АНАЛИЗ МАСШТАБИРУЕМОСТИ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ КООПЕРАТИВНОЙ ИГРЫ

А.С. Кириллов

Оренбургский госуниверситет

В работе рассматриваются подходы к решению вычислительно-сложной задачи построения оптимальной коалиции и распределение выигрыша в кооперативной игре, заданной множеством биматричных игр для n игроков, с использованием параллельных вычислительных технологий.

Введение

Алгоритмы решения большинства игр имеют экспоненциальную временную сложность, для кооперативных игр необходимо осуществлять комбинаторный перебор всех возможных коалиций, а также применять симплекс-метод для решения серии задач линейного программирования [4, 6–9].

Понятно, что решение задач кооперативных игр – процесс трудоемкий и с увеличением числа игроков трудоемкость решения задачи только повышается [5].

Использование параллельных реализаций алгоритмов решения различных игр позволит, во-первых, увеличить размеры задачи (количество игроков, число их стратегий и т.д.), решаемой за то же время, и, во-вторых, значительно сократить время получения результатов, что обеспечивает большую интерактивность системе поддержки принятия решений при взаимодействии с пользователем, а также позволяет использовать решающий модуль в системах реального времени с критичным временем отклика, например, в планировщике задач для грид-системы [1].

В предыдущих работах [6, 7] были рассмотрены основные понятия теории кооперативных игр и способы их решения, такие, как алгоритм решения кооперативной игры, заданной биматричными играми.

Для оценки эффективности разработанных параллельных алгоритмов обычно рассматривают масштабируемость параллельного алгоритма на вычислительном кластере. При этом параллельный алгоритм называют масштабируемым, если при росте числа процессоров он обеспечивает увеличение ускорения при сохранении постоянного уровня эффективности использования процессоров.

1. Параллельный алгоритм решения кооперативной игры, заданной биматричными играми с использованием итеративного алгоритма Брауна-Робинсона

На основании вышесказанного нами разработаны параллельные алгоритмы решения кооперативной игры, заданной набором биматричных игр с использованием технологии MPI [6, 7, 9]. Алгоритмы основаны на решении задачи линейного программирования (симплекс метода) и на итеративном алгоритме Брауна–Робинсона.

Параллельный алгоритм поиска оптимальных стратегий с использованием линейного программирования и симплекс метода [6] позволяет получить точные решения стратегических игр как подзадач решения кооперативной игры. Но при этом для стратегических игр большого размера не всегда можно получить решения на ЭВМ, ввиду проблем нехватки памяти для размещения решаемой игры и накоплении ошибок.

Для преодоления этих проблем возможно использование итерационных методов решения матричных игр, в частности, мы рассмотрели метод Брауна–Робинсона [3, 1] и разработали параллельный алгоритм решения кооперативной игры, заданной набором биматричных игр, итерационным методом Брауна-Робинсона [10] с использованием технологии MPI.

Схема работы параллельного алгоритма представлена на рис. 1.

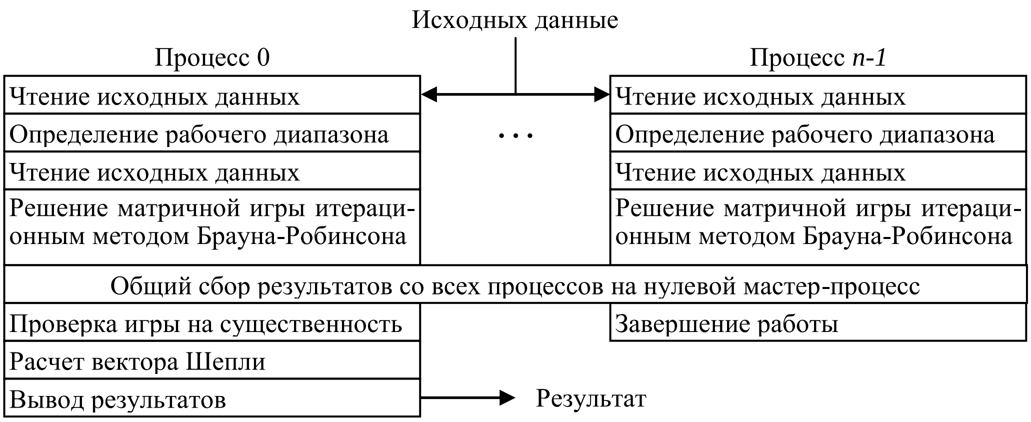


Рис. 1. Схема работы параллельного алгоритма решения кооперативной игры с использованием итерационного метода Брауна-Робинсона

2. Разработка и проведение вычислительного эксперимента

Тестирование разработанного параллельного алгоритма производилось на вычислительном кластере Оренбургского государственного университета.

Для запуска тестовых задач использовались четыре двухпроцессорных вычислительных узла на базе процессоров Intel Xeon 5440, на каждом узле использовались 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 последовательно для 4, 5, 6, 7 и 8 игроков, при этом каждый игрок имел 4, 5, 6, 7 и 8 стратегий. Для каждой задачи формировалось пять произвольных наборов исходных данных. Полученные наборы исходных данных затем решались последовательно на 1, 2, 4 и 8 процессорах, при этом замерялось время выполнения каждой задачи. Затем для каждой тройки значений количества игроков, количества стратегий и количества процессоров вычислялось среднее время работы параллельной программы на всех пяти различных наборах исходных данных. На основании полученных значений времени выполнения были построены графики времени работы и эффективности распараллеливания программ для 5, 6, 7, 8 игроков с 4, 5, 6, 7 и 8 стратегиями для варианта с использованием симплекс метода и итерационного метода Брауна–Робинсона. Графики представлены на рис. 2 – 7 соответственно.

3. Анализ результатов вычислительного эксперимента

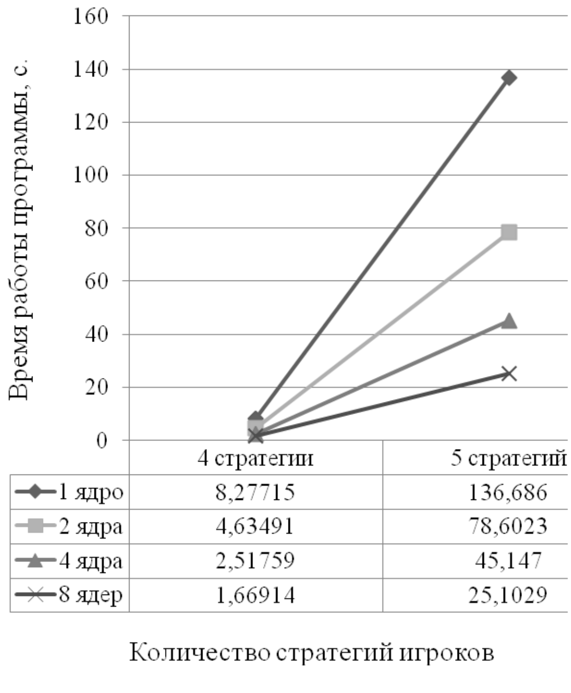
На рис. 2 и 3 представлены графики, из которых видно, что при выполнении точного алгоритма решения кооперативной игры с использованием симплекс метода не достигается приемлемая эффективность работы параллельной программы, и также не достигается приемлемая масштабируемость алгоритма. Это прежде всего связано с тем, что решаемые подзадачи в виде матричных игр имеют различные количества стратегий для коалиции, т.е. имеют различные количества строк в матрицах, что сильно меняет общий размер решаемой симплекс задачи, и, как следствие, появляется дисбаланс распределения подзадач между вычислительными узлами.

Кроме того, для семи игроков с количеством стратегий больше 5 не удалось получить хоть какого-либо решения, т.к. решаемая задача не уложилась в размер выделенной оперативной памяти вычислительного кластера. Аналогично обстоят дела и для игры с количеством игроков 8 и более при любом количестве стратегий. Это обстоятельство сильно ограничивает возможность практического использования данного алгоритма.

Однако в среднем эффективность разработанного параллельного алгоритма с использованием симплекс метода (отношение времени работы последовательного алгоритма к произведению времени работы параллельного алгоритма и количества процессов) составляет величину порядка 40%.

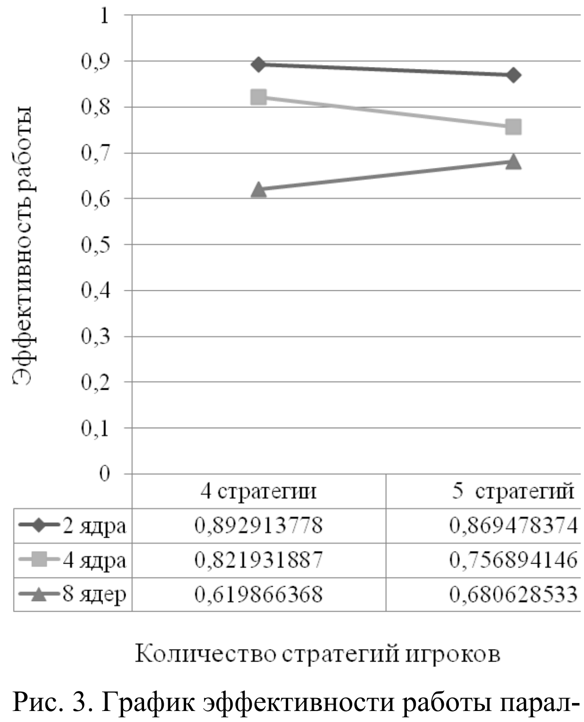
Обратная ситуация наблюдается при реализации параллельного алгоритма для метода Брауна-Робинсона. На графиках (рис. 4–7) видно, что с увеличением числа игроков и числа стратегий у них эффективность работы параллельного алгоритма решения кооперативной игры с использованием итерационного метода Брауна–Робинсона стремится к 100%.

Если сравнить графики на рисунках 2 и 4, то видно, что время выполнения параллельного алгоритма с использованием итерационного метода на несколько порядков меньше времени выполнения параллельного алгоритма с использованием симплекс метода на одних и тех же исходных данных.



	4 стратегии	5 стратегий
1 ядро	8,27715	136,686
2 ядра	4,63491	78,6023
4 ядра	2,51759	45,147
8 ядер	1,66914	25,1029

Рис. 2. График зависимости времени работы параллельной программы от количества процессоров и количества стратегий для семи игроков с использованием симплекс метода



	4 стратегии	5 стратегий
2 ядра	0,892913778	0,869478374
4 ядра	0,821931887	0,756894146
8 ядер	0,619866368	0,680628533

Рис. 3. График эффективности работы параллельной программы в зависимости от количества процессоров и количества стратегий для семи игроков с использованием симплекс метода

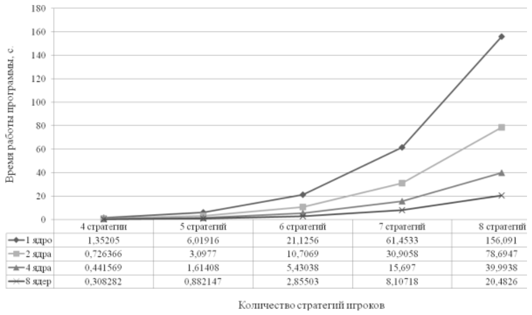


Рис. 4. График зависимости времени работы параллельной программы от количества процессоров и количества стратегий для семи игроков с использованием итерационного метода Брауна–Робинсона

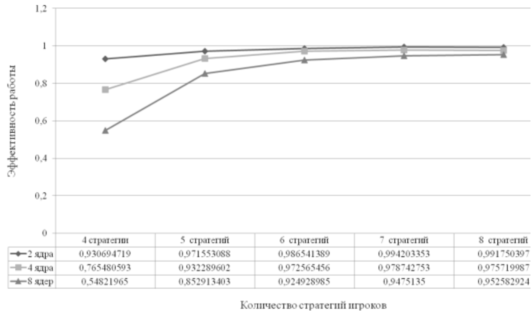


Рис. 5. График эффективности работы параллельной программы в зависимости от количества процессоров и количества стратегий для семи игроков с использованием итерационного метода Брауна–Робинсона

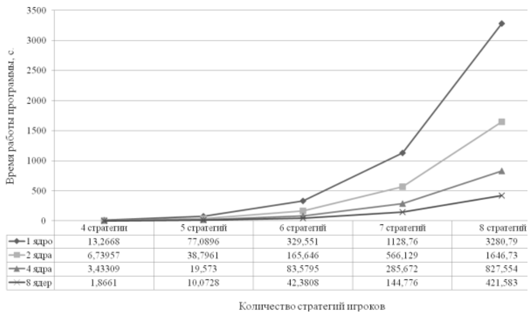


Рис. 6. График зависимости времени работы параллельной программы от количества процессоров и количества стратегий для восьми игроков с использованием итерационного метода Брауна–Робинсона

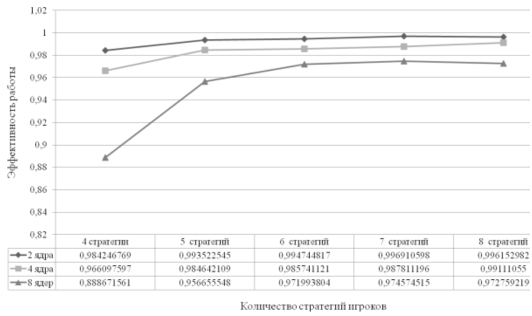


Рис. 7. График эффективности работы параллельной программы в зависимости от количества процессоров и количества стратегий для восьми игроков с использованием итерационного метода Брауна–Робинсона

4. Рекомендации по применению разработанных алгоритмов

На основании результатов вычислительного эксперимента можно предложить следующие варианты использования разработанных параллельных алгоритмов.

1. *Размер исходных данных небольшой (не больше 7 игроков и количество стратегий не более 8), и при этом требуется точное решение кооперативной игры.*

В этом случае нужно применить параллельный алгоритм решения кооперативной игры с использованием симплекс метода. В этом случае мы получим точное решение и приемлемое время, необходимое для решения задачи.

2. *Размер исходных данных небольшой (не больше 7 игроков и количество стратегий не более 8), и при этом не требуется точное решение кооперативной игры.*

В этом случае можно применить параллельный алгоритм решения кооперативной игры как с использованием симплекс метода, так и с использованием итеративного метода Брауна–Робинсона. В первом случае мы получим точное решение и приемлемое время, а во втором – решение, близкое к точному, и практически «мгновенное» решение задачи.

3. *Размер исходных данных большой (больше 7 игроков), и при этом требуется точное решение кооперативной игры.*

В этом случае вначале нужно применить параллельный алгоритм решения кооперативной игры с использованием итеративного метода Брауна–Робинсона. В результате мы получим решение, близкое к точному, и выигрышную коалицию. Затем, узнав выигрышную коалицию, мы можем применить симплекс метод только для выигрышной коалиции для получения точного решения кооперативной игры.

4. Размер исходных данных большой (больше 7 игроков), и при этом не требуется точное решение кооперативной игры.

В этом случае вначале нужно применить параллельный алгоритм решения кооперативной игры с использованием итеративного метода Брауна–Робинсона. В результате мы получим решение, близкое к точному, и выигрышную коалицию.

Литература

1. Brown G.W. Iterative solution of games by fictitious play // Activity analysis of production and allocation (T. C. Koopmans, ed.). New York: Wiley. 1951. P. 374-376.
2. Carroll T.E., Grosu D. Formation of virtual organizations in grids: A game-theoretic approach // Economic Models and Algorithms for Distributed Systems, Autonomic Systems. 2010. Part I. P. 63–81.
3. Robinson J. The annals of mathematics, second series. 1951. Vol. 54, No. 2 (Sep.). P. 296-301.
4. Данциг Дж. Линейное программирование, его применения и обобщения. М.: Прогресс, 1966. 600 с.
5. Крушевский А.В. Теория игр. Киев.: Вища школа, 1977. 216 с.
6. Нестеренко М.Ю., Кириллов А.С. Разработка и анализ высокопроизводительных параллельных алгоритмов решения кооперативных игр // Вестник ЮУрГУ. Серия Математическое моделирование и программирование. 2011. Вып. 8, № 17 (234). С. 92-101.
7. Нестеренко М.Ю., Леонов Д.В., Болгова Е.В., Кириллов А.С. Разработка программного обеспечения для моделирования конкурентного рынка на кластерных системах // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. Март–апрель 2008. № 54. Технологии высокопроизводительных вычислений и компьютерного моделирования. Санкт-Петербург, СПбГУ ИТМО, 2008. С. 156–161.
8. Оуэн Г. Теория игр. М.: Мир, 1971. 229 с.
9. Свидетельство о регистрации программы для ЭВМ № 2010614972. Решение кооперативных игр n-игроков с использованием технологии параллельного программирования MPI / М.Ю. Нестеренко, А.С. Кириллов. Зарегистрировано в Реестре программ для ЭВМ 29.07.2010 г.
10. Свидетельство о регистрации программы для ЭВМ № 2012612496. Решение кооперативных игр n-игроков с использованием итерационного алгоритма Брауна–Робинсона с помощью технологии параллельного программирования MPI / А.С. Кириллов. Зарегистрировано в Реестре программ для ЭВМ 07.03.2012 г.