Пусть - максимальный поток. Определим два множества V и W вершин сети: пусть G обозначено основание орграфа D, соответствующего рассматриваемой сети; тогда вершина z сети содержится в V в том и только в том случае, если в G существует простая цепь $v = v_0 \to v_1 \to v_2 \to ... \to v_{m-1} \to v_m = z$, обладающая тем свойством, что любое ее ребро $\{v_i, v_{i+1}\}$ соответствует либо ненасыщенной дуге (v_i, v_{i+1}), либо дуге (v_{i+1}, v_i), через которую проходит ненулевой поток. (Заметим, что вершина v, очевидно, содержится в V). Множество W состоит из всех тех вершин, которые не принадлежат V.

Покажем теперь, что W не пусто и, в частности, содержит вершину w. Если это не так, то w принадлежит V, и тогда в G существует простая цепь $v \to v_1 \to v_2 \to ... \to v_{m-1} \to w$, обладающая указанным выше свойством. Выберем положительное число , удовлетворяющее следующим двум условиям:

- оно не превышает ни одного из чисел, необходимых для насыщения дуг первого типа;
- оно не превышает потока через любую из дуг второго типа.

Очевидно, что если потоки через дуги первого типа увеличить на , а потоки через дуги второго типа уменьшить на , то величина потока увеличится на . Но это противоречит нашему предположению о том, что - максимальный поток, и, следовательно, w содержится в W.

Для завершения доказательства обозначим через E множество всех дуг вида (x, z), где x принадлежит V, а z принадлежит W. Ясно, что E является разрезом. Более того, мы видим, что каждая дуга (x, z) из E насыщена, так как в противном случае вершина z также принадлежала бы V. Следовательно, пропускная способность множества E должна равняться величине потока , а поэтому E и есть искомый разрез.

Приложение

Теорема о максимальном потоке и минимальном разрезе позволяет проверять, максимален данный поток или нет, но только для достаточно простых сетей. Разумеется, на практике приходится иметь дело с большими и сложными сетями, и в общем случае трудно найти максимальный поток простым подбором. Опишем один алгоритм нахождения максимального потока в любой сети с целочисленными пропускными способностями. Перенесение этого алгоритма на сети с рациональными пропускными способностями осуществляется тривиальным образом и предоставляется читателю.

Шаг 1. Сначала подберем поток , обладающий ненулевой величиной (если такой поток существует). Стоит отметить, что чем больше величина выбранного нами начального потока , тем проще будут последующие шаги.

Шаг 2. Исходя из N, строим новую сеть N' путем изменения направления потока на противоположное. Более точно, любая дуга a, для которой $(a) = 0$, остается в N' со своей первоначальной пропускной способностью, а любая дуга a, для которой $(a) \neq 0$, заменяется дугой a с пропускной способностью $(a) - (a)$ и противоположно направленной дугой с пропускной способностью (a).

Шаг 3. Если в сети N' мы сможем найти ненулевой поток из v в w, то его можно добавить к первоначальному потоку и получить в N новый поток $'$ большей величины. Теперь можно повторить шаг 2, используя при построении сети N' новый поток $'$ вместо . Повторяя эту процедуру, мы в конце концов придем к сети N', не содержащей ненулевых потоков; тогда соответствующий поток будет максимальным потоком.

УДК 519.6

Нина Ивановна Костюкова
Институт вычислительной математики и математической геофизики СО РАН

СЕТЕВОЕ ПЛАНИРОВАНИЕ И УПРАВЛЕНИЕ©

Введение

В нашей стране разработаны системы планирования и управления «СПУ». В основе этих систем лежат сетевые графики. Системы «СПУ» успешно применялись, например, при сооружении ТЭЦ в Лисичанске, Буштырской тепловой электростанции, Челябинского блюминга-автомата «1300», при ремонте мартеновской печи завода «Серп и молот», при реконструкции доменной печи в «Запорожстали» и т.д.

Сетевая модель была применена в США при создании баллистических ракет «Поларис», предназначенных для оснащения атомных подводных лодок американского военно-морского флота. В сложном комплексе работ при этом участвовало свыше 6000 фирм, работы выполнялись на территории 48 штатов Америки, а сетевой график включал в себя более 10000 событий.

Сетевой график

Всякий намеченный комплекс работ, необходимых для достижения некоторой цели, называют *проектом*. Проект (или комплекс работ) подразделяется на отдельные работы. Каждая отдельная работа, входящая в комплекс (проект), требует затрат времени. Некоторые работы могут выполняться только в определенном порядке. При выполнении комплекса работ всегда можно выделить ряд *событий*, то есть итогов какой-то деятельности, позволяющих приступить к выполнению следующих работ. Если каждому событию поставить в соответствие вершину графа, а каждой работе - ориентированное ребро, то получится некоторый граф. Он будет отражать последовательность выполнения отдельных работ и наступление событий в едином комплексе. Если над ребрами проставить время, необходимое для завершения соответствующей работы, то получится *сеть*. Изображение такой сети называют сетевым графиком. Сетевой график состоит из двух типов основных элементов: работ и событий. *Работа* представляет собой выполнение некоторого мероприятия (например, погрузка боезапаса или переход корабля в пункт базирования). Этот элемент сетевого графика связан с затратой времен и расходом ресурсов. Поэтому работа всегда имеет начало и конец. Кроме того, каждая работа должна иметь определение, раскрывающее ее содержание (например, уяснение боевой задачи, приготовление корабля к походу и т.д.).

На сетевом графике работа изображается стрелкой, над которой проставляется ее продолжительность или затрачиваемые ресурсы, или то и другое одновременно. Работа, отражающая только зависимость одного мероприятия от другого, называется *фиктивной работой*. Такая работа имеет нулевую продолжительность (или нулевой расход ресурсов) и обозначается пунктирной стрелкой.

Начальная и конечная точки работы, то есть начало и окончание некоторого мероприятия (например, окончание приготовления корабля к бою), называются *событиями*. Следовательно, событие, в отличие от работы, не является процессом и не сопровождается никакими затратами времени или ресурсов.

Событие, следующее непосредственно за данной работой, называется *последующим событием* по отношению к рассматриваемой работе. *Событие*, непосредственно предшествующее рассматриваемой работе, называется *предшествующим*.

Наименования «предшествующий» и «последующий» относятся также и к работам. Каждая входящая в данное событие *работа* считается *предшествующей* каждой выходящей работе, и наоборот, каждая выходящая *работа* считается *последующей* для каждой входящей.

Из определения отношения «предшествующий-последующий» вытекают свойства сетевого графика.

Во-первых, ни одно событие не может произойти до тех пор, пока не будут закончены все входящие в него работы. Во-вторых, ни одна работа, выходящая из данного события, не может начаться до тех пор, пока не произойдет данное событие. И, наконец, ни одна последующая работа не может начаться раньше, чем будут закончены все предшествующие ей.

Событие обозначается кружком с цифрой внутри, определяющей его номер.

Из всех событий, входящих в планируемый процесс, можно выделить два специфических - событие начала процесса, получившее название *исходного события*, которому присваивается нулевой номер, и событие конца процесса (*завершающее событие*), которому присваивается последний номер. Остальные события нумеруются так, чтобы номер предыдущего события был меньше номера последующего.

Для нумерации событий применяется следующий способ. Вычеркиваются все работы, выходящие из события с номером «0», и просматриваются все события, в которых оканчиваются эти вычеркнутые работы. Среди просмотренных находятся события, которые не имеют входящих в них работ (за исключением уже вычеркнутых). Они называются *событиями первого ранга* и обозначаются (вообще, в произвольном порядке) числами натурального ряда, начиная с единицы (на Рис. 1.1 это событие 1). Затем вычеркиваются все работы, выходящие из событий первого ранга, и среди них находятся события, не имеющие входящих работ (кроме вычеркнутых). Это - события второго ранга, которые нумеруются следующими числами натурального ряда (например, 2 и 3 на Рис. 1.1). Проделав таким способом $(k-1)$ шаг, определяют *события* $(k-1)$*-го ранга* , и просматривая события, в которых эти работы заканчиваются, выбирают события, не имеющие ни одной входящей в них работы (кроме вычеркнутых). Это события k -го ранга, и нумеруются они последовательными числами натурального ряда, начиная с наименьшего, еще не использованного числа при предыдущей нумерации на $(k-1)$-м шаге.

Сетевой график содержит конечное число событий. Поскольку в процессе вычеркивания движение осуществляется в направлении стрелок (работ), никакое предшествующее событие не может получить номер, больший, чем любое последующее. Всегда найдется хотя бы одно событие соответствующего ранга, и все события получат номера за конечное число шагов.

Работа обычно кодируется номерами событий, между которыми они заключены, то есть парой (i, j), где i - номер предшествующего события, j - номер последующего события.

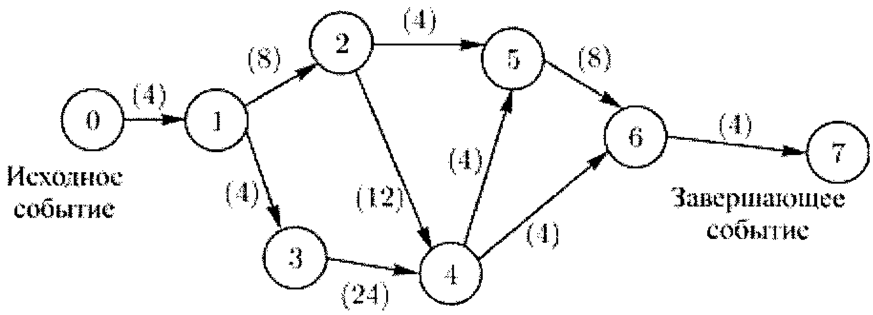

Рис. 1.1

В одно и то же событие могут входить (выходить) одна или несколько работ. Поэтому свершение события зависит от завершения самой длительной из всех входящих в него работ.

Взаимосвязь между работами определяется тем, что начало последующей работы обусловлено окончанием предыдущей. Отсюда следует, что нет работ, не связанных началом и окончанием с другими работами через события.

Последовательные работы и события формируют цепочки (пути), которые ведут от исходного события сетевого графика к завершающему. Например, путь $0 \to 1 \to 2 \to 5 \to 6 \to 7$ сетевого графика, показанного на Рис. 1.1, включает в себя события $0,1,2,5,6,7$ и работы $(0-1),(1-2),(2-5),(5-6),(6-7)$.

На основании изложенного можно сказать, что ранг события - это максимальное число отдельных работ, входящих в какой-либо из путей, ведущих из нулевого (исходного) события в данное. Так, события первого ранга не имеют путей, состоящих более чем из одной работы, ведущих в них из 0 (например, событие 1 на Рис. 1.1). События второго ранга связаны с 0 путями, которые состоят не более чем из двух работ, причем для каждого события второго ранга хоть один такой путь обязательно существует. Например, на Рис. 1.1 событие 4 - событие третьего ранга, так как пути, ведущие в это событие из 0, включают только три работы - $(0-1),(1-3)$ и $(3-4)$ или $(0-1),(1-2)$ и $(2-4)$.

Построенный таким образом сетевой график в терминах теории графов представляет собой направленный граф.

На рисунке изображен сетевой график. Граф, не содержащий циклов и имеющий только один исток и только один сток, называется *направленным графом*. Сетевой график есть ориентированный связный асимметрический граф с одним истоком, одним стоком и без циклов, то есть это направленный граф. При этом вершинами графа служат события сетевого графика, а дугами (ребрами) - работы сетевого графика.

Продолжительность работы представляет собой, в терминах теории графов, длину дуги. Следовательно, длина пути - это сумма длин всех дуг, образующих данный путь, то есть $T = \sum t_{i,j}, t_{i,j} \in T$, где символом $t_{i,j}$ обозначается дуга, которая соединяет вершины i и j и направлена от вершины i к вершине j.

Правила построения сетевого графика

Обычно сетевой график строится от исходного события к завершающему, слева направо, то есть каждое последующее событие изображается несколько правее предыдущего.

В планируемых процессах часто встречаются сложные комплексные связи, когда две или более работ выполняются параллельно, но имеют общее конечное событие, или когда для выполнения одной из работ необходимо предварительно выполнить несколько работ, а для другой, выходящей из общего для них события, предварительным условием является выполнение только одной из предшествующих работ и т.д. Изображение в сетевой модели подобных параллельных или дифференцированно зависимых работ выполняется следующим образом.

В случае, когда наступление события (например, 3 на Рис. 1.2) возможно в результате завершения двух работ $(1-3)$ и $(2-4)$, но в то же время существует событие 4, зависящее от завершения только одной из этих работ (например, $(2-4)$), вводится фиктивная работа $(4-3)$ (см. Рис. 1.2).

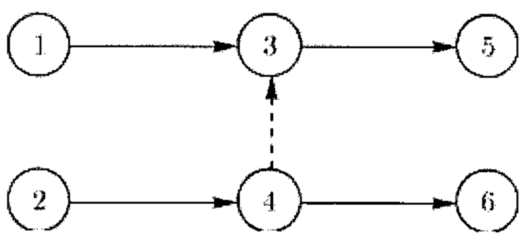

Рис. 1.2

Если одно событие (например, 1 на Рис. 1.3) служит началом двух (например, $(1-2)$ и $(1-3)$ или нескольких работ, заканчивающихся в другом событии (3 на Рис. 1.3)), то для их различия также вводится фиктивная работа $(2-3)$ (см. Рис. 1.3). С помощью фиктивной работы в сетевом графике могут быть отражены и двусторонние связи (зависимости).

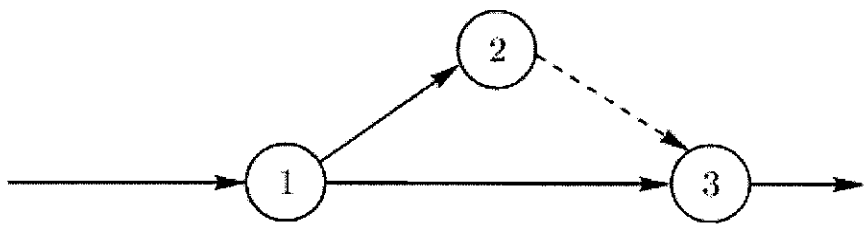

Рис. 1.3

Пусть, например, имеются три процесса А, В, С. При этом окончание процесса С зависит от результатов процессов А и В. В этом случае возникают двусторонние зависимости, которые можно изобразить так, как показано на Рис. 1.4.

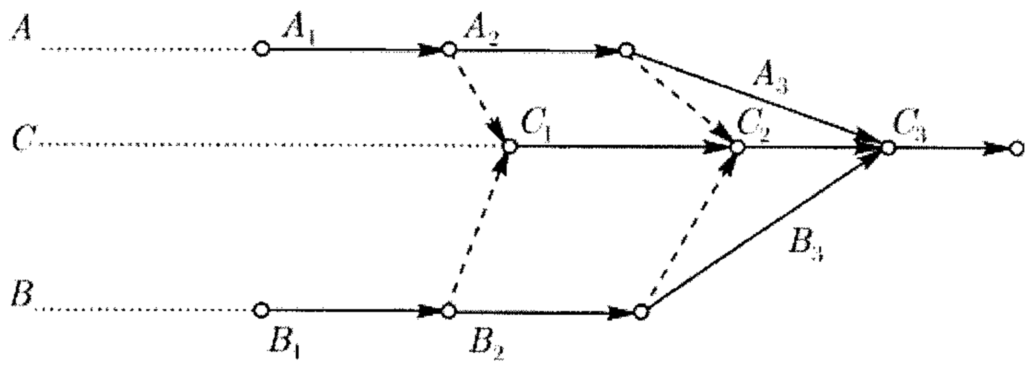

Рис. 1.4

Другое правило построения сетевого графика заключается в том, что если несколько работ может начаться не после полного, а после частичного выполнения определенной работы, то последнюю работу целесообразно представить как сумму ее частей, расчлененных событиями $(1,2,3,4$ и 5 на Рис. 1.5). И в то же время, группу работ целесообразно представить одной работой, если в этой группе имеется по одному начальному и конечному событию (1 и 4 на Рис. 1.6).

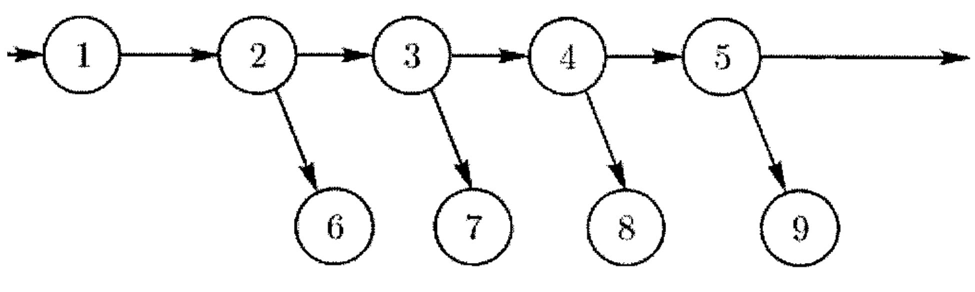

Рис. 1.5

Для отображения времени и места поступления дополнительных ресурсов (например, пополнение личного состава, топлива и т.д.) и другой информации на сетевом графике закрашенным кружком изображаются так называемые подставки (Рис. 1.7). При наличии двух и более работ, выходящих из события, с которым необходимо связать подставку, последняя соединяется с дополнительно введенным событием через фиктивную работу (Рис. 1.7).

После построения сетевого графика проверяется отсутствие работ, имеющих одинаковые коды. При наличии таких работ вводятся дополнительные события и фиктивные работы. Кроме того, сетевой график должен содержать только одно исходное событие и только одно завершающее событие.

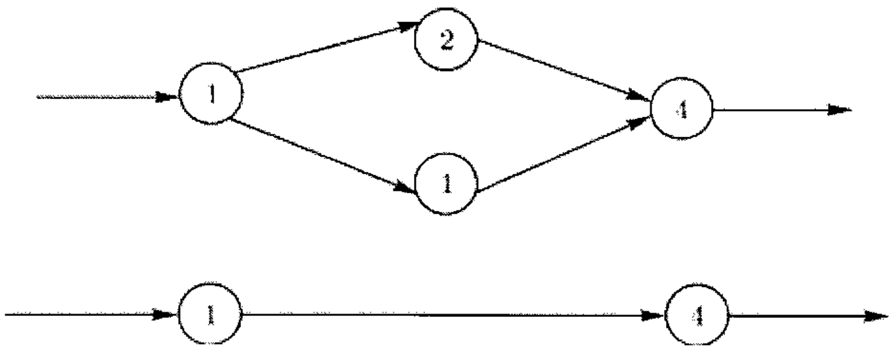

Рис. 1.6

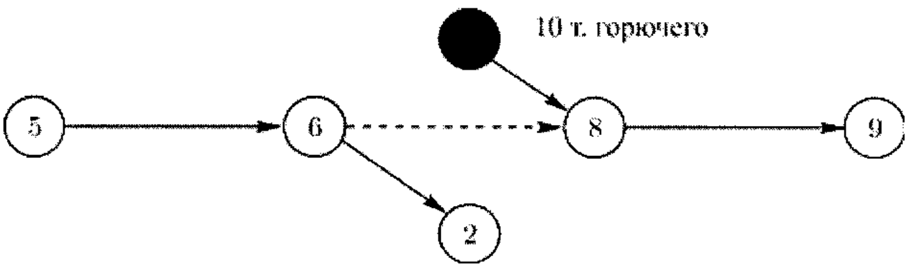

Рис. 1.7

Если эти условия не выполнены, то необходимо добавить еще одно исходное событие и соединить его стрелками с имеющимися несколькими начальными событиями или добавить еще одно конечное событие, к которому ведут стрелки от нескольких имеющихся конечных событий.

Сетевой график не должен иметь циклов, то есть таких путей, в которых конец последней работы совпадает с началом первой работы. Сетевой график, имеющий хотя бы один цикл, не может быть реализован, так как ни одна из работ, входящих в такой цикл, никогда не может начаться.

Анализ сетевой модели

Параметры сетевой модели. Параметрами сетевой модели являются:

- наиболее ранее возможное время наступления j-го события, обозначаемое символом $T_p(j)$;
- самое позднее допустимое время наступления i-го события, обозначаемое символом $T_n(i)$;
- резерв времени данного события, обозначаемый символом R_i;
- полный резерв времени работы (i, j), обозначаемый символом $r_n(i, j)$;
- свободный резерв времени работы (i, j), обозначаемый символом $r_c(i, j)$.

Наиболее раннее возможное время наступления j-го события определяется следующий рекуррентной формулой:

$$T_p(j) = \max_{i \in \Gamma_o^{-1}} \{T_p(i) + t_{ij}\} \tag{1.1}$$

где t_{ij} - продолжительность (i, j)-й работы; Γ_j^{-1} - множество событий, предшествующих j-му событию.

Вычисления по формуле (1.1) выполняются шаг за шагом, двигаясь в порядке нумерации событий.

Самое позднее допустимое время наступления события i определяется с помощью аналогичной рекуррентной формулы, но обращаясь не к предшествующим, а к последующим событиям.

$$T_n(i) = \min_{j \in \Gamma_i} \{T_n(j) - t_{ij}\}, \tag{1.2}$$

где Γ_i - множество событий, следующих за i-м событием.

Для определения $T_n(i)$ по формуле (1.2) надо двигаться от конечного события n к исходному событию 0, при этом $T_n(n) = T_p(n)$.

Резервом времени данного события называется разность между $T_n(i)$ и $T_p(i)$, которая вычисляется по формуле

$$R_i = T_n(i) - T_p(i) \tag{1.3}$$

Полный резерв времени работы (i, j) вычисляется по формуле

$$r_n(i, j) = T_n(j) - T_p(i) - t_{ij} \quad (1.4)$$

Свободный резерв времени работы (i, j) вычисляется по формуле

Список литературы

1. **Костюкова Н. И.** Графы и их применение: комбинаторные алгоритмы для программистов // Интернет-Университет информационных технологий. БИНОМ: Лаборатория знаний, 2007.

УДК 519.6

Нина Ивановна Костюкова, Андрей Евгеньевич Кудинов
Институт вычислительной математики и математической геофизики СО РАН

ОСНОВНЫЕ ПРИНЦИПЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ©

Определение моделирования

Моделированием называют построение модели того или иного явления реального мира. В общем виде ***модель*** - это абстракция реального явления, сохраняющая его существенную структуру таким образом, чтобы ее анализ дал возможность определить влияние одних сторон явления на другие или же на явления в целом. В зависимости от логических свойств и связей моделей с отображаемыми явлениями можно все модели разделить на три типа: изобразительные, аналоговые и математические.

Изобразительная модель отражает внешние характеристики явления и подобна оригиналу. Это наиболее простая и конкретная модель. Являясь в общем описательной моделью, она, как правило, не дает возможности установить причинные связи явления и соответственно определить или предсказать последствия изменений различных параметров явления. Характерная особенность такой модели - близкое совпадение ее свойств со свойствами отображаемого объекта. Эти свойства обычно подвергаются метрическому преобразованию, т.е. берется определенный масштаб.

В ***аналоговых моделях*** свойство данного явления отображается посредством свойств другого явления. Так, например, любая диаграмма представляет аналоговую модель некоторого явления. К аналоговым моделям относятся также морские карты, на которых совокупностью условных обозначений отображается совокупность свойств той или иной акватории. Преимущество аналоговой модели перед изобразительной состоит в том, что она позволяет отображать динамику явления. Другим преимуществом является большая универсальность этой модели: путем ее изменения можно отобразить различные процессы данного явления.

Математическая модель

Математическая модель является самой сложной и наиболее общей и абстрактной по сравнению с изобразительной и аналоговой. В ней для отображения свойств изучаемого явления используются символы математического или логического характера. Особые трудности возникают при решении задач с большой размерностью, расплывчатой постановкой, неопределенностью информации и т.д. В постановке таких задач появляются неклассические моменты, такие, как плохая формализуемость, нестандартность, противоречивость.

Остановимся на понятии плохо формализуемой задачи, которое появляется в результате решения потока серьезных прикладных задач в самых различных областях. Это могут быть и формализованные правила рассуждений, и правила логического вывода. Математические модели служат отражению и анализу некоторых свойств действительных объектов. Рассмотрим один из видов математических моделей, характеризующихся простой структурой и широко применяющихся в приложениях. Модели такого вида содержат следующие элементы:

1. вектор x параметров, измеряемых на объекте $x = [x_1, ..., x_n]$, где x_i - значение i-го параметра, которое является чаще всего вещественным числом. Можно назвать x вектором состояния объекта. Если изучается динамика моделируемого объекта во времени t, то считается, что состояние в каждой момент t описывается вектором $x(t) = [x_1(t), ..., x_n(t)]$;
2. вектор $y(t)$ параметров, которые не могут быть непосредственно измеренными;
3. неизвестные связи между переменными координатами векторов $x(t)$ и $y(t)$;
4. связи между переменными, являющиеся неизвестными;
5. математический аппарат исследования соотношений (связей).

В качестве примера можно привести имитационные модели, описывающие возможные пути развития сложных технико-экономических и природных систем.
