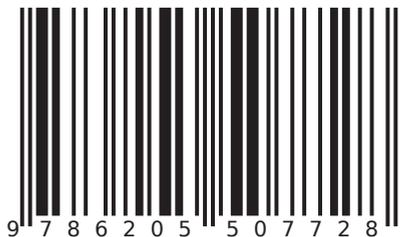


В монографии излагаются результаты научного исследования по алгоритмизации решения оптимизационных задач. Приведены теоретические предпосылки и принципы построения систем автоматизированного проектирования для решения классов оптимизационных задач инженерных конструкций. Совершенствование методов решения оптимизационных задач предполагает алгоритмизацию оптимизационных моделей и алгоритмов для построения автоматизированной системы проектирования инженерных конструкций на основе системного подхода и алгоритмических методов. Приводятся результаты расчета по оптимизации тонкостенных элементов инженерных конструкций и сооружений. Монография предназначена для ученых, докторантов, магистрантов и студентов.

Якубов С.Х. - д.т.н., профессор кафедры «Общетехнические дисциплины» Высшего Военного Авиационного училища Р. Узбекистан. Автор 200 работ.
 Хамраев А.А.- начальник Высшего Военного Авиационного училища Р. Узбекистан. Автор 5 работ.
 Хушбоков И.У.- преподаватель кафедры «Цифровые инф. технологии» Денауского института предпринимательства. Автор 10 работ.



Сабир Халмурадович Якубов
 Абдулла Адхамович Хамраев
 Исмоил Уролмахаматович Хушбоков

Алгоритмизация оптимизационных моделей и алгоритмов

для автоматизированного проектирования
 инженерных конструкций



**Сабир Халмурадович Якубов
Абдулла Адхамович Хамраев
Исмоил Уролмахаматович Хушбоков**

Алгоритмизация оптимизационных моделей и алгоритмов

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

Сабир Халмурадович Якубов
Абдулла Адхамович Хамраев
Исмоил Уролмахаматович Хушбоков

Алгоритмизация оптимизационных моделей и алгоритмов

для автоматизированного проектирования
инженерных конструкций

LAP LAMBERT Academic Publishing RU

Imprint

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:

LAP LAMBERT Academic Publishing

is a trademark of

Dodo Books Indian Ocean Ltd. and OmniScriptum S.R.L Publishing group

Str. Armeneasca 28/1, office 1, Chisinau-2012, Republic of Moldova, Europe

Printed at: see last page

ISBN: 978-620-5-50772-8

Copyright © Сабир Халмурадович Якубов,
Абдулла Адхамович Хамраев, Исмоил Уролмахаматович Хушбоков
Copyright © 2022 Dodo Books Indian Ocean Ltd. and OmniScriptum S.R.L
Publishing group

FOR AUTHOR USE ONLY

С.Х. ЯКУБОВ, А.А. ХАМРАЕВ, И.У. ХУШБОКОВ

**АЛГОРИТМИЗАЦИЯ ОПТИМИЗАЦИОННЫХ МОДЕЛЕЙ И АЛГОРИТМОВ ДЛЯ
АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ИНЖЕНЕРНЫХ
КОНСТРУКЦИЙ**

FOR AUTHOR USE ONLY

2022

В монографии излагаются результаты научного исследования по алгоритмизация решения оптимизационных задач. Приведены теоретические предпосылки и принципы построения систем автоматизированного проектирования для решения классов оптимизационных задач инженерных конструкций. Совершенствование методов решения оптимизационных задач предполагается алгоритмизации оптимизационных моделей и алгоритмов для построения автоматизированной системы проектирования инженерных конструкций на основе системного подхода и алгоритмических методов. Приводятся результаты расчета по оптимизации тонкостенных элементов инженерных конструкций и сооружений.

Монографии предназначено для ученых, докторантов, магистрантов и студентов.

FOR AUTHOR USE ONLY

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
ГЛАВА 1. МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ И ИХ АЛГОРИТМИЗАЦИИ.....	9
1.1. Анализ методов и алгоритмов решения оптимизационных задач.....	9
1.1.1. Минимизация функций без ограничений.....	9
1.1.2. Задачи с ограничениями.....	12
1.1.3. Алгоритмы случайного поиска.....	14
1.2. Прикладные программы задач оптимизации.....	19
1.3. Оптимальное проектирование инженерных конструкций.....	21
1.4. Принципы разработки алгоритмических банков для оптимизационных задач.....	24
1.5. Принцип определения оптимальной последовательности алгоритмов для классов оптимизационных задач.....	28
1.6. Постановка и анализ проблемы создания алгоритмической системы решения классов оптимизационных задач.....	31
ГЛАВА 2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ.....	35
2.1. Понятия оптимального и рационального решений.....	35
2.2. Расчетной модели деформируемых систем.....	35
2.3. Постановка задач параметрической оптимизации.....	38
2.4. Схема получения оптимального решения.....	38
2.5. Математическая модель оптимизации пластинчатых конструкций со сложной конфигурацией.....	39
2.6. Математические модели оптимизации арок, цилиндрических и конических оболочек переменной толщины и алгоритмы их разрешения.....	44
2.6.1. Уравнения равновесия арок, цилиндрических и осесимметричных конических оболочек переменной толщины.....	44
2.6.2. Алгоритмы решения прямой задачи расчета арок, цилиндрических и конических оболочек.....	48
ГЛАВА 3. РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ СЛУЧАЙНОГО ПОИСКА ДЛЯ АЛГОРИТМИЧЕСКОЙ СИСТЕМЫ ОПТИМИЗАЦИИ (АСО).....	52
3.1. Формирование основных требований к алгоритмам АСО.....	52
3.2. Алгоритм поиска П-1 и П-2.....	54
3.2.1. Алгоритм поиска П-1.....	54
3.2.2. Алгоритм поиска П-2.....	59
3.3. Алгоритмы глобального поиска.....	62
3.3.1. Алгоритм глобального поиска ГП-3.....	62
3.3.2. Алгоритм специального случайного поиска для оптимизации конструкций.....	68
ГЛАВА 4. РАЗРАБОТКА АЛГОРИТМИЧЕСКОЙ СИСТЕМЫ ОПТИМИЗАЦИИ.....	71
4.1. Банк постановки задач.....	71
4.2. Банк данных.....	77
4.3. Построение банков признаков и моделей.....	79
4.3.1. Б3 – банк признаков.....	79
4.3.2. Б4 – банк моделей.....	83
4.4. Построение банка алгоритмов и вычислительного банка.....	86
4.4.1. Б5 - банк алгоритмов.....	86
4.4.2. Б6 – вычислительный банк.....	91
ГЛАВА 5. РЕЗУЛЬТАТЫ РАСЧЕТА ОПТИМИЗАЦИИ ТОНКОСТЕННЫХ ЭЛЕМЕНТОВ ИНЖЕНЕРНЫХ КОНСТРУКЦИЙ.....	94

5.1. Весовая оптимизация инженерных конструкций типа пластин и оболочек.....	94
5.2. Оптимизация замкнутых цилиндрических оболочек, нагруженных осевой сжимающей силой.....	97
5.3. Оптимизация осесимметричных усеченных конических оболочек и кольцевых пластин переменной толщины.....	101
5.4. Оптимизация упругих пластин сложной конфигурации.....	106
ЗАКЛЮЧЕНИЕ.....	110
ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА.....	112

FOR AUTHOR USE ONLY

ВВЕДЕНИЕ

Оптимизации и проектирования инженерных конструкций является одна из наиболее сложных и актуальных задач механики. При оптимизации конструкций за целевую функцию принимают вес конструкции, стоимость, частоту колебаний и т.д. Наиболее широко ставится задача проектирования конструкций минимального веса, которые находят широкое применение в таких отраслях народного хозяйства как строительство, ракетостроение, самолетостроение, кораблестроение и др. Решение ряд практически важных задач по расчету и оптимизации оболочечных и пластинчатых конструкций, позволяет получить значительный народно хозяйственный эффект.

При постановке оптимизационных задач в области проектирования инженерных конструкций типа пластин и оболочек, можно встретить двойное истолкование (понимание) системного анализа (подхода): с одной стороны – это анализ любой реально существующей системы; с другой – формирование параметров системы для достижения поставленных целей. В реальных условиях эти две стороны неразрывны, поскольку нельзя создать систему, обеспечивающую поставленные цели, без анализа содержания и определения реальных процессов, которые приведут к желаемому результату. Системный анализ обеспечивает условия совместной оптимизации, как структурных частей системы (ее подсистемы), так и системы в целом, а также программного обеспечения компьютера. Конечной целью использования системного анализа при проектировании является реальное проектирование системы, ее подсистем и компонентов для достижения оптимальных эффективности и экономичности. Несмотря на то, что в системном анализе отсутствуют строго определенные правила, основные признаки достаточно полно раскрыты в [120].

С учетом специфики процесса проектирования оболочечных и пластинчатых конструкций и решаемых задач основные особенности системного подхода могут быть отображены следующими положениями [157]. Во-первых, в качестве оптимизируемой проектируемой оболочечных и пластинчатых конструкции принимается определенный, соответствующий выполнению функций, комплекс элементов, разделенный заданными свойствами и обладающий абстрактными связями с внешними условиями и системами.

В этом комплексе в процессе исследований каждому элементу можно придавать желаемые свойства без учета реальных характеристик, чтобы выявить возможный вклад этих свойств в изучаемые процессы и, следовательно, обосновать требования к перспективному решению данного элемента. В практических задачах оптимизации принимается, что свойства элементов и их функционально-технические характеристики известных и поэтому процессы функционирования рассматриваются в области допустимых (с учетом принятых ограничений) решений систем. Как в первом, так и во втором, а также в случае программного обеспечения (разработка алгоритмических комплексов) оценка рассматриваемого комплекса производится с учетом совокупности известных процессов и явлений и взаимосвязи между ними. Все это выдвигает на первый план такие особенности модели проектируемых инженерных конструкций, которые способствуют выяснению механизма функционирования данного комплекса в целях выбора наименьшего веса или затрат. При этом, следует отметить, во всех случаях система включает понятие о целом, состоящем из взаимосвязанных, взаимодействующих и взаимозависимых частей. При этом свойства этих частей зависят от системы в целом, а свойства системы – от свойств её частей.

Во-вторых, для конкретной проектируемой инженерной конструкции типа оболочек и пластин должно быть определено место в общей структуре других систем. Системный подход требует обоснованного выделения исследуемой системы в общем составе систем, предназначенных для поддержания нормирующих параметров, разделения её на подсистемы. Инженерные конструкции составными или основными элементами, которых являются оболочки или пластин рассматриваются как самостоятельный объект

изучения и оптимизации, но с учетом необходимого обмена информацией со смежными и внешними системами и внутри её – между подсистемами. Выбранная общая структура систем должна четко очертить границы исследуемой системы и способствовать выделению (структуризации) таких её подсистем, которые по своим размерам доступны для исследований и однородны по описанию. Все это обеспечивает организацию связей на каждом очередном уровне спуска от системы к отдельным элементам сверху-вниз с последующей передачей полученной агрегированной информации наверх (снизу-вверх). При этом, как общей структуре систем компенсации, так и подсистемам конструкций и сооружений должны быть присущи свойства целостности: изменения, возникшие в какой-либо из их частей, сказываются как на других частях, так и на всей их совокупности.

В-третьих, инженерная конструкция или сооружения представляются в виде модели. При проектировании сложных систем, какими являются инженерные конструкции типа пластин и оболочек, требуются знания о количественных и качественных закономерностях поведения системы и отдельных её элементов в зависимости от характера изменения многочисленных факторов (параметров).

Модель должна быть сходна с оригиналом, но и отличаться от него. Её отличительные особенности проявляются в том, что она подвергается таким преобразованиям в нужном направлении, которые невозможны при непосредственном исследовании оригинала.

Математическое моделирование позволяет изучать только те параметры оригинала, которые имеют математическое описание, адекватно отображающее поведение оригинала. При разработке модели очень важно освободиться от связей и отношений, которые затрудняют познание объекта исследований в соответствии с поставленными целями. При этом важно, чтобы ясные в своей основе идеи не обростали тяжелыми и громоздкими подробностями. Выбор модели является центральной частью работы по формированию методологии исследования и зависит от основной идеи, определяющей поиск экстремума функции цели.

Для решения ряда оптимизационных задач могут быть применены известные математические методы поиска экстремума функций нескольких переменных, например, в классической математике это решение системы линейных уравнений, полученных при приравнении нулю частных производных исследуемой функции по оптимизируемым параметрам, и метод неопределенных множителей Лагранжа. Эти методы справедливы при отсутствии ограничений на оптимизируемые параметры или при ограничениях в виде равенства. При ограничениях в виде неравенства используются методы нелинейного математического программирования, подразделяемые по признаку организации процесса поиска по методу слепого и направленного поиска. К первому из них относятся метод сплошного перебора вариантов с их упорядочением по критериям эффективности и метод статистических испытаний (метод Монте-Карло). К методу направленного поиска относятся градиентный, наискорейшего спуска, координатного спуска и т.д. Имеются и другие методы нелинейного программирования. В конечном итоге выбор метода определяется с учетом многих соображений, из которых не последняя роль принадлежит удобству обращения к алгоритму, длительности счета и т.д. Очевидно также, что решение задач требует неформальных действий, возможности вмешиваться в процесс счета и получить промежуточные результаты для реализации диалогового режима.

Известно [120], что выбор модели во многом зависит от интуиции, опыта, навыков неформального мышления, от представлений о сути взаимосвязи входов и выходов для самой системы. При этом прежде всего следует уточнить задачи, решению которых должна способствовать модель. При постановке данной задачи модель должна обеспечивать:

а) возможность обобщения любых исходных условий (факторов) в такую форму расчетной информации, которая существенно облегчает целенаправленный выбор конкурирующих вариантов конструкций и сооружений, состава подсистем и режимов их функционирования (НДС); б) исследование характера взаимосвязи

определяющих параметров систем и подсистем в зависимости от условий функционирования объекта; в) представление определяющих параметров в виде координат состояния системы, использование которых позволяет вычислить любые технико-экономические показатели, как отдельной подсистемы, так и системы в целом.

Модель конструкции может быть детерминированной по условию взаимосвязи входов и выходов для самой системы. Обойтись одной моделью практически не удастся, необходима система моделей - совокупность взаимосвязанных моделей отдельных подсистем. Система моделей должна создавать возможность самостоятельного решения отдельных задач без нарушения их последующего согласования с учетом всех связей между подсистемами.

В-четвертых, для оценки качества решений проектируемых конструкций и сооружений выбирается комплекс показателей. Как правило, цель системного анализа заключается в том, чтобы при всех возможных характеристиках внешних связей добиться наилучшего (оптимального) решения проектируемых конструкций и сооружений по их конструктивным, экономическим и другим показателям. Однако оптимум и оптимальность – не абсолютные понятия, они требуют точного определения критериев оптимальности, т.е. главных признаков, на основании которых производится сравнение эффективности различных решений. Решение, наилучшее в одних условиях и по одному критерию, может оказаться далеко не лучшим в других условиях и по другому критерию. Оптимизация по одному критерию (субоптимизация) чаще всего для технических систем производится по приведенным и затратам (в настоящем исследовании в качестве целевой функции принят вес конструкции).

В пятых, результаты анализа на модели конструкций должны переноситься на реальные системы. Для переноса решений на реальный объект требуется уверенность в адекватности решения. Адекватность оценивается аналогией свойств реального объекта и модели по основным признакам. Адекватность достигается в том случае, если модель полностью отображает напряженно-деформируемое состояние (НДС) реально существующих проектируемых конструкций и сооружений типа пластин и оболочек.

Перечисленные и принятые к исполнению основные положения системного подхода характеризуют лишь исходную основу метода, однако эффективность его использования полностью зависит от избираемого способа их реализации.

Для систематизации и обобщения сведений об основных признаках системного анализа, которые способствуют представлению разрозненных данных в упорядоченном виде с меньшим числом существенного переменных, необходимо: систематизировать взаимосвязь между системами, предназначенными для поддержания нормируемых параметров; проанализировать многочисленные исходные условия, найти форму их обобщения; пригодную для классификации определяющих условий НДС системы; выявить целесообразную классификацию конструкции или сооружения, способствующего целенаправленному выбору их конкурирующих вариантов; определить принципы декомпозиции систем, исходя из анализа их совокупности как единого целого; сформулировать исходную основу для построения математической модели конструкции или сооружения; классифицировать оптимизационные задачи, возникающие в практике исследований и проектирования.

Глава 1. МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ И ИХ АЛГОРИТМИЗАЦИИ

1.1. Анализ методов и алгоритмов решения оптимизационных задач

Рассмотрим вкратце характеристики и особенности основных методов решения задачи математического программирования.

Методы линейного программирования [13,45,74,142] решает задачу математического программирования в случае линейных F и f_i . В этом случае для решения всех задач линейного программирования можно применить симплекс – метод.

В этом случае, когда целевая функция F или ограничения f_i нелинейны, встает задача нелинейного программирования. Для решения этой задачи разработано множество методов с различной степенью сложности и скоростью сходимости. Однако, неперенным условием успешной работы этих методов является требование выпуклости целевой функции и области поиска.

Различает два типа задач нелинейного программирования:

- 1) задачи минимизации целевой функции без ограничений (безусловная оптимизация);
- 2) задачи при наличии ограничений (условная оптимизация).

Наименование полно разработаны методы условной оптимизации, что связано с резким усложнением задач и соответствующих алгоритмов при наличии ограничений.

Методам нелинейного программирования посвящена довольно обширная литературных источника [9,11,45,74,135,141]. Поэтому мы вкратце остановимся на наиболее распространенных методах, их положительных и отрицательных сторонах.

1.1.1. Минимизации функций без ограничений. Эти методы предполагают решение задачи минимизации функции $F(X)$, определенный в n -мерном евклидовом пространстве E^n , т.е. $X(x_1, x_2, \dots, x_n)$ – n -мерный вектор, при помощи итерационного процесса типа $X_{k+1} = X_k + \alpha_k \cdot P_k$, где P_k – вектор, определяющий направление движения из точки X_k ; $\alpha_k = \text{const}$ – определяет длину шага в направлении P_k .

Способом определения P_k и α_k и отличаются приведенные ниже методы.

Метод наискорейшего спуска. Направление спуска является антиградиентным в точке X_k :

$$X_{k+1} = X_k - \alpha_k \cdot F'(X_k), \alpha_k > 0, k = 0, 1, \dots \quad (1.1)$$

В координатной форме процесс (1.1) записывается так:

$$X_{k+1}^i = X_k^i - \alpha_k \frac{\partial F(X_k)}{\partial X^i}, i = \overline{1, n}.$$

Метод получил широкое распространение вследствие простоты и возможности применения для минимизации широкого класса функций.

Величина α_k выбирается следующим образом:

- 1) выбирается произвольное значение $\alpha > 0$ (одно и то же на всех итерациях) и определяется точка $X = X_k - \alpha F'_k$;
- 2) вычисляется $F(X) = F(X_k - \alpha F'_k)$;
- 3) производится проверка неравенства:

$$F(X) = F(X_k) \leq \varepsilon \alpha (F'_k, -F'_k) \quad (1.2)$$

где $0 < \varepsilon < 1$ – произвольно выбранная константа;

$(F'_k, -F'_k)$ – скалярное произведение векторов.

4) если неравенство (1.2) выполняется, то значение α и берется в качестве искомого $\alpha_k = \alpha$, иначе производится дробление α до выполнения неравенства (1.2).

Множитель α_k можно выбрать так, чтобы процесс сходил к стационарной точке при удовлетворении следующих условий:

- $F(X)$ ограничена снизу;
- $F'(X)$ удовлетворяет условию Липшица.

$$\|F'(X) - F'(Y)\| \leq R \|X - Y\|,$$

где R – некоторая константа.

К точке минимума X^* процесс сходится при выполнении дополнительных условий:

$F(X)$ – дважды непрерывно дифференцируемая;

- матрица вторых производных удовлетворяет условиям:

$$m \|Y\|^2 \leq (F''(X)Y, Y) \leq M \|Y\|^2, M \geq m \geq 0$$

при любых $X, Y \in E^n$.

Для скорости сходимости справедливы оценки:

$$\|F_k - F_*\| \leq q^k \|F_0 - F_*\|; \|X_k - X^*\| \leq cq^{k/2};$$

$$c < \infty; 0 < q < 1.$$

Метод довольно плохо сходится при минимизации овражных функций.

Метод Ньютона с регулировкой шага. Итерация:

$$X_{x+1} = X_x - \alpha_k (F''_k)^{-1} F'_k, \alpha_k > 0 \quad (1.3)$$

В координатной форме процесс (1.3) записывается так:

$$X_{k+1}^i = X_k^i - \alpha_k \sum_{j=1}^n \varphi_{ij} \frac{\partial F(X_k)}{\partial X^j}; \quad i = \overline{1, n}; j = \overline{1, n},$$

где φ_{ij} – элементы матрицы $(F''_k)^{-1}$;

j – индекс строки.

Метод можно представить в виде

$$F''_k P_k = -F'_k; X_{x+1} = X_k + \alpha_k P_k,$$

или в координатной форме:

$$\sum_{j=1}^n \frac{\partial^2 F(X_k)}{\partial x^i \partial x^j} P_k^j = -\frac{\partial F(X_k)}{\partial x^i};$$
$$x_{k+1}^i = x_k^i + \alpha_k p_k^i, \quad i = \overline{1, n}.$$

Следовательно, для определения вектора P_k можно решать систему линейных уравнений вместо того, чтобы обращать матрицу $F''(X_k)$. Из вышеизложенного видно,

что метод Ньютона может применяться лишь для минимизации тех функций, у которых существует обратимая ограниченная матрица вторых производных $F''(X)$. Скорость сходимости метода выше чем у градиентных методов, но затраты времени на один шаг гораздо больше.

В практике применяются некоторые модификации метода Ньютона:

а) $X_{k+1} = X_k - \alpha_k (F_0'')^{-1} F_k'$; $\alpha_k > 0$ т.е. для спуска применяется одна и та же матрица $(F_0'')^{-1}$, но при этом скорость сходимости ниже, чем в основном методе.

б) $X_{k+1} = X_k - \alpha_k (F_k'')^{-1} \cdot F_k'$; $\alpha_k > 0$.

Здесь матрица $(F_k'')^{-1}$ обновляется через каждые t шагов. Способ выбора α_k аналогичен спуску.

Метод двойственных направлений. Итерация $X_{k+1} = X_k - \alpha_k \cdot A_k^{-1} F_k'$; $\alpha_k > 0$, где A_k – определяется с помощью системы уравнений

$$A_k (X_{k-i} - X_{k-i-1}) = F'(X_{k-i}) - F'(X_{k-i-1}); \quad i = \overline{0, n-1}.$$

Для обеспечения сходимости точки X_{k-i} , X_{k-i-1} следует выбирать достаточно близкими друг другу.

Этот метод относится к так называемым «квазиньютоновским» методам. При его помощи можно минимизировать строго выпуклые гладкие функции независимо от выбора начального приближения. Скорость сходимости у этого метода сверхлинейная и близка к методу Ньютона при меньших затратах на один шаг. Отрицательной стороной метода является требование большой памяти ЭВМ.

Метод сопряженных направлений (сопряженных градиентов). Наиболее широко этот метод применяется для минимизации квадратичных функций:

$$F(X) = \frac{1}{2}(AX, X) + (b, X) + C,$$

где $AX > 0$ при любом $X \neq 0$.

Минимизация сводится к обращению матрицы A . Если известна матрица A^{-1} , то решение находится по формуле:

$$X^* = X_0 - A^{-1} F_0' = -A^{-1} b,$$

где X_0 – произвольная точка.

Итерация метода сопряженных направлений:

$$X_{k+1} = X_k + \alpha_k p_k; \quad K = \overline{0, n-1},$$

где

$$\alpha_k = - \frac{(F_0', P_k)}{(AP_k, P_k)}.$$

P_i – сопряженные вектора, удовлетворяющие условию:

$$(P_i, AP_j) = 0; \quad i \neq j.$$

Векторы P_k строятся следующим образом:

$$\begin{aligned}
 P_0 &= -F'(X_0); \\
 P_{k+1} &= -F'(X_{k+1}) + \beta_{k+1}P_k; \\
 \beta_{k+1} &= -\frac{[F'(X_{k+1}), F'(X_{k+1}) - F'(X_k)]}{(F'(X_k), P_k)}; k = 0, 1, \dots
 \end{aligned}$$

Этот метод определяет минимум квадратичной функции за число шагов, не превышающее n . По своим характеристикам он лучше градиентного метода и при минимизации других функций.

1.1.2. Задачи с ограничениями

1. Задача квадратичного программирования. Минимизируется квадратичная функция:

$$F(X) = \frac{1}{2}(X, CX) + (d, X) \quad (1.4)$$

при ограничениях:

$$(a_i, X) - b_i \leq 0; \quad i \in J^-; \quad (1.5)$$

$$(a_i, X) - b_i = 0; \quad i \in J^0. \quad (1.6)$$

Здесь $X \in E^n$; $a_i \in E^n$; $d \in E^n$; $i \in J^- \cup J^0$,
 b_i – числа, C – симметричная положительно определенная матрица размерности $n \times n$, т.е. $(X, CX) \geq 0$ для всех X . J^- и J^0 – конечные множества индексов.

Минимизация (1.4) при ограничениях (1.5, 1.6) производится процессом [38].

$$\begin{aligned}
 P_{-1} &= -(I - P)F'(X_0); \\
 X_{k+1} &= X_k + \alpha_{k+1}P_{k+1}; \\
 P_{k+1} &= -(I - P)F'(X_k) + \frac{\|(I - P)F'(X_k)\|^2}{\|(I - P)F'(X_{k-1})\|^2} P_k; \\
 \alpha_{k+1} &= -\frac{(F'(X_k), P_{k+1})}{(P_{k+1}, CP_{k+1})}; \quad k = 0, 1, \dots
 \end{aligned}$$

где X_0 – точка, удовлетворяющая ограничениям (1.5, 1.6); I – единичная матрица;

$P = A^T(A, A^T)^{-1}A$, где A – матрица, в которой строками служат вектора Q_i .

A^T – транспонированная матрица A .

Этот алгоритм можно трансформировать и для решения общей задачи математического программирования

2. Метод возможных направлений. Этот метод предназначен для решения задачи выпуклого программирования:

$$\begin{aligned}
 F(X) &\rightarrow \min; \\
 f_i(X) &\leq 0, \quad i = 1, 2, \dots, m; \\
 Ax - b &= 0;
 \end{aligned}$$

где $f_i(x)$ – выпуклые, непрерывно дифференцируемые функции;
 A – матрица, размерностью $l \times m$;
 b – мерный вектор.

Для успешной работы методы требуется, чтобы градиенты функции удовлетворяли условию Липшицы:

$$\|f'_i(X_1) - f'_i(X_2)\| \leq C \|X_1 - X_2\|$$

и, кроме того

$$f'_i(X) \leq K \text{ для всех } X,$$

где K – какая-либо const.

Основную идею метода можно выразить следующим образом. Каким-либо способом находят $X_0 \in D$ (D – допускаемая область), определяют направление $P \in E^n$ такое, что при малых α выполняются условия:

$$(X_0 + \alpha p) \in D \text{ и } F(X_0 + \alpha p) < F(X_0)$$

такое направление называется возможным. Сдвинувшись вдоль направления на шаг α , получили следующую точку $X_1 = X_0 + \alpha_1 p_1$ и т.д. Для определения возможных направлений решаются вспомогательные задачи, которые могут быть сведены к задачам линейного программирования. Подробно описывается метод с необходимыми доказательствами в [32].

3. Метод условного градиента. Предназначен для решения задачи выпуклого программирования. Дадим его краткое описание.

Определяется каким-либо образом $X_k \in D$. Вычисляют $F'(X_k)$ и находят точку минимума линейной функции $(F'(X_k), Z)$ в области D . Пусть это будет точка $Z(X_k)$. Полагают $P_k = Z(X_k) - X_k$ и $X_{k+1} = X_k + \alpha_k P_k, \alpha_k > 0$. Точка X_{k+1} принимается за исходную и т.д.

Как видим, метод может применяться для минимизации нелинейной функции в области, в которой задача минимизации линейной функции решается легко.

4. Метод штрафной функции. Это наиболее широко распространенный метод решения задач с ограничениями. Основная идея этого метода состоит в том, что задача нахождения условного минимума целевой функции заменяется задачей определения безусловного минимума некоторой вспомогательной функции, которая не отличается от исходной целевой функции $F(X)$ внутри допустимой области D и резко возрастает вне этой области. Пусть требуется минимизировать $F(X) \in E^n$ при ограничениях

$$f_i(X) \leq 0; \quad i = \overline{1, m}.$$

Все функции f_i должны быть непрерывны.

Положим

$$\varphi(t) = \begin{cases} t^2, & t \geq 0, \\ 0, & t < 0. \end{cases}$$

Составим функцию:

$$\psi(X, r) = r \sum_{i=1}^m \varphi(f_i(X)).$$

Очевидно, что

$$\begin{aligned} \psi(X, r) &= 0 \text{ при } X \in D, \\ \psi(X, r) &> 0 \text{ при } X \notin D. \end{aligned}$$

Вспомогательная задача состоит в минимизации функции

$$F_0(X, r) = F(X) + \psi(X, r).$$

Естественно, что решение вспомогательной задачи будет приближенным, однако степень приближения можно изменять, варьируя значением коэффициента r .

Далее мы не будем перечислять другие методы нелинейного программирования, так как основные идеи достаточно полно видны при рассмотрении вышеприведенных методов.

Все детерминированные методы решения задач нелинейного программирования требуют непрерывности целевой функции и ограничений, наличие первых, вторых производных, выпуклости, удовлетворение условию Липшица и т.д.

Все эти требования резко сужают круг задач, для решения которых могут эффективно применяться эти методы. Такие задачи, как многоэкстремальные, с многосвязной областью поиска D , при наличии фактора неопределенности и др. не могут решаться подобными методами. Более широкие возможности имеют статистические методы поиска. О них речь пойдет ниже.

1.1.3. Алгоритмы случайного поиска

Поставленную выше задачу оптимизации можно решить при помощи различных алгоритмов случайного поиска. Применение этого или иного алгоритма к решению задачи математического программирования обусловлено особенностями конкретной задачи и возможности алгоритма [9,31,39,44,45,48,55,56,57,68,74,89,90,111-116,123-125,128,143,145].

Л.А. Растринг в работах [90,123-125] объекты оптимизации классифицирует по трем признакам:

- а) дискретность или непрерывность;
- б) число экстремумов;
- в) наличие помех.

Дискретным называется такой объект, определение показателя качества занимает много времени.

Непрерывный объект дает возможность определить значение показателя качества практически без затрат.

В связи с этим, процесс поиска может быть шаговым или непрерывным в зависимости от объекта.

Непрерывные алгоритмы записываются в виде дифференциального уравнения:

$$\frac{dX}{dt} = Y[X, F(X)], \quad (1.7)$$

где $X(x_1, x_2, \dots, x_n)$ – вектор оптимизируемых параметров;

$F(X)$ – целевая функция;

Y – непрерывная векторная функция алгоритма поиска.

Шаговый алгоритм записывается в конечноразностной форме, т.е. в виде выражения для рабочего смещения в пространстве оптимизируемых параметров:

$$\Delta X = \Phi[X, F(X)], \quad (1.8)$$

где под Φ – подразумевается алгоритм определения рабочего шага.

Шаговые алгоритмы более универсальны, т.к. при помощи их можно оптимизировать как дискретные, так и непрерывные объекты, тогда как непрерывные алгоритмы могут работать только с непрерывными объектами.

Далее алгоритмы случайного поиска делятся в зависимости от количества экстремума на алгоритм локального поиска (одно экстремальные объекты) и глобального поиска (многоэкстремальные объекты).

Кроме того, алгоритмы делятся на алгоритмы с самообучением и без него. В последнем случае поиск подразделяется на локальный и не локальный. Локальным называют такую процедуру поиска, при которой поведение не зависит от предыстории. Нелокальный поиск связывает соседние этапы.

Рассмотрим некоторые шаговые алгоритмы локального поиска (одноэкстремальные объекты) [48,49,111-117].

1. Алгоритм с парной пробой. В случайном направлении определяем случайным единичным вектором $\Xi = (\xi_1, \xi_2, \dots, \xi_n)$ по обе стороны от исходного состояния $X = (x_1, x_2, \dots, x_n)$ делаются пробы. Значение целевой функции в точках $X_{1,2} = X \pm g\Xi$, где g – величина пробного шага, и определяют направление рабочего шага. Этот шаг делается в направлении наименьшего значения $F(X)$:

$$\Delta X_{k+1} = a\Xi \operatorname{sign}[F(X_k - g\Xi) - F(X_k + g\Xi)] \quad (1.9)$$

где a – величина рабочего шага.

Этот алгоритм склонен к «блужданию» даже в том случае, если минимум найден, т.к. найдя экстремум, алгоритм тут же уводит систему в сторону.

Этот алгоритм локальный, т.к. приращения ΔX не зависят от предыдущих шагов.

2. Алгоритм с возвратом при неудачном шаге. Этот алгоритм делает шаг в случайном направлении Ξ . При неудаче, т.е. при $F(X_{k+1}) \geq F(X_k)$ система возвращается в первоначальное состояние, откуда делается новый шаг.

Этот алгоритм можно записать в виде следующего выражения для шага:

$$\Delta X_{k+1} = \begin{cases} a\Xi & \text{при } F(X_k) < F(X_{k-1}); \\ -\Delta X_k & \text{при } F(X_k) \geq F(X_{k-1}). \end{cases} \quad (1.10)$$

Этот алгоритм не делает шагов в неудачном направлении, однако скорость сходимости его к окрестности минимума не велика.

3. Алгоритм с пересчетом при неудачном шаге.

Является модификацией алгоритма 2. Возврат в исходное состояние при неудачной попытке не делается, а система снова делает случайный шаг, отсчитанный из старого состояния, т.е. возврат как бы пересчитывается вместе с последующим случайным шагом, в связи, с чем алгоритм имеет более высокое быстродействие. Выражение для смещения имеет вид:

$$\Delta X_{k+1} = \begin{cases} a\Xi & \text{при } F(X_k) < F_{k-1}^0; \\ -\Delta X_k a\Xi & \text{при } F(X_k) \geq F_{k-1}^0, \end{cases} \quad (1.11)$$

где $F_k^0 = \min_{j=1,2,\dots,k} F(x_j)$ – наименьшее значение целевой функции за K предыдущих шагов поиска.

4. Алгоритм наилучшей пробы. Из исходной точки X_k делается m случайных независимых проб $X_k + g\Xi_j$; ($j = \overline{1, m}$) и запоминается тот шаг, который привел к наибольшему снижению целевой функции и в этом направлении делается рабочий шаг:

$$\Delta X_{k+1} = a\Xi^*, \quad (1.12)$$

где Ξ^* – направление наилучшей пробы, которая удовлетворяет следующему выражению:

$$F(X_k + g\Xi^*) = \min F(X_k + g\Xi_j), \quad j = \overline{1, m}, \quad (1.13)$$

где g – величина пробного шага, Ξ_j – единичные случайные независимые векторы, равномерно распределенные по всем направлениям пространства параметров. Этот алгоритм имеет большую скорость сходимости к экстремуму. Особенно ощутимо его превосходство при нелинейных функциях.

Выше описанных алгоритмов вполне достаточно для определения идеи и основных методов, которые применяются при случайном поиске. Существует еще несколько разновидностей алгоритмов локального поиска, которые являются различными модификациями выше описанных алгоритмов и не вносят ничего принципиально нового в случайный поиск.

Как уже отмечалось выше в алгоритмах случайного поиска возможно применение самообучения. Под самообучением понимается целенаправленный процесс изменения поведения поиска, т.е. такую деформацию вероятных свойств случайности, целью которой является повышение быстрействия, точности, надежности и других характеристик поиска.

Рассмотрим некоторые *виды самообучения* [56,57,133,165].

1. Самообучение методом исключения. Этот алгоритм опирается на то очевидное положение, что если число возможных направлений системы конечно, то исключение из рассмотрения неудачных направлений повышает вероятность отыскания удачных направлений.

Недостатком этого алгоритма является необходимость иметь довольно громоздкую память для хранения неудачных направлений.

2. Алгоритм координационного самообучения с произвольным законом изменения вероятности. Вероятность выбора положительного шага вдоль i – той координаты является определенной функцией некоторого параметра $W_i^{(N)}$, которой называется «памятью» по i -той координате:

$$P_i^{(N)} = P(W_i^{(N)}). \quad (1.14)$$

Вид функции (1.4) может быть различным, но всегда должен быть монотонным и неубывающим.

Алгоритм обучения реализуется путем изменения параметра памяти. Один из возможных вариантов изменения памяти [66]:

$$W_i^{(N+1)} = W_i^{(N)} - \delta \cdot \text{sign}(\Delta x_i^{(N)} \cdot \Delta F_N) \quad (1.15)$$

где δ «шаг по памяти» ($\delta > 0$) – величина, определяющая скорость обучения. При $\delta = 0$ обучения нет. При неудачном шаге память изменяются таким образом, что вероятность выбора этого направления при последующем шаге уменьшаются.

При работе алгоритмов поиска с самообучением типа (1.15) существует опасность «передетерминирования» поиска, т.е. непомерного увеличения параметра памяти w_i , в результате чего поиск практически производиться не будет. Чтобы избежать этого, память ограничивают сверху. Иногда память ограничивают и снизу, т.е. $c_1 \leq w \leq c_2$. Тогда выражение для определения памяти примут вид:

$$w_i = \begin{cases} c_1 & \text{при } w_i < c_1; \\ w_i & \text{при } c_1 \leq w_i \leq c_2; \\ c_2 & \text{при } w_i > c_2. \end{cases} \quad (1.16)$$

3. Алгоритм самообучения с направляющей сферой. Этот алгоритм является алгоритмом непрерывного самообучения[50]. Работает он следующим образом. Случайные точки выбираются на поверхности n -мерной гиперболы, а сама эта гиперсфера несколько выдвинута в направлении вектора памяти w . Таким образом, найденные направления имеют тенденцию в сторону w , причем эта тенденция тем больше, чем больше выдвинута гиперсфера в направлении w .

Направление пробного шага в данном алгоритме определяется следующей формулой:

$$\Xi = \frac{w + c\Xi^0}{|w + c\Xi^0|}, \quad (1.17)$$

где Ξ^0 – случайный единичный вектор, равномерно распределенный по всем направлениям пространства параметров;

c – некоторая постоянная (радиус гиперболы).

4. Алгоритм с направляющим конусом. В пространстве параметров образуют гиперконус, ось которого совпадает с направлением вектора памяти, имеющего при вершине угол 2ψ . Вокруг вершины конуса, как относительно центра, проводят гиперсферу радиуса g . Конус отсчит от этой сферы часть поверхности, на которой случайно выбирают точки, определяющие дальнейшее направление поиска.

Жесткость обучения в этом случае зависит от угла раскрытия конуса 2ψ . При малом угле имеется возможность «передетерминирования» поиска.

5. Непрерывный алгоритм самообучения с направляющим гиперэллипсоидом. Строится эллипсоид, ориентированный вдоль вектора памяти w , полуоси гиперэллипсоида определяются по следующим выражениям[50]:

$$\left. \begin{aligned} b &= \frac{1}{2} \sqrt{\frac{4 - |w|^2}{n}}; \\ b_1 &= \sqrt{1 - (n-1)b^2} \end{aligned} \right\} \quad (1.18)$$

b_1 – длина полуоси гиперэллипсоида, направленная вдоль памяти w .

$b_2 = b_3 = \dots = b_n = b$ – остальные полуоси.

Направление поиска ищется внутри указанного гиперэллипсоида.

Выше перечисленные алгоритмы самообучения являются наиболее распространенными в существующих алгоритмах случайного поиска.

Наименее полно в общей теории случайного поиска разработана теория алгоритмов глобального поиска. Это связано с принципиальными трудностями, возникающими при попытке отыскать глобальный экстремум. В этом случае возникают такие проблемы как:

- а) обеспечение глобальности поиска;
- б) условие окончания глобального поиска.

Последнее условие представляет собой сложную самостоятельную проблему. Существуют различные приемы, разрешающие эту проблему [48-50,165].

Алгоритмы глобального поиска делятся на три категории:

- а) независимый;
- б) блуждающий;
- в) комбинированный.

Одним из «независимых» алгоритмов глобального поиска является случайный перебор. Метод Монте-Карло определяют случайные точки в области поиска и запоминается наилучшее значение целевой функции. Сходимость этого алгоритма к глобальному минимуму чрезвычайно медленная, к тому же затруднительно поставить условия окончания поиска. По существу такой поиск почти ничем не отличается от простого перебора.

Усиление этого алгоритма возможно при введении адаптации. После определенного количества независимых испытаний выделяется подобласть около точки с наименьшим значением $F(X)$ и новые испытания проводятся в этой подобласти и подобласти и т.д. Иначе говоря, поиск стягивается вокруг наилучшей точки, и координаты минимума при этом уточняются. Этот прием позволяет уменьшить затраты на поиск, но они все же остаются значительными, и оптимизация сложной конструкции при помощи этого алгоритма не под силу даже современным ЭВМ. Другим недостатком подобного алгоритма является то, что при недостаточно плотном распределении начальной серии проб всегда имеется возможность оставить глобальный минимум вне рассмотрения, стягивая поиск около какого-либо локального минимума.

Более мощны, с точки зрения затрат на поиск глобального минимума, блуждающие алгоритмы глобального поиска. Эти алгоритмы предусматривают применение самообучения в процессе поиска, причем самообучение в них сознательно «передетерминируется». Это делается для того, чтобы позволить алгоритму преодолевать «перевалы» с тем, чтобы находить новые «бассейны» локальных минимумов. Блуждающие алгоритмы находят не сами локальные минимумы, а точки, «подозрительные» на близость к локальным минимумам. После окончания работы блуждающего алгоритма глобального поиска необходимо включение в поиск алгоритмов локального поиска, которые поочередно из «подозрительных» точек близлежащие минимумы, из которых затем определяются глобальной. Одним из недостатков «блуждающих» алгоритмов является трудность их настройки на конкретную задачу. Если самообучение слишком «передетерминировать», то алгоритм не будет реагировать на улучшение $F(X)$ и поиск не даст результатов. Если же самообучение недостаточно «передетерминировать», то появляется возможность того, что алгоритм не сможет преодолеть какой-либо «перевал» и «застрянет» в каком-то минимуме, не обязательно глобальном. К тому же при недостаточно жестком самообучении увеличиваются непроизводительные затраты на преодоление «перевалов», так как при этом алгоритм часто движется по спирали.

Различные модификации «блуждающих» алгоритмов глобального поиска были разработаны Л.А. Растргиным [123-125], Ю.Я. Тарнопольским [137] и др.

«Комбинированные» алгоритмы глобального поиска представляют собой комбинацию алгоритмов «Монте-Карло», дающего случайные точки, равномерно распределенные в области поиска, и алгоритма локального поиска, производящего поиск ближайших минимумов из точек, данных алгоритмом Монте-Карло. Такие алгоритмы имеют преимущества по сравнению с «блуждающими», так как при работе

«комбинированных» алгоритмов исключаются затраты на преодоление «перевалов». К комбинированным алгоритмам относятся алгоритмы, предложенные Л.А. Растригиным [90,123-125], А.И.Половинкиным [117]. Общим недостатком всех перечисленных алгоритмов глобального поиска является то, что локальные минимумы при глобальном поиске могут многократно отыскиваться и непроизводительные затраты при этом могут составлять большую часть общих затрат [2,3,50,51,113-118,137,141,146].

1.2. Прикладные программы задач оптимизации

В последнее время все более широкое распространение получили разработка и применение пакетов прикладных программ оптимизации (ППП-ОП). Как и ППП, предназначенных для решения других классов задач, ППП-ОП имеют две основные черты [7,8,12,14,15,16,18,19,32,33,34-36]:

- ППП – это совокупность программных средств, ориентированных на решение определенного класса задач;

- в ППП входит монитор (управляющая программа) и набор программных модулей. По способу общения с пользователем различают ППП: библиотечного типа, со специализированным и универсальным монитором. Кроме этого, по принципу работы можно различать ППП, работающие в автоматическом (пакетном) или в интерактивном (диалоговом «человек-машина») режимах.

Кроме перечисленных общих черт ППП-ОП имеют свои особенности. Они могут быть как инвариантными по отношению к классам оптимизируемых объектов, так и ориентированными на них.

ППП-ОП посвящена довольно обширная литература [38], в том числе и обзорная [10], по которой можно полно ознакомиться с имеющимися разработками в этой области. Поэтому мы не будем подробно останавливаться на обзоре, а рассмотрим принципы организации и особенности некоторых, наиболее интересных на наш взгляд ППП-ОП.

Пакет программ «Оптимизация» [38] выполнен как пакет с универсальным монитором и для решения широкого класса задач нелинейного программирования. Этот пакет инвариантен по отношению к объекту и может использоваться в различных САПР. При этом постановка задач оптимизации и их решение происходит в интерактивном режиме с управлением через алфавитно-цифровой дисплей. При автономном использовании пакета задача оптимизации в виде загрузочного модуля подготавливается пользователем, а вся работа происходит в пакетном режиме.

Авторы делят задачи нелинейного программирования на 4 класса: задачи безусловной оптимизации; задачи глобальной оптимизации в гиперкубе; задачи минимизации с ограничениями в виде неравенств; общие задачи нелинейного программирования (с ограничениями типа равенств и неравенств). Для решения этих классов задач в пакете предусмотрено 4 типа оптимизационных модулей. Для всех задач выработана единая форма представления численных характеристик – паспорт задачи, который заполняется пользователем. В пакете также принято единое представление (также в форме паспортов) характеристик всех оптимизационных модулей, входящих в состав пакета и основанных как на регулярных, так и на статистических методах поиска.

Отличительной особенностью пакета «Оптимизация» по сравнению с большинством существующих ППП-ОП является автоматический выбор оптимизационного модуля для решения конкретной задачи, который происходит в два этапа. На первом этапе рассматривается условие применимости модулей к решению данной задачи. В этой процедуре «формального выбора» решаемой задаче из состава и библиотеки и ставится в соответствие несколько оптимизационных модулей по некоторым формальным признакам. На втором этапе производится процедура «оптимального

выбора». Из всех формально допустимых модулей выбирается тот, у которого оценка эффективности по заданному пользователем критерию (наджность, быстродействие, точность) выше. По всем допустимым задачам и по каждому подклассу задач лицевой счет каждого модуля содержит следующие оценки:

- общее количество задач (K_1);
- количество решенных задач (K_2);
- время, затраченное на все задачи (T_1);
- время, затраченное на решенные задачи (T_2);
- количество вычислений функций для всех поставленных задач (Kf_1);
- количество вычислений функций для решенных задач (Kf_2);
- средняя фактическая точность решения по функции (для тестовых задач) (E_f);
- средняя фактическая точность по аргументам (E_0).

На основании этих оценок определяются окончательные оценки эффективности модуля: надежность $P = K_2/K_1$; быстродействие $T = Kf_2/K_2$; точность $E = E_f$.

По мере использования пакета и решения из задач лицевые счета модулей пополняются.

Оценки эффективности могут быть только «интегральные» - по всем поставленным и решенным модулем задачам, но и могут быть отнесены к различным подклассам задач. Отнесение задачи к подклассу происходит таким же образом, как и при допустимости задачи на этапе формального выбора, т.е. по паспорту задачи.

Применение такого двухэтапного подхода к проблеме автоматизации выбора алгоритмов в ППП-ОП считаем весьма целесообразным и перспективным. Однако, следует заметить, что авторы не рассматривают вопросы автоматизации настройки алгоритмов на подклассы задач, переключения алгоритмов в ходе решения задачи, или образования оптимизирующей последовательности алгоритмов, автоматизации дополнительного исследования моделей в необходимых случаях, т.к. идентификация задачи по формальным признакам, заданным в паспорте задачи, может привести в ряде случаев к слишком «широким» подклассам. Эти вопросы, на наш взгляд, должны найти отражение при создании самообучающихся систем оптимизации.

Разработанный под руководством А.И.Половинкина ППП-ОП является частью пакета для автоматизации поискового конструирования [117] и представляет собой пример проблемно-ориентированного ППП. В нем применяются разработанные авторами эффективные алгоритмы случайного поиска как локального, так и глобального экстремума. Пакет может работать как в автоматическом, так и в интерактивном режимах. В нем предусмотрены процедуры постановки задач, формирования модели (из библиотеки моделей) и выбора алгоритма оптимизации (из пакета программ оптимизации). Управление пакетом производится при помощи универсального монитора. Проблемную ориентированность пакета определяет класс конструируемых объектов, для которого предназначен общий ППП поискового конструирования.

В целом ППП-ОП, их состоянию и проблемам развития можно сделать вывод о том, что пакеты библиотечного типа пока наиболее распространены и лишь в последнее время получили развитие пакеты с универсальными мониторами. Большое и вполне оправданное развитие имеет интерактивный режим в ППП-ОП. Однако, следует заметить, что как пакеты библиотечного типа, так и диалоговые пакеты предполагают значительную степень подготовки пользователя. А т.к. пользователь применяет пакет для решения практических задач из своей области деятельности, то в этом случае необходима его высокая классификация сразу в двух областях: методах оптимизации и основной специальности, что достижимо далеко не всегда. Поэтому разработка пакетов и систем с универсальным монитором, работающих в пакетном режиме, имеющих универсальные мониторы, способные автоматически выбирать оптимальные алгоритмы для решения поставленной задачи имеет также большое научное и прикладное значение.

Отметим также закономерно все большее развитие проблемно-ориентированных ППП-ОП, которые, обладая максимальным быстродействием (ввиду применения специально ориентированных на объект алгоритмов), меньшей занимаемой памятью компьютера, позволяют автоматически формировать математические модели задач оптимизации класса объектов и выдавать всю необходимую информацию по проектируемому объекту.

1.3. Оптимальное проектирование инженерных конструкций

Широкое применение в промышленном и гражданском строительстве тонкостенных конструкций приводит к тому, что к ним предъявляются все более жесткие требования, которые связаны не только с экономией средств, материалов при возведении различных сооружений, но и с созданием конструкций минимального веса, что имеет первостепенное значение в таких отраслях народного хозяйства, как самолетостроение и ракетостроение.

Широкие исследования по вопросу определения рациональных форм арок провели А.И.Виноградов [22-27] и В.А.Киселев [58,59]. В.А. Киселев в [58] исследовал различные виды арок, выявляя теоретико-рациональные формы. При этом основные результаты были получены для трехшарнирных и двухшарнирных арок. А.В.Виноградов в [22-27] занимался поиском теоретически рациональных форм бесшарнирных арок. Полученные результаты [21,48,155-169] позволили сделать заключение что теоретически наиболее выгодными по объему являются равнопрочные бесшарнирные арки с мощными пятнами. Однако при проектировании наличие каких-либо ограничений может привести к тому, что теоретико-рациональная форма арки не достижима. Нахождение оптимальной формы арки в этом случае классическими приемами, затруднительно или вовсе невозможно. В таких случаях целесообразно применение вариационных методов [1,20,93,127,143,167]. Большое внимание уделяется вопросам проектирования арок по предельным нагрузкам. Такой подход к проектированию арок позволяет вскрыть все ресурсы конструкции и в некоторых случаях значительно снизить вес по сравнению с проектированием по упругому расчету. К этому направлению относятся работы [116,151,165]. С проектированием арок наименьшего веса при продольном изгибе можно ознакомиться по работе [22].

Оптимальное проектирование оболочек представляет собой задачу более сложную, чем описанная выше задача оптимального проектирования арок. Это связано с большей трудностью решения прямой задачи расчета конструкции. Именно с этими трудностями связано почти полное отсутствие работ по оптимальному проектированию оболочек (особенно не осесимметричных моментных) до недавнего времени. Лишь бурное развитие вычислительной техники, вычислительной математики и особенно такого ее раздела как математическое программирование привело к возможности решать задачи оптимизации оболочек. При этом, в исследованиях, посвященных оптимальному проектированию оболочек выделяются три основных направления [19,33,51,52,77,78-87,94,106,107,108,129,131,132,146,147-151,154,155-169]:

1. Методы конструктивной оптимизации.
2. Итерационные методы оптимизации, предусматривающие многократное решение прямой задачи поверочного расчета.
3. Прямые методы оптимизации.

К направлению конструктивной оптимизации относятся работы, в которых проектирование оптимальной конструкции осуществляется на основе аналитического решения соответствующей прямой задачи, включающей в общем виде геометрические параметры. К этому направлению, в основном, относятся задачи устойчивости

подкрепленных конструкций, и основные практические результаты получены именно в области устойчивости подкрепленных оболочек.

Опишем постановку задачи для оптимизации каркасированной цилиндрической оболочки, нагруженной осевой сжимающей силой. Если при этом оказывается возможным n форм потери устойчивости панели, выделяемой подкрепляющими элементами), то в формулы, определяющие соответствующие критические нагрузки, вводится $n+1$ геометрических параметров. n из них выражаются через одно из условий равноустойчивости по различным формам, а последний из дополнительных, например, равенства критической силы заданному значению, максимума критической силы, минимума веса. Полученные при этом уравнения позволяют определить все геометрические параметры. К этому направлению можно отнести работы [19,34-37,40,42,48,49,59,60,76,96-100,132].

Недостатком метода конструктивной оптимизации, кроме сложности решения полученной системы уравнений, является то, что такая постановка задачи предполагает наличие жестких ограничений в форме равенств. Получающаяся при этом система уравнений имеет единственное решение, которое может оказаться невыполнимым по конструктивным или технологическим соображениям. В связи с этим более целесообразным представляется записывать ограничения в форме неравенств, что ведет к привлечению для оптимизации аппарата математического программирования. В работе И.Н.Гинзбурга и С.Н.Кана [34,35] производится выбор оптимальных параметров цилиндрической оболочки, нагруженной осевой сжимающей силой. Рассматривается круговая изотропная цилиндрическая оболочка заданной длины L и материала, нагруженная осевой сжимающей нагрузкой N . Определяют толщину оболочки δ и радиус средней поверхности R , дающие минимум веса оболочки: $G = 2\pi\gamma RL\delta$, где γ - удельный вес материала; при выполнении условий прочности и устойчивости. Задача оптимизации ставится как задача математического программирования, которую авторы решают при помощи метода множителей Лагранжа, обобщенного на случай, когда ограничения даны в виде неравенств. Применение этого метода приводит к системе нелинейных алгебраических уравнений, решение которой позволяет определить искомые параметры. При этом сами авторы подчеркивают, что применение метода Лагранжа к подобным задачам может привести к труднопреодолимым сложностям и применение методов нелинейного программирования позволит решать более сложные задачи.

Эту же задачу решили Ю.М. Почтман и Г.В.Филатов [120,121] при помощи метода случайного поиска. Они использовали алгоритм независимого глобального поиска с адаптацией. Применение этого алгоритма позволило избежать математических трудностей, но привело к значительному количеству прямых расчетов оболочки.

Второе из отмеченных направлений в теории оптимального проектирования предусматривает процесс последовательного улучшения конструкции, согласно принятому критерию оптимальности. Это направление включает в себя как итерационные методы оптимизации, так и методы математического программирования, а также методы случайного поиска.

В работах В.П.Малкова [78-87], И.Н.Калинина [52] оптимизация осесимметричных оболочек вращения производится итерационным путем. За оптимальную конструкцию В.П.Малков считает оболочку дискретной равнонапряженности. Он расчленяет оболочку на n -частей и подбирает толщину, постоянную в пределах одного участка, каждой части. Из начального состояния $H^0 = (h_1^0, h_2^0, \dots, h_n^0)$ производится итерация в зависимости от напряжения в каждой части. Конструкция проводится к такой, в которой максимальное напряжение в каждой части равно или близко к допускаемому. Подобный же подход к оптимальному проектированию осесимметричных подкрепленных оболочек применяет И.Н.Калинин. В качестве оптимизируемых параметров он рассматривает толщину обшивки, площадь поперечного сечения шпангоутов, расположение шпангоутов по длине оболочки. Применение итерационного процесса, как показано в результатах, дает

неплохие результаты. Хотя сходимость итераций в подобных случаях не доказана, и, даже, для некоторых задач возможно расхождение итерационного процесса, по результатам работ [48] можно сделать вывод о том, что в тех случаях, когда удается построить быстро сходящийся итерационный процесс, оптимальное проектирование конструкций этим методом может дать неплохие результаты.

Оптимизация конструкций с учетом надежности их функционирования нашла отражение в работах Р.К.Фурунжиева [144,145], Ю.Н.Почтмана [119,120] и др. При этом целевая функция принимается в виде:

$$F(X) = C(X, Y, Q, \Lambda, t), \quad (1.19)$$

где C – нелинейный стохастический оператор;

$X = (x_1, x_2, \dots, x_n)$ – вектор номинальных значений оптимизируемых параметров конструкции в начале эксплуатации;

$Y = (y_1, y_2, \dots, y_b)$ – вектор параметров, определяющих напряженно-деформированное состояние системы;

$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$ – случайные отклонения параметров конструкции от номинальных значений с нулевым математическим ожиданием;

$Q = (q_1, q_2, \dots, q_g)$ – вектор случайных возмущений;

t – время.

Требуется определить $X_{\text{опт}}$, удовлетворяющие допустимому множеству параметров состояния Y , определяемых условиями надежности, сообщающие экстремум целевой функции (1.19). Граница допустимой области Y соответствует предельным состояниям конструкции по прочности, жесткости, трещиностойкости, а также по величине допустимых ускорений в различных точках объекта.

Для решения подобных задач наиболее целесообразно применение методов случайного поиска.

К указанному направлению относятся работы [6,9,31,38,44,55-57,68,74,95,111-116,123-125,137,139,140,143].

К третьему направлению теории оптимального проектирования относятся работы, в которых рассматривается задача о выборе оптимального содержания армирующих элементов в композиции. При этом форма конструкции, структура пакета и нагрузки считаются заданными. Сформулированная задача решена для осесимметрично нагруженной пластинки в работе [4,5] и безмоментной ортотропной оболочки вращения [19,28,29,53,54,60-67,69-73,77-88,94,104-110].

Более подробно с вопросами оптимального проектирования конструкций можно ознакомиться по обзорным работам [8,9,27,40,122,136,138,161].

Интересен подход А.И.Половинкина [117] к оптимизации конструкций. Он предложил так выбирать пространство параметров, чтобы они могли определить все возможные схемы конструкции, способные функционировать в данных условиях. Причем это могут быть как известные, так и не известные, так и не известные схемы. Это пространство параметров названо универсальным пространством параметров (УПП). Например, если требуется определить оптимальную конфигурацию сечения бруса, имеющего ось симметрии, то УПП можно построить следующим образом:

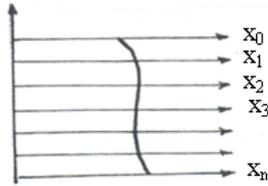


Рис. 1.1

Придавая различные значения компонентам x_i , можно добиться любой конфигурации сечения. Увеличивая количество переменных x_i , мы, тем самым, увеличиваем точность нахождения оптимальной конфигурации сечения. Если по смыслу задачи требуется, чтобы линия контура сечения была плавной, без изломов и т.д. то в алгоритмах, предложенных А.И.Половинкиным, предусмотрено наложение некоторой функциональной зависимости на изменения параметров Δx_i , с тем, чтобы линия контура на каждом шаге поиска соответствовала наложенным ограничениям. Таким образом, из рассмотрения сразу же выводятся схемы непригодности для функционирования в поставленных условиях.

Применение методов случайного поиска, как это видно из сделанного обзора, к оптимальному проектированию моментных оболочек, в настоящее время еще мало. Это связано с тем, что разработанные ранее алгоритмы случайного поиска (особенно глобального) предполагали слишком большое количество решений прямой задачи для нахождения минимума целевой функции. Это видно по результатам работы Ю.М.Почтмана и Г.В. Филатова [119,120], в которой было затрачено 4500 испытаний для нахождения глобального минимума, что, безусловно, очень много. Дальнейшее развитие теории случайного поиска и практики его применения для решения задач оптимизации конструкций, позволяет решать задачи оптимального проектирования со значительно меньшими потерями. В частности решение задач глобальной оптимизации арок и цилиндрических оболочек при помощи алгоритмов ГП-3 и ГП-4, описанных в [117] потребовало значительно меньших затрат, чем в [120]. Эти алгоритмы, ввиду соизмеримости затрат при оптимизации с другими итерационными методами и методами нелинейного программирования, позволяют производить оптимизацию достаточно сложных конструкций. Кроме того, универсальность ГП-3, ГП-4 и их способность решать многоэкстремальные задачи (что неспособны любые другие методы оптимизации) являются теми положительными качествами, которые позволяют решать при их помощи различные задачи без изменения алгоритмов.

1.4. Принципы разработки алгоритмических банков для оптимизационных задач

Исследования алгоритмизации и решения задач оптимизации инженерных конструкций описана в п. 1.6, реализуется на ЭВМ в виде алгоритмической системы [46-50,96-103,111-116,155-169], структурная схема которая имеет вид (рис. 1.2):

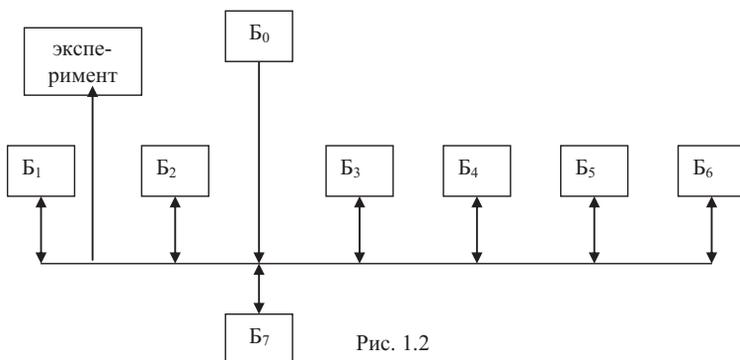


Рис. 1.2

Система в общем виде состоит из шести алгоритмических банков $B_i (i = 1, 6)$, двух вспомогательных B_0 и B_7 , а также экспериментальной установки, связанной с системой каналами связи.

Опишем кратко назначение банков для оптимизирующей системы. B_0 – банк постановки задачи – предназначен для записи задачи в форме удобной для дальнейшей переработки системой.

B_1 – банк данных – служит для хранения всех численных данных, необходимых для решения задачи.

B_2 – банк законов – предназначен для хранения основных законов, управлений, граничных и начальных условий, ограничений, характеризующих модели различных классов задач, и для формирования математической модели конкретной задачи.

B_3 – банк признаков – служит для хранения и переработки логической информации, формирования на ее базе значений признаков, необходимых для идентификации каждой конкретной задачи и выводе ее математической модели.

B_4 – банк моделей – служит для исследования и идентификации математических моделей задач.

B_5 – банк алгоритмов предназначен для хранения всех необходимых данных об имеющихся алгоритмах решения оптимизационных задач, а также для выбора оптимального алгоритма для каждой конкретной модели.

B_6 – банк прикладных программ – предназначен для хранения всех поисковых программ, соответствующих алгоритмам из B_6 , формирования рабочих программ решения задачи с передачей на счет.

B_7 – оперативный банк. В нем находится управляющая программа и происходят все промежуточные операции.

Алгоритмические банки, кроме B_1 и B_6 , предназначены для переработки логической информации. Все алгоритмические банки имеют общую структуру, т.е. они состоят из информационной и операционной частей. В информационных частях банков содержится логическая и символьная информация в закодированном виде. Исключения составляют B_1 , где в информационной части содержатся значения исходных данных, коэффициентов, констант и т.д. и B_6 , в информационной части которого содержится пакет прикладных программ, оформленных в виде стандартных процедур, или модулей.

В операционных частях банков содержатся сервисные программы, производящие переработку логических и символьных массивов из информационных частей и представляющие результаты для дальнейшего функционирования системы.

Алгоритмические банки (АБ) оптимизационной системы должны удовлетворять нижеследующим требованиям.

B_0 должен обеспечить:

1. унификацию постановок экстремальных задач;
2. их корректность;
3. удобную запись исходных данных;
4. связь между внешним языком постановки задачи и внутренним языком системы;
5. максимальную простоту постановки задачи;
6. контроль за правильностью постановки задачи.

Вышеперечисленные требования к B_0 предьявляет сама идея системного подхода к решению классов задач. Обеспечить выполнение их целесообразно при помощи применения специальных карт постановки, которые должны способствовать максимальной формализации и унификации постановок задач с учетом специфики каждого класса. Для связи информации, заносимой в карту с внутренним языком системы возможно применение различных способов. Укажем некоторые из них:

1. Для постановки задач разрабатывается специальный входной язык высокого уровня, способный описать все задачи требуемого класса, учитывающий специфику их и доступный специалистам соответствующего направления. При этом с помощью транслятора, разработанного для этого языка, информация, записанная на карту переводится во внутренний язык системы.

2. В карте постановки предусмотреть специальные признаковые массивы, элементы которых принимают значение «1» и «0» в зависимости от наличия или отсутствия соответствующей информации. Таким образом, информацию о постановке задач система получит от этих массивов. При таком подходе алгоритмические части в в постановке задачи (функции, уравнения и т.д.) следует сразу записывать на языке системы.

Первый подход к связи постановки задачи с системой универсальнее второго, но в реализации гораздо сложнее. Оба подхода дают возможность записывать задачу технику-программисту без привлечения специалистов высокого класса.

В B_0 необходимо предусмотреть блок проверки постановки задачи, цель которого - выявить возможные ошибки до начала работы всей системы, что необходимо для экономии машинного времени. Этот блок может быть оформлен в виде небольшой сервисной программы, которая начинает работать сразу же после ввода в компьютер информации с карты постановки.

Таким образом в информационной части B_0 – содержится информация, занесенная на карту постановки задачи, а в операционной блок проверки, операторы ввода исходных данных в B_1 и признаковов массивов в B_3 .

B_1 должен обеспечить:

1. прием и хранение всех данных из B_0 ;
2. прием и хранение промежуточных данных, вычисляемых в процессе исследования и решения задачи;
3. хранение всех данных (констант, коэффициентов и т.д.), общих для данного класса задач;
4. рациональное размещение всех данных в памяти компьютер.

Как видно из перечисленных требований информационная часть B_1 делится на сохраняемую (п.3) и не сохраняемую (пп. 1,2), которая в свою очередь делится на вводимую (п.1) при вводе каждой новой задачи и промежуточную (п.2).

Требование п.4. – естественно требование экономии памяти компьютер. Это требование можно удовлетворить, создав сервисную программу, способную в зависимости от характера каждой задачи резервировать в памяти компьютер массивы необходимой длины. Отсюда видно содержание операционной части B_1 , в которой должны содержаться программы экономии памяти, поиска и обработки данных.

B_2 – должен обеспечить:

1. хранение основных законов, уравнений, формул, необходимых для вывода мат. моделей;

2. направленную переработку информационной части;

3. формирование мат. моделей;

Следовательно информационная часть B_2 состоит из записанных на специальных алфавитах выражений (п.1). Операционная часть B_2 должна содержать программы поиска, переработки (аналитического дифференцирования и интегрирования, алгебраических действий, подстановки, приведение подобных членов и т.д.) и формирования моделей.

В общем случае для решения задач механики деформируемого твердого тела описанные требования к B_2 сохраняют силу. Однако, в ряде случаев формализовать построение мат. моделей чрезвычайно сложно (например в задачах проектирования строительных и др. конструкций, где кроме законов и теорий механики, учитываются требования СНиПов, ГОСТов и др. нормативов), а порой невозможно (например в задачах сложного нагружения). В этих случаях содержимое B_2 может быть изменено и вместо полной системы вывода мат. модели могут содержаться отдельные процедуры, формирующие наиболее часто встречающиеся частные модели. Возможен также вариант системы с пустым B_2 . В этом случае модель целиком задается в B_0 .

B_3 – должен обеспечить:

1. хранение логической информации о признаковом пространстве, описывающем все характерные особенности решаемых классов задач;

2. воспринимать и расшифровывать поступающую информацию из B_0 ;

3. перерабатывать всю информацию и определять значения признаков, обеспечивающих вывод и идентификацию моделей, выбор алгоритмов и составление рабочих программ.

Отсюда видно, что информационная часть B_3 состоит из неизменяемой и изменяемой частей, которые содержат логическую информацию. В оперативной части находятся программы анализа и расшифровки информации, полученной из B_0 , сравнения ее с признаковым пространством и формирования всех признаков.

B_4 – должен обеспечить:

1. хранение логической информации о возможных мат. моделях, принадлежащих рассматриваемым классам задач;

2. анализ модели, поступившей из B_2 ;

3. идентификацию модели.

Таким образом, информационная часть B_4 также состоит из изменяемой и неизменяемой частей. Операционная часть состоит из программ исследования и идентификации моделей.

Исследования моделей в системах решения оптимизационных и обратных задач различных классов инженерных конструкций осуществляется по двум направлениям:

1. Анализ и идентификация функций и операторов;

2. Анализ и идентификация области поиска.

Первое направление обеспечивается применением процедур, построенных на основе формализации определенных методов математического и функционального анализа.

Второе же, кроме этого, должно дополняться программами статистического анализа.

B_5 должен обеспечивать:

1. хранение всех данных об алгоритмах (для решения прямых, обратных и оптимизационных задач), заложенных в систему;

2. выбор алгоритмов решения прямой задачи;

3. построение вычислительной схемы и разрешающих уравнений;

4. выбор алгоритмов оптимизации.

Информационная часть B_5 обеспечивает выполнение первого требования. В операционной же части в общем случае должны находиться процедуры [48]:

- построение вычислительных схем;
- построение разрешающих уравнений;
- проверки сходимости и устойчивости;
- выборы оптимальной последовательности алгоритмов оптимизации;
- выработка сигнала выбора и подстановки в разрешающие уравнения исходных

данных;

- выработки сигнала выбора программ из пакета прикладных программ банка B_6 .

О проблеме выбора оптимальной последовательности алгоритмов оптимизации и методах ее решения сказано в п. 1.5.

B_6 должен обеспечить:

1. хранение пакета прикладных программ;
2. составление рабочей программы;
3. счет.

Об информационной части B_6 упоминалось выше, а операционная, как видно из требований, состоит из программ, реализующих генерацию из стандартных модулей или обращение к последовательности соответствующих процедур с настройкой их параметров.

B_7 должен обеспечить:

1. управление системой;
2. выполнение промежуточных операций.

Управление системой подразумевает обеспечить связи АБ, передачу промежуточной информации между ними, установление последовательности работы АБ.

Принцип работы B_7 может быть построен как на автоматическом, так и на интерактивном режимах.

В автоматическом режиме система работает по следующей последовательности. Из B_0 информация заносится в B_1 , B_3 и B_7 . По значениям признаков, сформированным в B_3 , из B_2 в B_7 переписываются соответствующие уравнения, формулы, граничные условия, вызываются процедуры операционной части B_2 , которые формируют математическую модель задачи. На следующем этапе с помощью содержимого B_4 происходит исследование и идентификация модели, после чего включается а работу B_5 . По результатам работы B_5 вызывается в B_7 соответствующие исходные данные из B_1 , и процедуры (модули) из B_6 и формируется рабочая программа, которая передается на счет.

Применение интерактивного режима позволит сделать работу системы более гибкой, т.к. в зависимости от особенностей задач можно будет изменять ход работы системы (иначе говоря выключать из работы часть АБ, если в них нет необходимости). Кроме того, при решении оптимизационных задач будет возможно вмешательство исследователя на любом этапе поиска экстремума, это обстоятельство в некоторых случаях позволит ускорить процесс решения, т.к. формализованной методики выбора оптимального алгоритма оптимизации нет, и часто интуиция и опыт исследователя являются решающим фактором в определении алгоритма для различных этапов поиска. Однако следует заметить, что применение интерактивного режима можно лишь при наличии высококвалифицированных специалистов в области механики твердого тела и математического программирования.

1.5. Принцип определения оптимальной последовательности алгоритмов для классов оптимизационных задач

Как известно [11,48,113-115,165] различные алгоритмы имеют различную скорость сходимости при решении задач математического программирования. Различие их

скорости сходимости выражается не только в абсолютных цифрах, но и в самом виде графика $\varepsilon(K)$, где ε - достигнутая точность (допустим – радиус ε - окрестности вокруг точного решения X^*), K – количество итераций. Подобные графики можно получить при тестировании различных алгоритмов. Так, графики $\varepsilon(K)$ применительно к алгоритмам случайного поиска и некоторых регулярных приведены в работах Л.А. Растригина [90,123-125], где, в частности, иллюстрируется улучшение сходимости случайного поиска по сравнению с градиентным методом с ростом оптимизируемых параметров.

Таким образом, используя зависимость $\varepsilon(K)$, полученные экспериментальным путем для различных алгоритмов на различных моделях с различным значением n – (количество оптимизируемых параметров), мы сможем, заложив эти данные в память ЭВМ, для решения задачи с заданной точностью ε (при условии близости модели к рассмотренным тестовым) однозначно выбирать алгоритм с точки зрения его оптимальности.

Допустим, что для некоторой тестовой задачи кривые $\varepsilon(K)$ для трех различных алгоритмов имеют вид (рис. 1.3).

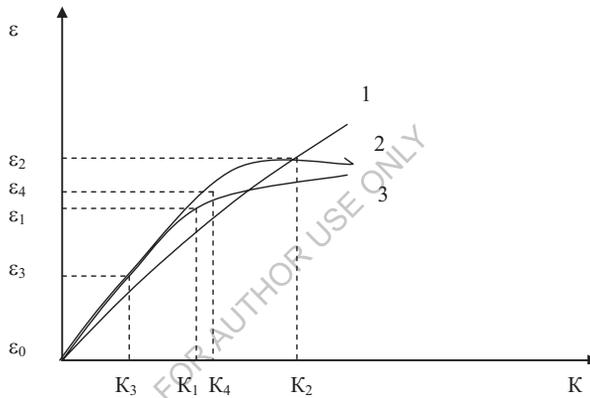


Рис. 1.3.

Как видно из рис. 1.3. на участке $\varepsilon_0 \varepsilon_1$, эффективнее алгоритм № 3, далее в зоне $\varepsilon_1 \varepsilon_2$ – № 2 и затем – № 1. Зная точки K_1 и K_2 пересечений кривых $\varepsilon_i(K)$, мы можем, задав минимум информации: номер модели, количество переменных, номера алгоритмов и точки пересечения, определять оптимальный алгоритм для каждой заданной точности.

Это довольно простой способ определения алгоритма с точки зрения оптимальности. Однако для использования подобной схемы выбора алгоритма исходной информации о задаче может оказаться недостаточно. Следовательно, потребуется произвести анализ модели с целью ее наиболее полной идентификации. Для этой цели необходимо привлечение аппарата статистического анализа, методов распознавания образов.

Пользуясь графиками $\varepsilon(K)$, построенными для различных моделей и алгоритмов, можно определять оптимальную последовательность алгоритмов с максимальной скоростью сходимости. Для этого необходимо исследовать кривые $\varepsilon'(K)$, т.е. первые производные $\varepsilon(K)$, определить точки их пересечения. При решении задачи с заданной ε целесообразно выбирать на каждом этапе поиска тот алгоритм, у которого $\varepsilon'(K)$ выше, чем у остальных, в точках K_3 и K_4 пересечения кривых $\varepsilon'(K)$ следует переходить на новый алгоритм (см. рис. 1.4).

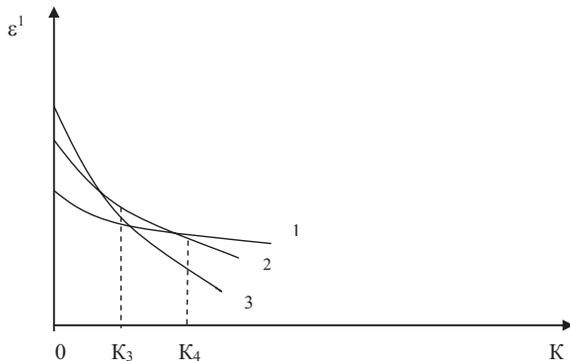


Рис. 1.4.

Как видим последовательность алгоритмов имеет более высокую скорость сходимости, чем каждый отдельный алгоритм.

Таким образом, имея в памяти компьютера данные о кривых $\varepsilon(K)$, можно определить не только оптимальный для данного класса задач алгоритм, но и их последовательность.

Выше говорилось об оптимальных алгоритмах, но не упоминалось о настройке алгоритмов. Известно, что все алгоритмы имеют свои параметры, от значения которых во многом зависит ход решения задачи. Правильно подобрать параметры алгоритма, с тем, чтобы он имел максимальную точность, скорость сходимости и надежность – задача сложная и может рассматриваться как оптимизационная, где максимизируется (минимизируется) критерий оптимальности алгоритма в пространстве его параметров. Следует отметить, что часто эта задача может оказаться многокритериальной (максимум точности, минимум испытаний, максимум надежности и т.д.).

Для решения задачи оптимизации алгоритмов с успехом можно применять алгоритмы случайного поиска. Регулярные методы в данном случае бесполезны, т.к. нет аналитического выражения целевой функции, не говоря уж о ее производных.

Кривые $\varepsilon(K)$ строятся по результатам решения тестовых задач только настроенных алгоритмов, т.е. прошедших этап оптимизации.

Для тестирования алгоритмов существует множество тестовых моделей, например описанных в [48,49,165]. Они между собой отличаются не только видом целевой функции, но часто одна и та же функция при различных значениях констант имеют различный вид бассейнов минимумов. Следовательно, для создания библиотеки тестовых функций их надо параметризовать, т.е. выделить все параметры, изменением которых можно добиться того или иного вида модели. Встречающиеся в литературе тестовые функции предлагаются при определенных значениях параметров, часто подобранных таким образом, чтобы показать предлагаемый автором алгоритм в лучшем свете. Когда же дело идет о библиотеке тестовых задач для тестирования пакета алгоритмов, то к ней предъявляются требования обеспечения максимального количества моделей при минимальных затратах памяти машины.

После определения оптимальных алгоритмов (или их последовательности) на тестовых задачах можно их применять к реальным моделям. Для этого необходимо решение следующих задач:

1. обезразмерить модель;
2. привести область существования решений ее к тестовым моделям;

3. стохастическая аппроксимация модели;
4. идентификация модели.

Две первые задачи решаются довольно просто. Обезразмерить модель можно следующим образом:

$$\bar{F} = \frac{F(X)}{F(X \max)} \quad \text{или} \quad \bar{F} = \frac{F(X)}{F(X \min)},$$

где $F(X)$ – текущее значение целевой функции, $F(X \max)$ или $F(X \min)$ – значение F при максимальных и минимальных значениях параметров. Приведенные области к заданным размерам производится при помощи линейного преобразования переменных. Например: если параметр меняется в пределах $a_i \leq x_i \leq b_i$, то, чтобы привести к отрезку $[0, 1]$,

следует ввести $\bar{x}_i = \frac{x - a_i}{b_i - a_i}$, тогда $0 \leq \bar{x}_i \leq 1$.

Третья задача заключается в аппроксимации модели некоей системой аналитических функций после приведения серии статистических экспериментов. По результатам этих экспериментов составляется таблица распределения значений целевой функции, которая аппроксимируется при помощи аналитических функций. Аппроксимация может производиться аналитически (беспоисково) путем решения соответствующей системы уравнений (что бывает слишком затруднительно) и при помощи поисковых методов, решая задачу минимизации невязок.

Здесь также встает оптимизационная задача, которая и решается при помощи алгоритмов, заложенных в систему.

Задача идентификации, или распознавания модели заключается в отнесении модели к тому или иному классу тестовых функций. Эта задача может решаться методами распознавания образов, а также поисковыми алгоритмами.

Задачи аппроксимации и идентификации моделей можно объединить, если за аппроксимирующие функции брать тестовые функции, заложенные в систему.

Решая поочередно задачу минимизации невязок путем определения значений констант тестовых функций, сравнивая между собой полученные невязки, можно модель отнести к той тестовой функции, у которой эта невязка минимальна. Если же определена некоторая норма близости модели и тестовых функций, то при получении на каком-либо этапе невязки не большей этой меры близости, идентификацию модели можно закончить.

Как решаются все перечисленные выше задачи в предлагаемой системе изложено ниже, в главе 3.

1.6. Постановка и анализ проблемы создания алгоритмической системы решения классов оптимизационных задач

Оптимизационные задачи, в частности задачи оптимизации инженерных конструкций, как видно из анализа, приведенного выше, предполагают использование широкого класса методов математического программирования от симплекс-алгоритма до глобальных алгоритмов случайного поиска. Постановка задач оптимизации и обратных задач расчета конкретных конструкций позволяет унифицировать методы их решения на основе применения перечисленных методов. Стоит отметить, что подобные задачи обладают рядом особенностей по сравнению с абстрактными задачами математического программирования, что позволяет разработать новые алгоритмы или модифицировать известные методы с ускоренной сходимостью. Из этих особенностей можно выделить следующее. Во-первых, при весовой оптимизации конструкций минимум целевой функции всегда будет находиться на одном или пересечении ограничений по прочности,

жесткости, устойчивости рассматриваемых конструкций. Эта особенность позволяет производить параметрическую адаптацию алгоритмов поиска. Во-вторых, задача прямого расчета конструкции, как правило требует на несколько порядков больше затрат машинного времени, чем вычисление целевой функции. Отсюда – возможность структурной адаптации алгоритмов с целью максимально уменьшить количество прямых расчетов конструкций. В - третьих, как прямые расчеты, так и обратные и оптимизационные для достаточно сложных конструкций производится при помощи численных методов. При этом очевидна целесообразность соотношения точности расчетной модели (которая может выражаться в количестве членов ряда координатных функций, узлов разностной сетки, конечных элементов) и положения поисковой системы в области поиска.

Следует отметить, что приведенные особенности задач механики твердого тела лишь предполагают возможность адаптации алгоритмов, но не создают предпосылок для создания одного оптимального для всех классов задач алгоритма. Невозможность создания подобного алгоритма обусловлена многообразием оптимизационных задач, техническими данными компьютера и временем, расходуемым на решение каждой поставленной задачи.

Множество методов решения задачи оптимизационных задач создает проблему оптимального алгоритма для решения конкретной задачи, а если учесть, что каждому методу соответствует множество алгоритмов с различными требованиями и характеристиками, то видно, что эта проблема может явиться достаточно сложной даже для специалиста высокого класса, тогда как для неквалифицированного пользователя может быть неразрешимой.

Совокупность требований, предъявляемых различными алгоритмами к задачам, таких как выпуклость, дифференцируемость определенное количество раз, обращаемость и достаточно хорошая обусловленность матрицы вторых производных целевой функции и ограничений, удовлетворение условию Липшица функций и их производных, отсутствие или наличие начальной точки и т.д. создает чрезвычайно большое множество ситуаций в выборе конкретного алгоритма, что исследователю трудно принять оптимальное решение. С другой стороны, проблему выбора оптимального алгоритма возможно и целесообразно автоматизировать, представив компьютеру производить необходимый анализ возможных вариантов.

Для классифицированного выбора алгоритма необходима информация о математических моделях задач более полная, нежели данные о выпуклости или невыпуклости целевой функции и ограничений. Для получения этой информации необходимо исследование моделей, которое может проводиться аналитически с применением ЭВМ и без, стохастически с определением закона распределения целевой функции и ограничений в области поиска, а также с применением специальных алгоритмов, позволяющих выявлять контуры области. Исследование моделей также представляется возможным автоматизировать в определенных рамках, что поможет более правильному выбору алгоритму или последовательности алгоритмов для решения конкретной задачи.

Для преодоления перечисленных трудностей, возникающих при постановке, анализе оптимизационной задачи, выборе и настройке алгоритмов наиболее целесообразным является системный подход, позволяющий комплексно рассматривать и решать встающие проблемы. Как видно из анализа методов оптимизации (п.1.1.), системы, или пакеты, предназначенные для решения оптимизационных задач, являющиеся частями САПР, а также отдельными системами, в основном, являются инвариантными по отношению к классам объектов. Это обстоятельство имеет две стороны. Положительная сторона заключается в том, что инвариантность пакета позволяет его подключать к другим системам (или к оптимизации других классов объектов при работе в индивидуальном режиме) по существу без доработок.

Отрицательные стороны инвариантности пакета состоит в том, что при этом не учитываются особенности объектов, в ряде случаев позволяющие значительно усилить алгоритмы, предназначенные для оптимизации именно данного класса объектов.

Из анализа также можно сделать вывод, что унификации постановок задач, исследованиям моделей, использованию результатов исследования моделей для автоматического выбора и настройки оптимальных алгоритмов в настоящее время уделяется неоправданно мало внимания. В предлагаемой работе рассматриваются и решаются вопросы автоматизации решения классов оптимизационных задач с единых позиций системного подхода. Причем рассматриваются вопросы создания как системы, инвариантной по отношению к объектам, так и проблемно-ориентированных ППП оптимального проектирования инженерных конструкций.

При постановке оптимизационных задач механики деформируемого твердого тела нет необходимости разрабатывать свою математическую модель для каждой задачи. Этот процесс успешно поддается формализации и автоматизации, что убедительно продемонстрировали труды В.К.Кабулова и его учеников [46-50,96-103,111-116,153,155-169].

На основе общих законов сохранения: массы, энергии и т.д. выводятся с помощью компьютера системы уравнений, описывающих любую математическую модель механики деформируемого твердого тела.

Таким образом для решения оптимизационных и обратных задач (назовем их в совокупности экстремальными) инженерных конструкций прослеживается цепочка, подлежащая формализации и автоматизации: вывод модели – исследование модели – выбор оптимального алгоритма, позволяющую максимально автоматизировать процесс постановки и решения оптимизационных задач.

Наиболее полна эта проблема может быть решена на основе алгоритмических методов, разработанных В.К.Кабуловым [46-50].

Общая схема алгоритмизации исследований, постановки и решения задач механики представлена на рис. 1.5.



Рис. 1.5.

Опыт здесь принимается в широком, философском смысле. Это опыт, накопленный человечеством, отраженный в монографиях, статьях, полученный в лабораториях и натуральных экспериментах. На этапе «опыт» предполагается создание информационных банков и широкая автоматизация экспериментов с разработкой средств сбора, передачи и обработки данных. На этапе «законы» накопленный опыт трансформируется в общие законы механики – законы сохранения и т.д. В алгоритмизации известные законы зашифровываются и вводятся в память компьютера. Новые законы формулируются по результатам автоматизированных экспериментов.

На основании общих законов решаются задачи, принадлежащие к различным классам (например, теории упругости, деформационной теории пластичности, реологические задачи и др.). Классификация этих задач и автоматическое распознавание классов выполняется на этапе «задачи».

Для каждой конкретной задачи по известным законам на этапе «модели» автоматически выводятся системы уравнений, являющихся математическими моделями задач.

На этапе «алгоритмы» происходит выбор конкретного алгоритма для решения полученной модели с учетом вопросов применимости и оптимальности.

После выбора алгоритма необходимо переходить к построению программ, реализующих счет, что и происходит на этапе «математическое обеспечение».

На последнем этапе «счет» создаются вычислительные комплексы, связанные с опытными установками, осуществляется увязка счета с опытом.

Таким образом, указанная система позволяет максимально автоматизировать творческий труд исследователя, начиная с постановки эксперимента и кончая обработкой численных результатов счета, оставляя ему неформализуемую часть творчества.

В связи с вышеуказанным наиболее перспективная помощь исследователям и инженерам представляется нам в разработке алгоритмической системы для решения задач механики сплошных сред [46-50,96,101-103, 165], а также (с учетом данных современных компьютеров) для более узких классов задач механики деформируемого твердого тела и строительной механики. Следует отметить, что разработку оптимизирующей системы, инвариантной по отношению к объектам также целесообразно вести, основываясь на указанных принципах алгоритмизации, все этапы которой необходимы и для функционирования инвариантной системы. Несколько изменена должна быть роль этапа «модели», т.к. в этом случае, очевидно, автоматизировать вывод и формирование математических моделей невозможно, но целью этапа «модели» остается исследование и идентификация моделей.

В данном параграфе рассмотрены лишь общие проблемы и вопросы, возникающие перед исследователями при постановке и решении оптимизационных задач, и требующих для своего решения системного подхода. Указаны пути реализации этого подхода, заключающиеся в создании АСО, о которой более подробно изложено ниже.

Глава 2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

2.1. Понятия оптимального и рационального решений

В настоящее время бурно развивается наука, занимающаяся изучением вопросов принятия решений для той или иной ситуации. Эта наука носит название «исследование операций» и позволяет заменить необоснованные, волюнтаристские решения научно обоснованными.

При проектировании и создании новых систем, конструкций, устройств, изделий, как правило, исходят из необходимости учитывать различные требования. Таковыми применительно к деформируемым механическим системам (конструкциям) являются прочность, жесткость, устойчивость всех элементов и узлов системы и т.д.

Решение, которое удовлетворяют всем заданным ограничениям, называется допустимым. Из допустимых в процессе решения экстремальных задач выбирается оптимальное или рациональное решение.

Под оптимальным будем понимать такое допустимое решение, которое является наилучшим с точки зрения выбранного критерия оптимальности.

Критерий оптимальности формируется на основе одного или нескольких критериев эффективности.

Но не все критерии эффективности ограничения, выражающие требования к проектируемым изделиям, могут быть формализованы, например, эстетичность, компактность системы, технологичность изготовления и пр. Такие критерии эффективности и ограничения анализируются путем качественной оценки проектируемого изделия экспертами и области эстетики, компоновки, технологии и т.д. В этом случае целесообразно ввести понятие рационального решения.

Под рациональным (от лат. ratio – разум, rationalis - разумный) будем понимать такое решение, которое получается неформальным путем, т.е. с учетом экспертных (или других неформальных) оценок.

При поиске оптимальной конструкции по принятому критерию оптимальности и ограничениям можно получить оптимальную, но рациональную конструкцию. Например, если требование технологичности не нашло количественной оценки и не вошло в число ограничений, то можно получить проект минимального веса (оптимальный по весу), но нерациональный проект в силу не технологичности. Более тяжелая конструкция, но удобная с технологической точки зрения, может оказаться более рациональной.

Поэтому проблему создания действительно рациональных изделий (конструкций) можно решить только благодаря творчеству различных специалистов, т.е. органическому сочетанию науки и искусства.

Оптимизационные задачи (и их различные постановки) представляют собой отдельные подзадачи в процессе разработки в целом рационального проекта.

2.2. Расчетной модели деформируемых систем

Всякая задача проектирования силовой конструкции начинается с выбора расчетной модели, которая получается путем идеализации.

Основные типы элементов деформируемых систем. При выборе расчетной модели, как правило, проектируемое изделие аппроксимируется типовыми элементами. Отметим основные из них.

Пространственный (массивный) элемент – тело произвольной формы, все габаритные размеры которого a, b, c ($a \leq b \leq c$) имеют, примерно, один порядок.

Оболочка – тело, форма которого определяется геометрической поверхностью, расположенной внутри него таким образом, что размер тела, нормальной к этой

поверхности (толщина h) существенно меньше по сравнению с габаритными размерами $a, b, c (a \geq b \geq c)$, т.е. $h \ll a$.

Пластина – тело, форма которого определяется срединной плоскостью, причем толщина h (постоянная или переменная) существенно меньше по сравнению с габаритными размерами a, b , т.е. $h \ll a (a \leq b)$.

Стержень – тело, форма которого образована движением центра тяжести плоской фигуры (постоянной или переменной площади) по некоторой траектории, при условии, что плоскость фигуры, называемая поперечным сечением стержня, остается перпендикулярной к данной траектории.

Выбор типовых элементов, как правило, связан с соответствующими математическими моделями описания состояния (напряженного, деформированного, устойчивости, колебаний и т.п.).

Конструктивные схемы. Классификация конструктивных схем может быть осуществлена по различным признакам. Например, в зависимости от используемых типовых элементов расчетные конструктивные схемы могут быть: пространственными, оболочечными, пластинчатыми, стержневыми и комбинированными.

К конструктивной схеме пространственного тела можно привести конструкции плотины, фундамента, корпусной детали машины, сложного узла соединения отдельных элементов и т.д.

К оболочечной конструктивной схеме могут быть отнесены конструкции купольного покрытия здания, участка трубопровода, отсека подводного аппарата и т.д.

К пластинчатой конструктивной схеме могут быть приведены отдельные подконструкции палубы судна, плоской крышки сосуда и т.д.

Примерами стержневых конструктивных схем являются фермы мостов, телемачт, опор линий электропередач, радиотелескопов и т.д.

К комбинированным конструктивным схемам относятся схемы, включающие различные типы элементов. Например, конструкция любого летательного аппарата представляет собой комбинированную (составную) конструкцию, состоящую из оболочек, стержней (стрингеров и шпангоутов), пластин.

Связи и их классификация. Все тела, как правило, взаимодействуют с внешней средой или другими телами.

Если вследствие каких-либо ограничений (условий) данное тело не может занять произвольную скорость, то такое тело называется несвободным. В этом случае ограничения (условия) называются связями. Связь налагает ограничения на изменение координат и скоростей фиксированных точек объекта (системы).

Связь на i -ю точку тела называется удерживающей (односторонней), если ее можно представить в виде неравенства

$$f(x_i, y_i, z_i, \dot{x}_i, \dot{y}_i, \dot{z}_i, t, A_j) \leq 0, \quad (2.1)$$

где x_i, y_i, z_i - координаты i -й точки, $\dot{x}_i, \dot{y}_i, \dot{z}_i$ - производные по времени t от координат i -й точки ($i = 1, 2, \dots, n$), A_j - заданные j -е параметры (константы).

В качестве функций f , как правило, применяются функции обобщенных перемещений.

Связь называется удерживающей, если ее можно представить в виде равенства

$$f(x_i, y_i, z_i, \dot{x}_i, \dot{y}_i, \dot{z}_i, t, A_j) = 0, \quad (2.2)$$

Если в связь (2.1) или (2.2) входит явно время t , то связь называется реономной или нестационарной*). Если же связь, не содержит параметра времени, то она называется склерономной или стационарной.

В том случае, когда в связь (2.1) или (2.2) производные по времени от координат $\dot{x}_i, \dot{y}_i, \dot{z}_i$ не входят, связи называются геометрическими или голономными. Так, уравнение удерживающей нестационарной голономной связи имеет вид

$$f(x_i, y_i, z_i, t, A_j) = 0, \quad (2.3)$$

Связь, выражение которой содержит производные по времени от координат, называется кинематической. При этом кинематическая связь будет называться неголономной, если выражения связи (2.1) – (2.2) путем интегрирования нельзя привести к голономной связи.

И наконец, различают идеальные и неидеальные связи. Под идеальными связями понимаются такие связи, реакции которых не содержат составляющих, обусловленных трением. Более общим определением идеальных связей является следующее: виртуальная работа реакций связей на любом виртуальном перемещении системы равна нулю.

Воздействия на деформируемые системы. Различают воздействия: силовые, температурные, кинематические, химические, различной природы облучения и др.

Силовые воздействия вызываются объемными и поверхностными силами. К объемным силам относят собственный вес, инерционные силы, магнитные силы и т.п. к поверхностным относят силы, возникающие от взаимодействия поверхностей твердых тел с жидкостью, глазами, со смесями, сыпучими средами, с другими конструкциями и телами.

Температурные воздействия могут вызывать изменение геометрической формы тела и изменение сил взаимодействия между его точками, изменение физико-механических свойств материалов.

Под кинематическими воздействиями понимаются изменения координат и скоростей координат точек деформируемых тел в силу эксплуатационных или других условий.

Химические воздействия изменяют свойства материалов во времени, что влияет на изменение напряженного и деформированного состояния механических систем. Так, надежность многих конструкций существенно зависит от химической агрессивности среды, в которой эксплуатируется конструкция.

При работе ряда современных конструкций в условиях облучения различной природы также могут существенно изменяться физико-механические свойства материалов, от которых зависит надежность конструкций. Такими облучениями могут быть радиоактивные, световые и др.

Воздействия могут быть статическими или динамическими, иметь случайный или детерминированный характер.

Механические свойства материалов. В расчетную модель необходимо закладывать такие механические свойства материалов, которые отвечают требованиям эксплуатации. Эти свойства предварительно изучаются экспериментальным путем. Свойства материалов характеризуют: модуль упругости, модуль сдвига, коэффициент Пуассона, предел пропорциональности, предел текучести, временное сопротивление в зависимости от различных параметров – относительной деформации, времени, скорости деформаций, температуры, интенсивности заданного облучения, влияния вынужденных колебаний определенной частоты и амплитуды, влияния агрессивной среды и т.п.

Учет статистической природы исходных данных в расчетной модели. Успех в создании рациональной конструкции зависит от удачного выбора расчетной модели. Расчетная модель должна быть достаточно простой, допускающей построение математической модели, и в то же время достаточно адекватно отражающей состояние реальной конструкции. Чаще всего в расчетной модели предполагается детерминированность формы и структуры проектируемого объекта, геометрических размеров элементов, внешних воздействий и свойств материалов. Между тем в действительности все исходные данные находятся под влиянием большого количества неучтенных факторов и поэтому в той или иной мере носят случайный характер. Например, изменчивость геометрических размеров и формы элементов конструкций играет существенную роль при потере устойчивости форм равновесия конструкции.

Почти все внешние нагрузки являются случайными. Случайный характер имеют и механические характеристики материала.

Случайный характер исходных данных в расчетной модели может быть учтен при выборе коэффициентов запаса согласно действующим нормам проектирования или применения статистических методов анализа состояния конструкции с учетом статистических исходных данных, входящих в расчетную модель.

2.3. Постановка задач параметрической оптимизации

Конечной целью проектирования является создание рациональных изделий, объектов, конструкций, систем и т.п., исходя из имеющихся ресурсов и возможностей.

Чтобы добиться наилучшего результата, необходимо оптимизировать на всех этапах (стадиях) проектирования.

Успехи в области теории экстремальных задач, в развитии численных методов решения краевых задач математической физики, в развитии компьютер обусловили интенсивное развитие теории и методов решения задач оптимизации в механике деформируемых систем.

В процессе проектирования возникает многообразие различных постановок задач оптимизации:

- поиск оптимальных типовых элементов,
- выбор оптимальных геометрических форм,
- выбор оптимальных структур конструкций,
- оптимальное распределение внутренних усилий за счет предварительных напряжений,
- подбор материалов,
- максимальное снижение концентрации напряжений в окрестности концентраторов,
- оптимальное формирование конструкций,
- создание конструкций минимального веса с заданной надежностью и т.д.

Общая постановка задач параметрической оптимизации может быть сформулирована следующим образом: требуется найти в n – мерном пространстве проектирования такую допустимую точку

$$x^* = \{x_1^*, x_2^*, \dots, x_i^*, \dots, x_n^*\}, \quad x^* \in D,$$

в которой достигается оптимальное значение $C(x^*)$ целевой функции $C(x)$, т.е.

$$C(x^*) = \underset{x \in D}{\text{opt}} C(x), \quad (2.4)$$

где $\text{opt} C(x)$ - оптимальное значение $C(x)$, D - область доступных решений

$$D = \{x : g_k(x) \leq 0, \quad k = \overline{1, m}\}, \quad (2.5)$$

Π – область поиска

$$\Pi = \{x : a_i \leq x_i \leq b_i, \quad i = \overline{1, n}\}, \quad \Pi \subset R_n^+, \quad (2.6)$$

a_i, b_i - наименьшее и наибольшее возможные значения i - го управляемого параметра x_i ,

$$R_n^+ = \{x : x_i \geq 0, \quad i = \overline{1, n}\}, \quad R_n^+ \subset E_n, \quad (2.7)$$

E_n - евклидово n - мерное пространство.

2.4. Схема получения оптимального решения

Рассмотрим укрупненную схему получения оптимального решения. Данная схема представлена на рис. 2.1.

Создать расчетную модель с полным учетом всех свойств реального объекта (деформируемой системы) принципиально невозможно. Искусство выбора расчетной модели состоит в том, чтобы выявить основные свойства реального объекта.

После выбора расчетной модели составляется математическая модель описания деформированного и напряженного состояний, динамических процессов и т.п. При связывающих все входящие параметры.

Наличие и предварительный анализ математической модели описания состояния деформируемой системы позволяет осуществить постановку задачи оптимизации и подобрать эффективный математический аппарат оптимизации.

Для оценки принимаемого решения (проекта) необходимо определить критерий оптимальности (критерий совершенства, критерий качества, критерий эффективности и т.п.). Целевая функция $C(x)$ является математической записью принятого критерия оптимальности.

Использование критерия оптимальности, ограничений, математической модели описания состояния, математического аппарата оптимизации (методов оптимизации) позволяет найти оптимальное решение, описывающее проектируемую конструкцию (модель объекта).

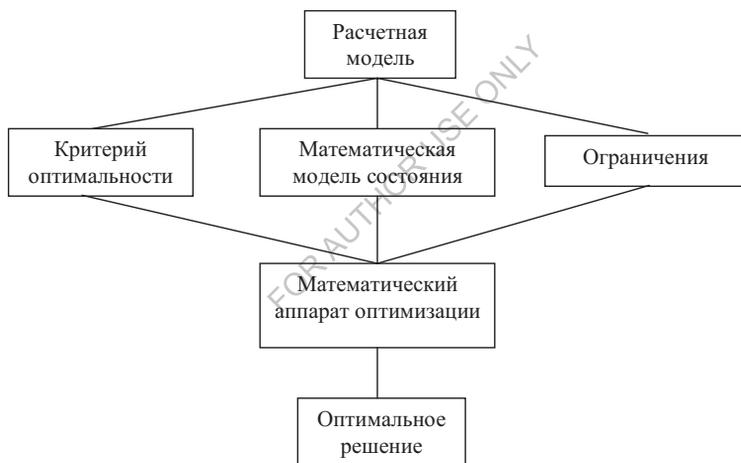


Рис. 2.1

Различные постановки задач оптимизации, методы решения задач и полученные результаты обсуждаются в ряде обзорных работ [2,8,19,31-33,48,77,78,81,82,88,104,108,116, 146-151,165].

2.5. Математическая модель оптимизации пластинчатых конструкций со сложной конфигурацией

При проектировании различных инженерных сооружений, а именно: объектов строительства, самолетостроения, ракетостроения, кораблестроения и др.- возникают задачи расчета и оптимизации пластинчатых элементов конструкции сложной конфигурации (пластины не прямоугольного очертания, с вырезами, многосвязные и т.д.).

Математическая сложность расчета подобных пластинчатых элементов произвольного очертания, особенно их оптимизации, привела к значительному отставанию научных исследований и публикаций по перечисленным вопросам от расчета и оптимизации «традиционных» видов элементов конструкций.

Задачу оптимизации конструкции поставим как задачу математического программирования: необходимо определить вектор $X(x_1, x_2, \dots, x_n)$ оптимизируемых параметров x_i ($i = 1, \bar{n}$), сообщающих целевой функции $F(x)$ экстремум (для определенности примем \min) при соблюдении ограничений на параметры $a_i \leq x_i \leq b_i$, ($i = 1, \bar{n}$) и функциональных ограничений $f_j(x) \leq 0$ ($j = 1, \bar{m}$). Эту задачу запишем в виде

$$\begin{aligned} F(X) &\rightarrow \min, \\ f_j(X) &\leq 0 \quad (j = 1, \bar{m}), \\ a_i &\leq x \leq b_i \quad (i = 1, \bar{n}). \end{aligned} \quad (2.8)$$

Рассмотрим несколько подробнее выражение (2.8). В качестве оптимизируемых параметров в элементах конструкций наиболее часто принимаются геометрические (толщина h пластины, радиус кривизны R_1 внешних, внутренних краев, вырезов и т.д.) и физические (модуль упругости E и др.). Характеристики нижние a_i и верхние b_i границы изменения параметров x_i задаются исходя из конструктивных, технологических, эксплуатационных и т.д. требований. За целевую функцию $F(X)$ наиболее часто принимается вес, расход материалов, стоимость конструкции.

Основными функциональными ограничениями конструкции. Основными функциональными ограничениями $f_j(X) \leq 0$ ($j = 1, \bar{m}$) при оптимизации инженерных конструкций подверженных различным внешним воздействиям, являются следующие.

1. Ограничения на напряженное состояние:

$$\max \sigma^{(\psi)}_{\text{экр}}(X) \leq [\sigma] \quad (\psi = 1, \bar{\varphi}). \quad (2.9)$$

Здесь ψ – номер варианта воздействия на конструкцию; $\max \sigma^{(\psi)}_{\text{экр}}(X)$ – максимальное эквивалентное напряжение конструкции, определяемое по принятой гипотезе или теории прочности при ψ - м варианте воздействия; $[\sigma]$ – допускаемое напряжение для материала конструкции при ψ - м варианте воздействия.

К канонической форме (2.8) ограничения (2.9) приводятся так:

$f_1(x)$

$$F_1(X) = \max \sigma^{(\psi)}_{\text{экр}}(X) - [\sigma] \leq 0.$$

2. Ограничения на деформированное состояние:

$$\max |u^{(\psi)}(X)| \leq [u] \quad \psi,$$

где $\max |u^{(\psi)}(X)|$ – максимальное перемещение поверхности конструкции при ψ - м варианте воздействия; $[u]$ – допускаемое перемещение поверхности конструкции.

В каноническом виде:

$$f_2(X) = \max |u^{(\psi)}(X)| - [u] \leq 0.$$

3. Условия устойчивости

$$P\psi \leq P_{кр.}$$

где $P\psi$ - сжимающее ψ -е воздействие, $P_{кр}$ - критическая сила воздействия на конструкцию.

В канонической форме:

$$f_3(x)$$

$$F_3(X) = P\psi - P_{кр} \leq 0.$$

4. Ограничения на частоты собственных колебаний.

При действии переменных (периодических) нагрузок с определенной частотой необходимо анализировать ограничения на частоты собственных колебаний:

$$\min \{ w_i^{(\psi)}(X) \} \geq [w] \psi \quad (i = 1, 2, \dots),$$

где $\min \{ w_i^{(\psi)}(X) \}$ - низшая частота собственных ψ -х колебаний, $[w] \psi$ - допускаемое наименьшее возможное значение частоты собственных ψ -х колебаний, назначаемое как расчетное значение вынужденных ψ -колебаний.

В канонической форме:

$$f_4(x)$$

$$F_4(X) = [w] \psi - \min \{ w_i^{(\psi)}(X) \} \leq 0$$

5. Ограничения на амплитуды механических колебаний:

$$a_0^{(\psi)}(X, w_i) \leq [a_0(w_i) \psi],$$

где $a_0^{(\psi)}(X, w_i)$ - максимальная амплитуда вынужденных ψ -х колебаний при частоте w_i , $[a_0(w_i) \psi]$ - допускаемая амплитуда.

Перечисленные выше ограничения наиболее распространены при оптимизации конструкций, однако при решении задач оптимизации конкретных конструкций могут быть наложены дополнительные конструктивные, технологические, эксплуатационные и другие ограничения.

Задача (2.8) оптимизации инженерных конструкции сложней конфигурацией представляет собой задачу нелинейного программирования, имеющая ряд специфических особенностей. Во первых, на вычисление целевой функции (вес, стоимость) требуется гораздо меньше времени, чем на проверку ограничений, которые требуют решения прямой задачи расчета конструкций; Во-вторых, глобальный минимум всегда будет находиться на какой-либо границе или на их стыке, иначе будем иметь запас материала, который можно убрать, не нарушая условий прочности, жесткости, устойчивости и т.д. В-третьих, вид функций $\sigma(X)$, $u(X)$, $P_{кр}(X)$ и т.д. априорно неизвестен и они могут быть определены только численно. Учитывая сказанное, для решения задачи (2.8) применим алгоритм глобального поиска ГП-3 [48,112], учитывающий перечисленные выше особенности. Алгоритм обладает повышенной скоростью сходимости и надежности.

Рассмотрим более подробно методы решения прямой задачи расчета.

Известно, что уравнения равновесия, колебания и устойчивости анизотропных пластин относительно моментов соответственно имеют вид [4].

$$\frac{\partial^2 M_1}{\partial x^2} + 2 \frac{\partial^2 M_{12}}{\partial x \partial y} + \frac{\partial^2 M_2}{\partial y^2} = q_1(x, y) \quad (2.10)$$

$$\frac{\partial^2 M_1}{\partial x^2} + 2 \frac{\partial^2 M_{12}}{\partial x \partial y} + \frac{\partial^2 M_2}{\partial y^2} + h(\sigma_x \frac{\partial^2 W}{\partial x^2} + \sigma_y \frac{\partial^2 W}{\partial y^2} + 2\sigma_{xy} \frac{\partial^2 W}{\partial x \partial y}) = 0 \quad (2.11)$$

$$\frac{\partial^2 M_1}{\partial x^2} + 2 \frac{\partial^2 M_{12}}{\partial x \partial y} + \frac{\partial^2 M_2}{\partial y^2} + m \frac{\partial^2 W}{\partial t^2} = q_2(x, y, t) \quad (2.12)$$

Здесь W - прогиб пластины, M_1, M_{12}, M_2 – изгибающие и крутящие моменты, $m = \gamma h / g$, γ - вес единицы объема, g – ускорение притяжения пластины, h – толщина.

Соотношение для M_1, M_{12}, M_2 в случаях, когда пластины являются изотропными, ортотропными и анизотропными, приведены в [4].

Подставляя в (2.10),(2.11) соотношения M_1, M_{12}, M_2 в случае, когда пластина является изотропной, ортотропной или в других случаях анизотропии, можно получить соответствующие уравнения. Эти уравнения приведены во многих руководствах по теории упругости. Поэтому нет необходимости здесь приводить их вид.

К уравнению (2.10),(2.11) добавляются краевые условия, а к уравнению (2.12) – и краевые и начальные условия.

Ниже приведены виды частот встречающихся краевых условий [1]:

а) жестко-зашемленный край

$$W|_{\Gamma=0}, \frac{\partial W}{\partial n}|_{\Gamma=0} = 0,$$

б) свободно-опертый край

$$W|_{\Gamma=0}, M_n|_{\Gamma=0} = (M_1 \cos^2 \alpha + M_{12} \cos \alpha \sin \alpha + M_2 \sin^2 \alpha)|_{\Gamma=0} = 0,$$

где $\alpha = (n, \hat{Ox})$ - угол между нормальным и Ox, Oy , $\beta = (n, \hat{Oy})$;

в) свободный край

$$M_n|_{\Gamma=0}, (\theta_n + \frac{\partial}{\partial S} M_{nr})|_{\Gamma=0} = 0,$$

где $\theta_n = M_1 \cos \alpha + M_2 \cos \beta$, $M_r = (M_2 - M_1) \cos \alpha \cos \beta + M_{12}(\cos^2 \alpha - \cos^2 \beta)$.

Кроме того, здесь возможны комбинации этих граничных условий в зависимости от способов закрепления краев пластин.

Начальные условия для уравнений (2.10) имеют вид

$$W(x, y, t)|_{t=t_0} = W_0(x, y), \quad \dot{W}(x, y, t)|_{t=t_0} = \dot{W}_0(x, y).$$

Формирование матриц для решения указанных выше задач осуществляется при помощи комбинации методов R – функции В.Л.Рвачева [126] и Бубнова-Галеркина [28].

Следует отметить, что [48] непосредственное применение метода Бубнова-Галеркина к решению уравнений (2.10),(2.11),(2.12) приводит к вычислительным трудностям. В данной работе в дальнейшем при формировании элементов матрицы размещающих уравнений будем использовать способ, предложенный в работе [48].

Здесь применение метода R – функций связано с построением координатных последовательностей, удовлетворяющих краевым условиям без каких-либо аппроксимаций.

Координатные последовательности, удовлетворяющие краевым условиям, представим в виде разложения

$$W = \sum_{i=1}^n T_i(t) B(\omega, \phi_i) = \sum_{i=1}^n T_i(t) W_i(x, y), \quad (2.13)$$

где $T_i(t)$ – неизвестные функции времени, подлежащие определению, $\{W_i(x, y)\}$ – полная, линейно-независимая система функций, которую будем строить при помощи метода R – функций В.Л.Рвачева [126].

Отметим, что в случае статики в представлении (2.12) вместо функции $T_i(t)$ будут неизвестные коэффициенты C_i .

Подставив (2.13) в (2.10)-(2.12) и выполняя обычно процедуры Бубнова-Галеркина, получим следующие уравнения:

$$AC=B, \quad (2.14)$$

$$A-\lambda B=0, \quad (2.15)$$

$$\mu \ddot{T} + AT = F, \quad (2.16)$$

$$T(t_0) = T_0, \quad \dot{T}(t_0) = \dot{T}_0,$$

где

$$A = \{a_{ij}\} = \frac{1}{S} \left\{ \iint_{\Omega} f_{ij} d\Omega \right\}, \quad b = \{b_{ij}\} = \frac{1}{S} \left\{ \iint_{\Omega} q_1 W_i d\Omega \right\},$$

$$B = \{b_{ij}\} = \frac{1}{S} \left\{ \iint_{\Omega} \varphi_{ij} d\Omega \right\}, \quad F = \{f_i\} = \frac{1}{S} \left\{ \iint_{\Omega} q_2 W_i d\Omega \right\},$$

$$\mu = \{m_{ij}\} = \frac{1}{S} \left\{ \iint_{\Omega} m W_i W_j d\Omega \right\}, \quad T(0) = \mu^{-1} T_1(t_0), \quad \dot{T}(0) = \mu^{-1} T_2(t_0)$$

$$T_1(t_0) = \frac{1}{S} \left\{ \iint_{\Omega} W_0 W_i d\Omega \right\}, \quad T_2(t_0) = \frac{1}{S} \left\{ \iint_{\Omega} \dot{W}_0 W_i d\Omega \right\},$$

$$\varphi_{ij} = \left(\sigma_x \frac{\partial^2 W_j}{\partial x^2} + \sigma_{xy} \frac{\partial^2 W_j}{\partial x \partial y} + \sigma_y \frac{\partial^2 W_j}{\partial y^2} \right) W_i,$$

$$f_{ij} = \left(\frac{\partial^2 W_i}{\partial x^2} + \nu \frac{\partial^2 W_i}{\partial y^2} \right) \frac{\partial^2 W_j}{\partial x^2} + \left(\frac{\partial^2 W_i}{\partial y^2} + \nu \frac{\partial^2 W_i}{\partial x^2} \right) \frac{\partial^2 W_j}{\partial y^2} + \frac{\partial^2 W_i}{\partial x \partial y} \cdot \frac{\partial^2 W_j}{\partial x \partial y}.$$

Для решения системы уравнений (2.14) применяются метод исключения Гаусса или метод наименьших квадратов и другие методы в зависимости от свойств матрицы. Для определения критической нагрузки применяется QL – метод [28].

Решение уравнения (2.13) при условии (2.16) находим различными численными методами: например, методом центральных разностей или методом Ньюмарка [30,93,107,127,130,142,154,165], или методом квадратурных сумм [48] и др.

Следует отметить, что при формировании матриц вычисления значений последовательности координатных функций и их производных n-го порядка осуществляются при помощи картежных операций [48]. Здесь значения интегралов вычисляются по n- точечной формуле Гаусса [17].

Описанный выше вычислительный алгоритм позволяет оптимизировать пластинчатых конструкции как постоянной, так и переменной толщины.

Таким образом, вычислительный алгоритм оптимизации пластинчатых конструкций состоит из следующих этапов:

1. Формирование целевой функции.
2. Формирование функциональных ограничений.
3. Формирование ограничений на параметры.
4. Прямой расчет.
5. Проверка выполнения условий прочности, жесткости, устойчивости и т.д.

В свою очередь, этап прямого расчета состоит из:

- построения последовательности координатных функций, удовлетворяющих граничным условиям поставленной задачи;

- формирования элементов матрицы разрешающего уравнения;

-решения разрешающего уравнения.

Следует отметить, разрешающие уравнение могут быть алгебраических, дифференциальных уравнений, в зависимости от рассмотренных задач в статической или динамической постановке.

2.6. Математические модели оптимизации арок, цилиндрических и конических оболочек переменной толщины и алгоритмы их разрешения.

Весовая оптимизация инженерных конструкций предусматривает минимизацию веса этих конструкций при воздействии на них заданных систем внешних сил с соблюдением условий сохранения необходимой прочности, устойчивости, жесткости конструкций.

Задача в общем виде может быть записана так:

$$\left. \begin{aligned} F(x) = G &\rightarrow \min; \\ G_{\max} &\leq [\sigma]; \\ P_{\max} &\leq P_{кр}; \\ U_{\max} &\leq [U], \end{aligned} \right\} \quad (2.19)$$

где G – вес конструкции;

σ_{\max} – максимальные напряжения в конструкции;

$[\sigma]$ - допускаемые напряжения;

P_{\max} – максимальная сжимающая сила;

$P_{кр}$ – критическая сила;

U_{\max} – максимальные перемещения в конструкции;

$[U]$ - допускаемые перемещения.

Проверка ограничений (2.19) возможна лишь после решения системы дифференциальных уравнений равновесия или движения рассматриваемых конструкций с соответствующими начальными или граничными условиями. Приведем эти системы для цилиндрических и конических оболочек.

2.6.1. Уравнения равновесия арок, цилиндрических и осесимметричных конических оболочек переменной толщины

Рассмотрим цилиндрическую оболочку переменной толщины, имеющую постоянный радиус R срединной поверхности, прямоугольную в плане и заданную в координатах α, β, \bar{r} (рис 2.3)

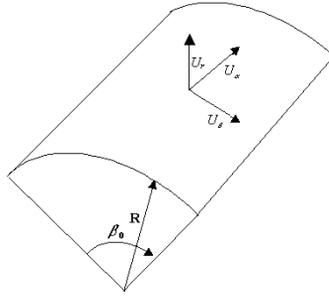


Рис. 2.3.

В этом случае:

$$R - \frac{h}{2} \leq \bar{r} \leq R + \frac{h}{2}; 0 \leq \beta \leq \beta_0; 0 \leq \alpha \leq \alpha_0$$

Компоненты перемещений показаны на рис. 2.3. U_z – направлено по внешней нормали к срединной поверхности; U_β - по касательной; U_α - по образующей.

Компоненты деформаций с учётом гипотезы Кирхгофа- Лява зависит от перемещений срединной поверхности U, V и W следующим образом

$$l_{\alpha\alpha} = \frac{1}{R} \left(\frac{\partial U}{\partial \alpha} - (r-1) \frac{\partial^2 W}{\partial \alpha^2} \right);$$

$$l_{\beta\beta} = \frac{1}{R} \left(\frac{\partial v}{\partial \beta} - \left(1 - \frac{1}{r}\right) \frac{\partial^2 W}{\partial \beta^2} + \frac{1}{r} W \right);$$

$$l_{\alpha\beta} = \frac{1}{R} \left(r \frac{\partial v}{\partial \alpha} + \frac{1}{r} \frac{\partial U}{\partial \beta} - \left(r - \frac{1}{r} \right) \frac{\partial^2 W}{\partial \alpha \partial \beta} \right);$$

(2.20)

$$z \frac{\partial e}{\partial e}$$

$$r = \frac{\bar{r}}{R}$$

Напряжения в оболочке определяются по формулам:

$$\sigma_{\alpha\alpha} = \frac{E}{1-\nu^2} (l_{\alpha\alpha} + \nu l_{\beta\beta});$$

$$\sigma_{\beta\beta} = \frac{E}{1-\nu^2} (l_{\beta\beta} + \nu l_{\alpha\alpha});$$

$$\tau_{\alpha\beta} = \frac{E}{2(1+\nu)} l_{\alpha\beta}.$$

(2.21)

Здесь E - модуль упругости; ν - коэффициент Пуассона.

Уравнение равновесия выводятся из вариационного принципа минимума энергии. В общем виде вариационное уравнение равновесия цилиндрической оболочки имеет вид:

$$\begin{aligned}
& \sum_{\kappa=1}^3 \int_V \sum_{i=1}^3 L_{ki}(U_i) \delta U_k dv + \sum_{\kappa=1}^3 \int_r \int_{\beta} \sum_{i=1}^3 F_{ki}^1(U_i) \Big|_0^{\alpha_0} \delta U_k dr d\beta + \\
& \sum_{\kappa=1}^3 \int_r \int_{\alpha} \sum_{i=1}^3 F_{ki}^2(U_i) \Big|_0^{\beta_0} \delta U_k dr d\alpha + \int_r \int_{\beta} \sum_{i=1}^3 \Gamma_{\alpha i}^2(U_i) \Big|_0^{\alpha_0} \delta \frac{\partial W}{\partial \alpha} \times \\
& dr d\beta + \int_r \int_{\alpha} \sum_{i=1}^3 \Gamma_{\beta i}^2(U_i) \Big|_0^{\beta_0} \delta \frac{\partial W}{\partial \beta} dr d\alpha = \sum_{i=1}^3 \int_{\alpha} \int_{\beta} X_i \delta U_i d\alpha d\beta,
\end{aligned} \quad (2.22)$$

где L_{ki} , Γ_{ki}^j – линейные дифференциальные операторы.

Полностью уравнения равновесия цилиндрической оболочки приведены в [28].

Уравнения равновесия арки получаются из (2.22) как частный случай при $U=0$ и $V=V(\beta)$, $W=W(\beta)$, т.е. при условии независимости перемещений V и W от координаты α .

Для осесимметричной конической оболочки, заданной в координатах α, γ, β (рис. 2.4) компоненты деформаций в зависимости от перемещений срединной поверхности V и W имеют вид:

$$\begin{aligned}
l_{33} &= \frac{dU}{d\alpha} - \gamma \frac{d^2 W}{d\alpha^2}; \\
l_{22} &= \frac{1}{\alpha \sin \Theta + \gamma \cos \Theta} \left[W \cos \Theta - \left(U - \gamma \frac{dW}{d\alpha} \right) \sin \Theta \right]
\end{aligned} \quad (2.23)$$

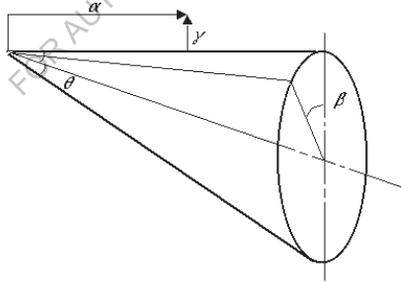


Рис. 2.4.

Напряжения определяются по формулам [34]:

$$\begin{aligned}
\sigma_{33} &= (\lambda + 2\mu) l_{33} + \lambda l_{22}; \\
\sigma_{22} &= \lambda \cdot l_{33} + (\lambda + 2\mu) l_{22}, \\
z \partial e & \\
\lambda &= \frac{E\nu}{(1-\nu)(1+2\nu)}; \mu = \frac{E}{2(1+\nu)}
\end{aligned} \quad (2.24)$$

Уравнения равновесия конических оболочек полностью приведены в [28,156]. Выпишем эти уравнения для осесимметричной конической оболочки переменной толщины.

$$L(X) = A^0 X^{IV} + B^0 X^{III} + C^0 X^{II} + D^0 X^I + E^0 X + Q^0 = O \quad (2.25)$$

$$A^0 = \begin{pmatrix} b_{29} & O \\ O & O \end{pmatrix}; B^0 = \begin{pmatrix} b_{28} & b_{24} \\ b_{17} & O \end{pmatrix}; C^0 = \begin{pmatrix} b_{27} & b_{23} \\ b_{16} & b_{13} \end{pmatrix};$$

$$D^0 = \begin{pmatrix} b_{26} & b_{22} \\ b_{15} & b_{12} \end{pmatrix}; E^0 = \begin{pmatrix} b_{25} & b_{21} \\ b_{14} & b_{11} \end{pmatrix}; Q^0 = \begin{pmatrix} \bar{X}_2 \\ \bar{X}_1 \end{pmatrix}; X = \begin{pmatrix} W \\ U \end{pmatrix};$$

$$b_{11} = [(\lambda + 2\mu)h/\alpha - \lambda h'] \sin \Theta;$$

$$b_{12} = -(\lambda + 2\mu) \cdot (n + h' \alpha) \sin \Theta;$$

$$b_{13} = -(\lambda + 2\mu) \alpha h \sin \Theta;$$

$$b_{14} = \left[(\lambda + 2\mu) \frac{h}{\alpha} - \lambda h' \right] \cos \Theta;$$

$$b_{15} = \left[\frac{(\lambda + 2\mu) h^3}{\alpha^2} - \lambda h \right] \cos \Theta;$$

$$b_{16} = (\lambda + 2\mu) \frac{h^2 h'}{4} \cos \Theta; b_{17} = (\lambda + 2\mu) \frac{h^3}{12} \cos \Theta;$$

$$b_{21} = (\lambda + 2\mu) \frac{h}{\alpha} \left(1 + \frac{h^2}{6\alpha^2} - \frac{hh'}{4\alpha} \right) \cos \Theta;$$

$$b_{22} = - \left[(\lambda + 2\mu) \left(\frac{hh'^2}{2} + \frac{h^2 h''}{4} + \frac{h^3}{12\alpha^2} \right) - \lambda h \right] \cos \Theta;$$

$$b_{23} = -(\lambda + 2\mu) \frac{h^2 h'}{2} \cos \Theta; b_{24} = -(\lambda + 2\mu) \frac{h^3}{12} \cos \Theta;$$

$$b_{25} = (\lambda + 2\mu) \frac{\cos^2 \Theta}{\sin \Theta} \frac{h}{\alpha} \left(1 + \frac{h^2}{6\alpha^2} - \frac{hh'}{4\alpha} \right);$$

$$b_{26} = - \left[(\lambda + 2\mu) \frac{1}{\alpha} \left(\frac{h^2 h'}{4} - \frac{h^3}{12\alpha} \right) - \lambda \frac{h h'^2}{2} \right] \sin \Theta;$$

$$b_{27} = \left[(\lambda + 2\mu) \left(\frac{h^2 h'}{2} + \alpha \left(\frac{hh'^2}{2} + \frac{h^2 h''}{4} - \frac{1}{\alpha^2} \frac{h^3}{12} \right) \right) + \lambda \frac{h^2 h'}{4} \right] \cdot \sin \Theta;$$

$$b_{28} = (\lambda + 2\mu) \left(\frac{h^3}{6} + \alpha \frac{h^2 h'}{2} \right) \sin \Theta;$$

$$b_{29} = (\lambda + 2\mu) \alpha \frac{h^3}{12} \sin \Theta;$$

$\bar{X}_{1,2}$ - поверхностные силы.

Решив уравнения (2.22, 2.25) при помощи алгоритмов, описанных ниже, найдем искомые функции – перемещения U, V, W, зная которые, по формулам (2.21., 2.24) определим напряжения в конструкции.

2.6.2. Алгоритмы решения прямой задачи расчета арок, цилиндрических и конических оболочек

Ввиду того, что оптимизация конструкций методами случайного поиска предполагает многократное решение прямой задачи расчета конструкций, к алгоритму прямого расчета следует предъявлять дополнительные требования о скорости действия, наряду с достижением требуемой точности. С другой стороны, выбор оптимального алгоритма составляет самостоятельную проблему, до настоящего времени еще не решенную до конца. Поэтому мы ограничились сравнением 2-х наиболее распространенных алгоритмов решения краевых задач (2.22, 2.25) – решение методом Ритца и конечно-разностным методом.

При решении задач расчета арок и цилиндрических оболочек методом Ритца [48] искомые перемещения U_i задаем в виде рядов:

$$\left. \begin{aligned} U &= \sum_n \sum_m X_n(\alpha) V_m(\beta) A_{nm}; \\ V &= \sum_n \sum_m Z_n(\alpha) Q_m(\beta) B_{nm}; \\ W &= \sum_n \sum_m \varphi_n(\alpha) \psi_m(\beta) C_{nm}. \end{aligned} \right\} \quad (2.26)$$

Варьируя по произвольным постоянным, получаем:

$$\left. \begin{aligned} \delta U &= \sum_n \sum_m X_n(\alpha) V_m(\beta) \delta A_{nm}; \\ \delta V &= \sum_n \sum_m Z_n(\alpha) Q_m(\beta) \delta B_{nm}; \\ \delta W &= \sum_n \sum_m \varphi_n(\alpha) \psi_m(\beta) \delta C_{nm}. \end{aligned} \right\} \quad (2.27)$$

Подставив (2.28) и (2.29) в (2.24), получим систему линейных алгебраических уравнений

$$\sum_m \sum_b A_{nm} A_{nmkt}^i + B_{nm} B_{nmkt}^i + C_{nm} C_{nmkt}^i = D_{kt}^i \quad (i = 1, \bar{3}) \quad (2.28)$$

где:

$$A_{nmkt}^1 = \int_V L_{11}(X_n, Y_m) X_k Y_t dV + \int_r \int_\beta \Gamma_{11}^1(X_n, Y_m) X_k Y_t \Big|_0^{\alpha_0} dr d\beta + \quad (2.29)$$

$$\int_r \int_\alpha \Gamma_{11}^2(X_n, Y_m) X_k Y_t \Big|_0^{\beta_0} dr d\alpha;$$

$$A_{nmkt}^2 = \int_V L_{21}(X_n, Y_m) Z_k Q_t dV + \int_r \int_\beta \Gamma_{21}^1(X_n, Y_m) Z_k Q_t \Big|_0^{\alpha_0} dr d\beta + \quad (2.30)$$

$$\int_r \int_\alpha \Gamma_{21}^2(X_n, Y_m) Z_k Q_t \Big|_0^{\beta_0} dr d\alpha;$$

$$A_{nmkt}^3 = \int_V L_{31}(X_n, Y_m) \varphi_k \psi_t dV + \int_r \int_\beta \left[\Gamma_{31}^1(X_n, Y_m) \varphi_k \psi_t + \Gamma_{41}^1(X_n, Y_m) \varphi_k' \psi_t \right] \Big|_0^{\alpha_0} dr d\beta +$$

$$\int_r \int_\alpha \left[\Gamma_{31}^2(X_n, Y_m) \varphi_k \psi_t + \Gamma_{41}^2(X_n, Y_m) \varphi_k' \psi_t \right] \Big|_0^{\beta_0} dr d\alpha;$$

(2.31)

$$B_{nmkt}^1 = \int_V L_{12}(Z_n, Q_m) X_k V_t dV + \int_r \int_\beta \Gamma_{12}^1(Z_n, Q_m) X_k V_t \Big|_0^{\alpha_0} dr d\beta + \quad (2.32)$$

$$\int_r \int_\alpha \Gamma_{22}^2(Z_n, Q_m) X_k V_t \Big|_0^{\beta_0} dr d\alpha;$$

$$B_{nmkt}^2 = \int_V L_{22}(Z_n, Q_m) Z_k Q_t dV + \int_r \int_\beta \Gamma_{22}^1(Z_n, Q_m) Z_k Q_t \Big|_0^{\alpha_0} dr d\beta + \quad (2.33)$$

$$\int_r \int_\alpha \Gamma_{22}^2(Z_n, Q_m) Z_k Q_t \Big|_0^{\beta_0} dr d\alpha;$$

$$B_{nmkt}^3 = \int_V L_{32}(Z_n, Q_m) \varphi_k \psi_t dV + \int_r \int_\beta \left[\Gamma_{32}^1(Z_n, Q_m) \varphi_k \psi_t \Gamma_{42}^1(Z_n, Q_m) \varphi'_k \psi'_t \right] \Big|_0^{\alpha_0} dr d\beta +$$

$$\int_r \int_\alpha \left[\Gamma_{32}^2(Z_n, Q_m) \varphi_k \psi_t + \Gamma_{42}^2(Z_n, Q_m) \varphi_k \psi'_t \right] \Big|_0^{\beta_0} dr d\alpha; \quad (2.34)$$

$$C_{nmkt}^1 = \int_V L_{13}(\varphi_n, \psi_m) X_k V_t dV + \int_r \int_\beta \Gamma_{13}^1(\varphi_n, \psi_m) X_k V_t \Big|_0^{\alpha_0} dr d\beta + \quad (2.35)$$

$$\int_r \int_\alpha \Gamma_{13}^2(\varphi_n, \psi_m) X_k V_t \Big|_0^{\beta_0} dr d\alpha;$$

$$C_{nmkt}^2 = \int_V L_{23}(\varphi_n, \psi_m) Z_n Q_m dV + \int_r \int_\beta \Gamma_{23}^1(\varphi_n, \psi_m) Z_n Q_m \Big|_0^{\alpha_0} dr d\beta + \quad (2.36)$$

$$\int_r \int_\alpha \Gamma_{23}^2(\varphi_n, \psi_m) Z_n Q_m \Big|_0^{\beta_0} dr d\alpha;$$

$$C_{nmkt}^3 = \int_V L_{33}(\varphi_n, \psi_m) \varphi_k \psi_t dV + \int_r \int_\beta \left[\Gamma_{33}^1(\varphi_n, \psi_m) \varphi_k \psi_t + \Gamma_{43}^1(\varphi_n, \psi_m) \varphi'_k \psi'_t \right] \quad (2.37)$$

$$\Big|_0^{\alpha_0} \left[dr d\beta + \int_r \int_\alpha \Gamma_{33}^2(\varphi_n, \psi_m) \varphi_k \psi_t + \Gamma_{43}^2(\varphi_n, \psi_m) \varphi_k \psi'_t \right] \Big|_0^{\beta_0} dr d\alpha;$$

$$D_{kt}^1 = \int_s X_1 X_k V_t ds; \quad (2.38)$$

$$D_{kt}^2 = \int_s X_2 X_k Q_t ds; \quad (2.39)$$

$$D_{kt}^3 = \int_s X_3 \varphi_k \psi_t ds; \quad (2.40)$$

Для арки перемещения V, W задаём в виде:

$$\left. \begin{aligned} V &= \sum_n^n Q_n(\beta) A_n; \\ W &= \sum_n^n \varphi_n(\beta) B_n. \end{aligned} \right\} \quad (2.41)$$

и система линейных алгебраических уравнений при этом имеет вид:

$$\sum_n A_n A_{nm}^i + B_n B_{nm}^i = C_m^i \quad (i=1,2) \quad (2.42)$$

Системы (2.16 и 2.41) решаем методом Гаусса. Определив коэффициенты (A_{nm}, B_{nm}, C_{nm}), по формулам (2.14, 2.42) определим значения перемещений V, W , зная которые по формулам (2.9) найдем напряжения σ_{ij} ;

С точки зрения экономии машинного времени более эффективным для решения дифференциальных уравнений с переменными коэффициентами является конечно-разностный метод [17, 41, 42, 62-67, 107].

Рассмотрим этот метод на примере решения задачи расчета осесимметричных конических оболочек [157].

Введем сетку $\omega_n = (x_m = mh, m = \overline{0, N})$ с шагом $h = \frac{1}{N}$ на отрезке $0 \leq x \leq 1, Y_m = \{W_m, U_m\}$ -сеточные функции в области ω_n .

Пользуясь центральными разностными формулами, аппроксимируем производные с точностью второго порядка

$$\begin{aligned} Y'_m &= \frac{N}{2}(Y_{m+1} - Y_{m-1}); \\ Y''_m &= N^2(Y_{m+1} - 2Y_m + Y_{m-1}); \\ Y'''_m &= \frac{N^3}{2}(Y_{m+2} - 2Y_{m+1} + 2Y_{m-1} - Y_{m-2}); \\ Y^{IV}_m &= N^4(Y_{m+2} - 4Y_{m+1} + 6Y_m - 4Y_{m-1} + Y_{m-2}). \end{aligned} \quad (2.43)$$

вместо уравнений (2.13) получим следующую систему алгебраических уравнений:

$$\bar{A}_m Y_{m-2} + \bar{B}_m Y_{m-1} + \bar{C}_m Y_m + \bar{D}_m Y_{m+1} + \bar{E}_m Y_{m+2} + Q_m = 0 \quad (2.44)$$

Здесь

$$\begin{aligned} \bar{A}_m &= A^\circ N^4 - B^\circ \frac{N^3}{2}; \\ \bar{B}_m &= -4A^\circ N^4 - B^\circ N^3 + C^\circ N^2 - D^\circ \frac{N}{2}; \\ \bar{C}_m &= 6A^\circ N^4 - 2C^\circ N^2 + E^\circ; \\ \bar{D}_m &= -4A^\circ N^4 - B^\circ N^3 + C^\circ N^2 - D^\circ \frac{N}{2}; \\ \bar{E}_m &= A^\circ N^4 + B^\circ \frac{N^3}{2}. \end{aligned}$$

Для $m^{\text{го}}$ уравнения (2.44) имеют место следующие зависимости:

$$\begin{aligned} Y_m &= f_m - H_m Y_{m+1} - F_m Y_{m+2}; \\ f_m &= G_m (Q_m - G'_m f_{m-1} - \bar{A}_m f_{m-2}); \\ H_m &= G_m (\bar{D}_m - G'_m F_{m-1}); F_m = G_m \bar{E}_m; \\ G_m &= [\bar{C}_m - \bar{A}_m F_{m-1} - G'_m H_{m-1}]^{-1}; G'_m = \bar{B}_m - \bar{A}_m H_{m-2}. \end{aligned} \quad (2.45)$$

применяя формулы (2.35) для $1^{\text{го}}, 2, \dots, N^{\text{го}}$ уравнения (2.44) (в случае свободного края) и $N-1^{\text{го}}$ уравнения (2.44) (в остальных случаях) находим значения последнего неизвестного вектора (Y_N, Y_{N-1}). Затем, пользуясь формулой (2.44), найдем значения сеточных функций Y_m во всех остальных узлах сетки. После чего по формулам (2.12) можно определить напряжения в осесимметричной конической оболочке.

Списанный алгоритм полностью подходит к расчету арки. Отличие будет лишь в значениях коэффициентов в уравнениях (2.44).

При сравнении методов Ритца и конечно-разностного применительно к расчету арки выяснилось, как это уже отмечалось в работах [49,108,166], что метод имеет преимущество в скорости.

В связи с этим для решения прямой задачи статистического расчета арок и осесимметричных конических оболочек нами применен конечно-разностный метод. Для расчета открытой цилиндрической оболочки применен метод Ритца, поскольку метод конечных разностей приводит в этом случае к слишком громоздкой системе алгебраических уравнений, что вызывает серьезные трудности как при решении прямой задачи, так и при оптимизации открытых цилиндрических оболочек.

FOR AUTHOR USE ONLY

Глава 3. РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ СЛУЧАЙНОГО ПОИСКА ДЛЯ АЛГОРИТМИЧЕСКОЙ СИСТЕМЫ ОПТИМИЗАЦИИ (АСО)

3.1. Формирование основных требований к алгоритмам АСО

Назначение АСО как системы, инвариантной по отношению к классу оптимизируемых объектов, предполагает решение ею разнообразных задач математического программирования. Это обстоятельство, естественно, выдвигает определенные требования к алгоритмам, включаемым в систему.

Одним из основных требований к алгоритмам является их универсальность. Алгоритмы, способные без перестройки решать максимально широкий класс задач обладают существенными привлекательными качествами в глазах исследователя, такими как:

- упрощение проблемы выбора алгоритма для решения конкретной задачи;
- уменьшение затрат времени на предварительный анализ задачи, т.е. определения выпуклости, дифференцируемости целевой функции и ограничений и т.д., что часто представляет собой трудно разрешимую проблему;
- сравнительно небольшое число алгоритмов, необходимое для включения в оптимизирующую систему, а следовательно и уменьшение количества и упрощение сервисных модулей, обслуживающих систему.

Однако, универсальность алгоритма имеет свои негативные стороны. Отметим следующие:

- невысокая скорость сходимости в ряде случаев;
- усложненная внутренняя структура алгоритма;
- увеличение объема памяти, занимаемого алгоритмом.

Рассматривая совокупность положительных и отрицательных сторон универсальных алгоритмов, можно прийти к выводу, что:

- а) в системах типа АСО необходимо наличие таких алгоритмов;
- б) необходимо наличие алгоритмов, менее универсальных, но более быстродействующих, простых и занимающих меньшую оперативную память ЭВМ, следовательно, способных решать задачи большей размерности.

Наиболее универсальны алгоритмы случайного поиска глобального экстремума. Однако, наиболее простые варианты этих алгоритмов (такие как случайный перебор, случайный перебор с адаптацией) имеют невысокую скорость сходимости, увеличение которой ведет к усложнению алгоритмов (включение в алгоритмы самообучения, адаптации, различных эвристических приемов), а следовательно и к увеличению объема требуемой оперативной памяти.

Алгоритмы случайного поиска локального экстремума обладают также большей универсальностью, но предназначены для решения задач с единственным экстремумом. Они более компактны, имеют, в основном, достаточную скорость сходимости и надежность нахождения экстремума.

Простота алгоритма является желательной для системы во-первых: с точки зрения их программной реализации, во-вторых: простота алгоритма подразумевает небольшой объем памяти, занимаемой алгоритмом, в-третьих: ведет к уменьшению времени работы самого алгоритма на проведение каждой итерации, что также имеет немаловажную роль в решении задач оптимизации.

Эффективность алгоритма – это естественное требование к нему инженеров и исследователей. Это требование может рассматриваться с двух позиций. Эффективность – максимальное быстродействие алгоритма при решении определенного класса задач, с другой стороны эффективность – достаточная надежность получения решения задачи с заданной точностью.

Наиболее широко известны своей простотой и эффективностью регулярные алгоритмы, такие как градиентные, наискорейшего спуска, сопряженных градиентов Ньютона. Они достаточно полно исследованы, их эффективность при решении определенных классов задач математически доказана, определены оценки скоростей сходимости.

Эффективность алгоритмов случайного поиска более трудно поддается аналитическому исследованию, поэтому большее внимание уделяется численным экспериментам в определении эффективности, достаточно высокая степень которой ими подтверждается, а также широкой практикой решения реальных оптимизационных задач [6,9,17,30,38,39,43, 44, 48,49,50,68,107,110,115,119,120,124, 128, 143,165].

Наиболее широко изучены и применяются алгоритмы безусловной оптимизации. Однако, для работы оптимизирующей системы одних таких алгоритмов недостаточно, т.к. условная оптимизация имеет большее значение особенно при оптимизации реальных объектов, когда в подавляющем большинстве случаев появляются те или иные ограничения, резко усложняющие оптимизационные задачи. Поэтому необходимость включения в состав системы алгоритмов условной оптимизации является очевидной.

Существует множество регулярных и, особенно, статистических алгоритмов условной оптимизации, причем стоит отметить, что последние более просты и гибки при «борьбе» с ограничениями. В п. 4.2 приведен ряд регулярных алгоритмов условной оптимизации, способных, при определенных условиях, решать задачи с функциональными ограничениями. Однако, выполнение этих условий не всегда возможно, особенно если ограничения заданы не в явном виде, а алгоритмически. В этом случае применимы лишь алгоритмы случайного поиска.

Известный и широко применяющийся метод «штрафных функций» решения задач условной оптимизации также более приемлем для методов случайного поиска. Этот метод имеет некоторые негативные стороны. Во-первых, достаточно сложно находить экстремумы с высокой точностью, во-вторых, применяя этот метод, сводящий условную оптимизацию к безусловной, приходится вычислять целевую функцию и ограничения одинаковое количество раз, т.к. они образуют единую новую целевую функцию. Это обстоятельство не всегда желательно, т.к. часто, и особенно при оптимизации инженерных конструкций, проверка ограничений требует на несколько порядков больше вычислений, чем для целевой функции. Таким образом, выявляется необходимость дифференцированного подхода к вычислению целевой функции и ограничений с целью ускорения сходимости алгоритмов. Для этого требуется разработка алгоритмов, специально направленных на уменьшение количества проверки функциональных ограничений при решении оптимизационных задач. Описание таких алгоритмов (как локального, так и глобального поиска) дано ниже.

Одним из путей повышения эффективности алгоритмов (в основном случайного поиска) является введение в них элементов эвристики, позволяющих снизить непроизводительные затраты на решение задачи. Эвристические алгоритмы более полно отражают опыт, интуицию, «здоровый смысл» разработчика и проблемную ориентированность алгоритма. Так при весовой оптимизации инженерных конструкций, подверженных влиянию внешних нагрузок, минимум веса конструкции будет находиться на границе прочности, жесткости или устойчивости. Это дает возможность путем введения некоторых масштабных коэффициентов резко ускорить процесс поиска. То же можно сказать и о других задачах условной оптимизации, если априорно известно, что экстремум находится на границе.

При решении многоэкстремальных задач, очевидно, правы следующие утверждения:

Утверждение 1. Для решения достаточно определение координат каждого локального минимума только один раз.

Утверждение 2. Для решения достаточно определение координат с заданной точностью ϵ лишь наиболее «глубоких» минимумов, из которых определяется глобальный.

Реализация приведенных утверждений путем введения в алгоритм глобального поиска элементов эвристики приводит к существенному ускорению сходимости алгоритма. Ясно, что многократное нахождение одного и того же минимума не дает новой информации и не ведет к повышению вероятности нахождения глобального минимума, а влечет лишь непроизводительные затраты. Этим недостатком страдают многие алгоритмы глобального поиска. Так же нет обычно никакой необходимости тратить время на уточненный поиск «мелких» неперспективных минимумов, т.е. тех результатов, которые не играют никакой роли для решения задачи, т.е. определения глобального минимума. Приведенные ниже результаты сравнения разработанных автором алгоритмов, включающих перечисленные элементы эвристики, с известными и достаточно эффективными алгоритмами на ряде задач подтверждают правоту выдвинутых утверждений и необходимость включения эвристики в поисковые алгоритмы.

Следует отметить необходимость включения в оптимизирующую систему алгоритмов, способных решать оптимизационные задачи при наличии дискретно изменяющихся параметров. Сложность решения подобных задач не снимает проблему дискретного программирования с повесткой дня, а предполагает развитие исследований в этой области. В п. 3.1. приведены некоторые подходы к решению дискретных задач, проанализированы их возможности. Требования к простоте и универсальности алгоритмов, а также, на наш взгляд, к единству внутренней идеологии их, предопределили разработку «дискретного» алгоритма. Концепции, предложенные в разработанных автором алгоритмах случайного поиска оказались достаточно эффективными и при решении задач, когда все или часть параметров являются целочисленными. Необходимая модернизация алгоритма не затронула его внутренней структуры; а лишь несколько модифицировала, и, вместе с тем, позволила решать задачи дискретного программирования с достаточной степенью надежностью.

Резюмируя вышесказанное, можно сформулировать следующие основные требования к алгоритмам, включаемым в оптимизирующую систему:

1. универсальность;
2. эффективность;
3. простота;
4. минимальный объем занимаемой памяти;
5. глобальность поиска;
6. способность решения задач условной оптимизации;
7. введение в алгоритмы элементов эвристики;
8. способность работы с дискретными параметрами.

Не обязательно, чтобы каждый алгоритм отвечал все перечисленным требованиям, но в совокупности алгоритмы, заложенные в систему должны обеспечить выполнение всего набора требований с целью обеспечения системе максимально широких возможностей.

3.2. Алгоритм поиска П-1 и П-2. Алгоритмы поиска П-1 и П-2 предназначены для оптимизации многопараметрических одноэкстремальных функций при любых ограничениях на функцию или на связь оптимизируемых параметров.

3.2.1. Алгоритм поиска П-1

1. **Область поиска.** Область, в которой производится минимизация целевой функции, может быть любой замкнутой областью, границы которой описываются удовлетворяющей ограничениям (1.1).

Наиболее распространенным случаем является тот, когда область поиска является n -мерным гиперпараллелепипедом:

$$a_i \leq x_i \leq b_i, \quad i = \overline{1, n}. \quad (3.1)$$

Поиск минимума в такой области производится наиболее просто ввиду простоты учета ограничений. Алгоритм поиска П-1, в силу линейности его тактики, наиболее удачно производит поиск именно в таких областях.

2. **Величина рабочего шага.** Величина рабочего шага задается самим исследователем для каждой задачи в зависимости от размеров области поиска и особенностей задачи. В процессе поиска минимума величина рабочего шага остается неизменной до тех пор, пока система не окажется в непосредственной близости от минимума, после чего производится уточнение координат минимума путем последовательного уменьшения величины рабочего шага вдвое. Такое уменьшение длины рабочего шага производится до тех пор, пока координаты локального минимума не будут найдены с заданной точностью ε .

3. **Определение начальной точки поиска.** Начальная точка поиска $X_0 \in D$ определяется процедурой Монте-Карло или задается.

4. **Организация поиска.** В поиске применен метод спуска в наилучшем направлении, о котором сообщалось в п. 1.1.1. Из исходной точки $X_K \in D$ делается m случайных проб $X_K + H\Xi_j \in D$, где $H = [h_1, h_2, \dots, h_n]$ – диагональная матрица с элементами h_i – модуль шага по i -той координате; $\Xi = (\xi_1, \xi_2, \dots, \xi_n)$ – случайный вектор с компонентами ξ_i – случайными числами, равномерно распределенными на отрезке $[-1, 1]$.

В каждой полученной точке $X_k + H\Xi_j$ определяется значение целевой функции и выбирается та проба $X_k + H\Xi^*$, которая сообщает функции качества максимальное уменьшение. Точки X_k и $X_k + H\Xi^*$ определяют направление спуска. В этом направлении происходит движение системы шагами $\Delta X = H\Xi^*$. После каждого шага происходит вычисление значения функции качества F_k и сравнение его со значениями после предыдущего шага F_{k-1} . Если $F_k \leq F_{k-1}$, то происходит дальнейший спуск, если же $F_k > F_{k-1}$, то происходит остановка системы, возвращение ее в последнее улучшенное состояние, из которого делаются серии m равномерно распределенных испытаний для определения дальнейшего направления поиска.

Таким образом, предлагаемая поисковая система является безинерционной, так как она мгновенно реагирует на отрицательный сигнал – ухудшение целевой функции.

Немедленная реакция на отрицательный сигнал – не единственная положительная черта алгоритма. То что он не нуждается в дополнительной настройке – также является его положительной чертой, которая уменьшает трудности работы с алгоритмом и делает его малозависимым от различных целевых функций. У инерционных алгоритмов быстрота реакции, а с нею быстрота поиска наилучшего направления связаны с величиной инерционности алгоритма, которая для различных задач должна быть различна. Эта дополнительная настройка «инерционных» алгоритмов несколько затрудняет работу с ними. Надо отметить, что алгоритмы с линейной тактикой хорошо работают с «пологими» функциями, когда последние близки к линейным. В случае сильно развитой нелинейности оптимизируемых функций и при значительной овражности их, эффективность алгоритмов с линейной тактикой снижается. Но большая простота и быстрота настройки таких алгоритмов по сравнению с инерционными алгоритмами с самообучением обеспечивают широкое практическое применение алгоритмов с линейной тактикой.

5. **Учет ограничений.** При учете ограничений нами выделено два случая: а) учет линейных ограничений типа (3.1); б) учет нелинейных ограничений (3.2) на параметры и учет ограничений на целевую функцию или на связь параметров.

В случае «а» учет ограничений производится наиболее просто, и учитываются они следующим образом:

$$x_i = \begin{cases} x_i & \text{при } a_i \leq x_i \leq b_i; \\ a_i(1 + K_{OT}) - K_{OT} \cdot x_i & \text{при } x_i \leq a_i; \\ b_i(1 + K_{OT}) - K_{OT} \cdot x_i & \text{при } x_i > b_i. \end{cases} \quad (3.2)$$

Здесь a_i и b_i соответственно нижняя и верхняя границы изменения i -того оптимизируемого параметра. K_{OT} – коэффициент отражения от границы. При $K_{OT}=0$ точка, нарушившая ограничение, возвращается на границу. При $K_{OT}>0$ точка «отражается» от стенки гиперпараллелепипеда вовнутрь области поиска. Коэффициент K_{OT} принимает значения $0 \leq K_{OT} \leq 1$. Он задается исследователем перед началом поиска для каждой конкретной задачи. Если «отраженные» точки не дают снижение минимизируемой функции, то из последней улучшенной точки делаются новые серии испытаний с уменьшением величины шага. Таким образом, осуществляется постепенный подход к границе. Для случая «б» такой алгоритм отражения построить не удается, а так как применение метода «штрафных функций» ведет к усложнению алгоритма и к смещению получаемого минимума относительно истинного, то поиск в случае сложных нелинейных ограничений мы построили следующим образом. Если точка нарушила границу, то система возвращается в последнее улучшенное состояние, не нарушающее ограничения. Из этого состояния делаются новые серии испытаний для нахождения «обхода» ограничения. Если такого направления найти не удастся, то происходит уменьшение длины рабочего шага, т.е. $\Delta X \cdot A_2$, где $A_2 < 1$ и т.д. При этом осуществляется постепенный подход к границе и минимум находится с заданной точностью.

6. **Условие окончания локального поиска.** Если при определении улучшающего минимизируемую функцию направления ни одна из проб не принесла удачи, то испытания повторяются снова. Если какое-то заданное количество подряд идущих серий испытаний не дали улучшения целевой функции, то происходит уменьшение длины рабочего шага вдвое и уточнение координат локального минимума, которое длится до выполнения условия:

$$K_H = K_0 \wedge \varepsilon > \Delta F^* \quad (3.3)$$

где: K_H – количество подряд идущих неудачных серий испытаний;

K_0 – коэффициент, характеризующий плотность просмотра окрестности минимума;

ε – заданная точность поиска; измеряется в единицах измерения функции цели;

ΔF^* – наибольшее приращение функции цели за неудачные серии поиска испытаний;

\wedge – знак логического умножения.

Коэффициент K_0 , как и ε , задается для каждой задачи.

Как видим, условие (3.3) прекращения локального поиска состоит из двух самостоятельных условий, соединенных знаком логического умножения. В виду того, что выбор направлений осуществляется случайным образом, т.е. при помощи датчиков псевдослучайных чисел, то прежде чем остановить поиск лучшего состояния системы, надо каким-то образом учитывать вероятность пропуска наилучшего направления. Эту вероятную сторону процесса поиска учитывает левая часть условия (3.3). Для увеличения

вероятности нахождения улучшенного состояния системы надо увеличить количество испытаний, проведенных из последней улучшенной точки.

Изменения плотности просмотра окрестности минимума можно добиться, изменяя константу K_0 . Увеличение вероятности нахождения лучшего состояния системы ведет к увеличению временных затрат на решение задачи, поэтому каких-либо общих рекомендаций на все случаи жизни дать, по видимому, нельзя. В каждом конкретном случае в зависимости от сложности задачи, степени ее важности, количество располагаемого машинного времени, отпущенного на решение данной задачи, исследователь сам задает значение константы K_0 .

Правая часть условия (3.3) характеризует точность нахождения координат локального минимума. Если левая часть условия (3.3) выполнена, а правая – нет, то происходит уменьшение модуля шага по каждой координате вдвое и переход к новым сериям испытаний.

При выполнении (3.3) последняя улучшенная точка считается локальным минимумом. Координаты локального минимума, а также значение функции качества в нем заносятся в память.

Таковы общие черты алгоритма П-1. На рис. 3. 1. представлена блок-схема алгоритма П-1.

Оператор 1 представляет собой процедуру Монте-Карло-выбирает начальную точку $X_0 \in D$.

Оператор 2 проверяет условие $R_s(X_0) \geq 0$ (1.1).

FOR AUTHOR USE ONLY

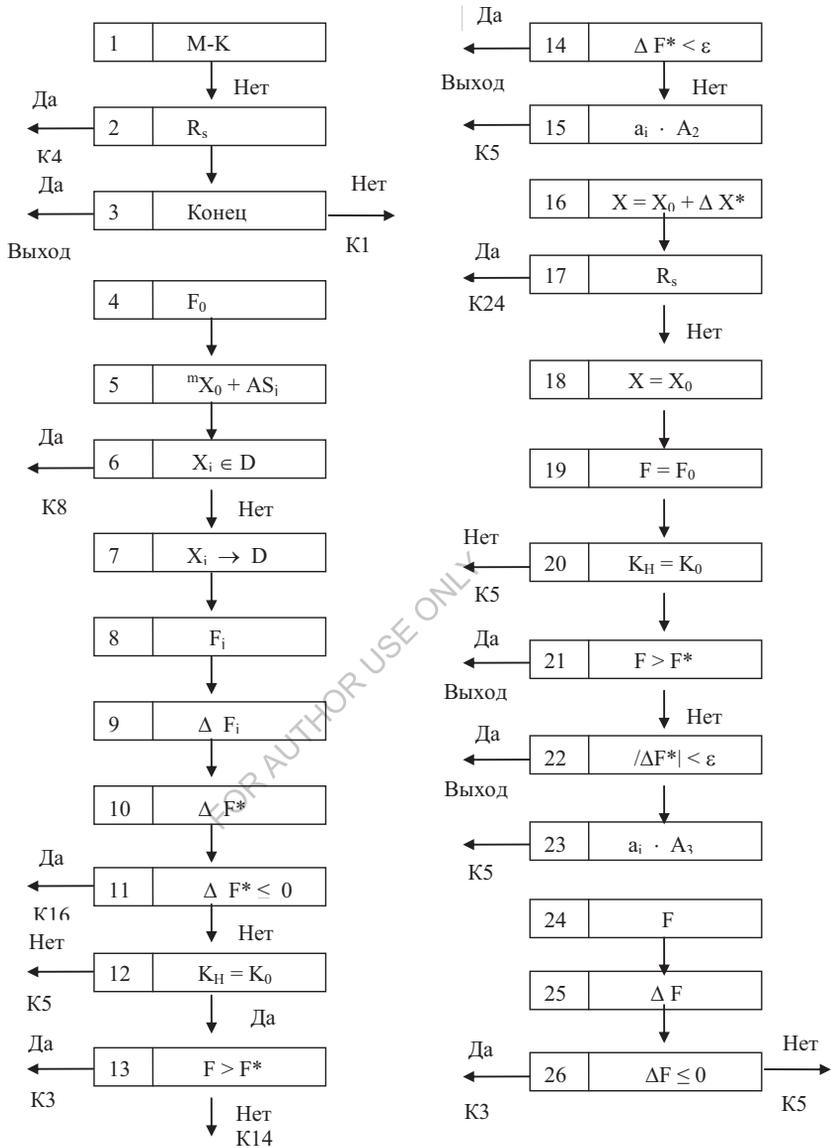


Рис. 3.1. Блок-схема П-1.

Если условие не выполняется происходит передача управления оператору 3, который подсчитывает количество неудачных точек и сравнивает с заданным числом C . Если все C попыток неудачны, то происходит остановка работы программы, т.к. считается, что просмотрена вся область D и вся она «затенена» ограничением (1.1). Если сделано попыток меньше C , то управление передается блоку 1, который выбирает новую точку. Оператор 4 вычисляет значение целевой функции в начальной точке X_0 .

Оператор 5 осуществляет серию m испытаний, равномерно распределенных относительно начальной точки X_0 .

Оператор 6 проверяет принадлежность проведенных испытаний области D , и в случае выхода какой-либо пробы за границы области D , и в случае выхода какой-либо пробы за границы области D , и в случае выхода какой-либо пробы за границы области, оператор 7 возвращает ее область D .

Операторы 8-10 вычисляют значения целевой функции, в каждой точке, приращения целевой функции и выбирает наименьшее приращение.

Операторы 11 сравнивает ΔF^* с нулем. При $\Delta F^* > 0$ операторы 12-15 производят проверку на окончание локального поиска и уточняют координаты локального минимума.

Оператор 16 осуществляет рабочий шаг в выбранном наилучшем направлении. После каждого рабочего шага оператор 17 производит проверку соблюдения ограничений (3.1), при нарушении которых операторы 18-23 производят шаг назад в последнее улучшенное положение, не нарушающее (3.1) и проверку на окончание локального поиска, а также уточнение координат локального минимума, если в том есть необходимость.

При соблюдении ограничений (3.1) управление передается оператору 24, который вычисляет значение целевой функции после каждого рабочего шага. Оператор 25 определяет приращение целевой функции. При $\Delta F \leq 0$ оператор 26 передает управление оператору 16, при $\Delta F > 0$ – оператору 5.

3.2.2. Алгоритм поиска П-2

В этом алгоритме вводится самообучение системы во время поиска. Это самообучение отличается от основных видов самообучения, применяемых в существующих алгоритмах случайного поиска – (по направляющему гиперконусу, по направляющей гиперболы, по направляющему гиперэллипсоиду, по координатному и т.д.) Введенный нами тип самообучения проще выше указанных в машинной реализации и очень гибкий в работе.

Алгоритм поиска П-2 является безинерционным, т.е. на каждое ухудшение целевой функции он реагирует немедленно.

1. Самообучение. Выбор начальной точки и определение наилучшего направления поиска здесь происходит так же как и в алгоритме поиска П-1. После того, как мы определили из m равномерно распределенных испытаний наилучшую пробу $X_k + H\Xi^*$ и выделили наилучшее направление поиска, определяемое по двум точкам X_k и $X_{k+1} = X_k + H\Xi^*$, включается система, самообучения алгоритма, которая помогает определить дальнейшее направление поиска. Относительно вектора $X_{k+2} = X_{k+1} + H\Xi^*$ делаем серию из m_1 испытаний $X_{k+2} + H\Xi_j^N$ где $\Xi^N = (\xi_1^N, \xi_2^N, \dots, \xi_n^N)$ – случайный вектор с компонентами ξ_i^N – случайными числами, нормально распределенными относительно нуля с дисперсией σ .

Таким образом, направления, не улучшающие или улучшающие слабо целевую функцию из рассмотрения исключаются, а все новые испытания концентрируются около выбранного наилучшего направления, улучшающего целевую функцию максимальным образом. Степень этой концентрации у лучшего направления определяется дисперсией σ ,

которая является величиной переменной и зависит от скорости изменения целевой функции.

Дисперсия σ подсчитывает после каждого рабочего шага по следующим рекуррентным выражениям:

$$\sigma_{k+1} = \begin{cases} A_4 \cdot \sigma_k & \text{при } \Delta F_k < \Delta F_{k-1}; \\ A_5 \cdot \sigma_k & \text{при } \Delta F_k \geq \Delta F_{k-1}; \\ A_6 \cdot H & \text{при } \sigma_{k+1} \geq A_6 \cdot H_j; \\ A_1 \cdot H & \text{при } K = 1 \vee \Delta F_k \geq 0. \end{cases} \quad (3.4)$$

Здесь k – номер шага при поиске локального минимума:

$\Delta F_k, \Delta F_{k-1}$ – приращение целевой функции соответственно за k -тый и за $k-1$ -й шаги.

$A_6 \cdot H$ – вектор, ограничивающий дисперсию σ сверху.

$$0 < A_4 < 1; \quad 0 < A_1 \leq 1;$$

$$A_5 > 1; \quad 0 < A_6 \leq 4.$$

Из m_1 испытаний $X_{k+2} + H\Xi^N$ опять выбирается наилучшее $X_{k+3} = X_{k+2} + H\Xi^{N*}$ и, если оно улучшает целевую функцию, относительно вектора $X_{k+4} = 2X_{k+3} - X_{k+1}$ делается снова серия из испытаний.

В целях экономии машинного времени, очевидно, что следует полагать $m_1 < m$, в силу того, что чрезмерная концентрация испытаний около одного направления не даст существенного улучшения направления, а затраты времени на проведение большой серии испытаний будут значительны.

Если серия из m_1 , нормально распределенных испытаний не даст улучшения целевой функции, то происходит переход к равномерно распределенным испытаниям.

Из выражения (3.4) видно, что с увеличением скорости улучшения целевой функции дисперсия уменьшается, а, следовательно, концентрация испытаний увеличивается. Так сделано из того соображения, что если направление поиска улучшается быстро, то оно приближается к наилучшему направлению из всех возможных, и, уменьшая дисперсию, мы уменьшаем зону поиска и увеличиваем вероятность нахождения наилучшего направления. При уменьшении скорости улучшения целевой функции мы удаляемся от направления, максимально улучшающего целевую функцию, и, следовательно, надо расширить зону испытаний, с тем, чтобы найти ускользящее наилучшее направление. Этого мы добиваемся, увеличивая дисперсию. Для того, чтобы дисперсия не вышла за разумные пределы, мы ее ограничиваем сверху.

Введение самообучения позволило резко увеличить скорость попадания системы в окрестность минимума.

Блок-схема П-2 представлена на рис. 3.2

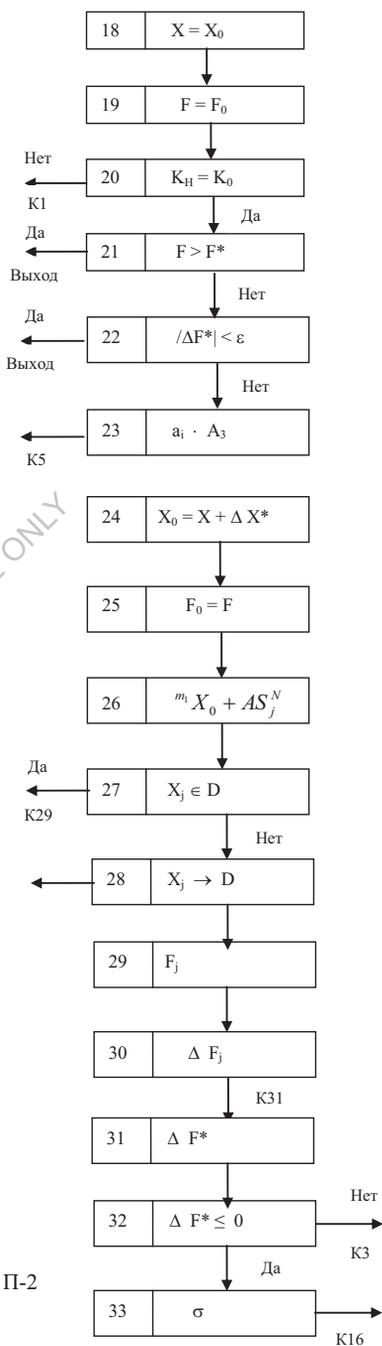
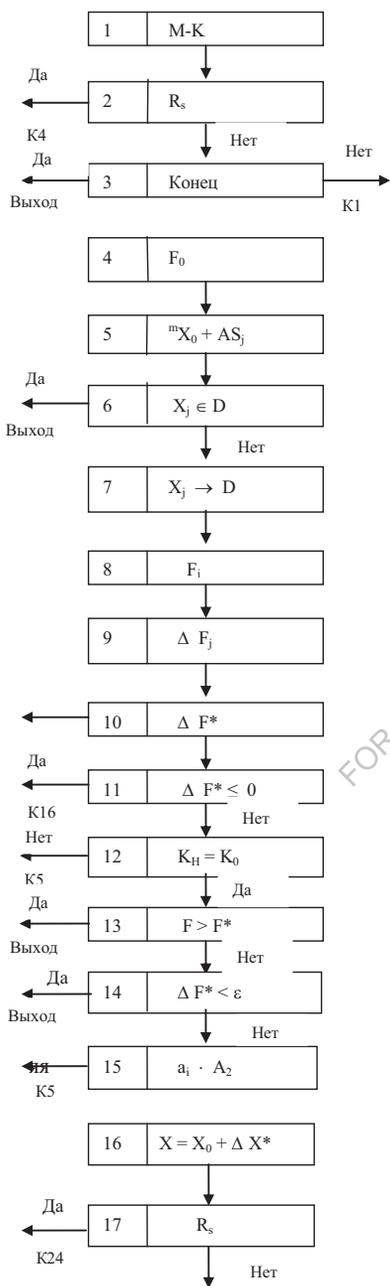


Рис. 3.2. Блок-схема П-2

Операторы 1-23 выполняют те же функции, что и в алгоритме поиска П-1.

Операторы 24 и 25 осуществляют передачу нулевых индексов последним улучшенным значениям X и F.

Оператор 26 организует серию m_1 нормально распределенных испытаний. Операторы 27-32 выполняют те же функции, что и операторы – 6-11. В случае выполнения поставленного условия оператор 32 передает управление оператору 33, который вычисляет дисперсию для следующих испытаний и передает управление оператору 16.

3.3. Алгоритмы глобального поиска

3.3.1. Алгоритм глобального поиска ГП-3. Алгоритм глобального поиска ГП-3 является результатом дальнейшего усиления и модернизации алгоритмов поиска П-1 и П-2 и предназначен для решения многоэкстремальных задач. Все, что описывалось в 3.2. в пунктах 1-7 в алгоритме ГП-3 соответствует алгоритмам П-1 и П-2. Кроме части п.5, касающейся поиска вблизи функциональных ограничений.

1. Обеспечение глобальности поиска.

Глобальность поиска обеспечивается синтезом алгоритма Монте-Карло, определяющего начальные точки поиска, равномерно распределенные в области поиска, производящего поиск локальных минимумов у начальных точек, определенных алгоритмом Монте-Карло.

2. Построение дополнительной системы ограничений.

Общим недостатком существующих алгоритмов глобального поиска является то, что один и тот же локальный минимум отыскивается по несколько раз. На это тратится много лишнего времени.

Предлагаемый алгоритм ГП-3 избавлен от указанного недостатка, так как по мере нахождения локальных минимумов, алгоритмом строится дополнительная система ограничений, не допускающая поисковую систему в уже найденные минимумы. Указанная дополнительная система ограничений строится следующим образом.

После нахождения $L^{го}$ локального минимума под пространство:

$$\sum_{i=1}^n (x_i - x_i^{(j)})^2 - R_j^2 \leq 0, \quad j = \overline{1, L} \quad (3.5)$$

считается «запретной зоной», доступ в которую запрещен. Здесь

x_i – координаты точки, производящей поиск;

$x_i^{(j)}$ – координаты $j^{го}$ локального минимума;

R_j – радиус j -той «запретной зоны», прикрывающей j -тый локальный минимум.

Радиус R_j определяется по следующей формуле:

$$R_j = K_n \cdot \sqrt{\sum_{i=1}^n (x_i^{(j_0)} - x_i^{(j)})^2}, \quad (3.6)$$

где: K_n – коэффициент полноты перекрытия бассейна локального минимума $0 < K_n \leq 1$; $x_i^{(j_0)}$ – координаты j -той начальной точки, из которой был найден j -тый локальный минимум.

После выполнения условия окончания локального поиска (3.3) последняя улучшенная точка считается центром локального минимума, все параметры которого

заносятся в память. Около этого центра описывается «запретная зона» радиусом R предельным по выражению (3.6). После этого процедура Монте-Карло ищет новую начальную точку, удовлетворяющую всем ограничениям и не попадающую ни в одну из «запретных зон»; из которой производится поиск нового локального минимума. Если при поиске система наткнулась на j -тую «запретную зону», то радиус последней увеличивается на величину ΔR_j , определяемую по выражению:

$$\Delta R_j = K_{II} \cdot \sqrt{\sum_{i=1}^n (x_i^{j0} - x_i)^2}, \quad (3.7)$$

где x_i – координаты точки, нарушившей «запретную зону». Коэффициент полноты перекрытия бассейна локального минимума K_{II} в (3.6) и (3.7) имеет одно и то же значение. Вообще K_{II} задается перед поиском и остается постоянным в течение всего глобального поиска. Значение K_{II} зависит от вида целевой функции. Если о минимизируемой функции ничего не известно, или известно, что она имеет минимумы «овражного типа», то коэффициент K_{II} желательно выбирать на отрезке, $0 < K_{II} \leq 0,1$, так как при больших значениях K_{II} возводимая «запретная зона» может перекрыть близлежащий минимум, что повлечет утерю глобальности поиска. Если же функция симметричная или имеет минимумы параболического профиля (гипотезу о минимумах параболического профиля для построения алгоритмов глобального поиска принимает, например, А.И. Половинкин [129]. Проведенные им исследования доказывают правомерность этой гипотезы), то значение K_{II} может быть выбрано ближе к 1. Увеличение значения K_{II} ведет к увеличению радиуса «запретных зон», что, в свою очередь, ведет к уменьшению затрат машинного времени на глобальный поиск.

Вообще же коэффициент K_{II} является одним из параметров настройки алгоритма на конкретную задачу, и правильный выбор значения его зависит от особенностей решаемой задачи и от навыка и интуиции исследователя.

Необходимость настройки алгоритма перед решением той или иной задачи для успешного решения ее возможно короткий отрезок времени является общим недостатком алгоритмов случайного поиска. С другой стороны это обстоятельство можно рассматривать как положительную черту, так как позволяет гибко подходить к решению различных максимальных задач, добиваясь максимального снижения затрат в каждом конкретном случае.

После увеличения «запретной зоны» на величину ΔR_j управление передается процедуре Монте-Карло, которая находит новую начальную точку и т.д. Таким образом, по мере протекания поиска, происходит постепенное «разрастание» «запретных зон», которые при этом выводят из рассмотрения неперспективные части области поиска.

Учет «запретных зон», в виду простоты, почти не сказывается на общем времени работы алгоритма, хотя ведет к значительной экономии машинного времени за счет уменьшения непроизводительных затрат.

3. Условие окончания глобального поиска.

Глобальный поиск прекращается после выполнения условия:

$$K_{33} = K_{III} \quad (3.8)$$

где K_{33} – количество попаданий в запретные зоны при поиске нового локального минимума;

K_{III} – коэффициент, характеризующий плотность просмотра области поиска.

Условие (3.8) в общем соответствует левой части условия окончания локального поиска (3.3), характеризующей вероятную сторону поиска. Оба эти условия имеют одну

природу и вытекает из очевидного предположения о том, что, если при каком-либо состоянии поисковой системы при проведении достаточного количества серий испытаний, равномерно распределенных по области поиска (для условия (3.8)) или в окрестности полученной точки (для условия (3.3)), не получаем улучшения состояния, то полученное состояние является наилучшим из всех возможных в окрестности его.

Величина $K_{ГП}$ непосредственно влияет на вероятность нахождения глобального минимума, так как при увеличении величины $K_{ГП}$ для прекращения глобального поиска, мы должны провести более длинную серию испытаний, что влечет за собой увеличение вероятности нахождения глобального минимума.

Как будет показано ниже, увеличение величины $K_{ГП}$ незначительно влияет на увеличение машинного времени, тогда как у существующих алгоритмов глобального поиска эта связь прямо пропорциональна.

Блок-схема алгоритма глобального поиска ГП-3 представлена на рис. 3.3.

Операторы 1, 3-16, 18, 22-38 – выполняют те же функции, что и операторы 1-33 алгоритма П-2.

Оператор 2 проверяет нарушение начальными точками «запретных зон». При попадании начальной точки в «запретную зону» управление передается оператору 20, который проверяет условия окончания глобального поиска.

Оператор 17 осуществляет «построение» «запретных зон» и вычисляет радиус возводимой «зоны» или «разрастающейся» «зоны».

Оператор 19 проверяет нарушение «запретных зон» точками, производящими поиск. При нарушении «запретных зон» управление передается оператору 20, который, в случае невыполнения условия окончания глобального поиска, передает управление оператору 22.

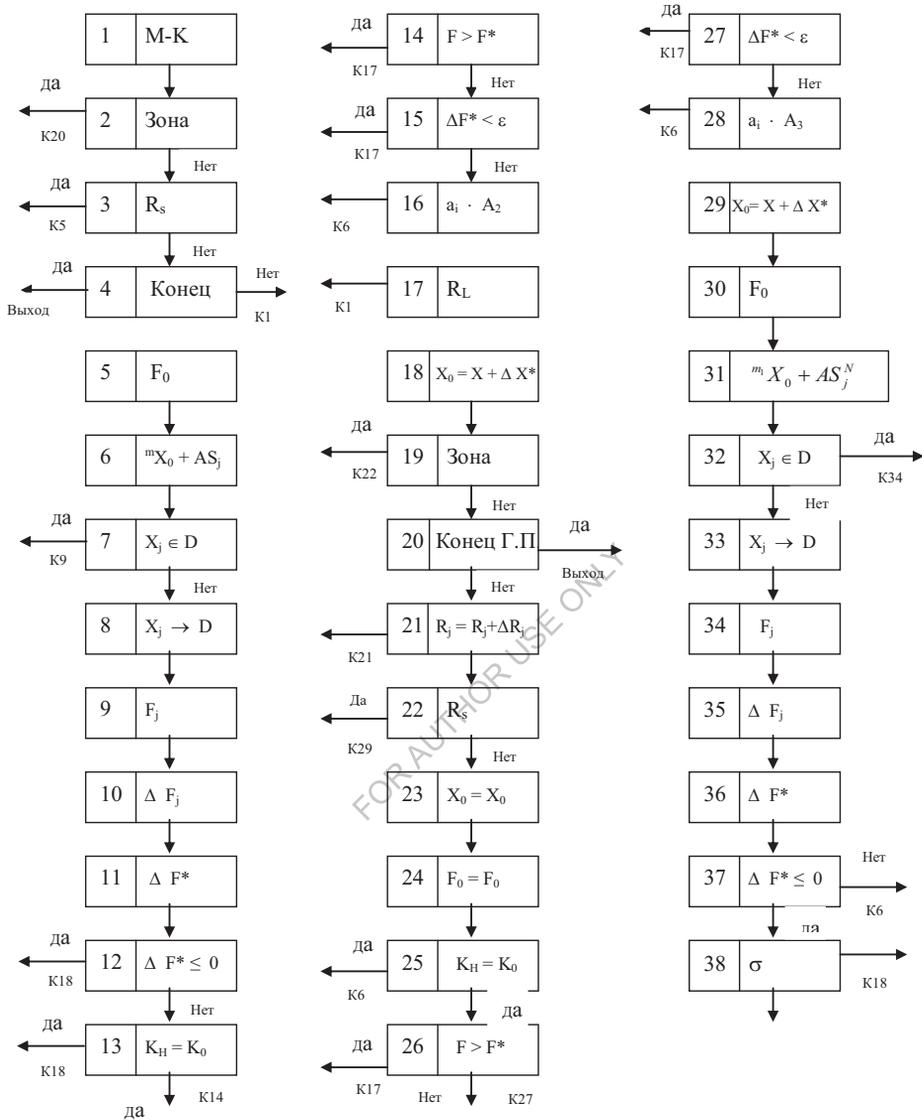


Рис. 3.3. Блок-схема ГП-3

4. Организация поиска вблизи функциональных ограничений.

В алгоритме ПП-3 организация поиска вблизи функциональных ограничений отличается от алгоритмов П-1 и П-2. Здесь для увеличения надежности нахождения условного минимума применен следующий эвристический прием. На каждом шаге после проведения очередной серии из m или m_1 испытаний все положительные пробы $\Delta F_j \leq 0$ заносятся в специальный массив и располагаются там таким образом, что на первом месте стоит проба с $|\Delta F|_{\max}$, за ней следует наилучшая из оставшихся положительных проб и т.д. Проверка ограничений производится в наилучшей пробе. При удовлетворении их поиск продолжается обычным порядком, иначе выбирается следующая проба и производится проверка ограничений в ней и т.д. Если все пробы, занесенные в массив не удовлетворяют ограничениям, то производится новая серия испытаний и т.д., как в алгоритмах поиска П-1 и П-2.

Введение этого приема хотя и несколько усложняет алгоритм, однако позволяет значительно улучшить характеристики поиска вблизи ограничений, особенно в тех случаях, когда углы между градиентами целевой функции и ограничений малы, ввиду большей плотности и дисперсности просмотра окрестности границ.

Возможности предложенного подхода к поиску вблизи границ проиллюстрируем на следующем примере решения задачи

$$F(X) = \frac{1}{(x_1 + 1)^2 + x_2^2} \rightarrow \min; \left. \begin{array}{l} x_1^2 + x_2^2 \geq 4; \\ x_1^2 + x_2^2 \leq 16; \\ -5 \leq x_{1,2} \leq 5. \end{array} \right\} \quad (3.9)$$

Точное решение этой задачи $F(X^*) = \frac{1}{25}$, $x_1^* = 4$, $x_2^* = 0$. Линия постоянного уровня целевой функции по характеру близки к верхней границе, составляя с ней довольно малый угол, что обуславливает трудность поиска условного минимума, особенно из достаточно удаленных от него начальных точек. На рис.3.4 показаны траектории поиска с применением предложенного приема и без него. Здесь отчетливо видно, что поисковая система приходила к окрестность минимума из очень неблагоприятных исходных точек, тогда как при отсутствии предлагаемого приема она быстро остановилась не доходя до минимума.

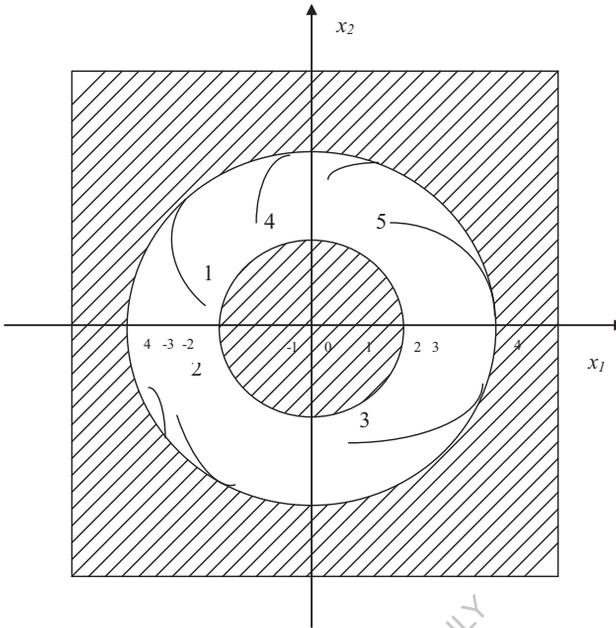


Рис. 3.4

5. Дальнейшее уменьшение затрат машинного времени на глобальный поиск.

При решении многоэкстремальной задачи в большинстве случаев нас интересует лишь глобальный минимум и минимумы, по значению целевой функции к нему. Поэтому тратить машинное время на уточненный поиск «мелких» неперспективных локальных минимумов не всегда целесообразно. В связи с этим, нами применен следующий прием, позволивший избежать излишние затраты машинного времени. Суть этого приема заключается в следующем. Первый локальный минимум отыскивается точно, т.е. с заданной точностью ε . При определении последующих локальных минимумов до уточнения координат делается проверка условия: $F_k \leq F^*$, где F_k – полученное значение целевой функции в последнем улучшенном состоянии; F^* – наилучшее известное значение целевой функции.

При выполнении условия ($F_k \leq F^*$) происходит уточнение координат локального минимума и, если значение целевой функции в нем меньше, чем F^* , то F^* принимает значение целевой функции в последнем минимуме. Если же условие ($F_k \leq F^*$) не выполняется, то поиск локального минимума прекращается.

Алгоритм глобального поиска ГП-3 построен таким образом, что в зависимости от значений некоторых его параметров он работает как ряд других алгоритмов (глобального и локального поиска). Глобальные алгоритмы:

1. При $K_n > 0$ - глобальный поиск с «запретными зонами»;
2. При $K_n=0$ – комбинированный алгоритм Монте-Карло с алгоритмом локального поиска (МКС).

В каждом алгоритме глобального поиска поиск локальных минимумов может производиться алгоритмами:

1. $m = 1 \wedge m_1 = 0$ – случайного спуска;
2. $m > 1 \wedge m_1 = 0$ – спуск по наилучшей пробе;

3. $m = 1 \wedge m_1 = 1$ – случайный спуск с линейной тактикой;
4. $m > 1 \wedge m_1 = 1$ – спуск по наилучшей пробе с линейной тактикой;
5. $m > 1 \wedge m_1 > 1$ – спуск по наилучшей пробе с самообучением.

Отметим, что, варьируя значениями A_2, A_3 , можно вышеперечисленные алгоритмы сделать как с адаптацией, так и без нее, хотя, очевидно, исключать адаптацию целесообразно.

Таким образом ГП-3 может моделировать различные тактики и стратегии поиска (как глобального, так и локального), присущих целому классу алгоритмов, что является, на наш взгляд, ценным качеством.

На базе ГП-3 создана его модификация, способная решать задачи при наличии целочисленных переменных. Основные концепции, заложенные в ГП-3 остались без изменений. Поэтому на этой модификации подробно останавливаться не будем.

Отладка программы поиска с целочисленными переменными производилась на примере известной теоремы Ферма:

$$x^n + y^n \neq z^n \text{ при } n > 2 \quad (3.10)$$

где x, y, z, n – целые числа. Задача ставилась следующим образом

$$F(X) = |x_1^{x_4} + x_2^{x_4} - x_3^{x_4}| \rightarrow \min; \quad (3.11)$$

$$1 \leq x_{1,2,3} \leq 5000;$$

$$3 \leq x_4 \leq 5.$$

В таблице 3.1 приведены результаты одного из решений задачи (3.11).

№ шага	F(X)	x_1	x_2	x_3	x_4
1	0, 495090879 10 ¹⁵	396	4717	115	4
10	0, 1026241714 10 ¹²	384	4681	112	3
20	0, 907943165 10 ¹¹	324	4494	88	3
30	0, 7112537128 10 ¹¹	236	4143	44	3
40	0, 4850806243 10 ¹¹	91	3647	23	3
50	0, 2702694968 10 ¹¹	1	3001	39	3
60	0, 1067466031 10 ¹¹	1	2202	134	3
70	0, 1204967420 10 ¹⁰	1	1067	214	3
75	0, 9509408000 10 ⁷	1	311	274	3
80	0, 2441520 10 ⁶	11	285	284	3
81	1000	10	284	284	3
82	343	7	284	284	3
83	216	6	284	284	3
84	64	4	283	283	3
85	8	2	282	282	3
86	1	1	281	281	5

3.3.2. Алгоритм специального случайного поиска для оптимизации инженерных конструкций. При оптимизации конструкций, находящихся под действием внешних нагрузок, всегда существуют ограничения по прочности, устойчивости, жесткости. И если при оптимальном проектировании конструкции за минимизируемую функцию принимается функция, вычисляющая вес конструкции, то минимум ее всегда находится на одной из этих границ или на стыке их, так как в противном случае можно всегда убрать

какое-то количество материала без ущерба прочности конструкции. В связи с вышесказанным нами предложен алгоритм ГП-4 с переменным шагом для оптимизации конструкции, отвечающий условиям прочности. Этот алгоритм можно рассматривать, как модификацию алгоритма ГП-3, так как поисковые системы этих алгоритмов, в основном, совпадают. Отличие заключается в том, что в алгоритме ГП-4 введен блок, рассчитывающий величину рабочего шага системы в процессе поиска. Введение этого блока повлияло также на процедуру Монте-Карло, определяющую начальные точки поиска.

Сущность предлагаемого метода определения величины рабочего шага заключается в том, что последний определяется в зависимости от степени удаления поисковой системы от границы прочности, то есть в зависимости от разности между допускаемым в конструкции напряжением и максимальным напряжением в конструкции, достигнутым в результате последнего шага.

В алгоритме ГП-4 масштаб рабочего шага определяется по следующей формуле:

$$MШ_{K+1} = MШ_K \cdot \text{sign}([\sigma] - [\sigma_k^{\max}]) \cdot \left| \frac{[\sigma] - \sigma_x^{\max}}{\sigma_x^{\max} - \sigma_{x-1}^{\max}} \right|^p; \quad (3.12)$$

$$MШ_0 = 1,$$

где K – счетчик шагов;

$[\sigma]$ – допускаемое напряжение;

$\sigma_k^{\max}, \sigma_{k-1}^{\max}$ – максимальное напряжение в конструкции, получаемые после K -го и $K-1$ -го шагов;

$P = \text{const}$ – зависит от степени нелинейности зависимости напряжений от оптимизируемых параметров.

Формула (3.12) определяет зависимость модуля рабочего шага от напряжений в конструкции. Точно таким же образом можно выразить зависимость $MШ$ от деформации в конструкции.

Для избежания излишних затрат при решении практических задач в случае, когда зависимость напряжений от оптимизируемых параметров установить трудно, в алгоритме предусмотрено ограничение модуля рабочего шага, определяемого по (3.12):

$$T_1 \leq MШ \leq T_2. \quad (3.13)$$

Определяя таким образом модуль шага, максимально снижаем затраты времени на подход к окрестности минимума, так как при большом удалении от границы делается большой шаг, позволяющий приблизиться к границе без лишних затрат времени.

Если начальная точка, которую определила процедура Монте-Карло находится за границей прочности, то, как видно из (3.12) модуль шага принимает отрицательное значение, что позволяет системе уходить в сторону увеличения целевой функции до перехода через границу прочности, после чего поиск минимума осуществляется обычным путем, описанным в п. 3.3.

При оптимизации конструкций, отвечающих условиям прочности, затрачиваемое на нахождение минимума времени можно представить в виде:

$$T = T_{HT} + T_{П} + T_{УТ}, \quad (3.14)$$

где T_{HT} – время, затрачиваемое на обработку начальных точек, лежащих за границей прочности;

$T_{П}$ – время подхода к окрестности минимума;

$T_{УТ}$ – время, затрачиваемое на уточнение координат минимума.

В некоторых случаях T_{HT} и $T_{П}$ составляют значительную часть общего времени.

При оптимизации конструкций при помощи алгоритма ГП-4 неудачных начальных точек нет и время $T_{HT}=0$. На подход к окрестности минимума также затрачивается мало времени ввиду переменной величины рабочего шага. В основном, время тратится на уточнение координат минимума. Так, при оптимизации арки (об этом будет говориться в п. 6.1.) в пространстве трех оптимизируемых параметров при помощи алгоритма ГП-4, в среднем, на отыскание одного минимума тратилось $6 \div 7$ шагов. Глобальный минимум, который определялся с повышенной точностью, найден за 28 шагов. Эта разница в затрачиваемых шагах целиком определяется увеличением точности поиска. При оптимизации той же арки при постоянном рабочем шаге на отыскание одного минимума тратилось 20-45 шагов.

При оптимизации стержневых конструкций по минимуму веса или стоимости, когда в качестве оптимизируемых параметров принимаются параметры, влияющие на жесткость стержней, оказывается целесообразной следующая модификация алгоритма ГП-4. Вводится не один коэффициент МШ, а несколько, по количеству стержней или параметров. Величина и знак коэффициента МШ зависит от напряженного состояния в j -том стержне, т.е.

$$MШ_{k+1}^i = MШ_k^i \cdot \text{sign}([\sigma]^i - \sigma_k^i) \cdot \left| \frac{[\sigma]^i - \sigma_k^i}{\sigma_k^i - \sigma_{k-1}^i} \right|^p \quad (3.15)$$

все члены те же, что и в (3.12), но берутся для i -того стержня.

При подобном определении коэффициента МШ, алгоритм будет отыскивать минимальное значение веса или стоимости конструкции при одновременном обеспечении функционирования всех стержней около или на границе прочности. Полученная таким образом поисково-итерационная процедура обеспечивает повышенную скорость сходимости к глобальному по сравнению с чисто поисковыми или чисто итерационными процедурами.

Глава 4. РАЗРАБОТКА АЛГОРИТМИЧЕСКОЙ СИСТЕМЫ ОПТИМИЗАЦИИ

4.1. Банк постановки задач

Для успешной работы алгоритмической системы при решении возможно наиболее широкого класса задач необходимо формализовать и унифицировать постановку задач. Банк постановки в предлагаемой системе для этих целей.

Рассмотрим требования, которые должны предъявляться к банку постановки. Во-первых, это требование унификации постановки задач. Иначе говоря все задачи, предназначенные к решению при помощи данной системы, должны ставиться единообразно. Во-вторых, постановки задач должны быть полностью формализованы и не допускать двойственности в понимании задачи. Это требование основывается на том, что система предназначена для работы в автоматическом режиме, т.е. без вмешательства человека в процессе решения, а следовательно и не требующая обязательного участия в работе специалистов в области решения задач математического программирования. При отсутствии подобных специалистов, способных конкретно поставить задачи оптимизации, принимает особое значение требования полной формализации задач с тем, чтобы даже техник-программист правильно ввел задачу в систему.

Для обеспечения этого важны также следующие условия:

- удобная запись исходных данных;
- максимальная простота постановки задачи.

Необходимость выполнения этих условий для организации B_0 очевидна и, на наш взгляд, не нуждается в какой-либо дополнительной аргументации.

Успешная работа системы невозможна без организации связи между внешним языком (языком постановки задачи) и внутренним языком компьютер, которую должен обеспечить B_0 .

При постановке задач возможны возникновения ошибок различного рода, связанных с действиями пользователя и программиста, обслуживающего систему. С целью экономии ресурса времени после ввода исходной информации в компьютер до начала работы всей системы необходимо провести контроль за правильностью постановки задачи с синтаксической стороны. Для этого в B_0 должны быть предусмотрены соответствующие сервисные программы.

Рассмотрим сформулированные требования к B_0 с позиции анализа функций технических систем, изложенные в [49], так как АСО можно также отнести к подобным системам.

Кратко опишем методику анализа функций технических систем (ТС), которой будем в дальнейшем пользоваться, придерживаясь, в основном, терминологии и обозначениями [49].

Функция ТС обозначается F . Если ТС многофункциональна, то её функции обозначаются F', F'', \dots . рассматриваемая система разбивается на функциональные элементы. Это могут быть элементы разных уровней.

Функциональные элементы первого уровня представляют собой укрупненные элементы, или блоки системы, каждый из которых имеет некоторое число самостоятельных функций по обеспечению работы других блоков и всей системы.

Для рассматриваемой АСО элементами первого уровня являются алгоритмические и вспомогательные банки B_i ($i=0,7$), каждый из которых имеет четко определенный круг функций. Элементы первого уровня обозначим B_i ($i=0,7$), а их функции соответственно F_0', F_0'', F_1', \dots .

Каждый из элементов первого уровня представляет собой ТС и состоит из элементов второго уровня $B_{0-1}, B_{0-2}, \dots, B_{1-1}, B_{1-2}, \dots$ со своими функциями $F'_{0-1}, F''_{0-1}, \dots, F'_{0-2}, \dots, F'_{1-1}, \dots$, которые в свою очередь могут состоять из элементов третьего уровня и т.д.

Таким образом можно произвести анализ функций системы и построить соответствующую таблицу вплоть до неделимых элементов (операторов).

Рассматривая B_0 , мы можем выделить его следующие функции:

F_0' - постановка задачи (унифицированная, формализованная);

F_0'' - ввод данных в компьютер;

F_0''' - контроль правильности постановки задачи;

F_0^{IV} - распределение данных по другим элементам первого уровня B_1 .

Для реализации каждой функции B_0 соответственно должны быть элементы второго уровня.

Рассмотрим их подробнее.

Из F_0' следует, что блок постановки задач предназначен для записи в форме, удобной для дальнейшей переработки системой.

Должна быть задана следующая информация о задаче.

1. Записана целевая функция на алгоритмическом языке. В нашем случае это АЛГОЛ-60 [134], т.к. система ориентирована на него.
2. Записаны все функциональные ограничения.
3. Указан вид целевой функции (линейная, квадратичная, выпуклая, многоэкстремальная, с дискретно изменяющимися параметрами).
4. Указан вид ограничений.
5. Дополнительные сведения о целевой функции. Сюда входят записанные на АЛГОЛе в аналитическом виде первые, вторые производные, сведения об удовлетворении их условию Лившица, данные об обращаемости матрицы вторых производных.
6. Такие же сведения об ограничениях.
7. Указано количество переменных.
8. Указано количество ограничений.
9. Точность определения экстремума.
10. Начальная точка, из которой будет производиться поиск локального экстремума.
11. Границы изменения параметров.
12. Количество дискретных параметров.
13. Указать, искать минимум или максимум.

Все эти сведения заносятся в специальную стандартного вида “карту постановки задач” (элемент второго уровня B_{0-1}), которая заполняется самим заказчиком или программистом. Информация, заложенная в этой карте вводится в систему.

Карта постановки задач имеет следующий вид.

Карта постановки задач.

Количество переменных N	Количество целочислен. пер-х NI	Точность по $x \ \varepsilon_x$	Точность по ε_u	П 1:27
Вид целевой функции				
линейная				1.
квадратичная				2.
выпуклая				3.
линейноэкстремальная				4.
враждебная				5.
искривленная				6.
различия $Ц(x)$				7.
различия $Ц'(x)$				8.
удовлетворение $Ц(x)$ условию ищущая				9.
обращаемость матрицы вторых производных				10.
Вид функциональных ограничений				
линейные				11.
выпуклые				12.
выпуклые				13.
искривленные				14.
различия $f'_i(x)$				15.
различия $f''_i(x)$				16.
удовлетворение $f'_i(x)$ условию ищущая				17.
нижняя граница $A [O:N]$				18.
верхняя граница $B [O:N]$				19.
начальная точка $X_0 [O:N]$				20.
табл $H [O:N]$				21.
нижняя граница $AI [O:NI]$				22.
верхняя граница $BI [O:NI]$				23.
начальная точка $X_0I [O:NI]$				24.
табл $HI [O:NI]$				25.
ограничения $f_i(x)=0$				26.
ограничения $f_i(x) \leq 0$				27.
функция цели $Ц(x)$				
$Ц'(x)$				
$Ц''(x)$				
$f'_i(x)$				
$f''_i(x)$				

Опишем подробнее карту постановки. В первой строке предусмотрено занесение оптимизация количество непрерывно изменяющихся переменных (N), о количестве целочисленных переменных (NI), необходимая точность отыскания координат минимума целевой функции по приращениям координат (ε_x) или по изменению функции цели (ε_u). Правый столбец занимает массив

П 1:27 - целочисленный, элементы которого принимают значения 0 или 1 в зависимости от отсутствия или наличия соответствующей информации. Так, если целевая функция линейна, то в первую клетку массива П заносим 1, а в остальные клетки, относящиеся к виду целевой функции - 0. Таким же образом поступают со всей информацией,

указанной на карте. Если оптимизация в виде целевой функции ничего не известно, то её надо считать многоэкстремальной и присвоить $\Pi_4=1$. То же можно сказать об ограничениях – при отсутствии информации их следует считать невыпуклыми.

Введение информации о наличии первых и вторых производных целевой функции и ограничений связано с применением регулярных алгоритмов математического программирования (градиентные, ньютоновские), сходимость которых доказана лишь при удовлетворении их

$\Pi'(x)$ условию Липшица с некоторой константой R (опять – таки неизвестной). Успешное применение метода Ньютона требует хорошей включение этой информации в карту постановки задач.

Значения нижней границы A , верхней границы изменения параметров B , координаты начальной точки X_0 , размер рабочего шага H задаются массивами действительных чисел. Причем возможно отсутствие информации о границах, но в этом случае необходимо задать X_0 . При отсутствии начальной точки возникает необходимость задания границ A и B . Для решения задачи желательно, но не обязательно задавать величины рабочих шагов H , т.к. в системе предусмотрено при отсутствии информации оптимизация величине шагов определять их автоматически в соответствии из размеров поиска.

Исходные данные, касающиеся целочисленных переменных задаются массивами целых чисел. Так же, как и при задании непрерывно изменяющихся переменных, здесь возможны различные варианты задания информации.

Ограничения, функция цели, первые и вторые их производные записываются на языке АЛГОЛ-60 самим заказчиком или программистом по представленным заказчиком алгоритмам[134].

Карта постановки задач здесь представлена схематически. Рабочая же карта состоит из нескольких листов, причем на ограничения, целевую функцию, производные выделяются отдельные листки. Функция $F_0''(x)$ – ввод данных в компьютер выполняется при помощи стандартных операторов ввода ($B_{0.2}$). Для этого информация, заданная на карте постановки задачи перфорируется в порядке её расположения на карте. Вводится в компьютер она поэтапно, т.е. сначала $N, NЦ, \varepsilon_x, \varepsilon_y$ и массив Π 1:27, а затем в виде исходных данных остальная информация до ограничений и целевой функции, которые, как уже писалось записываются на языке компьютер и оформляются в виде отдельных модулей.

В целях максимального облегчения заказчику правильной постановки задачи и исключения возможности затрат машинного времени на исследование системой заведомо неправильно поставленной задачи (при отсутствии необходимой или при наличии взаимоисключающей информации) при запуске системы предусмотрена работа блока проверки постановки задач ($B_{0.3}$), т.е. выполнение $F_0'''(x)$.

Этот блок состоит из ряда сервисных логических программ, производящих проверку поступившей информации. При наличии ошибок блок проверки прерывает работу системы и выдаёт на печать информацию о месте, где допущена ошибка.

Так, если не указан вид целевой функции, то на печать машина выдаст : **«не указан вид целевой функции»**. Если указано одновременно, что целевая функция одновременно разных видов – последует информация **«уточнить вид целевой функции»**. Если не указан вид ограничений – **«не указан вид ограничений»**. При задании вида ограничений в случае одновременного наличия ограничений разных видов (линейные, выпуклые, невыпуклые, дискретные) указывается лишь один вид. Например: если есть дискретные ограничения, то независимо от наличия других ограничений указываются его ограничения дискретные и признак $\Pi_4=1$. Если дискретные ограничения отсутствуют, то в случае наличия невыпуклых ограничений $\Pi_3=1$ и т.д. Поэтому, если при постановке задачи указаны ограничения различных видов, то произойдет выдача **«уточнить вид ограничений»**. Если количество чисел в массивах $A, B, X_0, H, AЦ, BЦ, X_0Ц, HЦ$ не

соответствует нужным числам $N+1$ или $NC+1$, то на печать пойдет информация “группа данных $>$ ($<$) длины массива оператор $INPUT\ N\ 3$ объект ввода $N\dots$ ”. Причем нижней границе A соответствует объект ввода $1, B-2, XO-3, H-4, AC-5, BC-6, XOЦ-7, HC-8$.

Приведем ещё примеры выдаваемой информации об ошибках. “отсутствие нижней (A или AC) (верхней) B или BC границы” – при отсутствии нижней (верхней) границы изменения параметров и начальной точки, если одна из границ задана. “отсутствие начальной точки- при отсутствии границ и XO ($XOЦ$)”.

Таким образом, функции B_{0-3} :

$F'_{0-3} \equiv F''_0$; F''_{0-3} - информирование пользователя о допущенных ошибках; F'''_{0-3} - останов работы системы.

Функцию F_0^{IV} - распределение данных по другим элементам первого уровня выполняет $B_{0.4}$ – блок распределения, который распределяет информацию между B_1 – банком данных; B_3 - банком признаков; B_4 – банком модулей. В B_1 идут исходные данные по задаче; в B_3 - признаковый массив P ; в B_4 – модули вычисления целевой функции и ограничений.

По итогам анализа функций B_0 составим таблицу анализа функций (см. табл. 4.1).

Таблица 4.1

Анализ функций B_0 .

Элемент	Функции
B_0 - банк постановок	F'_0 - постановка задачи (унифицированная, формализованная); F''_0 - ввод информации в ЭВМ; F'''_0 - контроль правильности постановки задачи; F_0^{IV} - распределение данных между другими элементами B_1, B_3, B_4 ;
$B_{0.1}$ - информационная часть (карта постановки)	$F_{0.1}$ –запись информации о задаче в удобной форме для дальнейшей переработки АСО
$B_{0.2}$ - операционная часть	$F'_{0-2} \equiv F''_0$; $F''_{0-2} = F''_0$; $F'''_{0-2} \equiv F_0^{IV}$.
$B_{0.2-1}$ – модуль ввода	F'_{0-2-1} - получение информации из $B_{0.1}$ $F''_{0-2-1} \equiv F'_{0-2}$;
$B_{0.2-2}$ – блок проверки	$F'_{0-2-2} \equiv F''_{0-2}$; F''_{0-2-2} - информирование пользователя об ошибках в постановке задачи; F'''_{0-2-2} - остановка работы системы; F^{IV}_{0-2-2} - сигнал для начала работы системы.
$B_{0.2-3}$ - блок распределения	$F_{0-2-3} \equiv F'''_{0-2}$.

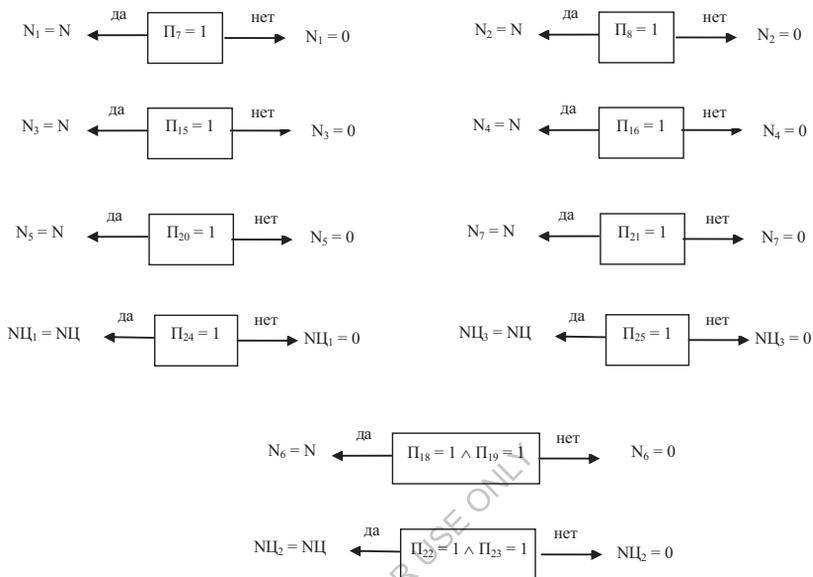
Функциональная схема B_0 представлена на рис. 4.1.

- Π - Массив признаков постановки задач;
 - A, B - массивы соответственно нижней и верхней границ изменения параметров;
 - X_0 - Массив начальных значений непрерывно изменяемых переменных;
 - H - массив модулей рабочих шагов по каждой непрерывно изменяемой переменной;
 - AI, BI - целочисленные массивы нижней и верхней границ изменения целочисленных переменных;
 - X_0I - Массив начальных значений целочисленных параметров;
 - HI - Массив размеров шагов целочисленных параметров;
 - C - Необходимые коэффициенты векторов и матриц целевой функции и ограничений
- К неизменяемой части исходных данных относятся данные
- об объёме МОЗУ, занимаемым каждой поисковой программой;
 - о количестве переменных, которое может взять та или иная программа;
 - о количестве функциональных ограничений, при которых может решаться задача той или иной программой.

По мере пополнения вычислительного банка новыми программами неизменяемая часть банка данных будет увеличиваться.

Ввиду того, что массивы исходных данных могут иметь большую равенств, следует обратить особое внимание на экономию резервируемой памяти компьютер. Это требуется еще вследствие того, что предусмотренные системой варианты постановок задач может содержать различное количество информации. Например: могут быть заданы или нет границы изменения параметров, начальная точка, шаг, первые, вторые производные и т.д. Все эти массивы имеют размерность N для непрерывно переменных и NI – для целочисленных, причем массив, в который заносятся результаты вычисления вторых производных целевой функции, имеет размерность $N \times N$, массив со значениями первых производных ограничений – $(M + M_1) \times N$ (где $M + M_1$ - количество всех функциональных ограничений), массив вторых производных ограничений $(M + M_1) \times N \times N$. Как видим, все эти массивы занимают значительную память компьютер, которая резко возрастает с увеличением числа оптимизируемых параметров. Для экономии памяти мы поступили следующим образом. ввели несколько новых целочисленных переменных: $N_1, N_2, N_3, N_4, N_5, N_6, N_7, NI_1, NI_2, NI_3$.

В зависимости от имеющейся в банке постановок задач информации, в частности от содержания массива признаков постановки Π , эти переменные получают значения 0 или $N (NI)$. Покажем блок-схему этой логической программы (Б_{1.2-1}).



После вышеописанного присвоения для работы системы резервируются массивы не размерностью от 0 или N (NI), а следующие:

$X_0[0 : N_5], A, B[0 : N_6], H[0 : N_1]$

- $Gr_1[0 : N_1]$ - массив, содержащий результаты вычисления первых производных целевой функции;
- $Gr_2[0 : N_2, 0 : N_2]$ - массив, содержащий результаты вычисления вторых производных целевой функции;
- $Gr_3[0 : M + M_1, 0 : N_3]$ - массив, содержащий результаты вычисления первых производных функциональных ограничений;
- $Gr_4[0 : M + M_1, 0 : N_3, 0 : N_3,]$ - массив вторых производных ограничителей;

$X_0H[0 : NI_1], AH, BH[0 : NI_2], HI[0 : NI_3]$

Таким образом достигается максимально возможное уменьшение требуемой памяти в зависимости от вида постановки задачи.

Функции F_1''' - поиск данных; F_1^{IV} - подстановки в B_4, B_6 выполняют операторы вызова ($B_{1-2.2}$) и подстановки ($B_{1-2.3}$). Анализ функций B_1 сведен в табл. 5.2.

Таблица 4.2

Элемент	Функции
B_1 – банк данных	F_1' – хранение исходных данных; F_1'' – распределение памяти; F_1''' – поиск данных; F_1^{IV} – передача в B_4, B_6
B_{1-1} – информационная часть	$F_{1-1} \equiv F_1'$;
B_{1-2} – операционная часть	$F_{1-2}' \equiv F_1''$; $F_{1-2}'' \equiv F_1'''$; $F_{1-2}''' \equiv F_1^{IV}$;
B_{1-2-1} – блок распределения памяти	$F_{1-2-1} \equiv F_{1-2}'$;
B_{1-2-2} – операторы вызова	$F_{1-2-2} \equiv F_{1-2}''$;
B_{1-2-3} – оператор передачи	$F_{1-2-3} \equiv F_{1-2}'''$;
B_{1-2-4} – монитор	F_{1-2-4} – управление B_{1-2-i} ($i = \overline{1,3}$)

Функциональная схема B_1 представлена на рис 4.2

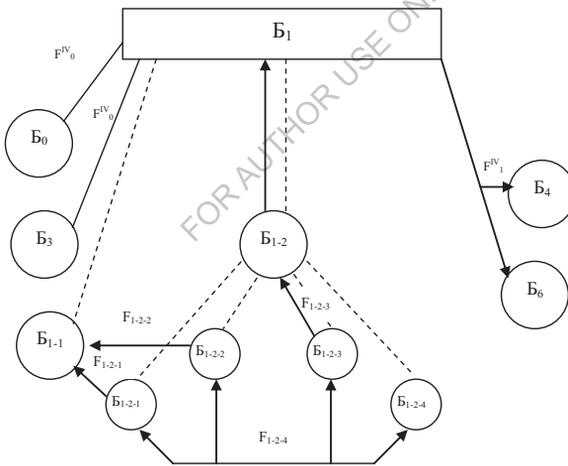


Рис.4.2. Функциональная схема B_1

4.3. Построение банков признаков и моделей

4.3.1. БЗ – банк признаков. Банк признаков служит для хранения и переработки логической информации, предназначенной для идентификации каждой конкретной задачи. Банк B_4 – моделей и B_5 – алгоритмов используют в своей работе признаки задачи, выявленные банком признаков.

Функциями B_3 являются:

F'_3 – хранение логической информации о признаковом пространстве, описывающем все модели, предусмотренные в B_0 ;

F''_3 – формирование значений признаков;

F'''_3 – передача значений признаков в B_1, B_4, B_5 .

Эти функции выполняют элементы второго уровня B_{3-1} – информационная часть B_3 и B_{3-2} – операционная часть B_3 , которая, в свою очередь, делится на элементы третьего уровня:

B_{3-2-1} – модуль формирования значений признаков;

B_{3-2-2} – модуль передачи значений признаков в B_1, B_4, B_5 .

B_{3-2-3} – монитор, управляющий работой B_{3-2-1} и B_{3-2-2} .

Для моделей задач математического программирования и алгоритмов, разрешающих эти модели, выделим следующие признаки:

1. *ПРЦ* – признак вида целевой функции. Этот признак учитывает не только сам вид функции, но и имеющуюся в постановке задачи дополнительную информацию к ней.

2. *ПР1* – признак вида ограничений №1. Учитывает вид функциональных ограничений, а также наличие первых, вторых производных и т.д.

3. *ПР2* – признак вида ограничений №2 – учитывает информацию о наличии того или иного вида ограничений (ограничений на параметры $a_i \leq x_i \leq b_i$), функциональных ограничений $f_i(X) = 0$; $f_i(X) \leq 0$, а также данных о начальной точке, размере шага и т.д.

4. *ПРН* – признак количества оптимизируемых параметров, которое может взять та или иная поисковая программа. Определяется после тестирования поисковых программ.

5. *ПРВ* – признаки скорости достижения определенной точности различных классов задач (тестовых) для каждой программы. Определяется после тестирования.

6. *ПР ϵ* – признаки точности ϵ , достигнутые программами в различных задачах (тестовых) за заданный отрезок времени T . Определяются после тестирования.

Признаки ПРЦ, ПР1, ПР2 формируются согласно табл. 4.3 – 4.5.

Таблица 4.3

i		Значения ПРЦ																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
1	Значения Π_i	1																		
2		0	1	1	1	1	1													
3		0						1	1	1	1	1								
4		0											1							
5		0												1	1	1	1	1	1	
6		0																		1
7		0		1					1						1					
8		0			1					1							1			
9		0				1					1							1		
10		0					1					1							1	

Приведем пояснения к табл. 4.3. Если в поставленной задаче целевая функция линейна, а в связи с этим первый элемент массива признаков постановки задач принимает значение 1, т.е. $\Pi_1=1, \Pi_i=0$ ($i = 2,10$), то ПРЦ=1. В табл. 4.3 содержание всех пустых ячеек есть ноль. ПРЦ=2, если имеем квадратичную функцию, но не имеем никакой дополнительной информации, иначе говоря $\Pi_2=1$. ПРЦ=3, если $\Pi_2=1 \wedge \Pi_7=1$, т.е. если

целевая функция квадратичная и заданы в аналитическом виде первые частные производные. ПРЦ=4 при $\Pi_2=1$ и $\Pi_8=1$, т.е. если заданы вторые частные производные (первые производные считаются заданными автоматически). ПРЦ=5, если заданы первые производные и доказано удовлетворение их условию Липшица с заданной константой R. ПРЦ=6, если заданы производные и доказана обращаемость матрицы вторых производных. ПРЦ=7, если о целевой функции известно только то, что она выпукла. Если же кроме этого заданы еще первые производные, то ПРЦ=8. При задании, кроме этого, вторых производных – ПРЦ=9. Если для выпуклой целевой функции с заданными первыми производными доказано их удовлетворение условию Липшица, то ПРЦ=11. ПРЦ=12, если целевая функция многоэкстремальная. При таких же условиях, как при квадратичной и выпуклой целевой функции, в случае овражной целевой функции признак вида ПРЦ принимает значения 14, 15, 16, 17. И в случае дискретно изменяющейся целевой функции ПРЦ=18.

Вышеперечисленные условия и значения признака ПРЦ охватывают, в основном, все реально возможные данные о виде целевой функции и сочетаниях дополнительной информации при постановке задач.

Признак вида ограничений ПР=1 принимает значения согласно табл. 4.4.

Таблица 4.4

i		Значения ПР1							
		1	2	3	4	5	6	7	8
11	Значения Π_i	1							0
12			1	1	1	1			0
13							1		0
14								1	0
15					1	1	1		0
16						1			0
17							1		0

ПР-1=1, если $\Pi_{11}=1$, т.е. имеем линейные ограничения. ПР-1=2, если $\Pi_{12}=1$ - ограничения выпуклы. ПР-1=3, если ограничения выпуклы и заданы первые частные производные ограничений. ПР-1=4, если ограничения выпуклы и заданы первые и вторые частные производные ограничений. ПР-1=5, если ограничения выпуклы, заданы первые производные ограничений и доказано удовлетворение их условию Липшица с заданной константой R. Если ограничения невыпуклые, то ПР-1=6; и в случае дискретно имеющихся ограничений ПР-1=7. Если функциональные ограничения не заданы, то ПР-1=8 ($\Pi_i = 0, i = \overline{11,17}$).

Признак ограничений ПР2 принимает значения согласно табл. 4.5.

ПР2=1, если заданы координаты начальной точки X_0 , но не заданы ни границы изменения параметров A, B, ни шаг H, и отсутствуют функциональные ограничения. Это соответствует задаче отыскания ближайшего безусловного минимума от X_0 в бесконечной области поиска D. Причем этом случае могут использоваться только регулярные методы поиска, т.к. в противном случае необходимо задавать величину шага H. ПР2=2, если не заданы границы A и B, шаг H, но задана начальная точка X_0 и функциональные ограничения вида $f_i(X) = 0$. Это случай отыскания ближайшего к X_0 условного минимума. Здесь применимы только регулярные алгоритмы с аналитическим вычислением производных и способных решать задачи с ограничениями указанного вида.

ПР2=3, если, кроме начальной точки X_0 , заданы ограничения вида $f_i(X) \leq 0$. Если же заданы ограничения обоих видов и X_0 (при отсутствии A, B и H), то ПР2=4, ПР2=5, если заданы X_0, A, B, H , но отсутствуют функциональные ограничения. Эта задача

отыскания ближайшего к X_0 условного или безусловного минимума в гиперпараллелепипеде. В этом случае возможно применение ПР2=6, если заданы только А и В. Это задача нахождения глобального безусловного или условного минимума в гиперпараллелепипеде. В этом случае применены только статистические методы поиска.

Таблица 4.5

i		Значения ПР2																													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27			
18	Значения Π_i					1	1	1	1	1	1	1	1	1																	
19						1	1	1	1	1	1	1	1	1																	
20		1	1	1	1	1																									
21						1					1	1	1	1																	
22																															
23																															
24																															
25																															
26																															
27																															

ПР2=7, если заданы А, В и ограничения $f_i(X)=0$. Это задача определения глобального минимума в области D, образованной перечислением гиперпараллелепипеда и ограничениями $f_i(X)=0$. Если заданы ограничения $f_i(X) \leq 0$, А и В, то ПР2=8, а в случае наличия обоих видов ограничений ПР2=9. Отличие в значениях ПР2 в связи с наличием или отсутствием того или иного вида ограничений связано с тем обстоятельством, что не все алгоритмы могут учитывать все виды ограничений.

ПР2=10, 11, 12, 13 соответственно, если заданы А, В и Н; А, В, Н и $f_i(X)=0$; А, В, Н и $f_i(X) \leq 0$; А, В, Н, $f_i(X)=0$, $f_i(X) \leq 0$.

Во всех этих случаях задача состоит в нахождении глобального условно или безусловного минимума в замкнутой области D.

Значения ПР2 от 14 до 23 включительно связано с наличием в оптимизационной задаче целочисленных переменных. Ситуации, влияющие на значения ПР2, в основном, соответствуют приведенным выше ситуациям с непрерывно изменяющимся переменными, с тем только отличием, что не рассматриваются ограничения вида $f_i(X)=0$, т.к. у автора нет данных об эффективных методах решения задач при наличии целочисленных переменных с подобными функциональными ограничениями. Кроме того, в вычислительный банк предлагаемой системы заложена разработанная авторам и программ [115], которая показала себя удовлетворительно при решении целочисленных и комплексных (т.е. при наличии как целочисленных, так и непрерывных переменных) задач с ограничениями вида $f_i(X) \leq 0$, тогда как при ограничениях $f_i(X)=0$ возникали значительные трудности вычислительного порядка.

Элементы B_3 и их функции сведем в табл. 4.6.

Таблица 4.6

Элемент	Функции
B_3 - банк признаков	F_3^I -хранение логической информации оптимизация признаковом пространстве; F_3^{II} -формирование значений признаков;

	F_3''' -передача значений признаков в Б ₁ , Б ₄ , Б ₅ .
Б ₃₋₁ – информационная часть	$F_{3-1}' \equiv F_3'$;
Б ₃₋₂ –операционная часть	$F_{3-2}' \equiv F_3'$, $F_{3-2}'' \equiv F_3''$.
Б ₃₋₂₋₁ – модуль формирования	$F_{3-2-1} \equiv F_{3-2}'$;
Б ₃₋₂₋₂ – модуль передачи	$F_{3-2-2} \equiv F_{3-2}''$;
Б ₃₋₂₋₃ - монитор	F_{3-2-3} - управление элементами Б ₃₋₂₋₁ и Б ₃₋₂₋₂

Табл. 4.6 представлен для анализа функции Б₃.
Функциональная схема Б₃ представлена на рис. 4.3.

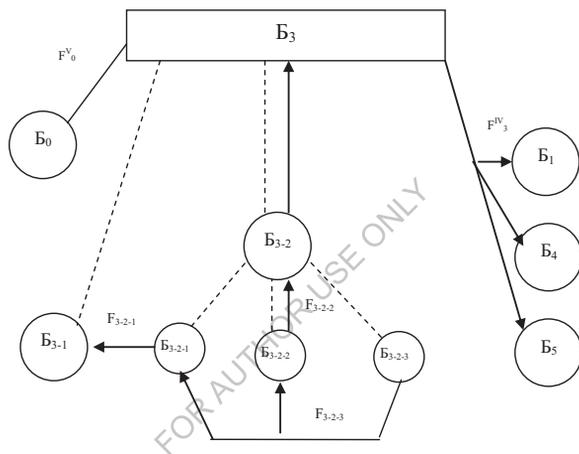


Рис. 4.3. Функциональная схема Б₃

4.3.2. Б₄ – банк моделей. Б₄ служит для исследования и идентификации математических моделей задач, т.е. его основные функции:

F_4' - идентификация моделей по признакам, сформированным в Б₃;

F_4'' - исследование моделей;

F_4''' - передача данных оптимизация моделях в Б₅.

Элементы второго уровня и их функции:

B_{4-1} – информационная часть, хранящая логическую информацию о моделях;

F_{4-1} – хранение информации о моделях;

B_{4-2} – операционная часть, производящая обработку B_{4-1} и выполняющая функции

$$F_{4-2}' \equiv F_4'; F_{4-2}'' \equiv F_4''; F_{4-2}''' \equiv F_4'''$$

B_{4-2} – делится на элементы третьего уровня;

B_{4-2-1} – модуль идентификации моделей по признакам;

B_{4-2-2} – блок исследования моделей;

B_{4-2-3} – оператор передачи данных в Б₅.

Блок исследования моделей, в свою очередь, делится на элементы четвертого уровня:

$B_{4-2-2-1}$ – модуль Монте-Карло, производящий наброс случайных равномерно распределенных точек по области поиска точек и вычисляющий значения целевой функции в них, приводя условные модели к безусловным при помощи штрафных функций.

$B_{4-2-2-2}$ – модуль формирования тестовых моделей;

$B_{4-2-2-3}$ – модуль оптимизации.

$B_{4-2-2-4}$ – модуль выбора тестовой модели, максимально приближенной к исследуемой модели.

Модель идентифицируется по следующим признакам:

1. Вид целевой функции;
2. Вид ограничений;
3. Наличие дополнительных сведений о целевой функции;
4. Наличие дополнительных сведений об ограничениях;
5. Количество переменных;
6. Количество ограничений.

Опишем подробнее метод идентификации моделей. Идентификатор модели – целочисленная переменная МОД – принимает значения от 1 до 144 в зависимости от значений признаков ПРЦ и ПР1. Значения МОД соответствуют порядковым номерам клеток матрицы, в которой строками служат значения признака ПРЦ (1,18), а столбцами – значения признака ПР1 (1,8). В табл. 4.7 показано соответствие значений МОД признакам ПРЦ и ПР1, а также, для наглядности номера алгоритмов, способных разрешить эти модели (см. B_5 - банк алгоритмов) .

Таблица 4.7

ПРЦ	ПР1							
	1.	2.	3.	4.	5.	6.	7.	8.
1.	¹ A ₁	² A _{10,12}	³ A ₁₂₋₁₆	⁴ A ₁₂₋₁₆	⁵ A ₁₂₋₁₆	⁶ A ₁₄₋₁₆	⁷ A ₁₆	⁸ -
2.	⁹ A ₁₂₋₁₆	¹⁰ A ₁₂₋₁₆	¹¹ A ₁₂₋₁₆	¹² A ₁₂₋₁₆	¹³ A ₁₂₋₁₆	¹⁴ A ₁₄₋₁₆	¹⁵ A ₁₆	¹⁶ A _{12,13}
3.	¹⁷ A ₁₂₋₁₆	¹⁸ A ₁₂₋₁₆	¹⁹ A _{7,8, 11}	²⁰ A ₁₂₋₁₆	²¹ A _{7,8,11-16}	²² A ₁₄₋₁₆	²³ A ₁₆	²⁴ A _{5,12,13}
4.	²⁵ A ₁₂₋₁₆	²⁶ A ₁₂₋₁₆	²⁷ A _{9,12-16}	²⁸ A _{9,12-16}	²⁹ A _{9,12-16}	³⁰ A ₁₄₋₁₆	³¹ A ₁₆	³² A _{12,13}
5.	³³ A ₁₂₋₁₆	³⁴ A ₁₂₋₁₆	³⁵ A ₁₂₋₁₆	³⁶ A ₁₂₋₁₆	³⁷ A ₁₂₋₁₆	³⁸ A ₁₄₋₁₆	³⁹ A ₁₆	⁴⁰ A _{2,4,12,13}
6.	⁴¹ A ₁₂₋₁₆	⁴² A ₁₂₋₁₆	⁴³ A ₁₂₋₁₆	⁴⁴ A ₁₂₋₁₆	⁴⁵ A ₁₂₋₁₆	⁴⁶ A ₁₄₋₁₆	⁴⁷ A ₁₆	⁴⁸ A _{3,12,13}
7.	⁴⁹ A ₁₂₋₁₆	⁵⁰ A ₁₂₋₁₆	⁵¹ A ₁₂₋₁₆	⁵² A ₁₂₋₁₆	⁵³ A ₁₂₋₁₆	⁵⁴ A ₁₄₋₁₆	⁵⁵ A ₁₆	⁵⁶ A _{12,13}
8.	⁵⁷ A ₁₂₋₁₆	⁵⁸ A ₁₂₋₁₆	⁵⁹ A ₁₂₋₁₆	⁶⁰ A ₁₂₋₁₆	⁶¹ A _{7,8,11-16}	⁶² A ₁₄₋₁₆	⁶³ A ₁₆	⁶⁴ A _{5,12,13}
9.	⁶⁵ A ₁₂₋₁₆	⁶⁶ A ₁₂₋₁₆	⁶⁷ A _{9,12-16}	⁶⁸ A ₁₂₋₁₆	⁶⁹ A ₁₂₋₁₆	⁷⁰ A ₁₄₋₁₆	⁷¹ A ₁₆	⁷² A _{12,13}
10.	⁷³ A ₁₂₋₁₆	⁷⁴ A ₁₂₋₁₆	⁷⁵ A ₁₂₋₁₆	⁷⁶ A ₁₂₋₁₆	⁷⁷ A ₁₂₋₁₆	⁷⁸ A ₁₄₋₁₆	⁷⁹ A ₁₆	⁸⁰ A _{2,4,12,13}
11.	⁸¹ A ₁₂₋₁₆	⁸² A ₁₂₋₁₆	⁸³ A ₁₂₋₁₆	⁸⁴ A ₁₂₋₁₆	⁸⁵ A ₁₂₋₁₆	⁸⁶ A ₁₄₋₁₆	⁸⁷ A ₁₆	⁸⁸ A _{3,12,13}
12.	⁸⁹ A ₁₄₋₁₆	⁹⁰ A ₁₄₋₁₆	⁹¹ A ₁₄₋₁₆	⁹² A ₁₄₋₁₆	⁹³ A ₁₄₋₁₆	⁹⁴ A ₁₄₋₁₆	⁹⁵ A ₁₆	⁹⁶ A ₁₄₋₁₆
13.	⁹⁷ A ₁₃₋₁₇	⁹⁸ A ₁₃₋₁₇	⁹⁹ A ₁₃₋₁₇	¹⁰⁰ A ₁₃₋₁₇	¹⁰¹ A ₁₃₋₁₇	¹⁰² A ₁₄₋₁₇	¹⁰³ A ₁₆	¹⁰⁴ A _{13,17}
14.	¹⁰⁵ A _{6,13-17}	¹⁰⁶ A ₁₃₋₁₇	¹⁰⁷ A ₁₃₋₁₇	¹⁰⁸ A ₁₃₋₁₇	¹⁰⁹ A _{7,8,11,13-}	¹¹⁰ A ₁₄₋₁₇	¹¹¹ A ₁₆	¹¹² A _{5,13,17}
15.	¹¹³ A ₁₃₋₁₇	¹¹⁴ A ₁₃₋₁₇	¹¹⁵ A _{9,13-17}	¹¹⁶ A ₁₃₋₁₇	¹¹⁷ A ₁₃₋₁₇	¹¹⁸ A ₁₄₋₁₇	¹¹⁹ A ₁₆	¹²⁰ A _{13,17}
16.	¹²¹ A ₁₃₋₁₇	¹²² A ₁₃₋₁₇	¹²³ A ₁₃₋₁₇	¹²⁴ A ₁₃₋₁₇	¹²⁵ A ₁₃₋₁₇	¹²⁶ A ₁₄₋₁₇	¹²⁷ A ₁₆	¹²⁸ A _{2,4,13,17}
17.	¹²⁹ A ₁₃₋₁₇	¹³⁰ A ₁₃₋₁₇	¹³¹ A ₁₃₋₁₇	¹³² A ₁₃₋₁₇	¹³³ A ₁₃₋₁₇	¹³⁴ A ₁₄₋₁₇	¹³⁵ A ₁₆	¹³⁶ A _{3,13,17}
18.	¹³⁷ A ₁₆	¹³⁸ A ₁₆	¹³⁹ A ₁₆	¹⁴⁰ A ₁₆	¹⁴¹ A ₁₆	¹⁴² A ₁₆	¹⁴³ A ₁₆	¹⁴⁴ A ₁₆

В табл. 4.7 внутри клеток слева в верхнем углу проставлено значение МОД. В клетках также поставлены номера алгоритмов, способных решить данную модель, взятых из табл. 4.10.

Метод исследования моделей описан выше, поэтому здесь на нем останавливаться не будем. Анализ функций B_4 представлен в табл. 4.8, а функциональная схема – на рис. 4.4.

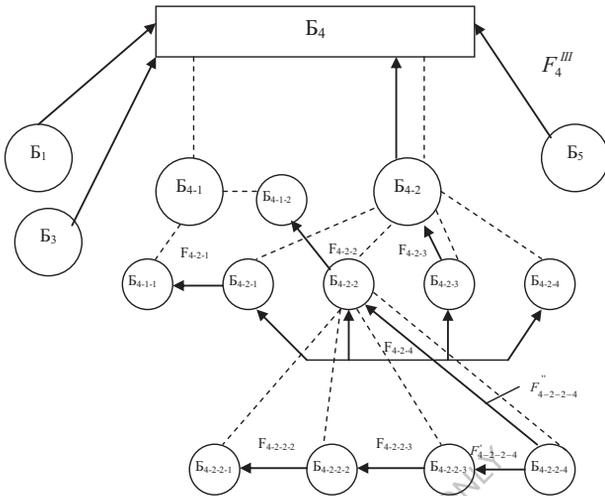


Рис.4.4. Функциональная схема B_4

Таблица 4.8

Элемент	Функции
B_4	F_4^I - идентификация; F_4^{II} -исследование; F_4^{III} -передача данных;
B_{4-1}	F_{4-1} – хранение информации
B_{4-2}	$F_{4-2}^I \equiv F_4^I$; $F_{4-2}^II \equiv F_4^{II}$; $F_{4-2}^III \equiv F_4^{III}$.
B_{4-2-1} – модуль идентификации по признакам	$F_{4-2-1} \equiv F_{4-2}^I$;
B_{4-2-2} – блок исследования	$F_{4-2-2} \equiv F_{4-2}^{II}$;
B_{4-2-3} - оператор передачи	$F_{4-2-3} \equiv F_{4-2}^{III}$;
B_{4-2-4} – монитор	F_{4-2-4} - управление
$B_{4-2-2-1}$ – модуль Монте –Карло	$F_{4-2-2-1}^I$ -наброс точек, равномерно распределенных по области; $F_{4-2-2-1}^{II}$ -переход к безусловной модели (штрафные функции); $F_{4-2-2-1}^{III}$ - вычисления безусловной модели в точках

Б ₄₋₂₋₂₋₂ – блок формирования тестовой модели	$F_{4-2-2-2}$ - формирование тестовой модели
Б ₄₋₂₋₂₋₃ – блок оптимизации	$F_{4-2-2-3}$ - оптимизация тестовых моделей
Б ₄₋₂₋₂₋₄ – модуль выбора тестовой модели;	$F'_{4-2-2-4}$ – определение тестовой модели, максимально приближенной к исследуемой модели. $F''_{4-2-2-4}$ – передача данных в Б ₄₋₂₋₂

4.4. Построение банка алгоритмов и вычислительного банка

4.4.1. Б₅ - банк алгоритмов. Банк алгоритмов служит для хранения всех необходимых сведений об имеющихся алгоритмах для решения экстремальных задач, а также для выбора оптимального алгоритма для каждой конкретной модели.

Наиболее общие методы и соответствующие характеристики им алгоритмов решения задач математического программирования заложенные в Б₅, представлены в табл. 4.9.

Таблица 4.9

№№ ПП	Название метода (алгоритма)	обозначение	примечание
1	Симплекс алгоритм	A ₁	БУ
2	Алгоритм наискорейшего спуска	A ₂	#
3	Алгоритм Ньютона	A ₃	#
4	Алгоритм двойственных направлений	A ₄	#
5	Алгоритм сопряженных градиентов	A ₅	#
6	Алгоритм сопряженных градиентов	A ₆	У
7	Алгоритм возможных направлений	A ₇	#
8	Алгоритм условного градиента	A ₈	#
9	Алгоритм Ньютона	A ₉	#
10	Алгоритм отсекающей гиперплоскости	A ₁₀	#
11	Алгоритм линеаризации	A ₁₁	#
12	Алгоритм П ₁ случайного поиска	A ₁₂	#
13	Алгоритм П ₂ случайного поиска	A ₁₃	#
14	Алгоритм ГПЗ глобального поиска	A ₁₄	#
15	Алгоритм ГП4 (спец. алгоритм)	A ₁₅	#
16	Алгоритм ГП5 дискретные модели	A ₁₆	#
17	Алгоритм овражный	A ₁₇	#

Опишем краткие характеристики и особенности алгоритмов:

1. Симплекс алгоритм – универсальный алгоритм для решения задач линейного программирования.

2. Метод наискорейшего спуска – сравнительно простой и может применяться для весьма широкого класса функций. Недостаток – плохая сходимость для овражных задач и в районе экстремума.

Условия, требуемые для сходимости алгоритма к стационарной точке:

1) $U(X)$ ограничена снизу;

2) $U'(X)$ удовлетворяет условия Липшица

$$\|U'(X) - U'(Y)\| \leq R \|X - Y\|;$$

к точке минимума:

3) $U(X)$ дважды непрерывно дифференцируема

4) матрица вторых производных удовлетворяет условиям:

$$m\|Y\|^2 \leq (U'(X)Y, Y) \leq M\|Y\|^2, \quad M \geq m > 0.$$

3. Метод Ньютона с регулировкой шага – может применяться лишь для тех функций, у которых существует обратимая ограниченная матрица вторых производных целевой функции. Скорость сходимости выше, чем у градиентных методов, на затраты вычислений на один шаг гораздо больше. Также методу Ньютона требуется больше машинной памяти.

4. Метод двойственных направлений – может применяться для минимизации строго выпуклых гладких функций независимо от выбора начального приближения. Скорость сходимости сверхлинейная и близка к методу Ньютона, однако требуется меньше вычислений на один шаг. Недостаток – требуется большая память.

5. Метод сопряженных направлений – применяется для минимизации выпуклых функций. Наиболее эффективно его применение к минимизации квадратичных функций, минимум которых он определяет за число шагов не превышающее N (N количество переменных). Скорость сходимости у этого метода выше, чем у градиентных, но ниже чем у Ньютоновских.

Методы для решения задач с ограничениями. 6. Метод сопряженных градиентов – применяется к задачам квадратичного программирования, для чего применяется проектирование градиентов на подпространство.

7. Метод возможных направлений – служит для решения задачи выпуклого программирования.

Требуемые условия:

1) $f_i(X)$ – ограничения – выпуклые, непрерывно дифференцируемые функции;

2) градиенты функции $f_i(X)$ удовлетворяют условия Липшица

$$\|f_i'(X) - f_i'(Y)\| \leq R\|X - Y\|$$

3) ограниченность градиентов $\|f_i'(X)\| \leq K$ для всех X . К недостаткам метода возможных направлений можно отнести то обстоятельство, что его скорость сходимости неизвестна.

8. Метод условного градиента – применяется для решения задач выпуклого программирования с такими областями поиска D , в которых задача минимизации линейной функции решается без труда.

Требует:

а) непрерывно дифференцируемой целевой функции в компактной выпуклой области D ;

б) удовлетворения градиентов целевой функции условию Липшица для всех точек области D . Скорость сходимости алгоритма невелика.

9. Метод Ньютона сходен с методом условного градиента, только требует на каждом шаге минимизации квадратичной функции в области D . Скорость сходимости высокая, если задача минимизации квадратичной функции решается просто.

10. Метод отсекающей гиперплоскости предназначен для решения задачи выпуклого программирования с ограничениями вида $f_i(X) \leq 0$, предусматривает решения вспомогательной задачи линейного программирования на каждом шаге. Требует значительной памяти. Скорость сходимости невелика.

11. Метод линеаризации – предназначен для решения общей задачи математического программирования.

Основные требования:

1. Непрерывная дифференцируемость $f_i(X)$;

2. Градиенты $f'_i(X)$ удовлетворяют условию Липшица.

Предположения, положенные в основу метода [19,85,161,169]:

Положим $F(X) = \max f_i(X)$;

$$I_\delta(X) = \{i \in I : f_i(X) \geq F(X) - \delta\}, \delta \geq 0.$$

1) $F(X) \geq 0$ при всех X ;

2) существуют такие константы $N > 0, \delta > 0$, что

а) множество $\Omega_N = \{X : U(X) + NF(X) \leq C_0\}$, $C_0 = U(X_0) + NF(X_0)$ ограничено;

б) градиенты функций $f_i(X)$ в Ω_N удовлетворяют условию Липшица

$$\|f'_i(X_1) - f'_i(X_2)\| \leq L \|X_1 - X_2\|;$$

в) задача квадратичного программирования

$$\min(U'(X_0), P) + \frac{1}{2} \|P\|^2, (f'_i(X), P) + f_i(X) \leq 0; i \in I_\delta(X)$$

разрешима относительно $P \in E^n$ при любом $X \in \Omega_N$;

г) существуют такие множители Лагранжа

$$U^i(X), i \in I_\delta(X), \text{ что } \sum_{i \in I_\delta(X)} U^i(X) \leq N$$

здесь $\|P\|$ – обозначает евклидову норму вектора P .

Алгоритмы 13-16 описаны в п.3.2–3.4.

Все они используют методы случайного поиска экстремума и отличаются друг от друга степенью сложности и организации поиска и, соответственно, сложностью решаемых задач. Приведем краткие характеристики. П-1 – случайный поиск локального условного или безусловного экстремума с линейной тактикой. Это наиболее простой алгоритм, требующий минимальной памяти компьютера. Успешно применяется в задачах со слабо выраженной нелинейностью целевых функций. П-2 – поиск локального экстремума с самообучением, введение которого в алгоритм позволило резко ускорить сходимость по сравнению с П1 особенно в задачах с ярко выраженной нелинейностью целевых функций.

ГП-3 – алгоритм поиска глобального экстремума в области D , в котором помимо самообучения применено несколько эвристических приемов, позволивших сократить непроизводительные затраты на поиск глобального экстремума.

ГП-4 – алгоритм глобального поиска, специально предназначенный для оптимизации конструкций по весу из условий прочности, устойчивости, жесткости. Это алгоритм адаптирующийся к меняющейся обстановке во время прохождения поиска экстремума, что позволило значительно сократить затраты на поиск.

ГП-5 – модификации алгоритма ГП-3, способная производить оптимизацию при наличии целочисленных переменных.

Эти алгоритмы очень универсальны, так как не накладывают на вид целевой функции и ограничения никаких условий, как регулярные алгоритмы.

Обратимся теперь к описанию построения и работы банка алгоритмов.

Выделим основные функции Б₅. К ним относятся:

F_5^I – выбор алгоритма для решения задачи;

F_5^II – параметрическая оптимизация алгоритма (настройка его на задачу);

F_5^III – построение оптимизирующей последовательности алгоритмов;

F_5^{IV} – передача информации в Б₆.

Элементы первого уровня:

Б₅₋₁ – информационная часть, обеспечивающая F_{5-1} – хранение информации о характеристиках алгоритмов;

B_{5-2} – операционная часть, которая обеспечивает выполнение следующих функций:

F'_{5-2} – выбор алгоритма;

F''_{5-2} – поиск оптимальных значений параметров алгоритма в B_{5-1} ;

$F'''_{5-2} \equiv F''_5, F''_{5-2}$ – построение ε – последовательности алгоритмов;

F''''_{5-2} – построение ε' – последовательности алгоритмов;

$F''''_{5-2} \equiv F''''_5$.

Элементы первого уровня состоят из следующих элементов второго уровня:

B_{5-1-1} – массив применимости алгоритмов;

B_{5-1-2} – массив оптимальных параметров алгоритмов;

B_{5-1-3} – массив ε – последовательностей алгоритмов для классов задач;

B_{5-1-4} – массив ε' – последовательностей алгоритмов для классов задач;

B_{5-2-1} – блок выбора алгоритма;

B_{5-2-2} – блок оптимизации алгоритма;

B_{5-2-3} – блок построения ε – последовательностей алгоритмов;

B_{5-2-4} – блок построения ε' – последовательностей алгоритмов;

B_{5-2-5} – монитор, управляющий элементами второго уровня.

Алгоритм, необходимый для решения задачи, выбирается с учетом следующих вопросов:

- 1) разрушаемость модели;
- 2) количество и характер дополнительных требований к модели, необходимых для успешной работы алгоритма;
- 3) требуемый объем памяти компьютер;
- 4) скорость действия.

Выше перечисленные вопросы отражают как применимость алгоритма к решению данной задачи (пункты 1,2), так и оптимальность (пункты 3,4).

Для решения этих вопросов применяются признаки $PR2$, $PR N$, $PR V$ (B_{5-1-2}), $PR2$ и $PR N$ выявляют применимость алгоритма, а $PR V$, $PR \varepsilon$ оптимальность.

Вопрос применимости алгоритмов должен решаться поэтапно.

1 этап. Применимость алгоритмов к моделям с учетом $PR2$ показана в табл. 4.10.

2 этап. Каждому алгоритму соответствует свое значение $PR N$ равное максимальному количеству параметров, которое может взять алгоритм. Таким образом, после определения алгоритмов, способных решить заданную модель с учетом $PR2$, идет сопоставление количества заданных параметров $K.З.П.$ с $PR N$ и выбирается алгоритм, у которого $K.З.П. \leq PR N$.

Таблица 4.10

A	PR2	МОД
1	1:13	1
2	1	40,80,128
3	1	48,88,136
4	1	40,80,128
5	1	24,64,112
6	1-5	17,105
7	1,3,5	19,21,61,109
8	1,3,5	19,21,61,109
9	1,3,5	27,28,29,67,115
10	1,3,5	2
11	1-13	19,21,61,109

12	1,3,5,6,8,10,12	9-13,16-21,24-29,32-37,40-45,48-53,56-61,64-69,72-77,2,3,4,5,80-85,88
13	1,3,5,6,8,10,12	3,4,5 те же, что А12,97-101,104-109,112-117,120-125,128-133,136
14	1,3,5,6,8,10,12	3,4,5,9-14,17-22,25-30,33-38,41-46,49-54,57-62,65-70,73-78,81-86,89-94,96-102,105-110,113-118,121-126,129-134
15	1,3,5,6,8,10,12	3,4,5,9-14,17-22,25-30,33-38,41-46,49-54,57-62,65-70,73-78,81-86,89-94,96-102,105-110,113-118,121-126,129-134
16	1,3,5,6,8,10,12,14-23	А14,7,15,23,31,39,47,55,63,71,73,87,95,103,111,119,127,135,137-144
17	1,3,5,6,8,10,12	97-102,104-110,112-118,120-126,128-134,136

Если нет дополнительного сигнала из B_4 об оптимизации алгоритма, то данные о нем передаются в B_6 . В случае же необходимости производится поиск оптимальных значений параметров алгоритма в B_{5-2} и передача их в B_6 . При отсутствии таковых производится оптимизация алгоритма, построение оптимизирующих последовательностей (в случае, если применимы несколько алгоритмов, с чем подробно сказано в п.4.4), занесение результатов в B_{5-1} (т.о. производится самообучение системы) и передача их в B_6 для формирования рабочей программы.

Анализ функций B_5 представлен в табл. 4.11, а функциональная схема – на рис 4.5.

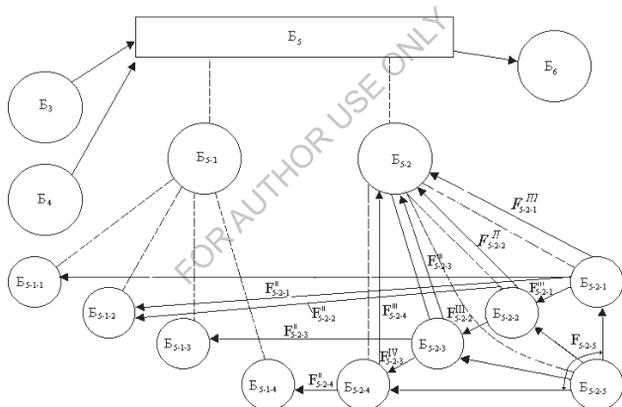


Рис. 4.5. Функциональная схема B_5

Таблица 4.11

Элемент	Функции
1	2
B_5 – банк алгоритмов	F'_5 – выбор алгоритма для разрешения модели; F''_5 – оптимизация алгоритма; F'''_5 – построение последовательности алгоритмов; F^{IV}_5 – передача информации в B_6
B_{5-1} – информационная часть	F_{5-1} – передача информации о характеристиках алгоритма
B_{5-2} – операционная часть	$F'_{5-2} \equiv F'_5$

	F_{5-2}'' – поиск оптимальных значений параметров алгоритма в B_{5-1} F_{5-2}''' – параметрическая оптимизация алгоритма. F_{5-2}^{IV} – построение ε последовательности F_{5-2}^V – построение ε' последовательности алгоритмов. F_{5-2}^{VI} – передача информации в B_6 .
B_{5-1-1} – массив применимости алгоритма;	F_{5-1-1} – хранение информации по применимости алгоритмов.
B_{5-1-2} – массив оптимальных параметров.	F_{5-1-2} – хранение оптимальных значений параметров для классов задач;
B_{5-1-3} – массив ε – последовательностей;	F_{5-1-3} – хранение информации об ε – последовательностях алгоритмов в для классов задач;
B_{5-1-4} – массив ε' – последовательностей;	F_{5-1-4} – хранение информации об последовательностях алгоритмов для классов задач.
B_{5-2-1} – блок выбора	F'_{5-2-1} – выбор алгоритмов по применимости; F''_{5-2-1} – поиск оптимальных значений параметров алгоритмов в B_{5-1-2} ; F'''_{5-2-1} – передача информации в B_{5-2} (для B_6); F^{IV}_{5-2-1} – передача информации в B_{5-2-2} для оптимизации параметров.
B_{5-2-2} – блок оптимизации	F'_{5-2-2} – формирование оптимизационной модели алгоритма; F''_{5-2-2} – параметрическая оптимизация алгоритмов;
	F'''_{5-2-2} – Передача информации в B_{5-2} для B_6 и B_{5-1-2} ; F^{IV}_{5-2-2} – передача информации в B_{5-2-3} для построения ε - последовательности.
B_{5-2-3} – Блок ε	F'_{5-2-3} – построения ε - последовательности; F''_{5-2-3} – передача информации в B_{5-1-3} ; F'''_{5-2-3} – передача информации в B_{5-2} (для B_6).
B_{5-2-4} – Блок ε'	F'_{5-2-4} – построения ε' - последовательности; F''_{5-2-4} – передача информации в B_{5-1-4} ; F'''_{5-2-4} – передача информации в B_{5-2} (для B_6).
B_{5-2-5} – монитор	F_{5-2-5} – Управление B_{5-2-i} ($i = \overline{1,5}$)

4.4.2. B_6 – вычислительный банк. Вычислительный банк предназначен для хранения всех поисковых моделей, соответствующих различным алгоритмам банка алгоритмов, выбора соответствующего модуля после получения информации из банка алгоритмов, формирования рабочей программы решения задачи с последующим счетом после получения из банка данных.

Таким образом функции Б₆:

F'_6 – хранение модулей;

F''_6 – формирование рабочей программы;

F'''_6 – подстановка исходных данных и значений параметров алгоритмов;

F^{IV}_6 – передача на счет.

Эти функции выполняют :

Б₆₋₁ – информационная часть;

Б₆₋₂ – операционная часть, которая, в свою очередь, состоит из элементов второго уровня:

Б₆₋₂₋₁ – блок формирования рабочей программы;

Б₆₋₂₋₂ – блок вызова и постановки данных;

Б₆₋₂₋₃ – монитор, управляющий Б₆₋₂₋₁ и Б₆₋₂₋₂.

При получении из Б₅ номера выбранного алгоритма Б₆ организует обращение к процедуре, реализующей указанный алгоритм с введением всех необходимых исходных данных и с последующей передачей, на счет. После прохождения счета на печать выводится информация оптимизация всех найденных минимумов: значение целевой функции в них и значения оптимальных параметров. Кроме того, в целях наглядности и для контроля работы системы на печать выводится информация оптимизация значениями всех признаков (*ПРЦ, ПР1, ПР2*) идентификатора модели *МОД* и номере выбранного алгоритма.

В поисковых модулях, соответствующих алгоритмам в случае не задания величин рабочего шага *H* предусмотрено задание этих величин в зависимости от границ *A* и *B*. Иначе говоря компоненты h_i вектора *H* получают значения

$$h_i = \frac{b_i - a_i}{20}.$$

В случае, если для данной задачи по совокупности всех признаков *ПРЦ, ПР2, ПРН, ПРV, ПРЭ* не имеется нужный алгоритм, то система выдает на печать: «**нужен новый алгоритм или изменить постановку задачи**» и прекращает счет.

Анализ функций Б₆ приведен в табл. 4.12.

Таблица 4.12

Элемент	Функции
Б ₆ - вычислительный банк	F'_6 - Хранение моделей ПП; F''_6 - Формирование рабочей программы; F'''_6 - Подстановка данных; F^{IV}_6 - Передача на счет
Б ₆₋₁ - информационная часть; Б ₆₋₂ - операционная часть; Б ₆₋₂₋₁ - блок формирования; Б ₆₋₂₋₂ - блок вызова данных	$F_{6-1} \equiv F'_6$; $F'_{6-2} \equiv F''_6$; $F''_{6-2} \equiv F'''_6$; $F'''_{6-2} \equiv F^{IV}_6$. $F'_{6-2-1} \equiv F'_{6-2}$; $F''_{6-2-1} \equiv F'''_{6-2}$. F'_{6-2-2} - вызов данных из Б ₁ , Б ₅ ; F''_{6-2-2} - подстановка данных в рабочую программу
Б ₆₋₂₋₃ - Монитор	F_{6-2-3} - управление Б ₆₋₂₋₁ , Б ₆₋₂₋₂

Функциональная схема Б₆ представлена на рис. 4.6.

Глава 5. РЕЗУЛЬТАТЫ РАСЧЕТА ПО ОПТИМИЗАЦИИ ТОНКОСТЕННЫХ ЭЛЕМЕНТОВ ИНЖЕНЕРНЫХ КОНСТРУКЦИЙ

5.1. Весовая оптимизация инженерных конструкций типа пластин и оболочек

Весовая оптимизация инженерных конструкций типа пластин и оболочек предусматривает минимизацию веса этих конструкций при воздействии на них заданных систем внешних сил с соблюдением условий сохранения необходимой прочности, устойчивости, жесткости конструкций.

Задача в общем виде записывается так:

$$\left. \begin{aligned} F(x) = G &\rightarrow \min; \\ G_{\max} &\leq [\sigma]; \\ P_{\max} &\leq P_{кр}; \\ U_{\max} &\leq [U] \end{aligned} \right\} \quad (5.1)$$

где G – вес конструкции; σ_{\max} – максимальные напряжения в конструкции;

$[\sigma]$ – допускаемые напряжения; P_{\max} – максимальная сжимающая сила; $P_{кр}$ – критическая сила; U_{\max} – максимальные перемещения в конструкции; $[U]$ – допускаемые перемещения.

Проверка ограничений (5.1) возможна лишь после решения системы дифференциальных уравнений равновесия или движения рассматриваемых конструкций с соответствующими начальными или граничными условиями. Рассмотрим более подробно методы решения прямой задачи расчета.

Известно, что уравнения равновесия, колебания и устойчивости анизотропных пластин относительно моментов соответственно имеют вид [4]:

$$\frac{\partial^2 M_1}{\partial x^2} + 2 \frac{\partial^2 M_{12}}{\partial x \partial y} + \frac{\partial^2 M_2}{\partial y^2} = q_1(x, y), \quad (5.2)$$

$$\frac{\partial^2 M_1}{\partial x^2} + 2 \frac{\partial^2 M_{12}}{\partial x \partial y} + \frac{\partial^2 M_2}{\partial y^2} + h(\sigma_x \frac{\partial^2 W}{\partial x^2} + \sigma_y \frac{\partial^2 W}{\partial y^2} + 2\sigma_{xy} \frac{\partial^2 W}{\partial x \partial y}) = 0, \quad (5.3)$$

$$\frac{\partial^2 M_1}{\partial x^2} + 2 \frac{\partial^2 M_{12}}{\partial x \partial y} + \frac{\partial^2 M_2}{\partial y^2} + m \frac{\partial^2 W}{\partial t^2} = q_2(x, y, t). \quad (5.4)$$

Здесь W – прогиб пластины, M_1 , M_{12} , M_2 – изгибающие и крутящие моменты,

$m = \gamma h / g$, γ – вес единицы объема, g – ускорение притяжения пластины, h – толщина.

Соотношения для M_1 , M_{12} , M_2 в случаях, когда пластины являются изотропными, ортотропными и анизотропными, приведены в [4]. Подставляя в (5.2), (5.3) соотношения M_1 , M_{12} , M_2 в случае, когда пластина является изотропной, ортотропной или в других случаях анизотропии, можно получить соответствующие уравнения. Эти уравнения

приведены во многих руководствах по теории упругости. Поэтому нет необходимости здесь приводить их вид.

В связи с этим для решения прямой задачи статистического расчета арок и осесимметричных конических оболочек нами применен конечно-разностный метод. Для расчета открытой цилиндрической оболочки применен метод Ритца, поскольку метод конечных разностей приводит в этом случае к слишком громоздкой системе алгебраических уравнений, что вызывает серьёзные трудности, как при решении прямой задачи, так и при оптимизации открытых цилиндрических оболочек [48,156,158,159].

Функция цели:

$$F(x) = \int_{\alpha} \int_{\beta} h(\alpha, \beta) R d\alpha d\beta \quad (5.5)$$

Для незамкнутых оболочек типа сводов считаются заданными: а) граничные условия; б) длина перекрытия – a ; в) ширина перекрытия – b ; г) материал оболочки: E - модуль упругости; ν - коэффициент Пуассона; γ - удельный вес; $[\sigma]$ - допускаемые напряжения; $[U]$ - допускаемые перемещения (если требуется выполнение ограничений по прочности и жесткости); д) система внешних нагрузок; е) другие ограничения (например: конструктивные, технологические и т.д.), если требуется их удовлетворение.

Оптимизируемыми параметрами являются параметры, определяющие закон изменения толщины оболочки, угол раствора оболочки, определяющий степень её крутизны.

Приведем результаты расчета задачи оптимизации цилиндрической оболочки, прямоугольной в плане, шарнирно-опертой по всему контуру, находящейся под равномерно распределенной нормальной нагрузкой интенсивностью q . Толщина оболочки постоянная $h = const$. Физические характеристики материала оболочки:

$E = 2 \cdot 10^6 \text{ кг} / \text{см}^2$; $[\sigma] = 2000 \text{ кг} / \text{см}^2$; $\nu = 0,5$; геометрические характеристики оболочки: $a = 150 \text{ см}$; $b = 100 \text{ см}$, нагрузка $q = 1 \text{ кг} / \text{см}^2$. Оптимизируемые параметры h, β_0 .

Ограничения на параметры:

$$\frac{\pi}{10} \leq \beta_0 \leq \pi;$$

$$0,1 \text{ см} \leq h \leq 3 \text{ см}$$

Минимизируемая функция – площадь поперечного сечения

$$S = R \cdot h \cdot \beta_0.$$

На конструкцию наложены ограничения:

$$\sigma_i \leq [\sigma],$$

где σ_i - интенсивность напряжений, определяемая по формуле

$$\sigma_i = \sqrt{(\sigma_{11} - \sigma_{12})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 + 6\tau_{23}^2} \quad (5.6)$$

Напряжения $\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \tau_{23}$ определяются после решения уравнений (5.6) методом Ритца. В качестве координатных функций выбираются балочные функции, которые в случае шарнирного опирания оболочки имеют вид

$$\begin{aligned}
 U_{mm} &= \cos \frac{n\pi\alpha}{\alpha_0} \sin \frac{m\pi\beta}{\beta_0}; \\
 V_{mm} &= \sin \frac{n\pi\alpha}{\alpha_0} \cos \frac{m\pi\beta}{\beta_0}; \\
 W_{mm} &= \sin \frac{n\pi\alpha}{\alpha_0} \cos \frac{m\pi\beta}{\beta_0}
 \end{aligned}
 \tag{5.7}$$

Оптимизация проводится при помощи алгоритма ГП-3 с точностью $\varepsilon \approx 2\%$ [4]. Результаты расчетов приведены в табл. 5.1.

Таблица 5.1

Результаты расчета оптимизации

В лок. миним.	S, см ³	h, см	β_0 , рад	σ_i , кг/см ²	Шаги
1	125,0179	0,956369	2,467197	1990	52
2	126,6418	1,125607	1,662033	1977	28
3	114,072	0,8343	2,6613	1989	39
4	165,1633	1,53425	1,320312	1993	42

На рис. 5.1, представлены кривые $\sigma_i(\alpha, \beta)$, соответствующие полученным минимумам.

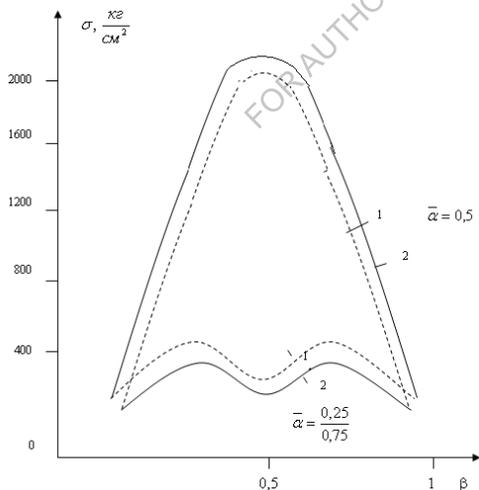


Рис. 5.1. Кривые $\sigma_i(\alpha, \beta)$, соответствующие полученным минимумам

По результатам решения задачи, видно, что применение цилиндрических оболочек переменной толщины позволяет снизить вес конструкции примерно на 14%, отсюда вывод о том, что имеется оптимизации веса, т.е. существенном снижении веса конструкции.

Необходимость оптимизации конструкций со всей очевидностью прослеживается в результатах решения приведенных задач. В то время как во всех найденных минимумах рассматриваемые конструкции находились на границе прочности или устойчивости, значения их весов значительно отличались, обе конструкции, обладая одинаковым запасом прочности, отличаются по весу почти в 20 %. Например, применение для оболочек переменной толщины (в виде различных законов $h(\beta)$ – для цилиндрических оболочек позволило в ряде случаев снизить вес конструкции ~14% (для цилиндрических оболочек) по сравнению с оболочками постоянной толщины, что указывает на оптимизацию - целесообразность применения переменной толщины, постановки и решения задач оптимизации при проектировании специальных оболочек минимального веса [7].

Задачи оптимизации инженерных конструкций типа пластин и оболочек сложны. Особую трудность представляет учет ограничений вследствие их разнообразия. Здесь и простейшие ограничения на параметры, типа $Q_i \leq x_i \leq b_i$, и функциональные (по прочности, устойчивости, жесткости). Причем в большинстве случаев для вычисления одного числа G_{max} необходимо решать систему дифференциальных уравнений в частных производных с соответствующими граничными условиями, что представляет определенные трудности. Оптимизируемые параметры могут меняться как непрерывно, так и дискретно. Целевая функция может быть также занята в неявном виде.

При весовой оптимизации конструкций время, требуемое на проверку ограничений, в основном, на несколько порядков больше, чем на вычисление целевой функции – веса конструкции, что предопределяет необходимость дифференцированного подхода к этим вычислениям с целью максимального сокращения количества проверки ограничений при сохранении необходимой надежности и точности нахождения оптимума. Многоэкстремальность задач оптимизации конструкций, о чем заявил ряд ученых, еще раз подтверждена полученными результатами: при решении задач найдено по несколько минимумов.

5.2. Оптимизация замкнутых цилиндрических оболочек, нагруженных осевой сжимающей силой

Рассмотрим изотропную круговую цилиндрическую оболочку заданной длины L и материала (задан модуль упругости E , допускаемые напряжения $[\sigma]$, удельный вес γ и коэффициент Пуассона - ν), нагруженную осевой сжимающей силой N . Нужно выбрать толщину оболочки δ и радиус срединной поверхности R так, чтобы выдерживая заданную нагрузку N , оболочка была минимального веса G . Таким образом, нужно найти минимум функции

$$G = 2\pi\gamma LR\delta \quad (5.8)$$

при условиях:

$$0,01\text{см} \leq \delta \leq 2,0\text{см};$$

$$1\text{см} \leq R \leq 100\text{см};$$

$$P_1^* = 2\pi E \delta^2 \left[3(1 - \nu^2) \right]^{\frac{1}{2}} \geq N; \quad (5.9)$$

$$P_2^* = \pi^3 E \delta R^3 L^{-2} \geq N;$$

$$P_3^* = 2\pi R \delta [\sigma] \geq N$$

Здесь P_1^* - критическая нагрузка потери устойчивости идеальной цилиндрической оболочки; P_2^* - критическая нагрузка потери устойчивости оси оболочки (считаем оболочку шарнирно-опертой по торцам); $\frac{N}{2\pi R \delta}$ - напряжения сжатия.

Ввиду того, что качество изготовления оболочки влияет на её несущую способность, представляет интерес учета влияния начальных несовершенств на критические нагрузки. При этом в ограничениях (5.9) в левую часть третьего неравенства вместо критических усилий идеальной оболочки подставить выражение, в котором учтено влияние начальных несовершенств на величину критического усилия [62]:

$$P_1^* = \frac{2\pi E \delta^2}{\sqrt{3(1-\nu^2)}} \left[\left(1 + K \frac{R}{\delta} \right)^{\frac{1}{2}} - \left(K \frac{R}{\delta} \right)^{\frac{1}{2}} \right] \geq N \quad (5.10)$$

Здесь коэффициент K учитывает качество изготовления оболочки.

Примем, как и в [53] $K=0,005$.

Подобную задачу впервые решили И.Н.Гинзбург и С.Н. Кан [34,35] при помощи обобщения метода множителей Лагранжа на случай, когда ограничения заданы в виде неравенств.

При этом авторы указали, что применение метода множителей Лагранжа для решения подобных задач не всегда является целесообразным, так как этот метод приводит к большому вычислительным трудностям, связанным с необходимостью в общем случае решать системы нелинейных уравнений.

Авторы решили численный пример при следующих физических и геометрических характеристиках оболочки:

$$N = 100m; E = 8,16 \cdot 10^5 \text{ кгс/см}^2; \gamma = 0,0078 \text{ кгс/см}^2;$$

$$[\sigma] = 1620 \text{ кгс/см}^2; \nu = 0,3; L = 300 \text{ см.}$$

Получены следующие результаты:

а) для идеальной оболочки можно брать любые δ и R , удовлетворяющие уравнению

$$\delta \cdot R = 9,8 \quad (5.11)$$

и ограничениям

$$\begin{aligned} 0,18 \text{ см} &\leq \delta \leq 1,63 \text{ см} \\ 6 \text{ см} &\leq R \leq 54,5 \text{ см} \end{aligned} \quad (5.12)$$

$$\text{При этом } G_{\min} = 2\pi \cdot 9,8 \cdot L \cdot \gamma = 144,1 \text{ кг}$$

б) для оболочки с начальными несовершенствами уравнение (5.11) остается, а ограничения:

$$\begin{aligned} 0,067 \text{ см} &\leq \delta \leq 0,262 \text{ см} \\ 37,4 \text{ см} &\leq R \leq 146,5 \text{ см} \end{aligned} \quad (5.12)$$

Вес оболочки при этом также равен 144,1 кг.

Эта же задача была решена И.М. Почтманом и Г.В. Филатовым пр помощи метода случайного поиска. Они применили так называемый «независимый» алгоритм глобального поиска с адаптацией [119]. Поиск они проводили в области, образованной ограничениями (6.26). весь процесс поиска был разбит на 9 этапов, в каждом из которых производилось по 500 независимых испытаний, равномерно распределенных в области поиска. После каждого этапа выбиралась наилучшая проба с наименьшим значением функции (6.22), образовывалась новая область поиска с центром в полученной наилучшей пробе и уменьшенная по сравнению с предыдущей областью поиска в определенное количество раз, и производилась новая серия из 500 проб. Поиск был завершен после 4500 испытаний. Достигнутое значение функции (5.8) $G = 145,019 \text{ кгс}$; $R = 13,02 \text{ см}$; $\delta = 0,758 \text{ см}$. Как видим, применение «независимого» алгоритма глобального поиска с адаптацией позволило получить неплохой результат, но при этом затраты машинного времени значительны.

Нами для решения этой задачи был применен алгоритм ГП-3, который позволил добиться лучшего результата, чем в [119], при гораздо меньших затратах машинного

времени (оптимизация затрат машинного времени можно судить по количеству проведенных испытаний).

При оптимизации идеальной оболочки с точностью $\varepsilon=0,3\%$ глобальный поиск проведен за 131 испытание. Достигнутое значение веса $G=144,5401$ кг. Как видим, испытаний проведено в 34 раза меньше, а полученное значение минимизируемой функции лучше, чем в [119]. Результаты расчетов приведены в табл. 5.2.

Таблица 5.2

В лок. мин	G кг	R см	δ см	P_1^* кг	P_2^* кг	P_3^* кг	Кол-во раб. шагов	Кол-во испыт
1	153,686	5,83375	1,79178	$99,62 \cdot 10^5$	10^5	$1,064 \cdot 10^5$	7	71
2	144,540	41,7244	0,23422	$1,702 \cdot 10^5$	$48,69 \cdot 10^5$	$1,0 \cdot 10^5$	10	60

На рис. 5.2 показаны траектории движения поисковой системы в области поиска.

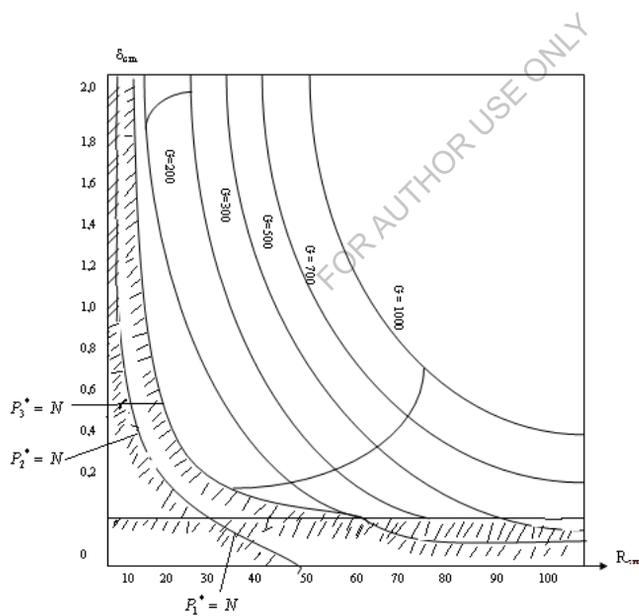


Рис.5.2

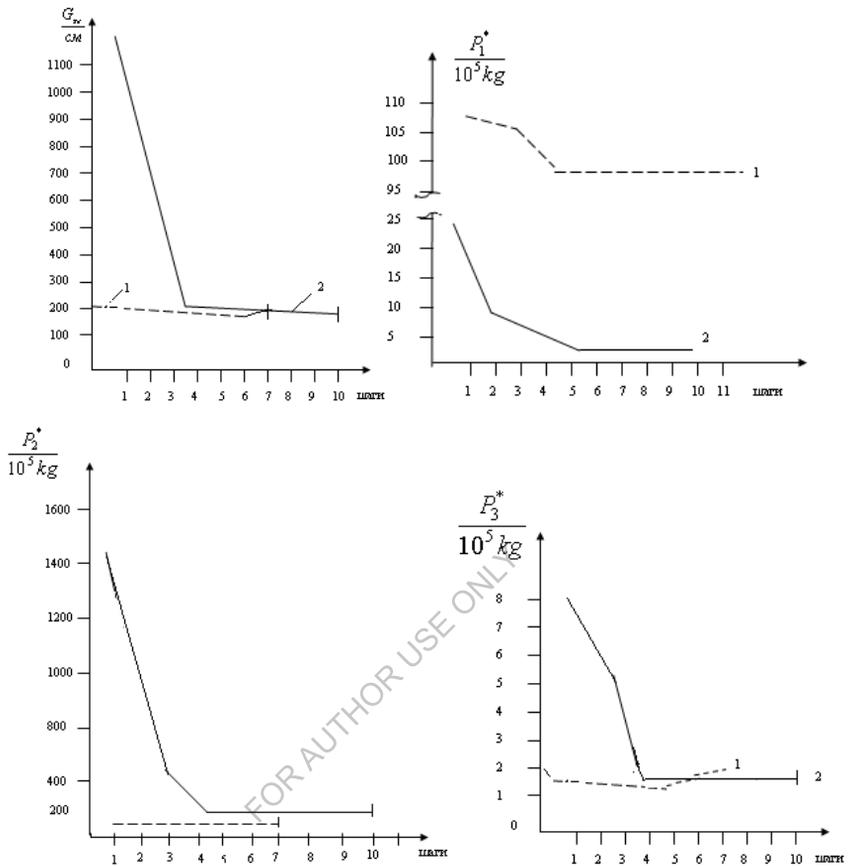


Рис.5.3

Изменение функции цели (5.8), а также величин критических нагрузок P_1^* , P_2^* , P_3^* при поиске обоих найденных минимумов показано на рис. 5.2, из рис.5.2 видно, что выход в окрестность минимума осуществлен за 4 рабочих шага во время поиска второго (глобального) минимума, и остальные 6 рабочих шагов затрачены на уточнение результатов. При поиске первого минимума после одного шага система находилась в непосредственной близости границы, а дальнейшие шесть шагов затрачены на уточнение результаты. Как видно из рис.5.2 ограничение $P_2^* = N$ в зоне первого минимума по существу параллельна линиям одинакового уровня функции (5.8) все же поисковая система в течение 6-ти рабочих шагов двигалась вдоль границы, пока происходило уменьшение функции цели.

При оптимизации оболочки с учетом начальных несовершенств глобальный минимум $G=144,5499$ кг был найден в точке $R=29,208$ см, $\delta=0,3366$ см после проведения 95 испытаний.

Сравнение алгоритма ГП-3 с «независимым» алгоритмом глобального поиска с адаптацией целиком в пользу алгоритма ГП-3, так как при гораздо меньших затратах получено лучшее значение минимизируемой функции цели.

5.3. Расчеты по оптимизации осесимметричных усеченных конических оболочек и кольцевых пластин переменной толщины [156,157]

Конструкции, включающие в себя конические оболочки, широко применяются в таких важных отраслях народного хозяйства, как ракетостроение, самолетостроение, турбостроение и т.д. В связи с этим актуальной задачей является проектирование конических оболочек минимального веса.

Задача оптимизация конических оболочек ставится следующим образом. Требуется перекрыть окружность радиусом R конической оболочкой, способной выдержать действующие внешние нагрузки $Z(\alpha)$ и $X(\alpha)$, которая при этом будет иметь минимальный вес. На высоту оболочки и величину радиуса отверстия при вершине (рис.5.4.) накладываются ограничения.

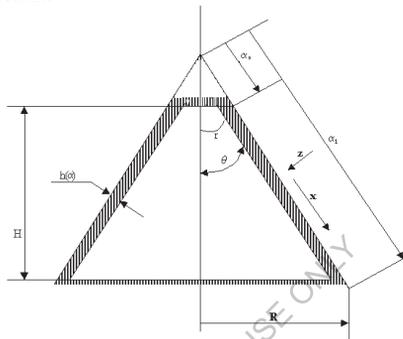


Рис. 5.4

При этом вес оболочки – минимизируемая функция

$$G = 2\pi \sin \Theta \gamma \int_{\alpha_0}^{\alpha_1} \bar{a} \cdot h(\alpha) d\alpha$$

где γ – удельный вес материала оболочки.

Оптимизируемые параметры такие:

1. угол конусности оболочки Θ ;
2. координата отверстия при вершине α_0 ;
3. параметры, определяющие толщину оболочки. Система ограничений следующая:

$$\Theta_{\min} \leq \Theta \leq \Theta_{\max};$$

$$\bar{\alpha}_{\min} \leq \bar{\alpha}_0 \leq \bar{\alpha}_{0\max};$$

$$h_{\min} \leq h \leq h_{\max};$$

$$H_{\min} \leq H \leq H_{\max};$$

$$r_{\min} \leq r \leq r_{\max};$$

$$\sigma_i \leq [\sigma];$$

$$\bar{\alpha}_0 = \frac{\alpha_0}{\alpha_1}; \alpha_1 = \frac{R}{\sin \Theta};$$

$$H = \frac{R}{\operatorname{tg}\Theta} - \alpha_0 \cos \Theta; r = \alpha_0 \cdot \cos \Theta,$$

где σ_i - интенсивность напряжений, определяемая по формуле

$$\sigma_i = \frac{\sqrt{2}}{2} \sqrt{(\sigma_\beta - \sigma_\alpha)^2 + \sigma_\alpha^2}$$

Напряжения σ_β и σ_α вычисляются после решения прямой задачи расчета осесимметричной конической оболочки.

При решении приведенных ниже задач количество узлов сетки N принималось равным 50. Задачи решались при следующих значениях физических и геометрических характеристик оболочки:

$$E = 2,1 \cdot 10^6 \text{ кг/см}^2; \nu = 0,3; [\sigma] = 2000 \text{ кг/см}^2;$$

$$\gamma = 0,0078 \text{ кг/см}^3; R = 100 \text{ см.}$$

Оболочка имеет жесткое защемление по обоим контурам. Оптимизация производилась с точностью $\varepsilon = 0,5\%$. Решены следующие задачи:

Задача 1. Произвести оптимизацию оболочки постоянной толщины под равномерно распределенной нормальной нагрузкой

$$Z = 10 \text{ кг/см}^2; X = 0.$$

Система ограничений такая:

$$\pi/4 \leq \Theta \leq 1,57;$$

$$0,1 \leq \bar{\alpha}_0 \leq 0,5;$$

$$0 \leq h \leq 5 \text{ см};$$

$$0 \leq H \leq 100 \text{ см};$$

$$10 \text{ см} \leq r \leq 50 \text{ см};$$

$$\sigma_i \leq [\sigma].$$

Результаты расчетов приведены в табл. 5.3.

Таблица 5.3

В мин.	G, кг	Θ , рад.	$\bar{\alpha}_0$	h, см	σ_i^{\max} кг/см ²
1	200,94	1,1545	0,50	1,00	1999,3
2	435,98	1,5700	0,50	2,3723	1988

Задача 2. Оболочка постоянной толщины имеет нагрузку $Z = 10 \sin \Theta$ кг/см²; $X = 10 \cos \Theta$ кг/см².

Остальные параметры те же, что и в задаче 1.

Результаты расчетов приведены в табл. 5.4.

Таблица 5.4

В мин.	G, кг	Θ	$\bar{\alpha}_0$	H, см	σ_i^{\max} кг/см ²
1	198,16	1,18756	0,500	1,00	1996,6
2	255,78	1,1938	0,1717	1,000	1992
3	442,37	1,5686	0,500	2,407	1930

Задача 3. Оболочка линейно-переменной толщины находится под нагрузкой

$$Z = 10 \text{ кг} / \text{см}^2; X = 0.$$

Толщина оболочки определяется по формуле

$$h = h_0 + h_1 \frac{\alpha - \alpha_0}{\alpha_1 - \alpha_0} \quad (5.13)$$

Ограничения следующие:

$$0,785 \leq \Theta \leq 1,57;$$

$$0,1 \leq \bar{\alpha}_0 \leq 0,5;$$

$$1 \leq h_0 \leq 3 \text{ см};$$

$$0 \leq H \leq 100;$$

$$\sigma_i \leq [\sigma]$$

Результаты расчетов приведены в табл. 5.5.

Таблица 5.5

В мин.	лок.	$G, \text{ кг}$	Θ	$\bar{\alpha}_0$	$h_0, \text{ см}$	$h_1, \text{ см}$	$\sigma_i^{\text{max}} \text{ кг} / \text{см}^2$
1		182,19	1,0335	0,500	1,00	-0,2671	1996
2		241,48	0,9196	0,1320	1,00	-0,3222	1971
3		374,23	1,2998	0,1186	1,7285	-0,3739	1896

Задача 4. Определить минимальный вес оболочки линейно-переменной толщины под нагрузкой

$$Z = 10 \cdot \sin \Theta \text{ кг} / \text{см}^2; X = 10 \cdot \cos \Theta \text{ кг} / \text{см}^2.$$

Параметры и ограничения, как в задаче 3.

Результаты даны в табл. 5.6.

Таблица 5.6

В мин.	лок.	$G, \text{ кг}$	Θ	$\bar{\alpha}_0$	$h_0, \text{ см}$	$h_1, \text{ см}$	$\sigma_i^{\text{max}} \text{ кг} / \text{см}^2$
1		164,34	0,9282	0,500	1,0438	-0,5897	1998
2		218,91	1,2056	0,500	1,0000	0,2026	1992
3		428,97	1,5700	0,500	2,4885	-0,2778	1991
4		499,03	1,5700	0,500	2,1692	0,4933	1998

Задача 5. Оптимизировать коническую оболочку, толщина которой меняется по закону

$$h = h_0 + h_1 \sin \frac{\alpha - \alpha_0}{\alpha_1 - \alpha_0} + h_2 \frac{\alpha - \alpha_0}{\alpha_1 - \alpha_0} \quad (5.14)$$

Компоненты нагрузки имеют значения:

$$Z = 10 \text{ кг} / \text{см}^2; X = 0$$

Ограничения такие:

$$\begin{aligned}
&0,785 \leq \Theta \leq 1,57; \\
&0,1 \leq \bar{\alpha}_0 \leq 0,5; \\
&1 \leq h_0 \leq 3; \\
&-1 \leq h_1 \leq 1; \\
&-1 \leq h_2 \leq 1; \\
&0 \leq H \leq 100; \\
&10 \leq r \leq 50; \\
&\sigma_i \leq [\sigma]
\end{aligned}$$

Результаты расчетов приведены в табл. 5.7.

Таблица 5.7

В лок. мин.	$G, \text{ кг}$	Θ	$\bar{\alpha}_0$	$h_0, \text{ см}$	$h_1, \text{ см}$	$h_2, \text{ см}$	$\sigma_i^{\max} \text{ кг/см}^2$
1	186,25	1,0751	0,5000	1,0000	-0,10577	-0,0743	1999,98
2	209,75	1,0980	0,5000	1,0824	0,3959	-0,5732	1957
3	279,64	1,1557	0,4631	1,8750	-1,0000	0,1624	1988
4	328,01	1,2311	0,5000	1,0000	0,5002	0,6558	1972
5	435,09	1,5666	0,5000	2,4945	-0,4954	-0,7966	1939
6	306,83	1,1945	0,2392	1,2163	0,5443	-0,5439	1978

Кривые $\sigma_1(\alpha)$, соответствующие 1,5,6 минимумам, приводятся на рис. 5.5.

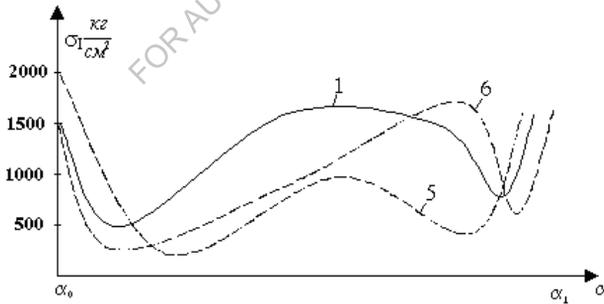


Рис. 5.5

Задача 6. Оптимизировать коническую оболочку, с толщиной переменной, определяемой по формуле (5.14), под нагрузкой

$$Z = 10 \cdot \sin \Theta \text{ кг/см}^2; X = 10 \cdot \cos \Theta \text{ кг/см}^2.$$

Результаты расчетов приведены в таблице 5.8.

Таблица 5.8

В лок. мин.	$G, \text{кг}$	Θ	$\bar{\alpha}_0$	$h_0, \text{см}$	$h_1, \text{см}$	$h_2, \text{см}$	$\sigma_i^{\max}, \text{кг}/\text{см}^2$
1	145,15	0,8693	0,5000	1,000	-0,4656	-0,1805	1999,9
2	379,36	1,3587	0,5000	2,347	-1,0000	0,7254	1982
3	183,26	0,9374	0,5000	1,000	-0,8034	0,5670	1982
4	143,13	0,8205	0,5000	1,000	-0,2729	-0,4618	1999,97
5	172,32	0,9331	0,1497	1,000	-0,5635	0,1015	1947
6	411,20	1,5700	0,5000	3,016	-0,7621	-0,5295	1962

Задача 7. Произвести оптимизацию кольцевой пластины постоянной толщины с защемленными краями, находящейся под действием равномерно распределенной внешней нагрузки $Z=10\text{кг}/\text{см}^2$.

Уравнения равновесия кольцевой пластины получаются из уравнений равновесия конической оболочки (5.2), если угол раствора $\Theta = \frac{\pi}{2}$. Все физические и геометрические характеристики пластин соответствуют характеристикам оболочки из предыдущих задач и при этом $r=50\text{см}$.

В результате оптимизации получены следующие результаты: $G=434,81\text{кг}$, $h=2,360\text{см}$, $\sigma_i^{\max}=1999,21$. Кривые $W(\alpha)$, $\sigma_2(\alpha)$, $\sigma_3(\alpha)$ и $\sigma(\alpha)$ приведены на рис. 5.6.

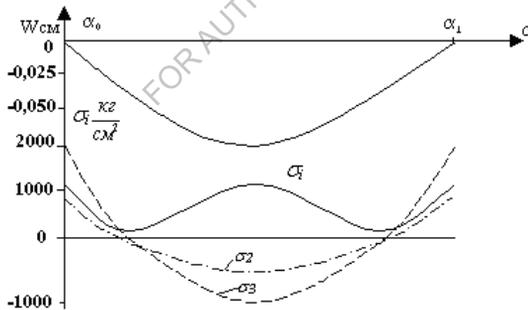


Рис.5.6.

Задача 8. Произвести оптимизацию пластины линейно-переменной толщины

$$h = h_0 + h_1 \frac{\alpha - \alpha_0}{\alpha_1 - \alpha_0}$$

Остальные данные те же, что и в задаче 7.

При решении получены следующие результаты: $G=417,26\text{кг}$, $h_0=2,734$, $h_1=-0,829$, $\sigma_i^{\max}=1999,99$.

Кривая $\sigma_7(\alpha)$ приведена на рис. 5.7.

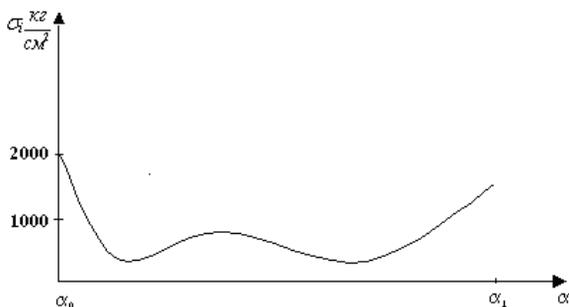


Рис.5.7

Задача 9. Оптимизация пластины с толщиной определяемой выражением:

$$h = h_0 + h_1 \sin \frac{\alpha - \alpha_0}{\alpha_1 - \alpha_0} + h_2 \frac{\alpha - \alpha_0}{\alpha_1 - \alpha_0}$$

Минимум веса $G=407,34\text{ кг}$ получен в точке:

$$h_0=2,999\text{ см}; h_1=-0,871; h_2=-0,411\text{ см}; \sigma_i^{\max}=1999,76\text{ кг/см}^2.$$

Кривые $W(\alpha)$, $\sigma_2(\alpha)$, $\sigma_3(\alpha)$ и $\sigma_1(\alpha)$ приведены на рис. 5.8.

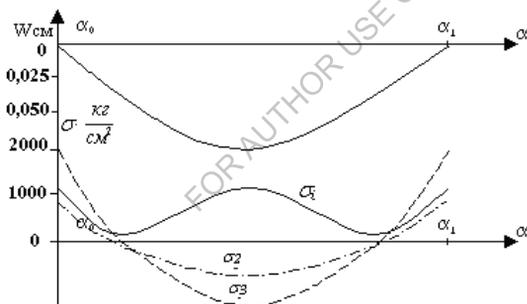


Рис.5.8

Как видим по результатам решения задач 7-9, применение переменной толщины позволило снизить вес пластины по сравнению с постоянной толщиной:

- при линейном изменении толщины на 4%;
- при нелинейном – на 6%.

5.4. Весовая оптимизация упругих пластин сложной конфигурации

Задача оптимизации пластин ставится как задача математического программирования: необходимо определить вектор оптимизируемых параметров $X = (x_1, x_2, \dots, x_n)$, сообщающих минимум целевой функции $F(X)$ при удовлетворении функциональных ограничений $f_j(X) \leq 0$ и ограничений на параметры:

$$a_i \leq x_i \leq b_i \quad (i = \overline{1, n}).$$

Эта задача описывается так:

$$\left. \begin{aligned} F(X) \rightarrow \min, \\ f_j(X) \leq 0, \quad j = \overline{0, m}, \\ a_i \leq x_i \leq b_i, \quad i = \overline{1, n}. \end{aligned} \right\} \quad (5.15)$$

Рассматриваем весовую оптимизацию пластин, т.е. $F(X)$ - вес пластины из изотропного материала, находящейся под действием внешней нагрузки q . Функциональные ограничения, учитывающиеся при оптимизации инженерных конструкций, а также численный алгоритм оптимизации пластинчатых конструкций сложной конфигурации подробно описан в [97-100]. В качестве оптимизируемого параметра принимается толщина, постоянная в пределах пластины.

Задача 1. Оптимизация по весу круглой жестко защемленной по всему контуру пластины, находящейся под равномерным внешним давлением $q = 10 \text{ кг}$. Радиус пластины $R = 100 \text{ см}$, модуль упругости $E = 2^{10} \cdot 6 \text{ кг/см}^2$, коэффициент Пуассона $\nu = 0.3$, допускаемые прогиб $[W] = 1 \text{ см}$ и эквивалентное $[\sigma_{\text{экр}}] = 2550 \text{ кг/см}^2$, удельный вес $\gamma = 7.8 \text{ г/см}^3$, $G(h) = \pi R^2 \gamma h \rightarrow \min$, $W_{\text{max}} \leq [W]$, $\sigma_{\text{max}}^{\text{экр}} \leq [\sigma_{\text{экр}}]$, $1 \text{ см} \leq h \leq 10 \text{ см}$.

Оптимизация проводилась с точностью до $\varepsilon = 0.01$. Получены следующие результаты:

$$G_{\text{min}} = 1141.19 \text{ кг}, \quad h = 4.6571 \text{ см},$$

$$W_{\text{max}} = 0.8206 \text{ см}, \quad \sigma_{\text{max}}^{\text{экр}} = 2549.99 \text{ кг/см}^2.$$

Эта задача имеет точное решение:

$$W_{\text{max}} = \frac{qR^4 12(1-\nu^2)}{64Eh^3}.$$

При полученном значении h имеем

$$W_{\text{max}} = 0.8446 \text{ см}.$$

Точность полученного приближенного решения удовлетворительна.

Задача 2. Оптимизация по всему кольцевой жестко защемленной по обоим контурам пластины, находящейся под равномерным внешним давлением интенсивностью $q = 10 \text{ кг/см}^2$.

Внешний радиус пластины $R = 100 \text{ см}$, внутренний - $r = 50 \text{ см}$. Остальные параметры те же, что и в задаче 1.

$$G(h) = \pi \gamma h (R^2 - r^2) \rightarrow \min$$

$$\text{Получены результаты: } G_{\text{min}} = 382.95 \text{ кг}, \quad h = 2.0837 \text{ см}, \quad \sigma_{\text{max}}^{\text{экр}} = 2549.82 \text{ кг/см}^2.$$

Задача 3. Оптимизация по всему полукольца (рис. 5.9). Все параметры те же, что и в задаче 2.

$$G_{\text{min}} = 134.48 \text{ кг}, \quad h = 1.126 \text{ см}, \quad \sigma_{\text{max}}^{\text{экр}} = 2549.99 \text{ кг/см}^2.$$

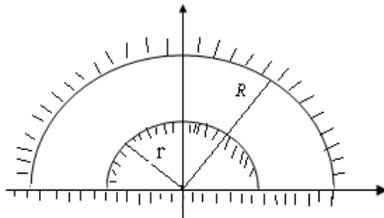


Рис. 5.9

Задача 4. Оптимизация по весу квадрата с круглым вырезом (рис. 5.10):
 $G(h) = (ab - \pi r^2) \gamma h$, $a = 200 \text{ см}$, $b = 200 \text{ см}$, $r = 50 \text{ см}$, $q = 10 \text{ кг/см}^2$,
 $W_{\max} = [W]$, $\sigma_{\max}^{\text{дог}} \leq [\sigma_{\text{дог}}]$, $1 \text{ см} \leq h \leq 10 \text{ см}$, $G_{\min} = 296.07 \text{ кг}$, $h = 3.9767 \text{ см}$,
 $\sigma_{\max}^{\text{дог}} = 2546.07 \text{ кг/см}^2$.

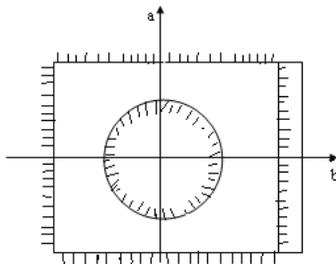


Рис. 5.10

Задача 5. Оптимизация по весу фигуры, показанной на рис.5.11:

$G(h) = (ab - 3\pi r^2) \gamma h$, $a = 200 \text{ см}$, $b = 200 \text{ см}$, $r = 20 \text{ см}$, $W_{\max} \leq [W]$, $\sigma_{\max}^{\text{дог}} \leq [\sigma_{\text{дог}}]$,
 $1 \text{ см} \leq h \leq 10 \text{ см}$, $G_{\min} = 725.96 \text{ кг}$, $h = 2.5689 \text{ см}$, $W_{\max} = 0.1262 \text{ см}$,
 $\sigma_{\max}^{\text{дог}} = 2549.9998 \text{ кг/см}^2$.

Задача 6. Оптимизация кольцевой пластины линейно переменной толщины.

Для упрощения выкладок принята полярная система координат (α - линейная, β - угловая).

Примем толщину в виде

$$h = h_0 + h_1 \frac{\alpha - r}{R - r}.$$

Остальные параметры те же, что и в задаче 2.

Целевая функция определяется из выражения

$$G(h) = e\pi \int_r^R \alpha h(\alpha) d\alpha.$$

Ограничения следующие:

$$\sigma_i^{\max} \leq [\sigma] = 2000 \text{ кг/см}^2,$$

$$W_{\max} \leq [W]$$

$$1 \text{ см} \leq h_0 \leq 3 \text{ см},$$

$$-1 \text{ см} \leq h_1 \leq 1 \text{ см},$$

где σ_i - интенсивность напряжений.

Получены результаты:

$$G_{\min} = 417.26 \text{ кг}, h_0 = 2.734 \text{ см}, h_1 = -0.826 \text{ см}, \sigma_i^{\max} = 1999.99 \text{ кг/см}^2.$$

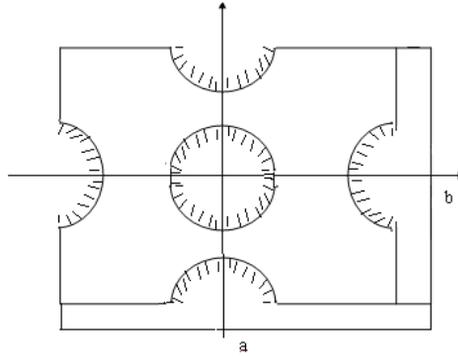


Рис. 5.11

Задача 7. Оптимизация кольцевой пластины нелинейно переменной толщины, т.е.

$$h(\alpha) = h_0 + h_1 \sin \frac{\alpha - r}{R - r} + h_2 \frac{\alpha - r}{R - r}$$

при следующих ограничениях:

$$\sigma_i^{\max} \leq [\sigma], W_{\max} \leq [W],$$

$$2 \text{ см} \leq h_0 \leq 3 \text{ см}, -1 \text{ см} \leq h_1 \leq 1 \text{ см}, -1 \text{ см} \leq h_2 \leq 1 \text{ см}.$$

Получены результаты:

$$G_{\min} = 407.34 \text{ кг}, h_0 = 2.999 \text{ см}, h_1 = -0.871 \text{ см},$$

$$h_2 = -0.411 \text{ см}, \sigma_i^{\max} = 1999.76 \text{ кг/см}^2$$

Видим, что применение нелинейно — переменной толщины привело к снижению веса пластины.

Задачи 1 и 2 имеют точные решения и приведены лишь для возможности контроля за результатами работы алгоритма [48]. Как видно по результатам решенных задач, основным ограничением (при взятых нами значениях $[\sigma_{\text{доп}}]$, $[\sigma]$ и $[W]$) являются ограничения по прочности, а алгоритм [48] обеспечил высокую степень приближения к границе. Разнообразие форм пластин указывает на широкие возможности применения алгоритма [49] к решению задач оптимизации пластин сложной конфигурации.

ЗАКЛЮЧЕНИЕ

Для дальнейшего повышения эффективности производства, экономии материалов, улучшения качества проектируемых объектов, сокращения сроков проектирования необходимо разработка новых систем автоматизированного проектирования на основе алгоритмических методов, который способных решать все перечисляемые задачи. Решение вопроса сводится к созданию систем оптимальных алгоритмов, функционирование которых в основном базируется на использовании методов оптимизации, системного анализа и теории распознавании образов.

В результате проведенных многолетних комплексных исследований различных известных и предложенных в данной работе оптимизационных задач и их алгоритмов, сравнительных и ресурсных испытаний с целью получения оптимальных решений были предложены обоснованные рекомендации по применению оптимальных систем алгоритмов в проектной практике.

Теоретические основы оптимизации инженерных конструкций базировались на создании их математических моделей оптимизации по весу. Построение таких моделей потребовала проведения как теоретических и так и машинных экспериментов. Большого фактического материала по моделям и их алгоритмов широко применяемые в проектном деле позволило впервые в расчетной практике разработать наиболее надежные (адекватные) системы оптимальных алгоритмов.

Проведенные исследования по построению математических моделей оптимизации и их алгоритмизация тонкостенных конструкций (пластинки, оболочки, арки) позволили распространить разработанные методы оптимизации алгоритмических систем на конструкции подземных сооружений с учетом динамических воздействий. Сформулированы основные требования к алгоритмам, включающимся в АСО и другие ППП, предназначенные для решения оптимизационных задач. Разработаны эффективные алгоритмы случайного поиска, отвечающие сформулированным требованиям и включенные в АСО. Они отличаются друг от друга степенью простоты и универсальности, способны решать широкий круг задач как безусловной, так и условной оптимизации, не предъявляя каких-либо требований моделям с непрерывно изменяющимися параметрами. Разработан и предложен алгоритм локального поиска ГП-3 глобального поиска, который решает многоэкстремальные задачи. Преимуществом этого алгоритма является экономию машинного времени. На базе ГП- 3 создана его модификация способная решать задачи при наличии целочисленных переменных.

Предложена алгоритмическая система для решения классов оптимизационных задач (АСО), основанная на работе алгоритмических банков (Б), каждый из которых имеет строго определенные функции. АСО работает в автоматическом режиме и предусматривает: унификацию постановок оптимизационных задач; контроль за правильностью постановки конкретной задачи; распределение памяти компьютера; исследование и идентификацию моделей; выбор алгоритмов из условий допустимости к решению поставленной задачи и оптимальности их; проведение параметрической оптимизации алгоритмов с целью получения скорости сходимости для класса задач; построение оптимальной последовательности алгоритмов; формирование рабочей программы, проведение счета.

Результаты вычислительного эксперимента по оптимизации тонкостенных инженерных конструкций показывают, что все найденные минимумы рассмотренные конструкции находились на грани прочности и устойчивости, тогда как значения их весов значительно отличается. Иначе говоря, обе конструкций, обладая одиноковым запасом прочности, отличаются по весу почти 20%. Результаты оптимизации арок и незамкнутые цилиндрические оболочки показывают, что при рассмотренных системах нагрузок рациональны более крутые типы конструкции. Применение для оболочек переменной толщины позволило в ряде случае снизить вес конструкций ~14% (для цилиндрических

оболочек) и на ~27% (для конических оболочек) по сравнению с оболочками постоянными толщинами, что говорит о целесообразности применения конструкций переменной толщины. Применение АСО для решения приведенных задач, несмотря на всю сложность и разнообразие, свидетельствует об эффективности системы и заложенных в нее алгоритмов.

Достоверность результатов проведенных расчетных исследований в широких пределах изменения основных определяющих параметров подтверждается хорошим (удовлетворяющим) совпадением их с теоретическими и вычислительными результатами других исследований.

По предложенным рекомендациям к проектированию инженерных конструкций по весовой оптимизации, был создан АСО защищенной авторским свидетельством и внедрены в расчетной практике. Ряд результатов работы могут быть успешно применяться в проектировании и строительстве инженерных конструкций, различного типа.

Результаты исследований, приведенные в работе, могут широко применяться в НИИ, КБ и проектных организациях, связанных с решением различных оптимизационных задач, а также при строительстве надземных и подземных инженерных конструкций и сооружений.

FOR AUTHOR USE ONLY

Использованная литература

1. Абовский Н.П., Андреев Н.П., Деруга А.П. Вариационные принципы теории упругости и теории оболочек. – М.: Наука, 1978. – 288 с.
2. Адаменко Г.М. Алгебраическая модель алгоритмов оптимизации. «Доклад АН СССР», 1978,22, № 9, 801-803.
3. Адаменко Г.М. Решение экстремальных задач в условиях неполной информированности. «Автоматика и вычислительная техника», 1980. № 4. - С. 53-59.
4. Амбарцумян С.А. Теория анизотропных пластин: Прочность, устойчивость и колебания. – 2-е изд. перераб. и доп.- М.: Наука, 1987. – 360 с.
5. Амирханов И.Г. К теории оболочек вращения наименьшего веса. Труды семинара по теории оболочек, Казанский физ.-тех. институт АН СССР. Вып.1, 1968.
6. Байсборд Э.М., Расстригин Л.А., Рубинштейн Я.С. Сходимость алгоритма случайного поиска по наилучшей пробе. Автоматика и вычислительная техника. I, 1973
7. Баничук Н.В. Оптимизация форм упругих тел. – М.: Наука, 1980.
8. Баничук Н.В. Введение в оптимизацию конструкций. – М.: Наука, 1986
9. Бараненко В.А., Почтман Ю.М., Филатов Г.В. О совместном использовании методов динамического программирования и случайного поиска в задачах оптимального проектирования. Строительная механика и расчет сооружений. I, 1973
10. Батищев Д.И. Пакеты программ диалоговой оптимизации (обзор). Материалы Всесоюзного научно-технического совещания «Пакеты прикладных программ САПР». Таллин, 1978. - С.82-85.
11. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М., Наука, 1965
12. Бинкевич Е.В., Вергейчик Л.В., Моссаковский В.И. О рациональном распределении материала силовых шпангоутов цилиндрических оболочек. Труды VII Всесоюзной конференции по теории пластин и оболочек. М., Наука, 1970
13. Борскаускас А.Э., Чирас А.А. Расчет упругопластических пластинок минимального веса с применением линейного программирования. Литовский механический сборник. 1968. № 2.
14. Болотин В.В., Гольденблат И.И., Смирнов А.Ф. Строительная механика: Современная состояние и перспективы развития. – 2-е изд. Перераб. и доп. – М.: Изд-во лит-ры по строительству, 1972. – 192 с
15. Буриев Т. Алгоритмизация расчета несущих элементов тонкостенных конструкций. – Ташкент: Фан, 1986. – 244 с.
16. Буриев Т. Расчет тонких плит на ЭВМ. – Ташкент: Фан, 1986. – 132 с.
17. Вабищевич П.Н. Численное моделирование. – М.: МГУ, 1993. – 152 с.
18. Валишвили Н.В. Методы расчета оболочек вращения на ЭЦВМ. – М.: Машиностроение, 1976. – 278 с.
19. Васильев В.В. Оптимальное проектирование пластин и оболочек. Труды VII Всесоюзной конференции по теории оболочек и пластинок. - М.: Наука, 1970. - С.721-735.
20. Васидзу К. Вариационные методы в теории упругости и пластичности. – М.: Мир, 1987. – 542 с.
21. Васильев В.Г. О наимыгоднейшем изменении поперечных сечений равнопрочных арок. Доклад высшей школы «Строительство». № 1, 1959.
22. Виноградов А.И. Вопросы расчета сооружений наименьшего веса. В кн. Труды Харьковского института инженеров железнодорожного транспорта. 25, М., Трансжелдориздат, 1955.
23. Виноградов А.И. О сходимости прочностного перерасчета в задачах оптимизации. Строительная механика и расчет сооружений. № 3, 1971.
24. Виноградов А.И. Об одном алгоритме теории оптимальных систем. Прикладная механика. Т. 3. № 7, 1967.

25. Виноградов А.И. Об оптимальном распределении усилий в стержневых системах и свойствах оптимальных систем. Прикладная механика. Т. 1. № 1, 1965.
26. Виноградов А.И. Оптимальные системы и применение ЭЦВМ при расчете сооружений. Труды Харьковского института инженеров железнодорожного транспорта. Вып. 91, 1967.
27. Виноградов А.И., Дорошенко О.И., Храповицкий И.С. Некоторые направления в теории оптимальных стержневых систем. Строительная механика и расчет сооружений. № 4, 1968.
28. Власов В.З. Общая теория оболочек. Т. 1, М., 1962.
29. Власов В.З., Леонтьев Н.Н. Балка, плиты и оболочки на упругом основании. М., 1960.
30. Волков Е.А. Численные методы. – М.: Наука, 1987. – 248 с.
31. Гайдуков А.Л. Применение случайного поиска при оптимальном проектировании. В кн. Прикладные задачи технической кибернетики. М., Советское радио, 1966.
32. Гилл Ф., Мюррей У. Практическая оптимизация. – М.: Мир, 1985. – 509 с.
33. Герасимов Е.Н. Задачи оптимизации в механике твердого деформируемого тела и теории проектирования инженерных конструкций. - Ижевск: ИМИ, 1979. - 80 с
34. Гинзбург И.Н., Канн С.Н. Выбор оптимальных параметров экстремально подкрепленной цилиндрической оболочки при осевом сжатии. Сб. Теория пластин и оболочек. М., Наука, 1971.
35. Гинзбург И.Н., Канн С.Н. Об одном методе выбора оптимальных параметров тонкостенной конструкции. Труды УП Всесоюзной конференции по теории пластин и оболочек. М., Наука, 1970.
36. Гребенюк Г.И. Об эффективной организации итерационных процессов конечномерной оптимизации конструкций. // Известие вузов. Строительство и архитектура, 1988.
37. Григолок Э.И., Кабанов В.В. Механика твердых деформируемых тел. М., 1969.
38. Григоренко В.П., Тоомсалу Т.А., Юлегиц Ю.Н. Организация выбора алгоритмов в пакете программ «Оптимизация». Материалы Всесоюзного научно-технического совещания. Пакеты прикладных программ САПР. Таллин, 1978. - С. 96-102.
39. Грудачев В.Г., Костромин Г.Я., Меркуров В.В., Половинкин А.И., Трахтенберг В.С. Некоторые результаты по исследованию локальных и глобальных алгоритмов случайного поиска. Материалы IX Всесоюзного семинара по случайному поиску экстремума и его приложениям. Харьков, 1973.
40. Давыдов С.С. Расчет и проектирование подземных конструкций. М., Стройиздат, 1950.
41. Дейнека В.С., Молчанов И.Н., Николенко Л.Д. Численное решение задачи о напряженно-деформированном состоянии толстой цилиндрической оболочки методом конечных элементов // Численные методы решения задач теории упругости и пластичности: Матриалы V Всесоюз. Конф. - Новосибирск, 1978. – С. 48-54
42. Дейнека В.С. Расчет методом конечных элементов некоторой ортотропной цилиндрической оболочки, регулярно подкрепленной кольцевыми ребрами жесткости. – Киев, 1977. – 18 с. (препринт – 77 21 / Ин-т кибернетики АН УССР).
43. Дмитриев Л.Т., Сосис П.М. Программирование расчета пространственных конструкций. – Киев: Гос. изд-во по строительству и архитектуре УССР, 1963. - 288 с.
44. Захаров Б.В. Об одной группе алгоритмов случайного поиска для решения многоэкстремальных задач. В сб. Труды III Всесоюзного симпозиума по экстремальным задачам. Поиск экстремума, Томск, 1969.
45. Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование. М., Наука, 1967.
46. Кабулов В.К. Алгоритмизация в механике сплошных сред. - Т.: Фан, 1980. - 304 с.
47. Кабулов В.К. Алгоритмизация в теории упругости и деформационной теории пластичности. -Т.: Фан, 1966. – 394 с.
48. Кабулов В.К., Назиров Ш.А., Якубов С.Х. Алгоритмизация решения оптимизационных задач. – Т.: Изд-во «Фан» АН РУз, 2008. – 204 с.

49. Кабулов В.К., Пискорский Л.Ф. О разработке алгоритмической системы для решения оптимизационных задач. Тезисы докладов Всесоюзной конференции «Современные методы и алгоритмы расчета и проектирования строительных конструкций с использованием ЭВМ». Таллин, 1979.
50. Кабулов В.К., Файзуллаев О, Назиров Ш.А. Ал-Хоразмий, алгоритм, алгоритмизация. - Ташкент: Фан, 2006. - 672 с.
51. Кавелергин Б., Кожевников А.А., Кузнецов Б.Б. Оптимальное проектирование подкрепленных сферических оболочек. –Прикладная механика, сб. IX, 1973, вып. 10.
52. Калинин И.Н. К проектированию оболочек вращения минимального веса. Ученые записки ГГУ. Вып. 138, Горький, 1971.
53. Канн С.Н. Строительная механика оболочек. М., Машиностроение, 1966.
54. Кармишин А.В., Лясковец В.А., Мяченков В.И., Фролов А.Н. Статика и динамика тонкостенных оболочечных конструкций. – М.: Машиностроение, 1975. - 376 с.
55. Карумидзе Г.В. Метод случайного поиска решения многоэкстремальных задач. Известия АН СССР. Техническая кибернетика, 6, 1969.
56. Кипа К.К. Непрерывное самообучение случайного поиска /направляющая сфера/. В кн. Алгоритмы и программы случайного поиска. Рига, «Зинатне», 1969.
57. Кипа К.К. Покоординатное обучение при случайном поиске. В кн. Алгоритмы и программы случайного поиска. Рига, «Зинатне», 1979.
58. Киселев В.А. Рациональные формы арок и подвесных систем. - М., Гос. изд. по строительству и архитектуре, 1953.
59. Киселев В.А. Строительная механика. – М., 1967.
60. Клюев П.А., Колдунов В.А. Численная модель расчета напряженно-деформированного состояния упругих оболочек из разнотипного материала // Сложные системы: моделирование и оптимизация. – Тверь: Тверь, гос. ун-т, 2001. – С. 141 – 151.
61. Ковалев В.А. Некоторые прикладные задачи механики сплошных сред. // Прикладные проблемы прочности и пластичности. – Межвуз. Сб. / М.: Товарищ. Науч. Изд. КМК. 1995. Вып. 53. – С.72 – 78.
62. Колдунов В.А., Кудинов А.Н., Черепанов О.И. Численные модели расчета оболочечных конструкций с трехмерных позиций. – Тверь: Тверь, гос. ун-т, 2006. – 88 с.
63. Колдунов В.А., Якубов С.Х. Численный расчет совместно упругого напряженно-деформированного состояния среды и ортотропной цилиндрической оболочки. –В. кн.: Инж.-физ. сб. –Томск: Изд-во Томского ун-та, 1988. –С. 34-36.
64. Колдунов В.А., Кудинов А.Н., Люкшин П.А., Мударисов Ш.Ш., Черепанов О.И. Анализ напряженно-деформированного состояния оболочечных конструкций с учетом анизотропии на основании пространственной численной схемы расчета // Теория пластин и оболочек: XI Всесоюз. Конф. – Таллин, 1983. – Ч.3. – С. 55-60.
65. Колдунов В.А., Лейцин В.Н., Пономарев С.В. Некоторые численные методы механики деформируемого твердого тела. – Томск: Изд-во Томск. ун-та, 1987. – 148 с.
66. Колдунов В.А., Люкшин П.А., Мударисов Ш.Ш., Черепанов О.И. Расчет НДС ортотропной цилиндрической оболочки в зоне краевого эффекта // Механика деформируемого твердого тела. – Томск: Томск. ун-т, 1987. – С. 86 – 90.
67. Колдунов В.А., Мударисов Ш.Ш., Черепанов О.И. Расчет круговой, подкрепленной ребрами цилиндрической оболочки на основании общих соотношений теории упругости // Механика сплошных сред. - Томск: Томск. ун-т, 1983. – С. 59 – 67.
68. Коротаева Л.Н., Пинишев А.В. Программа нахождения глобального экстремума функции многих переменных. В кн. Алгоритмы и программы случайного поиска. Рига; «Зинатне», 1969.
69. Кудинов А.Н. Актуальные проблемы математического моделирования процессов деформирования и устойчивости неоднородных систем // IV международ. науч. симпозиум «Устойчивость и пластичность в МДТТ»: Тезис. докл. – Тверь, 1998. – С.12-14.

70. Кудинов А.Н., Колдунов В.А. Актуальные вопросы математического моделирования деформируемых систем // Науч. конф. Профессорско – преподавательского состава и сотрудников госбюджетных и хоздоговорных тем: Тезис. докл. - Тверь: Твер. гос. ун-т, 1993. – С. 47-48.
71. Кудинов А.Н., Якубов С.Х. Исследование устойчивости цилиндрической оболочки, погруженной в грунт. – //Прикладные вопросы деформируемых тел. –Томск: Изд-во Томского ун-та, 1984. – С.17- 21.
72. Кудинов А.Н., Якубов С.Х. Исследование устойчивости круговой цилиндрической оболочки, погруженной в грунт. – В кн.: Инженерно-физический сб. –Томск: Изд-во Томского ун-та, 1987. –С.47-51.
73. Кудинов А.Н., Якубов С.Х. Устойчивость цилиндрической ортотропной оболочки, погруженной в сыпучую среду, при неравномерном внешнем давлении. –В кн.: Исслед. по строит. мех. и строит. конструкциям. –Томск, 1989. – С. 62-67.
74. Кузьминков Д.М., Чернецкий В.И. Метод поиска глобальных экстремумов в задачах линейного и нелинейного программирования при произвольных ограничениях. В кн. Алгоритмы и программы случайного поиска. Рига, «Зинатне», 1979.
75. Курицкий В.Я., Аксеенко Г.П., Виткин Ю.В. и др. Применение пакетов прикладных программ по экономико-математическим методам в АСУ. М. Статистика, 1980. –С.196.
76. Лоскутов В.Г., Носов А.М., Малоков В.Н., Конончук В.С. Принцип построения входного языка пакета программ стохастической и недеференцируемой оптимизации. В сб. «Вычислительные аспекты в пакетах программ и опыт решения оптимизационных задач». Киев, 1981. - С. 22-26.
77. Мажид К.И. Оптимальное проектирование конструкций. – М.: Высш. Шк., 1979. – 237 с.
78. Малков В.П., Стронгин Р.Г. Оптимизация конструкции по весу из условия прочности. Ученые записки ГГУ, серия – механика. Вып. 134, Горький, 1971.
79. Малков В.П., Тарасов В.Л. Дискретно равнонапряженная тонкостенная конструкция и конструкция минимального веса // Изв. АН СССР. МТТ. 1974. № 5.
80. Малков В.П., Туринцева Г.Д. Оптимизация сосуда под давлением из условия прочности. Ученые записки ГГУ, серия – механика. Вып. 108, 2, Горький, 1970.
81. Малков В.П., Угодчиков А.Г. Оптимальное проектирование составных тонкостенных конструкций. В сб. «Автоматизированное оптимальное проектирование инженерных объектов и технологических процессов». Вып. 2, 1974, ГГУ, Горький
82. Малков В.П., Угодчиков А.Г. Оптимизация упругих систем. – М.: Наука, 1981. – 288 с.
83. Малков В.П. Параметрическая оптимизация механических систем // Прикладные проблемы прочности и пластичности: Межвуз. сб. / М.: Товарищ. науч. изд. КМК. 1995. Вып. 53. С. 103-111.
84. Малков В.П. Повышение эффективности инженерной оптимизации // Межвуз. сб. / М.: Товарищ. науч. изд КМК. 1996. Вып. 54. С. 131-140.
85. Малков В.П., Маркина М.В. Поэтапная параметрическая оптимизация: / Н. Новгород: Изд-во ННГУ, 1998. - 142 с.
86. Малков В.П. Энергетический критерий материалоемкости // Конструирование алгоритмов и решение задач математической физики / АН СССР. Ин-т прикл. Математики им. Кельдиша. М., 1987.
87. Малков В.П. Энергоемкость механических систем. Н,Новгород. Изд-во Нижегород. Ун-та.1995. С.252.
88. Маневич А.И. Устойчивость и оптимальное проектирование подкрепленных оболочек. – Киев: Вища школа, 1979. – 152 с.
89. Медведев Г.А. Случайный поиск в многоэкстремальных задачах. Тезисы докладов IV симпозиума по экстремальным задачам. Каунас, 1969.

90. Мелешко В.И., Растринин Л.А. Состояние и проблемы развития случайного поиска. Материалы IX Всесоюзного семинара «Случайный поиск экстремума», Киев, «Наукова думка», 1974, С.5-16.
91. Михайловский Е.И., Торопов А.В. Математические модели теории упругости.- Сыктывкар: Изд-во СГУ, 1995.- 251с.
92. Михалевич В.С., Шор Н.З., Галустова Л.А. и др. Вычислительные методы выбора оптимальных проектных решений. Киев, «Наукова думка», 1977. С.176.
93. Михлин С.Г. Вариационные методы в математической физике. – М.: Наука, 1970. - 512 с.
94. Моисеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации, М., «Наука», 1978.
95. Морозов В.Д., Малков В.П. Алгоритм поиска глобального экстремума при весовой оптимизации тонкостенных конструкций. В сб. «Автоматизированное оптимальное проектирование инженерных объектов и технологических процессов». Вып. 1, 1974, ГГУ, Горький.
96. Назиров Ш.А. Вопросы создания проблемно-ориентированного языка и алгоритмической системы для проведения научных исследований и инженерных расчетов. – В кн. «Современные проблемы алгоритмизации и программирования». Докл. и тезисы Республ.научн. конф. Ташкент, 5-7 сентября 2001г. – Т., 2001, с. 39 – 40.
97. Назиров Ш.А., Пискорский Л.Ф «Вычислительный алгоритм оптимизации пластинчатых конструкций со сложной конфигурацией» Вопросы вычисл. и прикл. матем. Т, 1995, Вып.-100. – С. 49-56.
98. Назиров Ш.А., Пискорский Л.Ф «Комплекс программ для расчета и оптимизации инженерных конструкции сложной конфигурации.» Тез.докл.конф."Математические моделирования и вычислительный эксперимент." Ташкент, 1994. –С. 215.
99. Назиров Ш.А., Пискорский Л.Ф «Комплекс программ для расчета и оптимизации пластинчатых конструкции со сложной формой» Алгоритмы. Ташкент, 1995, Вып.80. –С. 41-54.
100. Назиров Ш.А., Пискорский Л.Ф, Бобокулов Ш.О, Нго Нгок Хынг. «Оптимизация упругих пластин сложной конфигурации.» Вопр. выч. и прикл. математики.Ташкент,1995, Вып.99. –С. 41-45.
101. Назиров Ш.А., Пискорский Л.Ф., Бобокулов Ш.О. «Алгоритмизация оптимизационных задач в теории упругости.» Узб.журн., "Проблемы информатики и энергетики"-Ташкент, Фан АН РУз, N3, 1997.
102. Назиров Ш.А., Якубов С.Х. Алгоритмическая система, автоматизирующая процессов оптимизации для проектирования инженерных конструкций и сооружений. - // Свидетельство Патентного ведомства Республики Узбекистан, DGU 01422, 2007.13.11.
103. Назиров Ш.А., Якубов С.Х. К вопросу автоматизации проектирования цилиндрических оболочек, подкрепленных ребрами жесткости//Вестник Каракалпакского отд. АН РУз, Нукус, 2012, №1. - С. 13- 17.
104. Немировский Ю.В. Оболочки абсолютно минимального веса. «Механика деформируемых сред», Куйбышев, 1978, № 3. С. 3-78.
105. Немировский Ю.В., Старостин Т.И. О возможности реализации безмоментного состояния оболочек путем армирования. Докл. АН СССР, 1971, 196, № 4. С. 797-800.
106. Образцов И.Ф., Васильев В.В., Бунаков В.А. Оптимальное армирование оболочек вращения из композиционных материалов. – М.: Машиностроение, 1977. – 144 с.
107. Образцов И.Г., Савельев Л.М., Хасанов Х.С. Метод конечных элементов в задачах строительной механики летательных аппаратов. – М.: Вышш. шк., 1985. – 392 с.
108. Образцов И.Ф. Проблемы проектирования тонкостенных конструкций из композиционных материалов. // Расчеты на прочность: Сб. науч. ст. – М.: Машиностроение, 1989. – Вып.30. – С. 3 – 6.

109. Орлов С.А. Методы статического расчета сборных железобетонных обделок тоннелей. М., Госстройиздат, 1961.
110. Перельмутер А.В., Сливкер В.И. Расчетные модели сооружений и возможность их анализа. – М.: Изд – во ДМК-пресс, 2007. – 600 с.
111. Пискорский Л.Ф. Алгоритм нахождения экстремумов многопараметрических многоэкстремальных функций. Автоматизированное оптимальное проектирование инженерных объектов и технологических процессов. Вып. 1, ГГУ, Горький, 1974.
112. Пискорский Л.Ф. Алгоритмы ГП2 и ГП3 поиска глобального экстремума функции многих переменных. Вопросы вычислительной и прикладной математики. Вып. 20, 1973.
113. Пискорский Л.Ф. К выбору оптимальных алгоритмов для решения классов оптимизационных задач. Известия АН УзССР, серия техн. наук. Вып.5, 1980.
114. Пискорский Л.Ф. Об автоматизации настройки поисковых алгоритмов на классы оптимизационных задач. Тезисы докладов респ. научно-техн. конф. «Методические основы разработки, привязки и эксплуатации САПР в строительстве». Ташкент, 1982.
115. Пискорский Л.Ф. Применение методов случайного поиска. К решению обратной задачи расчета фундаментальных плит силосных корпусов. Известия АН УзССР, серия техн. наук. Вып.5, 1982.
116. Пискорский Л.Ф., Вопросы оптимального проектирования арок. Вопросы вычислительной и прикладной математики. Вып. 21, Ташкент, 1974.
117. Половинкин А.И., Бобков Н.К., Буш Г.Я. и др. Автоматизация поискового конструирования. М., Радио и связь. 1981. -344 с.
118. Попырин Л.С. Математическое моделирование и оптимизация теплоэнергетических установок. – М.: Энергия, 1978. – 410 с.
119. Почтман Ю.М., Филатов Г.В. Оптимальное проектирование ребристых цилиндрических оболочек при совместном осевом сжатии и внутреннем давлении методом случайного поиска на ЭЦВМ. Тезисы докладов VI всец. конф. по применению ЭЦВМ в строительной механике. Ленинград, 1971.
120. Почтман Ю.М., Филатов Г.В. Расчет оптимальных упругих конструкций, стесненных ограничениями, с помощью методов случайного поиска. Изв. ВУЗов /серия строительство и архитектура/ 1978.
121. Прочность. Устойчивость. Колебания. Справочник. Под ред. И.А.Бергера и Я.Г.Пановко. – М.: Машиностроение, 1968. - Т.1.-832 с.
122. Рабинович Н.М. К расчету стержневых систем минимального веса. В сб. «Исследование по теории сооружений». Вып. 1.... Стройиздат...
123. Растрингин Н.А. Алгоритм случайного поиска. В кн. Алгоритмы и программы случайного поиска. Рига, «Зинатне», 1979.
124. Растрингин Н.А. Случайный поиск с линейной тактикой. Рига, «Зинатне», 1973.
125. Растрингин Н.А. Статистические методы поиска. М., Наука, 1988.
126. Рвачев В.Л., Курпа Л.В. R – функция в задачах теории пластин. – Киев: Наукова думка, 1987. – 176 с.
127. Розин Л.А. Вариационные постановки задач для упругих систем. – Л.: ЛГУ, 1978. – 224 с.
128. Рубинштейн Я.С. О сходимости алгоритмов случайного поиска. Автоматика и вычислительная техника, 1969, I.
129. Руденко В.И., Мороз Л.Р. Проектирование оболочек большого диаметра. – Транс. стр – во, 1985, №5, с.27-28
130. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. – М.: Наука, 1978. -592 с.
131. Свитски Г., Кери Н. Цилиндрические конструкции минимального веса. Ракетная техника и космонавтика, 1963, № 10.
132. Сергеев Н.Д., Богатырев А.И. Проблемы оптимального проектирования конструкций. Изд-во литературы по строительству, Л., 1971.

133. Смирнова Л.Ф., Рябцев А.Т. Непрерывный алгоритм самообучения случайного поиска (гиперэллипсоид). В кн. Алгоритмы и программы случайного поиска. Рига, «Зинатне», 1969.
134. Сосис П.М. АЛГОЛ-60 и применение его в строительной механике. – Киев: Будівельник, 1965. – 324 с.
135. Стронгин Р.Г. Многоэкстремальная минимизация. «Автоматика и телемеханика». 1970, 7.
136. Тамазян Н.Н. Определение оптимальных диаметров напорных трубопроводов. - // Гидротехника и мелиорация, 1983, №10, с.60 -61
137. Гарнопольский Ю.Я. Алгоритм глобальной оптимизации многопараметрических функций методом случайного поиска. В кн. Алгоритмы и программы случайного поиска. Рига, «Зинатне», 1969.
138. Тонкостенные оболочечные конструкции: Теория, эксперимент и проектирование. Пер. с англ. К.Г. Бомштейна, А.М. Васильева; Ред. Э.И. Григолюк – М.: Машиностроение, 1980. – 607 с.
139. Уайлд Д. Дж. Методы поиска экстремума. М., 1967.
140. Фадеева В.Н. О фундаментальных функциях оператора X, IV. Труды матем. ин-та им. В.А.Стеклова АН СССР, т. XXVIII, М., 1949.
141. Фмакко А., Мак-Кормек Г. Нелинейное программирование. Методы последовательной безусловной минимизации. М., 1972.
142. Форсайт Дж., Молер К. Численное решение систем линейных алгебраических уравнений. - М.: Мир, 1969. – 166 с.
143. Фрайнт М.Я. Применение случайного поиска к задачам оптимального проектирования. Строительная механика и расчет сооружений. 1970, № 1.
144. Фурунжиев Р.И. Вероятностная оптимизация конструкций с учетом критерия надежности. Материалы IX Всесоюз. семинара «Случайный поиск экстремума». Киев, «Наукова думка», 1974. С.140-143.
145. Фурунжиев Р.И. Оптимизация параметров конструкций методом случайного поиска с самообучением. Научно-техническая конф. «Повышение эффективности жилищно-гражданского строительства». Минск, 1971.
146. Чжу С.Я., Прагер В. Последние достижения в оптимальном проектировании конструкций. Сб. пер. «Механика», М., «Мир», № 6, 118, 1969.
147. Черноусько Ф.Л., Баничук Н.В. Вариационные задачи механики и управления. – М.: Наука, 1973.
148. Чирас А.А. Теория оптимизации в предельном анализе твердого деформируемого тела. – Вильнюс: «Минтис», 1971.
149. Чирас А.А., Боркаукас А.Э., Каркаукас Р.П. Теория и методы оптимизации упругопластических систем. – Л.: Стройиздат, 1974. – 280 с.
150. Чирас А.А. Методы линейного программирования при расчете упругопластических систем. Л., Стройиздат, 1969. -198 с.
151. Чирас А.А., Чижас А.П. К вопросу общей постановки задач рационального проектирования упругопластических одномерных систем. Литовский механический сборник. Вильнюс, № 1, 1967.
152. Шалтенис В.Р. Об одном методе многоэкстремальной оптимизации. Автоматика и вычислительная техника, 1971, 3.
153. Эрдниев О. П., Якубов С. Х. Вычислительный эксперимент оптимизации конструкции бесшарнирной арки постоянной толщины//IX Международная конференция по математическому моделированию, посвященная 75-летию Владимира Николаевича Врагова (Якутск, 27 июля – 1 августа 2020 г.): Тезисы докладов /под ред. д.ф.-м.н. С.В. Попова. - Якутск: Издательский дом СВФУ, 2020. - С 111.
154. Яковлев М.Ф., Левченко И.С., Спири В.Е. О численном расчете на прочность цилиндрической оболочки, подкрепленной ребрами // Численные методы решения задач

теории упругости и пластичности: Материалы IV Всесоюз. Конф. – Ч.2. – Новосибирск, 1976. – С. 114-125.

155. Якубов С.Х. Системный анализ оптимизации проектирования инженерных конструкций и сооружений //Проблемы оптимизации сложных систем: Докл. Седьмой междунар. Азиатской школы-семинара. – Ташкент, 2011. – с.154-163.

156. Якубов С.Х. Оптимизация проектирования осесимметричных усеченных конических оболочек // Вестник Каракалпакского отделения АН РУз, Нукус, 2012, №2. - С. 95-98.

157. Якубов С.Х. Оптимальное проектирование кольцевых пластин переменной толщины // Вестник Каракалпакского отделения АН РУз, Нукус, 2012, №3. - С. 8-11.

158. Якубов С.Х. Модели и методы оптимального проектирования инженерных конструкций типа арки // Архитектура. Строительство. Дизайн. - Ташкент, 2012, №4. - С. 38-40.

159. Якубов С.Х. Математические модели и алгоритмы оптимизации цилиндрических оболочек // Вестник ТУИТ. - Ташкент, 2014, №1. - С. 117-121.

160. Якубов С.Х., Соатов Х.С. Автоматизация процесса проектирования конструкции свода подземных сооружений//Вестник ТУИТ. - Ташкент, 2014, №2. - С. 92-96.

161. Якубов С.Х. Автоматизация процесса проектирования инженерных конструкций. Проектирование оптимальных конструкций. Монография. - Deutschland, Palmarium Academic Publishing, Saarbrucken, 2014. - 77 p.

162. Якубов С.Х., Соатов Х.С., Рахманов К.С. Алгоритмическая система для решения классов оптимизационных задач//Свидетельство о депонировании объектов интеллектуальной собственности. Регистрационный № 2129. Программа ЭВМ. 14.10.2014 г.

163. Якубов С.Х., Соатов Х.С., Рахманов К.С. Алгоритмы решения нелинейных и многомерных задач оптимизации инженерных конструкций и сооружений // Свидетельство о депонировании объектов интеллектуальной собственности. Регистрационный № 2149. Программа ЭВМ. 06.11.2014 г.

164. Якубов С.Х. Алгоритмические методы построения баз знаний в интеллектуальных системах//Современные материалы, техника и технологии. Научно-практический журнал, 2016, №1(4). - С. 238-243.

165. Якубов С.Х. Методы и алгоритмы синтеза и анализа конструкторских и технологических решений в системе автоматизированного проектирования инженерных конструкций и сооружений. - М.: ИНФРА-М, 2019. - 164 с.

166. Yakubov S.H. To a question of generalization of laws of weight optimization thin-walled constructions. Fifth World Conference on Intelligent Systems for Industrial Automation, WCIS – 2008. b-Quadrat Verlag, 2008. P.128-132.

167. Yakubov S. H. Models and algorithms for decision making in computer-aided design engineering for constructions and buildings// Proceedings of Eleventh International Conference on Application of Fuzzy Systems and Soft Computing (Paris, France, September 2-3, 2014), «ICAFS – 2014», b – Quadrat Verlag, Paris, 2014.- pp. 111-118.

168. Yakubov S. H. Construct a mathematical model for the multidimensional and multicriteria optimization problems cylindrical shells on the basis of system approach. Proceedings of Eighth World Conference on Intelligent Systems for Industrial Automation – «WCIS – 2014», b – Quadrat Verlag, Tashkent, 2014. - Pp.479-482.

169. Yakubov S. H. Problems of development algorithmic methods of construction of knowledge bases in the intellectual //Ninth World Conference “Intelligent Systems for Industrial Automation”, WCIS-2016, 25-27 October 2016, Tashkent, Uzbekistan, b – Quadrat Verlag, 2016, pp. 264-267.

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

**More
Books!**



yes
I want morebooks!

Buy your books fast and straightforward online - at one of world's fastest growing online book stores! Environmentally sound due to Print-on-Demand technologies.

Buy your books online at
www.morebooks.shop

Покупайте Ваши книги быстро и без посредников он-лайн – в одном из самых быстрорастущих книжных он-лайн магазинов! окружающей среде благодаря технологии Печати-на-Заказ.

Покупайте Ваши книги на
www.morebooks.shop

KS OmniScriptum Publishing
Brivibas gatve 197
LV-1039 Riga, Latvia
Telefax: +371 686 20455

info@omniscryptum.com
www.omniscryptum.com

OMNIScriptum



FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY