АКАДЕМИЯ СТРОИТЕЛЬСТВА И АРХИТЕКТУРЫ
ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Кафедра «Организации строительства»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению практических занятий
по дисциплине

«Системно-поточная организация строительства»

Автор

Ключникова О.В.

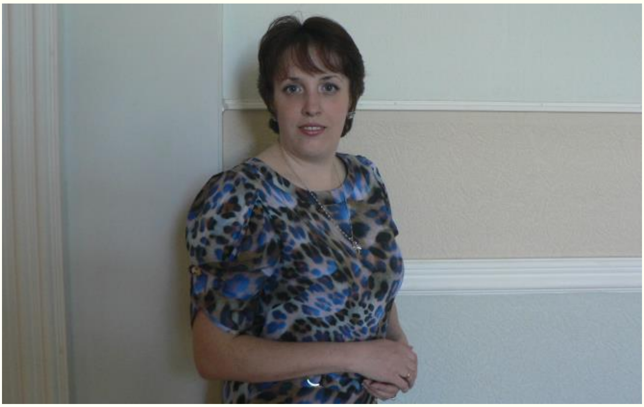
Ростов-на-Дону, 2016

Аннотация

Методические указания предназначены для студентов всех форм обучения подготовки магистров по направлению 08.04.01 «Строительство» по профессионально-образовательной программе «Теория и практика организационно-технологических и экономических решений»

Автор

к.т.н., доцент каф. «Организации строительства» Ключникова О.В.



Оглавление

1.ОБЩИЕ ПОЛОЖЕНИЯ О ПОНЯТИИ МОДЕЛИРОВАНИЯ

Для любой задачи управления характерна множественность ее решений. Кроме того, постоянное усложнение техники и технологии строительного производства и связанное с ним усложнение процесса управления, делают выбор оптимального решения чрезвычайно трудным.

Модель представляет собой абстрактное отображение наиболее существенных характеристик, процессов и взаимосвязей реальных систем.

По свойствам модели можно судить о наиболее существенных свойствах объекта, которые аналогичны и в модели, и в объекте и являются основными для исследований и решений определенного круга задач. Модель содержит и порождает информацию, адекватную информации оригинала.

Различают физические и абстрактные виды моделей. Физичекская модель – это материальная система, выполняемая из других материалов и в размерах отличных от объекта. Абстрактная модель создается математическими, языковыми методами.

В управлении организации, планировании и проектировании в строительстве применяются линкйные графические модели и сетевые модели.

Классификация сетевых моделей:

1. по организационной структуре;
2. по характеристикам (ориентирован на событие или работу);
3. по характеру решаемых задач;
4. по периоду функционирования во времени (цикличные или не цикличные);
5. по использованию средств переработки информации (автоматизированные и не автоматизированные).

Граф – это геоматрическая фигура, состоящая измножества точек и соединяющих ихлиний. В сетевом моделировании применяется ориентированный граф. В строительном производстве используется при построении сетевых графиков способ, при котором ориентированный граф изображает работу, а вершины события, результат этой работы.

2.ОСНОВНЫЕ СОСТАВЛЯЮЩИЕ ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ ПРИМЕНИТЕЛЬНО К РЕШЕНИЮ ЗАДАЧ СТРОИТЕЛЬНОГО ПРОИЗВОДСТВА

Имеется n возможных мест размещения производственных подразделений предприятия, выполняющих заданную производственную программу. Предположим, что типы всех объектов одинаковы, то есть эффект от их размещения зависит только от пункта размещения.

В целях формального описания задачи введем двоичную переменную x_i, которая равна 1, если объект размещается в пункте i и 0 в противном случае. Тогда простейшую задачу оптимального размещения можно сформулировать следующим образом.

Задача 1. Определить **{**x_i**}**, $i=\overline{1,n}$, максимизирующие

$$A(x)=\sum_i a_i x_i \qquad (1)$$

при ограничении

$$\sum_i b_i x_i \le B, \qquad (2)$$

где a_i - эффект от функционирования подразделения в пункте i; b_i - затраты на его размещение в пункте i; B - объем средств, выделенных на размещение объектов.

Рассматриваемая задача (1)-(2) является классической «задачей о ранце», методы решения, которой хорошо разработаны.

Но данная постановка задачи не учитывает ряд условий, которые могут оказаться существенными: очень часто существует ограничение на число объектов, размещаемых в одном районе. В этом случае, соответствующее ограничение имеет вид:

$$\sum_i x_i \le p, \qquad (3)$$

где p - максимальное число объектов, которые целесообразно разместить в данном районе.

Если имеется несколько районов возможного размещения ресурсов типа мощности и в k-м районе имеется множество p_k возможных пунктов размещения, то получаем систему ограничений:

$$\sum_{i\in p} x_i \le p_k, \quad k=\overline{1,\ r}, \qquad (4)$$

где p_k - максимальное число объектов, которые целесообразно размещать в k-м районе, r - число районов.

В условиях линейно-протяженного строительства существенным является условие неразмещения двух производственных единиц в близких или соседних пунктах. Для описания ограничений подобного вида удобнее всего с использованием теории графов. С этой целью будем рассматривать граф, вершины которого соответствуют пунктам размещения, а ребра соединяют соседние пункты. Если обозначить через U - множество ребер графа, описывающих расположение соседних пунктов, то в этом случае ограничения, связанные с неразмещением двух единиц ресурса типа мощности в соседних пунктах, принимают вид

$$x_i+x_j\le 1,\ (i,\ j)\in U \qquad (5)$$

В том случае если ограничения на величину финансовых средств не учитываются, рассматриваемая задача (1), (5) будет являться задачей нахождения независимого множества вершин графа, характеризуемого максимальной суммой весов a_i.

Таким образом, другую возможную постановку задачи можно сформулировать так:

Задача 2. Найти $\{x_i\}$, $i=\overline{1,n}$, максимизирующие (1) при ограничениях (2) и (4).

Возможным обобщением задач 1 и 2 является учет вида распределяемого ресурса, то есть будем считать, что распределяемые ресурсы типа мощности не являются одинаковыми (распределяемые производственные подразделения имеют специализацию). В этом случае необходимо учесть как пункта размещения, так и специализацию распределяемого объекта. Для этой цели введем еще один индекс, соответствующий типу распределяемого объекта. Тогда примем, что a_{ij} - эффект, b_{ij} - затраты, если объект i-го типа разместился в пункте j. Двоичная переменная x_{ij} равна 1, если объект типа i размещается в пункте j, и нулю в противном случае. Пусть число типов объектов равно m.

Задача 3. Определить $\{x_{ij}\}$, $i=\overline{1,m}$, $j=\overline{1,n}$, максимизирующие

$$A(x) = \sum_{i,j} a_{ij} x_{ij} , \qquad (6)$$

при ограничении

$$\sum_{i,j} b_{ij} x_{ij} \le B , \qquad (7)$$

$$\sum_{i} x_{ij} \le D_j , \quad j = \overline{1,n} . \qquad (8)$$

Соотношение (8) учитывает ограничение на число размещаемых объектов разных типов в каждом пункте.

В задаче 3 не учитывается так называемый синергетический эффект от размещения объектов разных типов в одном пункте.

С целью учета синергетического эффекта можно рассматривать размещаемые в одном месте разнотипные объекты, как комплекс. Это позволяет учесть синергетический эффект. В этом случае в ограничении (8) следует положить все $D_j = 1$, так как в одном пункте можно разместить не белее одного комплекса.

Анализируя возможность учета ограничения (4) в задаче 3, приходим к заключению, что это является более сложной задачей, так как речь идет о функционировании комплексов разных типов. Предположим, что в каждом комплексе можно выделить определяющий тип объекта, а все остальные объекты, входящие в комплекс, являются, дополняющими (это равносильно выделению ведущего процесса). Это позволяет учитывать ограничения вида (4) только по определяющему типу объектов, что существенно упрощает решение исходной задачи, так как в этом случае все сложные объекты (комплексы) разбиваются на непересекающиеся множества по определяющему типу объектов, а ограничения (4) записываются для каждого такого множества объектов.

Рассмотрим возможный подход к решению второй задачи, то есть задачи, в которой учитывались бы ограничения, связанные с нецелесообразностью размещения в одном районе (или в близких пунктах) большого числа объектов.

Рассмотрим множество P пунктов, в которых целесообразно размещать не более p объектов. Такую ситуацию достаточно удобно описывать с помощью графов. В общем случае, структура такого представления уже не будет являться деревом и поэтому необходимо применение общего метода сетевого программирования. Для этого в общем случае необходимо разделить все вершины множества Р на две, разделив при этом на две части и величины эффекта (рис. 1):

$a_i = u_i + a_i', i \in P.$ (9)

Такая операция позволяет получить граф в виде дерева, что позволяет использовать метод дихотомического программирования. В этом случае рассматриваются две подзадачи.

Первая заключается в определении $\{x_i\}$, максимизирующих

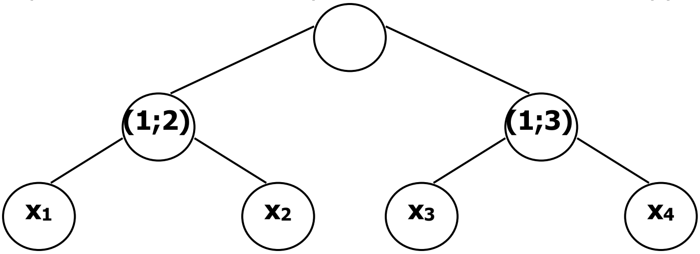


Рисунок 1. Сетевое представление задачи размещения

$$\sum_i x_i a_i',$$ (10)

при ограничении (2), а вторая - в определении $\{x_i\}$, $i \in P$, максимизирующих

$$U(x) = \sum_{i \in p} u_i x_i,$$ (11)

при ограничении (3).

Следует отметить, что решение второй задачи очевидно: для размещения необходимо выбрать пункты с наибольшими u_i.

Теперь пусть $A(u)$, $U(u)$ - значения целевых функций (10), (11) в оптимальных решениях соответствующих задач, тогда величина

$A(u) + U(u)$ (12)

является верхней оценкой для целевой функции исходной задачи (1), (2), (3). Оценочная (двойственная) задача заключается в нахождении **$\{u_i\}$**, а значит и

$a_i' = a_i - u_i,$ (13)

- минимизирующих оценку (12).

Анализируя рассматриваемую задачу (11), (13), следует отметить, что если в ее решении $x_i = 0$, то положив $u_i = u_{min}$, где u_{min}

- минимальная величина u_i среди $i \in P$ таких, что $x_i = 1$, мы не увеличим оценку (12), поскольку $U(u)$ не изменится, а $A(u)$ не увеличится. Следовательно, примем $u_i = u_{min}$ для всех $i \in P$ таких, что $x_i = 0$. Рассмотрим любое $u_i > u_{min}$ (очевидно, что $x_i = 1$ в решении задачи (11), (3)) и положим $u'_i = u_{min}$. В этом случае величина $u(a)$ уменьшается на разность $u_i - u_{min}$, а величина $A(u)$ может увеличиться не более чем на ту же разность $u_i - u_{min}$. Таким образом, оценка (12) не увеличится, и мы получили оптимальное решение оценочной задачи, в котором $u_i = u$ для всех $i \in P$. Это означает, что в оптимальном решении оценочной задачи все u_i одинаковы, то есть $u_i = u$, $i \in P$. Тем самым оценочная задача сведена к определению u, минимизирующего

$$pu + \max_x \sum_i (a_i - u)x_i, \qquad (14)$$

где $x = \{x_i\}$ удовлетворяют ограничениям (2).

Приведем описание алгоритма:

1 шаг. Принимаем $u = 0$ и решаем задачу (10), (2). Если в полученном решении выполняется условие $\sum_{i \in p} x_i \le P$, то это решение является оптимальным. Иначе переходим к шагу 2.

2 шаг. Увеличиваем u на некоторую величину $\delta > 0$ (выбор шага δ представляет собой отдельную задачу) и снова решаем задачу (10), (2). Если в полученном решении $\sum_{i \in p} x_i = P$, то это решение является оптимальным. Если в полученном решении $\sum_{i \in p} x_i > P$, то повторяем шаг 2. Если же $\sum_{i \in p} x_i < P$, то из двух решений (полученных на данном и на предыдущем шаге) берем решение с минимальной величиной оценки (13).

Таким образом, получили метод нахождения верхней оценки для целевой функции исходной задачи (1), (2), (3). Зная оценку сверху, можно применить метод ветвей и границ, либо взять решение, полученное на последнем шаге в качестве приближенного решения.

Приведем еще несколько частных случаев для задачи второго типа, когда удается разработать эффективные алгоритмы.

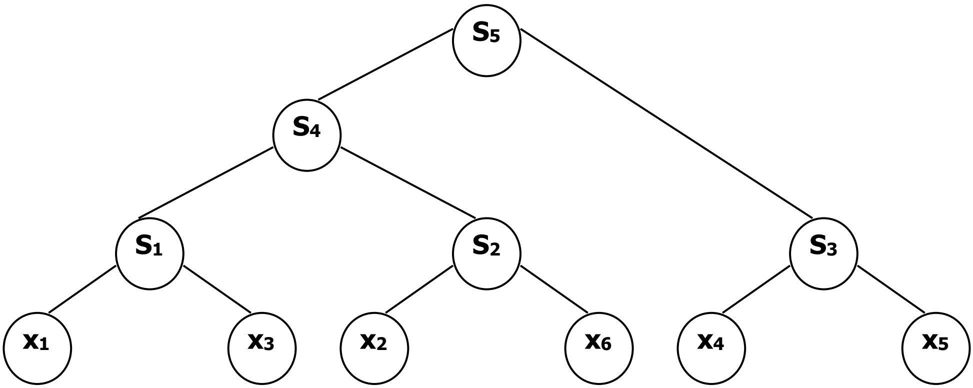


Рисунок 3. Дерево возможных решений

Рассмотрим возможность учета ограничения на размещения объектов в близких пунктах. В этом случае, как уже отмечалось выше, удобно использовать граф, ребра которого отражают нецелесообразность размещения двух объектов в соответствующих пунктах. Пример такого графа приведен на рис. 2. Особенностью такого графа является отсутствие ребер, имеющих общие вершины.

Эта особенность позволяет применить метод дихотомического программирования. По заданному графу построим структуру дихотомического представления таким образом, чтобы на нижних

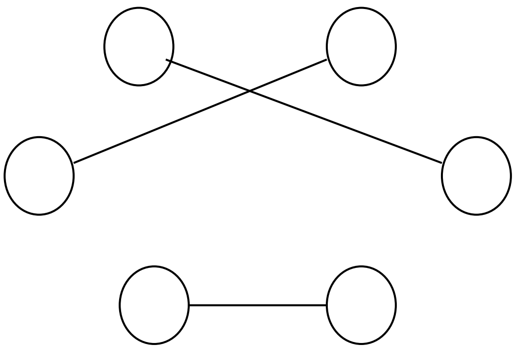


Рисунок 2. Граф учета ограничения на близость пунктов размещения объектов

уровнях дихотомического дерева находились смежные вершины графа (рис. 2.).

Таким образом, на первом шаге решаются задачи оптимизации для смежных вершин, то есть вершин, соединенных ребрами.

В том случае, когда граф, задающий возможные размещения, имеет произвольную структуру, рассмотрим *V*-множество ребер графа, являющихся паросочетанием (паросочетанием называется множество ребер, не имеющих общих вершин). Получим верхнюю оценку (1) на основе метода сетевого программирования.

Для этой цели определим два частных графа $G_1(X,V)$ и $G_2(X,W)$, где $W = U / V$. В качестве весов вершин в графе G_1 примем

$$a_{i1} = a_{j1} = \min(a_i, a_j),\ (i, j) \in V, \qquad (15)$$

а в качестве весов вершин в графе G_2

$$a_{i2} = a_i - a_{i1}, \quad i \in X. \qquad (16)$$

Обозначим A_1 максимальную величину (1) в графе G_1, A_2 - максимальную величину (1) в графе G_2.

Теорема. Выражение $A_m = A_1 + A_2$ является верхней оценкой величины (1).

В графе $G_2(X,W)$ часть вершин имеет веса, равные 0. Поэтому эти вершины можно исключить из графа вместе с инцидентными им ребрами. Оставшийся граф будем обозначать $G_2(Y,W)$, где Y - множество вершин с ненулевыми весами.

Пример. Рассмотрим граф рис. 4. Веса вершин указаны в нижних половинах соответствующих кружков.

Рассмотрим паросочетание (дуги, входящие в данное паросочетание выделены на рис. 4 двойными линиями) $V = \{(2.8);(3.4);(5.6);(1.7)\}$. Граф $G_2(Y,W)$ приведен на рис. 5.

Находим $A_1 = 10 + 8 + 11 + 15 = 44$, $A_2 = 3 + 10 + 9 + 2 = 24$, $A_m = 44 + 24 = 68$. Следует отметить, что оптимальные решения в графах G_1 и G_2 совпадают. Из основной теоремы сетевого программирования следует, что решение Q = (1, 3, 5, 8) является оптимальным решением исходной задачи.

Рассмотрим граф, в котором имеется дуга (1,5) (эта дуга показана пунктиром на рис. 4 и 5). В этом случае $A_2 = 2 + 9 + 10 = 21$, $A_m = 44 + 21 = 65$.

Так как оптимальные решения в графах G_1 и G_2 не совпадают, то величина $A = 65$ является только верхней оценкой.

Применяем метод ветвей и границ.

Для ветвления возьмем вершину 5, при этом множество всех решений разбиваем на два подмножества. В первом подмножестве x_5=1, то есть вершина 5 входит в искомое множество независимых вершин, а во втором – не входит, то есть x_5=0.

Оценка первого подмножества. Если x_5=1, то x_1=x_4=x_6=0.

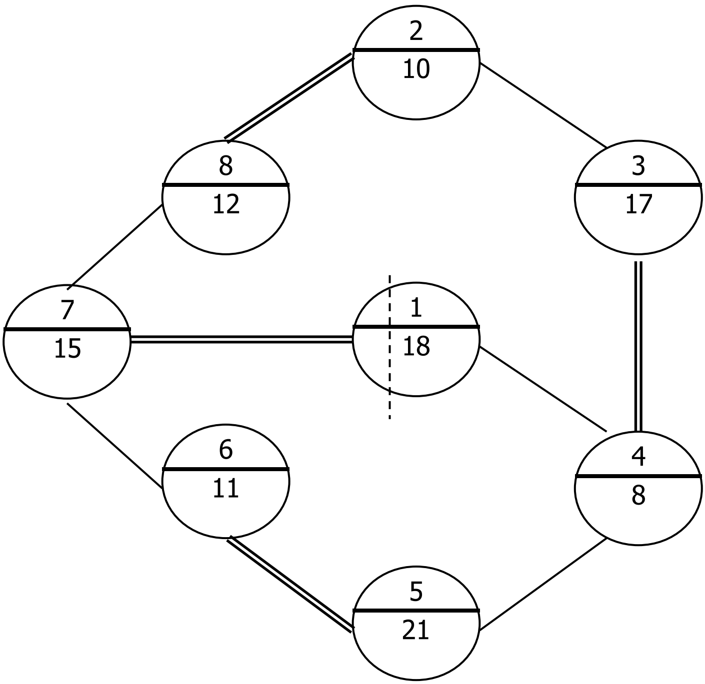


Рисунок 4. Исходный граф

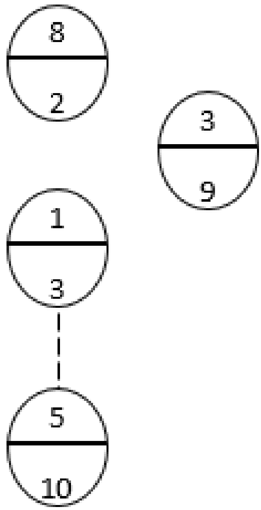


Рисунок 5. Частный граф $G_2(X,W)$

В оставшемся подграфе решение очевидно $x_3=x_7=1$, $x_1=x_2=x_8=0$. Имеем оптимальное решение в первом подмножестве с величиной A = 53.

Оценка второго подмножества. Так как $x_5=0$, то можно положить
$a_5=0$ и применить описанный выше алгоритм получения верхней оценки. Заметим, что граф G_2 будет уже другим, так как $a_{51}=a_{61}=0$, $a_{52}=0$, $a_{62}=11$. Этот граф приведен на рис. 6. Он состоит из четырех изолированных вершин.

Имеем C_1 = 33, C_2 = 25, C = 58, причем соответствующее решение
$x_1 = x_3 = x_6 = x_8 = 1$, остальные $x_i = 0$ является оптимальным решением во втором подмножестве.

Сравнивая оба решения, мы видим, что оптимальным решением исходной задачи является решение во втором подмножестве. В целях сокращения объема вычислений граф G_2 желательно иметь как можно более простым, следовательно, необходимо выбирать паросочетание с максимальным числом ребер.

Приведем описание алгоритма получения верхних оценок.

1 шаг. Строим паросо- четание с максимальным чис-

лом ребер.

2 шаг. Строим граф G_2. Если для этого графа получаем легко разрешимый случай задачи определения независимого множества с максимальной суммой весов, то определяем величину верхней оценки. В противном случае переходим к шагу 3.

3 шаг. Для графа G_2 выполняем шаги 1 и 2.

Так как при построении графа G_2 число вершин уменьшается примерно в два раза, то, как правило, после небольшого числа шагов получаем легко разрешимый случай.

На первом шаге приведем алгоритм построения паросочетания с максимальным числом ребер.

Примем $C_{ij} = 1$, если $(i, j) \in U$, где U - множество ребер графа и $C_{ij} = 0$, если $(i, j) \notin U$. Пусть W – произвольное паросочетание. Обозначим $x_{ij} = 1$, если ребро $(i, j) \in W$ и $x_{ij} = 0$ в противном случае. Получаем задачу:

$$L = \sum_{(i,j)} c_{ij} x_{ij} \to \max, \qquad (17)$$

при ограничениях

$$\sum_{j} x_{ij} \le 1, \quad i = \overline{1, n}. \qquad (18)$$

Следовательно, можно считать, что граф полный и число вершин графа четно (в противном случае всегда можно добавить одну вершину с нулевыми длинами инцидентных дуг).

Описание алгоритма.

1 шаг. Берем произвольное ребро графа, ясно, что это ребро образует максимальное паросочетание в подграфе из двух вершин.

k-шаг. Пусть получено оптимальное паросочетание в подграфе из k вершин. Добавим к подграфу еще одну вершину. Находим чередующийся цикл максимальной длины, проходящей через эту вершину. Если длина этого цикла положительна, то строим новое паросочетание с большим числом ребер.

Вычислительную сложность алгоритма можно уменьшить, если на первых шагах добавлять к подграфу по паре вершин, соединенных ребрами (если это возможно). В этом случае каждый раз будем получать паросочетание с максимальным числом ребер, поскольку в подграфе из $2k$ вершин число ребер паросочетания не может быть больше k.

На практике часто возникает необходимость определения минимально необходимого числа размещаемых единиц ресурса с целью обеспечения нормального выполнения запланированной производственной программы. Это приводит к необходимости рассматривать так называемую задачу о покрытии множества. Рассмотрим формальную постановку задачи. Пусть имеется n областей, в которых возможно размещение единиц ресурса. Топология возможного размещения может быть задана в произвольном виде: либо в виде фрагмента топографической карты с указанием квадратов возможного размещения (подобный способ задания легко трансформируется в табличную форму), либо в форме графа и т.п. Введем двоичную переменную x_j, которая принимает значение равное единице, если выбранное решение состоит в том, чтобы в j-ой области расположить единицу ресурса, и нулю в противном случае.

Каждая точка размещения будет характеризоваться некоторой двоичной величиной a_{ij}, называемой коэффициентом покрытия. Коэффициент покрытия принимает значение равное 1 в том случае, когда i-ая единица ресурса находится в зоне, покрываемой j-ой областью, и 0 в противном случае.

Учитывая, что по условиям подготовки строительного производства требуется определить минимально необходимое число единиц ресурса, предназначенных для размещения в рассматриваемом районе, то поставленная задача сводится к задаче о полном покрытии множества и может быть записана в следующем виде:

$$\sum_{j=1}^{n} x_j \to \min, \ \sum_{j=1}^{n} a_{ij} x_j \geq 1, \ i = \overline{1, \ n}, \ x_j = 0; \ 1, \ j = \overline{1, \ n}. \tag{19}$$

Утверждение 1. Решение задачи (19) эквивалентно решению булева уравнения следующего вида:

$$\prod_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_j = 1. \tag{20}$$

По условию задачи необходимо таким образом выбрать двоичные переменные, чтобы выполнялись ограничения вида

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \; i = \overline{1, \; n},$$

а целевая функция принимала минимальные значения. Ясно, что минимум целевой функции будет достигаться в том случае, когда все ограничения будут выполняться в форме равенств, так как отклонение ограничения в сторону большую единицы будет соответствовать тому факту, что появятся дополнительные переменные x_j отличные от нуля и увеличивающие значение целевой функции. Это означает, что оптимальному решению будет соответствовать решение, при котором все ограничения будут выполняться только в виде равенств. Это и будет соответствовать решению булева уравнения, определяемого соотношением вида (20).

Следовательно, задача (19) свелась к необходимости решения булева уравнения (20), которое позволяет минимизировать число переменных x_j, принимающих значение 1 (так как каждое равенство x_j = 1 означает, что в *j*-ой области расположена точка измерения). Уравнение (20) эквивалентно требованию, чтобы каждое из выражений, заключенных в скобки, равнялось 1 (в каждой зоне должен быть размещен хотя бы один объект).

Упростим выражение (20) с использованием основных соотношений булевой алгебры и в частности свойства «поглощения».

Если непосредственно раскрыть выражение (20), то получится булевский многочлен *m* степени. Полученное уравнение будет выполняться тогда, когда одно из слагаемых будет равно 1, а все остальные 0. Таким образом, каждое из слагаемых булева уравнения (20) будет соответствовать одному из возможных вариантов решения поставленной задачи.

Будем осуществлять упрощение на основе свойства «поглощения» до тех пор, пока ни одна пара скобок каждого из слагаемых в левой части рассматриваемого уравнения не будет содержать выражений с совпадающими членами. Это дает возможность понизить степени некоторых слагаемых булева уравнения. В данном случае, показатель степени у переменных x_j будет характеризовать число областей, которые можно охватить, разместив единицу производственного ресурса в рассматриваемом пункте. Понятно, что чем выше степень у переменной x_j, тем большее число пунктов можно обеспечить воздействием размещаемого в рассматриваемом пункте одного ресурса типа мощности. Учитывая, что максимально возможная степень каждого слагаемого в выражении (20) ограничена величиной *n*, то увеличение степени у одного из сомножителей слагаемого ведет к уменьшению общего числа сомножителей, входящих в данное слагаемое, что будет соответствовать уменьшению общего числа размещаемых объектов. Следовательно, в каждом слагаемом выражения (20) общее число сомножителей будет соответствовать минимально необходимому числу размещаемых объектов, которые необходимо расположить в точках, характеризуемых этими сомножителями. Таким образом, имеет место следующее утверждение:

Утверждение 2. Оптимальное решение задачи вида (19) будет определяться слагаемым наибольшей степени выражения (20). При этом минимально необходимое количество единиц ресурса *N* будет определяться следующим выражением:

$$N = n - \sum_{k=1}^{q} \left(p_k - 1\right), \qquad (21)$$

где p_k – показатель степени сомножителя; *q* – число сомножителей.

Если непосредственно раскрыть скобки, то получится многочлен *n*-ой степени, содержащий все возможные варианты решения поставленной задачи, сводящиеся к тому, что ресурсы необходимо размещать в каждой из рассматриваемых *n* точек. Сокращение числа используемых единиц ресурса возможно лишь за счет того, что по условиям размещения единиц производственных ресурсов в некоторых точках размещения возможно проведение работ, относящихся сразу к нескольким объектам. Очевидно, что такие случаи при раскрытии скобок в выражении (20) будут характеризоваться наличием показателей степени выше единицы у переменных x_j, относящимся к точкам размещения, позволяющим осуществить такое совмещение.

Рассмотрим применение приведенного алгоритма на примере. Пусть возможная схема возможного размещения единиц производственных ресурсов задана графом, приведенном на рис. 10. В этом случае задача о покрытии множества запишется в следующем виде: минимизировать соотношение

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \to \min$,

при ограничениях на размещение:

$(x_1 + x_4 + x_7) \geq 1$; $(x_2 + x_3 + x_8) \geq 1$; $(x_3 + x_4 + x_2) \geq 1$; $(x_4 + x_5 +$

$x_1+x_3)\geq 1$;

$(x_5+x_6+x_4)\geq 1$; $(x_6+x_7+x_5)\geq 1$; $(x_7+x_8+x_1+x_6)\geq 1$; $(x_8+x_2+x_7)\geq 1$;

$x_j=0,1 \quad j=1,2,\ldots 8$.

Исходя из условия, что в каждом из пунктов, возможно размещение одной единицы ресурса типа мощности, приходим к необходимости решения следующего булева уравнения:

$(x_1+x_4+x_7)\cdot(x_2+x_3+x_8)\cdot(x_3+x_4+x_2)\cdot(x_4+x_5+x_1+x_3)\cdot(x_5+x_6+x_4)\cdot(x_6+x_7+x_5)\cdot$

$\cdot(x_7+x_8+x_1+x_6)\cdot(x_8+x_2+x_7)=1$.

Используя утверждение 2, можно найти слагаемое, содержащее переменные $\boldsymbol{x_j}$ в степени, более высокой, чем первая. Если раскрыть скобки, то легко установить, что такими членами будут:

$$x_4^4\cdot x_7^3\cdot x_2+x_4^4\cdot x_7^3\cdot x_3+x_4^4\cdot x_7^3\cdot x_8+F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)=1$$

, (22)

где $F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)$ - остаток исходного многочлена степени *n*.

Анализируя выражение (22), приходим к заключению, что минимально необходимое количество единиц ресурса будет равняться трем. При этом возможно только одно решение: $x_4=1,\ \ x_7=1,\ \ x_2=1,$ то есть ресурсы необходимо размещать во втором, четвертом и седьмом пунктах.

Два других решения

$x_4=1,\ \ x_7=1,\ \ x_3=1, x_4=1,\ \ x_7=1,\ \ x_8=1$ не приемлемы, так как нарушают ограничения на близость располагаемых производственных подразделений.

К сожалению, получение полиноминальной формы (22) путем преобразования уравнения (20) по оценкам с трудом поддается программированию, если даже и осуществляется без всяких затруднений «вручную». Если же число областей и единиц распределяемого ресурса было бы значительно больше, для решения задачи потребовался бы алгоритм, предполагающий применение ПЭВМ. Можно было бы, например, заменить условие $x_i=0$ или 1 *i* =1,2, …, 12 условием $0\leq x_i\leq 1$ и искать решение путем обычного линейного программирования, которое, наверняка, не привело бы к оптимальному решению в целых числах. Однако можно было бы округлить до 1 все значения x_i < 1; полученное при этом решение, вообще говоря, могло бы оказаться неоптимальным, но зато задача решалась бы просто. Так, например, для случая, когда имеется 100 областей и 30 единиц ресурса типа мощности, для получения с помощью ПЭВМ решения, оптимального в смысле обычного линейного программирования, потребовалось бы всего несколько секунд. По аналогии с изложенными выше задачами, рассмотрим специальный алгоритм, приведенный ниже.

Предварительный шаг. Создать матрицу размерностью m х n и заполнить ее нулями. Число строк в данной матрице будет равно количеству ресурсов типа мощности, предполагаемых для размещения, то есть числу ограничений задачи, а число столбцов- числу рассматриваемых областей. Если схема предполагаемого размещения задается в виде графа, то построенная матрица будет являться аналогом матрицы смежности для заданного графа.

Задать номер строки и номер столбца введенной матрицы равный нулю, то есть i=0, j=0. Максимально возможный показатель степени также принимается равным нулю, то есть q=0.

1 шаг. Увеличиваем номер рассматриваемой строки на единицу, то есть задаем $i=i+1$.

2 шаг. Просматриваем i-ое ограничение с целью нахождения номеров переменных, входящих в состав этого ограничения. При нахождении в данном ограничении переменной с произвольным номером k в k-ом столбце j-ой строки проставляется единица.

3 шаг. Проверяется, остались еще строки исходной матрицы, подлежащие заполнению. Если строки остались, то происходит переход на 1 шаг. Если строк для последующего заполнения нет, то переходят к четвертому шагу.

4 шаг. Подсчитывается сумма по столбцам и выбирается столбец с максимальной суммой. Номер столбца и сумма фиксируется, а максимально возможный показатель степени увеличивается на единицу, то есть $q=q+1$.

5 шаг. Строки найденного столбца с максимальной суммой, в которых стоят единицы, обнуляются. Проверяется выполнение условия $q=m$. Если условие выполняется, то вычисление заканчивается, если нет, то происходит переход на 4 шаг.

Анализируя полученные решения можно сформулировать эвристическое правило, позволяющее находить пункты возможного размещения объектов при определении минимально необходимого их числа.

Эвристическое правило 1. В качестве пунктов возможного размещения производственных ресурсов выбираются вершины максимальной степени (то есть вершины, имеющие максимальное число инцидентных дуг).

Следует отметить, что полученное в этом случае решение, удовлетворяя требованиям минимальности необходимого числа размещаемых единиц ресурса, в общем случае не будет соответствовать оптимальному размещению при других критериях, например минимизации на размещение затрат на размещение или же максимизации эффекта, получаемого от данного размещения ресурсов типа мощности. Поэтому приходится решать соответствующую задачу комбинаторного программирования.

В этом случае, для получения решения, близкого к оптимальному, можно рекомендовать использование следующего эвристического правила:

Эвристическое правило 2. Для размещения ресурсов типа мощности пункты выбираются по возрастанию (убыванию) эффекта (затрат) от размещения. В том случае, если не удается разместить все ресурсы, предназначенные для размещения, то размещение необходимо начать с пункта, имеющего более низкие характеристики.

Например, если необходимо разместить 4 единицы производственного ресурса, причем схема размещения задана графом, приведенным на рис. 2.15. Используя эвристическое правило 2, в качестве исходной точки для размещения выбираем точку с максимальным значением эффекта, то есть точку 5 с эффектом 21. Такое размещение обеспечивает влияние на точки 4 и 6, которые исключаются из дальнейшего рассмотрения. Далее, наиболее выгодно использовать для размещения точку 1 с эффектом 18. Затем 3 и 8. В итоге получаем решение, совпадающее с оптимальным и дающее эффект 21+18+17+12=68.

3. ЗАДАЧА. ПРИМЕНЕНИЕ ЭЛЕМЕНТОВ ТЕОРИИ ГРАФОВ ПРИ РАСПРЕДЕЛЕНИИ РЕСУРСОВ ТИПА МОЩНОСТИ

Все многообразие ресурсов, используемых в производстве можно разделить на два принципиально различных класса: складируемые или материально-технические ресурсы и не складируемые, иначе называемые ресурсами типа мощности. Существует довольно значительное количество моделей, описывающих распределение материально-технических ресурсов, но вот распределению ресурсов второго типа библиография гораздо меньше, хотя в условиях строительного производства, когда фронты работ могут быть разнесены н пространстве на значительные расстояния, такая задача представляется весьма актуальной.

Задачи размещения связаны с решением проблем наилучшего расположения в определенных регионах таких систем обслуживания, как торговые центры, посты пожарной охраны, фабрики, аэропорты, склады

и т. д. Рассмотрим такие задачи размещения, для которых областью допустимых точек размещения центров обслуживания является некоторый граф, т. е. эти центры могут располагаться в какой-либо вершине или на какой-либо дуге графа.

В задачах размещения есть два основных критерия оценки качества размещения: минимизация максимального расстояния и минимизация суммы расстояний. Соответственно имеем и две основные задачи.

Рассмотрим следующее задание: найти максимальный поток и минимальный разрез в транспортной сети, используя алгоритм Форда–Фалкерсона (алгоритм расстановки пометок). Построить граф приращений. Проверить выполнение условия максимальности построенного полного

потока. Источник – вершина 1, сток – вершина 8.

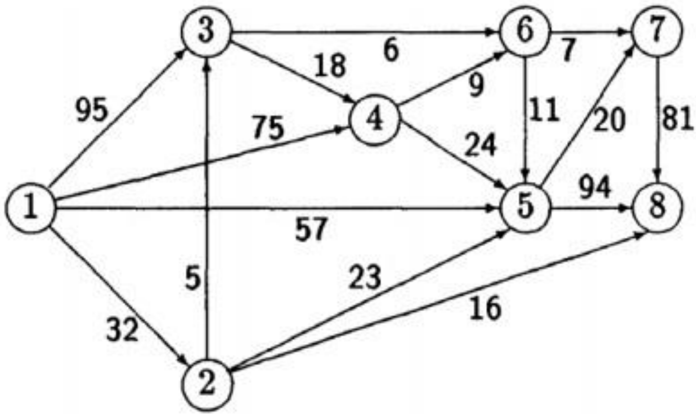


Рисунок 7. Исходная модель транспортной сети

Решение: С помощью алгоритма Форда-Фалкерсона найдем наибольший поток из 1 в 8 (см. рис 7).

Шаг 1. Выбираем произвольный поток, например, 1-3-6-7-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 6. Уменьшаем пропускные способности дуг этого потока на 6, насыщенную дугу 3-6 вычеркиваем.

Шаг 2. Выбираем произвольный поток, например, 1-4-5-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 24. Уменьшаем пропускные способности дуг этого потока на 24, насыщенную дугу 4-5 вычеркиваем.

Шаг 3. Выбираем произвольный поток, например, 1-5-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 57. Уменьшаем пропускные способности дуг этого потока на 57, насыщенную дугу 1-5 вычеркиваем.

Шаг 4. Выбираем произвольный поток, например, 1-2-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 16. Уменьшаем пропускные способности дуг этого потока на 16, насыщенную дугу 2-8 вычеркиваем.

Шаг 5. Выбираем произвольный поток, например, 1-2-5-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 13. Уменьшаем пропускные способности дуг этого потока на 13, насыщенную дугу 5-8 вычеркиваем.

Шаг 6. Выбираем произвольный поток, например, 1-2-5-7-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 3. Уменьшаем пропускные способности дуг этого потока на 3, насыщенную дугу 1-2 вычеркиваем.

Шаг 7. Выбираем произвольный поток, например, 1-4-6-7-8. Его пропускная способность равна минимальной из всех пропускных способностей входящих в него дуг, то есть 1. Уменьшаем пропускные способности дуг этого потока на 1, насыщенную дугу 6-7 вычеркиваем.

Шаг 8. Суммарный поток 6+24+57+16+13+3+1+8=128. Величина разреза 6+9+24+57+32=128.

Процедура распределения не складируемых ресурсов может рассматриваться как задача нахождения максимальности построенного полного потока. Причем физическая сущность располагаемых объектов, как правило, оказывает влияние на вид ограничений и критерии оптимальности, выбираемые для оценки размещения. При рассмотрении ряда возможных подобных задач, учитывая, что в качестве объекта размещения рассматриваются производственные подразделения строительной организации.

Следует отметить, что полученное в этом случае решение, удовлетворяя требованиям минимальности необходимого числа размещаемых единиц ресурса, в общем случае не будет соответствовать оптимальному размещению при других критериях, например минимизации на размещение затрат на размещение или же максимизации эффекта, получаемого от данного размещения ресурсов типа мощности. Поэтому приходится решать соответствующую задачу комбинаторного программирования. В этом случае, для получения решения, близкого к оптимальному, можно рекомендовать использование следующего эвристического правила: для размещения ресурсов типа мощности пункты выбираются по возрастанию (убыванию) эффекта (затрат) от размещения. В том случае, если не удается разместить все ресурсы, предназначенные для размещения, то размещение необходимо начать с пункта, имеющего более низкие характеристики.

4. ТЕМЫ РЕФЕРАТОВ

1. Достоинства и недостатки применения теории графов в строительном производстве.
2. Отечественная и зарубежная история использования теории графов для целей планирования в строительстве.
3. Примеры приложения теории графов.
4. Основные понятия теории графов.
5. Экстремальные пути и контуры на графах. Задача о кратчайшем пути.
6. Экстремальные пути и контуры на графах. Задача о ране,задача поиска контура минимальной длины и задача поиска контура минимальной средней длины.
7. Экстремальные пути и контуры на графах. Путь максимальной эффективности, путь максимальной эффективности с учетом штрафов.
8. Псевдопотенциальные графы.
9. Задача о максимальном потоке, задача о назначении.
10. Задачи календарно-сетевого планирования и управления. Задачи определения продолжительности проекта.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бурков В.Н., Заложнев А.Ю., Новиков Д.А. Теория графов в управлении организационными системами. М.: Синтег, 2004. – 124 с.
2. Емеличев В.А,, Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М.: Наука, 2001. – 384 с.
3. Новиков Д.А. Сетевые структуры и организационные системы. М.: ИПУ РАН, 2003. – 102 с.
4. Баркалов С.А., Тельных В.Г., Ключникова О.В. Оценка работоспособности инженерных инфраструктур при произвольной топологии. Вестник Воронежского гос. технического университета. 2011. Том 7. № 7. – С. 183-188.
5. Баркалов С.А., Ключникова О.В. Определение эксплуатационных характеристик инженерных сетей произвольной топологии. Системы управления и информационные технологии. 2011. Т. 45. № 3.1. С. 117-122.
6. Шипилов В.Н., Ключникова О.В. Организационно-технологическая модель распределения ресурсов типа мощности. Вестник Воронежского государственного технического университета. 2011. Т. 7. № 9. С. 157-163.
7. Берж К. Теория графов и ее применения. М.: Иностранная литература, 1962. –319 с.
8. Бурков В.Н., Горгидзе И.А., Ловецкий С.Е. Прикладные задачи теории графов. Тбилиси: Мецниереба, 1974. – 234 с.
9. Бурков В.Н., Ланда Б.Д., Ловецкий С.Е., Тейман А.И., Чернышев В.Н. Сетевые модели и задачи управления. М.: Советское радио, 1967. – 144 с.
10. Оре О. Теория графов. М.: Наука, 1968. – 352 с.