

РАЗРАБОТКА И АНАЛИЗ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ КООПЕРАТИВНЫХ ИГР

М.Ю. Нестеренко, А.С. Кириллов

DEVELOPMENT AND ANALYSIS OF HIGH PERFORMANCE PARALLEL ALGORITHMS FOR SOLVING COOPERATIVE GAMES

M.U. Nesterenko, A.S. Kirillov

В работе рассматривается подход к решению вычислительно-сложной задачи – построение оптимальной коалиции и распределение выигрыша в кооперативной игре заданной множеством биматричных игр для n игроков с помощью решения стратегических игр и использованием параллельных вычислительных технологий.

Ключевые слова: теория игр, кооперативные игры, параллельный алгоритм.

An approach to solving computationally complex problems – construction of an optimal coalition and distribution of the profit in a cooperative game of a given set of bimatrix games for n players with solving strategic games and the use of parallel computing technologies is discussed in paper.

Keywords: game theory, cooperative games, the parallel algorithm.

1. Актуальность

Возросшие запросы производства и экономики, развитие информационных систем и технологий предъявляют все большие требования к процессу принятия решений и управления в различных областях человеческой деятельности. При создании систем поддержки принятия решений активно используется аппарат теории игр. При этом очень актуальна проблема разработки эффективных алгоритмов для поиска оптимальных стратегий в игровых моделях. На практике требуется, чтобы алгоритмы были способны обрабатывать большое количество информации в режиме реального времени.

Среди направлений теории игр особое место занимают кооперативные игры, например, в разработке оптимальных стратегий производства, безопасности систем, эффективной ценовой политики и др. В кооперативных играх игроки имеют возможность принимать совместные решения и, в некоторых случаях, перераспределять выигрыши между собой. В кооперативной игре задаются возможности и предпочтения различных групп игроков (коалиций), для которых строятся оптимальные стратегии и задаются распределения между ними суммарных выигрышей. Результатом решения кооперативной игры являются оптимальная коалиция, в которую следует вступить игрокам, их оптимальные смешанные стратегии и выигрыш коалиции. Особого внимания заслуживают задачи игрового моделирования. Примером подобной задачи может быть моделирование конкурентного рынка в различных отраслях, где игроки имеют возможность объединяться в коалиции, конкурировать за соответствующие ниши рынка.

Разработка параллельных алгоритмов для решения задач теории игр – новое направление в алгоритмической теории игр. На сегодняшний день имеются лишь единичные публикации в этой области, например работы Уиджера и Гросу для некооперативных игр [1, 2, 3]. Для решения кооперативных игр параллельные алгоритмы пока не разработаны.

Существуют различные способы решения кооперативных игр, например: решение на основе функции полезности Неймана – Моргенштерна, решение игры в виде С-ядра и в стратегиях угроз, однако эти подходы не определяют оптимальных стратегий игроков внутри коалиций. А решение игры в виде С-ядра дает целое множество допустимых решений, не определяя оптимального решения. Кроме того, эти подходы рассматривают кооперативные игры на основе характеристических функций и опускают способы получения значений характеристической функции, что само по себе является отдельной и трудоемкой задачей.

Как видно, решение задач кооперативных игр процесс трудоемкий, и с увеличением числа игроков, трудоемкость решения задачи только повышается [4].

2. Понятие кооперативной игры

Кооперативной игрой [4, 5, 6, 7, 8, 9] n лиц называется игра вида $\Gamma = (N, v)$, где $N = \{1, 2, \dots, n\}$ – множество игроков, которым разрешено вести переговоры и объединяться в коалиции, а $v : 2^N \rightarrow R$ – характеристическая функция, определяющая наибольший уверенно получаемый выигрыш $v(K)$ для каждой возможной коалиции $K \subseteq N$ [7].

3. Алгоритм решения кооперативной игры, заданной биматричными играми

Обозначим через N множество всех игроков, $N = \{1, 2, \dots, n\}$, а через K – любое его подмножество. Пусть игроки из K договариваются между собой о совместных действиях и, таким образом, образуют одну коалицию. Образовав коалицию, множество игроков K действует как один игрок против остальных игроков, и выигрыш этой коалиции зависит от применяемых стратегий каждым из n игроков.

Функция v , ставящая в соответствие каждой коалиции K наибольший, уверенно получаемый ею выигрыш $v(K)$, называется характеристической функцией игры [4]. Всего значений характеристической функции может быть 2^n , по количеству коалиций, причем коалиция с номером 0 является пустой коалицией, т.е. состоящей из пустого множества игроков, и ее выигрыш по условию всегда равен нулю.

В большинстве работ рассматриваются только классические кооперативные игры, при этом в них не приводится способ получения значений характеристических функций. При этом считается, что значения характеристических функции заранее известны, либо их можно легко получить на основании какой-либо информации об игроках.

Но, в общем случае, это не так. В практике принятия решений заранее неизвестны значения характеристических функций, имеется только предполагаемое множество стратегий игроков, т.е. их возможные действия, которые приводят к получению прибыли или приводят к убыткам. И в таких случаях получение значений характеристических функций становится сложной и трудоемкой задачей.

Одним из способов решения этой задачи является использование биматричных игр. При этом кооперативная игра задается множеством биматричных игр между каждой парой игроков. Данный подход позволяет вычислить значения характеристической функции.

Назовем систему $\Gamma = (N, B)$, состоящую из множества игроков N и набора биматричных игр $B = \{B_{xy}, x \in N, y \in N, x \neq y\}$, кооперативной игрой на основе биматричных игр. Для решения такой игры необходимо иметь или составить исходный набор биматричных игр.

Легко увидеть, набор биматричных игр будет состоять из $n \cdot (n - 1)$ матриц, как набор размещений из n по 2.

Оговоримся сразу, что количество стратегий всех игроков одинаково и равно k , в противном случае можно добавить пустые «фиктивные» стратегии.

Набор биматричных игр B строится исходя из исходных данных задачи для каждой пары игроков $x, y \in N$ в виде матрицы размера $k \times k$:

$$B_{xy} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \dots & b_{kk} \end{pmatrix},$$

где элементы матрицы $b_{ij}(i, j = \overline{1, k})$ – выигрыши игрока x , при выборе игроком x стратегии i , в игре с игроком y , при выборе игроком y стратегии j .

Согласно [4,8,10], характеристическая функция для коалиции S определяет максимальный уверенно получаемый ею выигрыш. Следовательно, для вычисления этого выигрыша целесообразно считать оставшуюся часть игроков, не входящих в S , в первом приближении антагонистически настроенными и действующими совместно. В результате чего получаем матричную игру A_{SP} коалиции $S \subseteq N$ и антикоалиции $P = N - S = \{p \in N, p \notin S\}$, которая строится на основании биматричных игр.

При этом выигрыш коалиции S в этой матричной игре будет равен значению характеристической функции для данной коалиции S . При решении игры в смешанных стратегиях $\underline{v} = \bar{v} = v(S) = |A_{SP}|$, где $|A_{SP}|$ – решение матричной игры A_{SP} в смешанных стратегиях.

Матрицы матричной игры A_{SP} коалиции S в игре с остальными участниками будет иметь размер:

$$k^m \times k^{n-m},$$

где k – количество стратегий одного игрока, а m – количество игроков в текущей коалиции. Такой размер матриц обусловлен тем, что матрицы A_{SP} строятся на основании комбинированных стратегий коалиций S . Матрицы матричной игры будут иметь следующий вид:

$$A_{SP} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k^{n-m}} \\ a_{21} & a_{22} & \dots & a_{2k^{n-m}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k^m 1} & a_{k^m 2} & \dots & a_{k^m k^{n-m}} \end{pmatrix}, \text{ при } 0 < m \leq n,$$

где a_{ij} – выигрыш игроков коалиции S в игре с антикоалицией P , причем i и j – комбинированные стратегии коалиций. Элемент a_{ij} вычисляется по следующему алгоритму:

1) из номера комбинированной стратегии коалиции S , равного i , отнимают 1 и переводят из десятичной системы счисления в систему счисления по основанию k (количество стратегий каждого игрока), записанную в обратной последовательности, обозначим такую систему счисления как система счисления по основанию kt , получившееся число имеет вид:

$$(i - 1)_{10} = (c_m c_{m-1} \dots c_1)_{kt},$$

где $c_v + 1 (v = \overline{1, m})$ – стратегия v -го игрока коалиции S .

2) из номера комбинированной стратегии антикоалиции P равного j отнимают 1 и переводят из десятичной системы счисления в систему счисления по основанию k , получившееся число имеет вид:

$$(j - 1)_{10} = (c_{n-m} c_{n-m-1} \dots c_1)_{kt},$$

где $c_w + 1 (w = \overline{1, n - m})$ – стратегия w -го игрока антикоалиции P .

3) на основании найденных стратегий каждого игрока для игровой ситуации (i, j) находят значение выигрыша коалиции S при игре против антикоалиции P :

$$a_{ij} = \sum_{b_{c_x c_y} \in B_{xy}, x, y \in S, x \neq y} b_{c_x c_y} + \sum_{b_{c_x c_y} \in B_{xy}, x \in S, y \in P} b_{c_x c_y}.$$

Для решения матричной игры, в основном, используют задачу линейного программирования [4, 8].

В результате решения матричной игры получаем значение v , которое будет равно значению характеристической функции $v(S)$ для коалиции S . После получения характеристической функции необходимо проверить игру на существенность, для чего необходимо проверить, удовлетворяет ли полученная характеристическая функция свойству супераддитивности: $\forall K, S \subseteq NK \cap S = \emptyset \Rightarrow v(K \cup S) > v(K) + v(S)$.

Если характеристическая функция удовлетворяет этому свойству, то такая игра является существенной кооперативной игрой, и, соответственно, экономически устойчивой, и для нее имеет смысл в дальнейшем производить поиск дележа.

Существует несколько способов решения кооперативных игр и получения дележа: С-ядро, решение Неймана – Моргенштерна и вектор Шепли.

С-ядро [4, 5] представляет собой множество недоминируемых дележей кооперативной игры. Оно устойчиво к отклонениям любой коалиции игроков.

Однако множество может быть пустым, тогда соответствующая кооперативная игра не будет иметь решения в рамках понятия С-ядра.

С-ядро в кооперативной теории представляет собой основной множественный принцип оптимальности. Однако остается проблема выбора подходящего дележа в каждом конкретном случае. Также С-ядро на практике часто оказывается пустым [4], поэтому были предложены другие методы решения кооперативной игры.

НМ-решение [5, 6]. НМ-решение кооперативной игры – результат работы ученых Дж. фон Неймана и О. Моргенштерна [10]. Они предложили в качестве решения рассматривать множество подмножеств дележей, обладающих определенными свойствами, каждое из них носит название НМ-решения.

НМ-решение обладает внутренней и внешней устойчивостью. Внутренняя устойчивость обеспечивается равноправностью его дележей, т.е. невозможностью найти пару дележей, чтобы один из них доминировал над другим. Смысл внешней устойчивости в том, что произвольный дележ, не принадлежащий НМ-решению, доминируется дележом из НМ-решения.

Заметим, что если игра сбалансирована, то НМ-решение содержит в себе в качестве подмножества С-ядро [5].

Существуют игры, не имеющие НМ-решения, но чаще всего кооперативные игры имеют огромное количество НМ-решений. Трудоемкость нахождения ограничивает их практическое использование.

Также имеет место проблема совместного выбора участниками одного из нескольких НМ-решений, в рамках которого будет определяться вектор распределения выигрыша. Это ограничивает возможность применения данного подхода.

Вектор Шепли [4, 5, 7] – принцип оптимальности распределения выигрыша между игроками в задачах теории кооперативных игр. Представляет собой вектор дележа $\varphi[v]$, в котором выигрыш каждого игрока равен его среднему вкладу в благосостояние коалиции при определенном механизме ее формирования.

Одним из недостатков вектора Шепли является то, что он не является селектором С-ядра в общем случае, хотя обеспечивает справедливое (в смысле эффективного вклада участников) распределение выигрыша.

Для нахождения дележа используется принцип оптимальности распределения выигрыша между игроками в задачах теории кооперативных игр, называемый вектором Шепли.

4. Параллельный алгоритм решения кооперативной игры, заданной биматричными играми

На основании вышесказанного нами разработан параллельный алгоритм решения кооперативной игры, заданной набором биматричных игр, с использованием технологии MPI.

Суть разработанного параллельного алгоритма состоит в следующем.

Каждым процессом выполняется считывание матриц, задающих биматричные игры, причем биматричные игры задаются в виде набора текстовых файлов, при этом нумерация файлов идет от 1 до $n \cdot (n - 1)$: номер 1 соответствует набору игроков (1, 2), номер 2 соответствует (1, 3) и т.д. до $n \cdot (n - 1)$, который соответствует набору $(n, n - 1)$.

После прочтения исходных данных каждый процесс вычисляет свой диапазон коалиций, с которыми он будет в дальнейшем работать. Для каждой коалиции из своего диапазона процесс формирует матричную игру по алгоритму, описанному выше, и решает полученную матричную игру с помощью симплекс-метода. Номера игроков, входящих в коалицию, определяются по номерам позиций единичных элементов двоичного представления порядкового номера коалиции.

Как только все процессы закончат расчет на своих диапазонах, результаты отсылаются нулевому процессу, который затем проверяет игру на существенность, и, если игра существенна, то вычисляется дележ по Шепли. Затем все результаты записывается в выходной файл.

Схема работы параллельного алгоритма представлена на рис. 1.

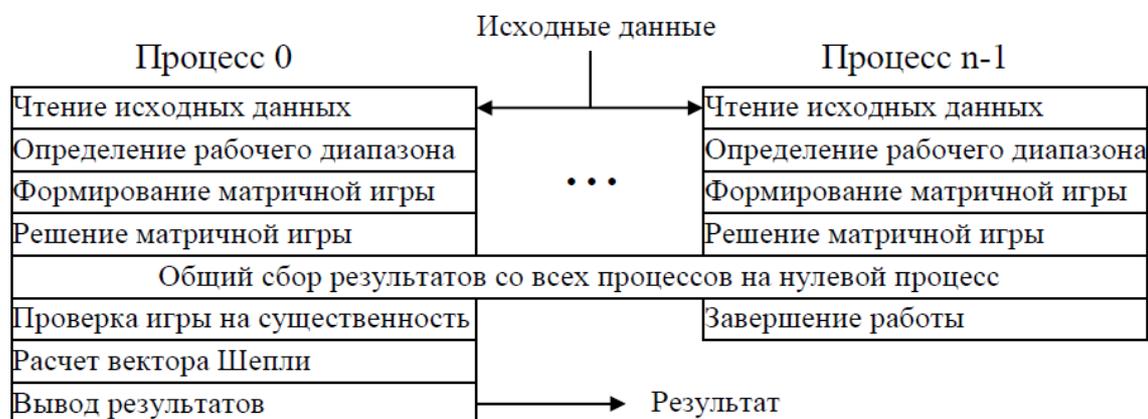


Рис. 1. Схема работы параллельного алгоритма

Тестирование разработанного параллельного алгоритма производилось на вычислительном кластере Оренбургского Государственного Университета, состоящего из следующих компонентов:

- головного узла (мастера) на базе двух процессоров Intel Xeon 3.2 ГГц, оперативной памятью 2 Гб и дисковым пространством 750 Гбайт;
- шести двухпроцессорных вычислительных узлов на базе Intel Xeon 3.2 ГГц, оперативной памятью 2 Гб и дисковым пространством 80 Гбайт (всего 12 вычислительных ядер);

– четырех двухпроцессорных вычислительных узлов на базе четырехъядерных процессоров Intel Xeon 5440, оперативной памятью 16 Гб и дисковым пространством 320 Гбайт (всего 32 вычислительных ядра).

Все узлы объединены вычислительной сетью Infiniband со скоростью 10 Гбит/с и управляющей сетью Ethernet со скоростью 1 Гбит/с. Суммарная пиковая производительность кластера составляет 451,6 GFLOP/s.

На кластере установлено следующее системное программное обеспечение:

- операционная система SUSE Linux 11;
- дистрибутивы MPI: OpenMPI 1.3.2, Mvapich2-0.9.8p3;
- компиляторы: GNU GCC 4.3.2, Intel C/C++/Fortran 11.0;
- менеджер ресурсов Torque-2.3.6.

Для запуска тестовых задач использовалось четыре двухпроцессорных вычислительных узла на базе процессоров Intel Xeon 5440, на каждом узле использовалось только два ядра. Такая методика тестирования была связана с тем, что решаемая задача кооперативных игр очень требовательна к объему оперативной памяти, и для каждого экземпляра задачи выделялось до 6 Гб памяти. Тестирование разработанного алгоритма производилось после-

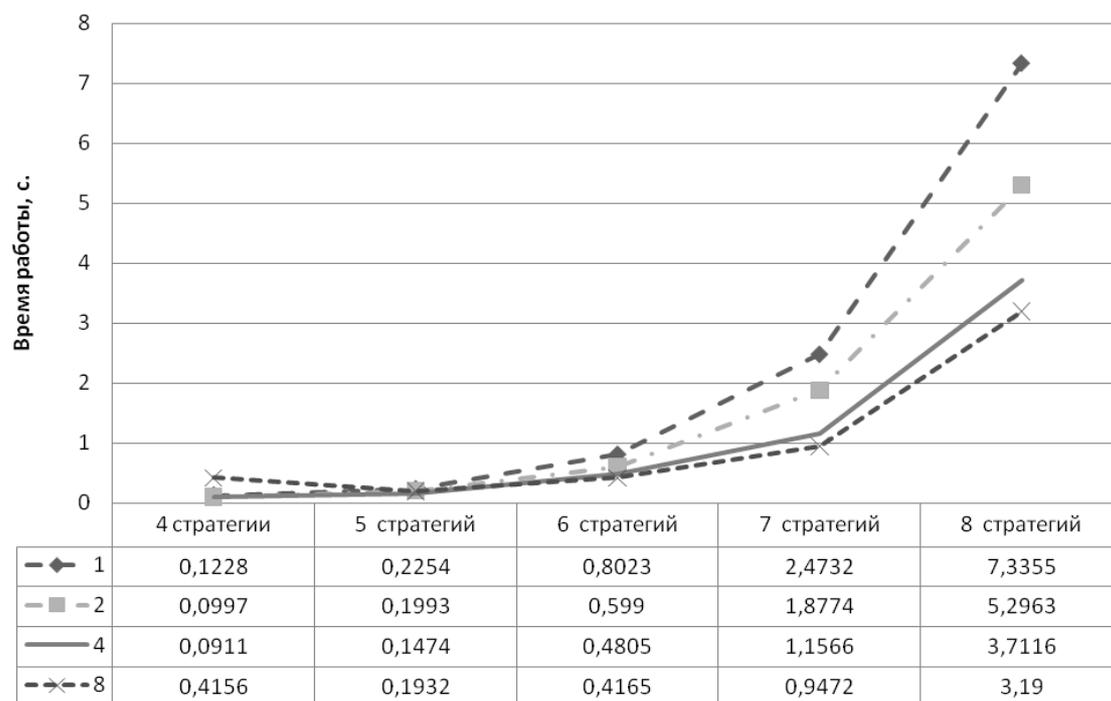


Рис. 2. График зависимости времени работы программы от количества процессоров и количества стратегий для пяти игроков

довательно для 4, 5, 6, 7 и 8 игроков, при этом каждый игрок имел 4, 5, 6, 7 и 8 стратегий. Для каждой задачи формировался произвольный набор исходных данных. Полученный набор исходных данных затем решался последовательно на 1, 2, 4 и 8 процессорах, при этом замерялось время выполнения каждой задачи.

Как видно из графиков, представленных на рис. 2, 3, 4 и 5, не удалось добиться приемлемого масштабирования разработанного параллельного алгоритма.

Это, прежде всего, связано с тем, что формирующиеся матричные игры требуют различного времени решения и нет сбалансированного метода распределения матричных игр по

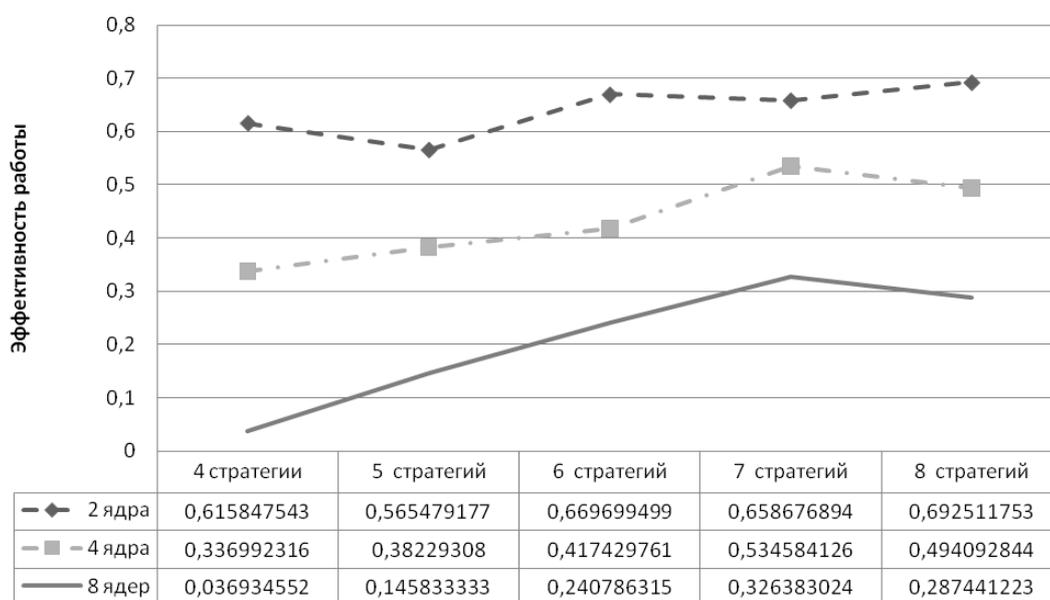


Рис. 3. График эффективности работы параллельной программы в зависимости от количества процессоров и количества стратегий для пяти игроков

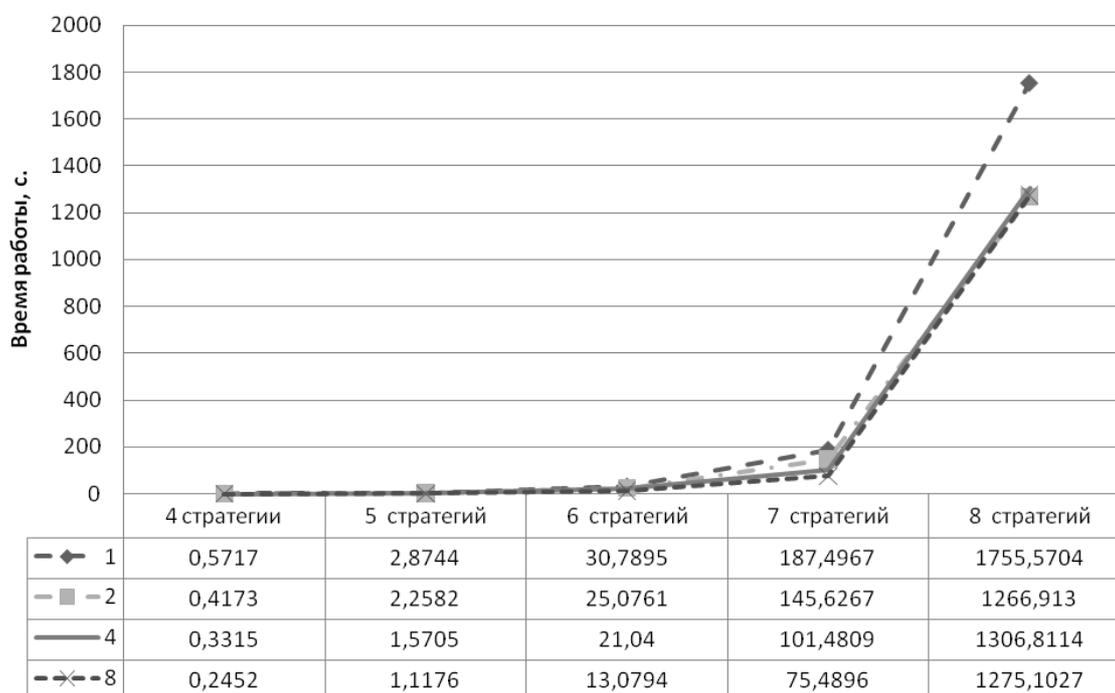


Рис. 4. График зависимости времени работы программы от количества процессоров и количества стратегий для шести игроков

процессам, для равномерного распределения нагрузки на процессы. Причем, существенно на размер матричной игры и, соответственно, на время решения задачи влияет количество

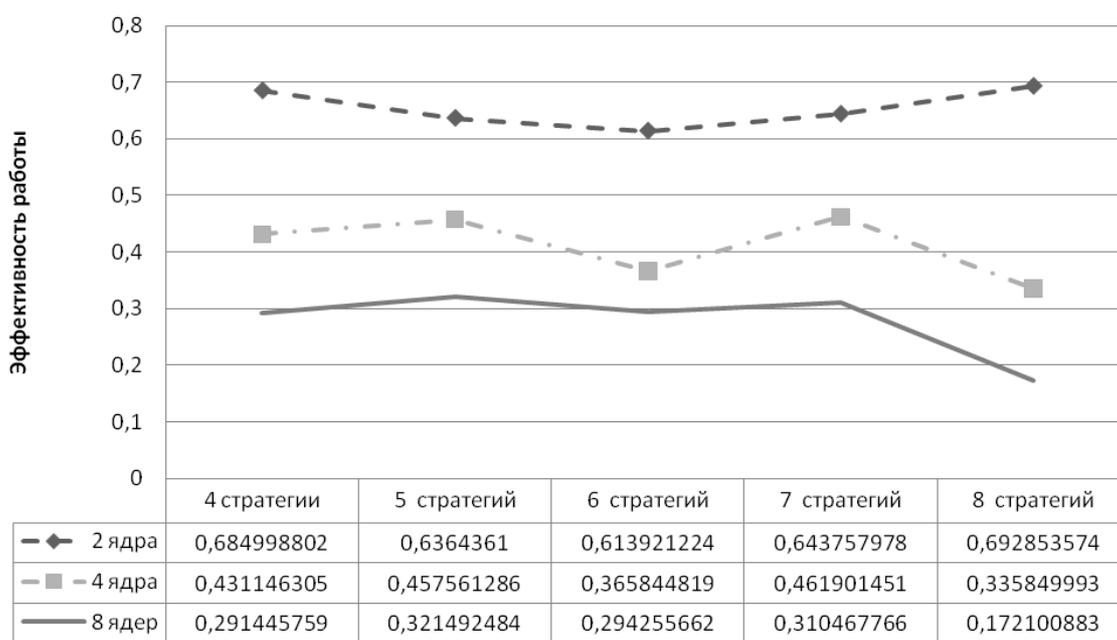


Рис. 5. График эффективности работы параллельной программы в зависимости от количества процессоров и количества стратегий для шести игроков

игроков в исследуемой коалиции. Так же при реализации данного параллельного алгоритма использовалась недостаточно эффективная реализация алгоритма решения задачи линейного программирования, что так же могло привести к низкому показателю эффективности.

Однако, в среднем, эффективность разработанного параллельного алгоритма (отношение времени работы последовательного алгоритма к произведению времени работы параллельного алгоритма и количества процессов) составляет порядка 40 %.

В дальнейшей работе планируется использовать биматричную игру двух игроков S и P для уточнения значения характеристической функции для коалиции S . Биматричная игра позволяет учитывать интересы остальных игроков, не входящих в S .

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг., номер контракта 14.740.11.0689.

Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2011».

Литература

1. Grosu, D. Computing Equilibria in Bimatrix Games by Parallel Support Enumeration / D. Grosu, J. Widger // In Proceedings of the 2008 international Symposium on Parallel and Distributed Computing (July 01 – 05, 2008). ISPD. IEEE Computer Society. – Washington, 2008. – P. 250 – 256.
2. Grosu, D. Computing Equilibria in Bimatrix Games by Parallel Vertex Enumeration / D. Grosu, J. Widger // Parallel Processing, 2009. ICPP '09. International Conference (22 – 25 Sept. 2009). – P. 116 – 123.
3. Grosu, D. Parallel Computation of Nash Equilibria in N-Player Game / D. Grosu, J. Widger // Computational Science and Engineering, IEEE International Conference. – 2009. – P. 209 – 215.

4. Крушевский, А.В. Теория игр / А.В. Крушевский. – Киев.: Вища Шк., 1977.
5. Губко, М.В. Управление организационными системами с коалиционным взаимодействием участников / М.В. Губко. – М.: ИПУ РАН (научное издание), 2003.
6. Губко, М.В. Теория игр в управлении организационными системами / М.В. Губко, Д.А. Новиков. – Изд. 2-е. – М., 2005.
7. Данилов, В.И. Лекции по теории игр / В.И. Данилов. – М.: Рос. экон. шк., 2002.
8. Оуэн, Г. Теория игр / Г. Оуэн. – М.: Мир, 1971.
9. Шеллинг, Т. Стратегия конфликта / Т. Шеллинг. – М.: ИРИСЭН, 2007.
10. Нейман, Д. Теория игр и экономическое поведение / Д. Нейман, О. Моргенштерн. – М.: Наука, 1970.

Нестеренко Максим Юрьевич, кандидат технических наук, доцент, кафедра «Математическое обеспечение информационных систем», Оренбургский государственный университет, n_mu@mail.ru.

Кириллов Алексей Сергеевич, аспирант, кафедра «Администрирование информационных систем», Оренбургский государственный университет, alexey.s.kirillov@gmail.com.

Поступила в редакцию 2 марта 2011 г.