УДК.65.012

Баркалов С.А., Бурков В.Н., Гилязов Н.М. Методы агрегирования в управлении проектами. М.: ИПУ РАН, 1999 – 55 с.

В работе рассматриваются задачи календарного планирования проектов (комплексов операций), связанные, в основном, с оптимальным распределением ограниченных ресурсов. Развиваемый в работе подход основан на идее агрегирования, то есть представления проекта (или его частей) в виде одной или нескольких операций. Рассматриваются задачи построения агрегированных описаний и задачи оптимального распределения ресурсов в проектах, представленных агрегированными операциями.

Рецензент: д.т.н. В. В. Кульба

Текст препринта воспроизводится в том виде, в котором представлен авторами.

Утверждено к печати Редакционным советом Института.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Управление проектами стало широко применяться начиная с 60-х годов. В настоящее время методология и методы управления проектами являются эффективным инструментом при реализации самых различных проектов от строительства дома до организации спортивных соревнований, реформирования предприятий и т.д. Центральной задачей в управлении проектами является задача формирования плана реализации проекта или задача календарного планирования. Как правило, эта задача связана с распределением ограниченных ресурсов по операциям проекта. Поэтому задачу календарного планирования называют часто задачей оптимального распределения ресурсов в проекте (комплексе операций). Отметим сразу, что эта задача в общем случае относится к сложным многоэкстремальным или комбинаторным задачам оптимизации. Точные эффективные методы получены только для небольшого числа частных постановок или для задач небольшой размерности. Поэтому для решения реальных задач календарного планирования развиваются два подхода.

Первый подход основан на использовании эвристических алгоритмов. Первая группа эвристических алгоритмов использует некоторые эвристические правила приоритетности операций при возникновении конфликтной ситуации, связанной с ограниченностью ресурсов. Вторая группа эвристических алгоритмов использует идею локальной оптимизации, то есть улучшения некоторого начального решения. Второй подход основан на идее агрегирования, то есть уменьшения числа операций проекта путем замены нескольких операций (подпроектов) одной операцией. Полученный агрегированный проект, как правило, допускает более эффективные методы решения (в силу меньшей размерности). Полученное агрегированное решение затем дезагрегируется в календарный план исходного проекта.

Метод агрегирования естественным путем вписывается в иерархическую организационную структуру системы управления проектом. Действительно, на верхнем уровне решения принимаются на основе агрегированных описаний руководителем всего проекта, а на нижних уровнях - руководителями подпроектов.

В работе рассматривается второй подход. Дается определение агрегирования и описание методов построения агрегированных операций. Рассматриваются методы решения агрегированных задач.

ГЛАВА 1. Методы построения агрегированных операций

1.1. Постановка задачи календарного планирования

Проектом называется некоторый процесс изменений, то есть не рутинный, не повторяющийся процесс, требующий специальных методов проектного управления. Для организаций, которые в основном занимаются реализацией проектов, рекомендуется и специальная форма управления (проектно-ориентированные организации).

Проект обычно представляют как некоторое множество операций (комплекс операций). Операция это процесс, требующий затрат времени и ресурсов. Для формального описания операции необходимо задать ее объем W и зависимость скорости (интенсивности) операции от количества ресурсов, ее выполняющих. Будем обозначать эту зависимость

$$w = f(u(t)),$$

где u(t) - вектор ресурсов в операции в момент t.

Пусть $t_н$ - момент начала операции, а t_0 - момент ее окончания. Тогда объем операции удовлетворяет условию

$$W = \int_{t_н}^{t_o} f\left[u(t)\right] dt .$$

Как правило, ресурсы участвуют в операции в определенных соотношениях, называемых набором ресурсов. Набор ресурсов можно представить в виде

$$u_j = \beta_j v, \quad j = \overline{1, m},$$

где m - количество видов ресурсов, v - интенсивность набора, β_j, - количество ресурса j-го вида на единицу мощности набора.

В качестве величины интенсивности набора, как правило. берется вид ресурса, который является основным (определяющим). Например, количество людей, выполняющих работу, определяет требуемое количество материалов, инструмента, рабочей одежды и т.д. Для определяющего ресурса, очевидно, $\beta = 1$. Ограничение на ресурсы теперь можно записать в следующем виде:

$$\sum_{i=1}^{n} \beta_{ij} u_i(t) \le N_j(t), \quad j = \overline{1, m}$$

где n - число операций комплекса, $N_j(t)$ - количество ресурсов j-го вида в момент t.

Ограничения на ресурсы часто связаны с ограниченностью финансов. Если обозначить c_j- стоимость единицы ресурсов j-го вида в единицу времени, а S(t) - объем финансирования в момент t, то ограничения, связанные с финансированием, принимают вид

$$\sum_{i=1}^{n} \sum_{j=1}^{m} c_j \beta_{ij} v_i(t) \le S_j(t).$$

Это ограничения типа мощности. Если ограничены средства, выделенные на проект, то получаем ограничения типа затрат:

$$\sum_{i=1}^{n} S_i \le Q,$$

где

$$S_i = \sum_{j=1}^{m} c_j \int_{t_{iн}}^{t_{io}} v_i(t)\, dt.$$

Наконец, если задан график Q(t) поступления ресурсов на проект (график финансирования проекта), то получаем следующие ограничения на ресурсы:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} c_j \beta_{ij} \int_{0}^{t} v_i(\tau)\, d\tau \le Q(t).$$

Задача оптимального распределения ресурсов (задача календарного планирования) заключается в определении распределения ресурсов $\overline{v}(t) = \{v_i(t)\}$ такого, что все операции комплекса выполнены за минимальное время (задача оптимальною быстродействия), либо потери, связанные с задержкой времени реализации комплекса или ряда его операций, минимальны (минимизация упущенной выгоды). Критерий минимизации упущенной выгоды, обычно, рассматривается в виде

$$\Phi = \sum_{i=1}^{n} q_i (t_i - d_i), \; t_i \geq d_i,$$

где d_i - желательный срок завершения i-й операции, q_i - потери в единицу времени при завершении операции позже d_i. Заметим, что и настоящее время в условиях дефицита финансовых средств. насыщенности рынка и материальных, и трудовых ресурсов. ограничивающим фактором являются финансовые ресурсы. Это позволяет рассматривать задачи календарного планирования, как задачи распределения ресурсов одного вида (финансовых ресурсов). Такой подход тем более обоснован, поскольку он позволяет сконцентрировать внимание именно на особенностях решения задач календарного планирования на основе агрегирования. Поэтому в дальнейшем, если это не оговорено особо, будем считать, что все операции выполняются ресурсами одного вида (финансовыми ресурсами). Будем обозначать далее

$$u(t) = \sum_{j=1}^{m} c_j \beta_{ij} v_i (t)$$

количество финансовых ресурсов на i-ой операции в момент t и, соответственно, f,(u) - скорость i-ой операции в зависимости от количества ресурсов.

1.2. Построение модели операции

Для решения задач календарного планирования необходимо, в первую очередь, получить описание всех операций, то есть определить объем каждой операции и зависимость f(u) скорости операции от количества ресурсов. Дело в том, что на практике, как правило, известны продолжительности операций при различных количествах ресурса на ней, то есть зависимости $\tau(v)$. Если операции выполняются с фиксированным уровнем ресурсов (v принимает только одно значение) или с постоянным уровнем ресурсов (количество ресурсов в процессе выполнения операции не меняется), то проблем не возникает. Действительно, в этом случае

$$\tau(v) = \frac{W}{f(v)} \quad \text{или} \quad f(v) = \frac{W}{\tau(v)},$$

и скорость операции определяется с точностью до параметра W (при известной зависимости $\tau(v)$ объем W может выбираться произвольно). Ситуация становится сложнее, если операция выполняется с переменным уровнем ресурсов.

Пример 1.1. Пусть операция состоит из двух частей, каждая из которых может выполняться при двух уровнях ресурсов. Обозначим τ_{ij} - продолжительность i-ой части при j-ом уровне ресурса, i = 1, 2; j = 1, 2;

$$T_1 = \tau_{11} + \tau_{21}$$

$$T_2 = \tau_{12} + \tau_{22}$$

- продолжительности операций при выполнении обеих частей, соответственно при первом и втором уровнях ресурсов. Пусть

$$\tau_{11} = 2,\ \tau_{21} = 3,\ \tau_{12} = 1,\ \tau_{22} = 2,\ T_1 = 5,\ T_2 = 3.$$

Примем объем операции W = 15. Тогда при T_1 = 5 средняя скорость операции w_1 = 3, а при T_2 = 3 – w_2 = 5. Пусть операция выполняется сначала при первом уровне ресурсов в течение 2 дней, а затем при втором уровне ресурсов. Тогда очевидно, что операция будет закончена за 4 дня, так как $\tau_{11} + \tau_{22} = 4$. Определим, однако, момент завершения операции на основе средних скоростей w_1 и w_2;. За два дня будет выполнено при скорости w_1 = 3 x(2) = 6 ед. объема операции. Оставшиеся 15 - 6 = 9 ед. объема при скорости w_2 = 5 будут выполнены за $^9/_5$ = 1,8 дня. В целом операция будет завершена за 3,8 дня, что меньше истинного срока t = 4. Если выполнять операцию сначала при втором уровне ресурса (w_2 = 5) в течение одного дня, а затем при первом (w_1 = 3), то операция, как легко проверить, завершится за $4^1/_3$ дня, что больше 4. Ошибки в определении времени завершения операции объясняются тем, что скорость завершения операции является средней величиной. Для уменьшения ошибки следует соответствующим образом выбрать объемы частей операции. Так, если взять объем первой части x_1 = 7,5, и объем второй части x_2 = 7,5, то ошибки не возникнет. Действительно, при выполнении первой части со скоростью w_1 = 3, она будет завершена за $^{7,5}/_3$ = 2,5 дня. Вторая часть при скорости w_2 = 5 будет завершена за $^{7,5}/_5$ = 1,5 дня, что в сумме даст 4 дня.

Рассмотренный пример показывает, насколько важно выбрать объемы частей операции при ее выполнении с переменной интенсивностью.

Рассмотрим задачу выбора объемов частей операции в общем случае. Пусть операция состоит из n частей, каждая из которых может выполняться при m различных уровнях ресурсов. Обозначим τ_{ij} -

продолжительность i-ой части при j-ом уровне ресурсов, y_i - объем i-ой части операции, Q_j - продолжительность операции при j-м уровне ресурсов. Обозначим далее x_{ij} = 1, если i-ая операция выполняется j-м уровнем ресурсов, x_{ij} = 0 в другом случае. Тогда продолжительность операции определяется на основе ее описания (то есть на основе объемов частей $\{y_i\}$ и продолжительностей $\{Q_j\}$) и будет равна

$$Q = \sum_{i,j} y_i Q_j x_i \,, \tag{1.2.1}$$

а истинная продолжительность

$$T = \sum_{i,j} x_{ij} \tau_{ij} \,. \tag{1.2.2}$$

Относительная ошибка описания операции составит

$$\delta = \left| 1 - \frac{Q}{T} \right| . \tag{1.2.3}$$

Поставим задачу определить объемы частей $\{y_i\}$ и продолжительности $\{Q_j\}$ так, чтобы ошибка δ была минимальной. Очевидно, что если существуют числа $\{y_i\}$ такие, что $\tau_{ij} = y_i T_j$, где $T_j = \sum_i \tau_{ij}$, то ошибка $\delta = 0$. Запишем условие (1.2.3) в виде

$$\begin{gathered} -\delta \le 1 - \frac{Q}{T} \le \delta \\ (1-\delta)T \le Q \le (1+\delta)T \,. \end{gathered} \tag{1.2.4}$$

Условия (1.2.4) можно представить в виде двух неравенств:

$$\sum_i \min_j \left((1+\delta)\tau_{ij} - y_i Q_j \right) \ge 0 \,,$$

$$\sum_i \min_j \left(y_i Q_j - (1-\delta)\tau_{ij} \right) \ge 0 \,.$$

Наконец, введя новые переменные:

$$u_i = \min_j \left((1+\delta)\tau_{ij} - y_i Q_j \right),$$

$$v_i = \min_j \left(y_i Q_j - (1-\delta)\tau_{ij} \right),$$

приведем задачу к следующему виду:

$$\delta \to \min ,$$

$$\sum_i u_i \ge 0 , \qquad (1.2.5)$$

$$\sum_i v_i \ge 0 ,$$

$$u_i = (1+\delta)\tau_{ij} - y_i Q_j, \quad i = \overline{1,n}, \; j = \overline{1,m} ,$$

$$v_i = y_i Q_j - (1-\delta)\tau_{ij} \quad i = \overline{1,n}, \; j = \overline{1,m} . \qquad (1.2.5)$$

Это задача нелинейного программирования. Если зафиксировать $\{y_ш\}$ или $\{Q_j\}$, то она становится задачей линейного программирования. Поэтому задачу (1.2.5) можно решать как последовательность задач линейного программирования, фиксируя сначала значения $\{Q_j\}$

(например, взяв $Q_j = \sum_i \tau_{ij}, \; j = \overline{1,m}$), а затем $\{y_i\}$ и т.д.

Заметим, что в ряде случаев величины $\{Q_j\}$ выбираются из других соображений. Так, например, мы хотим получить описание операции с линейной зависимостью скорости операции от количества ресурсов, то есть $f(w) = u = {}^{W}/_{Q}$. В этом случае $Q = {}^{W}\!/_{u_j}$, где u_j - количество ресурсов, соответствующее j-му уровню. Если принять $W = 1$, то $Q_j = {}^{1}\!/_{u_j}$ становится известным и задача (1.2.5) является задачей линейного программирования.

Рассмотрим приближенный метод построения описания операции. В его основе лежит идея приближенного представления чисел τ_{ij} в виде y_iQ_j. Относительная ошибка такого представления составит

$$\delta = \max_{i,j}\left|1 - \frac{y_iQ_j}{\tau_{ij}}\right|. \qquad (1.2.6)$$

Представим (1.2.6) в виде

$$(1-\delta_i)\max_j \frac{\tau_{ij}}{Q_j} \le y_i \le (1+\delta_i)\min_j \frac{\tau_{ij}}{Q_j} \qquad (1.2.7)$$

или

$$\delta_i = \frac{\max\limits_j \frac{\tau_{ij}}{Q_j} - \min\limits_j \frac{\tau_{ij}}{Q_j}}{\max\limits_j \frac{\tau_{ij}}{Q_j} + \min\limits_j \frac{\tau_{ij}}{Q_j}}\ , \qquad (1.2.8)$$

$$\delta = \max_i \delta_i\ .$$

Легко показать, что задача минимизации δ сводится к задаче минимизации $\varepsilon = \max\limits_i \varepsilon_i$, где

$$\varepsilon_i = \frac{\max\limits_j \tau_{ij}w_j}{\min\limits_j \tau_{ij}w_j}, \quad w_j = \frac{1}{Q_j}. \qquad (1.2.9)$$

Условие (1.2.9) легко сводится к виду

$$\frac{w_j}{w_k} \ge \frac{1}{\varepsilon}\max_i \frac{\tau_{ik}}{\tau_{ij}} = \frac{1}{\varepsilon} q_{kj}. \qquad (1.2.10)$$

Обозначим $\mathbf{l}n\ w_j = \lambda_j$, $\mathbf{l}n\ \varepsilon = \eta$, $\mathbf{l}n\ q_{kj} = \mathbf{l}_{kj}$. Тогда система (1.2.10) сведется к виду

$$\lambda_j - \lambda_k \ge \mathbf{l}_{kj} - \eta\ , \quad j, k = \overline{1,m}\ . \qquad (1.2.11)$$

Необходимо определить минимальную $\eta \ge 0$, при которой система (1.2.11) имеет решение.

Определим полный граф с m вершинами и длинами дуг ($\mathbf{l}_{kj}$- η). Как известно, система (1.2.11) имеет решение, если в графе отсутствуют контуры положительной длины.

Алгоритм решения задачи.

1 шаг. Положим η_0 = 0. Определим контур μ_1 положительной длины. Если таких контуров нет, то задача решена ($\eta = \eta_0 = 0$). Если $L(\mu_1) > 0$, то полагаем длины дуг равными $\mathbf{l}_{ij}$- η_1, где

$$\eta_1 = \frac{L(\mu_1)}{n(\mu_1)}$$

($\eta(\mu)$ - число дуг контура μ).

k-й шаг. Определяем контур μ_k положительной длины, при длинах дуг $\mathbf{l}_{ij}$- η_{k-1}. L_kЕсли $L_k = L(\mu_k) \le 0$, то $\eta = \eta_{k-1}$. Если $L(\mu_k) > 0$, то определяем

$$\eta_k = \frac{1}{n_k} L_k ,$$

где $n_k = n(\mu_k)$, и переходим к следующему шагу.

За конечное число шагов будет получена величина η такая, что в графе отсутствуют контуры положительной длины при длинах дуг $\mathbf{l}_{ij}$ - η. Соответствующие потенциалы вершин $\{\lambda_i\}$ и величина η определяют оптимальное решение задачи (1.2.11), а значит (1.2.10) и (1.2.7). Определив δ_i, мы можем найти y_i и, следовательно, получить полное описание операции.

Пример 1.2. Возьмем $\tau_{11} = 2$, $\tau_{21} = 3$, $\tau_{12} = 1$, $\tau_{22} = 2$. Имеем

$$q_{12} = \max\left[\frac{\tau_{11}}{\tau_{12}};\frac{\tau_{21}}{\tau_{22}}\right] = \max\left(\frac{2}{1};\frac{3}{2}\right) = 2 ,$$

$$q_{21} = \max\left[\frac{\tau_{12}}{\tau_{11}}; \frac{\tau_{22}}{\tau_{21}}\right] = \max\left(\frac{1}{2}; \frac{2}{3}\right) = \frac{2}{3}.$$

В данном случае задачу можно решить не переходя к логарифмам, определяя вместо длины контура его усиление. Усиление контура (1,2,1) равно $^4/_3 > 1$. Следовательно, $\varepsilon = {}^2\!/\!_{\sqrt{3}}$, $w_1 = 1$, $w_2 = \sqrt{3}$. Имеем $Q_1 = 1$, $Q_2 = {}^1\!/\!_{\sqrt{3}}$,

$$\delta_1 = \frac{\max\left(\frac{Q_1}{\tau_{11}}; \frac{Q_2}{\tau_{12}}\right) - \min\left(\frac{Q_1}{\tau_{11}}; \frac{Q_2}{\tau_{12}}\right)}{\max\left(\frac{Q_1}{\tau_{11}}; \frac{Q_2}{\tau_{12}}\right) + \min\left(\frac{Q_1}{\tau_{11}}; \frac{Q_2}{\tau_{12}}\right)} \approx 0{,}09$$

$$\delta_2 = \frac{\max\left(\frac{Q_1}{\tau_{21}}; \frac{Q_2}{\tau_{22}}\right) - \min\left(\frac{Q_1}{\tau_{21}}; \frac{Q_2}{\tau_{22}}\right)}{\max\left(\frac{Q_1}{\tau_{21}}; \frac{Q_2}{\tau_{22}}\right) + \min\left(\frac{Q_1}{\tau_{21}}; \frac{Q_2}{\tau_{22}}\right)} \approx 0{,}06.$$

Соответственно получаем

$$y_1 = (1 - \delta_1) \cdot \max\left(\frac{Q_1}{\tau_{11}}; \frac{Q_2}{\tau_{12}}\right) = \frac{0{,}92}{\sqrt{3}}$$

$$y_2 = (1 - \delta_2) \cdot \max\left(\frac{Q_1}{\tau_{21}}; \frac{Q_2}{\tau_{22}}\right) = \frac{0{,}94}{3}.$$

1.3. Идеальное агрегирование

Агрегирование, то есть представление сложной модели (описываемой большим числом параметров) в упрощенном (агрегированном) виде (описываемой небольшим числом параметров) не только эффективный метод решения задач большой размерности, но едва ли не единственный подход к принятию решений на высших уровнях управления. И дело здесь не в том, что ограничены наши возможности в решении задач большой размерности. Главная причина агрегированного описания сложных моделей в том, что руководитель (лицо, принимающее решение) способен принимать эффективные решения, оперируя только небольшим числом существенных факторов (порядка 7-8 факторов).

Отсюда следует, что подход к решению задач большой размерности на основе построения агрегированных моделей адекватен иерархическому построению организационных систем. Очевидно, что упрощенное описание является приближенным (ошибка агрегирования), однако, это упрощение окупается повышением эффективности принятия решения на основе агрегированных моделей. Большой интерес представляют случаи идеального агрегирования, то есть агрегирования с нулевой ошибкой. Дадим формальные определения агрегирования и ошибки агрегирования в задачах календарного планирования.

Определение 1. Агрегированием комплекса операций называется его представление в виде комплекса с меньшим (как правило, значительно меньшим) числом операций.

Это определение обобщает определение агрегирования, данное в [I], где под агрегированием понималось представление комплекса операций в виде одной операции.

Пусть задан класс M возможных ограничений N(t) на количество ресурсов, выделенных для реализации проекта. Обозначим $T_m[N(t)]$ - минимальное время реализации проекта при графике использования ресурсов N(t), а $T_a[N(t)]$ - минимальное время реализации агрегированного проекта при том же графике N(t). Разность

$$\varepsilon[N(t)] = \left| 1 - \frac{T_a[N(t)]}{T_m[N(t)]} \right| \qquad (1.3.1)$$

определяет ошибку агрегирования при заданном графике N(t). Ошибку агрегирования для всех возможных графиков $N(t) \in M$ будем оценивать выражением

$$\varepsilon = \max_{N(t) \in M} \varepsilon[N(t)] \qquad (1.3.2)$$

Определение 2. Агрегирование с нулевой ошибкой называется идеальным.

Приведем примеры идеального агрегирования комплекса операций.

Пример 1.3. Рассмотрим комплекс из n независимых операций. Обозначим W_i - объем i-ой операции, $f_i(u)$ - скорость i-ой операции. Пусть f_i - вогнутые функции. В этом случае, как доказано В.Н. Бурковым [2], все операции начинаются одновременно и заканчиваются также одновременно, причем скорости операций удовлетворяют соотношениям

$$f_i[u_i(t)] = w(t)W_i, \quad i = \overline{1, n}, \qquad (1.3.3)$$

где w(t) определяется из уравнения

$$\sum_{i=1}^{n} \xi_i[w(t)\, W_i] = N(t),$$

ξ_i – функция, обратная f_i.

Момент завершения комплекса определяется из условия

$$\int_0^T w(t)\, dt = 1 . \qquad (1.3.4)$$

Определим скорость агрегированной операции $w_э(t)$ как решение уравнения (1.3.4), а ее объем примем за $W_э = 1$. Очевидно, что для любого N(t) имеем $T_a[N(t)] = T_m[N(t)]$, то есть $\varepsilon = 0$.

Пример 1.4. Рассмотрим комплекс из n последовательных операций объема W_i $i=\overline{1,n}$ и скоростями $f_i(u_i) = \beta_i f(u_i)$. Определим агрегированную операцию с объемом $W_a = \sum_{i=1}^{n} \frac{W_i}{\beta_i}$ и скоростью $w_a = f(u)$. Нетрудно показать, что для любого N(t) имеет место $T_a[N(t)] = T_m[N(t)]$, то есть $\varepsilon = 0$.

На основании этих примеров можно утверждать, что если комплекс состоит из однородных операций (операций, скорости которых удовлетворяют соотношениям $f_i = \beta_i f$, где f вогнутые функции) и имеет последовательно параллельную структуру, то такой комплекс допускает идеальное агрегирование в одну операцию. Существует класс зависимостей $f_i(u_i)$, при которых возможно идеальное агрегирование любого комплекса операций. Это так называемые степенные зависимости вида

$$f_i(u) = u_i^{\alpha}, \ \alpha < 1, \ i=\overline{1,n}$$

Для случая степенных зависимостей доказано, что существует агрегированное представление комплекса в виде одной операции объема $W_э$ и со скоростью $f = u^{\alpha}$ такое, что для любого N(t) имеет место $T_m[N(t)] = T_a[N(t)]$ [3]. Таким образом, задача сводится к определению объема агрегированной операции (этот объем назван эквивалентным объемом комплекса).

Известны несколько методов определения эквивалентного объема. Первый метод основан на решении задачи распределения ресурсов при заданном уровне ресурсов N. Если $T_{min}(N)$ - минимальное время реализации проекта, то эквивалентный объем проекта определяется выражением

$$W_э = T_{min}(N) \cdot T^{\alpha}$$

Второй метод основан на решении задачи минимизации затрат при заданном сроке реализации проекта. При этом зависимость затрат на i-ую операцию от ее продолжительности определяется выражением

$$s_i(\tau_i) = \frac{W_i^{1/\alpha}}{\tau_i^{\frac{1-\alpha}{\alpha}}}, \quad i = \overline{1, n}.$$

Если $s_{min}(T)$ – величина минимальных затрат, то эквивалентный объем проекта определяется выражением

$$W_э = s_{min}^{\alpha} T^{1-\alpha}.$$

Опишем еще один метод определения эквивалентного объема, основанный на геометрической аналогии. Для этого введем понятие размерности комплекса операций.

Определение 3. Размерностью комплекса операций называется максимальное число независимых операций.

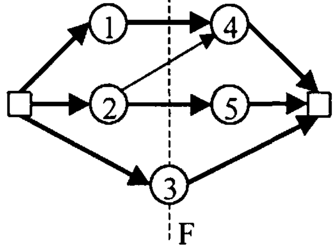

Рис. 1.1.

На рис. 1.1 приведен пример комплекса, имеющего размерность 3. Множество состояний комплекса размерности m можно изобразить в виде некоторой области m-мерного фазового пространства. Для этого определим множество M путей, покрывающих сеть, то есть таких, что каждая вершина сети принадлежит хотя бы одному пути. Как известно, минимальное число таких путей равно размерности комплекса [3]. Обозначим Q_i - множество вершин сетевого графика, принадлежащих пути μ_i (если вершина принадлежит нескольким путям, то оставляем ее только в одном из множеств Q_i). Пути μ_i выделены на рис. 1.1 толстыми дугами. Поставим в соответствие каждому пути μ_i координатную ось y_i фазового пространства, а последовательности вершин $k \in \mu_i$ последовательность отрезков длины W_k на оси y_i (рис. 1.2). Точка 0 соответствует начальному состоянию комплекса (ни одна операция не начата), а точка A - конечному состоянию (все операции завершены). Чтобы отобразить зависимости между операциями различных путей, «вырежем» из параллелограмма на рис. 1.2 соответствующие области. Полученная область полностью описывает множество возможных состояний комплекса. Любому процессу выполнения операций соответствует траектория, соединяющая т.0 с т. A и проходящая в области возможных состояний. Определим расстояние между любыми двумя точками y_1 и y_2 следующим образом:

$$\rho\left(y^1, y^2\right) = \left(\sum_{j=1}^{m} \left| y_j^1 - y_j^2 \right|^{1/\alpha} \right)^{\alpha}$$

В [3] показано, что эквивалентный объем комплекса операций равен длине кратчайшей траектории, соединяющей т.0 с т. A.

Опишем алгоритм определения кратчайшей траектории, использующий геометрическую аналогию.

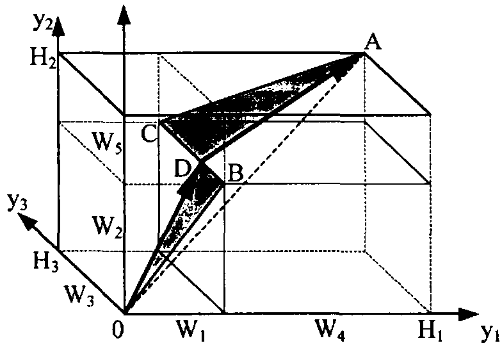

Рис. 1.2.

Проводим прямую, соединяющую т.0 с т. А. Эту прямую можно описать параметрически в виде

$$y_i(t)=t\cdot H_i,$$

где $H_i = \sum_{k\in\mu_i} W_k$ (см. рис. 1.2). Определяем минимальное t, начиная с которого прямая выходит за пределы области возможных состояний. Геометрически это означает, что кратчайшая траектория должна проходить через отрезок ВС на рис. 1.2. Как известно, в этом случае треугольники OBD и ACD должны быть подобными. Это позволяет определить координаты точки D на рис. 1.2 или, соответственно, фронт работ F на рис. 1.1.

Далее процедура повторяется. Определяем минимальное t, начиная с которого траектория выходит за пределы области возможных состоянии, далее определяем соответствующую точку на границе области (из условия подобия треугольников) и т.д., пока не получим траекторию, состоящую из отрезков прямых, целиком лежащую в области возможных состояний.

На втором этапе происходит корректировка полученной траектории. А именно, рассматриваем три последовательных точки излома траектории и корректируем, если это необходимо, положение средней точки из условия подобия треугольников.

Пример 1.5. Рассмотрим комплекс из 5 операций (рис. 1.3, нижние числа в вершинах равны объемам операций). Пусть a = '/z, то есть $f_i(u_i)= \sqrt{u_i}$, $i=\overline{1,5}$.

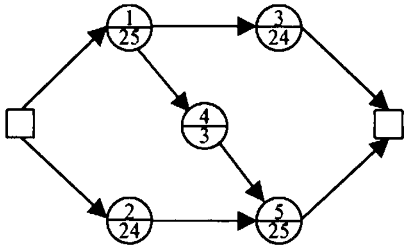

Рис. 1.3.

Задачу определения кратчайшей траектории удобно решать в параллельной системе координат (рис. 1.4). В этой системе любой точке фазового пространства состояний соответствует фронт F, то есть линия, проходящая через соответствующие координаты этой точки на параллельных координатных осях y_1, y_2 и y_3. Из рис. 1.4. видно, что кратчайшая траектория обязательно пройдет через фронты F_1 и F_2, причем между фронтами траектория будет отрезками прямой линии. Рассмотрим три последовательных фронта $F_н$, F_1 и F_2. Из условия подобия треугольников имеем уравнение

$$\frac{x}{24-x}=\frac{25}{\sqrt{9+(24-z)^2}}. \qquad (1.3.5)$$

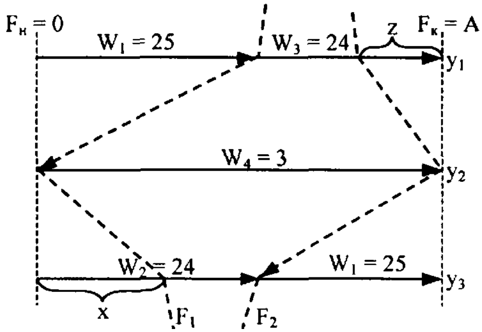

Рис. 1.4.

Рассматривая три последовательных фронта F_1, F_2 и $F_к$, получаем аналогично

$$\frac{z}{24-z}=\frac{25}{\sqrt{9+(24-x)^2}}\,. \qquad (1.3.6)$$

Получили систему двух нелинейных уравнений с двумя неизвестными. Ее решение можно получить на основе итерационного алгоритма. Берем начальное значение x_0 и из уравнения (1.3.6) определяем z_0. На основе z_0 определяем x_1 из уравнения (1.3.5), затем z_1 и т.д. Для определения начального значения x_0 рассматриваем три фронта – F_0, F_1 и F_k. Имеем из условия подобия треугольников

$$\frac{x}{49-x}=\frac{25}{\sqrt{24^2+9}}\,.$$

Из этого уравнения находим x_0:

$$x_0=\frac{25\cdot 49}{25+\sqrt{3^2+24^2}}\,.$$

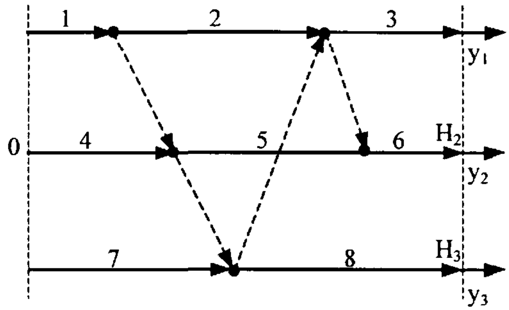

Рис. 1.5.

Изображение комплекса операций в параллельной системе координат позволяет в ряде случаев выписать выражение для эквивалентного объема в аналитическом виде. На рис. 1.5 представлен комплекс из 8 операций размерности 3. При этом объемы операций на каждой координатной оси y_i умножены на нормирующий множитель α_i, так что фронты F_0 и F_k являются параллельными прямыми. Пунктирные стрелки на рис. 1.5 показывают зависимости между операциями, принадлежащими различным координатным осям. Пусть все пунктирные стрелки имеют направление слева направо, то есть начало дуги лежит левее ее конца. В этом случае прямая линия, соединяющая начальный фронт с конечным, является допустимой траекторией, и поэтому эквивалентный объем равен

$$W_э = \left(\sum_{i=1}^{m} H_i^{1/\alpha} \right)^{\alpha} = \left[(W_1 + W_2 + W_3)^{1/\alpha} + (W_4 + W_5 + W_6)^{1/\alpha} + (W_7 + W_8)^{1/\alpha} \right]^{\alpha}.$$

1.4. Методы приближенного агрегирования линейных моделей

Линейная зависимость скорости операции от количества ресурсов широко применяется на практике. Без ограничения общности линейную зависимость можно представить в виде

$$f_i(u_i) = \begin{cases} u_i, & u_i \le a_i \\ a_i, & u_i \ge a_i \end{cases}. \tag{1.4.1}$$

Действительно, в общем случае линейная зависимость имеет вид

$$f_i(u_i) = \begin{cases} k_i \cdot u_i, & u_i \le a_i \\ k_i \cdot a_i, & u_i \ge a_i \end{cases}.$$

Как известно, описание операции инвариантно к умножению скорости и объема операции на любое положительное число. Поэтому, положив $\tilde{f}_i = \frac{1}{k_i} f_i$, $\tilde{W}_i = \frac{1}{k_i} W_i$, мы получим зависимость (1.4.1)

Метод агрегирования рассмотрим сначала для случая независимых операций. Обозначим $\tau_i = W_i / a_i$ минимальную продолжительность i-ой операции,

$$T = \max_i \tau_i$$

минимальную продолжительность комплекса из n независимых операций. Положим

$$A = \sum_{i=1}^{n} \frac{W_i}{T} = \frac{W_э}{T}.$$

Величины A и T примем за параметры агрегированной операции. Эквивалентный объем агрегированной операции определяется как сумма объемов операций, а зависимость скорости агрегированной операции от количества ресурсов N имеет вид (1.4.1), то есть

$$f_э(N)=\begin{cases} N, & N \le A \\ A, & N \ge A \end{cases}.$$

Для обоснования предложенного метода агрегирования докажем, что при $N(t) \le A$ ошибка агрегирования равна 0. Действительно, момент T окончания агрегированной операции определяется из уравнения

$$\int_0^{T_m} N(t)dt = W_э.$$

Положим

$$u_i(t) = \frac{N(t)}{A}\cdot\frac{W_i}{T}.$$

Заметим, что $u_i(t) \le a_i$ для всех i. Поэтому момент завершения i-ой операции определяется из уравнения

$$\int_0^{T_i} u_i(t)dt = W_i = \frac{W_i}{A\cdot T}\int_0^{T_i} N(t)dt.$$

Подставляя $T = W_э/A$ получаем, что $T_i = T_m$. Более того, если $N(t) = N$, то есть количество ресурсов не меняется во времени, то ошибка агрегирования равна 0 при любом $N > A$. Действительно, продолжительность агрегированной операции $T_m = W_э/A = T$, то есть совпадает с продолжительностью комплекса операций. Учитывая стремление к выравниванию уровня ресурсов, выполняющих каждый комплекс операций, можно утверждать, что предложенный метод агрегирования будет давать практически небольшие ошибки агрегирования.

Пусть теперь комплекс операций состоит из последовательности из п операций. В данном случае идеальное агрегирование, то есть представление комплекса в виде другого комплекса с меньшим числом операций, возможно только в случае, если несколько соседних операции имеют равные значения a_i. Если несколько соседних операций имеют близкие значения a_i то возможно приближенное агрегирование с ошибкой, зависящей от степени близости a_i.

Рассмотрим задачу определения относительной ошибки приближения a_i одним значением A для всех n операций. Относительная ошибка определяется выражением

$$\varepsilon = \max_i \left| 1 - \frac{A}{a_i} \right| . \tag{1.4.2}$$

Представляя (1.4.6) в виде

$$1 - \varepsilon \le 1 - \frac{A}{a_i} \le \varepsilon ,$$

получаем после несложных преобразований

$$(1 - \varepsilon) \max_i a_i \le A \le (1 + \varepsilon) \min_i a_i .$$

Минимальная относительная ошибка определяется выражением

$$\varepsilon = \frac{a_{max} - a_{min}}{a_{max} + a_{min}} , \tag{1.4.3}$$

а окончательное значение

$$A = (1 - \varepsilon) a_{max} = (1 + \varepsilon) a_{min} = \frac{2 a_{min} a_{max}}{a_{min} + a_{max}} . \tag{1.4.4}$$

Определим (n+1)-вершинный граф с вершинами 0, 1, 2, … , n. Вершины i, j (i < j) графа соединим дугой (i, j), длина которой равна относительной ошибке при агрегировании (j – i) операций (i+1), (i+2), … , j в одну. Обозначим эту длину $\varepsilon_{i,j}$.

Пусть задана допустимая ошибка приближения ε. Поставим задачу определить агрегированный комплекс с минимальным числом операций так, чтобы относительная ошибка при замене нескольких последовательных операций одной агрегированной операцией не превышала е. Для решения этой задачи достаточно исключить из графа все дуги i, длины которых превышают ε, и в оставшемся частичном графе определить путь, соединяющий начальную вершину 0 с конечной вершиной n и имеющий минимальное число дуг. Эта задача легко решается алгоритмами поиска кратчайших путей в графе.

Пример 1.6. Комплекс состоит из шести последовательных операций, данные о которых приведены в таблице 1. Длины дуг ε_{ij} графа, определяемые на основе выражения (1.4.3), приведены в таблице 2.

Пусть ε = 0,2. Граф с дугами, длины которых не превышают 0,2, приведен на рис. 1.6.

Таблица 1.

i	**1**	**2**	**3**	**4**	**5**	**6**
$\mathbf{W_i}$	12	16	10	15	8	12
$\mathbf{a_i}$	3	4	5	5	4	6
$\mathbf{T_i}$	4	4	2	3	2	2

Таблица 2.

i \ j	**1**	**2**	**3**	**4**	**5**	**6**
o	0	1/7	1/4	1/4	1/4	1/3
1		0	1/9	1/9	1/9	1/5
2			0	0	1/9	1/5
3				0	1/9	1/5
4					0	1/5
5						0

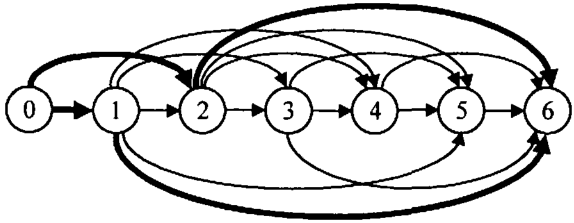

Рис. 1.6.

Толстыми дугами выделены пути с минимальным числом дуг. Как видно из рисунка, имеются два пути - (0, 2, 6) и (0, 1, 6)- из двух дуг. Они определяют два оптимальных варианта агрегирования комплекса. В первом варианте объединяются операции 1 и 2 в одну операцию с величиной А, определяемой по выражению (1.4.4):

$$A_1 = A(1,2) = \frac{2 \cdot 3 \cdot 4}{7} = \frac{24}{7} = 3\frac{3}{7},$$

а также объединяются операции 3, 4, 5 и 6 в одну операцию с величиной

$$A_2 = A(3,4,5,6) = \frac{2 \cdot 6 \cdot 4}{10} = \frac{24}{5} = 4{,}8.$$

Во втором варианте операция 1 остается, а объединяются операции 2, 3, 4, 5 и 6 в одну операцию с величиной

$$A_2 = A(2,3,4,5,6) = \frac{2 \cdot 6 \cdot 4}{10} = 4{,}8.$$

Если взять ε = 0,12, то агрегирование в две операции невозможно. Действительно, соответствующий граф, приведенный на рис. 1.7, не имеет путей из двух дуг.

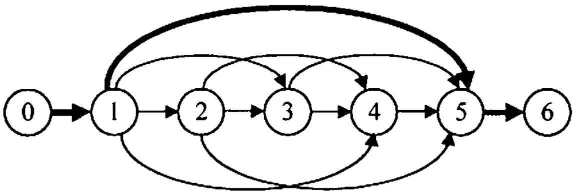

Рис. 1.7.

Оптимальный вариант всего один. В этом варианте объединяются операции (2, 3, 4, 5) в одну агрегированную операцию со значением

$$A_2 = A(2,3,4,5) = \frac{2 \cdot 4 \cdot 5}{9} = \frac{40}{9} = 4\frac{4}{9}$$

Рассмотрим теперь произвольный комплекс операций. Возьмем продолжительности всех операций равными минимальным значениям τ_i, и определим критический путь в сети. Обозначим через $T_{кр}$ длину критического пути. Поставим задачу выравнивания ресурсов, то есть определения максимально равномерного графика ресурсов, требуемых для выполнения комплекса операций за время $T_{кр}$. Эту задачу удобнее рассматривать, когда сетевой график изображен в форме «операции-дуги», то есть дуги сетевого графика соответствуют операциям, а вершины - событиям (моментам завершения одной или нескольких операций). Задача выравнивания ресурсов эффективно решается при заданном упорядочении 0, 1,2, ... , m моментов завершения событий (m+1 - число событий). Эта задача рассматривалась еще в шестидесятых годах [4]. В [5] для ее решения предлагалась гидродинамическая модель (переток жидкости между сообщающимися сосудами), а в [4] задача решалась методом квадратичного программирования. Мы рассмотрим геометрический подход к решению задачи. Обозначим A_k, - общий объем операций, которые должны быть выполнены в первых k интервалах, B_k - общий объем операций, которые могут быть выполнены в первых k интервалах. Для определения A_k необходимо определить правосдвинутый график использования ресурсов, то есть график, соответствующий началу всех операций в наиболее поздние моменты времени. Для определения B_k необходимо определить левосдвинутый график использования ресурсов, то есть график, соответствующий началу всех операций в наиболее ранние моменты времени. На основе этих графиков определяются значения A_k и B_k $k = \overline{1, m}$.

Построим на плоскости графики зависимости A_k и B_k от соответствующих моментов совершения событий наиболее поздних поздних $T_k^п$ и наиболее ранних $T_k^р$. Область между двумя графиками определяет множество возможных состояний комплекса операций, определяемых как объем операций, выполненный к соответствующему моменту t. Любому процессу выполнения операций комплекса соответствует траектория, соединяющая точку 0 с точкой К на рис. 1.8. Тангенс угла наклона этой траектории определяет количество ресурсов в соответствующий момент времени. Очевидно, что решению задачи выравнивания уровня ресурсов соответствует кратчайшая траектория, соединяющая точку 0 с точкой К. Точки излома траектории определяют моменты изменения количества ресурсов. Поиск кратчайшей траектории можно осуществлять непосредственно на плоскости с помощью карандаша и линейки.

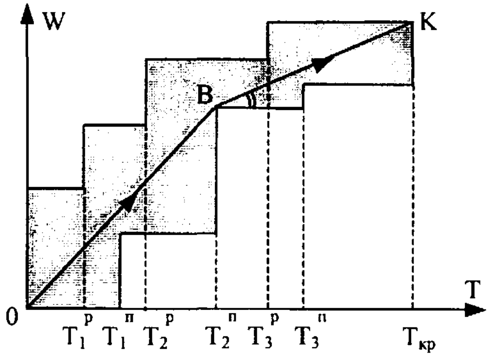

Рис.1.8.

Пример 1.7. Рассмотрим комплекс из четырех операций (рис 1.9).

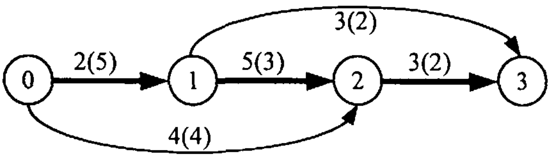

Рис. 1.9.

Числа у дуг на рис. 1.9 равны a_i, а числа в скобках - минимальным продолжительностям τ_i. Критический путь выделен толстыми дугами. График зависимостей $\{A_k\}$ и $\{B_k\}$ приведены на рис. 1.10.

Из рисунка видно, что кратчайшая траектория проходит через точку B. Таким образом, мы имеем оптимальный график использования ресурсов, состоящий из двух участков. На отрезке [0, 5]

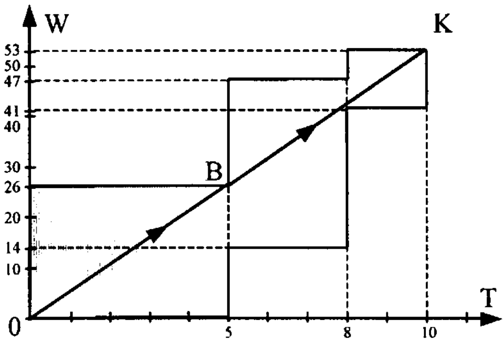

Рис. 1.10.

уровень ресурсов равен $N_1 = {}^{26}/_5 = 5{,}2$, а на отрезке [5, 10] – $N_2 = {}^{27}/_5 = 5{,}4$. Поэтому комплекс можно представить в виде двух последовательных агрегированных операции. Первая агрегированная операция объединяет две операции - (0, 1) и (0, 2), имеет объем $W_i = 26$ и величину $A_1 = 5{,}2$. Вторая агрегированная операция объединяет три операции - (1, 2), (1, 3) и (2, 3) имеет объем $W_2 = 27$ и величину $A_2 = 5{,}4$. Допуская относительную ошибку порядка 2% можно весь комплекс заменить одной агрегированной операцией объемом $W = 53$ и величиной $A_2 = 5{,}3$.

Пример 1.8. Рассмотрим комплекс из шести операций, рис. 1.11.

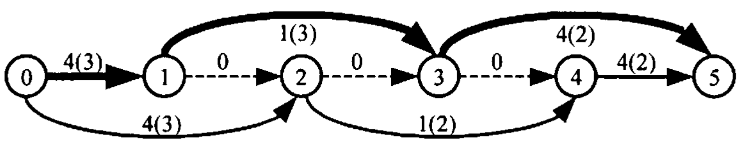

Рис. 1.11.

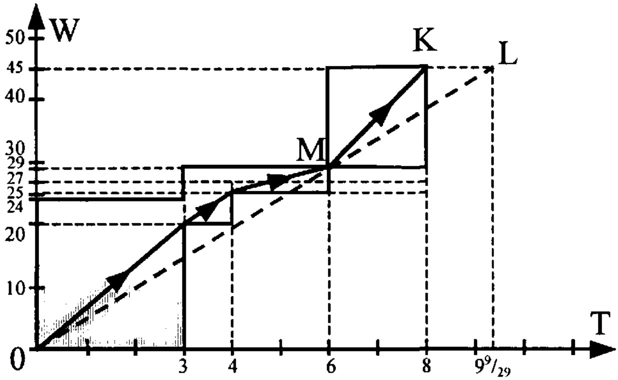

Рис. 1.12.

График {Ак}, {Вк} приведен на рис. 1.12. В данном случае удается уменьшить число операций за счет агрегирования только до четырех. Однако, если требование равномерной загрузки ресурсов на комплексах является существенным, можно добиться этого, увеличив минимальную продолжительность агрегированного комплекса. Пунктирная прямая OL на рис. 1.12, проходящая через точку M, соответствует агрегированию комплекса в одну операцию с параметрами $T_{min}= 9\,{}^{9}/_{29}$, $A = 4^{5}/_{6}$.

ГЛАВА 2. Оптимальное распределение ресурсов в агрегированных комплексах

Задача оптимального распределения ресурсов для агрегированных комплексов, по сути дела, не отличается от обычных задач распределения ресурсов. Особенность только в том, что число операций в агрегированном комплексе невелико, что позволяет в ряде случаев применять точные алгоритмы решения задач. Так, например, как будет показано ниже, при заданном упорядочении событий сети задача оптимального распределения ресурсов сводится к задаче выпуклого программирования, что позволяет решать задачу путем перебора возможных упорядочений событий. В этой главе рассматриваются как известные, так и новые методы решения задачи распределения ресурсов для агрегированных комплексов.

2.1. Сети с упорядоченными событиями

Пусть события сети пронумерованы согласно их упорядочению. Обозначим R_s – множество операций, которые могут выполняться в s-ом интервале (в интервале между (s-1) и s-ым событиями, $s = \overline{1, r}$), Q_i – множество интервалов, в которых может выполняться i-ая операция, x_{is} – объем i-ой операции, выполняемый в s-ом интервале, $\Delta_s(z_s)$, где $z_s = \{x_{is}:\ i \in R_s\}$, минимальная продолжительность s-го интервала, как функция z_s. Продолжительность комплекса, очевидно, равна

$$T(z) = \sum_{s=1}^{r} \Delta_s(z_s) \tag{2.1.1}$$

Теорема 1. [2] T(z) – выпуклая функция z.

Задача. Определить $\{x_{is} \geq 0,\ i=\overline{1,n},\ s=\overline{1,r}\}$, удовлетворяющие условиям:

$$\sum_{s\in Q_i} x_{is} = W_i, \quad i=\overline{1,n} \tag{2.1.2}$$

и минимизирующие T(z).

Согласно теореме 1 это задача выпуклого программирования.

Рассмотрим применение приведенных выше общих результатов к задаче распределения дискретных ресурсов, когда количество ресурсов на каждой операции фиксировано. На практике такие задачи возникают при формировании календарных планов работы специализированных бригад. Обозначим, как и раньше, τ_i – продолжительность i-ой операции, a_{ij} – фиксированное количество ресурсов j-го вида на i-ой операции. Будем обозначать множество независимых операций комплекса через вектор $x = (x_1, x_2, \dots, x_n)$, где $x_i = 1$, если i-ая операция принадлежит множеству, и $x_i = 0$ в противоположном случае. Вектор x, удовлетворяющий условиям

$$\sum_{i=1}^{n} x_i a_{ij} \leq N_j, \quad j=\overline{1,m}, \tag{2.1.3}$$

будем называть допустимым вектором. Пусть $x^1, x^2, \dots, x^q$ – множество допустимых векторов. Обозначим через u_s продолжительность интервала, в котором выполняются операции, соответствующие вектору x^s.

Задача. Определить u_s, $s=\overline{1,q}$, удовлетворяющие условиям

$$\sum_{s\in Q_i} x_i^s u_s = \tau_i, \quad i=\overline{1,n}$$

и минимизирующие $T = \sum_{s=1}^{q} u_s$.

Это задача линейного программирования, в которой матрица ограничений задается косвенно, в виде (2.1.3). Пусть x^1, x^2, … , x^n – базисные вектора некоторого начального решения. Обозначим через y^i вектор x, для которого $x_i = 1$, $x_j = 0$, $j \neq i$. Выразим y^i через базисные векторы. Пусть

$$y^i = \sum_{j=1}^{n} c_{ij} x^j \tag{2.1.4}$$

и обозначим $b_i = \sum_{j=1}^{n} c_{ij}$.

Рассмотрим следующую вспомогательную задачу: определить вектор x, удовлетворяющий (2.1.3) и максимизирующий

$$c = \sum_{i=1}^{n} x_i b_i \; . \tag{2.1.5}$$

Если в оптимальном решении этой задачи $c \leq 1$, то начальное решение оптимально. В противном случае оптимальное решение вспомогательной задачи определяет вектор, который нужно ввести в базис согласно процедуре симплекс-метода.

Замечание 1. Так как допустимый вектор определяет множество независимых операций, то задача (2.1.3), (2.1.5) решается для каждого множества R_s независимо.

Замечание 2. Для исключения зацикливания процедуры следует запоминать вектора, исключаемые из базиса в случае, если им соответствует u = 0, до тех пор, пока не будет исключен вектор со значением u > 0.

Задача (2.1.3), (2.1.5) является задачей линейного целочисленного программирования с переменными, принимающими значения 0, 1, и в общем случае не имеет эффективных методов решения. Рассмотрим несколько практически важных случаев, когда задача (2.1.3), (2.1.5) принимает более простой вид.

I) Пусть операции разбиты на m типов так, что операции j-го типа выполняются ресурсами j-го вида в количестве a_i ($i = \overline{1, n}$). Обозначим через R_{js} множество операций j-го типа, которые могут выполняться в s-ом интервале. В этом случае задача (2.1.3), (2.1.5) разбивается на несколько независимых подзадач, - для каждого $R_{js} \neq \varnothing$:

$$c_{js} = \sum_{i \in R_{js}} b_i x_i \to \max ,$$

$$\sum_{i \in R_{js}} a_i x_i \le N_j . \qquad (2.1.6)$$

Задача (2.1.6) известна как «задача о ранце» и решается достаточно эффективно методом динамического программирования.

Начальное решение оптимально, если

$$c_{js} \le 1,\ \ j = \overline{1, m} ,\ s = \overline{1, r}$$

$$c_{js} = 0, \text{ если } R_{js} = \varnothing.$$

Задача (2.1.6) решается элементарно, если все $a_i = 1$. Очевидно, в этом случае c_{js} равно сумме N_j положительных максимальных b_i (или просто сумме всех положительных b_i, если их число меньше, чем N_j).

Пример 2.1. Рассмотрим комплекс из пяти операций (рис. 2.1), данные о которых приведены в таблице 3. (номера операций указаны у дуг на рисунке).

Таблица 3.

i	**1**	**2**	**3**	**4**	**5**
t_i	2	4	6	5	2
a_i	6	5	4	4	6

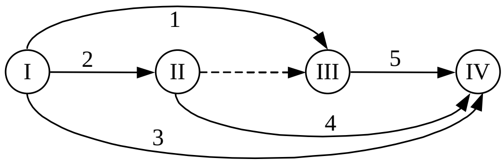

Рис. 2.1.

Все операции выполняются ресурсами одного вида, количество которых равно N = 11.

1. Получение начального решения. Для получения начального решения мы применим эвристический алгоритм, согласно которому приоритет имеет операция с меньшим значением позднего момента начала $t_i^п$ (это правило называется распределением по степени критичности операций). В случае одинаковых $t_i^п$ начинаем любую операцию. Значения $t_i^п$ приведены ниже.

i	1	2	3	4	5
$t_i^п$	0	1	1	2	5

Применяя вышеприведенное эвристическое правило, получим следующее решение:

$$x^1 = (1, 1, 0, 0, 0),\ u_1 = 2$$
$$x^2 = (0, 1, 1, 0, 0),\ u_2 = 2$$
$$x^3 = (0, 0, 1, 1, 0),\ u_3 = 4$$
$$x^4 = (0, 0, 0, 1, 1),\ u_4 = 1$$
$$x^5 = (0, 0, 0, 0, 1),\ u_5 = 1$$

Продолжительность комплекса составляет

$$T = \sum_{s=1}^{5} u_s = 10 .$$

Выразим векторы y^i через базисные векторы:

$$y^5 = x^5, \qquad b_5 = 1$$

$$y^4 = x^4 - x^5, \qquad b_4 = 0$$

$$y^3 = x^3 - x^4 + x^5, \qquad b_3 = 1$$

$$y^2 = x^2 – x^3 + x^4 - x^5, \qquad b_2 = 0$$

$$y^1 = x^1 – x^2 + x^3 – x^4 + x^5, \qquad b_1 = 1$$

Получаем следующую задачу о ранце:

$$x_1 + x_3 + x_5 \rightarrow \max ,$$

$$6x_1 + 4x_3 + 7x_5 \leq 1.$$

Задача разделяется на три независимых подзадачи:

$$\begin{gathered} x_1 + x_3 \rightarrow \max , \\ 6x_1 + 4x_3 \leq 11; \end{gathered} \tag{2.1.7}$$

$$\begin{gathered} x_3 \rightarrow \max , \\ 4x_3 \leq 11; \end{gathered} \tag{2.1.8}$$

$$\begin{gathered} x_3 + x_5 \rightarrow \max , \\ 4x_3 + 7x_5 \leq 11. \end{gathered} \tag{2.1.9}$$

Оптимальное решение задачи (2.1.7): $x_1 = x_3 = 1$, задачи (2.1.8): $x_3 = 1$, задачи (2.1.9): $x_3 = x_5 = 1$. Решение задач (2.1.7) и (2.1.9) имеют одну и ту же величину с = 2.Поэтому вводим в базис любой вектор, например

$$x^6 = (0, 0, 1, 0, 1) = y^3 + y^5 = x^3 – x^4 + 2x^5.$$

Определяем согласно процедуре симплекс-метода

$$u_6 = \min\left(\frac{u_3}{1}; \frac{u_5}{2}\right) = \frac{1}{2}$$

и исключаем вектор x^5. Новое решение

$$u_1^1 = u_1 = 2; \;\; u_2^1 = u_2 = 2; \;\; u_3^1 = u_3 - \frac{1}{2} = 3{,}5;$$

$$u_4^1 = u_4 + \frac{1}{2} = 1{,}5; \;\; u_6^1 = \frac{1}{2}; \;\; T_1 = 9{,}5.$$

Заменяя вектор

$$x^5 = ½(x^6 – x^3 + x^4)$$

в формулах для y^i, мы получим

$$
\begin{array}{ll}
y^1 = x^1 - x^2 + {}^1/_2x^3 - {}^1/_2x^4 + {}^1/_2x^6, & b_1 = {}^1/_2 \\
y^2 = \quad x^2 - {}^1/_2x^3 + {}^1/_2x^4 - {}^1/_2x^6, & b_2 = {}^1/_2 \\
y^3 = \quad {}^1/_2x^3 - {}^1/_2x^4 + {}^1/_2x^6, & b_3 = {}^1/_2 \\
y^4 = {}^1/_2x^1 + {}^1/_2x^3 - {}^1/_2x^6, & b_4 = {}^1/_2 \\
y^5 = \quad -{}^1/_2x^3 + {}^1/_2x^4 + {}^1/_2x^6, & b_5 = {}^1/_2
\end{array}
$$

Новые задачи о ранце имеют вид:

$$½(x_1 + x_2 + x_3) \to \max,$$

$$6x_1 + 5x_2 + 4x_3 \le 11;$$

$$½(x_2 + x_3 + x_4) \to \max,$$

$$5x_2 + 4x_3 + 4x_4 \le 11;$$

$$½(x_3 + x_4 + x_5) \to \max,$$

$$5x_2 + 4x_3 + 4x_4 + 7x_5 \le 11.$$

Все три задачи имеют оптимальное значение с = 1. Поэтому полученное решение является оптимальным.

Последовательность векторов x^s полученного оптимального решения не является единственной. Этим в ряде случаев можно воспользоваться для того, чтобы найти решение с минимальным числом прерываний операций. В работе [3] показано, что задача минимизации числа прерываний операций сводится к задаче коммивояжера. В нашем примере число прерываний равно 1 и уменьшить его нельзя, поскольку при выполнении комплекса без прерываний операций продолжительность его будет целым числом. Заметим, что начальное решение, полученное на основе эвристического правила является оптимальным решением без прерываний операций, поскольку продолжительность комплекса в этом решении T = 10 является ближайшим целым числом, которое больше минимальной продолжительности $T_1 = 9,5$.

Как уже отмечалось выше, описанный метод дает оптимальное решение при заданном упорядочении событий сети. Если возможных упорядочений несколько, то необходимо решить задачу при каждом из них и выбрать лучшее решение. Однако, в ряде случаев можно судить об оптимальности полученного решения без перебора других упорядочений событий. Для этого необходимо решить задачу о ранце, не предполагая упорядоченности всех событий сети. Если в результате решения этой задачи будет получен вектор со значением c > 1, то следует решить задачу при новом упорядочении, при котором этот вектор будет допустимым. В нашем примере достаточно рассмотреть следующую задачу о ранце:

$$\tfrac{1}{2}(x_1 + x_2 + x_3 + x_4 + x_5) \to \max,$$

$$6x_1 + 5x_2 + 4x_3 + 4x_4 + 7x_5 \le 11.$$

Легко видеть, что в оптимальном решении этой задачи c = 1, так что лучшего решения не существует, какое бы упорядочение событий мы ни взяли.

Для иллюстрации изменения упорядочения событий рассмотрим следующий пример.

Пример 2.2. Рассмотрим тот же сетевой график, что и в примере 2.1 с данными, приведенными в таблице 4.

Таблица 4.

i	**1**	**2**	**3**	**4**	**5**
t_i	4	5	3	2	6
a_i	6	5	4	4	6

Начальное решение в данном случае будет следующим:

$$x^1 = (1, 1, 0, 0, 0), \ u_1 = 4$$

$$x^2 = (0, 1, 1, 0, 0), \ u_2 = 1$$

$$x^3 = (0, 0, 1, 0, 1), \ u_3 = 2$$

$$x^4 = (0, 0, 0, 1, 1), \ u_4 = 2$$

$$x^5 = (0, 0, 0, 0, 1), \ u_5 = 2$$

Продолжительность комплекса T = 11. Действуя, как и в примере 2.1, мы получим следующую задачу о ранце:

$$x_2 + x_5 \to \max ,$$

$$5x_2 + 6x_5 \le 11.$$

При заданном упорядочении событий эта задача распадается на две независимые задачи и, очевидно, каждая из них будет иметь величину c = 1. Следовательно, полученное начальное решение является оптимальным при заданном упорядочении.

Заметим, однако, что если изменить очередность событий II и III, то операции 2 и 5 становятся независимыми. В этом случае оптимальное решение задачи о ранце будет следующим:

$$x_2 = x_5 = 1$$

со значением c = 2. Поэтому вводим в базис вектор

$$x^6 = y^2 + y^5 = x^2 - x^3 + 2x^5,$$

$$u_6 = \min\left(\frac{u_2}{1}; \frac{u_5}{2}\right) = 1.$$

Исключаем из базиса любой из векторов, например вектор x^5. Имеем

$$u_1^1 = u_1 = 4;\ \ u_2^1 = u_2 - u_6 = 0;\ \ u_3^1 = u_3 + u_6 = 3;$$
$$u_4^1 = u_4 = 2;\ \ u_6^1 = 1;\ \ T_1 = 10.$$

Повторяя процедуру, убеждаемся, что полученное решение является оптимальным.

Предложенный метод решения задачи позволяет также определить, на какую величину необходимо увеличить количество ресурса для того, чтобы было возможно дальнейшее уменьшение продолжительности комплекса. Для этого достаточно определить уровень ресурсов N, при котором решение задачи о ранце **c** больше чем 1, что позволит ввести в базис новый вектор.

2.2. Оптимальность эвристического правила по степени критичности операций

Правило распределения ресурсов по степени критичности операций (правило СК) является одним из самых распространенных эвристических правил. Согласно этому правилу, приоритет операции определяется величиной позднего момента начала (чем меньше поздний момент начала, тем выше приоритет операции). При этом продолжительности операций принимаются равным минимальным, то есть предполагается, что каждая операция выполняется максимальным количеством ресурсов a_i. Очевидно, что максимальный приоритет имеют критические операции. Рассмотрим пример, который иллюстрирует причину возможной неоптимальности решения, полученного на основе правила по степени критичности операций.

Пример 2.3. Рассмотрим комплекс из шести операций с линейными зависимостями скоростей операций от количества ресурсов и выполняемых ресурсами одного вида (рис. 2.2).

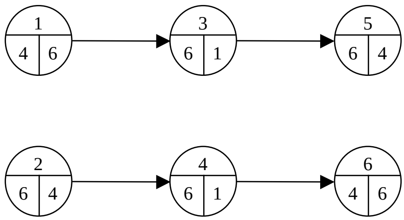

Рис. 2.2.

Верхнее число в вершине на рис. 2.2 указывает номер операции, нижнее слева – минимальную продолжительность τ_i, нижнее справа – максимальное количество ресурсов a_i. Пусть N = 5. Поскольку обе операции 1 и 2 являются критическими, то распределяем ресурсы прямопропорционально величинам a_1 и a_2, то есть $u_1 = 3$, $u_2 = 2$. Нетрудно показать, что распределение ресурсов, прямопропорциональное величинам a_i критических операций, сохраняет их критическими. Через 8 дней операция 1 завершится и начнется операция 3. При этом количество ресурсов на операции 2 увеличивается до $u_2 = 4$, так как операция 3 требует всего одной единицы ресурсов. Через 2 дня завершается операция 2 и начинается операция 4. С этого момента используется всего две единицы ресурсов в течении четырех дней, пока не закончится операция 3. Далее процесс идет опять при полной загрузке ресурсов, и комплекс завершается за 24 дня. На рис. 2.3 приведен график использования ресурсов. Видно, что график имеет «провал» в интервале (10, 14). Этот провал называется «узким местом» графика. В данном случае процесс следует организовать так, чтобы минимизировать ширину «узкого места».

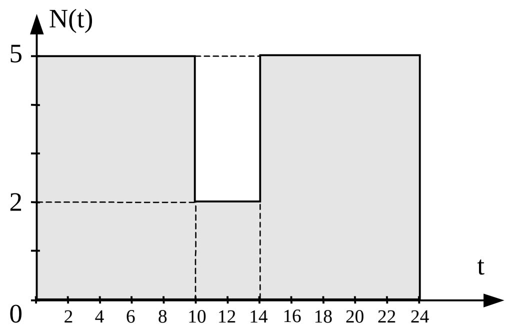

Рис. 2.3.

Для этого следует в первую очередь начать операцию 1, направив на нее все ресурсы. Через 4,8 дня операция 1 будет завершена. Далее выполняются операции 2 и 3, затем 4 и 5 и, наконец, операция 6. Продолжительность комплекса будет минимальной и равна 21,6 дня.

Таким образом, наличие «узких мест» на графике использования ресурсов является признаком возможной неоптимальности календарного плана.

Рассмотрим некоторый фронт работ F(t) в момент t , то есть множество операций, которые выполняются или могут выполняться в этот момент. Как уже отмечалось, степенью критичности операции $i \in F(t)$ называется ее полный резерв времени в этот момент, определенный при условии, что все еще не выполненные операции (или их части) выполняются при максимальном количестве ресурсов. Согласно правилу СК, операции с меньшим полным резервом времени не получают ресурса до тех пор, пока более критические операции не получили максимально возможного количества ресурса. Более того, операции с одинаковой степенью критичности (с равными полными резервами времени) получают ресурс таким образом, что их степени критичности остаются одинаковыми.

Покажем, что в этом случае операции с одинаковой степенью критичности должны получать ресурс прямопропорционально величинам a_i. Действительно, если $u_i = \gamma a_i$ для операций с одинаковыми степенями критичности, и это распределение ресурсов имело место в течении интервала Δ, то минимальная продолжительность этих операций уменьшилась на $\delta = u_i\Delta/a_i = \gamma\Delta$, то есть на одну и ту же величину. На эту же величину увеличился поздний момент начала оставшейся невыполненной части операции, а значит, полные резервы времени этих операций остались равными. Из сказанного выше следует простое свойство полных резервов времени.

Свойство при применении правила СК. Полные резервы времени с течением времени не увеличиваются.

Доказательство. Заметим, что если бы все операции фронта F(t) получили максимальное количество ресурса в интервале длительности Δ, то их минимальные продолжительности уменьшились бы на Δ, и полные резервы времени остались бы без изменения. Однако, в силу правила СК, ресурсы в первую очередь получают критические операции. Поэтому уменьшение минимальных продолжительностей критических операций всегда не меньше, чем всех остальных операций фронта. Поэтому полный резерв любой операции не увеличивается. Отсюда, в частности, следует, что критические операции остаются всегда критическими.

Опираясь на это свойство, докажем оптимальность правил СК для случая независимых операций.

Теорема 2. В случае независимых операций правило СК всегда дает оптимальное решение.

Доказательство. Рассмотрим пример графика использования ресурсов, изображенного на рис. 2.4. Момент завершения комплекса определяется моментом завершения критических операций.

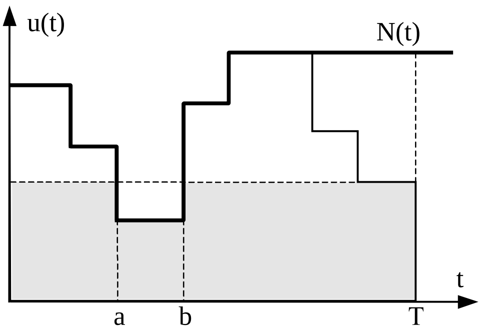

Рис. 2.4.

В силу правила СК и доказанного выше свойства, эти операции в любой момент времени имеют приоритет в получении ресурсов перед всеми другими операциями. Поэтому либо они используют весь ресурс (интервал (a, b) на рисунке), либо они выполняются максимальным количеством ресурса. Очевидно, что выполнить критические операции за время, меньшее чем T, невозможно.

Рассмотрим теперь комплекс операций, который состоит из m независимых путей, каждый из которых, в свою очередь, состоит из n_i операций. Обозначим a_{ij} – максимальное количество ресурсов на j-ой операции i-ой цепочки.

Теорема 3. Если $a_{ij} \geq a_{ij+1}$, $j = \overline{1, n_i - 1}$, $i = \overline{1, m}$, то правило СК всегда дает оптимальное решение.

Доказательство, по сути дела, повторяет доказательство теоремы 2.

2.3. Задача календарного планирования при учете совмещения агрегированных операций

Интересным вариантом агрегированного описания сложного проекта является его представление в виде n агрегированных операций, зависимость между которыми учитывается с помощью так называемых коэффициентов совмещения.

Коэффициент совмещения по началу k_{ij} означает, что работу j можно начинать только когда выполнена определенная часть k_{ij} работы i. Коэффициент совмещения по концу q_{ij} означает, что после завершения работы i необходимо выполнить не менее определенной части q_{ij} работы j. С помощью коэффициентов совмещения можно описывать как технологические, так и ресурсные зависимости. Наличие коэффициентов совмещения k_{ij} и q_{ij} не означает, что работа i должна начаться (окончиться) раньше чем начнется (окончится) работа j, а означает только, что они имеют смысл, если работа i начнется (завершится)ранее работы j. Более того, работа j может начаться (окончиться) раньше, чем работа i. В этом случае появляются, соответственно, коэффициенты совмещения k_{ji} и q_{ji}. Если очередности начала и окончания работ определены, можно построить обычную сетевую модель, и задача сводится к оптимальному распределению ресурсов на этой модели. Для того, чтобы убедиться в этом, достаточно рассмотреть две работы – i, j.

Пусть работа i начинается и заканчивается раньше работы j. В этом случае получаем сетевой график, показанный на рис. 2.5.

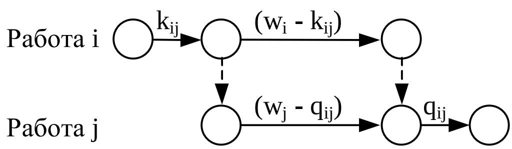

Рис. 2.5.

Если работа i начинается раньше работы j, а заканчивается позже, то получаем сетевой график, показанный на рис. 2.6.

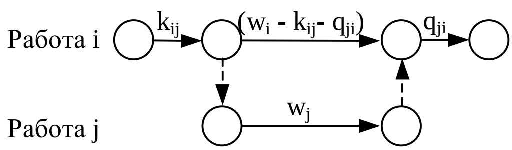

Рис. 2.6.

Аналогично можно изобразить остальные случаи.

Пример 2.4. Рассмотрим проект из трех операций, коэффициенты совмещения которых приведены ниже:

$$K = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 0 & 4 \\ 4 & 2 & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 0 & 3 & 1 \\ 2 & 0 & 3 \\ 4 & 2 & 0 \end{pmatrix}.$$

Пусть очередность начала и окончания работ одна и та же – (1, 2, 3), то есть работа 1 начинается (заканчивается) раньше работы 2, а работа 2 – раньше работы 3.Сетевой график, соответствующий такой очередности начала и завершения работ приведен на рис. 2.7.

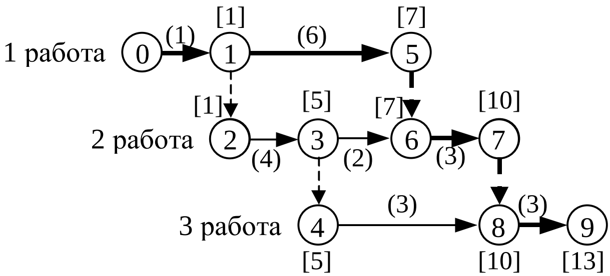

Рис. 2.7.

Примем, что продолжительности работ заданы и коэффициенты совмещения измеряются в единицах времени. Продолжительность первой работы равна 7, второй – 9, третей – 6. Продолжительность проекта, определяемая критическим путем (показан на рис. 2.7 толстыми дугами) равна 13.

Задача. Определить очередность начала и окончания работ, при которой продолжительность проекта минимальна.

Поставленная задача относится к классу NP-трудных задач типа задачи коммивояжера. Поэтому для ее решения следует применять либо эвристические методы, либо методы локальной оптимизации, либо методы типа ветвей и границ. Рассмотрим метод получения оценки снизу для данной задачи, предполагая, что очередность начала работ такая же, как очередность их окончания. Для этого определим

$$a_i = \min_{j \neq i} k_{ij},$$

$$b_i = \min_{j \neq i} q_{ji},$$

$$d_i = \tau_i - b_i.$$

Рассмотрим задачу, которая является несколько более общей, чем задача Джонсона о двух станках.

Имеется n деталей. Каждая деталь сначала обрабатывается на первом станке в течении времени a_i. Затем деталь доставляется на второй станок (время доставки равно $(d_i - a_i)$)и обрабатывается на нем в течении времени b_i. Одновременно на одном станке может обрабатываться только одна деталь. Требуется определить очередность обработки деталей, при которой продолжительность обработки всех деталей минимальна. Пусть детали обрабатываются в очередности их номеров. Тогда продолжительность обработки определяется выражением

$$T = \max_i \left(\sum_{j=1}^{i-1} a_j + d_i + \sum_{i}^{n} b_j \right) = B + \max_i \left(\sum_{j=1}^{i-1} c_j + d_i \right),$$

$$\text{где } B = \sum_{1}^{n} b_j, \;\; c_j = a_j - b_j.$$

Обозначив $S = T - B$, приведем это выражение к виду

$$S - \sum_{j=1}^{i-1} c_j \geq d_i, \quad i = \overline{1, n}. \tag{2.3.1}$$

Системе неравенств (2.3.1) можно дать другую содержательную интерпретацию.

Задача о лекторе. Лектор должен посетить n городов. Для поездки в город i ему нужна сумма d_i. В городе i лектору оплачиваются транспортные расходы d_i, кроме того, он получает за лекцию некоторую сумму b_i, а тратит в городе i сумму a_i. Какую минимальную сумму нужно иметь лектору, чтобы посетить все города?

Очевидно, что условие (2.3.1) является необходимым условием поездки в город i, если до этого лектор посетил города от 1 до (i-1). Из этой содержательной интерпретации сразу следует правило выбора оптимальной очередности посещения городов. Очевидно, что сначала следует посещать города, в которых оплата лекций b_i превышает расходы a_i, то есть после посещения которых сумма денег у лектора увеличивается. Столь же очевидно, что эти города лектору следует посещать в очередности возрастания (неубывания) транспортных расходов d_i. Далее лектор посещает города, в которых он тратит больше чем получает, но уже в очередности убывания (невозрастания) транспортных расходов.

Пример 2.5. Получим оценку снизу для задач из примера 2.4. Имеем:

$$a_1 = 1,\ b_1 = 2,\ c_1 = -1,\ d_1 = 5;$$

$$a_2 = 3,\ b_2 = 2,\ c_2 = 1,\ d_2 = 7;$$

$$a_3 = 2,\ b_3 = 1,\ c_3 = 1,\ d_3 = 5.$$

Согласно полученному правилу, сначала обрабатывается первая деталь, затем - вторая, и затем – третья. Продолжительность обработки составит 11 дней. Фактическая продолжительность проекта при этой очередности равна13, как было показано ранее. Теперь можно организовать ветвление, то есть разбиение множества всех решений на подмножества.

Рассмотрим три подмножества. В первом подмножестве первой выполняется первая операция. Оценка снизу остается прежней, то есть T(1) = 11, хотя фактическая продолжительность равна 13.

Во втором подмножестве первой выполняется вторая операция. Оценка снизу равна T(2) = 12. Заметим, что фактическая продолжительность в данном случае также равна 12, то есть оценка снизу является точной.

В третьем подмножестве первой выполняется третья операция. Оценка снизу также равна 12 и определяется очередностью (3, 1, 2). Однако, фактическая оценка при этой очередности равна 14.

Таким образом, оптимальное решение находится в подмножестве 2. Ему соответствует очередность работ (2, 1, 3) с продолжительностью проекта T = 12 (рис. 2.8).

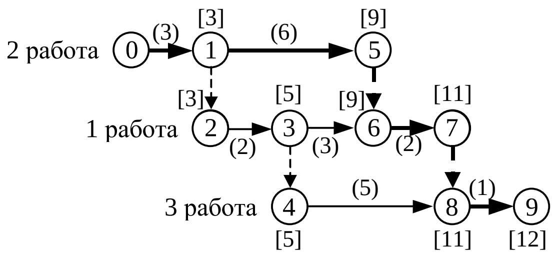

Рис. 2.8.

ЗАКЛЮЧЕНИЕ

Рассмотренные в работе методы построения календарных планов на основе агрегирования комплексов операций позволяет преодолеть «проклятие размерности» и получить эффективные решения реальных задач распределения ресурсов. Наша задача была – показать работоспособность методов агрегирования на примере степенных и линейных зависимостей скоростей операций от количества ресурсов, а также рассмотреть точные методы решения агрегированных задач, позволяющих решать задачи с небольшим числом агрегированных операций.

Дальнейших исследований требует проблема агрегирования комплекса операций с нелинейными зависимостями произвольного вида, наличием ресурсов нескольких типов.

Интересно также рассмотреть методы решения агрегированных задач по критерию упущенной выгоды.

ЛИТЕРАТУРА

1. *Бурков В.Н., Квон О.Ф., Цитович Л.А.* Модели и методы мультипроектного управления. / Препринт - М.: ИПУ РАН, 1998.
2. *Бурков В.Н.* Распределение ресурсов как задача оптимального быстродействия. – АиТ №7, 1966.
3. *Burkov V.N.* Problems of optimum distribution of resources. – Control and Cybernetics. Vol. 1 (1972), №1/2.
4. *Воронов А.А., Петрушинин Е.П.* Решение задачи оптимального распределения ресурсов методом квадратичного программирования. – АиТ №5, 1965.
5. *Разумихин Б.С.* Задача об оптимальном распределении ресурсов. – АиТ №7, 1965.