

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РАЗДЕЛЕНИЯ ДВУХ ВЫПУКЛЫХ НЕПЕРЕСЕКАЮЩИХСЯ МНОГОГРАННИКОВ С ИСПОЛЬЗОВАНИЕМ ФЕЙЕРОВСКИХ ОТОБРАЖЕНИЙ

А.В. Ершова, И.М. Соколинская

1. Введение.

Задача разделения двух выпуклых непересекающихся многогранников имеет большое значение теоретического и прикладного характера в распознавании образов, включающем задачи дискриминации, классификации и др. [1]. Задача сильной отделимости может быть решена в ходе итерационного процесса, использующего операцию проектирования. Однако на практике применение этого метода существенно ограничивается тем, что далеко не всегда удастся построить конструктивную формулу для вычисления проекции точки на выпуклое множество. Поэтому целесообразно заменить операцию проектирования последовательностью фейеровских отображений [2]. Указанный метод был предложен И.И. Ереминым в работе [3].

Фейеровские методы относятся к итерационным методам проекционного типа, применяемым для решения систем линейных неравенств и задач линейного программирования. Их конструкция, то есть конструкция соответствующих фейеровских операторов, базируется на той или иной суперпозиции элементарных проектирований, а именно – проектирований на полупространства. Проектирование на полупространство, задаваемое линейным неравенством, является тем элементарным «кирпичиком», из совокупности которых и конструируется процедура генерации последовательности $\{x_k\}$, решающей в пределе задачу: будь то просто система линейных неравенств или задача линейного программирования. Этот класс методов интересен с разных точек зрения, в частности, с точки зрения использования их для нестационарного (эволюционного) моделирования [4].

Фейером было введено понятие сходимости последовательности x_k , $k=0,1,2,\dots$, относительно множества $M \subset \mathbb{R}^n$ в смысле: $\|x_k - y\| < \|x_k - x\|$ для любых k и $y \in M$.

В данной работе строится и исследуется итерационный алгоритм **F** решения задачи сильной отделимости, основанный на методе И.И. Еремина, рассматриваются теоретические и реализационные аспекты алгоритма **F**. Описывается параллельная версия данного алгоритма и приводятся результаты вычислительных экспериментов на суперкомпьютере «СКИФ-Урал» (с кластерной архитектурой).

2. Теоретический анализ.

Сформулируем задачу сильной отделимости [1]. Пусть даны два выпуклых непересекающихся многогранника $M \subset \mathbb{R}^n$ и $N \subset \mathbb{R}^n$, заданные системами линейных неравенств:

$$\begin{aligned} M &= \{x \mid AX \leq b\} \neq \emptyset, \\ N &= \{x \mid BX \leq d\} \neq \emptyset, \\ M \cap N &= \emptyset. \end{aligned} \quad (1)$$

Задача сильной отделимости – это задача нахождения слоя наибольшей толщины, разделяющего M и N . Сильная отделимость, по существу, эквивалентна задаче отыскания расстояния между M и N в смысле метрики

$$p(M, N) = \min \{ \|x - y\| \mid x \in M, y \in N \}. \quad (2)$$

Проведем касательные гиперплоскости через $x \in M$ и $y \in N$, для которых выполняется условие (2), так, чтобы эти гиперплоскости были параллельны. Тогда перпендикуляр, опущенный с одной гиперплоскости на другую, и будет определять слой наибольшей толщины, разделяющий многогранники M и N .

Задача сильной отделимости может быть решена с помощью итерационного алгоритма на основе фейеровских отображений. Дадим определение фейеровского отображения [4].

Пусть $\varphi \in \{\mathbb{R}^n \rightarrow \mathbb{R}^n\}$. Отображение φ называется *M-фейеровским*, если $\varphi(y) = y$, $\forall y \in M$; $\|\varphi(x) - y\| < \|x - y\|$, $\forall y \in M$, $\forall x \notin M$.

Последовательность $\{x_k\} \in \mathbb{R}^n$, $\{x_k\} \cap M = \emptyset$ называется *M-фейеровской*, если $\|x_{k+1} - y\| < \|x_k - y\|$, $\forall k$, $\forall y \in M$.

Рассмотрим фейеровское отображение, приведенное в монографии [4].

Пусть в \mathbb{R}^n задана конечная система линейных неравенств

$$Ax \leq b: l_j(x) = (a_j, x) - b_j \leq 0, j = 1, \dots, m, \quad (3)$$

где $a_j \neq 0$ для любого j . Определим $l_j^{plus}(x)$ следующим образом:

$$l_j^{plus}(x) = \max\{l_j(x), 0\}, j=1, \dots, m. \quad (4)$$

Тогда *фейеровское отображение* имеет вид:

$$\varphi(x) = x - \sum_{j=1}^m \alpha_j \lambda_j \frac{l_j^{plus}(x)}{\|a_j\|^2} a_j \quad (5)$$

для любой системы положительных коэффициентов $\{\alpha_j > 0\}$, $j=1, \dots, m$, таких, что $\sum_{j=1}^m \alpha_j = 1$ и коэффициентов релаксации $0 < \lambda_j < 2$.

Пусть M – выпуклое замкнутое множество, заданное системой линейных неравенств, φ – непрерывное M -фейеровское отображение. Пусть $\{x_i\}$ – последовательность, порождаемая отображением φ . Пусть $x_0 \in \mathbb{R}^n$ – произвольное начальное приближение. Имеет место следующая фундаментальная теорема сходимости [2].

Теорема. Процесс $\{x_{i+1} \in \varphi(x_i)\}$, $\forall i$, порожденный любым фейеровским отображением при φ любым начальном x_0 , сходится к некоторому решению x' системы выпуклых неравенств: $\forall i \{x_{i+1} \in \varphi(x_i)\} \rightarrow x' \in M$.

Рассмотрим, как фейеровское отображение может быть использовано для решения задачи сильной отделимости двух выпуклых непересекающихся многогранников.

3. Методика решения задачи сильной отделимости.

Определим алгоритм **F** разделения выпуклых многогранников с помощью фейеровских отображений [5].

Алгоритм F. Пусть даны два выпуклых непересекающихся многогранника $M \subset \mathbb{R}^n$ и $N \subset \mathbb{R}^n$, заданные системами линейных неравенств (1). Пусть φ_M и φ_N – непрерывные M и N -фейеровские отображения. Пусть задано произвольное начальное приближение $z_0 \in \mathbb{R}^n$. Алгоритм состоит из следующих шагов:

Шаг 0. $k := 0$.

Шаг 1. Найдем точку x_k как результат многократного последовательного применения отображения φ_M к точке z_k : $x_k := \lim_{u \rightarrow \infty} \varphi_M^u(z_k)$.

Шаг 2. Найдем точку y_k как результат многократного последовательного применения отображения φ_N к точке z_k : $y_k := \lim_{u \rightarrow \infty} \varphi_N^u(z_k)$.

Шаг 3. Положим: $z_{k+1} := \frac{x_k + y_k}{2}$.

Шаг 4. $k := k + 1$.

Шаг 5. Перейти на шаг 1.

Применение алгоритма **F** для нахождения слоя наибольшей толщины, разделяющего два выпуклых многогранника M и N , заключается в последовательном вычислении точек $x_k \in M$ и $y_k \in N$. Процесс завершается при выполнении следующего критерия завершения: $\max\{\|x_k - x_{k-1}\|, \|y_k - y_{k-1}\|\} < \varepsilon$, для некоторого фиксированного $\varepsilon > 0$.

Известно [4], что если в алгоритме **F** точки x_k и y_k получать как проекции точек x_{k-1} и y_{k-1} на многогранники M и N соответственно, то для случая выпуклых многогранников процесс будет сходиться к циклу неподвижности. При этом в пределе мы получим отрезок, определяющий слой наибольшей толщины. И.И. Ереминым была высказана гипотеза, что алгоритм **F**, использующий фейеровские приближения также будет сходиться к циклу неподвижности.

Авторами настоящей статьи была выполнена программная реализация алгоритма **F** на языке C++ (тексты программы доступны в Интернет по адресу <http://life.susu.ru/discr/>). С помощью этой программы было проведено экспериментальное исследование поведения алгоритма **F** для различных выпуклых многогранников. Во всех случаях наблюдалась сходимость процесса к циклу неподвижности. При этом в результате мы получали отрезок, определяющий слой наибольшей толщины с заданной точностью. Проведенные вычислительные эксперименты по исследованию поведения алгоритма **F** на масштабируемых модельных и случайных задачах подтвердили эффективность предложенного подхода.

4. Методика параллельной реализации алгоритма.

При решении практических задач экономико-математического моделирования часто встречаются задачи с большим количеством переменных и ограничений. Размер типичной средней задачи может составлять 20 000 переменных и 5 000 ограничений [6]. В отдельных случаях количество переменных может превышать 100 000, а количество ограничений – 20 000. Подобные задачи при решении требуют значительных вычислительных мощностей. В связи с этим актуальной является задача разработки параллельной версии алгоритма сильной отделимости.

Рассмотрим метод построения параллельного алгоритма решения данной задачи, в основе которого лежит подход разбиения на подвекторы. Пусть $\varphi(x)$ – произвольное фейеровское отображение относительно множества M , $\varphi \in \{\mathbb{R}^n \rightarrow \mathbb{R}^n\}$. Разобьем вектор $x \in \mathbb{R}^n$ на r подвекторов $x = [x_1, \dots, x_r]$, где $x_i \in \mathbb{R}^{n_i}$ ($i=1, \dots, r$) и $\mathbb{R}^n = \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_r}$. Обозначив через $\pi_i(x)$ проекцию x на \mathbb{R}^{n_i} , определим отображения $\varphi_i(x) = \pi_i(\varphi(x)) + \sum_{j \neq i} \pi_j(x)$ и положим

$$\bar{\varphi}(x) = \alpha \sum_{i=1}^r \pi_i(\varphi_i^K(x)), \quad 0 < \alpha < 1 \quad \text{при некотором фиксированном натуральном } K.$$

При определенных условиях отображение $\bar{\varphi}$ будет M -фейеровским. Таким образом, на базе одного фейеровского отображения мы сконструировали другое, обладающее большим ресурсом параллелизма. Действительно, значения $\varphi_i^K(x)$ могут вычисляться независимо друг от друга для различных $i=1, \dots, r$. При этом мы получаем две степени свободы для регулировки баланса загрузки процессорных узлов. Во-первых, увеличивая K , мы можем повышать вычислительную нагрузку на процессорный узел между двумя соседними итерациями, вычисляющими $\bar{\varphi}(x)$. Во-вторых, мы можем произвольным образом перераспределять координаты вектора x между процессорами.

При реализации параллельной версии алгоритма **F** применение фейеровского отображения K раз к каждому подвектору оформлялось в виде независимого процесса. Далее происходил обмен вычисленными $\varphi_i^K(x)$ между процессорами. Указанный подход позволяет без какой-либо модификации выполнять параллельную программу на различном количестве процессоров. Для организации обменов данными между процессами была использована система

Многогранник M

параллельного программирования MPI, основанная на передаче сообщений [7]. При реализации параллельной версии алгоритма **F** мы использовали передачу сообщений на основе односторонних коммуникаций MPI-2 [8]. Разработанная параллельная версия алгоритма **F** может использоваться без каких-либо изменений как на

$$\begin{cases} x_1 - 2x_2 \leq 0 \\ x_1 - 2x_3 \leq 0 \\ \dots \\ x_1 - 2x_n \leq 0 \\ x_1 + 2x_2 \leq 20000 \\ x_1 + 2x_3 \leq 20000 \\ \dots \\ x_1 + 2x_n \leq 20000 \\ -x_1 \leq 0 \\ -x_2 \leq 0 \\ \dots \\ -x_n \leq 0 \end{cases}$$

Многогранник N

$$\begin{cases} x_1 - 2x_2 \leq 20000 \\ x_1 - 2x_3 \leq 20000 \\ \dots \\ x_1 - 2x_n \leq 20000 \\ x_1 + 2x_2 \leq 40000 \\ x_1 + 2x_3 \leq 40000 \\ \dots \\ x_1 + 2x_n \leq 40000 \\ -x_1 \leq -20000 \\ -x_2 \leq 0 \\ \dots \\ -x_n \leq 0 \end{cases}$$

многопроцессорных системах с общей памятью, так и на системах с массовым параллелизмом, включая кластерные системы. Исходные тексты программ доступны в Интернет по адресу <http://life.susu.ru/dscr/>.

5. Вычислительные эксперименты.

Для проведения вычислительных экспериментов мы использовали сконструированную нами модельную задачу *Model-n*. Для такого типа задач можно легко аналитически вычислить точное значение толщины максимального разделяющего слоя. Поэтому они хорошо подходят для проверки корректности алгоритма и изучения его масштабируемости. Задача *Model-n* имеет следующий вид (n задает размерность задачи):

Для всех задач *Model-n* толщина максимального разделяющего слоя равна 10 000. На Рис. 1 изображены многогранники M и N , задаваемые задачей *Model-3* ($n=3$), и итерации последовательного алгоритма **F**, приводящие к циклу неподвижности.

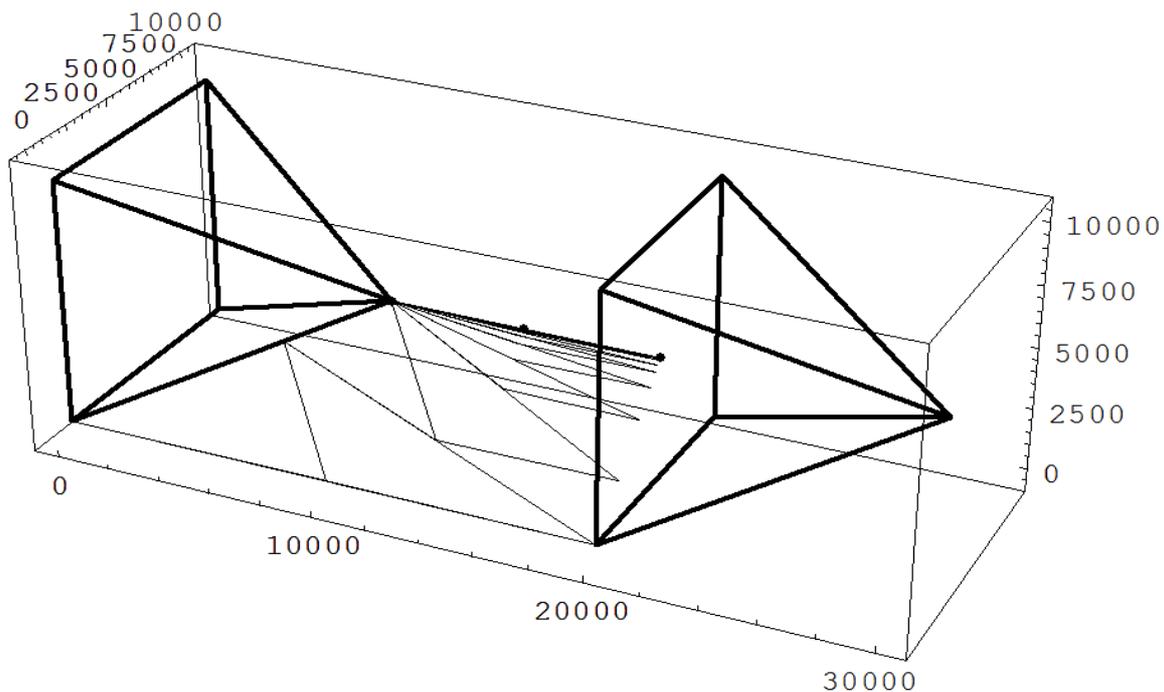


Рис. 1. Итерации последовательного алгоритма F для задачи Model-3.

Для проведения экспериментов был использован суперкомпьютер «СКИФ-Урал» (с кластерной архитектурой) с характеристиками, представленными в Табл. 1.

Число лезвий blade-системы / процессоров / ядер	166 / 332 / 1328
Тип процессора	Intel Xeon E5472 (4 ядра по 3.0 GHz)
Оперативная память	1.33 TB
Дисковая память	49.29 TB
Система хранения данных	Panasas ActiveStorage 5100 (20 TB)
Тип системной сети	InfiniBand (20Gbit/s, макс. задержка 2 мкс)
Тип управляющей (вспомогательной) сети	Gigabit Ethernet
Сервисная сеть	СКИФ ServNet
Пиковая производительность	16 TFlops
Производительность на тесте LINPACK	12.2 TFlops
Операционная система	SUSE Linux Enterprise Server 10
	Windows HPC Server 2008
Система бесперебойного электропитания	APC Symmetra 160 kVA

Табл. 1. Характеристики вычислительного кластера.

В первой серии вычислительных экспериментов исследовалась зависимость количества итераций и времени решения задачи от количества независимых без обмена итераций K . Это фиксированный натуральный параметр алгоритма, обозначающий количество применений фейеровского отображения к каждому подвектору. Увеличивая K , мы можем повышать вычислительную нагрузку на процессорный узел между двумя соседними итерациями.

Результаты эксперимента представлены на Рис. 2 и Рис. 3. Эксперименты проводились для трех размерностей: $n=256$, $n=512$ и $n=1024$ при фиксированном количестве процессоров $p=128$. Значения K варьировались от 150 до 1450. Графики показывают, что количество итераций и время решения задачи для всех размерностей уменьшаются при увеличении K , принимая горизонтально-асимптотический характер при значениях K больших 1000.

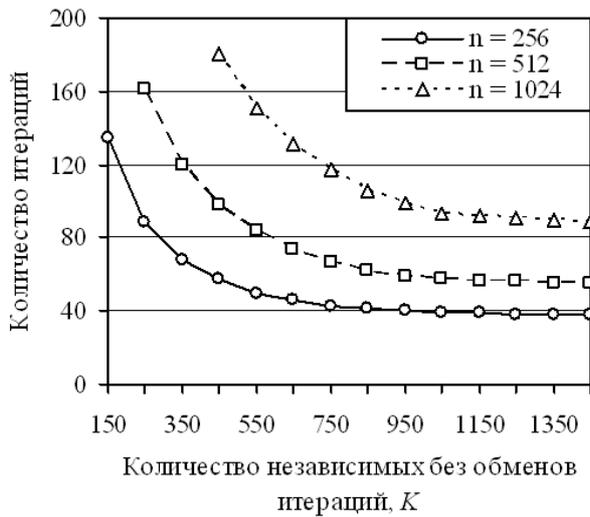


Рис. 2. Зависимость количества итераций задачи от количества независимых без обменов итераций

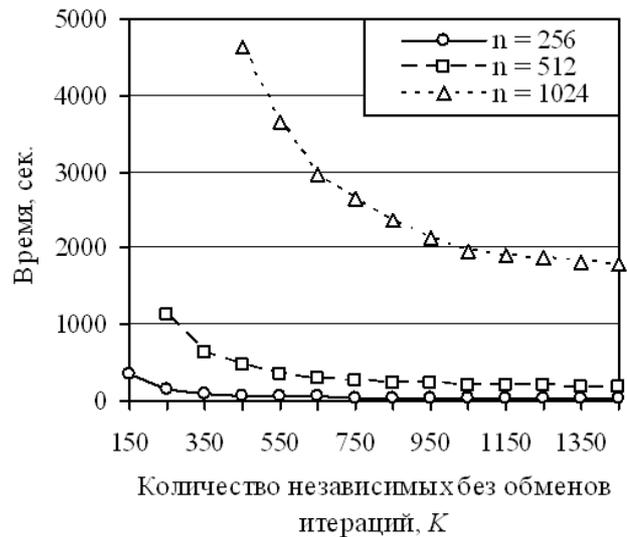


Рис. 3. Зависимость времени решений задачи от количества независимых без обменов итераций

Во второй серии вычислительных экспериментов исследовалась зависимость времени решения задачи от независимых без обмена итераций K при фиксированной размерности задачи $n=512$. При этом задача вычислялась на разном количестве процессоров: $p=64$, $p=128$ и $p=256$.

Результаты экспериментов показывают (Рис. 4), что время решения задачи стабильно уменьшается для всех трех вариантов, стабилизируясь при значениях K больших 1000.

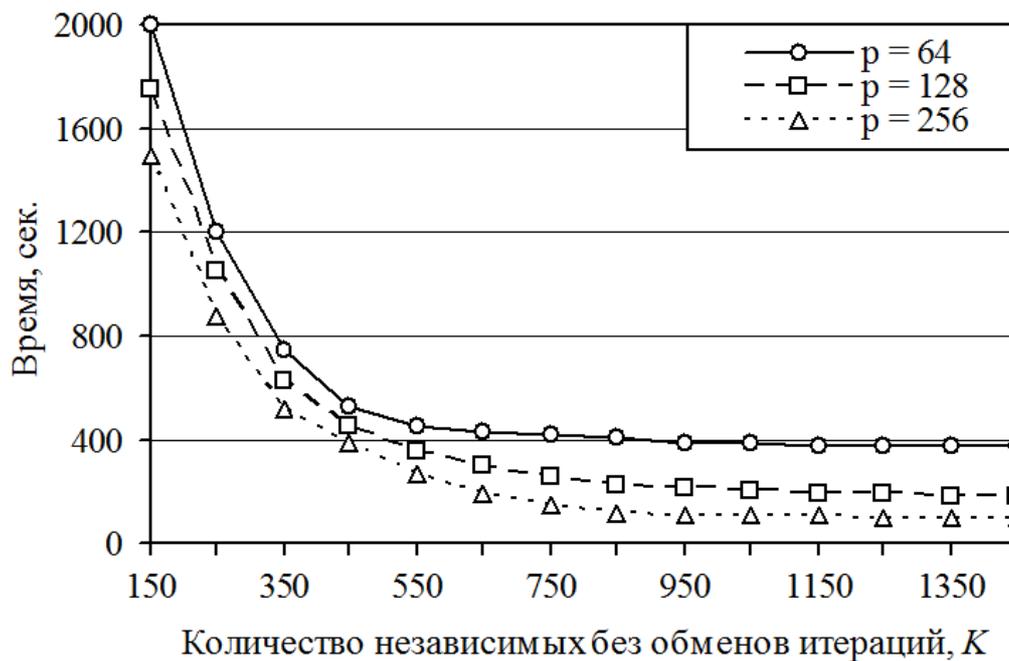


Рис. 4. Зависимость времени решения задачи от количества независимых без обменов итераций.

Проведенные эксперименты показывают, что для данной серии задач хорошим выбором является значение параметра алгоритма $K=1000$.

В третьей серии экспериментов исследовалось ускорение параллельного алгоритма. Результаты этого исследования представлены на Рис.5. Эксперименты проводились для трех размерностей: $n=256$, $n=512$ и $n=1024$ при фиксированном количестве независимых без обмена итераций $K=1000$. Ускорение вычислялось по формуле $\alpha = t_p / t_8$, где t_8 - время, затраченное на решение задачи Model-n на

8 процессорах, t_p - время, затраченное на решение этой же задачи на p процессорах. Для каждой размерности варьировалось количество процессоров, используемых для решения задачи. Мы видим, что масштабируемость существенно зависит от размерности. Чем больше размерность, тем лучше масштабируется задача.

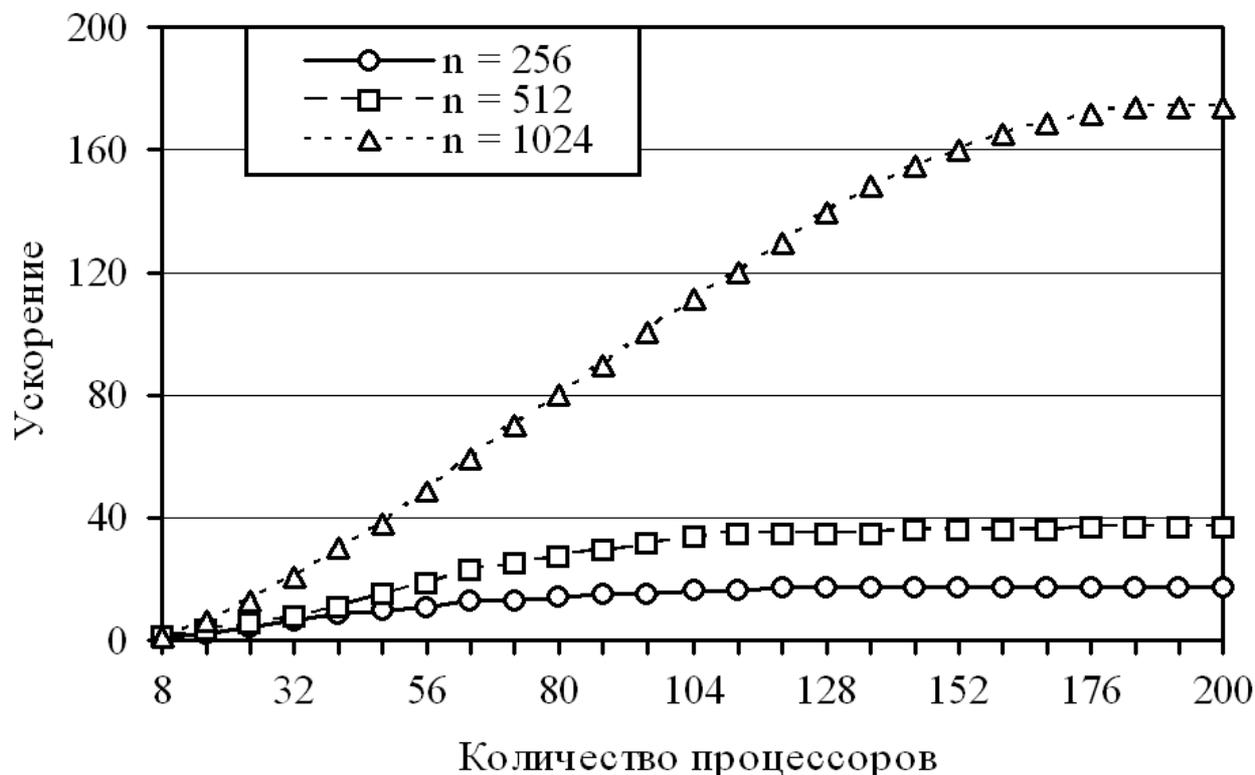


Рис.5. Ускорение для Model-n.

Таким образом, можно сделать вывод о том, что параллельный алгоритм F показывает хорошую масштабируемость на многопроцессорных системах с массовым параллелизмом при решении задач разделения двух выпуклых непересекающихся многогранников.

6. Заключение.

В данной работе был рассмотрен итерационный алгоритм F для решения задач разделения двух выпуклых непересекающихся многогранников слоем наибольшей толщины. Этот алгоритм базируется на использовании фейеровских отображений в качестве аналога операции проектирования. Предложена параллельная версия алгоритма F, которая может быть использована без каких-либо изменений как на многопроцессорных системах с общей памятью, так и на системах с массовым параллелизмом, включая кластерные системы. Проведены вычислительные эксперименты на высокопроизводительном кластере по исследованию поведения алгоритма F на масштабируемых модельных задачах *Model-n* и по исследованию масштабируемости описанной параллельной версии алгоритма F, подтверждающие эффективность предложенных подходов. Исходные тексты параллельной версии свободно доступны в Интернет по адресу: <http://life.susu.ru/discr/>.

Работа поддержана грантом РФФИ № 09-01-00546а.

ЛИТЕРАТУРА:

1. И.И. Еремин Фейеровские методы сильной отделимости выпуклых полиэдральных множеств // Известия вузов. Сер. математика. -2006. -№ 12. -С. 33-43.
2. И.И. Еремин, В.Д. Мазуров Нестационарные процессы математического программирования. -М.: Наука, 1979. -288 с.
3. И.И. Еремин Итеративная отделимость непересекающихся многогранников ограничениями // Теория управления и теория обобщенных решений уравнений Гамильтона-Якоби: Тр. Междунар. семинара, посв. 60-летию акад. А.И. Субботина. -Екатеринбург: Издательство Урал. ун-та, 2006. -Т.2. -С. 16-24.
4. И.И. Еремин Теория линейной оптимизации. -Екатеринбург: Изд-во "Екатеринбург", 1999. -312 с.

5. А.В. Ершова Алгоритм разделения двух выпуклых непересекающихся многогранников с использованием фейеровских отображений // Системы управления и информационные технологии. -2009. -№1(35). -С. 53-56.
6. J.J.H. Forrest, J.A. Tomlin Implementing the simplex method for the optimization subroutine library // IBM Systems Journal. -1992. -Vol. 31, No. 1. -P. 11-25.
7. J.J. Dongarra, S.W. Otto, M. Snir, D. Walker A message passing standard for MPP and workstations // Communications of the ACM. -1996. -Vol. 39, No. 7. -P. 84-90.
8. W. Gropp, S. Huss-Lederman, A. Lumsdaine, E. Lusk, B. Nitzberg, W. Saphir, M. Snir MPI - The Complete Reference: Volume 2, The MPI Extensions. -MIT Press, 1998.