

## ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МЕТОДА СЕТЕВОГО ОПЕРАТОРА ДЛЯ СИНТЕЗА РОБАСТНОЙ СИСТЕМЫ УПРАВЛЕНИЯ КОСМИЧЕСКИМ АППАРАТОМ

Дивеев А.И.<sup>2</sup>, Уваров А.С.<sup>1</sup>, Хамадияров Д.Б.<sup>1</sup>

<sup>1</sup>ФГБОУ Российский университет дружбы народов, ул. Орджоникидзе, 3, Москва, Россия, 115419, aidiveev@mail.ru

<sup>2</sup>Федеральное государственное бюджетное учреждение науки Вычислительный центр им. А.А. Дородницына Российской академии наук, ул. Вавилова, 40, Москва, Россия, 119333

---

Рассматривается задача синтеза робастного управления. Приведена математическая формулировка задачи. Для достижения свойства робастности системы управления множества значений неопределенных параметров покрывается сеткой. Расчет функционалов производится для каждого узла сетки. Оценка качества определяется суммой значений функционалов для всех точек расчета. Неопределенные параметры и интенсивность разбиения их множеств задаются непосредственно исследователем. Для решения задачи синтеза используется метод сетевого оператора. С целью повышения эффективности расчета использованы методы параллельных вычислений. Разработанный параллельный алгоритм метода сетевого оператора вычисляет одновременно вариации для нескольких независимых базисных решений и для нескольких точек множества неопределенных параметров, что позволяет эффективно нагружать процессоры в процедуре расчета функционалов. Проведен анализ эффективности алгоритма при разных параметрах расчета и распределении нагрузки на ядра процессоров. Представлены результаты применения разработанного алгоритма для прикладной задачи синтеза робастного управления спуском космического аппарата на Луну в условиях параметрической неопределенности.

---

Ключевые слова: синтез системы управления, метод сетевого оператора, параллельный алгоритм, робастные системы управления

## THE PARALLEL ALGORITHM OF THE NETWORK OPERATOR METHOD FOR SYNTHESIS OF A SPACECRAFT ROBUST CONTROL SYSTEM

Diveev A.I.<sup>2</sup>, Uvarov A.S.<sup>1</sup>, Khamadiyarov D.B.<sup>1</sup>

<sup>1</sup>Cybernetics and mechatronics department, Peoples' Friendship University of Russia, Ordjonikidze str., 3, Moscow, Russia, 115419, aidiveev@mail.ru

<sup>2</sup>Dorodnicyn Computer Center of Russian Academy of Sciences, Vavilov str., 40, Moscow, Russia, 119333

---

The problem of the robust control synthesis is considered. The mathematical formulation of the problem is given. The set of uncertain parameters values is covered by a grid to achieve the robustness. The functionals are calculated for each grid node. Quality is estimated by the sum of functional values for all calculation points. Uncertain parameters and the partition intensity are set directly by a researcher. To solve the synthesis problem the network operator method is used. To improve calculation efficiency parallel programming approaches are used. The developed parallel algorithm of the network operator method calculates at the same time variations of several independent basis solutions and several points of the uncertain parameters set. This approach lets to load processors efficiently by the functional calculation procedure. The calculations efficiency is analyzed by different calculation parameters and cores loading distribution. The results of developed algorithm application for the robust control synthesis problem of spacecraft landing on the moon by parametric uncertainty are presented.

---

Keywords: system control synthesis, network operator method, parallel algorithms, robust control systems

Статья посвящена численному решению задачи синтеза системы управления. Синтез системы управления обеспечивает нахождение функции, описывающей зависимость управления от координат пространства состояний объекта. Рассматривается задача синтеза робастной системы управления, в которой модель объекта управления имеет множество неопределенных параметров. С целью обеспечения свойства робастности множество неопределенных параметров покрывается сеткой определенных значений. Для решения задачи применяется метод сетевого оператора [1,2,5], который использует кодирование

математического выражения в форме целочисленной матрицы. Метод сетевого оператора относится к классу современных методов символьной регрессии. Все эти методы производят поиск оптимального решения в форме кода с помощью эволюционных алгоритмов [4]. Эффективность этих алгоритмов определяется мощностью начального множества возможных решений и количеством циклов преобразования этого множества. В рассматриваемой задаче множество неопределенных значений существенно усложняет процесс поиска решения. Для повышения эффективности метода сетевого оператора применяются параллельные технологии создания алгоритмов. В работе представлен анализ параллельных алгоритмов метода сетевого оператора. Рассматриваются подходы, обеспечивающие параллельное вычисление функционалов на множестве значений неопределенных параметров и параллельное вычисление вариаций для множества базисных решений [3,7-9]. Рассматривается пример синтеза робастной системы управления спуском космического аппарата (КА) на поверхность Луны.

#### **Задача численного синтеза робастного управления**

Модель объекта управления задана системой обыкновенных дифференциальных уравнений:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u}), \quad (1)$$

где функция  $\mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{u})$  - вектор-функция с векторными аргументами,  $\mathbf{x}$  – вектор состояния,  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u}$  – вектор управления,  $\mathbf{u} \in U \subseteq \mathbb{R}^m$ ,  $n \geq m$ ,  $\mathbf{y}$  – вектор неопределенных параметров,  $\mathbf{y} \in Y \subseteq \mathbb{R}^l$ ,  $U, Y$  - замкнутые и ограниченные множества.

Для системы (1) заданы начальные условия, значения которых могут зависеть от неопределенных параметров:

$$\mathbf{x}(0) = \mathbf{g}^0(\mathbf{x}^0, \mathbf{y}), \quad (2)$$

где  $\mathbf{x}^0$  – вектор заданных начальных значений.

Заданы терминальные условия:

$$\varphi_i(\mathbf{x}(t_f)) = 0, i = \overline{1, k}, k \leq n, \quad (3)$$

где  $t_f$  – время процесса управления определяется из соотношения:

$$t_f = \begin{cases} t, \max_i \{|\varphi_i(\mathbf{x}(t_f))|\} = 0, t \leq t^+ \\ t^+, \text{ иначе} \end{cases}, \quad (4)$$

где  $t^+$  задано и определяет верхнюю границу возможного времени управления.

Задан функционал качества:

$$J = \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \rightarrow \min, \quad (5)$$

Необходимо найти функцию управления в виде:

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \quad (6)$$

где  $\mathbf{h}(\mathbf{x})$  - вектор-функция  $\mathbf{h}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  со следующими свойствами:

$$\mathbf{h}(\mathbf{x}(t, \mathbf{y})) \in U,$$

$$\int_0^{t_f} f_0(\mathbf{x}(t, \mathbf{y}), \mathbf{h}(\mathbf{x}(t, \mathbf{y}))) dt = \min_{\mathbf{u} \in U} \int_0^{t_f} f_0(\mathbf{x}(t, \mathbf{y}), \mathbf{u}) dt, \quad (7)$$

$$\varphi_i(\mathbf{x}(t_f, \mathbf{y})) = 0, i = \overline{1, k},$$

где  $\mathbf{x}(t, \mathbf{y})$  - решение системы дифференциальных уравнений  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{h}(\mathbf{x}))$ ,  $\forall \mathbf{y} \in Y$ .

Для решения данной задачи (1-6) используем численный метод сетевого оператора. Множество неопределенных параметров заменяем конечным множеством определенных значений из заданного множества:

$$\tilde{Y} = \{\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^P\}. \quad (8)$$

Заменяем терминальные условия (3) дополнительным функционалом, а функционал качества (5) заменяем суммой функционалов, вычисленных для каждой точки из множества неопределенных параметров (8). В результате получаем задачу многокритериальной оптимизации с критериями:

$$J_1 = \sum_{i=1}^P \left( \sum_{j=1}^k \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \right)_{\mathbf{y}^i} \rightarrow \min, \quad (9)$$

$$J_2 = \sum_{i=1}^P \left( \sum_{j=1}^k |\varphi_j(\mathbf{x}(t))| \right)_{\mathbf{y}^i} \rightarrow \min. \quad (10)$$

### Метод сетевого оператора

Метод сетевого оператора был описан во многих работах [1,2,5]. Метод сетевого оператора позволяет представить математическое выражение в виде целочисленной матрицы, что дает больше гибкости при программировании данного метода. В данной работе не производится никаких аналитических изменений в исходной постановке задачи, однако в данном случае выбраны неопределенные параметры объекта, значения которых задаются множеством точек в заданном интервале. Данное преобразование придает свойство робастности системе управления. При этом функционалы высчитываются для каждой точки из множества неопределенных параметров, а результирующим считается сумма функционалов по всем точкам каждого параметра. В данной работе применяется параллельная реализация метода сетевого оператора. Такая реализация позволяет рассмотреть большее количество возможных решений без существенного увеличения временных затрат, а соответственно, шанс найти необходимое решение возрастает. Также в

данной реализации предоставляется возможность сократить время всего алгоритма за счет распараллеливания процесса вычисления функционалов.

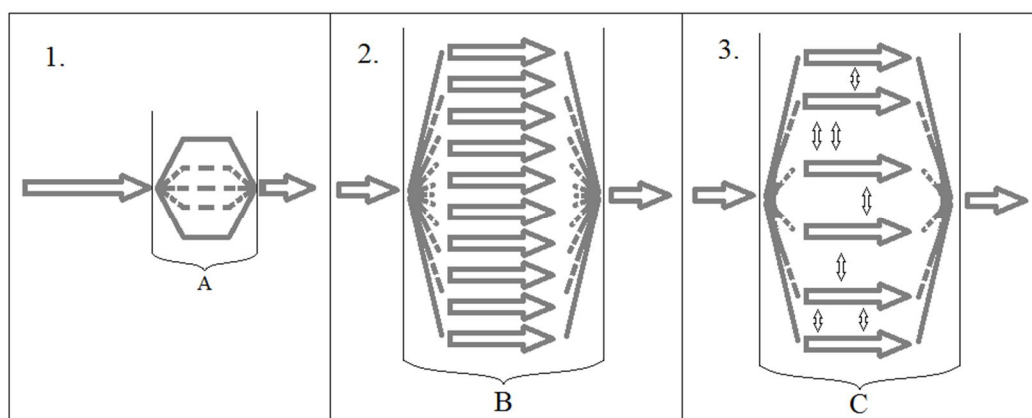
### Принципы распараллеливания

Поисковая часть метода сетевого оператора основана на генетическом алгоритме. Существует несколько моделей распараллеливания генетических алгоритмов [7-9], из них три основных, а остальные являются их расширениями и комбинациями:

1. Master-slave (ведущий-ведомый) модель. Модель, при которой основной поток (ведущий, master) обрабатывает единственную популяцию. Таким образом, весь генетический алгоритм производится master-поток. Остальные потоки (ведомые, slave) лишь получают от master-потока задания на расчет значений критериев (функционалов). В силу существования ведущего и ведомых потоков, данную модель часто называют master-slave моделью (ведущий-ведомый). Такая модель особенно эффективна, если критерии описываются сложными математическими выражениями;

2. Клеточная модель. В данной модели популяция разбивается на мелкие части (клетки). Такая модель часто именуется выражение «мелкозернистый» параллелизм (fine grained parallelism). Основной идеей данной модели является разделение популяции на мелкие группы хромосом (обычно по одной или две). Таким образом, хромосомы могут взаимодействовать только внутри своей подгруппы;

3. Распределенная модель. В данной модели популяция также разбивается на части, но не настолько мелкие, как в клеточной модели. Поэтому данная модель также носит название «крупнозернистый» параллелизм (coarse grained parallelism). Субпопуляции, полученные после разбивки основной популяции, могут обмениваться своими хромосомами (процесс миграции). Субпопуляции часто называют островами, из-за чего сама модель часто именуется также и островным подходом или островной моделью параллелизма.



**Рис.1.** Три подхода распараллеливания. **А** – часть алгоритма, вычисляющая значения функционалов; **В** – часть алгоритма, в которой выполняются генетические алгоритмы в каждой субпопуляции в отдельной потоке; **С** – часть алгоритма, в которой выполняются генетические алгоритмы в каждой субпопуляции в отдельном потоке, при этом они могут обмениваться хромосомами

Существуют гибридные (смешанные) модели распараллеливания генетических алгоритмов [7], которые основаны на одной из трех вышеописанных моделей, однако вбирают в себя некоторые принципы остальных моделей. Например, модель ведущий-ведомый разделяют на синхронную и асинхронную. Разница заключается в наличии возможности продолжения вычислений ведущим потоком без ожидания ведомых. Распределенную модель с миграциями иногда разделяют на островную модель и модель ступенек. Разница состоит в том, что при островной модели миграции хромосом могут происходить между любыми субпопуляциями (мигранты, путешествующие по островам), а при модели ступенек миграция происходит только между соседними субпопуляциями.

Иногда процедуру миграции полностью исключают из алгоритма, но вместо этого включают понятие пересекающихся областей субпопуляций. Тогда в процессе исполнения алгоритма в операции скрещивания внутри одной субпопуляции могут попасть хромосомы из области общего доступа. Так решается проблема обмена между субпопуляциями.

Существуют так называемые грязные генетические алгоритмы, для которых также существуют параллельные реализации. Чаще всего грязный генетический алгоритм имеет два цикла: внешний и внутренний. Основная параллельная область исполняется на стадии внутреннего цикла, но внешний цикл также поддается распараллеливанию, однако это зависит от реализации алгоритма.

Самым простым подходом распараллеливания генетических алгоритмов считается подход изолированного исполнения. Алгоритм реализуется так, что на каждом отдельном процессоре или ядре полностью исполняется последовательный алгоритм. Таким образом, на многопроцессорной системе запускается сразу несколько последовательных алгоритмов, которые никак не связаны друг с другом. Соответственно, после того, как каждый процессор завершит вычисления, будет получено большее множество возможных решений, чем при работе лишь одной копии алгоритма.

### **Синтез системы управления посадкой КА на поверхность Луны**

Эксперимент проводился на модели КА, движение которого задано системой из пяти дифференциальных уравнений (1-4):

$$\begin{aligned} \dot{x}_0 &= \frac{g_E \cdot (P_{0c} + u_1) \cdot \cos(u_0 - x_1)}{x_4} - g_{Mh} \cdot \cos x_1 \\ \dot{x}_1 &= \frac{g_E \cdot (P_{0c} + u_1) \cdot \sin(u_0 - x_1)}{x_4} - g_{Mh} \cdot \sin x_1 \\ \dot{x}_2 &= \frac{x_0 \cdot \cos x_1}{1000} \end{aligned} \quad , \quad (11)$$

$$\begin{aligned} \dot{x}_3 &= \frac{x_0 \cdot \sin x_1}{(R_M + x_2) \cdot 1000} \\ \dot{x}_4 &= -\frac{(P_{0c} + u_1)}{P_{udc}} \\ g_{Mh} &= g_M \cdot \left( \frac{R_M}{R_M + x_2} \right)^2, \end{aligned} \quad (12)$$

где  $x_0$  – скорость движения КА в текущий момент времени,  $x_1$  – угол наклона траектории КА в текущий момент времени,  $x_2$  – высота полета КА относительно поверхности Луны в текущий момент времени,  $x_3$  – дальность полета КА на текущий момент времени,  $x_4$  – масса КА с учетом топлива,  $g_E$  – значение ускорения свободного падения на Земле,  $g_M$  – значение ускорения свободного падения на поверхности Луны,  $g_{Mh}$  – значение ускорения свободного падения на некоторой высоте над поверхностью Луны,  $P_{0c}$  – номинальная тяга двигателя КА (кг),  $P_{udc}$  – удельная тяга двигателя КА (с),  $R_M$  – радиус Луны;  $u_0$  – управление углом наклона траектории,  $u_0(x_1, x_2) \in [0; \pi]$ ,  $u_1$  – управление тягой двигателя КА,  $u_1(x_1, x_2) \in [-80; 80]$ .

Начальные условия заданы следующим образом:

**Таблица 1**

Начальные условия

Угол наклона траектории	Начальная масса с учетом топлива	Начальная скорость	Начальная высота	Начальное отклонение угловой дальности
$\theta^-(0) = -1.65$ рад	$m(0) = 940$ кг	$V(0) = 1689$ м/с	$h^-(0) = 17$ км	$\varphi(0) = 0$ рад
$\theta^+(0) = -1.55$ рад			$h^+(0) = 20$ км	

Терминальные условия заданы следующим образом:

- Конечная скорость: 0 м/с;
- Конечная высота: 1.5 км.

Функционал качества и функционал цели управления для данной системы заданы в следующем виде:

$$J_1 = \sum_{i=1}^P \left( |h_f - h(t_f)| + v_1 \cdot |\varphi - \varphi(t_f)| \right) \rightarrow 0, \quad (13)$$

$$J_2 = \sum_1^k \left( |V_f - V(t_f)| + v_2 \cdot |\varphi_f - \varphi(t_f)| \right) \rightarrow 0, \quad (14)$$

где  $v_1 = 0.1$ ,  $v_2 = 0.1$  весовые коэффициенты. В отличие от [2] в данном эксперименте использовался функционал с учетом дальности отклонения от планируемой точки посадки.

Для данного эксперимента в качестве неопределенных параметров выбраны начальная высота, с которой начинается процесс посадки и начальный угол наклона КА. Начальная

высота рассчитывается как точка перицентра Луны (периселений), но для обеспечения робастности системы рассматривается конечное множество точек, в котором может находиться КА. Начальный угол наклона КА может также отличаться от заданного и, соответственно, задается конечным множеством точек:

$$y_1 = \{y_1^{(1)}(t_0), \dots, y_1^{(l)}(t_0)\} \Leftrightarrow \{h^{(1)}(t_0), \dots, h^{(l)}(t_0)\}, \quad (15)$$

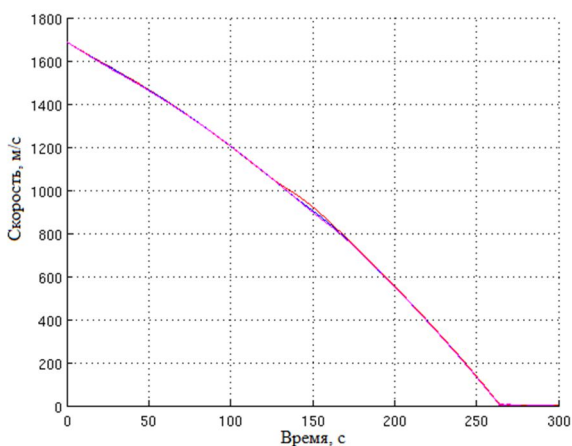
$$y_2 = \{y_2^{(1)}(t_0), \dots, y_2^{(l)}(t_0)\} \Leftrightarrow \{\theta^{(1)}(t_0), \dots, \theta^{(l)}(t_0)\}, \quad (16)$$

где  $y_1$  – отклонение по начальной высоте,  $y_2$  – отклонение по начальному углу. Величина  $l$  – выбирается в зависимости от величины интервала  $|h^{(1)}(t_0) - h^{(l)}(t_0)|$  или  $|\theta^{(1)}(t_0), \dots, \theta^{(l)}(t_0)|$  и от необходимой степени робастности по определенному параметру. Однако необходимо учесть, что высокая дискретность данных интервалов влечет за собой увеличение времени расчетов, так как расчеты функционалов производятся для каждой комбинации неопределенных параметров. Таким образом, для случая, когда  $P = 2$  и  $l = 2$ , функционалы будут рассчитаны  $P \times l = 4$  раза.

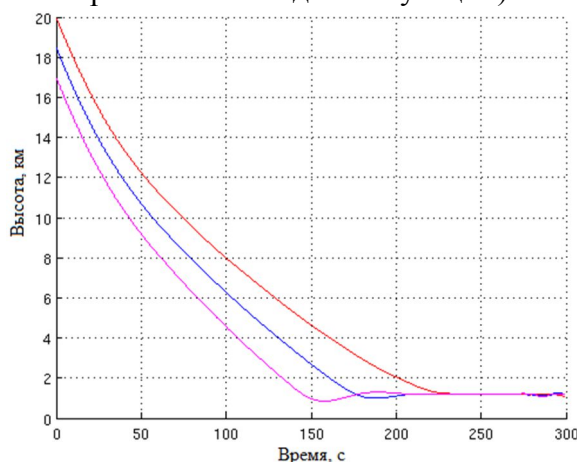
В данном эксперименте применяется комбинация модели ведущий-ведомый и островной модели распараллеливания. Выбор модели ведущий-ведомый обуславливается необходимостью вычислять функционалы для нескольких точек из области неопределенных параметров. Выбор островной модели обуславливается содержанием большого количества хромосом в каждой популяции. Для достижения большей эффективности распараллеливания применяется комбинация данных моделей. Перед началом работы алгоритма исследователю предлагается выбрать количество процессоров, используемых на распределение по островам. Учитывая данное значение, программа автоматически вычисляет количество оставшихся процессоров, которое используется для вычисления функционалов. Также перед началом работы исследователь задает количество поколений, пар в поколении, эпох (определяет размер цикла поколений, после которого происходит отбор лучшего решения от каждого острова). По прохождении эпохи каждый остров принимает отобранную лучшую хромосому в качестве базисной, и дальнейшая работа происходит именно с ней (миграция хромосом). Каждый остров имеет в своем распоряжении определенное количество процессоров, которые используются для расчета функционалов. Параллельность алгоритма расчета функционалов заключается в независимом расчете значения функционала для каждой точки из множества точек каждого неопределенного параметра. Работа организована по алгоритму, при котором поток, освободившись от вычисления функционала при определенной точке неопределенных параметров, тут же переходит на следующую точку. Соответственно, в таком режиме запускаются сразу несколько потоков. Каждое полученное значение функционала автоматически суммируется к значению функционала для каждой хромосомы. Таким

образом, такой процесс заметно ускоряет процедуру расчета функционалов. Параметры генетического алгоритма выглядят следующим образом:

- размер эпохи (количество поколений, после которых происходит обмен): 25;
- количество пар в поколении (количество сгенерированных пар за поколения): 128;
- общее количество поколений: 500;
- размерность каждой популяции (количество хромосом в каждой популяции): 1000;



**Рис.2.** Изменение скорости.



**Рис.3.** Изменение высоты при разных НУ.

На рис. 2 и рис. 3 показаны графики изменения скорости и высоты при разных начальных условиях:  $h^{(1)}(0) = 17$  км,  $h^{(2)}(0) = 18.5$  км,  $h^{(3)}(0) = 20$  км. Таким образом, КА выходит на необходимую высоту с разных начальных условий и достигает на ней нулевую скорость.

При тех же параметрах алгоритма был проеден сравнительный анализ времени вычислений, результаты которых представлены ниже.



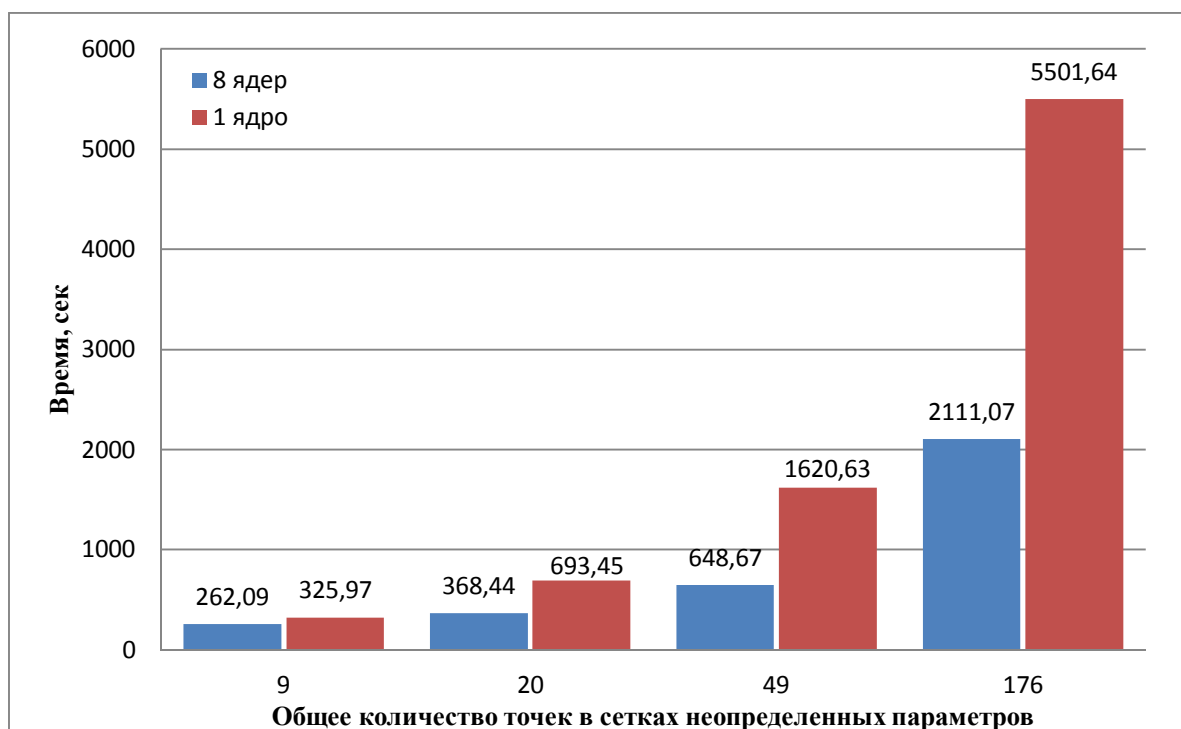
**Рис.4.** Зависимость времени от количества ведомых ядер

На рис. 4 показан график зависимости времени выполнения алгоритма от количества ведомых ядер (выделенных на вычисление функционалов) при одних и тех же начальных параметрах. По графику видно, что при увеличении количества ведомых ядер время расчета уменьшается, но лишь до определенного момента. В частности, при выделении 16 ядер на вычисление функционалов, время начинает расти, из-за выделения большего времени на создание и уничтожение потоков.

Для проведения следующего сравнительного анализа времени использовались следующие параметры генетического алгоритма и распределения ядер:



- размер эпохи (количество поколений, после которых происходит обмен): 10;
- количество пар в поколении (количество сгенерированных пар за поколения): 32;
- общее количество поколений: 16;
- размерность каждой популяции (количество хромосом в каждой популяции): 32;



**Рис.5.** Зависимость времени от размерности сетки неопределенных параметров и количества ведомых ядер.

На рис. 5 показан график зависимости времени от размерности сетки неопределенных параметров и количества ведомых ядер (выделенных на вычисление функционалов). По графику видно, что разница между временем однопоточного и многопоточного вычислений растет с увеличением размерности сетки параметров.

В следующей таблице представлено отношение времени выполнения при 1 ядре к времени при 8 ядрах:

**Таблица 2**

Отношение времени выполнения при 1 ядре к времени при 8 ядрах

Размерность сетки	$3 \times 3 = 9$	$4 \times 5 = 20$	$7 \times 7 = 49$	$16 \times 11 = 176$
Отношение	1,24	1,88	2,49	2,61

Вычисления производились на вычислительной машине HP ProLiant DL785 G5 (8 4-х ядерных процессоров по 2,2 ГГц, 48 ГБ оперативной памяти), работающей на операционной системе Linux Mageia 4 x86\_64.

Для решения задачи численного синтеза системы управления посадкой космического аппарата был разработан параллельный метод сетевого оператора, позволяющий рассмотреть большее количество возможных решений, чем при вычислении с использованием одного ядра. Соответственно, это увеличивает шансы нахождения подходящего решения.

Разработанный алгоритм позволяет балансировать нагрузку на ядра процессоров, распределяя ее между независимыми базисными потоками и вычислением функционалов для каждой точки из множества неопределенных параметров.

### Список литературы

1. Дивеев А.И. Метод сетевого оператора. М.: ВЦ РАН, 2010. 178 с.
2. Дивеев А.И. Численный метод сетевого оператора для синтеза системы управления с неопределенными начальными значениями// Известия РАН Теория и системы управления. 2012, № 2. С. 63-78.
3. Дивеев А.И., Уваров А.С., Хамадияров Д.Б. Параллельный алгоритм численного синтеза робастной системы управления на основе метода сетевого оператора, XI международный симпозиум «Интеллектуальные системы INTELS' 2014», 30.06 – 04.07 2014 г. – Москва: РУДН.
4. Карпенко А.П., Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов// Информационные технологии, 2012, №7. С.2-32.
5. Diveev A., Khamadiyarov D., Shmalko E., Sofronova E.: Intellectual Evolution Method for Synthesis of Mobile Robot Control System// IEEE Congress on Evolutionary Computation (CEC 2013), Cancún, México, P. 24-31.
6. Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, Massachusetts, London, MA: MIT Press. 1992. 819 p.
7. Luque G., Alba E., Parallel Genetic Algorithms, 367 p., ISBN 978-3-642-22084-5, 2011.
8. Nowostawski M., Riccardo P., Parallel Genetic Algorithm Taxonomy, KES'99, 1999.
9. Stender J., Parallel Genetic Algorithm: Theory & Applications, 1994, 223 p.

### Рецензенты:

Никульчев Е.В., д.т.н., профессор, проректор по научной работе НОУ ВО «Московский технологический институт», г.Москва;

Забудский Е.И., д.т.н., профессор кафедры Электропривод и электротехнологии, ФГБОУ ВПО Российский государственный аграрный университет – МСХА им. К.А. Тимирязева, г.Москва;

Петров М.Н., д.т.н., профессор, профессор кафедры «Системный анализ и исследование операций», Сибирский государственный аэрокосмический университет им. М.Ф. Решетнева, г.Красноярск.