УДК 519.6

Применение метода частиц в стае к задаче поиска оптимального управления дискретными детерминированными системами.

А.В. Пантелеев, Е.А. Алёшина

Аннотация

Рассмотрена задача поиска оптимального управления дискретной детерминированной системой с помощью метода частиц в стае. Сформирован детальный алгоритм решения поставленной задачи с использованием различных модификаций метода. На основе разработанного алгоритма создано программное обеспечение, позволяющее не только решить поставленную задачу, но и отобразить результат в наглядной форме. Эффективность программного обеспечения продемонстрирована на примерах.

Ключевые слова

Оптимальное управление, дискретная детерминированная система, метаэвристические алгоритмы, метод частиц в стае, глобальная оптимизация.

Введение

В данной работе рассматривается применение метода частиц в стае [1-3] к задаче синтеза оптимального программного управления детерминированными дискретными системами [4]. Для частного случая, когда система линейная, разработаны алгоритм и программа, позволяющие решать задачу при различных значениях параметров метода. С помощью сформированного программного обеспечения решено несколько тестовых примеров, на которых продемонстрирована эффективность предложенного подхода. Полученный результат может быть использован при решении разнообразных задач параметрической оптимизации сложных ракетно-космических систем, в которых делается переход от непрерывных моделей движения летательных аппаратов к их дискретным аналогам путем дискретизации.

Постановка задачи

Поведение модели объекта управления описывается разностным уравнением

$$x(t+1) = f\left(t, x(t), u(t)\right), \; t = 0,1,\dots,N-1, \tag{1}$$

где x – вектор состояния системы, $x \in \mathbf{R}^n$; u – вектор управления, $u \in U(t) \subseteq \mathbf{R}^q$, $U(t)$ – некоторое замкнутое выпуклое множество допустимых значений управления; t – дискретное время, $t \in T = \left[0,1,\dots,N-1\right]$, число шагов N задано, $f\left(t,x,u\right) = \left(f_1\left(t,x,u\right),\dots,f_n\left(t,x,u\right)\right)^T$ – заданная вектор-функция,.

Начальное состояние системы задано

$$x(0) = x_0. \tag{2}$$

Правый конец траектории свободен. Предполагается, что при управлении используется информация только о дискретном времени t, т.е. применяется так называемое программное управление.

Множество допустимых процессов $\mathcal{D}(0, x_0)$ – это множество пар $d = \left(x(\cdot), u(\cdot)\right)$, включающих траекторию $x(\cdot) = \left\{x_0, x(1),\dots,x(N)\right\}$ и управление $u(\cdot) = \left\{u(0), u(1),\dots u(N-1)\right\}$, $u(t) \in U(t)$, удовлетворяющих уравнению состояния (1) с начальным состоянием x_0.

На множестве допустимых процессов определен функционал качества управления

$$I(d) = \sum_{t=0}^{N-1} f^0\left(t, x(t), u(t)\right) + F\left(x(N)\right), \tag{3}$$

где $f^0\left(t,x,u\right)$, $F\left(x\right)$ – заданные функции.

Требуется найти такую пару $d^* = \left(x^*(\cdot), u^*(\cdot)\right) \in \mathcal{D}(0, x_0)$, что

$$I(d^*) = \min_{d \in \mathcal{D}(0, x_0)} I(d) \qquad (4)$$

Стратегия поиска решения

Пусть имеется стая из NP частиц с позициями $\left\{x^1, x^2, ..., x^{NP}\right\} \subseteq D \subset R^n$. На каждом этапе (итерации) у частицы с номером j выбирается случайное число соседей $NI^j \sim R\left[NI_{\min}, NI_{\max}\right]$ из членов стаи, определяемое равномерным (R) распределением на отрезке $\left[NI_{\min}, NI_{\max}\right]$, где $NI_{\min}$ – минимальное число соседей, $NI_{\max}$ – максимальное число соседей (не превышающее NP). Среди соседей находится частица с номером n_j и позицией $\tilde{x}^{n_j,k}$, которой соответствует наименьшее значение целевой функции. Каждая частица с номером j хранит информацию о своей лучшей позиции $\tilde{x}^{j,k}$, достигнутой в течение k итераций. Скорость частицы $v^{j,k+1}$ с учётом текущего положения и скорости, своей наилучшей позиции $\tilde{x}^{j,k}$ в прошлом, позиции $\tilde{x}^{n_j,k}$ лидера среди соседей вычисляется по формуле:

$$v^{j,k+1} = \omega v^{j,k} + \alpha r_1\left(\tilde{x}^{j,k} - x^{j,k}\right) + \omega\beta r_2\left(\tilde{x}^{n_j,k} - x^{j,k}\right), \qquad (5)$$

где k – номер итерации; $v^{j,k}, x^{j,k}$ – скорость и позиция частицы на k-й итерации соответственно; ω – весовой параметр, характеризующий инерцию (память) частицы; α и β – параметры, учитывающие степень влияния лидера среди соседей и прошлого опыта частицы (весовые коэффициенты) на её последующую скорость; r_1 и r_2 – случайные параметры, заданные равномерным законом распределения $R\left[0;1\right]$.

Новое положение частицы находится суперпозицией текущего положения и перемещения частицы в единицу времени (т.е. скорости):

$$x^{j,k+1} = x^{j,k} + v^{j,k+1}. \qquad (6)$$

Если новая позиция $x^{j,k+1}$ оказалась лучше, чем $\tilde{x}^{j,k}$, необходимо в качестве наилучшей позиции $\tilde{x}^{j,k+1}$ принять $x^{j,k+1}$. Расчёт новой скорости и новой позиции по формулам (5) и (6) производится для $j = 1,...,NP$, т.е. для всех частиц стаи. После этого выполняется следующая итерация, затем следующая, и так до выполнения условий окончания расчета. В качестве условия окончания обычно выбирается равенство числа итераций k некоторому наперед заданному числу K (максимальному числу итераций).

При использовании метода частиц в стае для решения задачи (1)-(4) будем искать наилучшее управление $u(\cdot)$. Позиция частицы под номером j на k-й итерации будет представлять собой вектор-строку $u^{j,k} = \left(u^{j,k}(0), u^{j,k}(1),...,u^{j,k}(N-1)\right)$. На каждой итерации будет формироваться новая позиция частицы $u^{j,k+1} = \left(u^{j,k+1}(0), u^{j,k+1}(1),...,u^{j,k+1}(N-1)\right)$, ее необходимо будет оценить с помощью критерия (3), который будет играть роль целевой функции метода частиц в стае. Для получения значения целевой функции потребуется:

а) найти траекторию системы $x^{j,k+1} = \left(x_0, x^{j,k+1}(1),...,x^{j,k+1}(N)\right)$, соответствующую управлению $u^{j,k+1}$, из уравнения состояния (1) с учетом условия (2);

б) вычислить значение критерия (3), соответствующее $u^{j,k+1}$ и $x^{j,k+1}$:

$$I\left(d^{j,k+1}\right) = \sum_{t=0}^{N-1} f^0\left(t, x^{j,k+1}(t), u^{j,k+1}(t)\right) + F\left(x^{j,k+1}(N)\right), \qquad (7)$$

где $d^{j,k+1} = \left(x^{j,k+1}(\cdot), u^{j,k+1}(\cdot)\right)$ – процесс из множества допустимых процессов $\mathcal{D}(0, x_0)$, $x^{j,k+1}(t), t = 0,...,N,$ определяется уравнением (1) и начальным условием (2):

$$x^{j,k+1}(t) = \begin{cases} f\left(t-1, x^{j,k+1}(t-1), u^{j,k+1}(t-1)\right), t = 1,\dots,N, \\ x_0, t = 0. \end{cases} \quad (8)$$

Алгоритм применения метода частиц в стае к задаче поиска оптимального управления дискретными детерминированными системами

Шаг 1. Задать NP – количество частиц в стае; $NI_{\min}$, $NI_{\max}$ – параметры выбора числа «соседей» в стае; K – максимальное число итераций; весовой коэффициент ω, характеризующий инерцию; параметры α и β, используемые при вычислении скорости частицы.

Шаг 2. Сформировать стаю частиц:

- положить число итераций $k = 0$;
- в области D допустимых решений задать NP частиц с позициями $u^{1,0}, u^{2,0}, \dots, u^{NP,0}$. Для этого использовать равномерный закон распределения, т.е. для каждой частицы с номером $j = 1,\dots,NP$ найти позицию $u^{j,0} = \left(u^{j,0}(0), \ldots, u^{j,0}(N-1)\right)$, полагая $u_i^{j,0}(0) \sim R\left[a_i, b_i\right]$, …, $u_i^{j,0}(N-1) \sim R\left[a_i, b_i\right]$, $i = 1,\dots,q$;
- положить начальные скорости равными нулю: $v^{j,0} = 0$, $j = 1,\dots,NP$.

Шаг 3. Для каждой частицы $j = 1,\dots,NP$ найти:

- случайное число NI^j соседей при помощи равномерного распределения на отрезке $\left[NI_{\min}, NI_{\max}\right]$, среди них найти частицу с номером и позицией $\tilde{u}^{n_j,k}$, которой соответствует наименьшее среди выбранных соседей значение целевой функции.
- наилучшую позицию $\tilde{u}^{j,k}$ в течение k итераций (при $k = 0$ положить $\tilde{u}^{j,0} = u^{j,0}$).

Шаг 4. Для каждой частицы в стае ($j = 1,\dots,NP$) найти новую скорость и новое положение по формулам (5) и (6). Для нового положения частицы $u^{j,k+1}$ вычислить значение критерия

$I\left(d^{j,k+1}\right)$ по формуле (7) с использованием формулы (8) для нахождения компонент вектора состояния $x^{j,k+1}$. Обозначим допустимый процесс, в который входит $\tilde{u}^{j,k}$, через $\tilde{d}^{j,k}$, $\tilde{d}^{j,k} = \left(\tilde{x}^{j,k}(\cdot), \tilde{u}^{j,k}(\cdot)\right)$. Если $I\left(d^{j,k+1}\right) < I\left(\tilde{d}^{j,k}\right)$, запоминается новая наилучшая позиция частицы: $\tilde{u}^{j,k+1} = u^{j,k+1}$, $I\left(\tilde{d}^{j,k+1}\right) = I\left(d^{j,k+1}\right)$.

Шаг 5. Проверка условия окончания работы алгоритма. Если $k < K$, полагаем $k = k+1$ и переходим к Шагу 3. Если $k = K$, расчёт окончен. В качестве решения задачи $u^* = \left(u^*(0), u^*(1), ..., u^*(N-1)\right)$ выбирается позиция частицы в стае с наименьшим значением целевой функции и вектор $x^* = \left(x_0, x^*(1), ..., x^*(N)\right)$, который рассчитывается по уравнению состояния.

Программное обеспечение

На основе разработанного алгоритма сформировано программное обеспечение на языке C# в среде Microsoft Visual Studio 2005. Разработан интерфейс пользователя, позволяющий: вводить параметры постановки задачи; задавать параметры метода; использовать модификации алгоритма; анализировать полученный результат. Для удобства реализации на компьютере в постановке задачи были сделаны следующие упрощения.

Уравнение состояния полагалось линейным:

$$x(t+1) = Ax(t) + Bu(t), \quad t = 0,1,...,N-1, \tag{7}$$

где A – матрица размеров $n \times n$, B – матрица размеров $n \times q$.

Функционал качества управления считался квадратичным или линейно-квадратичным (для случая $n = 1$):

$$I_1(d) = \sum_{t=0}^{N-1}\left(x^T(t)Sx(t) + u^T(t)Qu(t)\right) + L\,x(N), \tag{8}$$

$$I_2(d)=\sum_{t=0}^{N-1}\left(x^T(t)Sx(t)+u^T(t)Qu(t)\right)+x^T(N)L\,x(N), \qquad (9)$$

где S – матрица размеров $n\times n$, Q – матрица размеров $q\times q$, L – матрица размеров $n\times n$ (для $n=2$) или число для $n=1$.

Множество допустимых значений управления $U(t)$ для каждого t полагалось равным прямому произведению отрезков $\left[a_{t,i},b_{t,i}\right]$, $i=\overline{1,q}$.

Пример 1. Даны модель объекта управления

$$x(t+1)=x(t)+u(t),$$

где $x\in\mathbf{R}$, $u\in\mathbf{R}$, $t=0,1,...,N-1$; начальное состояние $x(0)=x_0$ и функционал $I=\sum_{t=0}^{N-1}\frac{1}{t+1}u^2(t)+2x(N)$. Требуется найти минимальное значение функционала и оптимальный процесс $\left(x^*(\cdot),u^*(\cdot)\right)$, на котором это значение достигается.

Сравнивая с (7) и (8), имеем: $n=1$, $q=1$, $A=1$, $B=1$, $S=0$, $Q=\frac{1}{t+1}$, $L=2$.

Будем использовать рекомендуемые значения параметров метода частиц в стае [1-3]: $NP=30$, $K=100$, $NI_{\min}=15$, $NI_{\max}=25$, $\omega=0{,}5$, $\alpha=\beta=0{,}5$.

На рис.1 показано главное окно программы.

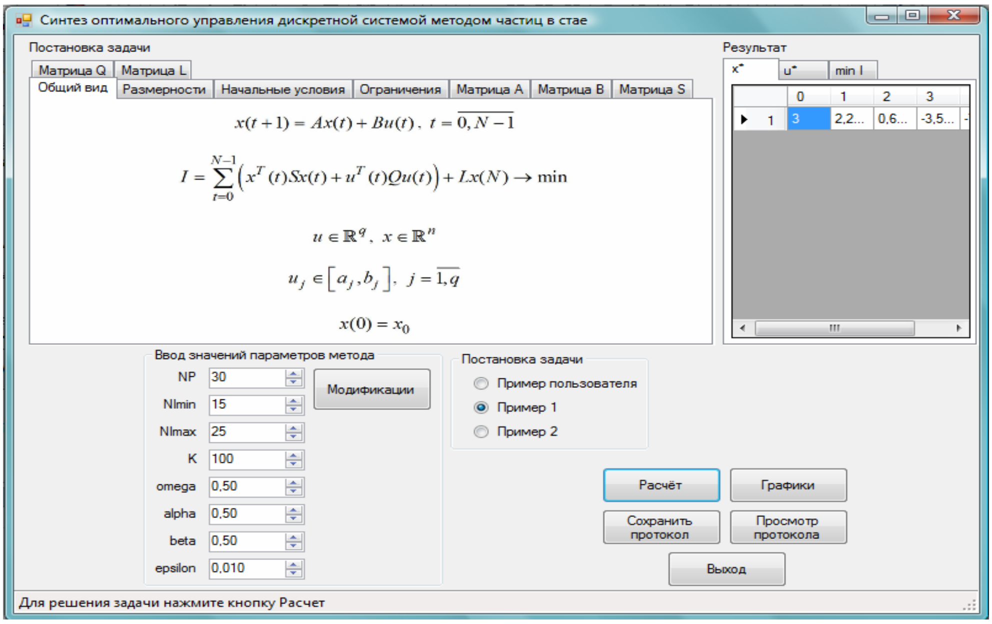


Рис. 1. Ввод параметров постановки задачи и метода

Структура главного окна программы такова: слева вводятся условие задачи и параметры метода частиц в стае. Кроме того, можно выбрать один из тестовых примеров для решения. Справа внизу находятся кнопки управления программой, а справа вверху отображается результат работы программы – таблицы со значениями оптимального управления, оптимальной траектории и минимальным значением критерия. Для ввода условия задачи необходимо заполнить поля ввода и таблицы на вкладках в левой части окна. В качестве значений параметров метода частиц в стае по умолчанию выбираются рекомендуемые параметры, это отображается в полях ввода. Значения этих параметров можно изменить. При выборе одного из тестовых примеров все поля заполняются автоматически. Для решения задачи необходимо нажать кнопку «Расчет». Программа не поддерживает пошаговое отображение процесса поиска решения, так как это действие существенно увеличивает время работы алгоритма. По окончании процесса поиска решения результат отображается в таблицах справа и на графиках, которые вызываются по кнопке «Графики» (см. рис. 2).

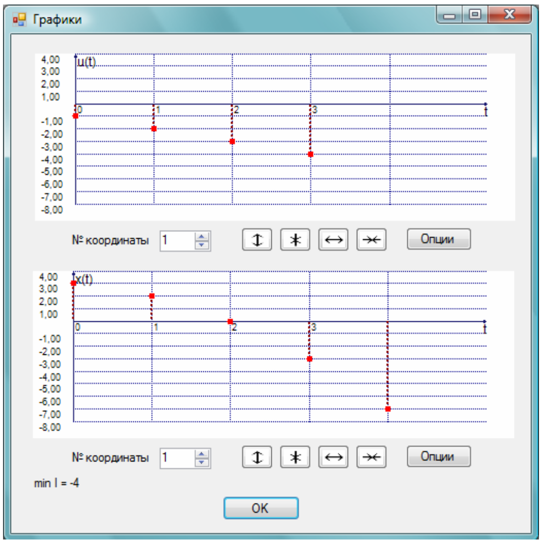
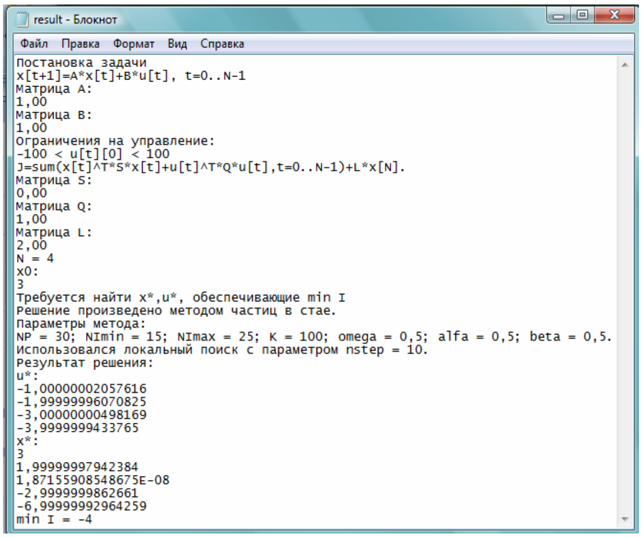


Рис. 2. Оптимальные траектория и управление



```
Постановка задачи
x[t+1]=A*x[t]+B*u[t], t=0..N-1
Матрица A:
1,00
Матрица B:
1,00
Ограничения на управление:
-100 < u[t][0] < 100
J=sum(x[t]^T*S*x[t]+u[t]^T*Q*u[t],t=0..N-1)+L*x[N].
Матрица S:
0,00
Матрица Q:
1,00
Матрица L:
2,00
N = 4
x0:
3
Требуется найти x*,u*, обеспечивающие min I
Решение произведено методом частиц в стае.
Параметры метода:
NP = 30; NImin = 15; NImax = 25; K = 100; omega = 0,5; alfa = 0,5; beta = 0,5.
Использовался локальный поиск с параметром nstep = 10.
Результат решения:
u*:
-1,00000002057616
-1,99999996070825
-3,0000000498169
-3,9999999433765
x*:
3
1,99999997942384
1,87155908548675E-08
-2,9999999862661
-6,99999992964259
min I = -4
```

Рис. 3. Протокол решения примера 1

Полученный результат можно сохранить в памяти компьютера, чтобы он был доступен после закрытия программы. Для этого необходимо нажать кнопку «Сохранить протокол». Для просмотра файла с записанным в него результатом необходимо нажать кнопку «Просмотр протокола». При решении примеров была выбрана модификация с использованием локального поиска, так как она является наиболее эффективной (подробнее об этом в [2-3]). На рис. 3 приведен протокол решения примера 1 при $N = 4$ и $x_0 = 3$, а на рис. 2 – графики оптимального управления и оптимальной траектории.

Точное решение задачи известно [4]: $u^*(t) = -t - 1$, $x^*(t) = x_0 - \frac{1}{2}t(t+1)$, $\min I = 2x_0 - \frac{1}{2}N(N+1)$. Для $N = 4$ и $x_0 = 3$ решение будет выглядеть так: $u^*(0) = -1$, $u^*(1) = -2$, $u^*(2) = -3$, $u^*(3) = -4$, $x^*(0) = x_0 = 3$, $x^*(1) = 2$, $x^*(2) = 0$, $x^*(3) = -3$, $x^*(4) = -7$, $\min I = -4$. Таким образом, можно оценить точность решения, которое получается в результате работы алгоритма. Как видно из протокола, использование модификации с локальным поиском дает решение, практически совпадающее с оптимальным.

Пример 2. Даны модель объекта управления:

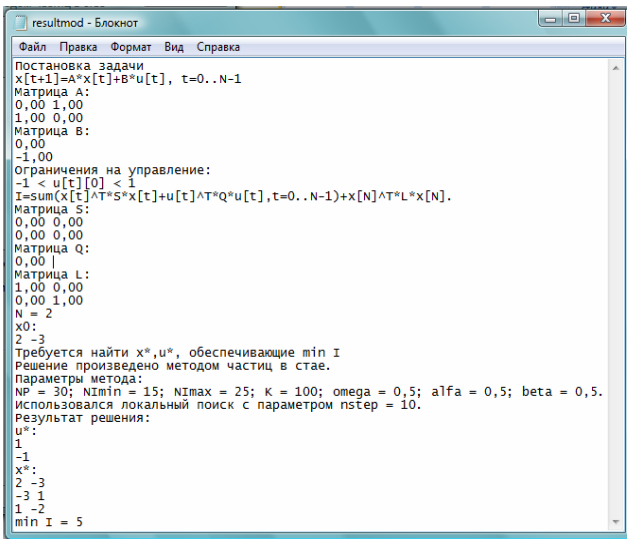
$$x_1(t+1) = x_2(t),$$

$$x_2(t+1) = x_1(t) - u(t),$$

где $x \in \mathbf{R}^2$, $|u| \le 1$, $t = 0,1$; начальные условия $x_1(0) = 2$, $x_2(0) = -3$ и функционал $I = x_1^2(2) + x_2^2(2)$. Требуется найти минимальное значение функционала и оптимальный процесс $\left(x^*(\cdot), u^*(\cdot)\right)$, на котором это значение достигается.

Сравнивая с (7) и (9), имеем: $N = 2$, $n = 2$, $q = 1$, $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, $S = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $Q = 0$, $L = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Решение задачи производилось с использованием рекомендуемых значений параметров метода частиц в стае $NP = 30$, $K = 100$, $NI_{\min} = 15$, $NI_{\max} = 25$, $\omega = 0{,}5$, $\alpha = \beta = 0{,}5$.

Применив модификацию с использованием локального поиска, получим $u^*(0) = 1$, $u^*(1) = -1$, $x^*(0) = x_0 = \begin{pmatrix} 2 & -3 \end{pmatrix}^T$, $x^*(1) = \begin{pmatrix} -3 & 1 \end{pmatrix}^T$, $x^*(2) = \begin{pmatrix} 1 & -2 \end{pmatrix}^T$, $\min I = 5$. Протокол приведен на рис. 4.



```
resultmod - Блокнот
Файл  Правка  Формат  Вид  Справка
Постановка задачи
x[t+1]=A*x[t]+B*u[t], t=0..N-1
Матрица A:
0,00 1,00
1,00 0,00
Матрица B:
0,00
-1,00
Ограничения на управление:
-1 < u[t][0] < 1
I=sum(x[t]^T*S*x[t]+u[t]^T*Q*u[t],t=0..N-1)+x[N]^T*L*x[N].
Матрица S:
0,00 0,00
0,00 0,00
Матрица Q:
0,00 |
Матрица L:
1,00 0,00
0,00 1,00
N = 2
x0:
2 -3
Требуется найти x*,u*, обеспечивающие min I
Решение произведено методом частиц в стае.
Параметры метода:
NP = 30; NImin = 15; NImax = 25; K = 100; omega = 0,5; alfa = 0,5; beta = 0,5.
Использовался локальный поиск с параметром nstep = 10.
Результат решения:
u*:
1
-1
x*:
2 -3
-3 1
1 -2
min I = 5
```

Рис. 4. Протокол решения примера 2

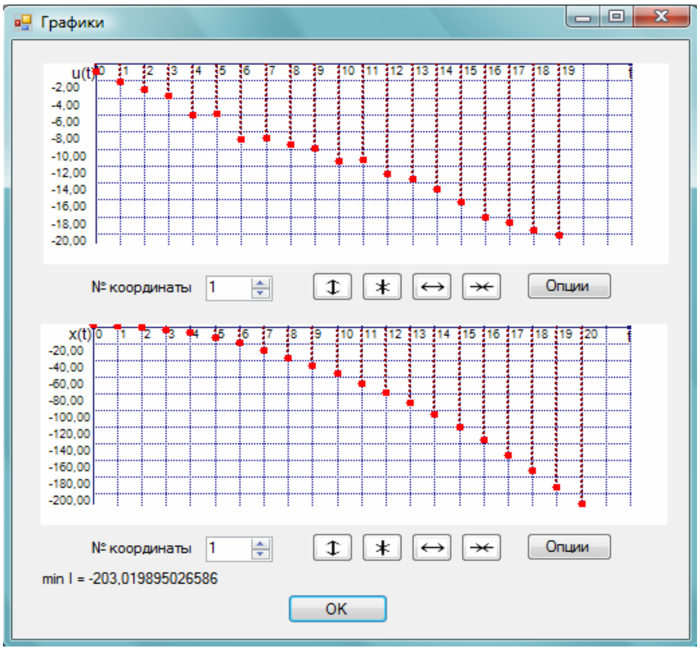


Рис. 5. Результаты решения примера 3

Пример 3. Решим пример 1 при $N = 20$ и тех же значениях параметров метода. Результаты представлены на рис. 5. Точное минимальное значение критерия для $N = 20$ и $x_0 = 3$ равно -204.

Работа выполнена в научно-образовательном центре "Математические методы оптимизации и идентификации аэрокосмических систем и летательных аппаратов", как часть работ по Государственному контракту 02.740.11.0471 в рамках Мероприятия 1.1 Федеральной целевой программы "Научные и научно-педагогические кадры инновационной России" на 2009-2013 гг.

Выводы

Рассмотрен алгоритм решения задачи нахождения оптимального программного управления дискретными детерминированными многомерными системами с помощью метода частиц в стае. Разработано соответствующее программное обеспечение, работоспособность которого продемонстрирована на примерах.

Библиографический список

1. Kennedy, J. and Eberhart, R. Particle Swarm Optimization // Сайт в Интернете http://www.engr.iupui.edu/~shi/Coference/psopap4.html.
2. Пантелеев А. В., Алёшина Е. А. Разработка алгоритмического и программного обеспечения метода частиц в стае // Научный вестник МГТУ ГА, серия Прикладная математика. Информатика, № 145, 2009. С. 32-39.
3. Пантелеев А.В. Метаэвристические алгоритмы поиска глобального экстремума.- М.: МАИ-ПРИНТ, 2009.
4. Пантелеев А.В., Бортаковский А.С. Теория управления в примерах и задачах.- М.: Высш. шк., 2003.

Сведения об авторах

Пантелеев Андрей Владимирович, Московский авиационный институт, заведующий кафедрой, доктор физико-математических наук, профессор, E-mail: dep805@mai.ru.

Алёшина Екатерина Александровна, Московский авиационный институт, студент, E-mail: katya_bzuk@mail.ru