параллелизма. Каково будет это превышение, может показать описываемое ниже имитационное моделирование этого алгоритма.

Приведенными алгоритмами не исчерпываются все возможные варианты методов управления ресурсами в ОССГ. Разработчик может использовать другие алгоритмы, более подходящие для исследуемой системы, исходя из ее возможностей и особенностей. Однако при этом он может воспользоваться теми способами моделирования алгоритмов управления ресурсами, которые, по мнению автора, являются достаточно универсальными.

3. Имитационное моделирование алгоритмов управления ресурсами

Предпочтение, которое может отдать пользователь тому или иному алгоритму управления процессом выполнения ОССГ с помощью привлечения свободных ресурсов, может быть основано на их имитационном моделировании. Для этих целей наиболее эффективным является, по мнению автора, построения траекторий в дереве состояний ОССГ [3] для каждого случайного набора времен прохождения дуг с включением алгоритмов управления ресурсами в механизм прохождения траекторий в дереве состояний.

Напомним, что состоянием ОССГ считается вектор, составленный из номеров всех дуг, проходимых в текущий момент времени. Каждая дуга дерева состояний помечена номером дуги сетевого графика, исходящей из состояния дерева, в котором данная дуга объявлена в качестве составляющей вектора состояния. Начальным состоянием дерева состояний назначается век-

тор из номеров дуг, исходящих из начальной вершины сетевого графика. Конечному состоянию соответствует пустой вектор.

Алгоритм построения дерева состояний состоит в рекурсивном пополнении состояний дерева, начиная с начального. Если предположить, что построено состояние s k -го уровня этого дерева, то для построения состояний $(k + 1)$ -го уровня, непосредственно связанных с состоянием s , нужно из s вывести дуги числом, соответствующим числу дуг сетевого графика, входящих в s . Каждая такая дуга u , помеченная номером выбранной дуги сетевого графика, ведет в некоторое состояние q , которое содержит дуги, входящие в s , за исключением u , и пополненное дугами, активируемыми в случае, если прохождение дуги u вызвало свершение некоторого события. Построение дерева состояний в лексикографическом порядке может вестись так, что если состояние, соответствующее рассматриваемой вершине, в процессе построения дерева уже было достигнуто ранее, то оно подчеркивается, и построение ее продолжения обрывается. В этом случае результатом такого построения будет укороченное дерево состояний. Если же процедура подчеркивания не используется, то результатом построения будет полное дерево состояний.

В качестве примера для проверки предложенных выше алгоритмов управления методом имитационного моделирования был выбран ОССГ, представленный на рис. 1. Нормальные независимые распределения времен прохождения дуг и их обозначения принимались в соответствии с данными, приведенными ниже.

$$m_{01} = m_{12} = m_{23} = 8, m_{02} = 5,5\sqrt{2}, m_{13} = 6, m_{14} = 6\sqrt{2},$$

$$m_{24} = 10\sqrt{2}, m_{34} = 7, m_{35} = 3, m_{45} = 9;$$

$$d_{01} = d_{14} = d_{02} = d_{12} = d_{13} = d_{23} = d_{24} = d_{34} = d_{35} = d_{45} = 1.$$

Было принято следующее предположение: скорость прохождения каждой дуги пропорциональна используемому ресурсу: $v_k = kv_1$, где k – число исполнителей, привлеченных для прохождения дуги; v_1 – скорость прохождения при одном исполнителе. Таким образом, при $k > 1$ время прохождения дуги, определенное для $k = 1$, уменьшается в k раз. Возможны и другие

зависимости времени прохождения дуги от числа исполнителей, определяемые пользователем.

Рассматриваемый ОССГ, как это показано в [3], имеет степень параллелизма, равную 4. Там же показано, что при выбранных нами распределениях времен прохождения дуг эта степень параллелизма достижима.

По этой причине в статистических экспериментах выбиралось 4 исполнителя. Однако при экспериментах с алгоритмом III было выявлено, что для отсутствия очередей необходимо наличие 5 исполнителей.

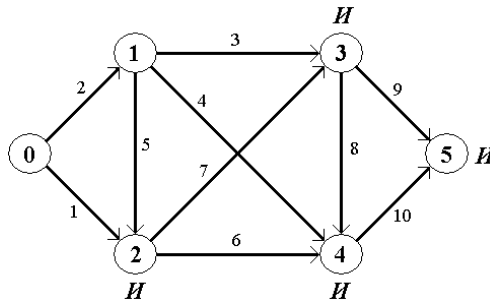


Рис. 1. Пример ОССГ

Моделирующие программы строились на основании двух принципов. Первоначально рассматривалась простая возможность построения программ, основанная на определении траекторий в укороченном дереве состояний при заданном случайном векторе времен прохождения дуг. Этот путь требовал предварительного построения укороченного дерева состояний, что может осуществляться машинным методом. Для стандартного режима без управления и алгоритмов I–II возможно было ограничиться построением укороченного дерева состояний, представленного на рис. 2.

Заметим, что дерево состояний ОССГ, в котором все вершины типа «И», имеет высоту, равную числу дуг сетевого графика. Такую же высоту будет иметь дерево состояний при наличии вершин типа «ИЛИ» без блокирования входных для них дуг [3]. При наличии вершин типа «ИЛИ» с блокированием дуг

(с остановкой прохождения всех входящих в эту вершину дуг после прихода первой по времени дуги) высота дерева состояний станет меньше на величину, равную суммарному числу входных дуг в вершины типа «ИЛИ» за вычетом числа таких вершин. Что касается ширины (полного) дерева состояний, то в общем случае ее можно оценить с помощью общего числа путей.

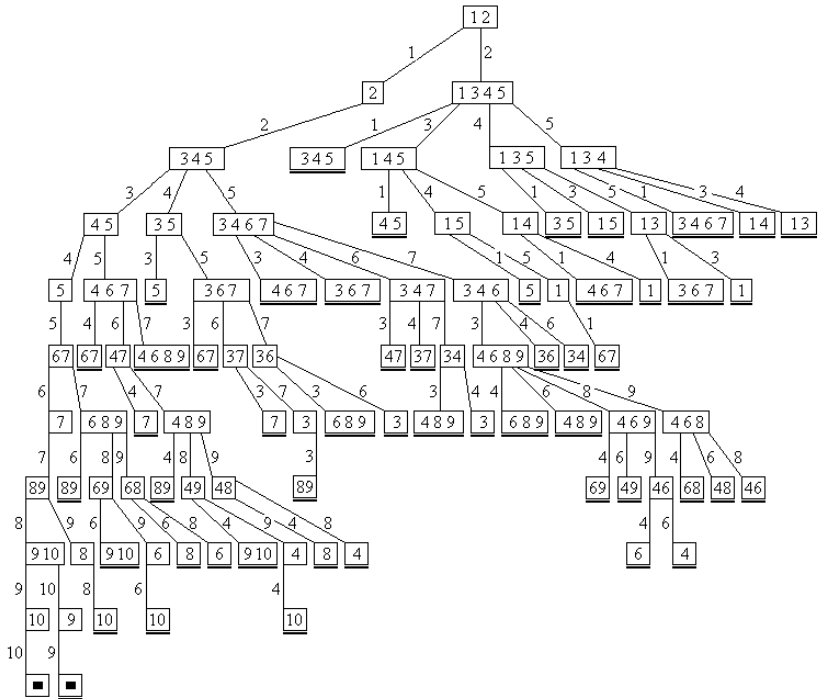


Рис. 2. Укороченное дерево состояний ОССГ по рис. 1

Однако для моделирования алгоритма III, как показано ниже, требовалось построение полного дерева состояний, что существенно увеличивало объем программы. Поэтому для алгоритма III этот метод моделирования не использовался.

По этой причине для моделирования всех алгоритмов был рассмотрен способ программирования, основанный на построе-

нии траекторий в полном дереве состояний без построения самого дерева. Такой метод предполагал построение траектории в дереве состояний для заданного набора случайных времен прохождения дуг непосредственно по графу ОССГ.

На рис. 3, *a* показан фрагмент сетевого графика, в котором события, соответствующие верхнему ряду вершин, свершились и проходятся дуги *a*, *b*, *c*, *d*. Степень параллелизма этого ОССГ равна 5. Вершины 1 и 2 имеют тип «И».

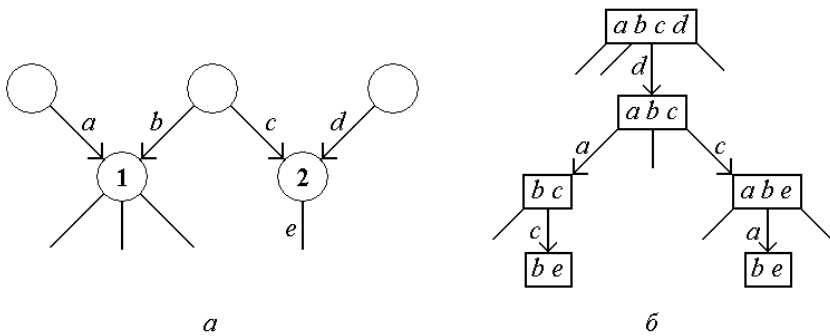


Рис. 3. Фрагмент ОССГ (а) и соответствующий ему фрагмент дерева состояний (б)

Если первой пройдена дуга *d*, то из состояния *abcd* на дереве состояний, фрагмент которого показан на рис. 3, б, произойдет переход в состояние *abc*. Если дуга *a* будет пройдена раньше дуг *b* и *c*, то произойдет переход в состояние *bc*, если же дуга *c* будет пройдена раньше дуг *a* и *b*, то произойдет переход в состояние *abe* и свершится событие 2. Далее из состояния *bc* по дуге *c* произойдет переход в состояние *be*, а из состояния *abe* по дуге *a* также в состояние *be*. Однако эти два состояния на дереве состояний могут обладать разными числами исполнителей для дуг *b* и *e*. Это окажется возможным, например, если дуги *a*, *b*, *c* и *d* проходились с помощью, соответственно, 1, 2, 1 и 1 исполнителей. Тогда, как нетрудно проследить, в левом состоянии *be* дуги *b* и *e* проходятся с помощью 2 и 3 исполнителей соответственно, а в правом состоянии *be* эти же дуги проходятся с по-

мощью 2 исполнителей каждая, что говорит о невозможности слияния этих двух одноименных состояний.

Второй способ моделирования имел преимущество перед способом, основанном на построении укороченного дерева состояний, определяемое практически независимым от размерности ОССГ объемом моделирующей программы.

Для верификации обоих методов было произведено сравнение результатов счета, получаемых для алгоритмов I–II обоими способами, показавшее их полное совпадение.

Результаты статистических испытаний четырех режимов выполнения ОССГ, в число которых входил стандартный режим без управления и три рассмотренных в п. 2 алгоритма I–III, приведены в таблице 1.

В таблице приняты следующие обозначения: m – оценка среднего времени выполнения ОССГ; σ – среднеквадратическое отклонение этого времени; Δ – ширина интервала доверия при уровне значимости $\alpha = 0,05$ и числе испытаний 10^5 .

Таблица 1.

Параметр	Стандарт	Алгоритмы управления		
		I	II	III
m	40,34	19,85	22,17	26,91
σ	2,04	0,78	1,02	2,44
Δ	$2,5 \cdot 10^{-2}$	10^{-2}	$1,3 \cdot 10^{-2}$	$3 \cdot 10^{-2}$

Как видно из таблицы 1, алгоритм I обеспечивает почти двукратное уменьшение среднего времени выполнения ОССГ. Однако дробление исполнительных ресурсов не всегда возможно. Менее эффективными в порядке убывания являются алгоритмы II и III, в которых дробление не используется. Однако относительная недостаточная эффективность компенсируется простотой подключения и использования ресурсов во время прохождения дуг.

Для ОССГ, показанного на рис. 1, но с вершинами 2 и 4 типа «ИЛИ» с блокированием входных для этих состояний дуг, укороченное дерево состояний имело вид, приведенный на рис. 4. По этому дереву было проведено моделирование алгоритмов I и II при тех же параметрах случайных времен прохождения дуг. В данном случае высота дерева была равна 7, а число возможных состояний было равно 28 (44 состояния при программировании по дереву с рис. 2), что обеспечивало более благоприятные условия для применения этого метода моделирования.

Результаты моделирования приведены ниже.

Стандарт: $m = 25,5, \sigma = 1,7, \Delta = 2,1 \cdot 10^{-2}$.

Алгоритм I: $m = 15,1, \sigma = 0,94, \Delta = 1,2 \cdot 10^{-2}$.

Алгоритм II: $m = 15,7, \sigma = 1,07, \Delta = 1,3 \cdot 10^{-2}$.

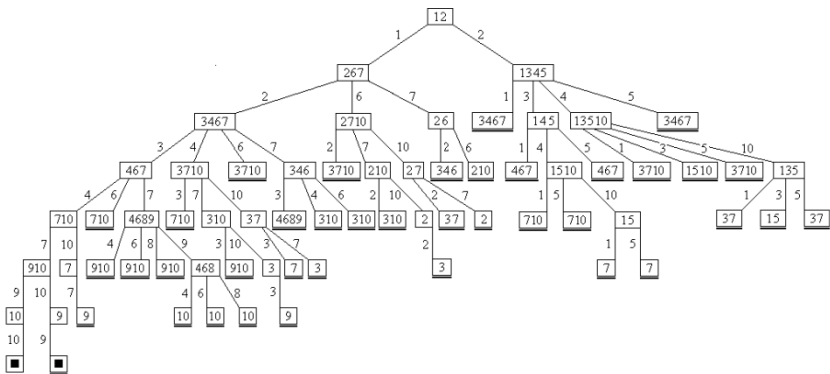


Рис. 4. Укороченное дерево состояний ОССГ со смешанными типами вершин и с блокированием дуг

4. Заключение

Предложен ряд алгоритмов привлечения резервных исполнительных ресурсов с целью ускорения прохождения отдельных дуг и как следствие ускорения процесса выполнения ОССГ. Для этих алгоритмов в статье предложен инструмент их анализа с помощью дерева состояний ОССГ. Основываясь на рассмотренных в статье результатах, пользователь может разрабатывать иные алгоритмы управления и проводить проверку их эффективности имитационными средствами предложенными способами.

Литература

1. ГОЛЕНКО-ГИНЗБУРГ Д.И. *Стохастические сетевые модели планирования и управления разработками*. – Воронеж: Научная мысль, 2010. – 283 с.
2. ИВАНОВ Н.Н. *Аналитико-имитационное моделирование обобщенных стохастических сетевых графиков // Управление большими системами*. – 2015. – Вып. 53. – С. 27–44.
3. ИВАНОВ Н.Н. *Степень параллелизма обобщенных стохастических сетевых графиков // Управление большими системами*. – 2017. – Вып. 65. – С. 6–23.
4. ИГНАТУЩЕНКО В.В., ИСАЕВА Н.А. *Резервирование взаимосвязанных программных модулей для управляющих параллельных вычислительных систем: организация, оценка отказоустойчивости, формализованное описание // Автоматика и телемеханика*. – 2008. – №10. – С. 142–161.

ALGORITHMS OF RESOURCE MANAGEMENT IN GENERALIZED STOCHASTIC NETWORKS

Nikolay Ivanov, Institute of Control Sciences of RAS, Moscow,
Doctor of Science (Moscow, Profsoyuznaya st., 65,
ivanov.nni@yandex.ru).

Abstract: Generalized stochastic network contains vertices corresponding to events, the fulfillment of which can occur in the discipline "AND" and "OR". These networks can be used for simulation of real-time control processes using parallel computing systems. The weight of edges in these networks are considered random variables distributed according to given laws. The main purpose of modeling of such networks is monitoring and ensuring of the specified directive execution time of the management process. The time management of the network schedule can be managed by utilization of free resources (processors, channels of parallel computing systems, etc.). For a generalized stochastic network, three methods of managing free resources are considered, aimed at accelerating the execution of a simulated control process in real time. Evaluation of the feasibility and verification of the effectiveness of the proposed methods is carried out on the basis of the state tree of the network modeling the management process. The state tree defines those states of the network process that require fewer resources than what is called the degree of parallelism. The degree of parallelism determines the maximum number of resources that ensures the absence of queues. Simulation tools are discussed that allow you to obtain comparative data on each of the proposed methods in the tree of states.

Keywords: the generalized stochastic network, the state tree of the network, the execution time of the network, simulation.

*Статья представлена к публикации
членом редакционной коллегии А.А. Лазаревым.*

Поступила в редакцию 15.06.2017.

Опубликована 31.05.2018.