

Глава 5.

В этой главе рассматриваются математические основы МДЦ. Для чтения этой главы, в отличие от остальных глав книги, требуется некоторая математическая подготовка. Впрочем, здесь мы не стали описывать сложные математические проблемы, связанные с топологией векторных пространств, алгебраической ренциальной геометрии, которые нам при ботки метода. В связи с этим данная глава доступна любому специалисту тики.

Прежде всего дадим математическую постановку задачи. В n -рному линейному пространству. Здесь мы будем считать, что заранее задана W , природу которого пока уточнять не будем. В частности, W функций, аргументом которых является время. Относительно критериев-прежнему предполагать, что их конечное число и, более того, m -мерным вектором, т.е. в качестве \mathbf{R}^m . Будем считать, что связь между решениями и значениями f , действующим из пространства W \mathbf{R}^m . Множество допустимых X . Конечно, $X \subset W$.

и-рых целей, которое в выбранных обозначениях является образом X f , т.е. $f(X)$. Напомним, что нас интересует случай $m \geq 3$, для ж-ную задачу. Эта задача усугубляется тем, что $f(X)$ рассматриваемых нами задачах, т.е. заданием множества X f , а не самого $f(X)$. В связи с этим приходится изучать различные конкретные задачи W , множеств X f . В § 1.4 уже упомянутых задачи:

- 1) W – линейное конечномерное, множество X – k -ый многогранник, а f – линейное;
- 2) $f(X)$ – ограниченное выпуклое;

3) W – конечномерное малой размерности, m
 $f(X)$ – ограниченное, не обязательно выпуклое.

Следует отметить, что, во-первых, к этим базовым задачам св

. Так, к ним можно свести такие задачи, как по-

ренциальных уравнений (дифференциальных включений), что сделано в §
 5.6, или для краевых задач для систем в частных пр изводных и т.д. Во-
 вторых, эти три типа задач соответствуют логике (и истории) развития
 МДЦ. В-третьих, методы построения ОЭП для этих задач аналогичны
 методам построения множества достижимых целей.

$f(X)$ о а-
 Q , пол
 равенств. В

$f(X)$ аппроксимируе а-

R^n . При построении ОЭП в первых двух случаях эта оболочка ап-
 Q

пространстве критериев. В третьем случае ОЭП аппроксимируется сум-
 ния.

Итак, в основу методики визуализации множеств достижимых целей

два этапа:

I этап –

$f(X)$ (или его ОЭП) в явном виде, удобном для визуализации. Этот этап
 должен выполняться автоматически, без участия человека (возможно, на
 мощных ЭВМ). Его можно трактовать как предобр и-
 ваемой задаче.

II этап – собственно визуализация, использующая явное пред е-
 ние множеств. Этот этап является существенно интерактивным (челове-
 -машинным), и поэтому на нем выдвигаются жесткие тр
 удобству и скорости получения изображения. Этот этап разумно реализо-
 вать на персональном компьютере или графической станции.

простыми объектами, демонстрация которых на ди о-
 стро. В книге уже было приведено достаточно примеров визуализации
 множества достижимых целей и его ОЭП. В первых двух случаях виз а-
 лизация основана на “быстрых” алгори
 сечений многогранника (или суммы многогранника и конуса). В третьем о-
 жения двумерных сечений аппроксимирующих шаров и конусов, либо на
 основе построения и изображения контура их объединения, что также
 лнено достаточно быстро.

Глава состоит из шести параграфов. В первых трех описываются методы $f(X)$ для трех базовых задач. В § 5.4 рассматривается методика аппроксимации ОЭП. В § 5.5 описано построение многоугольного множества, в § 5.6 – $f(X)$ для динамических систем.

5.1. Построение образа выпуклого многогранника при

Рассмотрим первый тип задач, когда пространство W – \mathbf{R}^n , множество $X \subset \mathbf{R}^n$, отображение $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ является линейным. Предполагается, что множество X задано системой ограничений, т.е.

$$X = \{x \in \mathbf{R}^n : Hx \leq h\}, \quad (1.1)$$

где H – заданная матрица и h – заданный вектор. Линейное отображение $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ задано формулой

$$f(x) = Fx. \quad (1.2)$$

$$f(X) = \{y \in \mathbf{R}^m : y = Fx, Hx \leq h\} \quad (1.3)$$

$$Q = \{y \in \mathbf{R}^m : Dy \leq d\}$$

в начале 70-х нами (см. [29]) было предложено использовать методы построения (ортогональных) проекций многогранных множеств на основе

Проекция многогранных множеств и свертывание неравенств.

Пусть $M \subset \mathbf{R}^p \times \mathbf{R}^q$ – (ортогональной) множества $M \subset \mathbf{R}^q$.

$$M_w = \{w \in \mathbf{R}^q : \exists v: (v, w) \in M\}.$$

Покажем, каким образом множество $f(X)$ может быть представлено в виде проекции некоторого множества. Рассмотрим график отображения $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ типа (1.2), заданного на множестве (1.1)

$$Z = \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^m : y = Fx, Hx \leq h\}. \quad (1.4)$$

Заметим, что множество $f(X)$, определенное согласно (1.3), можно

$$f(X) = \{y \in \mathbf{R}^m : \exists x: y = Fx, Hx \leq h\}.$$

$$f(X) = \{y \in \mathbf{R}^m : \exists x: (x, y) \in Z\},$$

т.е. множество $f(X)$ является проекцией множества $Z \subset \mathbf{R}^n$. По (1.1), (1.2), задача построения $f(X)$ сводится к построению множества $Q \subset \mathbf{R}^m$. Напомним, что эта проекция должна быть построена как множество решений системы линейных неравенств.

Заметим, что при $m = n$ вектор x может быть найден очень легко: $x = F^{-1}y$. Поэтому

$$f(X) = \{y \in \mathbf{R}^m : HF^{-1}y \leq h\}.$$

Таким образом, вектор x множества $f(X)$ за счет простого обращения матрицы F (а именно эта ситуация обычно встречается в прикладных задачах) этим простым способом задачу нахождения множества $f(X)$ в \mathbf{R}^m полностью решить не удастся. Импедансы позволяют исключить из описания (1.4) только часть координат вектора x . При этом они выражаются через остальные и подставляются в неравенства. Для того чтобы исключить оставшиеся координаты, можно использовать методы свертывания систем линейных неравенств, предложенные в первой половине XIX века.

$$M \subset \mathbf{R}^p \times \mathbf{R}^q$$

$$M_w = \{w \in \mathbf{R}^q : Dw \leq d\}. \quad (1.5)$$

При этом предполагается, что выпуклое многогранное множество M имеет вид

$$M = \{(v, w) \in \mathbf{R}^p \times \mathbf{R}^q : Av + Bw \leq c\}, \quad (1.6)$$

где A, B, c – заданные матрицы и вектор, а задача построения M_w сводится к построению множества $Dw \leq d$.

Поскольку методы свертывания недостаточно известны, мы в дальнейшем дадим их краткое описание.

Метод свертывания Фурье.

В предложенном Фурье, последовательно строятся проекции множества M на $\mathbf{R}^{p-1} \times \mathbf{R}^q, \mathbf{R}^{p-2} \times \mathbf{R}^q, \dots, \mathbf{R}^q$.

Исключение производится сложением неравенств, предварительно умноженных на некое число. Этот процесс называется свертыванием неравенств. Метод Фурье в

уравнений, однако в нем нужно учитывать, что неравенства можно трицательные числа.

$p = 1$, т.е. $v \in \mathbf{R}^l$. Тогда систему неравенств (1.6)

$$a_i v + \langle b_i, w \rangle \leq c_i, i = 1, 2, \dots, N, \quad (1.7)$$

a_i – числа, b_i – векторы. Неравенства системы (1.7) ра и-
 Π_+ , Π_- , Π_0 н-
 a_i v . В систему, описыва M_w , сле-
 Π_0 , а также все во

$$\langle a_j b_i - a_i b_j, w \rangle \leq a_j c_i - a_i c_j, i \in \Pi_-, j \in \Pi_+,$$

Π_+ , а другое – Π_- .

д-
 полагается действовать точно так же, взяв систему, полученную после
 исключения предыдущей переменной, в качестве исходной.

Опыт показывает, что если число неравенств в исходной системе со-
 ставляет несколько десятков, то при использовании алгоритма Фурье
 только быстро,

любой современной ЭВМ. Кроме того, сло о-
 гранника, описываемого системой лине ных неравенств, резко растет с
 ростом числа неравенств (подробнее об этом говорится в § 5.5), так что
 лем, возникающих при реализ ции МДЦ.

К счастью, имеется ряд соображений, которые значительно о а-
 ют ситуацию. Главное из них состоит в том, что среди совок о-
 а-

(т.е. они являются зависимыми неравенствами), и поэтому удале-
 ние их из системы не влияет на множество решений. Выявление и е-
 –

ния методов свертывания.

Для того чтобы понять, как устранить зависимые неравенства в рам-
 ках методов свертывания, используем некоторые результаты теории ли-
 нейных неравенств, в частности, леммы Фаркаша и Але –
 Цзи.

–Фань Цзи. в-
 венств. На

ней, в частности, основывается теория двойственности линейного про-
 граммирования. Лемма имеет две формулировки – геометрическую, как
 утверждение о двойственных многогранных конусах, и алгебраическую.
 нятий.

$$K = \{b: b = A\lambda, \lambda \geq 0\},$$

A – матрица, составленная из произвольной конечной совокупности векторов $\{a_1, a_2, \dots, a_m\}$.

Конусом,

K , называется множество

$$K^+ = \{u: \langle u, b \rangle \geq 0, b \in K\}.$$

K , заданного матрицей A , сопр

$$K^+ = \{u: uA \geq 0\}.$$

K^+ , в свою очередь, можно определить сопряженный

$$K^{++} = \{w: \langle u, w \rangle \geq 0, u \in K^+\}.$$

(геометрическая формулировка). Для многогранного конуса K

$$K^{++} = K.$$

(алгебраическая формулировка). Пусть дана матрица A . Для некоторого вектора b

$\lambda \geq 0$, что $b = A\lambda$, тогда, когда для любого решения системы $uA \geq 0$ имеет

$$\langle u, b \rangle \geq 0.$$

Очевидным следствием леммы Фаркаша, удобным для анализа методов свертывания, является лемма Александра–Фань Цзи.

Система $Av \leq b$ разрешима тогда и только тогда, когда для любого решения системы

$uA = 0, u \geq 0$

$$uA = 0, u \geq 0 \tag{1.8}$$

$$\langle u, b \rangle \geq 0.$$

U матрицу, строки которой представляют собой конечное число векторов, порождающих многогранный конус решений (1.8). Тогда лемма может быть переформулирована в следующем виде.

Система $Uv \leq b$ разрешима тогда и только тогда, когда

U – матрица, строки которой являются векторами, порождающими конус решений системы (1.8). Система $Uv \leq b$

разрешима тогда и только тогда, когда

$$Ub \geq 0.$$

Как уже говорилось, на лемме Фаркаша базируется теория линейного программирования. В свою очередь, лемма Александра–Фань Цзи

используется в теории

многранного множества (1.6) следующим образом. Рассмотрим систему

Рассмотрим систему

$$Av \leq b, \quad \text{где } b = c - Bw.$$

–Фань Цзи,

$w \in M_w$ имеет место тогда и только тогда, когда

$$U(c - Bw) \geq 0.$$

$$M_w = \{ w \in \mathbf{R}^q: UBw \leq Uc \}. \quad (1.9)$$

U согласно (1.8) неотрицательны, то следствием леммы является тот факт, что элементы матрицы D

д
венств, что и проис-
ходит в алгоритме Фурье.

Заметим, что существует бесконечное число матриц U , породив конус решений (1.8). Среди них есть матрицы с минимальным числом строк. Такие матрицы U_f а-ми решений (1.8). Известно, что алгоритм Фурье при $p > 1$ обычно не позволяет получить систему (1.9), соответствующую U_f , т.е. такую что $D = D_f = U_f B$, $d = d_f = U_f c$.

Идеи, которые могут быть использованы для построения U_f , предложены в работе Т.С.Мощкина и других [44]. В связи с тем, что эти тельств, фон Бургер [64] переформулировал их и дал доказательства. На основе этих утверждений С.Н.Черников сформулировал алгор с-тем неравенств, позволяющие найти D_f d_f . Опишем идею алгори

Алгоритмы свертывания Черникова.

е-
конуса, т.е.

таких векторов, которые не могут

векторов, принадлежащих этому конусу. Известно, что минимальный набор векторов, порождающих острый (не соде ненулевой размерности) конус – купность его крайних векторов. Конус решений (1.8) содержится в неотрицательном ортанте, поэтому он ров.

$$D_f \quad d_f$$

результат, получе методов свертывания систем линейных неравенств. Мы при это утверждение в несколько иной форме, которая, на наш взгляд, более проста для понимания, так как опирается на общеизвестное понятие аффинной независимости векторов. Оказывается, найти

¹ Материал, излагаемый в следующем разделе, представляет собой упр изложение некоторых идей книги [56].

векторы конуса решений системы (1.8) можно на основе проверки A .

Векторы $a_i, i = 1, 2, \dots, r$, называются u -
 , если существует тако $\lambda \in \mathbf{R}^r$, что

$$\sum_{i=1}^r \lambda_i a_i = 0, \quad \sum_{i=1}^r \lambda_i = 0, \quad \exists \lambda_i \neq 0, \quad i = 1, 2, \dots, r.$$

Необходимое и достаточное условие того, что некоторое реш (1.8) является крайним вектором, дано в следующей лемме, кот приводим без доказательства.

Вектор u , принадлежащий конусу решений системы (1.8), является крайним вектором конуса тогда и только тогда, когда совокуп A , соответствующая положительным коорд u , аффинно независима.

$$D_j \quad d_j$$

ия. Обозначим через I_j системы (1.6), которые вошли в линейную комбинацию при образовании j -го неравенства системы (1.5). Это множество n о-го неравенства. Из определения аффинности I_j ясно, что если

$$I_i \subset I_j \quad (1.10)$$

для некоторого неравенства (1.5) с номером i , то строки матрицы A I_j аффинно зависимы. Из сформулированной леммы можно с-теме (1.5): если $I_i \subset I_j$, что выполняется (1.10), то j -неравенство является зависимым.

Оказывается, что этого соображения вполне достаточно, чтобы мо-

$U_j B w \leq U_j c$. Этот алгоритм был назван

Легко показать, что матрица из p $(p+1)$ аффинно независимых строк. Поэтому если $|I_j| > p + 1$ ($|I_j|$ – е-ство элементов в индексе), то j -мим. По- (1.10), в котором требуется осуществлять попарное сравнение индексов для всех пар неравенств системы, p проверкой числа элементов в индексе. Алгоритм, использующий такую проверку, называется 2 .

вания.

Опишем его шаг. Пусть из системы (1.6) исключены перемене v о-лучена система типа (1.5), причем для каждого i - (1.5) I_i . Пусть, далее, требуется исключить из систе-

² Возможно, что кроме подсчета числа элементов индекса существуют и другие предварительные проверки, которые позволят реже проверять условия (1.10).

мы (1.5) еще одну переменную. Как и в методе Фурье, все неравенства Π_+ , Π_- , Π_0 . Прежде всего, в новую систему Π_0 , сохранив индекс неравенств без изменения. Далее, в новую систему будем включать линейные комбинации пар неравенств, одно из которых взято из Π_+ , а другое – Π_- . В отличие от алгоритма Фурье, при этом будем образовывать не все комбинации, а только те, индексы которых удовлетворяют условиям, описываемым ниже. Пусть в системе (1.5) имелись неравенства $i \in \Pi_+$, $j \in \Pi_-$.

$$b_i^1 w_1 + \sum_{k=2}^q b_i^k w_k \leq c_i,$$

$$b_j^1 w_1 + \sum_{k=2}^q b_j^k w_k \leq c_j.$$

$$\sum_{k=2}^q (b_j^1 b_i^k + b_i^1 b_j^k) w_k \leq b_j^1 c_i + b_i^1 c_j$$

с индексом $(I_i \cup I_j)$ включим в новую систему тогда и только тогда, когда

$$1) |I_i \cup I_j| \leq p + 2;$$

$$2) I_k \subseteq I_i \cup I_j, \quad k\text{-го неравенства системы (1.5), что}$$

$$I_k \subseteq I_i \cup I_j.$$

На этом шаг алгоритма завершается. Заметим, что условие 1) сформулировано здесь в упрощенном виде, на самом деле его можно несколько усилить.

Обсуждение алгоритмов. Обратим внимание на то, что, хотя число получаемых неравенств значительно меньше, чем в методе Фурье, число операций не уменьшается. Это связано с тем, что в методе Фурье для каждого неравенства необходимо вычислять значения b_i^k и c_i . Сказанное относится к числу операций. Расчеты показывают, что эффективность методов сводится к тому, что для систем очень малой размерности ($p \approx 10$, $n \approx 10$) в то же время, встречаются практические задачи, в которых методы свертывания эффективны, поскольку численно. Опыт показывает, что большое влияние на протекание процесса

³. Таким образом, эти методы гарантируют получение конечного результата лишь для систем очень малой размерности ($p \approx 10$, $n \approx 10$). В то же время, встречаются практические задачи, в которых методы свертывания эффективны, поскольку численно. Опыт показывает, что большое влияние на протекание процесса

ленно. Опыт показывает, что большое влияние на протекание процесса

³ темы, а также на процесс исключения переменных в алгоритме Фурье рассмотрено в работах [38-40].

ть исключения переменных, поэтому

ния.

Обратим внимание на то, что даже минимальная матрица U_f M_w . Действительно, матрица U_f шений (1.8), который A , но не от конкретных B c . Более того, матрица U_f должна содержать строки, дающие описание B c !

ничений. Для того чтобы расширить область применения методов свертывания, требуется использовать специальные методы, предназначенные для систем неравенств.

Прежде всего, дадим математическую формулировку проблемы. В

$$\langle a_i, x \rangle \leq b_i, i = 1, 2, \dots, N,$$

$a_i \in R^p$ – заданные строки, b_i – заданные числа, $x \in R^p$, необходимо выделить подсистему, имеющую то же множество решений, но не состоящую из независимых неравенств.

Отметим, что проблема устранения зависимых неравенств в линейном программировании (ЛП) ⁴. Большинство методов основано либо на графическом методе, либо на некоторых эвристических приемах. Выяснилось, что не удается найти универсальное решение несущественных ограничений, эффективных для решения задач. Одни алгоритмы работают быстро, но требуют слишком больших временных затрат, другие работают быстро, но не дают никаких оценок того, сколько еще осталось ⁵. Поэтому каждая из задач требует экспериментов

ных неравенств.

Таким образом, методы свертывания систем линейных неравенств n -мерных множеств при их линейных отображениях, если число переменных не превышает нескольких десятков. Подчеркнем, что при их исследовании и “почти зависимых” неравенств, а также при исключении переменных. Это означает, что при использовании методов свертывания не удается автоматизировать, и приходится

⁴ См., например: [75].

⁵ Предложенные нами методы описаны и обсуждены в [6, 7].

ку, хорошо знающему особенности того или

6.

Существенным недостатком методов свертывания является то, что в том случае, когда процесс свертывания не удается провести до конца, в результате. С другой стороны, большое достоинство этих методов заключается в возможности пространства образов. В то же время, очевидно, что для моделей $f(X)$ для систем (1.3) при большой размерности пространства образов. В то же время, очевидно, что для моделей пригодны.

Обратим внимание на тот факт, что исключение “почти зависимых” неравенств из системы, описывающей очередную проекцию, приводит к

заменяется задачей ее аппроксимации. Спрашивается, зачем при построении аппроксимации исходить из описания точной проекции, а не

(например, искать аппроксимирующий многогранник с минимальным числом вершин при заданной точности)? Такой подход был использован в методе уточнения оценок проекции [3] с целью преодоления основных недостатков свертывания, состоящего в отсутствии при

и проекции в том случае, когда задачу не удается решить до конца.

Метод уточнения оценок проекции.

проекции (УОП) аппроксимация проекции (1.5) осуществляется на основе $P^0, P^1, \dots, P^k, \dots$, сходя-

$M_w \subset R^p$, т.е. к

р-

фа. При этом последовательность многогранников строится итеративно на основе решения некоторой вспомогательной задачи свертывания.

k -

о-

P^k , причем:

а) $P^k \subset M_w$;

б) вершины P^k

M_w ;

в) P^k

но:

– в виде множества вершин $\{w^{(1)}, w^{(2)}, \dots, w^{(r)}\}$;

–

$$P^k = \{w \in R^q : \langle c_j, w \rangle \leq d_j, j = 1, 2, \dots, N\}, \quad (1.11)$$

$c_j - R^q, d_j - \text{числа.}$

⁶ В свое время был реализован пакет прикладных программ “Потенциал”, с помощью которых решались задачи оптимизации с различными ограничениями. Целью работы является использование методов свертывания [9].

P^{k+1} состоит в следующем. Для всех граней P^k , задав c_j $d_j, j = 1, 2, \dots, N$, решая

$$\langle c_j, w \rangle \rightarrow \max \text{ при } (v, w) \in M. \quad (1.12)$$

Среди полученных решений $(v_{(j)}, w_{(j)})$, $j = 1, 2, \dots, N$, выбирается такое, что точка $w^{(j^*)}$ принадлежит P^k . Далее полагается $P^{k+1} = \text{conv} \{ w^{(j^*)}, P^k \}$.

Заметим, что для работы алгоритма на $(k+1)$ - P^k в виде (1.11). Для решения этой задачи в ритм.

- 1) Находятся неравенства системы (1.11), нарушаемые то $w^{(j^*)}$;
- 2) $w^{(i)}, i = 1, 2, \dots, L$, принадлежащие граням, соответствующим нарушаемым неравенствам;
- 3) $K \subset \mathbf{R}^q$, задаваемый совокупностью $w^{(i)} - w^{(j^*)}, i = 1, 2, \dots, L$;
- 4) K линейных неравенств (что эквивалентно построению K^+);
- 5) P^k добавляются неравенства, задаваемые ребрами K^+ $w^{(j^*)}$;
- 6) из описания исключаются неравенства, нарушаемые точкой $w^{(j^*)}$.

венств. Обратим внимание на то, что при решении задачи а \mathbf{R}^m , а не в исходном пространстве переменных. Поскольку m относительно мало, то последнее свойство позволяет применять метод УОП и другие методы, разработанные на его основе, для аппроксимации $f(X)$ в задачах большой размерности ($n \approx 1000$). Наш опы

решения задач построения множества достижимых целей. Было подмечено, что метод УОП может быть использован для аппроксимации любых ограниченных выпуклых множеств, для которых можно рассчитать опорную функцию, т.е. решить задачу (1.12) для любого c_j . Именно в такой постановке рассматриваются современные алгоритмы достижимых целей, рассмотренные в следующем параграфе. На основе метода УОП был предложен алгоритм, в которых использованы следующие идеи метода УОП: – итеративное построение многогранников аппроксимации;

- P^k, P^{k+1} на два независимых шага:
- 1) выбор некоторой "присоединяемой" точки w^* тех или иных задач оптимизации;
 - 2) построение $\text{conv} \{P^k, w^*\}$ в виде (1.11), которое осуществит.

5.2. Адаптивные алгоритмы аппроксимации

многогранников, аппроксимирующих с любой степенью точное компактное тело (т.е. множество с непустой внутренней частью). Пусть R^m . Пусть в R^m

(в.к.т.) C , т.е. для любого u , принадлежащего R^m : $\langle u, u \rangle = 1$, можно рассмотреть

$$g_C(u) = \max \{ \langle u, y \rangle : y \in C \}.$$

Итерационные алгоритмы аппроксимации в.к.т. многогранников. аппроксимации в.к.т. C

$P^0, P^1, \dots, P^k, \dots$ с растущими, в которых последующий многогранник строится на основе предыдущего, с использованием некоторых рамок C . Далее будем рассматривать алгоритмы, позволяющие каждого в.к.т. $C \subset R^m$ P^k ,

$$\lim_{k \rightarrow \infty} \delta(P^k, C) = 0,$$

$\delta(\cdot, \cdot)$ – метрика Хаусдорфа, т.е.

$$\delta(C_1, C_2) = \max \{ \sup \{ d(x, C_2) : x \in C_1 \}, \sup \{ d(x, C_1) : x \in C_2 \} \}.$$

Такие алгоритмы будем называть аппроксимирующими.

ции, основанных, от алгоритмов,

тел (таких как симплекс, параллелепипед, эллипсоид), является возможность аппроксимации в.к.т. с любой степенью точности. За это преимущество, однако, приходится платить довольно высокой ценой: как показывают практика и теоретические оценки, сложность

⁷ или гиперграней, т.е. граней размерности $m-1$.

размерности аппроксимируемого тела. Тем не менее, мы были вынужде-
 с тем, что в МДЦ нас интересует форма границы аппроксимируемого
 множества, а не только область, где это множество находится.

разработки алгоритмов, оптимальных с точки зрения сложности описания
 аппроксимирующих многогранников. Кроме того, по

ни, представляют интерес
 алгоритмы, оптимальные с точки зрения числа вычислений опорной
 функции. Такие алгоритмы рассматриваются на графе.

е-
 дующего многогранника по предыдущему. В данном параграфе, в основ-
 ном, рассматриваются алгоритмы, использующие так называемые схемы
 выполнения. Обозначим через $P(C)$ множество многогранников, вершины которых принадле-
 жат границе ∂C аппроксимируемого в.к.т. C .

$P^k \in P(C)$. Тогда $(k+1)$ -я итерация состоит из двух шагов.

Шаг 1. Выбирается точка $y^* \in \partial C$;

Шаг 2. Строится $P^{k+1} = \text{conv} \{ y^*, P^k \}$.

Очевидно, что если в некоторой схеме выполнения многогранника

$P^0 \in P(C)$, то и $P^k \in P(C)$ для любого

k . Если для многогранника $P \in P(C)$ через $N(P)$ обозначить
 число вершин, то

$$N(P^k) \leq N(P^0) + k. \quad (2.1)$$

Обратим внимание на то, что описанный выше алгоритм УОП базируется
 на схеме выполнения.

а-
 ми решения двух задач:

1) $y^* \in \partial C$;

2) $P^{k+1} = \text{conv} \{ y^*, P^k \}$ в требуемом виде.

схем, например, схемы отсечения. Пусть аппроксимируется н
 в.к.т. C . Обозначим через $Q(C)$ класс выпуклых многогранников, граница
 которых принадлежит ∂C .

$Q^k \in Q(C)$. Тогда $(k+1)$ -й шаг.

Шаг 1. Выбирается направление $u^* \in S$;

Шаг 2. Строится $Q^{k+1} = Q^k \cap \{y \in R^m: \langle u^*, y \rangle \leq g_C(u^*)\}$.

К алгоритмам, использующим схему отсечения, относятся ра-
т-
н-
ников.

Адаптивные алгоритмы.

адаптивные (активные) схемы, в которых в методе выбора то $y^* \in \partial C$
 P^k . В этом случае выбор $y^* \in$
 ∂C C в той мере, в какой P^k C .
тивными, нежели неадап-
тивные (пассивные), основанные, как пр вило, на построении априорной
сетки на сфере направлений, т.е. не учитывающие конкретную форму
8. На колько нам известно, впервые адаптивный
па, основанный на схеме восполне-
ния, был предложен для аппроксимации части границы двумерного вы-
пуклого тела [69].

о-
ритмов рассмотрим класс алгоритмов, в дальнейшем обознача
 A^0 . Алгоритмы этого класса основаны на одном частном случае схем
восполнения и являются развитием алгоритма УОП. В алгоритмах класса
 A^0 $y^* \in \partial C$ P^k в-
 P^k в виде множества решений системы линейных неравенств. Для
о на шаге 2 алгоритма восполнения решается достаточно сложная
 $P^{k+1} = \text{conv} \{y^*, P^k\}$ в требуемом виде.

Пусть аппроксимируется в.к.т. C . Обозначим через $U(P)$ коне

о-
 $P^k \in P(C)$. Очевидно, что множество $U(P)$ зад но, если много-
P а-
венств.

A^0 .

Перед началом $(k+1)$ - $P^k \in P(C)$ в
ервенств.

Шаг 1: находим $u^* \in U(P^k)$, на котором достигается

$$\max \{ (g_C(u) - g_{P^k}(u)) : u \in U(P^k) \}; \quad (2.2)$$

$y^* \in \partial C$, что

$$\langle u^*, y^* \rangle = g_C(u^*).$$

Шаг 2: находим $U(P^{k+1})$ для $P^{k+1} = \text{conv} \{y^*, P^k\}$ на основе
построения $\text{conv} \{y^*, P^k\}$ в виде множества решений

8 в работе [14] приведены экспериментальные данные, показывающие, что адап-
п-
тивные.

венств.

Предполагается, что перед началом работы алгоритма задан и $P^0 \in P(C)$, методы построения которого здесь рассматриваться не будут [60].

A^0 . Во-первых, благодаря решению задачи (2.2) можно получить грубую оценку $\delta_{\square}(P^k, C)$.
-вторых, возможна визуальная оценка этого расстояния на основе од-
 P^k о-

$$S^k = \{y \in R^m: \langle u, y \rangle \leq g_C(u), u \in U(P^k)\},$$

C .
-третьих, как и во всех схемах восполнения, в алгоритмах класса A^0
 y^* $U(P^{k+1})$, вообще говоря, не связаны
между собой. Конкретный алгоритм класса A^0

$\text{conv}\{y^*, P^k\}$ на втором. Примером алгоритма класса A^0 о-
ритм УОП.

исследований, показывающие, что схема выбора точки $y^* \in \partial C$, исполь-
 A^0 , приводит к оптимальной по ь-
ности многогранников аппроксимации, а затем обсудим испол
нами методы построения $\text{conv}\{y^*, P^k\}$.

О многогранниках наилучшей аппроксимации.

огогранников, генерируемых алгоритмами, с некоторой “образцовой” по-
следовательностью многогранников, дающих на а-
цию. Многогранники наилучшей аппроксим
следующим образом.

Пусть задано некоторое в.к.т. C . Классическим результатом те
выпуклых множеств [71, 72] является тот факт, что среди мног
с числом вершин, не более чем N (класс таких многогра

$P_N(C)$, на $P_N \in P_N(C)$, на котором достигается
 $\delta_{\square}(C, P_N)$. Этот многогранник н и-
(м.н.а.). Он м а-

C . Известно, что $\delta(C, P_N) \rightarrow 0$ при $N \rightarrow \infty$. Для в.к.т. с дважды не-
н-

k_C K_C такие, что

$$k_C N^{-2/(m-1)} \leq \delta_{\square}(C, P_N) \leq K_C N^{-2/(m-1)}. \quad (2.3)$$

в-
е-
довательностью м.н.а.

Хаусдорфовы схемы.

о-
о-

вых схем восполнения (и отсечения) и хаусдорфовых алг ритмов, которые были введены в работах Г.К.Каменева [14, 15]. Схему восполнения называют хаусдорфовой для в.к.т. C $\gamma > 0$, если она порождает последовательность многогранников $\{P^k\}_{k=0,1,\dots}$

$$\delta_{\square}(P^k, P^{k+1}) \geq \gamma \delta_{\square}(P^k, C), k = 0, 1, \dots$$

Показано, что ха щими.
 $N(P^k) -$ P^k . Доказано, что для в.к.т. с дважды диффе-
 $\delta_{\square}(C, P^k)$ асимптот
 $N(P^k)^{-2/(m-1)}$, т.е. совпадает с порядком аппро симации м.н.а. Это утвер-
 ку числа вершин.

и-
 малыны по порядку числа вершин, возникает вопрос об асим
 значениях постоянных аппроксимации. Пр е-
 ских и экспериментальных исследований в этой области.

$F = \{P^k\}_{k=0,1,\dots}$ – последовательность многогранников, сходя-
 щаяся к в.к.т. C .

$$\eta(F) = \liminf_{k \rightarrow \infty} \frac{\delta(C, P_{N(P^k)})}{\delta(C, P^k)}$$

F . Для
 последовательностей м.н.а. имеем $\eta(F) = 1$; для последовател ностей, не
 оптимальных по порядку числа вершин, имеем $\eta(F) = 0$; в остальных
 случаях имеет место $0 < \eta(F) < 1$.

Показано, что для в.к.т. с трижды непрерывно дифференциру е-
 F γ

$$\eta(F) \geq \frac{1}{4} \left(\frac{m-1}{m+1} \frac{\mathcal{G}_{m-1}}{m} \gamma^m \right)^{2/m-1} \frac{\rho_{\min}}{\rho_{\max}},$$

ρ_{\min} ρ_{\max} – ∂C , а
 \mathcal{G}_m – R^m ми. В част-
 ности, \mathcal{G}_2 составляет около 1.2, \mathcal{G}_3 составляет около 1.5, а при $m \geq 4$ отно-
 \mathcal{G}_m имеют место оценки, например, $1.65 \leq \mathcal{G}_4 \leq 1.77$.

A^0 . т-
 мы обладают замечательными свойствами, возникает вопрос о том, суще-
 ления, являющиеся хаусдорфовыми. Было пока-
 зано [16], что алг A^0 :

- 1) ю-
 бого в.к.т. C ;
- 2) γ в процессе аппроксимации стрем
 $1/\alpha(C)$, где $\alpha(C) -$ C ;

3) для в.к.т. с дважды непрерывно дифференцируемой границей об-
 ы; стремящейся к единице.

Итак, теоретическое исследование алгоритмов класса A^0 показывает, что
 для гладких тел они оптимальны по порядку числа вершин. Б лее того,
 для в.к.т. с трижды не н-

$$\eta(F) \geq \frac{1}{4} \left(\frac{m-1}{m+1} \frac{\rho_{m-1}}{m} \right)^{2/m-1} \frac{\rho_{\min}}{\rho_{\max}}.$$

Было осуществлено и экспериментальное исследование [12] одн
 A^0 . Рассматривалась аппроксимация двух-, трех-
 четырехмерных эллипсоидов, для которых удалось получить эксперимен-
 ритма. Важность ляется тем фактом, что
 к ним принадлежит шар, являющийся наиб
 аппроксимации в метрике Хаусдорфа. Помимо сказанного, задача эффек-
 б-

следующие основные выводы:

- 1) A^0 , как и следует из теоретического анализа, явл о-
 гранников;
- 2) - A^0 превосходит 1/2; е-
- 3) тет с ростом размерности; A^0
- 4) ла. A^0

Таким образом, теоретические и экспериментальные исследования под-
 н-
 л-
 нения.

"под-над". л-
 нения, т.е. к описанию методов построения $\text{conv} \{ y^*, P^k \}$. В фундамен-
 тальной книге Препараты и Шеймоса [51] рассмотрена схема построения
 выпуклой оболочки многогранника и точки, к

"под-над" ("beneath-beyond"). Опишем эту схему на трехмерном прим ре.

Пусть имеется многогранник ABCDEF (см. рис. 2.1a), и требуе о-
 ки G. Не-
 трудно понять, что в эту новую выпуклую оболочку войдут все грани
 исходного многогранника, не видимые из точки G (т.е. ABCD, BCE, CDE,

DEF и ADF), и не войдут грани, видимые из точки G (ABF и BEF, заштрихованные на рисунке). Кроме того, выпуклые новые грани (см. рис 2.1б). Этими новыми гранями являются грани минимального конуса с вершиной G, содержащего многогранник ABCDEF.

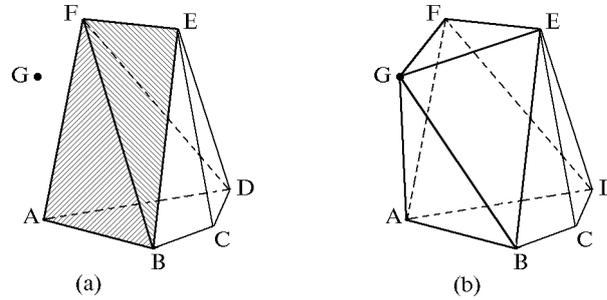


Рис. 2.1

Каждая такая новая грань лежит в гиперплоскости, проходящей через ребро многогранника, лежащее на границе между видимой и невидимой из точки G частями поверхности многогранника. Эта граница, выделенная на рисунке жирной линией, состоит из ребер AB, BE, EF и FA. Таким образом, каждая новая грань выпуклой оболочки проходит через пересечение

многогранника, одна из которых видима, а другая — невидима из точки G. Например, новая грань BEG проходит через ребро BE, которое является пересечением видимой грани BEF и невидимой грани BCE. Эта схема легко обобщается на случай пространства R^m , если учесть, что граница между видимой из точки G частями многогранника в R^m имеет размерность $m-2$.

схемы "под-над" требуется леммы:

- определить, является ли гипергрань видимой из точки;
- определить, являются ли две гиперграни соседними;
- преобразовать представление многогранника к новому виду.

е-

ния многогранника и определяют конкретный вид алгоритма.

алгоритмы, основанные на методе "под-над", которые были предложены в неопубликованных работах Kallay и Seidel. В работе Kallay для представления многогранника хранятся координаты его вершин, а также для каждой гиперграни задана ориентация полупространства, граница которого проходит через вершины данной гиперграни, по отношению к присоединяемой точке. Для ре-

об ориентации полупространства, граница которого проходит через вершины данной гиперграни, по отношению к присоединяемой точке. Со-

деляется при помощи графа принадлежности.

Б-
бинаторная процедура. Алгоритм Seidel является двойственным к алгоритму Kallay, он оказался оптимальным по порядку числа раций в пространствах четной размерности. Информация об опыте реализации и использования этих алгоритмов отсутствует.

а-
венств.

и-
симо от Kallay и Siedel. Описанный в п. 5.1 алгоритм Буше реализует схему “под-над”. Поскольку для реализации МДЦ требовалось

линейных неравенств, реализация схемы “под-над” в наших исследованиях оказалась иной, чем в чисто комбинаторных алгоритмах типа Kallay. В

еще один алгоритм (алгоритм Черных) [59], используя а-
стоящее время.

Описание алгоритма Черных начнем с того, что рассмотрим построения выпуклой оболочки заданной совокупности точек $\{v^1, v^2, \dots, v^s\} \subset \mathbf{R}^m$ в виде множества решений системы линейных неравенств. Согласно определению, точка $y \in \mathbf{R}^m$

$\{v^1, v^2, \dots, v^s\}$, если существуют такие $\lambda_1, \lambda_2, \dots, \lambda_s$, что

$$y = \sum_{i=1}^s \lambda_i v^i, \quad \sum_{i=1}^s \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = 1, 2, \dots, s. \quad (2.4)$$

$\mathbf{R}^s \times \mathbf{R}^m$ $\lambda_1, \lambda_2, \dots, \lambda_s$ у. Рас-

Λ в этом пространстве, зад

системой (2.4). Согласно определению проекции, выпуклая совокупность точек $\{v^1, v^2, \dots, v^s\}$ является проекцией множества Λ \mathbf{R}^m у. Таким образом, для по

выпуклой оболочки достаточно из системы (2.4) исключить переменные λ .

λ из системы (2.4) в порядке их номеров. Будем считать, что $s > m + 1$ и точки находятся в общем положении. Тогда все требуемые детерминанты не равны нулю. Тогда переменные $\lambda_1, \lambda_2, \dots, \lambda_{m+1}$ можно выразить через остальные переменные, используя имеющиеся. Дальнейшее исключение переменных будем проводить по

Можно проверить, что после исключения части переменных $\lambda_1, \lambda_2, \dots, \lambda_q$, где $q \geq m+1$, мы фактически получаем описание выпуклой оболочки v^1, v^2, \dots, v^q (для этого в полученной системе надо лишь приравнять

$\lambda_{q+1}, \dots, \lambda_s$). Более того, исключение λ_{q+1} из системы (2.4) соответствует приравнению v^{q+1}

v^1, v^2, \dots, v^q . Бл

выпуклой оболочки фактически не требуется информация о тех точках, ся, поэтому не нужно заранее знать ни сами будущие точки, ни их число. Таким образом, данный алгоритм последовательно.

Алгоритм Черных также построен по схеме “под-над”. Хранение индексов неравенств, которые используются в алгоритме сокраще фундаментального свертывания, соответствует частичному хранению д- лежности вершин граням. Таким образом, фактиче хранится в памяти в виде системы неравенств, к т- ветствует гиперграницы многогранника, причем при каждом неравенстве хранится информация о номерах вершин, принадлежащих соо ю- щей грани.

Вследствие отличного от Kallay представления многогранника, в ал- основных проблем схемы. Прежде всего, вопрос видимости грани реша- т- во, соответствующее этой грани.

Далее, вопрос о выборе соседних граней в выпуклой оболочке точек v^1, v^2, \dots, v^q становится центральным моментом алгоритма. Если бы при λ из системы (2.4) применялся алгоритм Фурье, то это привело бы к комбинированию всех пар неравенств, соответствующих одной видимой и одной невидимой граням (без учета соседства). Понятно, что в этом случае число n тастрофически, а алгоритм не соответствовал бы схеме “под-над”. Использование сокращенного фундаментального водит к тому, что комбинируются именно такие пары неравенств, которые соответствуют соседним граням, т.е. комбинирование сходит в соответствии со схемой “под-над”.

Черных решается достаточно просто: преобразование индексов не представляет труда, а коэффициенты неравенства, соответствующего новой грани, находятся сложением двух неравенств, описывающих две соседние грани, с положительными весами.

выше, чем в чисто комбинаторном алгоритме Kallay, поскольку для иден- и- мых и невидимых граней. В рамках метода МДЦ, однако, время его работы оказывается вполне удовлетворительным, так как само выяснение

операций, а метод МДЦ обычно не требует той стадии, на которой время выпо доминировать. В случае n

m (до 7-8) алгоритм Черных

позволяет строить выпуклые оболочки множеств, состоящих из нескольких сотен точек даже на персональных ЭВМ.

Как было сказано, алгоритм Бушенкова также построен на о-идеях, составляющих схему “под-над”. В нем при присоединении каждой

v^1, v^2, \dots, v^r , лежащие на границе ви-
гранника. Поскольку в исходном варианте этого алгоритма не предпола-
галось хранить комбинаторную структуру многогранника, то такая сово-

всех вершин многогранника во все неравенства системы. Затем грани
 λ

равенств, определяющих этот конус

$$y - p^* = \sum_{i=1}^r \lambda_i (v^i - p^*), \quad \lambda_i \geq 0, \quad i = 1, 2, \dots, r, \quad (2.5)$$

ния.

Сравним алгоритмы Бушенкова и Черных. Во-первых, алгоритм Чер-
ных имеет более простую структуру и требует меньше времени и памяти,

ной системы (2.5) целый
 λ_i , число которых равно чи- шин, принадлежащих
видимым граням.

-вторых, алгоритм Черных является более надежным, по-
для него разработана методика контроля погрешности, во-
округления. При исследовании этого вопроса оказа-

абстрактного симплицеального комплекса. Благодаря этому удалось лег-
ш-
лой оболочки, которая получается путем подста-

системы, при лиженно описывающей эту выпуклую оболочку. Заметим,

шин выпуклой оболочки (для работы основной части алгоритма Чер-
ных, в отличие от алгоритма Бушенк ва, этого не требуется).

В то же время, в алгоритме Бушенкова не требуется проверять сосед-
ство всех пар, составленных из видимой и невидимой граней. По-
существу, в алгоритме Бушенкова при присоединении очередной верши-
ны рассматривается не многогранник в целом, а лишь его видимая часть и
те грани невидимой части, которые примыкают к ней. Благодар

последовательность локальных подзадач, при решении каждой из кото-
щей многогранник в целом. С одной
стороны, это свойство алгор тма Бушенкова может стать важным, когда
число граней окажется велико. С другой стороны, именно это разбиение

к-
 гранника, что крайне необходимо для контроля погрешности
 при наличии ошибок округления.

л-
дварительных оценках граней.
 л-
 горитма Черных является то, что он позволяет создать эффекти
 л-
 A^0 , в котором почти без дополнительных за

$$g_C(u) - g_{P^{k+1}}(u) \quad (2.6)$$

$$u \in U(P^{k+1}) \setminus U(P^k).$$

Идея предварительного оценивания величин (2.6) для таких но
 была впервые предложена и реализована Каменевым, однако при исполь-

$$\langle c_i, y \rangle \leq d_i \text{ и } \langle c_j, y \rangle \leq d_j$$

$$\Delta_i \quad \Delta_j, \text{ и таким образом имеет вид}$$

$$\langle \Delta_i c_i + \Delta_j c_j, y \rangle \leq \Delta_i d_i + \Delta_j d_j.$$

И пусть известны (например, в результате непосредственного реш
 задачи нахождения значения опорной функции) величины $e_i \quad e_j$ такие,
 т-

$$\langle c_i, y \rangle \leq d_i + e_i \text{ и } \langle c_j, y \rangle \leq d_j + e_j.$$

Очевидно, что тогда все точки аппроксимируемого множества удо
 о-
 также и неравенству, являющемуся следствием этих двух нера-

$$\langle \Delta_i c_i + \Delta_j c_j, y \rangle \leq \Delta_i d_i + \Delta_j d_j + (\Delta_i e_i + \Delta_j e_j). \quad (2.7)$$

Это означает, что величина $\Delta_i e_i + \Delta_j e_j$
 (2.6) для $\Delta_i c_i + \Delta_j c_j$ (т.е. для нормали вновь полученного неравенства).

Наличие оценок (2.7) часто позволяет не решать трудоемкую з
 о-
 жества для части из вновь получаемых неравенств. Опыт показал, что в
 среднем доля неравенств, для
 вычислять опорную функцию, уменьшается с ростом размерности m
 числа вершин многогранника. Так, при $m = 3$ эта доля обычно составляет
 около 80%, а при $m = 6$ – только около 25%. Описанная идея предвари-
 тельного оценивания показывает, что хотя шаги схемы восполнения неза-
 висимы, между ними можно уст
 связь.

Метод сближающихся многогранников.

начале параграфа вопросу об алгоритмах, оптимальных по поря

ов опорной функции аппроксимируемого тела. Ясно, что такие алгоритмы крайне важны в задачах большой размерности (порядка сотен или тысяч). Алгоритм сближающихся многогранников (СМ) обладает таким свойством. Пусть аппроксимируется некоторое в.к.т. C .

$P^k \in \mathcal{P}(C)$ и $Q^k \in \mathcal{Q}(C)$ ⁹. Тогда $(k+1)$ -шагов.

Шаг 1: находим $u^* \in U(P^k)$, на котором достигается

$$\max \{ (g_{Q^k}(u) - g_{P^k}(u)) : u \in U(P^k) \},$$

$$y^* \quad \partial C, \text{ что}$$

$$\langle u^*, y^* \rangle = g_C(u^*).$$

Шаг 2: а) находим $U(P^{k+1})$ для $P^{k+1} = \text{conv} \{ y^*, P^k \}$ на основе построения $\text{conv} \{ y^*, P^k \}$

вершин;

б) положим

$$Q^{k+1} = Q^k \cap \{ y \in \mathbf{R}^m : \langle u^*, y \rangle \leq g_C(u^*) \}.$$

Обратим внимание на тот факт, что на каждой итерации реш

$$g_C(u^*) = \max \{ \langle u^*, y \rangle : y \in C \} \quad C: \text{находит-} \quad y^*.$$

Как видно, алгоритм СМ включает в себя как схему восполнения, так и схему отсечения. Потребовав P^0 , чтобы вершины исходного многогранника Q^0 , получим многогранники, также обладающие этим свойством.

оптимальность метода по порядку числа вершин. Оптимальность вычислений опорной функции очевидна.

Заключение. к-

множеств достижимых целей в случае их выпуклости, ограничиваемых. Для тех случаев, когда множества достижимых целей не являются выпуклыми, были предложены методы, рассматриваемые в следующем параграфе.

⁹ Напомним, что из этого условия следует $P^k \subset C \subset Q^k$.

5.3. Аппроксимация невыпуклых

$f(X)$ в невыпуклом случае состоит в том, что отдельный выпуклый многогранник не может аппроксимировать это множество. Далее, алгоритмы а

$f(X)$. Поэтому метод, описываемый здесь, базируется на идеях вариантного расчета [53]. Задавая различные варианты точек из множества X , можно критериев. В этом случае, однако, возникает проблема выбора точек из X и, в случае, когда число точек велико, проблема представления результатов расчетов.

критериев, $f(X)$. Этот X, P $f(X)$, объединение окрестностей которых $f(X)$ с заданной точностью и полнотой. Изображение графики не представляет труда.

Сейчас нам не важно, как в деталях устроены отображение $f: W \rightarrow R^m$ $X \in W$. Потребуем лишь, чтобы множество $f(X)$ было ограниченным. Во многих приложениях на множестве X $\mu_X, \mu_X \neq 0$, и способ получения достаточно большого числа X , согласованных с этой мерой (например, равномерно распределенных на X). При достаточно общих предположениях f $f(X)$ меру μ_Y так, что мера образа равна мере прообраза:

$$\mu_Y(C) = \mu_X(f^{-1}(C)), C \subset Y. \quad (3.1)$$

$f^{-1}(C)$ понимается полный прообраз множества C X . Обозначим через $d(z, q)$ расстояние между точками $z, q \in R^m$. Пусть $\varepsilon > 0$. Через $(Q)_\varepsilon$ ε - Q , т.е. множество точек, отстоящих от Q на расстояние, меньшее ε . Множество Y' называть (ε, η) - $Y = f(X)$, если оно удовлетворяет следующим двум требованиям:

1) $Y' \subset (Y)_\varepsilon$; ранстве:

2) заданной полноте по мере, индуцированной в пространстве критериев:

$$\mu_Y(Y' \cap Y) \geq \eta \mu_Y(Y), \quad 0 < \eta < 1.$$

Первое требование означает, что для любой точки $y \in Y'$ $x^* \in X$ такое, что $d(y, f(x^*)) < \varepsilon$. Второе требование означает, что в множестве Y' η -аза. Таким образом, первое требование означает, что множество Y' очень размыто, а второе – $Y = f(X)$.

Y некоторое конечное число точек, обо T и назовем базой покрытия. Обозн $\eta_T(s)$ вероятность попадания образа точки из X s - T . Эта

$$\eta_T(s) = \mu_Y((T)_s \cap Y) / \mu_Y(Y).$$

$\eta_T(s)$ назовем функцией полноты покрытия. При $s > 0$ функция $\eta_T(s)$ есть функция распределения случайной величины, равной расстоянию X T . Поэтому для $\eta_T(s)$ можно использовать статистические методы, основанные на расчете $f(x)$ и X .

Y T $\eta_T(s)$. Тогда множество $(T)_s$ есть $(s, \eta_T(s))$ - $f(X)$. При этом величина $1 - \eta_T(s)$ есть ошибка 1-го рода, т.е. вероя б-ки в гипотезе о том, что из $x \in X$ $f(x) \in (T)_s$. Т ким образом, задача построения (ε, η) - $Y = f(X)$ св T , что

$$\eta_T(\varepsilon) \geq \eta. \quad (3.2)$$

Опишем один из способов получения такого множества.

X T из образа этой точки. Далее будем повторять однотипные итерации:

$$- \left(\begin{matrix} N \\ N \end{matrix} \quad \eta \quad \varepsilon \right);$$

$\eta_T(\varepsilon)$ и проверим условие окончания (3.2);

– если (3.2) выполнено, то закончим работу, в противном случае по T тем образом, который отстоит от T с-стояние;

– перейдем к следующей итерации.

Можно показать, что при условии ограниченности $f(X)$ и любых $\eta < 1$ $\varepsilon > 0$ приведенный алгоритм позволяет за конечное число итераций T – базу покрытия, удовлетворяющую условию (3.2), т.е. построить (ε, η) - $f(X)$. Могут быть предл

T , более экономно использ расчетов. Они, однако, выходят за рамки настоящего излож ния [18].

Пусть теперь построена (ε, η) - $Y = f(X)$ в виде $Y' = (T)_\varepsilon$. Множество Y' имеет простое явное описание как система из $|T|$ (где $|...|$ – мощность множества) шаров радиуса ε .

$$\max \{ \lambda_i |x_i^1 - x_i^2| : i = 1, 2, \dots, m \},$$

λ_i – некоторые положительные веса. В этом случае шары, используя $f(x)$, являются параллелепипедами со сторонами, параллельными осям координат, и множество $(T)_\varepsilon$ быстро изображено средствами компьютерной графики (в нашей книге, как можно заметить, использовался именно этот случай). Наконец, для каждой точки $(T)_\varepsilon$ образ T , для которой легко может быть восстановлен образ.

5.4. Об аппроксимации оболочки Эджворта

В этой главе мы обсуждали вопросы построения (и аппроксимации) множества достижимых целей и не касались вопроса его ОЭП. Этот вопрос рассматривается в данном параграфе. Как увидит читатель, задачи построения ОЭП близки к задачам построения множества достижимых целей, поэтому разр

задач, изучавшихся ранее в данной главе.

В данном параграфе будем для определенности предполагать, что в

. Тогда критериальная точка y^* задана $y \leq y^*$. В этом случае ОЭП представляет собой множество Y_p , выражаемое следующим образом:

$$Y_p = \{y \in \mathbf{R}^m : y \leq y^*, \quad y^* \in Y\}. \quad (4.1)$$

Заметим, что множество Y_p можно представить в ином, иногда более простом виде $Y_p = Y + (-\mathbf{R}_+^m)$, где \mathbf{R}_+^m – неотрицательный m -мерный вектор.

а-венств. Рассмотрим системы, изучавшиеся в § 5.1, т.е. пространство $\mathbf{W} \subset \mathbf{R}^n$, множество $X \subset \mathbf{R}^n$, n -линейное отображение $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ является линейным. Будем предполагать, что X

$$X = \{x \in \mathbf{R}^n : Hx \leq h\},$$

H и h – заданные матрица и вектор, а линейное отображение f задано матрицей F , т.е. $f(x) = Fx$.

$Y = f(X)$, которое в данном случае задано

$$f(X) = \{y \in \mathbf{R}^m: y = Fx, Hx \leq h\},$$

$$Q = \{y \in \mathbf{R}^m: Dy \leq d\}, \quad (4.2)$$

в § 5.1 было предложено использовать построения проекции многогранника

$$Z = \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^m: y = Fx, Hx \leq h\}$$

на пространство критериев. Для построения ОЭП в том же явном виде (4.2) также используем построение проекций. Согласно (4.1), ОЭП в дан-

$$Y_p = \{y \in \mathbf{R}^m: y \leq Fx, Hx \leq h\},$$

поэтому для построения его явного представления (4.2) достаточно по-

$$Z = \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^m: y \leq Fx, Hx \leq h\}$$

на пространство критериев. Таким образом, нет принципиальных различий между построением ОЭП на пространстве критериев и его ОЭП. При этом остаются в силе все замечания и недостатки этих методов, приведенные в § 5.1. Этот метод был предложен в [36], но не нашел широкого применения в связи с появлением алгоритмов

оптимизации. Ими являются алгоритмы, которые называются алгоритмами граничными. Они используются для построения ОЭП, как и алгоритмы аппроксимации множества достижимых целей, основанные

на построении выпуклой оболочки множества достижимых целей и некоторой дополнительной точки, расщепления задач оптимизации. При этом предполагается, что множество достижимых

целей выпуклым. В рассматриваемых алгоритмах множества достижимых целей

представлены в виде выпуклой оболочки множества точек этого множества.

Пусть $V = \{v^1, v^2, \dots, v^s\}$, принадлежащих к \mathbf{R}^m . Выпуклая оболочка этой совокупности точек определяется

$$\text{conv} V = \{y \in \mathbf{R}^m: y = \sum_{i=1}^s \lambda_i v^i, \sum_{i=1}^s \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \dots, s\}.$$

ОЭП выпуклой оболочки этого множества точек в случае (4.1) определяется так:

$$(\text{conv}V)_p = \{y \in \mathbf{R}^m: y \leq \sum_{i=1}^s \lambda_i v^i, \sum_{i=1}^s \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \dots, s\}. \quad (4.3)$$

Заметим, что это определение отличается от определения выпуклости. В

степени аналогично приведенному в § 5.2 описанию алгоритма симметрии выпуклого тела.

основании ее определения, т.е. на основании системы равенств (4.3), задающей $(\text{conv}V)_p$ в неявном виде. Для построения ОЭП Z_p н-

$y \lambda$:

$$Z_p = \{(\lambda, y) \in \mathbf{R}^s \times \mathbf{R}^m: y \leq \sum_{i=1}^s \lambda_i v^i, \sum_{i=1}^s \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \dots, s\},$$

$y \in \mathbf{R}^m$. Это можно сделать при помощи λ из системы равенств и неравенств, заданных Z_p .

$$\begin{cases} -\sum_{i=1}^s \lambda_i v_l^i + y_l \leq 0, & l = 1, 2, \dots, m; \\ \lambda_i \geq 0, & i = 1, 2, \dots, s; \\ \sum_{i=1}^s \lambda_i = 1, \end{cases}$$

$v_l^j - l - v^j \in V$. Эта задача аналогична задаче построения $\text{conv}V$, и методы решения этих задач аналогичны. Пер λ исключим, используя последнее равенство. Остальные переменные λ свертывания.

е-
е-
вное пространство. В
ограниченными множествами (в частности, конусами) [52], в результате чего алгоритмы построения этих тел ми.

Здесь мы не будем останавливаться на этом вопросе подробно [62], заметим лишь, что алгоритму построения выпуклой оболочки. Как и при построении оболочки, здесь точки совокупности V находятся последовательно,

т.е. допускается, чтобы очередная точка стала известной после того, как построена ОЭП выпуклой оболочки предыдущих точек. Этим можно достичь следующих целей.

воспользоваться алгоритмами, предназначенными для аппроксимации выпуклых компактных тел, достаточно лишь небольшой модификации: в \mathbf{R}_+^m .

тел многогранниками. Нетрудно заметить, что в ходе работы этих алгоритмов будут формироваться только такие грани, векторы нормалей \mathbf{R}_+^m . В результате будет уточняться описание Y_p , которая соответствует недоминируемой Y .

и фигур.

базовую задачу. Для аппроксимации ОЭП в этом случае используются системы простых фигур, каждая из которых представляет собой шар и конуса $(-\mathbf{R}_+^m)$. Для построения систем таких фигур используются алгоритмы, почти полностью совпадающие с алгоритмами, изложенными в § 5.3. Единственным отличием является то, что к совокупности базовых

управлении, принадлежащем \mathbf{R}_+^m .

§ 5.3, в вычислительных множествах в качестве шаров используются параллелепипеды. Поэтому при аппроксимации шаров используются параллелепипеды. Таким образом, аппроксимация осуществляется системами сдвинутых ортантов.

5.5. Построение двумерных сечений

визуализацию. Как уже говорилось, при этом используется точное или $f(X)$ (или $f(X) + (-\mathbf{R}_+^m)$), построенное в

полупространств, либо в виде объединения конусов. Отметим заранее, что методика визуализации, описанная далее,

решений при нескольких критериях. Вообще говоря, после того, как построены $f(X)$, могут быть использованы и другие средства визуализации.

Как уже говорилось, основой визуализации многомерных многогранников является двумерное сечение. Напомним его определение. Пусть $I^0 = \{1, 2, \dots, m\}$, $I \subset I^0$, $|I| = 2$, $I^* = I^0 \setminus I$. Обозначим $R(I)$ подпростран-

Напомним, что каждое неравенство системы (5.3) определяет п у-пространство (в данном случае – полуплоскость), а коэффициенты (a_{i1} , a_{i2}) являются координатами вектора внешней нормали к этой полуплоскости. Каждая вершина искомого многоугольника является крайней точкой множества решений одной из подсистем системы (5.3), состоящей из двух неравенств. Крайнюю точку множества р
3-
вольных неравенств системы (5.3) б
венств.

Для того чтобы найти вершины многоугольника, неравенств системы (5.3) выделить неравенства, соответств о-нам этого многоугольника, расположив их в порядке обхода вдоль грани-цы. Узловые точки соседних неравенств будут искомыми вершинами многоугольника. Такое расположение нер и-о-тив часовой стрелки. Поскольку пр о-гоугольников, представляется удобным заранее расположить в таком теме (5.3) (отделив при этом неравенства, у ента равны нулю). Этот порядок одинаков для всей лельных сечений, поэтому такое упорядочение для всей серии няется только один раз и выступает в качестве предобработки. Для ¹⁰, имеющие порядок вр н-
 $N \ln N$.

следует отобрать неравенства, существенные для описания данного сече-ния, т.е. соответствующие сторонам искомого мног ¹¹.

упорядоченной системы (5.3). Возможность последовательной обработки основана на следующем очевидном факте: если некот

системы, то они будут зависимыми и для всей совокупности неравенств (5.3). В процессе обработки очередного неравенства происходит модификация набора неравенств, бывших существенными для подмножества неравенств, рассмотренного р нее. Возможно, часть из них станет зави-венства.

¹⁰ См., например, [22].

¹¹ Точнее говоря, необходимо исключить зависимые неравенства (см. § 5.1) из системы (5.3) при заданных x_j^* , j, I^* .

Рассмотрим пример (см. рис. 5.1). Здесь после обработки некоторой совокупности неравенств построена ломаная с узлами A, B, C, D (порядок выстроен стрелками).

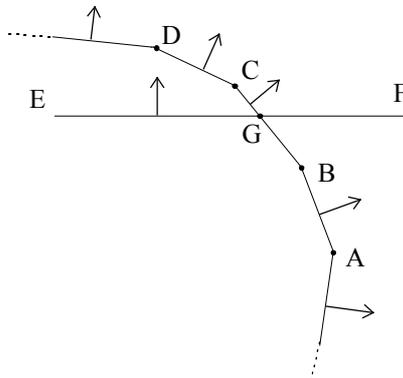


Рис. 5.1

конечных лучей, выходящих из точек A и D. Внешние нормали на рисунке.

неравенство задает прямую EF. Видно, что узлы D и C не являются вершинами ломаной “выше” линии EF устраняются. В ломаную добавляется новый узел G (образуется звено BG) и луч GE, выходящий из этой

точки.

Обратим внимание, что упорядоченная система неравенств (5.3) является циклической, т.е. за последним неравенством следует первое. После обработки всех неравенств (если сечение не пусто), для того чтобы построить ломаную и получить многоугольник, следует продолжить работу алгоритма, обрабатывая отобранные неравенства во втором разе, пока не будет вычислена точка.

Построение двумерных сечений объединения параллелепипедов.

помощью наложения сечений отдельных шаров, либо с построения контура объединения совокупности их сечений. Как уже говорилось, в качестве шаров в книге используются параллелепипеды. В этом случае используются параллельные оси координат. Алгоритмы предложены, например, в [51] и имеют временную сложность, $N \ln N$, где N – число прямоугольников, описывающих визуализируемое множество. Недостатком указанных алгоритмов является сложность их реализации. Результаты численных экспериментов.

сложностью, пропорциональной N^2 . Алгоритм основан на методе сканирования: контур строится по мере продвижения заметной прямой, распо-

¹² Подробнее см. [68], где также показано, что благодаря предварительному упорядочению системы (5.3) число операций, осуществляемых при построении многоугольника, пропорционально N .

ложенной вдоль одной из осей координат. Поя

о-

¹³

Пусть имеется набор прямоугольников, изображенный на рис. 5.2. На W – заметающая прямая в ее начальном положении (до -либо событий). Направление заметания – вверх, т.е. в Y . Следующий рис. 5.3 иллюстрирует организацию информации о прямоугольниках, отслеживаемую в процессе перемещения заметающей прямой.

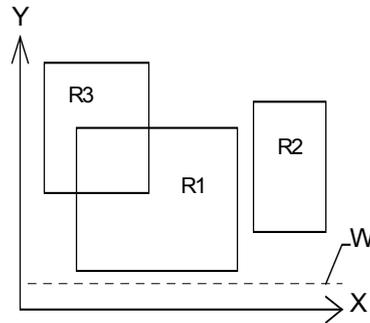


Рис. 5.2

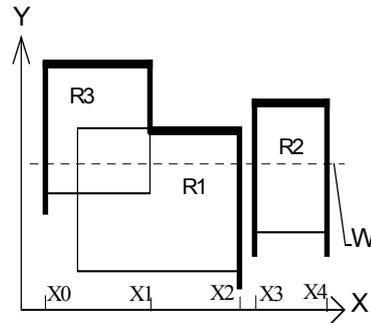


Рис. 5.3

Поддерживается структура, описывающая верхнюю и нижнюю границы контура объединения прямоугольников, пересекающих линию заметания (на рисунке верхняя граница показана жирными линиями). Она хранится в виде упорядоченного набора интервалов оси абсцисс: $(-\infty, X_0)$, (X_0, X_1) , (X_1, X_2) , (X_2, X_3) , (X_3, X_4) , $(X_4, +\infty)$; для каждого интервала хранится значение ординаты Y . Для интервалов, лежащих вне перекрываемых (на рис. 5.3 – интервалы $(-\infty, X_0)$, (X_2, X_3) , $(X_4, +\infty)$), принимаем $Y = -\infty$.

ях:

- Y – ордината (событие “вхождение нового прямоугольника в сечение”);
- Y – ордината Y – либо интервала (событие “завершение интервала”).

Интересно, что для описываемого алгоритма вычислительные экспоненциальные

от числа прямоугольников (рассматривались множества, описанные десятками тысяч прямоугольников).

¹³ Подробнее см. [20].

Подход, описанный в этом разделе, дает возможность производить операции с изображениями, что не удается в случае простого наложения сечений шаров.

Построение двумерных сечений ОЭП.

сдвинутых органтов. Сечение такой системы представляет собой о-онную убывающую или возрастающую ломаную, состоящую из звеньев, параллельных осям координат. Это свойство позволяет использовать простой алгоритм, аналогичный алгоритму построения вершин выпуклого многоугольника, предназначенный для визуализации большого числа сечений ОЭП. В этом алгоритме используется предобработка в виде упорядоченного списка о-онных чисел органтов в системе.

5.6. О построении множеств достижимых целей

Во введении к данной главе указывалось, что МДЦ применим для анализа моделей достаточно произвольной природы. Математически говоря, это выражалось в том, что мы не указывали конкретное пространство W рассматриваемых решений (стратегий). На практике, конечно, приходится

определить W , множество X $f: W \rightarrow R^m$, для которых удается разработать алгоритмы $f(X)$. Так, были рассмотрены

три базовые задачи, методы аппроксимации $f(X)$ в §§ 5.1-5.3. В данном параграфе показывается, что к базовым задачам е-ских моделей, описываемых дифференциальными включениями (или системами управляемых дифференциальных уравнений).

Основные понятия.

¹⁴. Рассматриваются системы типа

$$\dot{x} = g(x, u, t), \quad (5.1)$$

t – время, x – n -мерный вектор фазовых координат, u – r -мерный вектор управлений, $g(x, u, t)$ – n -мерная функция, $t \in [0, T]$ и всех $u \in R^r, x \in R^n$. На вектор управлений и $t \in [0, T]$ наложено

$$(u(t), x(t)) \in Y(t), \quad (5.2)$$

¹⁴ См., например, [26, 49, 57].

$Y(t) \subset \mathbf{R}^r \times \mathbf{R}^n$ – заданное семейство множеств, зависящих от времени. Кроме того, задано множество $\Gamma(0) \subset \mathbf{R}^n$

$$x(0) \in \Gamma(0). \quad (5.3)$$

Пусть определены классы, к которым принадлежат векторы $u(\cdot), x(\cdot)$. Например, $u(\cdot) \in L_\infty^r[0, T]$, где $L_\infty([0, T])$ – пространство измеримых, почти всюду ограниченных на $[0, T]$ функций с нормой L_∞ , а $x(\cdot) \in L_\infty^n([0, T])$, где $L_\infty([0, T])$ – пространство абсолютно непрерывных функций на $[0, T]$ с нормой L_∞ . Пусть $W = L_\infty^r[0, T] \times L_\infty^n[0, T]$ и пусть задано отображение $f: L_\infty^r[0, T] \times L_\infty^n[0, T] \rightarrow \mathbf{R}^m$,

ставящее в соответствие паре $(u(\cdot), x(\cdot))$ некоторую совокупность значений критериев. Обозначим через X совокупность пар $(u(\cdot), x(\cdot)) \in L_\infty^r[0, T] \times L_\infty^n[0, T]$, удовлетворяющих (5.1)-(5.3). Тогда множество $f(X)$.

Существует альтернативное, зачастую более удобное описание динамических систем. Пусть

$$\dot{x} \in G(x, t), \quad (5.4)$$

$G(x, t) \subset \mathbf{R}^n$ – семейство множеств, заданное при $t \in [0, T]$ и всех $x \in \mathbf{R}^n$. Ясно, что система (5.1), (5.2) приводится к виду (5.4). С другой стороны, положив $v \equiv \dot{x}$

$$\dot{x} = v, \quad v \in G(x, t),$$

переформулируем (5.1), (5.2). Таким образом, для системы (5.4), (5.3) мы имеем (5.1)–(5.3). Заметим, что динамическую систему удобно задавать в виде (5.4) вместо (5.1), (5.2) в том случае, когда $G(x, t)$ представляет собой множество векторов v , например, если отображение $f(u(\cdot), x(\cdot))$ задано в виде $f(x(\cdot))$.

Обозначим $f(x(\cdot))$: критерии – $x(T)$, т.е. векторы $x(T)$. Множество достижимости S системы (5.1)-(5.3) (или (5.4),(5.3)) и обозначать $\Gamma(T)$. Это множество дисплея.

– это случай терминальных критериев, т.е. критериев, заданных в $x(T)$:

$$f(x(\cdot)) = \tilde{f}(x(T)).$$

Возможно также рассмотреть и более сложные критерии, определяемые на траекториях, например, зависящие от состояния системы $x(t)$ и времени t .

траекторий. Все они могут быть сведены к виду $f(x(\cdot)) = \tilde{f}(x(T))$. Подчеркнем, что зависимость только от траекторий – хотя и важный, но только частный случай критериев.

Далее будут рассмотрены лишь терминальные критерии. В этом случае $\Gamma(T)$.

с-
тем. Рассмотрим дифференциальное включение (5.4). Заметим, что если $G(x, t)$ в (5.4) на его выпуклую оболочку, т.е. $\text{conv } G(x, t)$, то при выполнении некоторых простых условий (см. [1]) множество достижимости не изменится. При каждом $t \in [0, T]$ рассмотрим

$Z(t) \subset \mathbf{R}^n \times \mathbf{R}^n$ – график многозначного отображения, действующего $\mathbf{R}^n \rightarrow \mathbf{R}^n$ и заданного множеством $\text{conv } G(x, t)$, т.е.

$$Z(t) = \{ (v, x) \in \mathbf{R}^n \times \mathbf{R}^n : v \in G(x, t), x \in \mathbf{R}^n \}.$$

Тогда (5.4) можно записать в эквивалентном виде

$$\dot{x} = v, \quad (v, x) \in Z(t). \tag{5.5}$$

Систему (5.1), (5.2) или (5.4) будем называть выпуклой, если она сводится к виду (5.5) с выпуклыми $Z(t), t \in [0, T]$. Зам. тим, что частным случаем выпуклых систем являются системы (5.1), (5.2) с линейным дифференциальным уравнением, т.е.

$$\dot{x} = A(t)x + B(t)u + a(t),$$

$A(t), B(t)$ и $a(t)$ – матрицы и вектор, заданные при $t \in [0, T]$, и с выпуклыми $Y(t), t \in [0, T]$. Кроме того, к выпуклым относятся и

$\Gamma(T)$ для в к-
 $\Gamma(0)$. Очевидно, что для них множество $\Gamma(T)$
выпукло. Поэтому проблему аппроксимации
о-
16,
гранниками.

р-
тивания. Разобьем отрезок $[0, T]$ на N $t_i = i\tau$, где
 $\tau = T/N, i = 0, 1, \dots, N$. Вместо системы (5.5) рассмотрим, напр мер,

¹⁵ Примеры приведены в [23]. В частности, при выполнении некоторых простых условий к ним относятся системы (5.1), (5.2) с дифференциальным уравнением

$$\dot{x} = \varphi(x)\psi(u) \quad \varphi(x).$$

¹⁶ изучено в работах [29, 31, 33].

$$\frac{x^{i+1} - x^i}{\tau} = v^i, \quad (v^i, x^i) \in Z^\Delta(t_i), \quad x^0 \in \Gamma^\Delta(0), \quad (5.6)$$

$i = 0, 1, \dots, N-1$, $Z^\Delta(t_i)$ – многогранное множество, аппроксимирующее $Z(t_i)$ с точностью Δ , $\Gamma^\Delta(0)$ – многогранное множество, аппроксимирующее $\Gamma(0)$ с точностью Δ . Заметим, что, вообще говоря, здесь не $Z(t)$ и $\Gamma(0)$, поэтому мы требуем заранее, чтобы эти множества можно было аппроксимировать многогранными множествами, не обязательно ограниченными, с любой степенью точности.

17 $\Gamma(T)$ пред-
 Γ_N , представляющее собой множество x^N для системы (5.6).

Γ_N –
сведение к первой базовой задаче. Если многогранные множества $\Gamma^\Delta(0)$ и $Z(t_i)$ заданы как множества решений системы линейных неравенств, и $y = x^N$, мы приходим к первой базовой задаче, в которой

$$(x^0, x^1, \dots, x^N, v^0, v^1, \dots, v^{N-1}) \in \mathbf{R}^{(N+1)n} \times \mathbf{R}^{Nr},$$

удовлетворяющие соотношениям (5.6), а критерии – $y \in \mathbf{R}^n$.
§ 5.1 указывалось, что, вообще говоря, методы свертывания позволяют

большого числа переменных.

В данном случае число переменных равно $(N + 1)n + Nr$, т.е. является довольно большим. Тем не менее, множество Γ_N построить за счет специфики системы (5.6), представляющей собой разностную аппроксимацию дифференциального включения (5.4).

Γ_N . В этом методе вводится понятие множества достижимости на отдельном шаге. Например, Γ_1 – x^1 , удовлетворяющих (5.6) для $i = 0$, т.е. векторов x^1 , для к x^0, v^0 , что

$$\frac{x^1 - x^0}{\tau} = v^0, \quad (v^0, x^0) \in Z^\Delta(0), \quad x^0 \in \Gamma^\Delta(0).$$

В свою очередь, Γ_2 – x^2 , для которых на x^0, x^1, v^0, v^1 , что

$$\frac{x^2 - x^1}{\tau} = v^1, \quad \frac{x^1 - x^0}{\tau} = v^0, \\ (v^1, x^1) \in Z^\Delta(t_1), \quad (v^0, x^0) \in Z^\Delta(t_0), \quad x^0 \in \Gamma^\Delta(0).$$

$\Gamma_3, \Gamma_4, \dots, \Gamma_i, \dots, \Gamma_{N-1}$ определяются аналогичным образом.

Заметим, что при определении множества Γ_2 решать задачу с самого начала, а определить его как множество x^2 , для x^1, v^1 , что

$$\frac{x^2 - x^1}{\tau} = v^1, \quad (v^1, x^1) \in Z^\Delta(t_1), \quad x^1 \in \Gamma_1.$$

Аналогичным образом, множество достижимых состояний на $(i+1)$ -м шаге Γ_{i+1} для которых найду x^i, v^i , что

$$\frac{x^{i+1} - x^i}{\tau} = v^i, \quad (v^i, x^i) \in Z^\Delta(t_i), \quad x^i \in \Gamma_i.$$

Поэтому, если множество Γ_i системы линейных неравенств, то нахождение множества Γ_{i+1} – первая базовая задача. Таким образом, множество Γ_{i+1} найдено с помощью методов свертывания. Поскольку исходное множество $\Gamma(0)$ аппроксимировано многогранным множеством $\Gamma^\Delta(0)$, заданным в виде множества решений системы линейных неравенств, то о шагам, последовательно находя $\Gamma_1, \Gamma_2, \dots, \Gamma_N$. При этом на каждом шаге с помощью свертывания требуется r вектор x^i (вектор x^i помощью равенств). Число таких шагов равно N .

е-имуществами:

- 1) попутно строятся промежуточные множества, которые могут представлять самостоятельный интерес;
- 2) - описание промежуточных множеств, что может позволить построить Γ_N для более широкого класса систем, чем в прямом тоде.

е-тод крупных шагов, в котором заранее выбираются нужное число номеров i , для которых будут построены множества Γ_i . Он обладает всеми преимуществами пошагового метода, но в то же время не имеет его - $\Gamma_i, i = 1, 2, \dots, N$, число

которых может оказаться слишком большим. Различные методы аппроксимированы в диалоговой системе (см. [9]).

п-проксимации в.к.т. многогранниками. В том случае, когда методы Γ_N большой сложности задачи, можно использовать методы аппроксимации, § 5.2. При этом требуется, чтобы множество $\Gamma(T)$ было ограниченным.

Γ_N для линейной системы (5.6). Эта задача § 5.2.

Другой подход состоит в использовании исходной, дифференциальной формы описания динамической системы. Дело в том, что аппроксимации в.к.т. многогранниками требуется решать вспомогательные задачи оптимизации, а для некоторых из динамических систем удается сделать в исходном виде, обычно с помощью принципа максимума Понтрягина [49]. В частности, задачи оптимизации можно решить для систем (5.1)-(5.3) с линейным дифференциальным уравнением, не зависящими от фазового состояния.

Заметим, что метод аппроксимации позволяет также строить промежуточные (или конечные) динамические системы.

динамических систем. $\Gamma(T)$ аппроксимированы Γ^A , где $\Gamma^A - \Gamma_N$ $f(\Gamma(T))$ достижимых значений критериев.

$f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ - линейное, то эта задача решается с § 5.2 (при этом Γ^A должно быть ограничено). Выбор метода зависит от Γ^A m .

Заметим, что для линейного отображения f могла бы быть сформулирована как построение (или аппроксимация)

для системы (5.6) без промежуточного построения множества Γ_N . Такой разумным, если, например, множество $\Gamma(T)$ аппроксимировано его разностным аналогом.

Рассмотрим теперь терминальные критерии, задаваемые нелинейными динамическими системами. В отличие от случая линейных критериев, здесь предва-
 $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$. Будем по-
 $\Gamma(T)$ много
 Γ^A
 ходимой.

§ 5.3, методы аппроксимации невыпуклых

размерности пространства переменных. В
 данном случае, при предварительной аппроксимации множества $\Gamma(T)$
 Γ^A , задача оказывается сведена к проблеме
 $f(\Gamma^A)$, где $\Gamma^A \subset \mathbf{R}^n$. Поскольку размерность пространства
 состояний обычно не слишком велика, методы
 нимыми.

динамических систем.

системы не удовлетворяют условию выпуклости, то описанные выше
 методы можно использовать для построения некоторого множества, со-
 $\Gamma(T)$. Это достигается¹⁸
 $Z(t)$, после чего можно построить множе-
 $f(\Gamma(T))$.

Ясно, что для аппроксимации множества достижимых целей в случае

использовать покрытие системами шаров. При этом пространство $L_\infty^r[0,$
 $T]$
 $u(\cdot)$
 числом параметров, например, считать, что

$$u(t) = \sum_{s=1}^S \alpha_s u^{(s)}(t),$$

α_s – неопределенные параметры, а $u^{(s)}(t)$ –
 s
 $= 1, 2, \dots, S$. Если это отвечает смыслу задачи, то множество $f(\Gamma(T))$ мо-
 жет быть аппроксимировано достаточно точно. Для этого можно исполь-
 зовать методику, описанную в § 4.3, без каких-либо дополнений. В част-
 ности, в экономических задачах, рассмотренных в §§ 3.4 и 3.5, именно
 ческих систем.

Замечание.

множеств достижимых целей для динамических систем, описываемых
 дифференциальными уравнениями (5.1) и дифференциальными включе-
 ниями (5.4). Ясно, что рассмотренные методы применимы и для разност-
 ных (многошаговых) уравнений, в которых состояние меняется от шага к
 шагу.

Заключение. Методы, рассмотренные в данном параграфе, по-
 яют строить множества достижимых целей для динамических систем, опи-

¹⁸ Подробнее этот вопрос рассмотрен в [23].

ениями. Это лишь один из примеров применения МДЦ в случае бесконеч-
ности. Могут быть рассмотрены и более общие системы, например, включающие уравнения в частных производ-

19

19

смотрены в [35].

с-