$$p_k = \frac{1}{k!}\left(\frac{\lambda}{\nu}\right)^k p_0, \quad 1 \leqslant k \leqslant n, \tag{2}$$

$$p_k = \frac{1}{n!\,n^{k-n}}\left(\frac{\lambda}{\nu}\right)^k p_0, \quad k \geqslant n, \tag{2'}$$

где p_k — вероятность нахождения в обслуживающей системе k требований; n — число каналов обслуживания; p_0 — вероятность отсутствия требований в системе;

$$p_0 = \left[\sum_{k=0}^{n-1} \frac{1}{k!}\left(\frac{\lambda}{\nu}\right)^k + \frac{\nu}{(n-1)!\,(n\nu-\lambda)}\left(\frac{\lambda}{\nu}\right)^n\right]^{-1}, \tag{3}$$

$$M_1 = \frac{p_n\lambda}{n\nu(1-(\lambda/n\nu))^2}, \tag{4}$$

где M_1 — математическое ожидание длины очереди; p_n — вероятность нахождения в системе n требований;

$$M_2 = \sum_{k=1}^{\infty} kp_k = M_1 + \frac{np_n}{1-(\lambda/n\nu)} + p_0 \sum_{k=1}^{n-1} \frac{1}{(k-1)!}\left(\frac{\lambda}{\nu}\right)^k, \tag{5}$$

где M_2 — математическое ожидание числа требований, находящихся в обслуживающей системе.

Все приведенные расчетные формулы справедливы при условии $\lambda < n\nu$, при выполнении которого очередь не может расти безгранично, т. е. система справляется с обслуживанием.

Функция $M = \varphi(n, l)$ имеет вид

$$M = \sum_{k=l+1}^{\infty} (k-l)p_k. \tag{6}$$

Найдем более удобное выражение для M, для чего рассмотрим два случая.

Случай 1. $l \geqslant n-2$. Используя формулы (2′) и (6′), получим:

$$M = \sum_{k=l+1}^{\infty} (k-l)\left(\frac{\lambda}{n\nu}\right)^{k-n} p_n = p_n \sum_{k=0}^{\infty} (k+1)\left(\frac{\lambda}{n\nu}\right)^{k-n+l+1} =$$

$$= p_n \left(\frac{\lambda}{n\nu}\right)^{l-n+1}\left[\sum_{k=0}^{\infty} k\left(\frac{\lambda}{n\nu}\right)^k + \sum_{k=0}^{\infty}\left(\frac{\lambda}{n\nu}\right)^k\right].$$

На основании выражений для сумм сходящихся бесконечных рядов

$$\sum_{k=0}^{\infty} a^k = \frac{1}{1-a} \quad \text{и} \quad \sum_{k=0}^{\infty} ka^k = \frac{a}{(1-a)^2},$$

а также учитывая формулу (4), окончательно имеем:

$$M = M_1\left(\frac{\lambda}{n\nu}\right)^{l-n}, \quad l \geqslant n-2. \tag{6'}$$

Случай 2. $l < n-2$. Из формул (2), (4) и (6) следует:

$$M = \sum_{k=l+1}^{n-2} (k-l)\frac{1}{k!}\left(\frac{\lambda}{\nu}\right)^k p_0 + \sum_{k=n-1}^{\infty} (k-l)\left(\frac{\lambda}{n\nu}\right)^{k-n} p_n =$$

$$= p_0 \sum_{k=l+1}^{n-2} \frac{k-l}{k!}\left(\frac{\lambda}{\nu}\right)^k + \frac{n\nu}{\lambda}\left(p_n \frac{n-l-1}{1-(\lambda/n\nu)} + M_1\right), \quad l < n-2. \tag{6''}$$

Полученные выражения (6′) и (6″) позволяют решать уравнения функции цели (1).

Рассмотрим пример решения задачи нахождения оптимального соотношения между численностью ремонтного персонала и количеством резервных приборов.

Пример. Система обслуживания характеризуется следующими параметрами: $\lambda = 1$, $\nu = 0{,}6$.

Величины, входящие в функцию цели:

$$c_1 = 0{,}5 \text{ руб/час}, \quad c_2 = 1 \text{ руб/час}, \quad c_3 = 0{,}05 \text{ руб/час}.$$

Пользуясь формулами (3), (4), (6′) и (6″), найдем значения M для различных n и l и сведем результаты вычислений в табл. 1.

Подставляя в уравнение (1) значения M, n, l, запишем полученные величины $F(n, l)$ в табл. 2.

Как видно из табл. 2, функция цели $F(n, l)$ достигает минимального значения при $n = 2$, $l = 16$. Таким образом, для нашего примера оптимальным будет вариант с двумя работниками ремонтной мастерской и 16 запасными приборами.

Подобным же способом можно рассчитать оптимальную численность ремонтного персонала и резерва аппаратуры автоматики практически при любых исходных данных, а также решать аналогичные задачи, относящиеся к другим видам оборудования.

ЛИТЕРАТУРА

1. А. Кофман, Р. Крюон. Массовое обслуживание. Теория и приложения. М., «Мир», 1965.

Поступила в редакцию
1 VII 1966

СРАВНЕНИЕ «ЯЗЫКА РАБОТ» И «ЯЗЫКА СОБЫТИЙ» ДЛЯ СОСТАВЛЕНИЯ СЕТЕВЫХ ГРАФИКОВ

И. М. ГЛАЗМАН, В. И. РУБЛИНЕЦКИЙ

(Харьков)

Известно, что существуют два способа составления сетевых графиков: на «языке событий», когда кружки обозначают события, а стрелки — работы и связи между событиями, и на «языке работ», когда события вовсе не вводятся, кружки обозначают работы, а стрелки — связи между работами. С формальной стороны эти два языка эквивалентны, но вопрос о методическом преимуществе того или иного языка требует, по нашему мнению, внимательного изучения.

В настоящее время преимущественное распространение получил язык событий, что отразилось в публикуемых инструктивных и учебных материалах, в том числе в недавно вышедших временных указаниях (Основные положения по разработке и применению систем сетевого планирования и управления. М., «Экономика», 1965). В этой книжке имеется лишь одно примечание, посвященное языку работ, в котором он упоминается и запрещается для использования в межотраслевых системах СПУ (см. стр. 16).

Следует отметить, что язык событий стал господствующим, а язык работ чуть ли не отменен без какого-либо серьезного сравнительного анализа обоих языков. Нам представляется, однако, что язык работ отменять рано, несмотря на вполне понятное стремление к унификации. Сейчас, пока действуют временные инструкции, нужно всесторонне продумать, широко обсудить и практически проверить сравнительные удобства обоих языков. Учитывая огромные масштабы применения СПУ в ближайшем будущем, нужно иметь в виду, что даже малые усовершенствования в методике могут обернуться большой пользой.

Настоящим письмом нам хотелось бы начать такое обсуждение. На протяжении почти трех лет мы имели непосредственное отношение к внедрению (как на языке работ, так и на языке событий) систем СПУ в строительных организациях и на промышленных предприятиях Харькова. На основании опыта мы пришли к выводу, что язык работ намного удобнее языка событий и что, обладая всеми достоинствами последнего, язык работ лишен некоторых недостатков языка событий, что мы и попытаемся показать.

Начнем с возражения весьма общего характера: о целесообразности введения самого понятия события. Такого понятия в традиционном планировании и управлении не существует, поэтому без особо веских аргументов вводить его не следует. Ведь существующая техническая документация описывает работы, а не события; управленческий персонал приучен иметь дело с работами, а не событиями; наконец, не события, а работы, как правило, имеют своего «хозяина», непосредственно ответственного за их выполнение.

Но, может быть, введение нового понятия — события — компенсируется тем, что при его помощи существенно улучшается качество планирования и управления? Оказывается, что нет. Зная лишь, какие события осуществились, а какие нет, невозможно управлять разработкой. Ведь если событие не осуществилось, то нужно еще знать, почему это случилось: то ли соответствующие работы совсем не начинались, то ли, например, всего одна осталась невыполненной. Для того чтобы вести управление с максимальным учетом реальной обстановки, все равно нужно знать, как выполнены работы, а не события, и принимать меры к тому, чтобы выполнить работы, после чего события произойдут уже автоматически. Поэтому мы полагаем, что проще сразу составить сетевой график на языке работ, не вводя лишних понятий.

В ходе преподавания основ СПУ мы убедились в том, что язык событий менее доступен для практических работников, чем язык работ. Это объясняется не только новыми понятиями, такими, как событие и фиктивная работа, но и совмещением в одном графическом знаке (стрелке) функций работы и связи.

Мы считаем, что в дальнейших учебных руководствах и программах по общему курсу СПУ должны освещаться оба языка.

Нам не известен ни один убедительный аргумент против составления сетей на языке работ. Правда, сторонники языка событий часто приводят возражение, что понятие работы не заключает в себе информации, необходимой для перехода к другим, следующим работам. Например, выполнение работы «проектирование узла A изделия» не дает возможности начать следующую работу, так как для нее нужно, чтобы были оформлены и сданы чертежи. Поэтому, дескать, нужно выполнение события: «проект узла A изделия разработан и передан в технологический отдел». Такая аргументация несостоятельна и возникает от того, что работа описана неполно. Возражение отпадет, если полнее определить работу, а именно: «проектирование узла A и сдача чертежей в технологический отдел».

Но события являются не только в некотором смысле лишним элементом в системах СПУ. Сети, составленные на языке событий, имеют еще один — и очень существенный — недостаток по сравнению с сетями, составленными на языке работ: состав кружков в сети может меняться, если добавляются новые связи между работами. Проиллюстрируем сказанное на примере. При этом мы будем пользоваться следующими обозначениями: в примерах на языке работ мы будем ставить буквы в кружках, в примерах на языке событий эти же буквы будем ставить на стрелках, изображающих работы, а буквы с одним или двумя штрихами мы будем использовать для обозначения событий, отвечающих соответственно началу и концу работы.

Пусть имеется следующий подграфик сетевого графика на языке работ, представляющий собой «вилку вправо» (рис. 1, *а*) и тот же подграфик на языке событий (рис. 1, *б*). Если нужно провести дополнительно связь между работой d того же графика и работой c, то в модели на языке работ достаточно поставить стрелку из d в c (рис. 1, *в*). В графике же на языке событий нужно дополнительно ввести новый кружок — событие c' (рис. 1, *г*). Но если нужно изобразить связь между d и b, то придется перестраивать график иначе, вводя новое событие b'. Совершенно аналогичные трудности возникают, если подграфик представляет собой «вилку влево» (рис. 2).

Зависимость набора событий от связей представляет неудобство даже в такой обычной операции, как дополнение сети новыми работами, а следовательно, и связями. Это неудобство становится особенно ощутимым при решении на ЭВМ оптимизационных задач путем варьирования ресурсных связей (т. е. порядка предоставления работам ресурсов).

Для того чтобы сделать график на языке событий устойчивым по отношению к изменению связей, нужно составлять графики так, как показано на рис. 1, *д* и 2, *д*, но это привело бы к тому, что сети на языке событий стали бы почти вдвое более громоздкими, чем на языке работ, не имея по сравнению с ними никаких преимуществ.

Даже такое незначительное обстоятельство, что в сети на языке событий работа обозначается двумя номерами, а на языке работ — одним, также является преимуществом языка работ, так как при программировании алгоритмов, используемых в СПУ, сэкономленные разряды номеров работ отводят для всяких полезных признаков, например, указывающих исполнителя работы, вид потребляемых ресурсов и т. п. Кодирование работы парой событий в сетях на языке событий представляется неудобным также в связи с отмеченной выше нестабильностью набора событий: введение новых связей даже при неизменном составе работ приводит, вообще говоря, к изменению кодов некоторых из этих работ.

Введение событий, на наш взгляд, может быть оправдано лишь для особых разработок (например, поисковых, научных), когда в зависимости от того, осуществится или нет данное событие, выбирается тот или иной вариант продолжения разработки. Но в этом особом случае события играют совершенно иную роль, чем та, которая отводится для них в обычных случаях.

Вследствие всего сказанного мы считаем язык работ более удобным для использования в сетевых моделях именно как рабочий инструмент, как язык, на котором

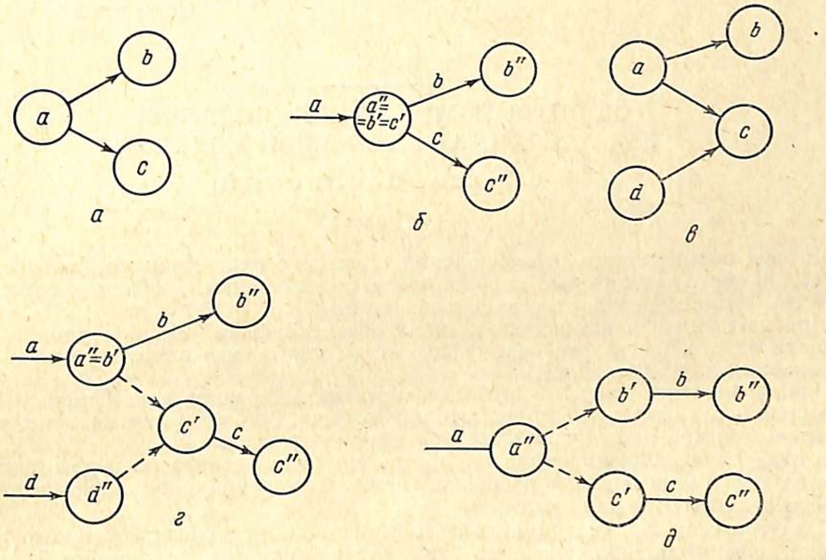

Рис. 1

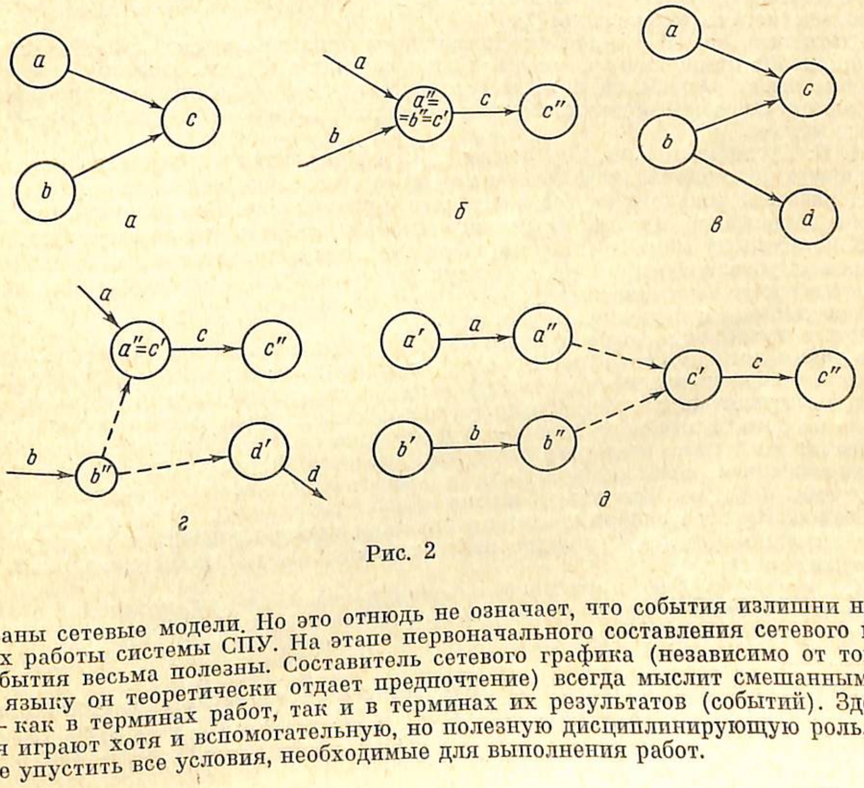

Рис. 2

записаны сетевые модели. Но это отнюдь не означает, что события излишни на всех этапах работы системы СПУ. На этапе первоначального составления сетевого графика события весьма полезны. Составитель сетевого графика (независимо от того, какому языку он теоретически отдает предпочтение) всегда мыслит смешанным образом — как в терминах работ, так и в терминах их результатов (событий). Здесь события играют хотя и вспомогательную, но полезную дисциплинирующую роль, помогая не упустить все условия, необходимые для выполнения работ.

Возможно, нам неизвестны некоторые достоинства языка событий, оправдывающие его повсеместное распространение в качестве рабочего языка систем СПУ. Мы просим компетентных специалистов высказаться по этому поводу. Одно несомненно: только всестороннее обсуждение этого вопроса поможет правильно решить, какой из двух возможных языков (а может быть, оба) нужно оставить в будущем.

Поступила в редакцию
9 II 1966

ОБ ОПТИМИЗАЦИИ ФУНКЦИОНАЛОВ, ЗАДАННЫХ СТАТИСТИЧЕСКОЙ МОДЕЛЬЮ *

В. А. СТАРОСЕЛЬСКИЙ

(Москва)

Методы оптимального распределения занимают существенное место среди математических методов, которые используются при решении проблем организации производства, экономики и исследования операций.

Многие практические задачи из этих областей были успешно решены методами математического программирования, и в настоящее время имеется достаточный опыт применения их на практике [1].

Однако по мѐре развития этой проблематики все чаще стали появляться задачи, связанные с оптимизацией процессов управления, для решения которых методы математического программирования оказываются непригодными.

В этих случаях применение аппарата математического программирования для решения задач оптимального распределения ввиду грубой формализации не представляет особого интереса для практики.

С другой стороны, создание методов решения задач математического программирования, сформулированных с необходимой общностью (нелинейный функционал, учет случайных факторов и др.), наталкивается на серьезные трудности, в частности на вычислительные.

Если же система достаточно сложна, то часто функцию цели вообще невозможно записать в явном виде. Для исследования процессов функционирования сложных систем с учетом широкого круга действующих факторов успешно используется метод статистического моделирования [2].

Статистическое моделирование позволяет в принципе вычислять значения любых функционалов от параметров системы, включенных в модель. Поэтому существенное значение имеет разработка методов решения задач математического программирования для случая, когда функция цели и ограничения не заданы аналитически, а с помощью модели.

Н. П. Бусленко и Г. А. Соколовым [3] был предложен метод, позволяющий оптимизировать функционал, заданный с помощью статистической модели. К сожалению, метод работает только в случае выпуклого функционала. Все рассуждения в [3] ведутся в общем виде, и вычислительная сторона вопроса почти не затронута.

К настоящему моменту этот метод отработан и в процессе применения частично изменен к условному примеру, где изучаются системы, представляющие собой системы массового обслуживания.

Задача имеет следующую постановку.

Пусть имеется система массового обслуживания с тремя каналами; заявки поступают двумя потоками по пуассоновскому закону распределения с параметрами λ_1 и λ_2. Заявка первого потока поступает на обслуживание на первый канал и, если он занят, поступает на второй. В случае занятости второго канала заявка теряется. Заявка второго потока поступает на обслуживание на второй канал, если он занят — на третий и в случае занятости его также теряется. Время обслуживания τ каждой заявки свободным каналом определяется показательным законом распределения с параметрами μ_1, μ_2, μ_3. Потеря требования приводит к убытку, равному α_1 или α_2 соответственно. Простой канала в единицу времени также приводит к убытку, равному соответственно c_1, c_2, c_3. Требуется найти такие μ_1, μ_2, μ_3, чтобы суммарный убыток был минимальным.

В литературе [4] есть описание решения похожей задачи, полученного аналитическим путем, но в данном случае мы преследовали цель отработки метода Бусленко-Соколова и поэтому целевую функцию находили при помощи модели Монте-Карло.

* Автор выражает благодарность руководителю группы математического моделирования сложных систем ЦЭМИ АН СССР Х. Ш. Маргулису, под чьим непосредственным руководством велась настоящая работа, а также чл.-корр. АН СССР Н. П. Бусленко за систематические консультации и советы.

Ниже приводятся результаты численных расчетов решения этой задачи по данному методу при следующих параметрах:

$\lambda_1 = 0{,}4$; $\lambda_2 = 0{,}6$; $\alpha_1 = 3$; $\alpha_2 = 4$; $c_1 = 7$; $c_2 = 8$; $c_3 = 9$ и переменных μ_1, μ_2, μ_3.

$$\mu^0 \begin{cases} \mu_1 = 1, \\ \mu_2 = 3, \\ \mu_3 = 4, \end{cases} \quad f(\mu^0) = 36625{,}2498; \qquad \mu^4 \begin{cases} \mu_1 = 8{,}75, \\ \mu_2 = 9{,}53125, \\ \mu_3 = 8{,}125, \end{cases} \quad f(\mu^4) = 29697{,}3809;$$

$$\mu^1 \begin{cases} \mu_1 = 5, \\ \mu_2 = 8{,}75, \\ \mu_3 = 7{,}5 \end{cases} \quad f(\mu^1) = 34257{,}3106; \qquad \mu^5 \begin{cases} \mu_1 = 7{,}5, \\ \mu_2 = 7{,}5, \\ \mu_3 = 5, \end{cases} \quad f(\mu^5) = 29543{,}7671;$$

$$\mu^2 \begin{cases} \mu_1 = 8{,}125, \\ \mu_2 = 5, \\ \mu_3 = 9{,}84375, \end{cases} \quad f(\mu^2) = 34009{,}4225; \qquad \mu^6 \begin{cases} \mu_1 = 5, \\ \mu_2 = 7{,}5 \\ \mu_3 = 1{,}71875, \end{cases} \quad f(\mu^6) = 29376{,}6765;$$

$$\mu^3 \begin{cases} \mu_1 = 7{,}5, \\ \mu_2 = 7{,}5, \\ \mu_3 = 8{,}75, \end{cases} \quad f(\mu^3) = 29799{,}5936; \qquad \mu^7 \begin{cases} \mu_1 = 5, \\ \mu_2 = 8{,}75, \\ \mu_3 = 5, \end{cases} \quad f(\mu^7) = 29356{,}9353;$$

$$\mu^8 \begin{cases} \mu_1 = 5, \\ \mu_2 = 8{,}75, \\ \mu_3 = 5, \end{cases} \quad f(\mu^8) = 29356{,}9383.$$

Как видим, оптимизация методом Бусленко — Соколова дает хорошие результаты.

ЛИТЕРАТУРА

1. Д. Б. Юдин, Е. Г. Гольштейн. Задачи и методы линейного программирования. М., «Сов. радио», 1961.
2. Н. П. Бусленко. Математическое моделирование производственных процессов на цифровых вычислительных машинах. М., «Наука», 1964.
3. Н. П. Бусленко, Г. А. Соколов. Об одном классе задач оптимального распределения. Экономика и матем. методы, 1965, т. 1, вып. 1.
4. А. Кофман, Р. Крюон. Массовое обслуживание. Теория и приложения. М., «Мир», 1965.

Поступила в редакцию
7 VI 1966

О РАСПРЕДЕЛЕНИИ ЧИСЛА ШАГОВ СИМПЛЕКС-МЕТОДА

Н. Н. ВОРОБЬЕВ

(Ленинград)

В работе [1] приводятся математическое ожидание и дисперсия числа шагов в методах линейного программирования типа симплекс-метода, если переход от одной вершины многогранника допустимых точек к другой, с меньшим значением целевой функции, осуществляется случайно, причем все такие вершины принимаются равновероятными. После этого вероятность тех или иных отклонений числа шагов от его математического ожидания оценивается на основе неравенства Чебышева.

Далее описывается несколько иной подход к вопросу, основанный на том, что в случае большого числа вершин многогранника доля точек, остающихся после очередного шага, может принимать значения, расположенные в интервале $(0, 1)$ «достаточно плотно». Поэтому распределение числа остающихся вершин мало отличается от равномерного на $(0, 1)$ (т. е. от распределения, имеющего на $(0, 1)$ единичную плотность, а вне этого интервала — нулевую) и может быть им заменено. Отсюда выводится, что при больших N_1 и N_2 число шагов, за которые задача поиска экстремальной вершины многогранника с N_1 вершинами сводится к аналогичной задаче для случая N_2 вершин $(N_2 < N_1)$, распределено асимптотически по закону Пуассона с параметром $\log N_1 / N_2$.

В самом деле, при переходе от числа N_1 вершин к числу N_2 мы как бы переходим от их полного числа к доле $\gamma = N_2 / N_1$. Каждый k-й шаг состоит в умножении доли вершин Y_{k-1}, имевшихся к началу этого шага, на долю X_k от Y_{k-1} вершин, остающихся после этого шага.

Таким образом, мы должны исследовать случайную величину Z, принимающую значение n в том и только в том случае, когда

$$X_1X_2, \ldots, X_{n-1}X_n = Y_n < \gamma < X_1X_2, \ldots, X_{n-1} = Y_{n-1}. \tag{1}$$