

Тема 7. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИНТЕЗА ПРОЕКТНЫХ РЕШЕНИЙ

Содержание

7.1. Постановка задач параметрического синтеза	1
7.1.1. Место процедур синтеза в проектировании	1
7.1.2. Критерии оптимальности	2
7.1.3. Задачи оптимизации с учетом допусков	4
7.2. Обзор методов оптимизации	5
7.2.1. Классификация методов математического программирования	5
7.2.2. Методы одномерной оптимизации	6
7.2.3. Методы безусловной оптимизации	7
7.2.4. Необходимые условия экстремума	10
7.2.5. Методы поиска условных экстремумов	11
7.3. Постановка задач структурного синтеза	12
7.3.1. Процедуры синтеза проектных решений	12
7.3.2. Задача принятия решений	13
7.3.3. Представление множества альтернатив	15
7.4.4. Морфологические таблицы	15
7.4.5. Альтернативные графы	16
7.4.6. Исчисления	17
7.4.7. Планирование процессов и распределение ресурсов	18
7.4. Методы структурного синтеза в САПР	19
7.4.1. Метод ветвей и границ	19
7.4.2. Системы искусственного интеллекта	20

7.1. Постановка задач параметрического синтеза

7.1.1. Место процедур синтеза в проектировании

Сущность проектирования заключается в принятии проектных решений, обеспечивающих выполнение будущим объектом предъявляемых к нему требований. Синтез проектных решений – основа проектирования; от успешного выполнения процедуры синтеза в определяющей мере зависят потребительские свойства будущей продукции. Конечно, анализ – необходимая составная часть проектирования, служащая для верификации принимаемых проектных решений. Именно анализ позволяет получить необходимую информацию для целенаправленного выполнения процедур синтеза в итерационном процессе проектирования. Поэтому синтез и анализ неразрывно связаны.

Синтез подразделяют на параметрический и структурный. Проектирование начинается со *структурного синтеза*, при котором генерируется принципиальное решение. Таким решением может быть облик будущего металлорежущего станка, или физический принцип действия датчика, или одна из типовых конструкций двигателя, или функциональная схема микропроцессора. Но эти конструкции и схемы выбирают в параметрическом виде, т.е. без указания числовых значений параметров элементов. Поэтому прежде чем приступить к верификации проектного решения, нужно задать или рассчитать значения этих параметров, т.е. выполнить *параметрический синтез*. Примерами результатов параметрического синтеза могут служить геометрические размеры деталей в механическом узле или в приспособлении, параметры режимов резания в технологической операции и т.п.

В случае если по результатам анализа проектное решение признается неокончательным, то начинается процесс последовательных приближений к приемлемому варианту проекта. Во многих приложениях для улучшения проекта удобнее варьировать значения параметров элементов, т.е. использовать *параметрический синтез на базе многовариантного анализа*. При этом задача параметрического синтеза может быть сформулирована как задача определения

значений параметров элементов, наилучших с позиций удовлетворения требований технического задания при неизменной структуре проектируемого объекта. Тогда параметрический синтез называют параметрической оптимизацией или просто *оптимизацией*. Если параметрический синтез не приводит к успеху, то повторяют процедуры структурного синтеза, т.е. на очередных итерациях корректируют или перевыбирают структуру объекта.

7.1.2. Критерии оптимальности

В САПР процедуры параметрического синтеза выполняются либо человеком в процессе многовариантного анализа (в интерактивном режиме), либо реализуются на базе формальных методов оптимизации (в автоматическом режиме). В последнем случае находят применение несколько постановок задач оптимизации.

Наиболее распространенной является детерминированная постановка: *заданы условия работоспособности на выходные параметры Y и нужно найти номинальные значения проектных параметров X , к которым относятся параметры всех или части элементов проектируемого объекта.*

Назовем эту задачу оптимизации *базовой*. В частном случае, когда требования к выходным параметрам заданы нечетко, к числу рассчитываемых величин могут быть отнесены также нормы выходных параметров, фигурирующие в их условиях работоспособности.

Если проектируются изделия для дальнейшего серийного производства, то важное значение приобретает такой показатель, как процент выпуска годных изделий в процессе производства. Очевидно, что успешное выполнение условий работоспособности в номинальном режиме не гарантирует их выполнения при учете производственных погрешностей, задаваемых допусками параметров элементов. Поэтому *целью оптимизации становится максимизация процента выхода годных*, а к результатам решения задачи оптимизации относятся не только номинальные значения проектных параметров, но и их допуски.

Базовая задача оптимизации ставится как задача математического программирования

$$\begin{aligned} & \underset{X \in D_x}{extr} F(X), \\ & D_x = \{X \mid \varphi(X) > 0, \psi(X) = 0\} \end{aligned} \quad (7.1)$$

где $F(X)$ – целевая функция, X – вектор управляемых (проектных) параметров, $\varphi(X)$ и $\psi(X)$ – функции–ограничения; D_x – допустимая область в пространстве управляемых параметров.

Запись (7.1) интерпретируется как задача поиска экстремума целевой функции путем варьирования управляемых параметров в пределах допустимой области.

Таким образом, для выполнения расчета номинальных значений параметров необходимо, во–первых, сформулировать задачу в виде (7.1), во–вторых, решить задачу поиска экстремума $F(X)$.

Сложность постановки оптимизационных проектных задач обусловлена наличием у проектируемых объектов нескольких выходных параметров, которые могут быть критериями оптимальности, но в задаче (7.1) целевая функция должна быть одна. Другими словами, проектные задачи являются многокритериальными, и возникает проблема сведения многокритериальной задачи к однокритериальной.

Применяют несколько способов выбора критерия оптимальности.

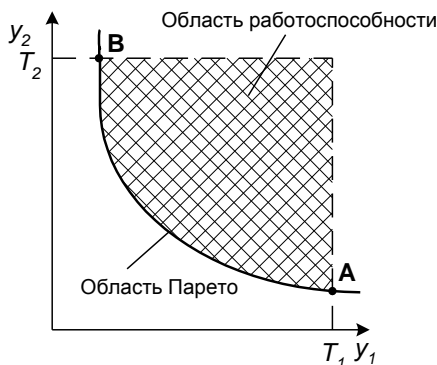


Рис. 7.1 Области Парето и работоспособности

В частном критерии среди выходных параметров один выбирают в качестве целевой функции, а условия работоспособности остальных выходных параметров относят к ограничениям задачи (7.1). Эта постановка вполне приемлема, если действительно можно выделить один наиболее критичный выходной параметр.

Но в большинстве случаев сказывается недостаток частного критерия (рис. 7.1). На этом рисунке представлено двумерное пространство выходных параметров u_1 и u_2 , для которых заданы условия работоспособности $u_1 < T_1$ и $u_2 < T_2$. Кривая АВ является

границей достижимых значений выходных параметров. Это ограничение объективное и связано с существующими физическими и технологическими условиями производства, называемыми условиями реализуемости.

Область, в пределах которой выполняются все условия реализуемости и работоспособности, называют *областью работоспособности*. Множество точек пространства выходных параметров, из которых невозможно перемещение, приводящее к улучшению всех выходных параметров, называют областью компромиссов, или *областью Парето*. Участок кривой АВ (см. рис. 7.1) относится к области Парето.

Если в качестве целевой функции в ситуации рис. 7.1. выбрать параметр u_1 , то результатом оптимизации будут параметры X , соответствующие точке В. Но это граница области работоспособности и, следовательно, при нестабильности внутренних и внешних параметров велика вероятность выхода за пределы области работоспособности. Конечно, результаты можно улучшить, если применять так называемый метод уступок, при котором в качестве ограничения принимают условие работоспособности со скорректированной нормой в виде

$$u_2 < T_2 + \Delta,$$

где Δ – уступка. Но возникает проблема выбора значений уступок, т.е. результаты оптимизации будут иметь субъективный характер. Очевидно, что ситуация не изменится, если целевой функцией будет выбран параметр u_2 , – оптимизация приведет в точку А.

Аддитивный критерий объединяет (свертывает) все выходные параметры (частные критерии) в одну целевую функцию, представляющую собой взвешенную сумму частных критериев

$$F(X) = \sum_{j=1}^m w_j y_j(X) \quad (7.2)$$

где w_j – весовой коэффициент, m – число выходных параметров. При чем, $\sum_{j=1}^m w_j = 1$.

Функция (7.2) подлежит минимизации, при этом если условие работоспособности имеет вид $y_j > T_j$, то $w_j < 0$.

Недостатки аддитивного критерия – субъективный подход к выбору весовых коэффициентов и неучет требований технического задания. Действительно в (7.2) не входят нормы выходных параметров.

Аналогичные недостатки присущи и *мультипликативному критерию*, целевая функция которого имеет вид

$$F(X) = \prod_{j=1}^m y_j^{w_j}(X) \quad (7.3)$$

Нетрудно видеть, что если прологарифмировать (7.3), то мультипликативный критерий превращается в аддитивный.

Более предпочтительным является *максиминный критерий*, в качестве целевой функции которого принимают выходной параметр, наиболее неблагоприятный с позиций выполнения условий работоспособности. Для оценки степени выполнения условия работоспособности j -го выходного параметра вводят запас работоспособности этого параметра S_j , и этот запас можно рассматривать как нормированный j -й выходной параметр. Например (здесь и далее для лаконичности изложения предполагается, что все выходные параметры приведены к виду, при котором условия работоспособности становятся неравенствами в форме $y_j < T_j$):

$$S_j = \frac{T_j - y_j}{T_j}$$

или

$$S_j = \frac{T_j - y_{номj}}{\delta_j}$$

где $y_{номj}$ – номинальное значение, а δ_j – некоторая характеристика рассеяния j -го выходного параметра, например, трехсигмовый допуск. Тогда целевая функция в максиминном критерии есть

$$F(X) = \min_{j \in [1:m]} Z_j(X)$$

Здесь запись $[1: m]$ означает множество целых чисел в диапазоне от 1 до m . Задачу (7.1) при максиминном критерии конкретизируют следующим образом:

$$F(X) = \max_{X \in D_X} \min_{j \in [1:m]} Z_j(X)$$

где допустимая область D_X определяется только прямыми ограничениями на управляемые параметры x_i : $x_{i \min} < x_i < x_{i \max}$.

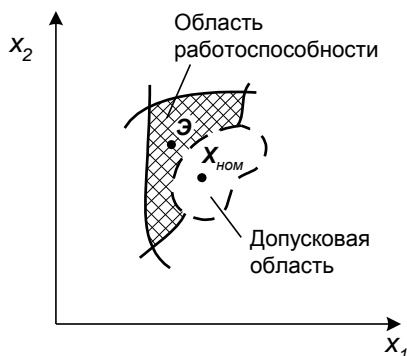


Рис.7.2.Области допусковая и работоспособности

7.1.3. Задачи оптимизации с учетом допусков

Содержательную сторону оптимизации с учетом допусков поясняет рис. 7.2, на котором представлены области работоспособности и допусковая в двумерном пространстве управляемых параметров. Если собственно допуски заданы и не относятся к управляемым параметрам, то цель оптимизации – максимальным образом совместить эти области так, чтобы вероятность выхода за пределы области работоспособности была минимальной.

Решение этой задачи исключительно трудоемко, так как на каждом шаге оптимизации нужно выполнять оценку упомянутой вероятности методами статистического анализа, а для сложных моделей объектов таким методом является метод статистических испытаний. Поэтому на практике подобные задачи решают, принимая те или иные допущения.

Например, если допустить, что цель оптимизации достигается при совмещении центров областей работоспособности \mathcal{E} и допусковой $\mathbf{X}_{\text{ном}}$, то оптимизация сводится к задаче *центрирования*, т.е. к определению центра \mathcal{E} . Задачу центрирования обычно решают путем предварительного нормирования управляемых параметров x_i с последующим вписыванием гиперкуба с максимально возможными размерами в нормированную область работоспособности.

Очевидно, что решение задачи центрирования позволяет не только оптимизировать номинальные значения проектных параметров, но и их допуски, если последние относятся к управляемым параметрам.

7.2. Обзор методов оптимизации

7.2.1. Классификация методов математического программирования

В САПР основными методами оптимизации являются поисковые методы. Поисковые методы основаны на пошаговом изменении управляемых параметров

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \Delta \mathbf{X}_k, \quad (7.5)$$

где в большинстве методов приращение $\Delta \mathbf{X}_k$ вектора управляемых параметров вычисляется по формуле

$$\Delta \mathbf{X}_k = h \mathbf{g}(\mathbf{X}_k). \quad (7.6)$$

Здесь \mathbf{X}_k – значение вектора управляемых параметров на k -м шаге, h – шаг, а $\mathbf{g}(\mathbf{X}_k)$ – направление поиска. Следовательно, если выполняются условия сходимости, то реализуется пошаговое (итерационное) приближение к экстремуму.

Методы оптимизации классифицируют по ряду признаков.

В зависимости от числа управляемых параметров различают методы *одномерной* и *многомерной* оптимизации, в первом из них управляемый параметр единственный, во втором размер вектора \mathbf{X} не менее двух. Реальные задачи в САПР многомерны, методы одномерной оптимизации играют вспомогательную роль на отдельных этапах многомерного поиска.

Различают методы *условной* и *безусловной* оптимизации по наличию или отсутствию ограничений. Для реальных задач характерно наличие ограничений, однако методы безусловной оптимизации также представляют интерес, поскольку задачи условной оптимизации с помощью специальных методов могут быть сведены к задачам без ограничений.

В зависимости от числа экстремумов различают задачи одно– и многоэкстремальные. Если метод ориентирован на определение какого–либо локального экстремума, то такой метод относится к *локальным методам*. Если же результатом является глобальный экстремум, то этот метод называют *методом глобального поиска*. Удовлетворительные по вычислительной эффективности методы глобального поиска для общего случая отсутствуют и потому на практике в САПР используют методы поиска локальных экстремумов.

Наконец, в зависимости от того, используются при поиске производные целевой функции по управляемым параметрам или нет, различают методы нескольких порядков. Если

производные не используются, то имеет место метод *нулевого порядка*, если используются первые или вторые производные, то соответственно метод *первого* или *второго порядка*. Методы первого порядка называют также градиентными, поскольку вектор первых производных $F(\mathbf{X})$ по \mathbf{X} есть градиент целевой функции

$$\mathbf{grad}(F(\mathbf{X})) = (\partial P/\partial x_1, \partial P/\partial x_2, \dots, \partial P/\partial x_n).$$

Конкретные методы определяются следующими факторами:

- 1) способом вычисления направления поиска $\mathbf{g}(\mathbf{X}_k)$ в формуле (7.6);
- 2) способом выбора шага h ;
- 3) способом определения окончания поиска.

Определяющим фактором является первый из перечисленных в этом списке, он подробно описан ниже.

Шаг может быть или постоянным, или выбираться исходя из одномерной оптимизации – поиска минимума целевой функции в выбранном направлении $\mathbf{g}(\mathbf{X}_k)$. В последнем случае шаг будем называть оптимальным.

Окончание поиска обычно осуществляют по правилу: если на протяжении r подряд идущих шагов траектория поиска остается в малой ε -окрестности текущей точки поиска \mathbf{X}_k , то поиск следует прекратить, следовательно, условие окончания поиска имеет вид

$$|\mathbf{X}_k - \mathbf{X}_{k-r}| < \varepsilon.$$

7.2.2. Методы одномерной оптимизации

К методам одномерной оптимизации относятся методы дихотомического деления, золотого сечения, чисел Фибоначчи, полиномиальной аппроксимации и ряд их модификаций.

Пусть задан отрезок $[A, B]$, на котором имеется один минимум (в общем случае нечетное число минимумов). Согласно *методу дихотомического деления* (рис. 7.3, а) отрезок делят пополам и в точках, отстоящих от центра C отрезка на величину допустимой погрешности q , рассчитывают значения целевой функции $F(C + q)$ и $F(C - q)$. Если окажется, что $F(C + q) > F(C - q)$, то минимум находится на отрезке $[A, C]$, если $F(C + q) < F(C - q)$, то минимум – на $[C, B]$, если $F(C + q) = F(C - q)$ – на $[C - q, C + q]$. Таким образом, на следующем шаге вместо отрезка

$[A, B]$ нужно исследовать суженный отрезок $[A, C]$, $[C, B]$ или $[C - q, C + q]$. Шаги повторяются, пока длина отрезка не уменьшится до величины погрешности q .

По *методу золотого сечения* (рис. 7.3, б) внутри отрезка $[A, B]$ выделяют две промежуточные точки C_1 и D_1 на расстоянии $s = aL$ от его конечных точек, где $L = B - A$ – длина отрезка. Затем вычисляют значения целевой функции $F(x)$ в точках C_1 и D_1 . Если $F(C_1) < F(D_1)$, то минимум находится на отрезке $[A, D_1]$, если $F(C_1) > F(D_1)$, то – на отрезке $[C_1, B]$, если $F(C_1) = F(D_1)$ – на отрезке $[C_1, D_1]$. Следовательно, вместо отрезка $[A, B]$ можно рассматривать отрезок $[A, D_1]$, $[C_1, B]$ или $[C_1, D_1]$, т.е. длина отрезка уменьшилась не менее чем в $L/(L - aL) = 1/(1 - a)$ раз.

Если подобрать значение a так, что в полученном отрезке

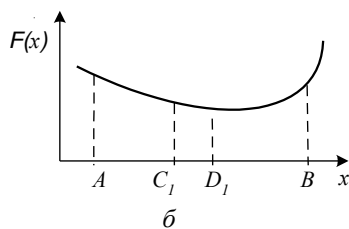
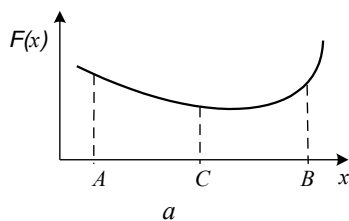


Рис. 7.3. Одномерная минимизация:

а – дихотомическое деление;
б – золотое сечение

меньшей длины одна из промежуточных точек совпала бы с промежуточной точкой от предыдущего шага, т.е. в случае выбора отрезка $[A, D_1]$ точка D_2 совпадет с точкой C_1 , а в случае выбора отрезка $[C_1, B]$ точка C_2 – с точкой D_1 , то это позволит сократить число вычислений целевой функции на всех шагах (кроме первого) в два раза.

Условие получения такого значения a формулируется следующим образом: $(1 - 2a)L_k = aL_{k+1}$, откуда с учетом того, что $L_k / L_{k+1} = 1/(1 - a)$, имеем $a = 0,382$. Это значение и называют *золотым сечением*.

Согласно *методу чисел Фибоначчи*, используют числа Фибоначчи R_i , последовательность которых образуется по правилу $R_{i+2} = R_{i+1} + R_i$ при $R_0 = R_1 = 1$, т.е. ряд чисел Фибоначчи имеет вид 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 147.... Метод аналогичен методу золотого сечения с тем отличием, что коэффициент a равен отношению R_{i-2}/R_i , начальное значение i определяется из условия, что R_i должно быть наименьшим числом Фибоначчи, превышающим величину $(B-A)/E$, где E – заданная допустимая погрешность определения экстремума. Так, если $(B-A)/E = 100$, то начальное значение $i=11$, поскольку $R_{11}=144$, и $a = 55/144 = 0,3819$, на следующем шаге будет $a = 34/89 = 0,3820$ и т.д.

По *методу полиномиальной аппроксимации* при аппроксимации $F(x)$ квадратичным полиномом

$$P(x) = a_0 + a_1x + a_2x^2 \quad (7.7)$$

выбирают промежуточную точку C и в точках A, B, C вычисляют значения целевой функции. Далее решают систему из трех алгебраических уравнений, полученных подстановкой в (7.7) значений A, B, C вместо x и вычисленных значений функции вместо $P(x)$. В результате становятся известными значения коэффициентов a_k в (7.7) и, исходя из условия $dP(x)/dx=0$, определяют экстремальную точку Ξ полинома. Например, если точка C выбрана в середине отрезка $[A, B]$, то

$$\Xi = \frac{C + (C - A)(F(A) - F(B))}{2(F(A) - 2F(C) + F(B))}$$

7.2.3. Методы безусловной оптимизации

Среди методов нулевого порядка в САПР находят применение методы Розенброка, конфигураций (Хука–Дживса), деформируемого многогранника (Нелдера–Мида), случайного поиска. К методам с использованием производных относятся методы наискорейшего спуска, сопряженных градиентов, метод Ньютона.

Метод Розенброка является улучшенным вариантом покоординатного спуска.

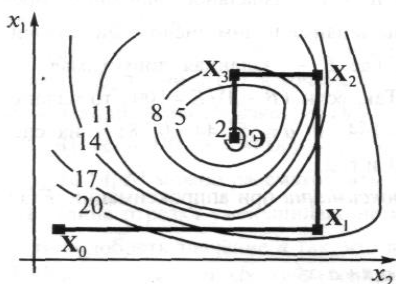


Рис. 7.4. Траектория покоординатного спуска

Метод покоординатного спуска характеризуется выбором направлений поиска поочередно вдоль всех n координатных осей, шаг рассчитывают на основе одномерной оптимизации, критерий окончания поиска $|\mathbf{X}_k - \mathbf{X}_{k-n}| < \varepsilon$, где ε – заданная точность определения локального экстремума, n – размерность пространства управляемых параметров. Траектория покоординатного спуска для примера двумерного пространства управляемых параметров показана на рис. 7.4, где \mathbf{X}_k – точки на

через центр тяжести **ЦТ** многогранника, причем рекомендуется X_4 выбирать на расстоянии d от **ЦТ**, равном $|\mathbf{CT} - X_1|$. Новая вершина X_4 заменяет худшую вершину X_1 . Если оказывается, что X_4 имеет лучшее значение целевой функции среди вершин многогранника, то расстояние d увеличивают. На рисунке именно эта ситуация имеет место и увеличение d дает точку X_5 . В новом многограннике с вершинами X_2, X_3, X_5 худшей является вершина X_2 , аналогично получают вершину X_6 , затем вершину X_7 и т. д. Если новая вершина окажется худшей, то в многограннике нужно сохранить лучшую вершину, а длины всех ребер уменьшить, например, вдвое (стягивание многогранника к лучшей вершине). Поиск прекращается при выполнении условия уменьшения размеров многогранника до некоторого предела.

Случайные методы поиска характеризуются тем, что направления поиска g выбирают случайным образом.

Особенностью *метода наискорейшего спуска* является выполнение шагов поиска в градиентном направлении

$$X_{k+1} = X_k + \frac{h \text{grad} F(X)}{|\text{grad} F(X)|}$$

шаг h выбирается оптимальным с помощью одномерной оптимизации.

При использовании метода наискорейшего спуска, как и большинства других методов, эффективность поиска существенно снижается в овражных ситуациях. Траектория поиска приобретает зигзагообразный вид с медленным продвижением вдоль дна оврага в сторону экстремума. Чтобы повысить эффективность градиентных методов, используют несколько приемов.

Один из приемов, использованный в *методе сопряженных градиентов* (называемом также методом Флетчера–Ривса), основан на понятии сопряженности векторов. Векторы A и B называют Q –сопряженными, если $A^T Q B = 0$, где Q – положительно определенная квадратная матрица того же порядка, что и размер N векторов A и B (частный случай сопряженности – ортогональность векторов, когда Q является единичной матрицей порядка N), A^T – вектор–строка, B – вектор–столбец.

Особенность сопряженных направлений для $Q=G$, где G – матрица Гессе, в задачах с квадратичной целевой функцией $F(X)$ заключается в следующем: одномерная минимизация $F(X)$ последовательно по N сопряженным направлениям позволяет найти экстремальную точку не более чем за N шагов.

Примечание. *Матрицей Гессе* называют матрицу вторых частных производных целевой функции по управляемым параметрам.

Основанием для использования поиска по G –сопряженным направлениям является то, что для функций $F(X)$ общего вида может быть применена квадратичная аппроксимация, что на практике выливается в выполнение поиска более чем за N шагов.

Метод Ньютона основан на использовании необходимых условий безусловного экстремума целевой функции $F(X)$

$$\text{grad } F(X) = 0. \quad (7.15)$$

Выражение (7.15) представляет собой систему алгебраических уравнений, для решения которой можно применить известный численный метод, называемый методом Ньютона. Корень системы (7.15) есть стационарная точка, т.е. возможное решение экстремальной задачи. Метод

Ньютона является итерационным, он основан на линеаризации (7.15) в окрестности текущей точки поиска \mathbf{X}_k

$$\mathbf{grad} F(\mathbf{X}) = \mathbf{grad} F(\mathbf{X}_k) + \mathbf{\Gamma}(\mathbf{X} - \mathbf{X}_k) = 0. \quad (7.16)$$

Выражение (7.16) – это система линейных алгебраических уравнений. Ее корень есть очередное приближение \mathbf{X}_{k+1} к решению $\mathbf{X}_{k+1} = \mathbf{X}_k^T \mathbf{\Gamma}^{-1}(\mathbf{X}_k) \mathbf{grad} F(\mathbf{X}_k)$.

Если процесс сходится, то решение достигается за малое число итераций, окончанием которых служит выполнение условия

$$|\mathbf{X}_{k+1} - \mathbf{X}_k| < \varepsilon.$$

Главный недостаток метода – высокая трудоемкость вычисления и обращения матрицы $\mathbf{\Gamma}$, к тому же ее вычисление численным дифференцированием сопровождается заметными погрешностями, что снижает скорость сходимости.

7.2.4. Необходимые условия экстремума

В задачах безусловной оптимизации необходимые условия представляют собой равенство нулю градиента целевой функции:

$$\mathbf{grad} F(\mathbf{X}) = 0.$$

В общей задаче математического программирования (7.1) необходимые условия экстремума, называемые условиями Куна–Таккера, формулируются следующим образом: для того чтобы точка \mathcal{E} была экстремальной точкой выпуклой задачи математического программирования (ЗМП), необходимо наличие неотрицательных коэффициентов u_i таких, что

$$u_i \varphi_i(\mathcal{E}) = 0, i = 1, 2, \dots, m; \quad (7.17)$$

и при этом соблюдались ограничения задачи, а также выполнялось условие

$$\mathbf{grad} F(\mathcal{E}) + \sum_{i=1}^m u_i \mathbf{grad} \varphi_i(\mathcal{E}) + \sum_{j=1}^L a_j \psi_j(\mathcal{E}) = 0 \quad (7.18)$$

где m – число ограничений типа неравенств; L – то же равенств; коэффициенты $a_j > 0$.

За приведенной абстрактной формулировкой условий скрывается достаточно просто понимаемый геометрический смысл. Действительно, рассмотрим сначала случай с ограничениями только типа неравенств. Если максимум находится внутри допустимой области \mathbf{R} , то, выбирая все $u_i = 0$, добиваемся выполнения (7.17); если же точка максимума \mathcal{E} лежит на границе области \mathbf{R} , то, как видно из левой части рис. 7.9, эту точку всегда соответствующим

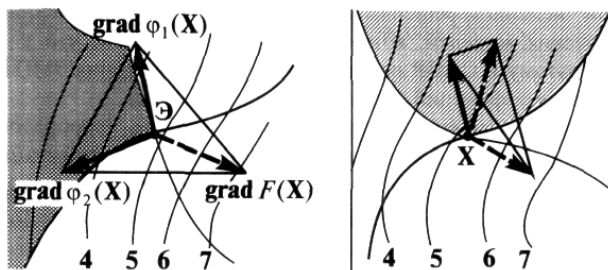


Рис. 7.9. Пояснение условий Куна–Таккера

подбором неотрицательных u_i можно поместить внутрь оболочки, натянутой на градиенты целевой функции $F(\mathbf{X})$ и функций–ограничений $\varphi_i(\mathbf{X})$. Наоборот, если точка не является экстремальной, то (7.17) нельзя выполнить при любом выборе положительных коэффициентов u_i , (см. правую часть рис. 7.9, где рассматриваемая точка \mathbf{X} лежит вне выпуклой оболочки, натянутой на градиенты).

Учет ограничений типа равенств очевиден, если добавляется последняя из указанных в (7.18) сумма.

7.2.5. Методы поиска условных экстремумов

Широко известен метод множителей Лагранжа, ориентированный на поиск экстремума при наличии ограничений типа равенств $\psi(\mathbf{X}) = 0$, т.е. на решение задачи

$$\text{extr}_{X \in R} F(X), \quad (7.19)$$

где $R = \{X \mid \psi(X) = 0\}$.

Суть метода заключается в преобразовании задачи условной оптимизации (7.19) в задачу безусловной оптимизации с помощью образования новой целевой функции

$$\Phi(X, \Lambda) = F(X) + \sum_{i=1}^L \lambda_i \psi_i(X)$$

где $\Lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_h)$ – вектор множителей Лагранжа; L – число ограничений.

Необходимые условия экстремума функции $\Phi(X)$:

$$\begin{aligned} \frac{\partial \Phi(X, \Lambda)}{\partial X} &= \frac{\partial F(X)}{\partial X} + \sum_{i=1}^L \lambda_i \frac{\partial \psi_i(X)}{\partial X} \\ \frac{\partial \Phi(X, \Lambda)}{\partial \Lambda} &= \psi(X) = 0 \end{aligned} \quad (7.20)$$

Система (7.20) содержит $n+L$ алгебраических уравнений, где n – размерность пространства управляемых параметров, ее решение дает искомые координаты экстремальной точки и значения множителей Лагранжа. Однако при численном решении (7.20), что имеет место при использовании алгоритмических моделей, возникают те же трудности, что и в методе Ньютона. Поэтому в САПР основными методами решения ЗМП являются методы штрафных функций и проекции градиента.

Основная идея *методов штрафных функций* – преобразование задачи условной оптимизации в задачу безусловной оптимизации путем формирования новой целевой функции $\Phi(X)$ введением в исходную целевую функцию $F(X)$ специальным образом выбранной функции штрафа $S(X)$:

$$\Phi(X) = F(X) + rS(X),$$

где r – множитель, значения которого можно изменять в процессе оптимизации.

Среди методов штрафных функций различают методы внутренней и внешней точки. Согласно методам внутренней точки (иначе называемым методами *барьерных функций*) исходную для поиска точку можно выбирать только внутри допустимой области, а для методов

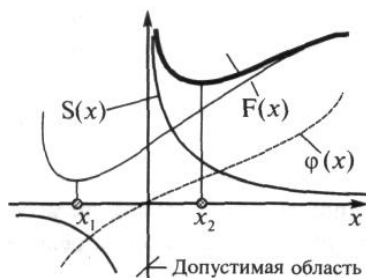


Рис. 7.10. Пояснение метода штрафных функций

внешней точки – как внутри, так и вне допустимой области (важно лишь, чтобы в ней функции целевая и ограничений были определены). Ситуация появления барьера у целевой функции $\Phi(x)$ и соотношение между условным в точке x_2 и безусловным в точке x_1 минимумами $F(x)$ в простейшем одномерном случае иллюстрируется рис. 7.10.

Чем больше коэффициент r , тем точнее решение задачи, однако при больших r может ухудшаться ее обусловленность. Поэтому в начале поиска обычно выбирают умеренные значения r , увеличивая их в окрестностях экстремума.

Основной вариант *метода проекции градиента* ориентирован на задачи математического программирования с ограничениями типа равенств.

Поиск при выполнении ограничений осуществляется в подпространстве $(n-m)$ измерений, где n – число управляемых параметров, m – число ограничений, при этом движение осуществляется в направлении проекции градиента целевой функции $F(\mathbf{X})$ на гиперплоскость, касательную к гиперповерхности ограничений.

Поиск минимума начинают со спуска из исходной точки на гиперповерхность ограничений. Далее выполняют шаг в указанном выше направлении (шаг вдоль гиперповерхности ограничений). Поскольку этот шаг может привести к заметному нарушению ограничений, вновь повторяют спуск на гиперповерхность ограничений и т.д. Другими словами, поиск заключается в выполнении пар шагов, каждая пара включает спуск на гиперповерхность ограничений и движение вдоль гиперповерхности ограничений.

7.3. Постановка задач структурного синтеза

7.3.1. Процедуры синтеза проектных решений

Принятие проектных решений охватывает широкий круг задач и процедур – от выбора вариантов в конечных и обозримых множествах до задач творческого характера, не имеющих формальных способов решения.

Соответственно в САПР применяют как средства формального синтеза проектных решений, выполняемого в автоматическом режиме, так и вспомогательные средства, способствующие выполнению синтеза проектных решений в интерактивном режиме. К вспомогательным средствам относятся базы типовых проектных решений, системы обучения проектированию, программно–методические комплексы верификации проектных решений, унифицированные языки описания технического задания и результатов проектирования.

Задачи синтеза структур проектируемых объектов относятся к наиболее трудно формализуемым. Существует ряд общих подходов к постановке этих задач, однако практическая реализация большинства из них неочевидна. Поэтому имеются лишь «островки» автоматического выполнения процедур синтеза среди «моря» проблем, ждущих автоматизации.

Именно по этой причине структурный синтез, как правило, выполняют в интерактивном режиме при решающей роли инженера–разработчика, а ЭВМ играет вспомогательную роль: предоставление необходимых справочных данных, фиксация и оценка промежуточных и окончательных результатов.

В ряде приложений, однако, имеются и примеры успешной автоматизации структурного синтеза. Среди них заслуживают упоминания в первую очередь задачи синтеза технологических процессов и управляющих программ для механообработки в машиностроении и некоторые другие.

Структурный синтез заключается в преобразовании описаний проектируемого объекта: исходное описание содержит информацию о требованиях к свойствам объекта, об условиях его функционирования, ограничениях на элементный состав и т.п., а результирующее описание должно содержать сведения о *структуре*, т.е. о составе элементов и способах их соединения и взаимодействия.

Постановки и методы решения задач структурного синтеза в связи с трудностями формализации не достигли степени обобщения и детализации, свойственной математическому обеспечению процедур анализа. Достигнутая степень обобщения выражается в установлении типичной последовательности действий и используемых видов описаний при их преобразованиях в САПР. Исходное описание, как правило, представляет собой техническое задание на проектирование, по нему составляют описание на некотором формальном языке, являющемся входным языком используемых подсистем САПР. Затем выполняют преобразования описаний, и получаемое итоговое для данного этапа описание документируют – представляют в виде твердой копии или файла в соответствующем формате для передачи на следующий этап.

Важное значение для развития подсистем синтеза в САПР имеют разработка и унификация языков представления описаний (спецификаций). Каждый язык, поддерживая выбранную методику принятия решений, формирует у пользователей САПР – разработчиков технических объектов определенный стиль мышления; особенности языков непосредственно влияют на особенности правил преобразования спецификаций. Примерами унифицированных языков описания проектных решений является язык Express – универсальный язык спецификаций для представления и обмена информацией в компьютерных средах.

7.3.2. Задача принятия решений

Имеется ряд подходов для обобщенного описания задач принятия проектных решений в процессе структурного синтеза.

Задачу принятия решения (ЗПР) формулируют следующим образом:

$$\text{ЗПР} = \langle \mathbf{A}, \mathbf{K}, \text{Мод}, \Pi \rangle,$$

где \mathbf{A} – множество альтернатив проектного решения, $\mathbf{K} = (\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_m)$ – множество критериев (выходных параметров), по которым оценивается соответствие альтернативы поставленным целям; Мод: $\mathbf{A} \rightarrow \mathbf{K}$ – модель, позволяющая для каждой альтернативы рассчитать вектор критериев; Π – решающее правило для выбора наиболее подходящей альтернативы в многокритериальной ситуации.

В свою очередь, каждой альтернативе конкретного приложения можно поставить в соответствие значения упорядоченного множества (набора) атрибутов $\mathbf{X} = \langle x_1, x_2, \dots, x_n \rangle$, характеризующих свойства альтернативы. При этом x_i может быть величиной типа real, integer, Boolean, string (в последнем случае величину называют *предметной* или *лингвистической*). Множество \mathbf{X} называют *записью* (в теории баз данных), *фреймом* (в искусственном интеллекте) или *хромосомой* (в генетических алгоритмах). Модель Мод называют структурно–критериальной, если среди \mathbf{X} , имеются параметры, характеризующие структуру моделируемого объекта. Основными проблемами ЗПР являются:

- компактное представление множества вариантов (альтернатив);
- построение модели синтезируемого устройства, в том числе выбор степени абстрагирования для оценки значений критериев;
- формулировка предпочтений в многокритериальных ситуациях (т.е. преобразование векторного критерия \mathbf{K} в скалярную целевую функцию);

- установление порядка (предпочтений) между альтернативами в отсутствие количественной оценки целевой функции (что обычно является следствием неколичественного характера всех или части критериев);

- выбор метода поиска оптимального варианта (сокращение перебора вариантов).

Присущая проектным задачам неопределенность и нечеткость исходных данных, а иногда и моделей, диктуют использование специальных методов количественной формулировки исходных неколичественных данных и отношений. Эти специальные методы либо относятся к области построения измерительных шкал, либо являются предметом теории нечетких множеств.

Измерительные шкалы могут быть:

1) абсолютными;

2) номинальными (классификационными), значения шкалы представляют классы эквивалентности, примером может служить шкала цветов; такие шкалы соответствуют величинам неколичественного характера;

3) порядковыми, если между объектами A и B установлено одно из следующих отношений: простого порядка, гласящее, что если A лучше B , то B хуже A , и соблюдается транзитивность; или слабого порядка, т.е. либо A не хуже B , либо A не лучше B ; или частичного порядка. Для формирования целевой функции $F(\mathbf{X})$ производится оцифровка порядковой шкалы, т.е. при минимизации, если A предпочтительнее B , то $F(\mathbf{X}_a) < F(\mathbf{X}_b)$, где \mathbf{X}_a и \mathbf{X}_b – множества атрибутов объектов A и B соответственно;

4) интервальными, отражающими количественные отношения интервалов: шкала единственна с точностью до линейных преобразований, т.е. $y = ax + b$, $a > 0$, $-\infty < b < \infty$, или $y = ax$, $a \neq 0$, или $y = x + b$.

В большинстве случаев структурного синтеза математическая модель в виде алгоритма, позволяющего по заданному множеству \mathbf{X} и заданной структуре объекта рассчитать вектор критериев \mathbf{K} , оказывается известной. Например, такие модели получаются автоматически в программах анализа типа *Spice*, *Adams* или ПА–9 для объектов, исследуемых на макроуровне. Однако в ряде других случаев такие модели неизвестны в силу недостаточной изученности процессов и их взаимосвязей в исследуемой среде, но известна совокупность результатов наблюдений или экспериментальных исследований. Тогда для получения моделей используют специальные *методы идентификации и аппроксимации* (модели, полученные подобным путем, иногда называют феноменологическими).

Среди методов формирования моделей по экспериментальным данным наиболее известны *методы планирования экспериментов*. Не менее популярным становится подход, основанный на использовании *искусственных нейронных сетей*.

Если же математическая модель $\mathbf{X} \rightarrow \mathbf{K}$ остается неизвестной, то стараются использовать подход на базе *систем искусственного интеллекта (экспертных систем)*.

Возможности практического решения задач *дискретного математического программирования* (ДМП) изучаются в теории сложности задач выбора, где показано, что задачи даже умеренного размера, относящиеся к классу **NP**–полных задач, в общем случае удается решать только приближенно.

Поэтому большинство практических задач структурного синтеза решают с помощью приближенных (эвристических) методов. Это методы, использующие специфические особенности того или иного класса задач и не гарантирующие получения оптимального решения. Часто они приводят к результатам, близким к оптимальным, при приемлемых затратах вычислительных ресурсов.

7.3.3. Представление множества альтернатив

Решению проблем упорядочения и описания множества альтернатив и связей между ними в конкретных приложениях посвящена специальная область знания, которую по аналогии с наукой описания множеств животных и растений в биологии можно назвать *систематикой*.

Простейший способ задания множества A – явное перечисление всех альтернатив. Семантика и форма описания альтернатив существенно зависят от приложения. Для представления таких описаний в памяти ЭВМ и доступа к ним используют *информационно-поисковые системы* (ИПС). Каждой альтернативе в ИПС соответствует поисковый образ, состоящий из значений атрибутов x_i и ключевых слов вербальных характеристик.

Явное перечисление альтернатив при представлении множества альтернатив возможно лишь при малой мощности A . Поэтому в большинстве случаев используют неявное описание A в виде способа (алгоритма или набора правил P) синтеза проектных решений из ограниченного набора элементов \mathcal{E} . Поэтому здесь $A = \langle P, \mathcal{E} \rangle$, а типичный процесс синтеза проектных решений состоит из следующих этапов:

- 1) формирование альтернативы A_i (это может быть выбор из базы данных ИПС по сформированному поисковому предписанию или генерация из \mathcal{E} в соответствии с правилами P);
- 2) оценка альтернативы по результатам моделирования с помощью модели Мод;
- 3) принятие решения (выполняется ЛПР – лицом, принимающим решение, или автоматически) относительно перехода к следующей альтернативе или прекращения поиска.

Для описания множеств P и \mathcal{E} используют следующие подходы.

1. *Морфологические таблицы* и *альтернативные И-ИЛИ-деревья*.
2. Представление знаний в *интеллектуальных системах* – фреймы, семантические сети, продукции.
3. Генетические методы.
4. Базы *физических эффектов* и *эвристических приемов*, применяемые при решении задач изобретательского характера.

7.4.4. Морфологические таблицы

Морфологическая таблица (M) представляет собой обобщенную структуру в виде множества функций, выполняемых компонентами синтезируемых объектов рассматриваемого класса, и подмножеств способов их реализации. Каждой функции можно поставить в соответствие одну строку таблицы, каждому способу ее реализации – одну клетку в этой строке. Следовательно, в морфологических таблицах элемент M_{ij} означает j -й вариант реализации i -й функции в классе технических объектов, описываемом матрицей M .

Другими словами, множество альтернатив можно представить в виде отношения M , называемого морфологической таблицей

$$\mathbf{M} = \langle \mathbf{X}, \mathbf{R} \rangle,$$

где \mathbf{X} – множество свойств (характеристик или функций), присущих объектам рассматриваемого типа; n – число этих свойств; $\mathbf{R} = \langle \mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_n \rangle$, \mathbf{R}_i – множество значений (способов реализации) i -го свойства, мощность этого множества далее обозначена N_i . При этом собственно множество альтернатив \mathbf{A} представлено композицией множеств \mathbf{R}_i , т.е. каждая альтернатива включает по одному элементу (значению) из каждой строки морфологической таблицы. Очевидно, что общее число альтернатив k , представляемых морфологической таблицей, равно

$$k = \prod_{i=1}^n N_i$$

Морфологические таблицы обычно считают средством неавтоматизированного синтеза, помогающим человеку просматривать компактно представленные альтернативы, преодолевать психологическую инерцию. Последнее связано с тем, что внимание ЛПР обращается на варианты, которые без морфологической таблицы оставались бы вне его поля зрения.

Собственно таблица \mathbf{M} не содержит сведений о способе синтеза. Однако на базе \mathbf{M} возможно построение методов синтеза с элементами алгоритмизации. В таких методах вводится метризация морфологического пространства. Морфологическое пространство составляют возможные законченные структуры, принимается, что расстояние между структурами \mathbf{C}_1 и \mathbf{C}_2 есть число несовпадающих элементов (каждая клетка \mathbf{M} есть один элемент). Поэтому можно говорить об окрестностях решений. Далее исходят из предположения о компактности «хороших» решений, которое позволяет вместо полного перебора ограничиваться перебором в малой окрестности текущей точки поиска. Таким образом, гипотеза о «компактности» и метризация пространства решений фактически приводят к построению математической модели, к которой можно применить методы дискретной оптимизации, например локальные методы.

К недостаткам \mathbf{M} относятся неучет запрещенных сочетаний элементов в законченных структурах и отражение состава элементов в структурах без конкретизации их связей. Кроме того, морфологические таблицы строят в предположении, что множества \mathbf{R}_i взаимно независимы, т.е. состав способов реализации i -й функции не меняется при изменении значений других функций. Очевидно, что предположение о взаимной независимости множеств \mathbf{R}_i оправдано лишь в сравнительно простых структурах. Последний недостаток устраняется путем обобщения метода морфологических таблиц при использовании метода альтернативных (И–ИЛИ) графов.

7.4.5. Альтернативные графы

Любую морфологическую таблицу можно представить в виде дерева (рис. 7.12). На рисунке функции представлены вершинами И (темные кружки), значения функций – вершинами ИЛИ (светлые кружки). Очевидно, что таблица представляет множество однотипных объектов, поскольку все они характеризуются одним и тем же множеством функций.

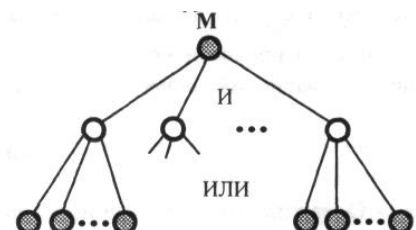


Рис. 7.12. Дерево, соответствующее морфологической таблице

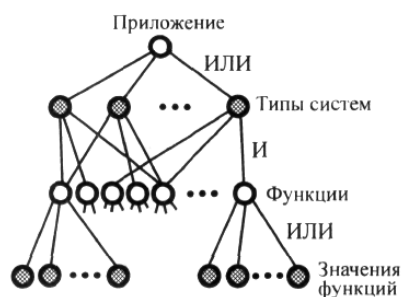


Рис. 7.13. И-ИЛИ-граф

Для разнотипных объектов применяют многоярусные альтернативные графы. Например, на рис. 7.13 показан двух-ярусный граф, в котором для разных типов объектов предусмотрены разные подмножества функций.

Если допустить некоторую избыточность при изображении И-ИЛИ-графа, то его можно превратить в И-ИЛИ-дерево, что ведет к определенным удобствам.

Очевидно, что И-ИЛИ-дерево можно представить как совокупность морфологических таблиц. Каждая И-вершина дерева соответствует частной морфологической таблице, т. е. множеству функций, так, что i -я выходящая ветвь отображает i -ую функцию. Каждая ИЛИ-вершина, инцидентная i -й ветви, соответствует множеству вариантов реализации i -й функции, при этом j -я исходящая из ИЛИ-вершины ветвь отображает j -й вариант реализации.

Алгоритмизация синтеза на базе И-ИЛИ-деревьев требует введения правил выбора альтернатив в каждой вершине ИЛИ. Эти правила чаще всего имеют эвристический характер, связаны с требованиями технического задания, могут отражать запреты на сочетания определенных компонентов структур.

Трудности эффективного решения задачи существенно возрастают при наличии ограничений, типичными среди которых являются ограничения на совместимость способов реализации разных функций, т.е. ограничения вида

$$C_{ij} \text{ and } C_{pq} = \text{false}, \quad (7.29)$$

где $C_{ij} = \text{true}$, если в оцениваемый вариант вошел элемент \mathcal{E}_{ij} , иначе $C_{ij} = \text{false}$. Условие (7.29) означает, что в допустимую структуру не могут входить одновременно элементы \mathcal{E}_{ij} и \mathcal{E}_{pq} . Совокупность ограничений типа (7.29) можно представить как систему логических уравнений с неизвестными C_{ij} . Тогда задачу синтеза можно решать эволюционными методами, если предварительно или одновременно с ней решать систему логических уравнений (задачу о выполнимости).

7.4.6. Исчисления

Очевидно, что в большинстве случаев структурного синтеза вместо нереализуемого явного представления всего множества проектных решений задают множество элементов и совокупность правил объединения этих элементов в допустимые структуры (проектные решения).

Эти множества элементов и правил часто представляют в виде *формальной системы* (исчисления), т.е. задача синтеза (ЗС) имеет вид

$$\text{ЗС} = \langle \mathcal{E}; \text{НТ}; \text{АК}; \Pi \rangle,$$

где \mathcal{E} – алфавит исчисления (алфавит представлен базовыми элементами, из которых синтезируется структура); **НТ** – множество букв, не совпадающих с буквами алфавита \mathcal{E} и служащих для обозначения переменных; **АК** – множество аксиом исчисления, под которыми понимают задаваемые исходные формулы (слова) в алфавите \mathcal{E} (например, соответствия функций и элементов); **Π** – множество правил вывода новых формул в алфавите \mathcal{E} из аксиом и

ранее выведенных корректных формул. Каждую формулу можно интерпретировать как некоторую структуру, поэтому синтез – это процесс вывода формулы, удовлетворяющей исходным требованиям и ограничениям.

Другие примеры компактного задания множества альтернатив **A** через множества **Э** и **П** связаны с использованием систем искусственного интеллекта, в которых **Э** есть база данных, **П** – база знаний, или эволюционных методов; или в которых **Э** – также база данных, **П** – множество эвристик, последовательность применения которых определяется эволюционными и генетическими принципами.

7.4.7. Планирование процессов и распределение ресурсов

В отдельную группу задач структурного синтеза выделяют планирование процессов и распределение ресурсов. В нее входят синтез технологических процессов в различных отраслях промышленности, проектирование вычислительных процессов для многопроцессорных систем и сетей, синтез логистических процессов (например, планирование перевозок грузов при наличии множества заказов и ограниченном числе транспортных средств), а также планирование работ при управлении проектами. Эти задачи объединяет общность ряда свойств и подходов к решению, как задач синтеза расписаний.

Базовым понятием в синтезе расписаний является понятие *работы* – элементарной планируемой части процесса. Нужно составить план выполнения работ, в котором фиксируются объемы работ, распределение ресурсов всех видов, моменты (даты) начала и окончания каждой работы, называемые *событиями* (или *вехами*), стоимости работ. *Ресурсы* – обеспечивающие компоненты деятельности, включающие исполнителей, энергию, материалы, оборудование и т.д.

С каждой работой можно связать функцию потребности в ресурсах. Различают ресурсы унарные и объемные. Единица *унарного* ресурса, называемая далее сервером, может одновременно выполнять не более одной работы, и по каждому виду ресурса на работу может быть назначен не более чем один сервер. Примерами унарных ресурсов могут быть токарный станок, процессор ЭВМ, водитель автомашины. Значение объемного ресурса (энергии, финансов, пропускной способности канала), назначаемое для конкретной работы, может быть выбрано в некотором диапазоне и от выбранного значения зависят длительность и (или) стоимость выполнения работы.

Результаты синтеза обычно представляют в виде таблиц и диаграмм. *PERT-диаграмма* (Program Evaluation and Review Technique) – сеть типа вершина–событие – ориентированный граф без контуров, имеющий одну исходную и одну завершающую вершины, в котором вершины поставлены в соответствие событиям, а дуги – работам. *Диаграмма Ганта* (Gantt diagram) – горизонтальная линейная диаграмма, на которой задачи проекта представляются протяженными во времени отрезками, распределенными между серверами и характеризующимися датами начала и окончания, задержками и возможно другими временными параметрами.

Для задач планирования процессов и управления проектами характерны следующие черты:

- 1) широкий диапазон размеров задач, причем верхняя граница диапазона может достигать

значений в десятки тысяч и более работ;

2) многокритериальность, основные критерии – время и стоимость выполнения плана, в качестве целевой функции часто выбирают стоимость, в число ограничений включают времена окончания работ и, возможно, ряд условий использования ресурсов;

3) разнообразие типов управляемых переменных, среди которых могут быть величины вещественные, целые, нечисловые.

Указанные особенности обуславливают сложность решения задач синтеза расписаний, являющихся задачами дискретной оптимизации.

7.4. Методы структурного синтеза в САПР

7.4.1. Метод ветвей и границ

Выбор метода поиска решения – вторая проблема после формализации задачи. Если при формализации все управляемые параметры удалось представить в числовом виде, то можно попытаться применить известные методы дискретного математического программирования.

Задача дискретного математического программирования определяется следующим образом:

$$\text{extr}_{X \in D} F(X), \quad (7.30)$$

$$D = \{X \mid W(X) > 0, Z(X) = 0\},$$

где $F(X)$ – целевая функция; $W(X)$, $Z(X)$ – вектор–функции, связанные с представленными в техническом задании требованиями и ограничениями; D – дискретное множество. В полученной модели, во–первых, каждый элемент множества рассматриваемых законченных структур должен иметь уникальное сочетание значений некоторого множества числовых параметров, вектор которых обозначим X . Во–вторых, необходимо существование одной или нескольких функций $\Phi(X)$, значения которых могут служить исчерпывающей оценкой соответствия структуры предъявляемым требованиям. В–третьих, функции $\Phi(X)$ должны отражать внутренне присущие данному классу объектов свойства, что обеспечит возможность использования $\Phi(X)$ в качестве не только средств оценки достигнутого при поиске успеха, но и средств указания перспективных направлений продолжения поиска. Эти условия выполнимы далеко не всегда, что и обуславливает трудности формализации задач структурного синтеза.

Однако наличие формулировки (7.30) еще не означает, что удастся подобрать метод (алгоритм) решения задачи (7.30) с приемлемыми затратами вычислительных ресурсов. Другими словами, применение точных методов математического программирования вызывает непреодолимые трудности в большинстве случаев практических задач типичного размера из–за их принадлежности к классу NP–трудных задач. Поэтому лидирующее положение среди методов решения задачи (7.30) занимают приближенные методы, в частности, декомпозиционные методы, отражающие принципы блочно–иерархического проектирования сложных объектов. Декомпозиционные методы основаны на выделении ряда иерархических уровней, на каждом из которых решаются задачи приемлемого размера.

Основу большой группы математических методов, выражающих стремление к сокращению перебора, составляют операции разделения множества вариантов на подмножества и отсеечения неперспективных подмножеств. Эти методы объединяются под названием *метода*

ветвей и границ. Основная разновидность метода ветвей и границ относится к точным методам решения комбинаторных задач. Рассмотрим эту разновидность.

Пусть имеется множество решений \mathbf{M} , в котором нужно выбрать оптимальный по критерию $F(\mathbf{X}_j)$ вариант, где \mathbf{X}_j – вектор параметров варианта $m_j \in \mathbf{M}$. Пусть также имеется алгоритм для вычисления нижней границы $L(\mathbf{M}_k)$ критерия $F(\mathbf{X}_j)$ в любом подмножестве \mathbf{M}_k множества \mathbf{M} , т.е. такого значения $L(\mathbf{M}_k)$, что $F(\mathbf{X}_j) > L(\mathbf{M}_k)$ при любом j (подразумевается минимизация $F(\mathbf{X})$).

Тогда основная схема решения задач в соответствии с методом ветвей и границ содержит следующие процедуры:

- 1) в качестве \mathbf{M}_k принимаем все множество \mathbf{M} ;
- 2) ветвление: разбиение \mathbf{M}_k на несколько подмножеств \mathbf{M}_q ;
- 3) вычисление нижних границ $L(\mathbf{M}_q)$ в подмножествах \mathbf{M}_q ;
- 4) выбор в качестве \mathbf{M}_k подмножества \mathbf{M}_p с минимальным значением нижней границы критерия (среди всех подмножеств, имеющих на данном этапе вычислений), сведения об остальных подмножествах \mathbf{M}_q и их нижних границах сохраняются в отдельном списке;
- 5) если $|\mathbf{M}_k| > 1$, то переход к процедуре 2), иначе одноэлементное множество \mathbf{M}_k есть решение.

Метод ветвей и границ в случае точного вычисления нижних границ относится к точным методам решения задач выбора и потому в неблагоприятных ситуациях может приводить к экспоненциальной временной сложности. Однако метод часто используют как приближенный, поскольку можно применять приближенные алгоритмы вычисления нижних границ.

7.4.2. Системы искусственного интеллекта

В теории интеллектуальных систем синтез реализуется с помощью экспертных систем (ЭС)

$$\text{ЭС} = \langle \text{БД}, \text{БЗ}, \text{И} \rangle,$$

где **БД** – база данных, включающая сведения о базовых элементах; **БЗ** – база знаний, содержащая правила конструирования вариантов структуры; **И** – интерпретатор, устанавливающий последовательность применения правил из **БЗ**.

Системы искусственного интеллекта (СИИ) основаны на знаниях, отделенных от процедурной части программ и представленных в одной из характерных форм. Такими формами могут быть продукции, фреймы, семантические сети. Реально функционирующие в современных САПР системы с базами знаний чаще всего относятся к классу ЭС.

Продукция представляет собой правило типа «если A , то B », где A – условие, а B – действие или следствие, активизируемое при истинности A . Продукционная **БЗ** содержит совокупность правил, описывающих определенную предметную область.

Фрейм – структура данных, в которой в определенном порядке представлены сведения о свойствах описываемого объекта. Типичный вид фрейма:

$$\langle \text{имя фрейма}; x_1 = p_1; x_2 = p_2; \dots; x_N = p_N; q_1, q_2, \dots, q_M \rangle,$$

где x_i – имя i -го атрибута, p_i – его значение, q_i – ссылка на другой фрейм или некоторую обслуживающую процедуру. В качестве p_i можно использовать имя другого (вложенного) фрейма, описывая тем самым иерархические структуры фреймов.

Семантическая сеть – форма представления знаний в виде совокупности понятий и явно выраженных отношений между ними в некоторой предметной области. Семантическую сеть удобно представлять в виде графа, в котором вершины отображают понятия, а ребра или дуги – отношения между ними. В качестве вершин сети можно использовать фреймы или продукции.

Экспертная система является типичной системой искусственного интеллекта, в которой **БЗ** содержит сведения, полученные от людей–экспертов в конкретной предметной области. Трудности формализации процедур структурного синтеза привели к популярности применения экспертных систем в САПР, поскольку в них вместо выполнения синтеза на базе формальных математических методов осуществляется синтез на основе опыта и неформальных рекомендаций, полученных от экспертов.