АКАДЕМИЯ НАУК СССР
УРАЛЬСКИЙ НАУЧНЫЙ ЦЕНТР
ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ

АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ ИГР

(Материалы по математическому обеспечению ЭВМ)

Свердловск 1984

АКАДЕМИЯ НАУК СССР
УРАЛЬСКИЙ НАУЧНЫЙ ЦЕНТР
ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ

АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ ИГР

(материалы по математическому обеспечению ЭВМ)

Свердловск, 1984

УДК 519.9

Алгоритмы и программы решения линейных дифференциальных игр (материалы по математическому обеспечению ЭВМ). Свердловск: УНЦ АН СССР, 1984.

Брошюра содержит набор алгоритмов и программ, предназначенных для решения некоторых типов линейных дифференциальных игр.

Материал рассчитан на вычислителей, инженеров и научных работников, интересующихся численными методами теории управления и теории дифференциальных игр.

Ответственные редакторы -
доктор физ.-мат.наук А.И.Субботин
кандидат физ.-мат.наук В.С.Пацко

$А\frac{20204-214(83)}{055(02)\ 7}БО$

Е.А.Исакова, Г.В.Логунова, В.С.Пацко

ПОСТРОЕНИЕ СТАБИЛЬНЫХ МОСТОВ В ЛИНЕЙНОЙ ДИФФЕРЕНЦИАЛЬНОЙ ИГРЕ С ФИКСИРОВАННЫМ МОМЕНТОМ ОКОНЧАНИЯ

I. Постановка задачи

Рассматривается линейная позиционная игра

$$\dot{x} = Ax + Bu + Cv \tag{I}$$

с фиксированным моментом окончания ϑ. Здесь $x \in R^n$ – фазовый вектор, A, B, C – постоянные матрицы размерностей $n \times n$, $n \times k$, $n \times e$ соответственно. Управляющий параметр u первого игрока в каждый момент $t \leq \vartheta$ выбирается из множества $\mu(t)P$, управляющий параметр v второго игрока – из множества $\nu(t)Q$. Множества P, Q – многогранники в пространствах R^k, R^e, скалярные функции $\mu(\cdot)$, $\nu(\cdot)$ линейные: $\mu(t) = a + bt$, $\nu(t) = c + dt$.

Выделим две координаты x_α, x_β фазового вектора x. Будем считать, что цель первого игрока – привести в момент ϑ выделенные координаты на заданное множество $M \subset R^2$, т.е. обеспечить включение

$$\begin{pmatrix} x_\alpha(\vartheta) \\ x_\beta(\vartheta) \end{pmatrix} \in M \ . \qquad (2)$$

Интересы второго игрока противоположны. Условимся, что M - выпуклый, замкнутый, ограниченный многоугольник. Требуется найти множество W всех начальных позиций (t_*, x_*), для каждой из которых первый игрок, используя позиционный способ управления [2], может гарантировать в момент ϑ включение (2). Следуя [2], будем называть W множеством позиционного поглощения или максимальным стабильным мостом.

Описанная дифференциальная игра n -го порядка эквивалентна [2,5] дифференциальной игре 2-го порядка

$$\dot{z} = X_{\alpha,\beta}(\vartheta,t)Bu + X_{\alpha,\beta}(\vartheta,t)Cv \ , \qquad (3)$$

где $X_{\alpha,\beta}(\vartheta,t)$ - матрица, составленная из α и β -ой строк фундаментальной матрицы Коши $X(\vartheta,t)$, соответствующей матрице A. Ограничения на управляющие параметры u, v в игре (3) такие же, как в игре (I). Момент окончания равен ϑ, целевое множество совпадает с M.

Поскольку матрица A постоянна, то матрица $X(\vartheta,t)$ зависит лишь от разности $\vartheta - t$. Введем обратное время $\tau = \vartheta - t$. Тогда система (3) и ограничения на управляющие параметры игроков перепишутся в виде

$$\dot{y} = -Y(\tau)Bu - Y(\tau)Cv \ , \qquad (4)$$

$$Y(\tau) = X_{\alpha,\beta}(\vartheta, t)\ , \ u \in \mu(\vartheta-\tau)P, \ v \in \nu(\vartheta-\tau)Q \ .$$

Пусть $\tilde{W}$ – максимальный стабильный мост для игры (4), т.е. множество всех начальных позиций (τ_*, y_*), для каждой из которых разрешима задача первого игрока о приведении фазового нектора в момент $\tau = 0$ на множество M. Положим

$$W_t = \{ x \in R^n : (t, x) \in W \}, \ \ t \leq \vartheta ,$$

$$\tilde{W}_\tau = \{ y \in R^2 : \ (\tau, y) \in \tilde{W} \}, \ \ \tau \geq 0 \ .$$

Справедливо соотношение [2,5]

$$W_t = \{ x \in R^n : \ Y(\vartheta - t)x \in \tilde{W}_{\vartheta - t} \} \ .$$

Описываемая ниже программа предназначена для построения множеств $\tilde{W}_\tau$. В качестве входной информации задаются матрицы A, B, C, вершины многогранников P, Q, вершины многоугольника M, числа a, b, c, α, определяющие функции $\mu(\cdot)$, $\nu(\cdot)$, шаг Δ попятной процедуры.

2. Описание основного алгоритма

Пусть $\tau_0 = 0$, $\tau_{i+1} = \tau_i + \Delta$, $i = 0, 1, 2, \ldots,$ — разбиение полуоси $\tau \geq 0$ с шагом $\Delta > 0$. Положим

$$Y^*(\tau) = Y(\tau_i),\ \mu^*(\tau) = \mu(\vartheta - \tau_i),\ \nu^*(\tau) = \nu(\vartheta - \tau_i),$$

$$\tau \in [\tau_i, \tau_{i+1}), \quad i = 0, 1, 2, \ldots .$$

Заменим игру (4) аппроксимирующей дифференциальной игрой

$$\dot{y} = -Y^*(\tau) B u - Y^*(\tau) C v, \qquad (5)$$
$$u \in \mu^*(\tau) P, \quad v \in \nu^*(\tau) Q$$

с фиксированным моментом окончания $\tau = 0$ и целевым множеством M. Пусть W^* — максимальный стабильный мост для игры (5) и

$$W^*_\tau = \{ y \in R^2 : (\tau, y) \in W^* \}, \ \tau \geq 0.$$

При любом $\tau \geq 0$ множество W^*_τ — выпуклый, замкнутый, ограниченный многоугольник [2,4]. Оценки близости W^*_τ к $\tilde{W}_\tau$ в метрике Хаусдорфа имеются в работах [1, 3]. Положим

$$P_i^* = \mu^*(\tau_i) Y^*(\tau_i) BP, \; Q_i^* = \nu^*(\tau_i) Y^*(\tau_i) CQ \; ,$$

$$W_i^* = W_{\tau_i}^* \; , \quad i = 0, 1, 2, \ldots \; .$$

Заметим, что $W_0^* = M$.

Опорная функция γ_{i+1}^* множества W_{i+1}^* , $i = 0, 1, 2, \ldots$, есть [4] выпуклая оболочка функции

$$\gamma_{i+1}(\ell) = \max_{m \in W_i^*} \langle \ell, m \rangle + \max_{p \in -\Delta P_i^*} \langle \ell, p \rangle - \max_{q \in \Delta Q_i^*} \langle \ell, q \rangle, \; \ell \in R^2 .$$

Поскольку W_i^* , P_i^* , Q_i^* – выпуклые многоугольники (возможно вырождение в точку или отрезок), то положительно-однородная функция γ_{i+1} – кусочно-линейная, и ее выпуклая оболочка может быть найдена точно после конечного числа итераций. При нахождении выпуклой оболочки используем также то свойство, что "локальная выпуклость" функции γ_{i+1} может нарушаться лишь вблизи направлений, определяемых внешними нормалями к многоугольнику Q_i^* Зная функцию γ_{i+1}^* , легко определить многоугольник W_{i+1}^* .

Пусть $\mathcal{M}$, $\mathcal{P}$, $\mathcal{Q}$ – выпуклые, замкнутые многоугольники (допускается вырождение в точку или отрезок). Опишем способ построения выпуклой оболочки φ^* функции φ , заданной на плоскости формулой

$$\varphi(\ell) = \max_{m \in \mathcal{M}} \langle \ell, m \rangle + \max_{p \in \mathcal{P}} \langle \ell, p \rangle - \max_{q \in \mathcal{Q}} \langle \ell, q \rangle . \qquad (6)$$

Этот способ используется в программе для построения выпуклой оболочки γ^*_{i+1} функции γ_{i+1}.

Введем вспомогательные понятия и обозначения. Для произвольного набора K единичных векторов на плоскости пусть $h(K)$ - количество векторов в наборе. Говоря о нумерации векторов в наборе, условимся вводить ее всегда против часовой стрелки. Пусть ℓ_r - вектор из K с номером r. Положим $\ell_{h(K)+1} = \ell_1$, $\ell_0 = \ell_{h(K)}$.

Скажем, что набор K обладает свойством F, если $h(K) \geq 3$ и при любом $r = 1, 2, \ldots, h(K)$ угол между векторами ℓ_r, ℓ_{r+1}, отсчитываемый против часовой стрелки от ℓ_r к ℓ_{r+1}, меньше π.

Для любых единичных векторов ℓ_*, ℓ^* $(\ell_* \neq -\ell^*)$ на плоскости символом $\Lambda(\ell_*, \ell^*)$ обозначим наименьший выпуклый, замкнутый конус с вершиной в нуле, содержащий ℓ_* и ℓ^*.

Пусть $N(\mathcal{M})$, $N(\mathcal{P})$, $N(\mathcal{Q})$ - совокупности единичных внешних нормалей к многоугольникам $\mathcal{M}$, $\mathcal{P}$, $\mathcal{Q}$. Положим $L = N(\mathcal{M}) \cup N(\mathcal{P}) \cup N(\mathcal{Q})$. Для дальнейшего нам нужно, чтобы набор L обладал свойством F. Удобно также, чтобы каждое из множеств $N(\mathcal{M})$, $N(\mathcal{P})$, $N(\mathcal{Q})$ содержало не менее двух векторов, и в случае двух векторов они были бы противоположно направлены. Эти условия выполнены, если многоугольники $\mathcal{M}$, $\mathcal{P}$, Q невырожденные. При вырождении какого-либо из трех многоугольников в точку или отрезок условимся сопоставлять ему в качестве множества нормалей совокупность из двух или более единичных векторов, причем так, чтобы набор L обладал свойством F . Когда число сопоставляемых векторов равно двум, будем брать их противоположно направленными. Если многоугольник вырождается в отрезок, потребуем, чтобы среди сопоставляемых векторов были векторы, ортогональные отрезку.

Из формулы (6) следует, что функция φ непрерывна, и при любом $1 \leq r \leq h(L)$ ее сужение на конус $\Lambda(\ell_r, \ell_{r+1})$ есть линейная функция. Кроме того, выпуклым является сужение φ на любой выпуклый конус с вершиной в нуле, не содержащий в своей внутренности векторов из $N(\mathcal{Q})$.

Для нахождения выпуклой оболочки φ^* функции φ построим конечную последовательность заданных на плоскости числовых функций $\varphi^{(k)}$, $k = 0,1,2,\ldots$. Примем $\varphi^{(0)} = \varphi$. Наряду с функциями $\varphi^{(k)}$ будем строить наборы векторов $L^{(k)} \subset L$ и $E^{(k)} \subset L^{(k)}$. Положим $L^{(0)} = L$, $E^{(0)} = N(\mathcal{Q})$.

Пусть при некотором $s \geq 0$ построены функция $\varphi^{(s)}$, упорядоченные против часовой стрелки наборы $L^{(s)}$, $E^{(s)}$ и выполнены условия:

а) набор $L^{(s)}$ обладает свойством F ,

в) функция $\varphi^{(s)}$ непрерывна, и при любом $1 \leq r \leq h(L^{(s)})$ ее сужение на конус $\Lambda(\ell_r^{(s)}, \ell_{r+1}^{(s)})$ ($\ell_j^{(s)}$ - вектор из $L^{(s)}$ с номером j) есть линейная функция,

с) сужение $\varphi^{(s)}$ на любой выпуклый конус с вершиной в нуле, не содержащий в своей внутренности векторов из $E^{(s)}$, есть выпуклая функция.

Перечисленным условиям, как отмечено выше, удовлетворяют функция $\varphi^{(0)} = \varphi$ и наборы $L^{(0)} = L$, $E^{(0)} = N(Q)$. Опишем правило перехода от $\varphi^{(s)}$, $L^{(s)}$, $E^{(s)}$ к $\varphi^{(s+1)}$, $L^{(s+1)}$, $E^{(s+1)}$.

Скажем, что функция $\varphi^{(s)}$ выпукла вблизи некоторого единичного вектора $y \in R^2$, если существует такой выпуклый конус S с вершиной в нуле, содержащий y в своей внутренности, что сужение $\varphi^{(s)}$ на S выпукло.

Перебирая все векторы из $E^{(s)}$, проверим выпуклость $\varphi^{(s)}$ вблизи каждого из них. Пусть e - произвольный вектор из $E^{(s)}$. Поскольку $E^{(s)} \subset L^{(s)}$, то вектор совпадает при некотором $1 \leq r \leq h(L^{(s)})$ с вектором $\ell_r^{(s)}$ из $L^{(s)}$. Для проверки выпуклости $\varphi^{(s)}$ вблизи вектора $e = \ell^{(s)}$ рассмотрим соседние к $\ell_r^{(s)}$ в $L^{(s)}$ векторы $\ell_{r-1}^{(s)}$, $\ell_{r+1}^{(s)}$ и найдем точку y_* пересечения прямых

$$\langle \ell_{r-1}^{(s)}, x \rangle = \varphi^{(s)}(\ell_{r-1}^{(s)}), \quad \langle \ell_r^{(s)}, x \rangle = \varphi^{(s)}(\ell_r^{(s)}) .$$

Если $\langle \ell_{r+1}^{(s)}, y_* \rangle \leqslant \varphi^{(s)}(\ell_{r+1}^{(s)})$, то функция $\varphi^{(s)}$ выпукла вблизи вектора e. Если $\langle \ell_{r+1}^{(s)}, y_* \rangle > \varphi^{(s)}(\ell_{r+1}^{(s)})$, она не является выпуклой. Это утверждение следует из непрерывности $\varphi^{(s)}$ и линейности ее на конусах $\Lambda(\ell_r^{(s)}, \ell_{r+1}^{(s)})$, $\Lambda(\ell_{r-1}^{(s)}, \ell_r^{(s)})$.

Пусть $D^{(s)} \subset E^{(s)}$ – совокупность всех векторов из $E^{(s)}$, вблизи которых $\varphi^{(s)}$ не является выпуклой. Если $D^{(s)} = \emptyset$, то функцию $\varphi^{(s+1)}$ и наборы $L^{(s+1)}$, $E^{(s+1)}$ не строим. Если $D^{(s)} \neq \emptyset$, рассмотрим набор $L^{(s)} \setminus D^{(s)}$. В случае, когда он не обладает свойством F, тройку $\varphi^{(s+1)}$, $L^{(s+1)}$, $E^{(s+1)}$ также не строим. Пусть набор $L^{(s)} \setminus D^{(s)}$ обладает свойством F. Положим $L^{(s+1)} = L^{(s)} \setminus D^{(s)}$. В качестве $E^{(s+1)}$ возьмем совокупность всех векторов из $L^{(s+1)}$, каждый из которых является соседним в $L^{(s)}$ к какому-либо вектору из $D^{(s)}$. Функцию $\varphi^{(s+1)}$ на плоскости определим при помощи условий: 1) $\varphi^{(s+1)}(\ell) = \varphi^{(s)}(\ell)$ для $\ell \in L^{(s+1)}$, 2) сужение $\varphi^{(s+1)}$ на любой конус $\Lambda(\ell_r^{(s+1)}, \ell_{r+1}^{(s+1)})$, $1 \leq r \leq h(L^{(s+1)})$, есть линейная функция. Нетрудно видеть, что для тройки $\varphi^{(s+1)}$, $L^{(s+1)}$, $E^{(s+1)}$ выполнены свойства, аналогичные свойствам а), в), с), оговоренным для тройки $\varphi^{(s)}$, $L^{(s)}$, $E^{(s)}$.

Применяя описанное правило перехода от $\varphi^{(s)}$, $L^{(s)}$, $E^{(s)}$ к $\varphi^{(s+1)}$, $L^{(s+1)}$, $E^{(s+1)}$, построим последовательно тройки $\varphi^{(k)}$, $L^{(k)}$, $E^{(k)}$, $k = 0, 1, 2, \ldots$. Процесс построения конечен. А именно, он заканчивается либо на некотором номере j, для которого $D^{(j)} = \emptyset$, либо на номере j, для которого $L^{(j)} \setminus D^{(j)}$ не обладает свойством F. В первом случае функция $\varphi^{(j)}$ выпукла и совпадает с выпуклой оболочкой φ^* функции φ. Во втором функция φ не имеет выпуклой оболочки. Последнее равносильно тому, что пересечение

$$\bigcap_{l \in L} \{y : \langle l, y\rangle \leqslant \varphi(l)\}$$

совпадающее при каждом $k = 0,1,2, \dots, j$ с пересечением

$$\bigcap_{l \in L^{(k)}} \{y : \langle l, y\rangle \leqslant \varphi^{(k)}(l)\},$$

является пустым.

Таков алгоритм овыпукления функции φ. Подчеркнем, что на каждом шаге k итерационного процесса овыпукления информация о кусочно-линейной функции $\varphi^{(k)}$ состоит из упорядоченного против часовой стрелки набора векторов $L^{(k)}$, определяющего разбиение плоскости на конусы линейности, и набора значений функции $\varphi^{(k)}$ на векторах этого набора. Указывается также набор $E^{(k)} \subset L^{(k)}$ "подозрительных" векторов, где может нарушаться "локальная выпуклость" функции $\varphi^{(k)}$. Переход к следующей функции $\varphi^{(k+1)}$ осуществляется за счет подправки функции $\varphi^{(k)}$ вблизи тех векторов набора $E^{(k)}$, где $\varphi^{(k)}$ не является выпуклой.

Для применения описанного алгоритма овыпукления при нахождении выпуклой оболочки γ^*_{i+1} функции γ_{i+1} следует положить

$$\varphi = \gamma_{i+1}, \; \varphi^* = \gamma^*_{i+1}, \; \mathcal{M} = W_i^*, \;\; \mathcal{P} = -\Delta P_i^*, \;\; \mathcal{Q} = \Delta Q_i^* \; .$$

Если при этом процесс построения $\varphi^* = \gamma^*_{i+1}$ заканчивается на номере j и $D^{(j)} = \varnothing$ (т.е. $\varphi^* = \varphi^{(j)}$), то вершины многоугольника W_{i+1}^{*} получаются как точки последовательного пересечения прямых $\langle \ell_r^{(\varphi)}, y \rangle = \varphi^{(\varphi)}(\ell_r^{(\varphi)})$, $1 \leq r \leq h(L^{(\varphi)})$.

3. Алгоритм сборки нормалей, просчет значений овыпукляемой функции

В алгоритме построения выпуклой оболочки φ^* функции φ использовалось упорядоченное против часовой стрелки множество L нормалей, снятых с многоугольников $\mathcal{M}$, $\mathcal{P}$, $\mathcal{Q}$. Опишем операцию сборки нормалей и способ подсчета значений функции φ по формуле (6).

Предположим, что установлено соответствие между номерами нормалей и вершинами многоугольников $\mathcal{M}$, $\mathcal{P}$, $\mathcal{Q}$. Принцип соответствия ясен из рис. I. Перенумеруем нормали (и соответственно вершины) многоугольников $\mathcal{P}$, $\mathcal{Q}$ так, чтобы номер I в $N(\mathcal{P})$ и $N(\mathcal{Q})$ имели нормали, ближайшие по углу слева к первой нормали из $N(\mathcal{M})$. Символом $\bar{m}_i$ обозначим вектор из $N(\mathcal{M})$ с номером i, пусть $\bar{p}_i$, $\bar{q}_i$ - векторы с номером i из $N(\mathcal{P})$, $N(\mathcal{Q})$ (в новой нумерации). Для обозначения вершин применим символы m_i, p_i, q_i (рис. 2).

Множество L формируется следующим образом. Среди векторов $\bar{m}_1$, $\bar{p}_1$, $\bar{q}_1$ по построению правее двух других по углу лежит вектор $\bar{m}_1$ (возможно совпадение $\bar{m}_1$ с $\bar{p}_1$ или $\bar{q}_1$). Вектор $\bar{m}$ записываем в L под номером I. Если $\bar{m}_1$ было отлично от $\bar{p}_1$, $\bar{q}_1$, то следующая тройка векторов состоит из $\bar{m}_2$, $\bar{p}_1$, $\bar{q}_1$. Если

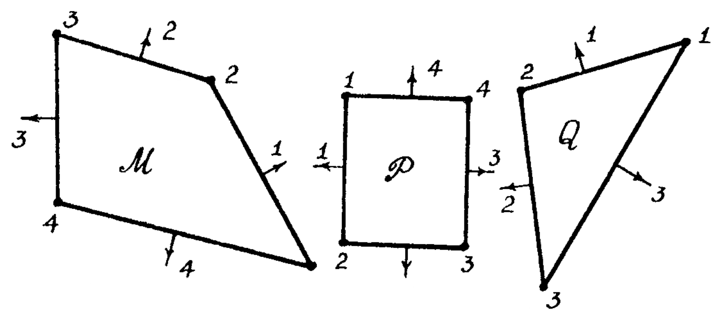

3
2
2
3
1
1
4
4
1
4
3
2
3
2
1
1
2
3
2
3
$\mathscr{M}$
$\mathscr{P}$
Q

Рис. 1

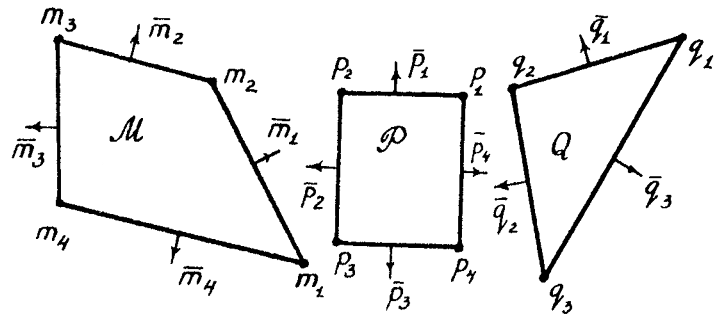

m_3
$\bar{m}_2$
m_2
$\bar{m}_1$
$\bar{m}_3$
m_4
$\bar{m}_4$
m_1
p_2
$\bar{p}_1$
p_1
$\bar{p}_4$
$\bar{p}_2$
p_3
$\bar{p}_3$
p_4
$\bar{q}_1$
q_2
q_1
$\bar{q}_2$
$\bar{q}_3$
q_3
$\mathscr{M}$
$\mathscr{P}$
Q

Рис. 2

$\bar{m}_1$ совпадало с $\bar{p}_1$ или $\bar{q}_1$, то при составлении новой тройки заменим не только $\bar{m}_1$ на $\bar{m}_2$, но и совпадающий с $\bar{m}_1$ вектор на вектор с номером 2 из того же множества. Так, при $\bar{m}_1 = \bar{p}_1$, $\bar{m}_1 \neq \bar{q}_1$ новая тройка будет $\bar{m}_2$, $\bar{p}_2$, $\bar{q}_1$. Если $\bar{m}_1 = \bar{p}_1 = \bar{q}_1$, то новая тройка состоит из векторов $\bar{m}_2$, $\bar{p}_2$, $\bar{q}_2$. Анализируя тройку, полученную на втором шаге, выделяем из нее правый по углу вектор и записываем его под номером 2 в L. Составляем далее третью тройку и т.д. Исчерпав векторы какого-либо из множеств $N(\mathcal{M})$, $N(\mathcal{P})$, $N(\mathcal{Q})$, в очередную тройку записываем первый вектор этого множества. Процесс сборки заканчиваем при получении тройки, состоящей (как и первая тройка) из векторов $\bar{m}_1$, $\bar{p}_1$, $\bar{q}_1$.

Для нормалей рис. 2 порядок сборки таков: $\bar{m}_1$, $\bar{m}_2$, $\bar{p}_1$, $\bar{q}_1$, $\bar{m}_3$, $\bar{q}_2$, $\bar{m}_4$, $\bar{p}_3$, $\bar{q}_3$, $\bar{p}_4$.

При выборе очередного вектора ℓ_i для записи в набор L параллельно на нем просчитывается значение $\varphi(\ell_i)$. Пусть $\bar{m}_s$, $\bar{p}_t$, $\bar{q}_n$ – тройка, из которой на i-ом шаге сборки выбран вектор ℓ_i. Учитывая соответствие между нормалями и вершинами многоугольников $\mathcal{M}$, $\mathcal{P}$, $\mathcal{Q}$, имеем

$$\varphi(\ell_i) = \langle \ell_i, \bar{m}_s \rangle + \langle \ell_i, \bar{p}_t \rangle - \langle \ell_i, \bar{q}_n \rangle .$$

3. Краткая характеристика программы

Программа написана на языке ФОРТРАН-ДУБНА для БЭСМ-6. Имя головной программы *IGRA*, подпрограммы: *FU*, *PAZP*, *BUBOP*, *NORMAN*, *PAZB*, *GATHER*, *A*, *BOF*, *PPOB*, *ZET*. Для интегрирования системы дифференциальных уравнений (5) используется программа из библиотеки стандартных программ ОИЯИ ДУБНА.

Программа *IGRA* предназначена для приближенного построения множеств $\widetilde{W}_{\tau_i}$ для заданных моментов обратного времени τ_i. Допускается в качестве целевого множества M, многогранников P, Q брать отрезок или точку. Предусмотрена возможность умножения M на коэффициент $\gamma>0$, что соответствует замене целевого множества M в игре (3) на множество $\gamma \cdot M$.

Используемые переменные

N – порядок системы (I),

AA(10,10) – матрица A системы (I),

CC(10,10) – матрица C системы (I),

BB(10,10) – матрица B системы (I),

MB – размерность вектора $p \in P$,

MC – размерность вектора $q \in Q$,

KB – количество вершин многоугольника P,

P(10,50) – массив вершин многоугольника P,

KC – число вершин многоугольника Q,

Q(10,50) – массив вершин многоугольника Q,

DL – шаг Δ по оси времени,

TK – момент окончания игры ϑ,

NTN – число моментов обратного времени, в которые происходит выдача результатов на печать,

$TN(40)$ – массив моментов обратного времени, в кото – рые происходит выдача на печать результатов счета,

MG – число вершин многоугольника W_i^* ,

$GB1(1000)$, $GB2(1000)$ – массивы, содержащие первые и вторые координаты вершин многоугольника W_i^* на i –ом шаге попятной процедуры,

$GN1(1000)$, $GN2(1000)$ – массивы, содержащие первые и вторые координаты нормалей к многоугольнику W_i^*,

KNG – число нормалей к W_i^* ,

$PB1(40)$, $PB2(40)$ – массивы, содержащие первые и вторые координаты вершин многоугольника P_i^* ,

MP – число вершин P_i^* ,

$PN1(40)$, $PN2(40)$ – массивы первых и вторых ко – ординат нормалей к многоугольнику $-P_i^*$,

KNP – число нормалей к многоугольнику $-P_i^*$,

$QB1(40)$, $QB2(40)$ – массивы первых и вторых ко – ординат вершин многоугольника Q_i^* ,

MQ – число вершин Q_i^* ,

$QN1(40)$, $QN2(40)$ – массивы первых и вторых ко – ординат векторов нормалей к многоугольнику Q_i^* ,

KNQ – число нормалей к многоугольнику Q_i^* ,

$F(1000)$ – массив значений функции γ_i на векторах $\ell \in L$,

MDL – максимально допустимое число нормалей в L ,

ICT, JCT – номера α, β двух координат фазового вектора x, которые надо привести на множество M,

$TT1$, $TT2$ – коэффициенты a, b, определяющие функцию $\mu(t)$,

$TT3$, $TT4$ – коэффициенты c, d, определяющие функцию $\nu(t)$,

$R(2,10)$ – массив, в котором содержатся элементы матрицы $V(\tau_i)$,

OS – множитель γ перед M.

Ввод данных и вывод результатов

Основная часть исходных данных вводится с перфокарт. Исключение составляет множество M, которое вводится посредством обращения к подпрограмме $PAZB$. Обращение имеет вид

```
CALL PAZB(OS,MG)
```

Подпрограмма $PAZB$ должна иметь следующую форму:

```
SUBROUTINE PAZB(OS,MG)
COMMON /G/ X,Y
DIMENSION X(1000), Y(1000)
{операторы, осуществляющие вычисление первых и
{вторых координат вершин M, засылку их в мас-
{сивы X,Y. MG - число вершин многоугольника
{M = W_0^*
RETURN
END
```

Необходимо, чтобы вершины M были расположены в массивах X, Y против часовой стрелки.

Порядок ввода остальной информации указан в таблице. Элементы матриц располагаются на перфокартах по строкам. Если информация умещается на одной карте, то в графе "Количество чисел на одной перфокарте" стоит прочерк.

Номер оператора *READ*	Вводимая информация	Формат ввода	Количество чисел на одной перфокарте
I.	N, MB, MC, KB, KC	I2	-
2.	ICT, JCT	I2	-
3.	DL, TK	F9.5	-
4.	NTN	I2	-
5.	массив TN	F7.4	I0
6.	MDL	I3	-
7.	массив AA	F9.5	6
8.	массив BB	F9.5	6
9.	массив CC	F9.5	6
I0.	массив P	F9.5	6
II.	массив Q	F9.5	6
I2.	OS	F9.5	-
I3.	$TT1, TT2, TT3, TT4$	F9.5	-

Результатом счета являются вершины многоугольников $W_i^* = W_{\tau_i}^*$, где τ_i - заданные моменты обратного времени (массив TN). Для каждого τ_i информация на печать выдается в следующем виде:

I строка $T = (\tau_i)$

2 строка $MG =$ (число вершин $W^*_{\tau_i}$)

3 строка $SET\ W$

4 строка (координаты вершин $W^*_{\tau_i}$ по 5 пар

⋮ координат в строке. Формат $F11.4$)

После просчета вершин $W^*_{\tau_i}$ для последнего из заданных моментов обратного времени на печать выдается запись $FINISH$. Если на i -ом шаге попятной процедуры произошел обрыв моста, т.е. $W^*_i = \varnothing$, то на печать выдается сообщение $ANGLE\ BETWEEN\ VECTORS\ IS\ MORE\ OR\ EQUAL\ THAN\ PI$ и счет прекращается. Если на $i+1$ -ом шаге общее число нормалей к множествам W^*_i , P^*_i , Q^*_i превышает заранее указанное число MDL , печатается сообщение $ARRAY\ IS\ FULL\ OVER$, затем многоугольник W^*_i заменяется на приближенный с меньшим числом нормалей и счет продолжается.

Особенности программы

Среди управляющих карт мониторной системы должна быть карта $*CALL\ FICMEMORY$, обеспечивающая расширение стандартной памяти.

Тестовая задача

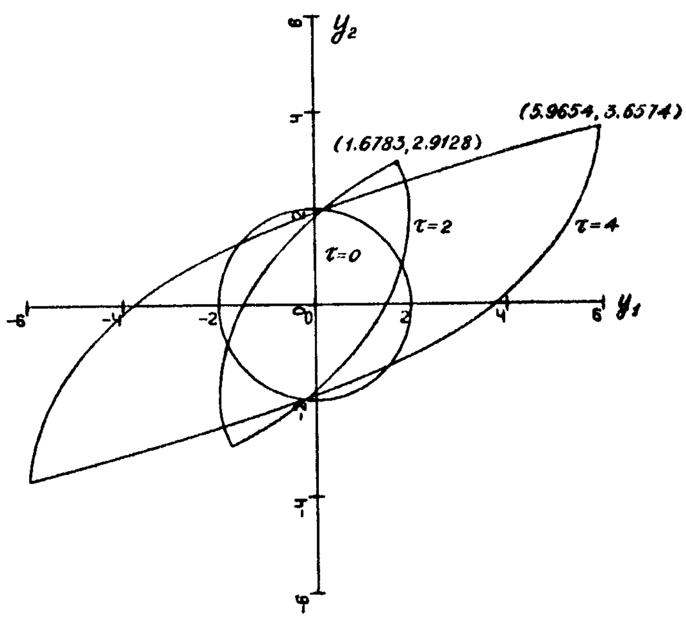
Рассмотрим дифференциальную игру второго порядка

$$\dot{x}_1 = x_2 + v$$

$$\dot{x}_2 = u$$

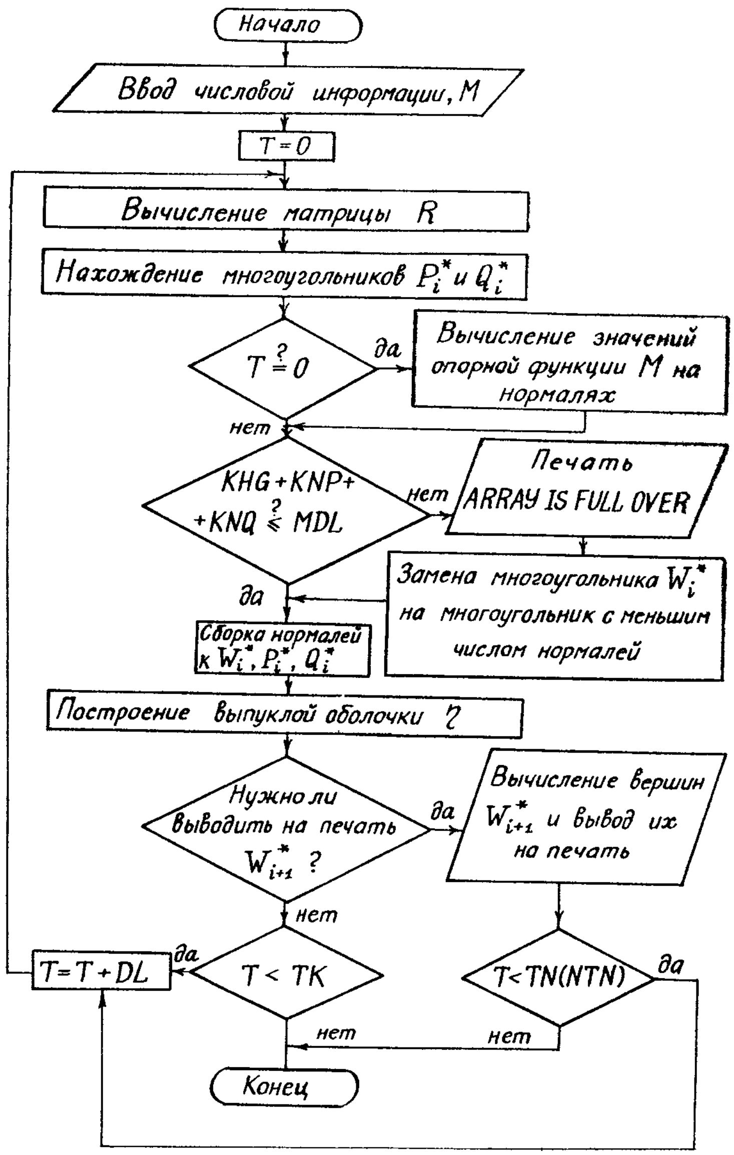
$$|u| \leq 1 \ , \ |v| \leq 1$$

с моментом окончания $\vartheta = 4$. В качестве целевого множества M



возьмем правильный I00-угольник, вписанный в круг радиуса 2. Первая точка имеет координаты (2,0). Множитель γ перед M примем равным I. На рисунке показаны множество M и сечения W_i^* для моментов обратного времени, равных 2,4. При вычислении полагалось Δ = 0.05. Рядом с характерными угловыми точками проставлены их координаты.

Блок-схема



Начало
Ввод числовой информации, M
T=0
Вычисление матрицы R
Нахождение многоугольников P_i^* и Q_i^*
$T \overset{?}{=} 0$
да
Вычисление значений опорной функции M на нормалях
нет
$KHG+KNP+KNQ \overset{?}{\leqslant} MDL$
нет
Печать ARRAY IS FULL OVER
Замена многоугольника W_i^* на многоугольник с меньшим числом нормалей
да
Сборка нормалей к W_i^*, P_i^*, Q_i^*
Построение выпуклой оболочки ?
Нужно ли выводить на печать W_{i+1}^* ?
да
Вычисление вершин W_{i+1}^* и вывод их на печать
нет
T=T+DL
да
T < TK
T<TN(NTN)
да
нет
нет
Конец

ТЕКСТ ПРОГРАММЫ

```
      PROGRAM IGRA
      DIMENSION PPP(10,50),
     *PPQ(10,50),F(1000),
     *AF(1000),NHPB(50)
      DIMENSION PN1(40),
     *PN2(40),QN1(40),QN2(40)
     *,RN1(1000)
      DIMENSION GN1(1000),
     *GN2(1000),GB1(1000),
     *GB2(1000)
      DIMENSION PB1(40),
     *PB2(40),PB3(40),PB4(40)
     *,RN2(1000),TN(40)
      DIMENSION QB1(40),
     *QB2(40),QB3(40),QB4(40)
     *,NHOM(50),NOM(50)
      DIMENSION AA(10,10),
      READ2,ICT,JCT
  335 FORMAT(10F7.4)
      READ335,DL,TK
      READ2,NTN
      READ335,(TN(I),I=1,NTN)
      DO 1111 I=1,NTN
 1111 TN(I)=TN(I)-DL
  336 FORMAT(2I3)
      READ336,MDL
      READ5,((AA(I,J),J=1,N),
     *I=1,N)
      READ5,((BB(I,J),J=1,MB)
     *,I=1,N)
      READ5,((CC(I,J),J=1,MC)
     *,I=1,N)
      READ5,((P(I,J),J=1,KO),
     *I=1,MB)
```

На страницах 147 - 158 идет текст программы на Фортране.

```
33 IND=1
   GOTO 66
41 J=J+1
   B(1,J)=A1
   B(2,J)=A2
66 I=I+1
   IF(I-NN-1)44,45,45
45 NN=J
   IF(IND-1)47,62,62
62 DO 61 K=1,2
   DO 61 I=1,NN
61 A(K,I)=B(K,I)
   IND=0
   GOTO 46
47 DO 48 I=1,NN
   B(1,I)=B(1,I)+S1
48 B(2,I)=B(2,I)+S2
   NN=NN+1
     B(1,NN)=S1
     B(2,NN)=S2
     DO 200 I=1,NN
     X1(I)=B(1,I)
 200 X2(I)=B(2,I)
     GOTO(1001,1002)IL
1001 RETURN
1002 DO 1005 I=1,NN
     A(1,I)=X1(I)
1005 A(2,I)=X2(I)
     DO 1006 I=1,NN
     X1(I)=A(1,NN+1-I)
1006 X2(I)=A(2,NN+1-I)
     RETURN
     ENTRY BUBOP1
     IL=2
     GOTO 1003
     END
```

Литература

I. Боткин Н.Д. Оценка погрешности численных построений в дифференциальной игре с фиксированным моментом окончания. – Пробл. управления и теории информ., 1982, т. II, № 4, с. 283-295.

2. Красовский Н.Н., Субботин А.И. Позиционные дифференциальные игры. – М.: Наука, 1974. – 456 с.

3. Пономарев А.П. Оценка погрешности численного метода построения альтернированного интеграла Понтрягина. – Вестн. Моск. ун-та, сер. 15. Выч. матем. и кибернет., 1978, № 4, с. 37-43.

4. Пшеничный Б.Н., Сагайдак М.И. О дифференциальных играх с фиксированным временем. – Кибернетика, 1970, № 2, с.54-63.

5. Субботин А.И., Ченцов А.Г. Оптимизация гарантии в задачах управления. – М.: Наука, 1981. – 288 с.