

# ОБОБЩЕННЫЕ СТОХАСТИЧЕСКИЕ СЕТЕВЫЕ ГРАФИКИ С НЕСТАНДАРТНЫМИ ДИСЦИПЛИНАМИ СВЕРШЕНИЯ СОБЫТИЙ

Н.Н. Иванов

Понятие обобщенного сетевого графика расширено допущением включения в эти графики вершин с нестандартными дисциплинами свершения событий в целях предоставления этим вершинам возможностей управлять различного рода динамическими процессами, протекающими в сетевых графиках. Рассмотрены общие принципы конструктивного описания сетевых графиков с подобными вершинами, которое может служить основой для проведения их статистического анализа.

**Ключевые слова:** обобщенный стохастический сетевой график, критический путь, дисциплина свершения события, время выполнения сетевого графика.

## ВВЕДЕНИЕ

Операционные системы вычислительных систем, управляющих техническими объектами в реальном времени (ОСРВ), помимо чисто синхронизационных функций, сводящихся к реализации сборок по «И» и «ИЛИ» для программных модулей управляющих комплексов, могут выполнять и другие функции управления, в первую очередь, обеспечивающие выполнение задач за время, не превосходящее директивный порог. Требуют решения с помощью ОСРВ и другие задачи управления, которые ставятся в соответствии с техническим регламентом. Такие функции по большей части не могут моделироваться обобщенными стохастическими сетевыми графиками (ОССГ) в силу ограниченности арсенала средств управления у последних. Шаг в направлении пополнения этого арсенала сделан в работе [1], в которой в ОССГ, моделирующей систему реального времени, допускалось введение управляющих дуг. Активация этих дуг может влиять на время выполнения моделирующего ОССГ. Введение таких дуг и подбор их параметров позволяет пользователю на стадии проектирования систем реального времени оценить их роль в обеспечении временной надежности этих систем.

В настоящей работе упор делается на введение в ОССГ особых вершин, которые также могут стать средством расширения моделирующих возможностей ОССГ и решения с их помощью задач, которые интересны пользователям и не могут быть

решены с помощью традиционных моделей, имеющих в своей основе стандартные сетевые графики.

Цель работы — показать на ряде примеров возможности создания особых вершин ОССГ с нестандартными дисциплинами свершения событий, ассоциированных с этими вершинами, и дать общее конструктивное описание алгоритма, позволяющего для заданного набора случайных времен прохождения дуг находить критический путь и время его прохождения.

Этот алгоритм в дальнейшем может быть положен в основу программ, на которых строится статистический метод анализа ОССГ с подобными особыми вершинами, который может проводиться при заданных законах распределения случайных времен прохождения дуг и позволяющий оптимизировать параметры графиков в целях достижения необходимых показателей моделируемых систем реального времени.

На ряде примеров проиллюстрированы возможности предлагаемых в работе приемов анализа ОССГ со специальными вершинами. При этом не ставилась цель представления всех вариантов таких вершин. Пользователь, применяя описанный подход, может создавать иные типы особых вершин с требуемой дисциплиной свершения событий, ассоциированных с ними, и методом статистических испытаний их оптимизировать.

## 1. ПРИМЕРЫ ОБОБЩЕННЫХ СТОХАСТИЧЕСКИХ СЕТЕВЫХ ГРАФИКОВ С ОСОБЫМИ ВЕРШИНАМИ

Привлечение сетевых графиков в качестве инструмента анализа поведения систем реального



времени обусловлено их чрезвычайной простотой (в сравнении, например, с сетями Петри) и наглядностью. Однако в существующем виде ОССГ представляют собой лишь примитивную основу для моделирования управляющих воздействий ОСРВ и не могут отразить все многообразие этих воздействий. Отсюда актуальна задача расширения изобразительных возможностей ОССГ с целью их приближения к механизмам функционирования ОСРВ. В силу многообразия этих механизмов не представляется возможным создать такую версию ОССГ, которая могла бы привлекаться для моделирования в каждом встречающемся на практике случае. Речь должна идти об универсальном методе включения в ОССГ таких дополнительных возможностей, которые могли бы разрабатываться пользователем самостоятельно применительно к каждому конкретному варианту ОСРВ.

В работе рассматривается путь расширения функциональных возможностей ОССГ, опирающийся на поведенческие свойства входных дуг для некоторых особых вершин. Времена прихода входных дуг и, следовательно, времена окончания действия некоторых управляющих команд и ответных реакций на них со стороны управляемого объекта могут служить входной информацией для ОСРВ и, соответственно, служить исходной информацией для ОССГ, моделирующего процесс управления.

Далее рассматривается ряд примеров специальных вершин с нестандартными дисциплинами свершения событий, которые могут встретиться в ОССГ, моделирующих процессы управления в системах реального времени. Отметим, что свершение событий, ассоциированных с этими вершинами, может происходить только на основе текущего состояния временных процессов прохождения входных дуг.

### 1.1. Вершины типа RUNTIME

В ОССГ могут быть включены особые вершины, предназначенные для контроля времени прихода в эти вершины с целью внесения корректив в динамику выполнения графика. В работе [1] этой цели служили управляющие дуги, активация которых изменяла условия свершения событий, ассоциированных с вершинами, в которые они входили, на такие, при которых происходило свершение этих событий в более ранние сроки по сравнению со стандартными. В настоящей работе предлагается иной тип воздействия на динамику выполнения ОССГ, связанный с изменением времени прохождения дуг, исходящих из особой вершины. Это изменение в сторону уменьшения может быть вызвано ОСРВ алгоритмическим способом с помощью разных средств (уменьшения учитываемых пара-

метров, понижения точности счета, уменьшения числа итераций и др.).

Вершины типа RUNTIME могут быть как типа «И», так и типа «ИЛИ». Управляющее воздействие активируется в том и только в том случае, если время свершения события по заданной дисциплине («И»/«ИЛИ») превосходит заданную константу ( $\tau$  — runtime constant).

### 1.2. Вершины типа SEQUENCE

Для этих вершин предполагается, что во время выполнения ОССГ приход входных дуг в некоторую вершину должен происходить в заранее установленном порядке. Соответственно этому порядку упорядочены входящие дуги в вершину типа SEQUENCE. Если этот порядок нарушается, что должно происходить с вероятностью, не превышающей заданную величину, то выполнение ОССГ каким-то образом корректируется (например, прерывается). Эта вероятность может быть определена в ходе статистических испытаний, и на их основе могут быть внесены корректирующие изменения во времена прохождения отдельных дуг.

### 1.3. Вершины типа $I_k$

Дисциплиной свершения событий — как бы промежуточной между дисциплинами «И» и «ИЛИ» — может служить описываемая далее дисциплина « $I_k$ ». Событие, происходящее по этой дисциплине, наступает после прихода по времени первых  $k$  дуг в соответствующую данному событию вершину, где  $1 < k < n$ ,  $n$  — число входящих в эту вершину дуг.

### 1.4. Вершины типа PRIORITY

Вершина типа PRIORITY вводится в тех случаях, когда свершение каких-либо двух событий 1 и 2 (им соответствуют вершины  $u_1$  и  $u_2$ , не связанные отношением следования  $\psi$ , при этом  $u_2$  — особая вершина) не может произойти одновременно в пределах некоторого назначенного промежутка времени. При этом событию 1 может быть отдан приоритет так, что в случае его свершения событие 2 не может свершиться ранее, чем через определенный промежуток времени  $\tau$  после свершения события 1. Что касается события 2, то его свершение может состояться беспрепятственно в случае, оговоренном выше, а также в том случае, если событие 1 еще не свершилось.

Приведенное определение относится к случаю одностороннего приоритета. При необходимости оно может быть применено и по отношению к двустороннему приоритету, для которого событию 2 может быть отдан приоритет так, что в случае его свершения событие 1 не может свершиться

ранее, чем через определенный промежуток времени  $\sigma$  после свершения события 2.

Вершины, по отношению к которым может быть введен приоритет, могут быть типа «И» или «ИЛИ», но в случае возникновения условий, связанных с введением приоритета, события в них свершаются с временной задержкой.

## 2. АЛГОРИТМЫ АНАЛИЗА ОБОБЩЕННЫХ СТОХАСТИЧЕСКИХ СЕТЕВЫХ ГРАФИКОВ С ОСОБЫМИ ВЕРШИНАМИ

Конструктивное описание алгоритмов построения критических путей и времени их прохождения в ОССГ с нестандартными дисциплинами свершения событий, рассматриваемое ниже, основывается на применении псевдокодов, созвучных с языками программирования C, Pascal и Java [2]. Названия алгоритмов, соответствующих ОССГ с особыми вершинами, определяются названиями этих специальных вершин.

Далее приняты следующие общие обозначения для всех приводимых в работе алгоритмов:  $V[G]$  — множество вершин ОССГ;  $(v, u)$  — дуга, соединяющая вершины  $v$  и  $u$  в указанном порядке;  $Adj[u]$  — множество всех непосредственных последователей вершины  $u$ ;  $d[v]$  — время прихода в вершину  $v$ ;  $\pi[v]$  — непосредственный предшественник вершины  $v$  такой, что дуга  $(\pi[v], v)$  вызвала изменение времени  $d[v]$ ;  $w(v, u)$  — время прохождения дуги  $(v, u)$ ;  $s$  — начальная вершина,  $r$  — конечная вершина ОССГ.

В основе алгоритмов лежит топологическая сортировка множества  $V[G]$ , которая заключается во введении на этом множестве отношения  $\phi$  линейного порядка, покрывающего частичный порядок  $\psi$ , определяемый ОССГ,  $u\psi v \Rightarrow u\phi v$ ,  $u, v \in V[G]$ . Если разместить все вершины графика на прямой в соответствии с линейным порядком  $\phi$ , то все стрелки сетевого графика окажутся направленными слева направо. Топологическая сортировка в указанном смысле будет обозначаться в тексте алгоритмов как процедура SORTING. Процедура сортировки достаточно подробно описана в работе [2]. Все описываемые далее алгоритмы начинаются с вызова этой процедуры.

Каждый алгоритм последовательно просматривает вершины графика в соответствии с линейным порядком  $\phi$  так, что при просмотре все цепочки дуг, ведущие из начального состояния в просматриваемое, пройдены. Этим обеспечивается выполнение условий для свершения событий, ассоциированных с этими вершинами, при этом алгоритм располагает временем прихода последней из входных дуг (для вершин типа «И») или первой по времени дуги (для вершин типа «ИЛИ»).

Следующая процедура, вызов которой требуется во всех алгоритмах, это процедура инициализации, которая будет обозначаться как INITIALIZE. Стандартный вид процедуры:

```
INITIALIZE
1  $d[s] \leftarrow 0$ 
2 for  $\forall v \in V[G]$  do
3   if  $v$  типа «И»
4     then  $d[v] \leftarrow 0$ 
5   if  $v$  типа «ИЛИ»
6     then  $d[v] \leftarrow \infty$ 
7    $\pi[v] \leftarrow \text{NIL}$ .
```

После присвоения всех начальных значений переменным, задействованным в каждом алгоритме, начинается просмотр всех вершин  $V[G]$  основной программой с привлечением следующих стандартных процедур.

Если просматриваемая вершина особая, то для нее в первую очередь должна быть вызвана процедура CONTROL. Эта процедура уникальна для каждого типа особых вершин.

Следующие две процедуры применяются для не особых вершин, однако имеются исключения, например, для вершин типа RUNTIME и PRIORITY (см. выше § 1).

RELAX ( $u, v, w$ ) (применяется для вершин типа «ИЛИ»)

```
1 if  $d[v] > d[u] + w(v, u)$ 
2   then  $d[v] \leftarrow d[u] + w(v, u)$ 
3    $\pi[v] \leftarrow u$ .
```

INCREASE ( $u, v, w$ ) (применяется для вершин типа «И»; вершины, имеющие одну входную дугу, отнесены к вершинам типа «И»)

```
1 if  $d[v] < d[u] + w(v, u)$ 
2   then  $d[v] \leftarrow d[u] + w(v, u)$ 
3    $\pi[v] \leftarrow u$ .
```

Если  $\pi[v] = \text{NIL}$ , то предшественник не определен, если же  $\pi[v] \neq \text{NIL}$ , то обращенная цепочка  $r, \pi[r], \pi[\pi[r]], \dots, s$  будет представлять собой критический путь, при этом  $d[r]$  — время прохождения критического пути.

Теперь алгоритм может быть представлен в виде следующей последовательности процедур.

### DAG\_LONGEST\_PATHS

```
1 SORTING
2 INITIALIZE
3 for всех  $u \in V[G]$  (в установленном линейном порядке  $\phi$ ) do
4   if  $u$  — особая вершина
5     then CONTROL
6   for  $v \in Adj[u]$  (для всех последователей вершины  $u$ ) do
7     if  $v$  имеет тип «ИЛИ»
8       then RELAX ( $u, v, w$ )
9     if  $v$  имеет тип «И»
10      then INCREASE ( $u, v, w$ )
11 Время выполнения сетевого графика  $T \leftarrow d[r]$ .
```



Все описываемые алгоритмы используют в качестве входной информации набор случайных времен прохождения дуг, получаемых с помощью генераторов случайных чисел с заданным распределением. Выбор вида этого распределения не принципиален и определяется пользователем. Предпочтительнее ограниченное нормально распределение, поскольку генерация нормально распределенных случайных векторов весьма проста и допускает зависимость между компонентами в соответствии с корреляционной матрицей [3].

Для рассмотренных выше четырех типов особых вершин процедуры CONTROL могут быть записаны в следующей форме.

В алгоритме RUNTIME процедура CONTROL имеет вид:

CONTROL ( $u, \tau$ ) ( $u$  — особая вершина,  $\tau$  — runtime constant)

```

1 if  $d[u] > \tau$ 
2   then for всех  $z \in Adj[u]$ 
3     do  $w(u, z) \leftarrow w(u, z)/k[u]$ .
```

**Примечание.** Постоянная  $k[u]$  может быть назначена произвольно,  $k[u] > 1$ . В данном случае  $k[u]$  постоянная величина для всех исходящих из вершины  $u$  дуг, но в общем случае ее значение может быть выбрано для каждой исходящей из вершины  $u$  дуги.

Процедура управления особой вершиной в алгоритме SEQUENCE:

CONTROL ( $u, d_i[u]$ )

```

1 if  $d_n[u] < \dots < d_1[u]$ 
2   then  $d[u] \leftarrow d_1[u]$ 
3      $\pi[u] \leftarrow z_1$ 
4 else  $\pi[u] \leftarrow NIL$ 
5    $Adj[u] \leftarrow \{r\}$  (выполнение ОССГ остановлено)
6    $d[r]$  не определен (критический путь не существует)
```

**Примечание.**  $d_i[u]$  — время прихода входной дуги ( $z_i, u$ ) в вершину  $u$ ,  $d_i[u] = d[z_i] + w(z_i, u)$ .

В этом алгоритме может контролироваться порядок прихода некоторого собственного подмножества дуг, входящих в особую вершину.

В алгоритме «И<sub>к</sub>» процедура CONTROL принимает вид:

CONTROL ( $u, d_i[u], k$ )

```

1 сортировка по возрастанию величин  $d_i[u]$ 
  ( $d_i[u]$  имеет тот же смысл, что и в алгоритме SEQUENCE,  $1 < k < n$ )
2  $d[u] \leftarrow d_k[u]$ 
3  $\pi[u] \leftarrow u_k$ .
```

Алгоритм PRIORITY в случае одностороннего приоритета содержит в себе следующую процедуру (сортировка вершин при этом предполагает, что  $u_1 < u_2$ ):

CONTROL ( $u, d[u_1], d[u_2], \tau$ )

```

1 if  $d[u_1] < d[u_2] < d[u_1] + \tau$ 
2   then  $d[u_2] \leftarrow d[u_1] + \tau$ 
3      $\pi[u_2] \leftarrow u_1$ .
```

Отношения приоритета между вершинами  $u_1, u_2$  и  $u_2, u_1$  являются взаимоисключающими. Поэтому для двустороннего приоритета потребуется выполнение программы анализа при одном и том же наборе случайных времен прохождения дуг ОССГ для двух топологических сортировок вершин  $V[G]$ , у которых  $u_1 < u_2$  и  $u_2 < u_1$ . Критическим путем следует назначить наидлиннейший путь из двух получаемых при этих сортировках.

### 3. ПРИМЕР

Статистические испытания ОССГ, приведенного на рисунке, проводились программно в соответствии с алгоритмами моделирования, описанными в § 1 и 2. Эти испытания проводились для независимых нормальных распределений при уровнях значимости, определяемых «правилом трех сигм». Далее величины  $m_{ij}$  и  $d_{ij}$  представляют собой математические ожидания и дисперсии времен прохождения дуг, соединяющих вершины с номерами  $i$  и  $j$ :

$$m_{01} = 5, \quad m_{02} = 3\sqrt{2}, \quad m_{12} = m_{23} = 9, \quad m_{13} = m_{34} = 6,$$

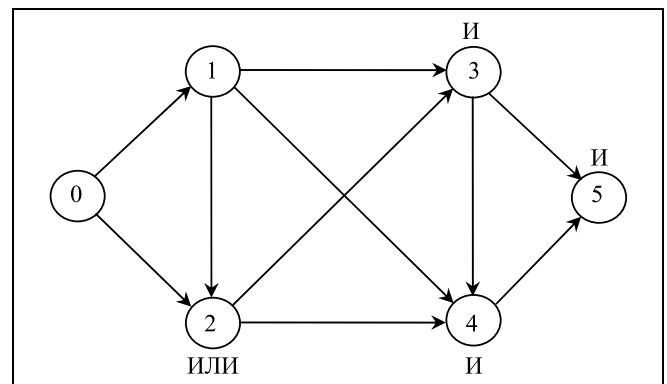
$$m_{14} = m_{24} = 6\sqrt{2}, \quad m_{35} = m_{45} = 3,$$

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

Времена прохождения дуг моделировались по методике, описанной в работе [3], с помощью двумерных изотропных векторов.

Во всех режимах в качестве особой вершины рассматривалась вершина 4. В режиме ТЕСТ никаких ограничений на дисциплины свершения событий в вершине 4 не накладывалось.

Временной параметр  $RT$  вершины 4 в режиме RUNTIME принимал значение, равное 24 (время свершения события, соответствующего вершине 4, не должно было превышать 24, при этом в случае невыполнения



Пример обобщенного стохастического сетевого графика

этого требования время прохождения дуги 45 уменьшалось в два раза).

Режим SEQUENCE происходил без останова выполнения программы для двух (из шести возможных) допустимых последовательностей прихода дуг в вершину 4: {14, 24, 34} и {24, 14, 34}.

В режиме  $I_k$  для вершины 4 было принято  $k = 2$ .

Режим PRIORITY для одностороннего приоритета рассматривался для вершины 3 по отношению к вершине 4 с временной константой задержки, равной 3 усл. ед. времени. Вершина 4 при этом имела тип «И» по отношению к входным дугам 14 и 24. Дуга 34 в этом режиме имела время прохождения  $\tau = 3$  и инициализировалась лишь в случае, если  $0 < d[4] - d[3] < 3$ .

Для всех рассмотренных режимов испытания проводились для одного и того же набора случайных векторов времен прохождения дуг.

Результаты проведенных экспериментов сведены в таблицу, где  $m$  — оценка среднего времени выполнения ОССГ без ограничений на дисциплину («И») свершения события, соответствующего вершине 4 (второй столбец) и с ограничениями в соответствии с указанными режимами (столбцы 3–6),  $\sigma$  — среднеквадратическое отклонение этого времени,  $\Delta$  — ширина интервала доверия для  $m$  при уровне значимости  $\alpha = 0,05$  и числе прогонов  $10^6$ ,  $P_i$  — относительная частота  $i$ -го пути, при этом пути, обозначенные номерами проходимых вершин, упорядочены: 1 — 0135, 2 — 0235, 3 — 0145, 4 — 0245, 5 — 01235, 6 — 02345, 7 — 01245, 8 — 01345, 9 — 012345.  $T_{\max}$  — максимальное время выполнения сетевого графика с указанием критического пути, на котором это значение достигнуто, по всем прогонам. В последних двух строках приведены относительные частоты  $P_{RT}$  и  $P_{SEQ}$  невыполнения условий режимов RUNTIME и SEQUENCE соответственно.

Строки, соответствующие путям 5, 7 и 9, удалены во всех режимах, поскольку все величины в них равны нулю.

Как видно из таблицы, режимы TEST, RUNTIME и SEQUENCE дают достаточно близ-

кие значения оценок среднего времени выполнения ОССГ, среднеквадратического отклонения, интервалов доверия и относительных частот критических путей.

Для режима RUNTIME, имеющего целью не допустить превышение директивного времени выполнения ОССГ (30 усл. ед. времени), удовлетворительный результат был получен при  $RT = 24$ , при этом параметр  $T_{\max}$  принял значение 29,69 при относительной частоте включения режима RUNTIME, равной  $P_{RT} = 2,5 \cdot 10^{-3}$ . Отметим, что уменьшение времени прохождения дуги 45 в 1,3 раза при включении режима RUNTIME также позволяло не допустить превышение директивного времени, при этом время  $T_{\max}$  принимало значение 29,74.

В режиме SEQUENCE, полагая  $m_{14} = m_{24} = 5\sqrt{2}$  (против первоначального значения  $6\sqrt{2}$ ), можно добиться снижения значения  $P_{SEQ}$  до  $4,5 \cdot 10^{-5}$ , т. е. почти в 25 раз.

## ЗАКЛЮЧЕНИЕ

Предложен новый тип сетевых графиков — обобщенные стохастические сетевые графики с вершинами, как традиционными, функционирующими в соответствии с дисциплинами «И» и «ИЛИ», так и специальными, функционирующими по особым правилам. Рассмотрены в качестве примеров четыре типа таких вершин и соответствующие им дисциплины свершения событий.

Предложена стандартная форма конструктивного описания алгоритма нахождения критического пути и времени его прохождения для любого из рассмотренных в статье типов особых вершин, а также для других возможных типов этих вершин, включаемых пользователями в обобщенные стохастические сетевые графики.

## ЛИТЕРАТУРА

1. Иванов Н.Н. Обобщенные стохастические сетевые графики с управляющими дугами // Проблемы управления. — 2017. — № 6. — С. 15–18.
2. Кормен Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ: 2-е изд. — М.: Вильямс, 2005. — 1296 с.
3. Ермаков С.М., Михайлов Г.А. Статистическое моделирование. — М.: Наука, 1982. — 296 с.

Статья представлена к публикации членом редколлегии В.Н. Бурковым.

Иванов Николай Николаевич — д-р техн. наук, ст. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ✉ niknivan@ipu.ru.

Результаты экспериментов

Параметр	Режимы				
	TEST	RUN-TIME	SEQU-ENCE	$I_2$	PRIOR-ITY
$m$	22,37	22,37	22,37	17,51	18,45
$\sigma$	1,89	1,89	1,89	1,37	1,93
$\Delta$	$7,4 \cdot 10^{-3}$	$7,4 \cdot 10^{-3}$	$7,4 \cdot 10^{-3}$	$5,3 \cdot 10^{-3}$	$7,5 \cdot 10^{-3}$
$P_1$	~0	~0	~0	0,0224	0,0161
$P_2$	0,0001	0,0001	0,0001	0,3665	0,1292
$P_3$	0,0011	0,0011	0	0,4353	0,3865
$P_4$	~0	~0	0	0,1747	0,1293
$P_6$	0,8712	0,8712	0,8721	0,0006	0,3289
$P_8$	0,1276	0,1276	0,1278	0,0005	0,01
$T_{\max}$	31,76/6	29,74/6	31,76/6	25/3	26,81/6
$P_{RT}$	—	$2,5 \cdot 10^{-3}$	—	—	—
$P_{SEQ}$	—	—	$1,11 \cdot 10^{-3}$	—	—