Л и т е р а т у р а

1. Б и р ю к В.И., Л и п и н Е.К., Ф р о л о в В.М. Методы проектирования конструкций самолетов. М.: Машиностроение, 1977.

2. К о м а р о в В.А., С о л о в о в А.В. Об опыте автоматизации проектирования силовых схем крыльев. В сб.: Материалы Всероссийской школы 1975 года по автоматизации проектирования. МФТИ, 1976.

3. У г о д ч и к о в А.Г. Численные методы и ЭВМ в решении проблем прочности. В сб.: Прикладные проблемы прочности и пластичности. Горький, 1975, вып. 1.

УДК 629.7.658.51:519.87

А.В. Соллогуб, Л.З. Чернис

ЗАДАЧА СИНТЕЗА МНОГОЭТАПНОГО ПРОЦЕССА ПРОЕКТИРОВАНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ КОМПЛЕКСОВ

Предполагается, что создание сложного технического комплекса (ТК) осуществляется в результате выполнения n последовательных этапов. В зависимости от сложности S создаваемого ТК (считаем, что введена шкала сложности ТК) и имеющегося опыта $\bar{L}$ (наличие прототипа, задела технической документации и методик, возможности заимствования отдельных элементов ТК) может быть определена трудоемкость каждого этапа создания ТК:

$$k_i^* = k_i^*(S, \bar{L}) \qquad (i=1,2,\ldots,n).$$

Трудоемкость k_i^* выражается в единицах основного продукта труда i-го этапа. Это может быть количество рассмотренных вариантов технических решений, количество листов технической документации или чертежей стандартного формата, количество типовых инструкций, методик, алгоритмов, количество проведенных испытаний и др.

Состояние выполнения i-го этапа в момент времени t_i с его начала может быть охарактеризовано двумя величинами:

$K_i(t_i)$ – количество произведенных единиц продукта труда i -го этапа, $P_i(t_i)$ – количество единиц продукта труда, выполненных с ошибками. Величину $P_i(t_i)$ будем называть загрязнением. Загрязнение $P_i(t_i)$ измеряется в тех же единицах, что и $K_i(t_i)$.

Введем понятие относительного загрязнения

$$\alpha_i(t_i) = P_i(t_i) / K_i(t_i).$$

Очевидно, на количество ошибок, возникающих на i -ом этапе, влияют ошибки, уже допущенные на предыдущих j -х этапах (j =1,2,..., i -1).

Количественно это влияние можно выразить через относительные загрязнения в конце предыдущих этапов:

$$\alpha_1^* = \alpha(t_1)\Big|_{t_1=\tau_1} = \alpha_i(\tau_1), \ldots, \alpha_{i-1}^* = \alpha_{i-1}(\tau_{i-1}),$$

где τ_j (j =1,2,..., i -1) - продолжительности j -ых этапов.

Загрязнение отрицательно сказывается на работе над проектом. Коллектив разработчиков стремится сдать заказчику законченный проект с уровнем загрязнения, не превышающим некоторый заранее заданный уровень. Поэтому в ходе выполнения проекта с загрязнением необходимо бороться. Часть труда затрачивается на поиск, локализацию и устранение ошибок, что приводит к необходимости выполнения повторных расчетов, повторных испытаний или экспериментальных работ, внесения изменений в чертежную и другую техническую документацию. Таким образом, в каждый момент времени t_i на i -ом этапе определяется доля труда β_i, идущая на борьбу с загрязнением и доля $1-\beta_i$, идущая на прямое выполнение проекта.

Существенную роль на состояние выполнения i -го этапа оказывает интенсивность труда $\Sigma_i(t_i)$ разработчиков проекта, занятых на i -ом этапе. С возрастанием $\Sigma_i(t_i)$ количество ошибок увеличивается, растет загрязнение $\alpha_i(t_i)$, время τ_i выполнения этапа сокращается. Уменьшение $\Sigma_i(t_i)$ приводит к увеличению τ_i и уменьшению загрязнения $\alpha_i(t_i)$.

Рассмотрим задачу: определить в каждый момент времени на каждом этапе проектирования интенсивность труда $\Sigma_i(t_i)$ и долю его $\beta_i(t_i)$, идущую на борьбу с загрязнением, так, чтобы в целом время выполнения проекта было минимальным, а относительное загрязнение в конце последнего этапа $\alpha_n(t_n)\big|_{t_n=\tau_n}$ не превышало α_n^*, то есть $\tau = \sum_{i=1}^{n}\tau_i \to min$ при условии $\alpha_n(\tau_n) \le \alpha_n^*$. При этом план работы, соответствующий каждому этапу , должен быть выполнен:

$$K_i(\tau_i) \geqslant K_i^* \quad (i=1,2,\dots,n).$$

Для решения задачи удобно рассматривать вспомогательную задачу. Предположим, что этапы проектирования, предшествующие i-ому этапу уже выполнены, причем продукция, произведенная на этих этапах, имеет относительное загрязнение $\alpha_1^*, \alpha_2^*, \dots, \alpha_{i-1}^*$ соответственно. Требуется в каждый момент времени t_i выполнения i-го этапа определить интенсивность труда $\sum_i(t_i)$ и долю его $\beta_i(t_i)$, идущую на борьбу с загрязнением, так, чтобы время τ_i выполнения i-го этапа было минимальным, при условии, что загрязнения в конце предыдущих этапов равны $\alpha_1^*, \dots, \alpha_{i-1}^*$, а в конце i-го этапа не превышают α_i^*, то есть $\tau_i \to min$ при условии $\alpha_i(\tau_i) \leqslant \alpha_i^*$, если известны $\alpha_j(\tau_j) = \alpha_j^* (j=1,2,\dots i-1)$. Здесь также подразумевается выполненным условие завершения работ на i-ом этапе:

$$K_i(\tau_i) \geqslant K_i^*.$$

Сформулированная вспомогательная задача относится к классу задач оптимального быстродействия. Управлениями здесь являются функции $\beta_i(t_i)$ и $\sum_i(t_i)$. В результате решения задачи получим: $\tau_i^{opt} = \tau_i^{opt}(\alpha_1^*, \dots, \alpha_i^*)$ – минимальное время выполнения i-го этапа при заданных относительных загрязнениях $\alpha_j^* (j=1,2,\dots,i)$ и оптимальные управления $\beta_i^{opt}(t_i)$, $\sum_i^{opt}(t_i)$ $(0 \leqslant t_i \leqslant \tau_i)$.

Для упрощения задачи будем считать в дальнейшем, что минимальное время выполнения i-го этапа зависит только от заранее заданного значения α_i^* и известного загрязнения в конце предыдущего этапа α_{i-1}^*, то есть $\tau_i^{opt} = \tau_i^{opt}(\alpha_{i-1}^*, \alpha_i^*)$.

Предположим, что мы эффективно решаем вспомогательную задачу, то есть для всех $i=1,2,\dots,n$ и любых α_{i-1}^*, α_i^* можем вычислить $\tau_i^{opt}(\alpha_{i-1}^*, \alpha_i^*)$. Это дает возможность решить основную задачу.

Обозначим через $T^{(k)}(\alpha_k^*)$ минимальное время выполнения части проекта, включающей k первых этапов, при условии, что относительное загрязнение в конце k-го этапа не превышает α_k^*. Тогда схема определения минимального времени выполнения всего проекта из n этапов при условии, что в конце последнего этапа относительное загрязнение не превышает α_n^*, примет вид:

$$k=1 \qquad T^{(1)}(\alpha_1^*) = \tau_1^{opt}(\alpha_1^*), \ \alpha_1^* \in A_1^*$$

$$k=2 \qquad T^{(2)}(\alpha_2^*) = \min_{\alpha_1^*} \left[T^{(1)}(\alpha_1^*) + \tau_2^{opt}(\alpha_1^*, \alpha_2^*)\right], \ \alpha_2^* \in A_2^*$$

. . . .

$$k=s \qquad T^{(s)}(\alpha_s^*) = \min_{\alpha_{s-1}^*} \left[T^{(s-1)}(\alpha_{s-1}^*) + \tau_s^{opt}(\alpha_{s-1}^*, \alpha_s^*)\right], \ \alpha_s^* \in A_s^*$$

. . . .

$$k=n \qquad T^{(n)}(\alpha_n^*) = \min_{\alpha_{n-1}^*} \left[T^{(n-1)}(\alpha_{n-1}^*) + \tau_n^{opt}(\alpha_{n-1}^*, \alpha_n^*)\right], \ \alpha_n^* \in A_n^*$$

где $A_i^* (i=1,2,\dots n)$ – множества значений α_i^*, для которых осуществляется табулирование функций $T^{(i)}(\alpha_i^*)$ на i-ом этапе.

Для выполнения этого алгоритма достаточно предварительно для всех $i = 1,2,\dots,n$ протабулировать $\tau_i^{opt} = \tau_i^{opt}(\alpha_{i-1}^*, \alpha_i^*)$ по α_{i-1}^* и α_i^*. Тем самым мы решим задачу для всех α_n^*, заданных в таблице функции $\tau_n^{opt} = \tau_n^{opt}(\alpha_{n-1}^*, \alpha_n^*)$.

Составляя на прямом шаге работы алгоритма для каждого этапа i ($i = 1,2,\dots,n$) таблицу

$$< \alpha_i^*, T^i(\alpha_i^*), \alpha_{i-1}^* >,$$

где α_i^* пробегает все значения из множества A_i^*, а α_{i-1}^* – значение относительного загрязнения на предыдущем этапе, обеспечивающее совместно со значением α_i^* минимальное время $T^i(\alpha_i^*)$, и просматривая полученные для всех этапов таблицы в обратном направлении, можно определить для каждого значения α_n^* цепочку значений относительных загрязнений $\alpha_n^*, \alpha_{n-1}^*, \dots, \alpha_1^*$, по этапам, обеспечивающих минимальное время выполнения проекта $T^{(n)}(\alpha_n^*)$. Далее, принимая во внимание результаты решения вспомогательных задач, где помимо аргументов α_{i-1}^*, α_i^* и значений функций $\tau_i^{opt}(\alpha_{i-1}^*, \alpha_i^*)$ хранятся оптимальные управления

$$\beta_i^{opt}(t_i), \ \Sigma^{opt}(t_i) \quad (0 \le t_i \le \tau_i)$$

легко отыскать оптимальные управления $\beta^{opt}(t)$, $\Sigma^{opt}(t) (0 \le t \le T^{(n)}(\alpha_n^*))$ для всего процесса реализации проекта в целом.

Если функции $\tau_i^{opt}(\alpha_{i-1}^*, \alpha_i^*)$ табулированы в L точках по каждой переменной (всего L^2 точек), то на выполнение алгоритма требуется не более nL^2 операций. Если на ход работы на i -ом этапе влияют относительные загрязнения нескольких, например, ν предыдущих этапов, то алгоритм несколько изменяется: на каждом шаге вычисляется величина

$$T^s(\alpha_s^*) = \min_{\alpha_{s-1}^*,\dots,\alpha_{s-\nu}^*} \left[T^{(s-1)}(\alpha_{s-1}^*) + \tau_s^{opt}(\alpha_{s-\nu}^*,\dots,\alpha_s^*) \right]$$

и табулируется по α_s^* .

Таким образом, если функции $\tau_i^{opt}(\alpha_{i-s}^*,\dots,\alpha_i^*)$ заданы таблицей в L точках по каждой переменной, то на выполнение алгоритма потребуется не более nL^{s+1} операций.

В целом, задача , решаемая алгоритмом, такова: при заданном количестве проектной продукции K_i^* (i =1,2,...,n) для каждого этапа и заданном загрязнении α_n^* в конце последнего этапа выбирается такой набор конечных загрязнений ($\alpha_1^*,\dots,\alpha_{n-1}^*$) и управлений по этапам, которые в целом обеспечивают выполнение проекта за минимальное время.

Процесс накопления проектной продукции и загрязнения на i -ом этапе в общем случае может быть описан дифференциальными уравнениями вида:

$$\dot{K}_i = f_{1i}(K_i, P_i, \beta_i, \Sigma_i, \alpha_{i-1}),$$
$$\dot{P}_i = f_{2i}(K_i, P_i, \beta_i, \Sigma_i, \alpha_{i-1}).$$

Уравнения учитывают, что скорости роста продукции и загрязнения на i -ом этапе в каждый момент времени t зависят от произведенных на этом этапе продукции $K_i(t)$ и загрязнения $P_i(t)$, доли $\beta_i(t)$ труда, затрачиваемого в момент t на борьбу с загрязнением, интенсивности труда $\Sigma_i(t)$ и относительного загрязнения в конце предыдущего этапа α_{i-1}^* .

Принимая во внимание, что качественная картина накопления продукции и загрязнения от этапа к этапу не меняется, опустим везде индекс i . Тогда вспомогательная задача может быть сформулирована как задача максимального быстродействия ($T \to \min$) для траектории, описываемой уравнениями

$$\dot{K} = f_1(K, P, \beta, \Sigma, \alpha_{-1}^*),$$
$$\dot{P} = f_2(K, P, \beta, \Sigma, \alpha_{-1}^*)$$

с закрепленными концами

$$k(0)=0\ ,\quad k(\tau)=k^*$$
$$P(0)=P\ ,\quad P(\tau)=\alpha^* k^*.$$

Ограничения на фазовые переменные:

$$k(t)\geqslant 0,\ P(t)\geqslant 0,\ \dot{k}(t)\geqslant 0,$$

то есть количество продукции и загрязнение неотрицательны и количество продукции не убывает.

Ограничения на управления: $0\leqslant\beta(t)\leqslant 1$, $\sum(t)\geqslant 1$, то есть доля труда, идущая на борьбу с загрязнением, неотрицательна и не превышает весь труд в данный момент, а интенсивность труда может изменяться от нормальной и выше.

Будем предполагать в дальнейшем, что на текущем этапе мы можем бороться не только с загрязнением, возникшим на данном этапе, но и с загрязнением, доставшимся от предыдущего этапа и выраженным в его единицах.

Рассмотрим модель этапа, описываемую системой дифференциальных уравнений вида:

$$\begin{cases} \dot{k}=(1-\beta-\beta')\sum b & \text{(I)} \\ \dot{P}=(1-\beta-\beta')\sum\left[j+a\alpha+a'\alpha_{-1}+h(\sum)\right]-\beta\sum d & \text{(2)} \\ \dot{P}_{-1}=-\beta'\sum d' & \text{(3)} \end{cases}$$

$$k(0)=P(0)=0\ ,\quad P_{-1}(0)=P_{-1}^*$$
$$k(\tau)=k^*;\quad P(\tau)=\alpha^* k^*,\quad P_{-1}(\tau)=0.$$

Здесь, как и прежде,

$$\alpha(t)=P(t)/k(t),$$
$$\alpha_{-1}(t)=P_{-1}(t)/k_{-1}^*.$$

Уравнение (I) показывает, что прирост продукции пропорционален интенсивности труда $\sum(t)$ и доле труда ($1-\beta-\beta'$), идущей на продуктивную работу (а не на борьбу с загрязнением) в данный момент времени. Здесь β – доля труда, затрачиваемого на борьбу с загрязнением, возникшим на данном этапе, β' – доля труда, идущего на борьбу с загрязнением с предыдущего этапа, b – коэффициент пропорциональности, учитывающий прирост продукции в единицу времени при $\sum(t)=1$ и $\beta+\beta'=0$.

Уравнение (2) показывает, что прирост загрязнения складывается из следующих факторов:

I. В процессе продуктивного труда возникают ошибки пропорционально приросту продукции вследствие:

а) того, что люди, а также средства автоматизации проектирования работают не безошибочно;

б) ошибок, уже допущенных на текущем этапе;

в) ошибок, допущенных на предыдущем этапе;

г) повышения интенсивности труда.

Эти факторы учитываются соответственно коэффициентами $j, a, a', h(\Sigma)$.

2. Уменьшение загрязнения происходит пропорционально доле труда, идущего на борьбу с ним, и интенсивности этого труда.

Член $a'\alpha_{-1}(t)$ показывает, что загрязнение с предыдущего этапа P_{-1} " переходит " в текущее загрязнение $P(t)$. Уравнение (3) описывает закон уменьшения, доставшегося с предыдущего этапа.

Следует отметить, что вследствие быстрого роста $h(\Sigma)$ верхнее ограничение на Σ не требуется, так как интенсивность труда из условия оптимальности не будет высокой.

Если положить

$$x = (K, P, P_{-1}), \quad u = (\beta, \beta', \Sigma),$$

где $x \in X = \{(K, P, P_{-1}) : K \geq 0,\ p \geq 0,\ P_{-1} \geq 0\}$,

$$u \in U = \{(\beta, \beta', \Sigma) : \Sigma \geq 1,\ \beta \geq 0,\ \beta' \geq 0,\ \beta + \beta' \leq 1\},$$

то алгоритм, решающий задачу быстродействия, может быть проиллюстрирован на системе вида

$$\dot{x} = f(x, u),\ x \in X,\ u \in U$$

где векторы x и u имеют одинаковую размерность ($x, u \in R^3$).

Заменим связи $\dot{x} = f(x, u)$ разностными уравнениями

$$x_j = x_{j-1} + \Delta t \cdot f(x_{j-1}, u_{j-1}),\ j = 1, 2, \ldots$$

где начальное состояние $x_0 = (0, 0, P_{-1}^*)$ задано.

Ограничения $x(t) \in X$ и $u(t) \in U$ заменяем ограничениями $x_j \in X$, $u_j \in U$.

Введем в R^3 сетку. Каждая компонента вектора x может принимать дискретный ряд значений. Таким образом вектор x принадлежит конечному подмножеству $\tilde{X} \subset X$.

Рассмотрим алгоритм формирования множеств достижимостей на каждом шаге процесса.

I) Определим все возможные состояния x_1, в которые можно попасть из состояния x_0 за один шаг, то есть за время Δt:

$$x_1 = x_0 + \Delta t f(x_0, u_0).$$

Так как x_1 и u_0 векторы одинаковой размерности, то $u_0 = \varphi(x_0, x_1)$.
Положим

$$X^{(1)} = \{x_1 \mid x_1 \in \tilde{X},\ u_0 = \varphi(x_0, x_1) \in U\}.$$

2) Определим все возможные состояния x_2, в которые можно попасть из состояний $x_1 \in X^{(1)}$ за время Δt:

$$X^{(2)} = \{x_2 \mid x_2 \in \tilde{X},\ \exists x_1 \in X^{(1)} : u_1 = \varphi(x_1, x_2) \in U\}$$

$$X^{(j)} = \{x_j \mid x_j \in \tilde{X},\ \exists x_{j-1} \in X^{(j-1)} : u_{j-1} = \varphi(x_{j-1}, x_j) \in U\}.$$

Искомое время

$$\tau = \Delta t \cdot \min\{j \mid x^* \in X^{(j)}\},$$

где

$$x^* \in X^* = \{(k, p, p_{-1}) : k \geqslant k^*, p \leqslant p^*, p_{-1} = 0\}.$$

Пусть $x_k = X^*$ – конец оптимальной траектории. Определяя последовательно величины $x_i \in X^{(i)}$ ($i = k-1, k-2, \dots, 0$), такие, что $u_i = \varphi(x_i, x_{i+1}) \in U$, находим оптимальную траекторию $x_0, x_1, \dots, x_k$ и соответствующее оптимальное управление $u_0, u_1, \dots, u_{k-1}$.
В случае, если решение задачи неоднозначно, в качестве оптимального выбирается любое из полученных решений.

Рассмотрим применение алгоритма к решению вспомогательной задачи. Нам необходимо протабулировать функцию $\tau(\alpha_{-1}^*, \alpha^*)$ по обоим аргументам. Задаем сетку по α_{-1}^* и α^*. Этим самым задается сетка по $P_{-1}^* \alpha_{-1}^*, k_{-1}^*$ и по $P^* = \alpha^* k^*$. Эта же сетка принимается для P_{-1} и P. Кроме того, выберем ветку по k. В результате получена сетка во множестве X. Алгоритм формирования допустимых множеств выполняется до тех пор, пока не будет определено минимальное время попадания фазовой точки в область, задаваемую множеством X^*. Это гарантирует выпуск продукции $k \geqslant k^*$, наличие в конце этапа загрязнения $p \leqslant p^*$ и устранение загрязнения, доставшегося с предыдущего этапа. Здесь k^* заданное фиксированное для рассматриваемого этапа число, а (p^*, p^*_{-1}) изменяется и принимает значения в узлах сетки. Применяя алгоритм для каждого α^*_{-1} и α^*, решим задачу табулирования функции $T(\alpha^*_{-1}, \alpha^*)$.

Если на каком-либо шаге работы алгоритма количество единиц произведенной продукции не достигло k^* , а загрязнение P^*_{-1} устранено, то доля труда, затрачиваемая на борьбу с этим загрязнением, используется в дальнейшем на выпуск основной продукции этапа.

На j -ом шаге алгоритма (j =1,2,...) рассматривается каждый элемент $x_j=(k, p, p_{-1})$ дискретного множества $\tilde{X}$, которое задано в R^3 в виде:

$$\tilde{X} = \left\{ x = (k, p, p_{-1}) \in R^3 : x \in [0, k^*] \times [0, P_{max}] \times [0, P_{-1max}] \right\},$$

где x - узел прямоугольной сетки с числом узлов $N_k, N_p, N_{p_{-1}}$ по осям k, p, p_{-1} соответственно. P_{max} выбирается из эмпирических соображений и соответствует такой величине, что ни в одной точке оптимального процесса уровень загрязнения ее не превосходит. P_{-1max}-это наибольшее из загрязнений в конце предыдущего этапа.

Для каждого элемента $x_j \in \tilde{X}$ определяется, войдет ли он в класс $X^{(j)}$, формируемый на j -ом шаге алгоритма. Для этого проверяется, существует ли элемент $x_{j-1} \in X^{(j-1)}$ и управление $u_{j-1}=(\beta_{j-1}, \beta^1_{j-1}, \Sigma_{j-1}) \in U$, переводящее x_{j-1} в x_j :

$$x_j = x_{j-1} + \Delta t \cdot f(x_{j-1}, u_{j-1}),$$

где f - вектор правых частей системы дифференциальных связей. Иначе говоря, проверяется, принадлежит ли множеству U вектор $u_{j-1}=(\beta_{j-1}, \beta^1_{j-1}, \Sigma_{j-1})$, определяемый из соотношений:

Σ - корень уравнения

$$\frac{k_j - k_{j-1}}{b \Delta t} h(\Sigma) - \Sigma d + \frac{k_j - k_{j-1}}{b \Delta t}\left[j + \frac{a^1 P_{-1,j-1}}{k^*_{-1}} + \frac{a P_{j-1}}{k_{j-1}} \right] -$$

$$- \frac{1}{\Delta t b d^1}\left[(P_{-1,j} - P_{-1,j-1}) d \cdot b - (k_j - k_{j-1}) d d^1 + (P_j - P_{j-1}) d^1 \cdot b \right];$$

$$\beta = - \frac{P_{-1,j} - P_{-1,j-1}}{\Delta t \Sigma d^1};$$

$$\beta = 1 + \frac{P_{-1,j} - P_{-1,j-1}}{\Delta t \Sigma d'} - \frac{K_j - K_{j-1}}{\Sigma b \Delta t}.$$

Если ни один $x_{j-1} \in X^{(j-1)}$ не переводится допустимым управлением в состояние x_j, то элемент x_j не включается в $X^{(j)}$, а если найдется хоть один, то включается.

Трудоемкость метода в целом равна трудоемкости вычисления таблиц функций $\tau_i = \tau_i(\alpha^*_{i-1}, \alpha^*_i)$, $i = \overline{1,n}$.
Для некоторого конкретного этапа, опуская индекс i, имеем

$$\alpha_{-1} \in [0, \alpha^{max}_{-1}] = A_{-1}, \ \alpha \in [0, \alpha^{max}] = A.$$

Для каждого $\alpha^*_{-1} \in A^{(N)}_{-1}$ применяется алгоритм построения областей достижимости, и для каждого $\alpha^* \in A^{(N)}$ фиксируется время $\tau(\alpha^*_{-1}, \alpha^*)$ достижения точки $(K^*, p^*, \alpha^*, \alpha^o_{-1})$, где $\alpha^o_{-1} = 0$.

В результате однократного применения алгоритма функция $\tau(\alpha^*_{-1}, \alpha^*)$ вычисляется при фиксированном α^*_{-1} и для всех $\alpha^* \in A^{(N)}$. В результате N - кратного применения алгоритма вычисляется таблица значений функции $\tau(\alpha^*_{-1}, \alpha^*)$ при всех $\alpha^*_{-1} \in A^{(N)}_{-1}$, $\alpha^* \in A^{(N)}$.

Если число этапов проектирования равно n, то трудоемкость метода составляет nN применений алгоритма, построения области достижимости.

В процессе работы алгоритма решения основной задачи можно либо пользоваться найденными таблицами, либо вычислять функции $\tau_i(\alpha^*_{i-1}, \alpha^*_i)$, $i = \overline{1,n}$ линейной интерполяцией по их табличным значениям.

Рассматриваемая модель многоэтапного процесса проектирования позволяет оценить влияние отдельных этапов и факторов на процессе проектирования в целом, оценить эффективность разрабатываемых средств автоматизации проектирования и экспериментальных исследований, выявить первоочередные задачи и этапы, автоматизация которых приводит к получению наибольшего эффекта.

Уместно заметить, что некоторые задачи борьбы с различными видами загрязнений уже ставились и решались зарубежными учеными, например в [I]. Алгоритм ,предлагаемый для построения множеств достижимости, во многом похож на известный алгоритм В.С. Михалевича "Киевский веник", описанный в [2].

Новизна работы состоит в том, что задача синтеза многоэтапного процесса проектирования сложных технических комплексов описывается в рамках простой и вместе с тем достаточно адекватной модели - задачи оптимального быстродействия, которая для многоэтапного процесса в целом решается методом динамического программирования [3] .

Л и т е р а т у р а

I. Э. К и л е р, М. С п е н с, Р. З е к х а у д е р. Оптимальный контроль над загрязнением окружающей среды. Математическая экономика. М.:Мир,I974.

2. М и х а л е в и ч В.С. Последовательные алгоритмы оптимизации и их применение.Кибернетика, №№I,2,I965.

3. Р. Б е л л м а н. Динамическое программирование.ИЛ,I960.

УДК 629.7.539.3

А.И. Данилин , В.А.Комаров

ПРИМЕНЕНИЕ АЛГОРИТМА ПРОЕКТИРОВАНИЯ С УЧЕТОМ ТРЕБОВАНИЙ ЖЕСТКОСТИ

В данной работе рассматривается одна конкретная задача отыскания рационального распределения материала по элементам конструкции,обеспечивающего одновременное выполнение требований прочности и жесткости. При этом используется автоматизированный метод проектирования, несущий в себе следующие основные идеи.

Требования жесткости рассматриваются в виде ограничений на упругие обобщенные перемещения конструкции под нагрузкой.

Выявлено, что если перемещение определять по формуле Максвелла--Мора, прикладывая единичную обобщенную силу в направлении нежелательных деформаций, то отрицательные значения интегралов Мора для каких-либо элементов конструкции свидетельствуют о том, что для достижения заданных деформаций необходимо увеличить податливость этих элементов. Другими словами, в зонах с отрицательными интегралами Мора нужно назначать толщины, минимально допустимые по прочности, технологичности или другим условиям.

В [I,2] решена задача оптимального распределения материала только по условиям жесткости и без учета зависимости усилий,действующих в элементах конструкции от распределения материала. При этом