Оценка степени манипулируемости известных процедур агрегирования для большого числа альтернатив и участников

Ю.В. Зонтов, А.А. Иванов, Д.С. Карабекян, В.И. Якуба
НИУ ВШЭ

В данной работе рассматривается проблема манипулирования, возникающая во время процедуры голосования. Представлены реализованные подходы к переработке существующих последовательных алгоритмов для обобщенных скоринговых процедур агрегирования, а также оценок степени манипулируемости известных процедур агрегирования для большого числа участников и альтернатив с использованием подхода параллелизма данных на суперкомпьютерном кластере ВШЭ. Обсуждаются полученные результаты расчетов и пути решения выявленных трудностей.

Ключевые слова: скоринговые процедуры, процедуры агрегирования, манипулируемость.

1. Введение

Манипулирование – это ситуация, которая возникает при голосовании, когда участник или группа участников записывают в бюллетень неискренние предпочтения, чтобы достичь для себя более выгодного для них результата голосования.

Процедура агрегирования – это правило вычисления результата голосования на основе выборов участников голосования.

Около 50 лет назад было доказано, что любая недиктаторская процедура агрегирования является манипулируемой [1, 2], после чего возник вопрос, какая процедура агрегирования является наименее манипулируемой [3]. Для решения этой задачи в литературе чаще всего используются два подхода: первый подход предполагает выведение точных [4] или приближенных [5] формул, второй подход предполагает использование компьютерного моделирования для поиска наименее манипулируемых процедур агрегирования [6, 7].

Для анализа степени манипулируемости используется индекс Нитцана-Келли (NK-индекс) [8, 9]:

$$\mathrm{NK} = \frac{P_0}{P},$$

где P_0 – количество всех профилей, в которых хотя бы один участник успешно манипулирует, P – число всех возможных профилей предпочтений участников. Таким образом, NK-индекс – это доля всех манипулируемых профилей.

Для вычисления индекса Нитцана-Келли необходимо для каждого профиля сгенерировать все возможные попытки манипулирования, затем, если хотя бы одна попытка манипулирования является успешной, то есть результат агрегирования для манипулирующего участника или группы участников лучше согласно их расширенным предпочтениям, то профиль помечается как манипулируемый. Если для профиля не существует ни одной успешной попытки манипулирования, то он помечается как неманипулируемый.

В данной работе рассматривается модель, в которой n участников голосования выбирают среди m альтернатив. У каждого участника есть предпочтение, выраженное линейным порядком, на множестве альтернатив, то есть одно из $m!$ предпочтений. Профиль P – это совокупность участников голосования и их предпочтений. Процедура агрегирования C(P) ставит в соответствие заданному профилю результат процедуры голосования.

Существует два подхода к тому, каким может быть результат голосования C(P): это одиночный и множественный выбор. В рамках концепции одиночного выбора предполагается, что

если между двумя или более альтернативами возникла ничья, то используется некоторый дополнительный механизм для определения победителя.

Один из популярных вариантов – использование алфавитного (лексикографического) принципа устранения несравнимости. При одиночном выборе процедура голосования дает на выходе одну альтернативу – победителя. Например, если альтернативы {a} и {b} набрали по 4 голоса, и возникла ничья, то выбор будет {a} по алфавитному принципу.

В модели множественного выбора предполагается, что в случае возникновения ничьи между двумя или более альтернативами, все такие альтернативы будут входить в результат коллективного выбора. Например, если альтернативы {a} и {b} набрали по 4 голоса, то выбор будет {a,b}.

В литературе известно несколько десятков процедур агрегирования. В данной работе рассматриваются 5 основных групп процедур: Scoring Rules (Скоринговые), Majority Relation-Based Rules (Мажоритарные), Value Function Rules (Основанные на функции полезности), Tournament Matrix Rules (Турнирные), Q-Paretian Rules (q-Паретовские), а также обобщенные скоринговые правила.

В качестве примеров скоринговых правил рассмотрим две классические процедуры агрегирования: правило относительного большинства и правило Борда.

Правило относительного большинства: В выбор входят альтернативы, которые являются лучшими для наибольшего числа участников голосования.

Правило Борда: Каждой альтернативе x ставится в соответствие число, равное мощности множества альтернатив, худших, чем x при выбранных предпочтениях участника. Сумма данных значений по всем участникам называется рангом Борда для альтернативы x. В выбор входят альтернативы с максимальным рангом.

Исходных предпочтений участника (например, $a \succ b \succ c$) может оказаться недостаточно для сравнения результатов процедуры голосования до и после манипулирования. Например, возникает вопрос, если до манипулирования результат был {b}, а после он стал {a, c}, то стал ли этот результат лучше или хуже для участника? Является ли более предпочтительным получить гарантированно вторую лучшую альтернативу или ничью между лучшей и худшей альтернативами? Для ответа на этот вопрос в литературе используются расширенные предпочтения, которые являются линейным порядком на множестве всех возможных результатов процедуры агрегирования.

Для случая 3 альтернатив это 4 метода построения расширенных предпочтений (при предположении, что исходные предпочтения участника $a \succ b \succ c$):

1. Лексимин
$$\{a\} \succ \{a,b\} \succ \underline{\{b\} \succ \{a,c\} \succ \{a,b,c\}} \succ \{b,c\} \succ \{c\}$$
2. Лексимакс
$$\{a\} \succ \{a,b\} \succ \underline{\{a,b,c\} \succ \{a,c\} \succ \{b\}} \succ \{b,c\} \succ \{c\}$$
3. Рискофил
$$\{a\} \succ \{a,b\} \succ \underline{\{a,c\} \succ \{a,b,c\} \succ \{b\}} \succ \{b,c\} \succ \{c\}$$
4. Рискофоб
$$\{a\} \succ \{a,b\} \succ \underline{\{b\} \succ \{a,b,c\} \succ \{a,c\}} \succ \{b,c\} \succ \{c\}$$

В литературе используются две наиболее популярные модели вероятностей профилей: Impartial Culture (IC) и Impartial Anonymous Culture (IAC).

В модели IC предпочтения участников равновероятны. Для случая n участников и m альтернатив у каждого участника возможно одно из $m!$ предпочтений, значит, общее количество профилей равно $(m!)^n$. Для случая 3 альтернатив и 100 участников всего будет $6.5 * 10^{77}$ различных профилей.

В модели IAC предполагается, что профили равновероятны с учетом анонимности, то есть два профиля считаются одинаковыми, если их можно получить переименованием участников.

В модели IAC используются «единогласные профили», то есть профили, когда все участники имеют одинаковые или почти одинаковые предпочтения, будут более вероятны, чем в модели IC. Количество профилей в модели IAC для случая n участников и m альтернатив равно

$C^{n}_{m!+n-1}$. Для случая $n = 100$ и $m = 3$ общее количество профилей в IAC составит 96.5 миллионов.

Стандартный механизм расчета манипулирумости состоит из следующих шагов [10]:

1. Генерация профилей;
2. Для каждого профиля – генерация всех возможных попыток манипулирования, то есть всех возможных изменений предпочтений манипулирующим участником или группой участников;
3. Подсчет нового результата коллективного выбора для каждой попытки манипулирования;
4. Расчет индекса NK: если хотя бы одна попытка манипулирования привела к более хорошему результату, то профиль считается манипулируемым, и неманипулируемым в противном случае.

Для случая IC их количество равно $(m!)^n$, поэтому в предыдущих исследованиях в этой области, например [11], генерируются не все профили, а 1 миллион случайных профилей. Вторая проблема – высокая вычислительная сложность проверки всех профилей на манипулируемость. Вычислительная сложность равна количеству всех профилей умноженному на количество всех возможных попыток манипулирования и на алгоритмическую сложность вычисления результата процедуры голосования для профиля.

В работе [10] показана оценка сложности для метода с генерацией 1 миллиона случайных профилей, для случая 3 альтернатив и 100 участников время работы алгоритма составляло 63 часа на персональном компьютере. Если бы расчеты производились для 96.5 млн профилей (общее число профилей в модели IAC для случая 100 участников), то расчет только одного случая со 100 участниками занял бы более 6000 часов (более 8 месяцев).

2. Постановка задачи

Как следует из предыдущего раздела, одной из ключевых проблем использования компьютерного моделирования для оценки степени манипулируемости процедур агрегирования является высокая сложность необходимых расчетов. Такая оценка предполагает решение задачи проверки множества профилей предпочтений участников на возможность успешного манипулирования. При этом количество всех возможных профилей предпочтений, как следствие, и сложность вычислений, быстро возрастают с ростом числа участников и альтернатив, что требует использования эффективных вычислительных схем [10].

Помимо вычислительной нагрузки, с ростом числа участников и альтернатив возрастает требуем объем постоянной и оперативной памяти. Размерность задачи оказывается слишком высока для решения на персональных компьютерах, поэтому мы использовали суперкомпьютерный кластер ВШЭ [12] для проведения расчетов.

В качестве исходных материалов в данной работе выступали существующие однопоточные реализации алгоритмов расчета [10] оценки манипулируемости, разработанные сотрудниками Международного центра анализа и выбора решений на языке C#.

На первом этапе работы был выбран курс на переписывание данного программного кода на язык C++ с целью наиболее эффективного использования возможностей как суперкомпьютерного кластера, так и известного быстродействия самого языка. Данный подход применялся для алгоритмов индивидуального манипулирования применительно к числу альтернатив более 5 и показал хорошие перспективы внедрения, однако были выявлены важные недостатки, в первую очередь дополнительные временные затраты на сверку результатов расчета старой и новой версии программ, различия в количестве и устройстве доступных библиотек и структур данных для языков C++ и C#.

На втором этапе работы, при решении задачи распараллеливания алгоритма расчета оценки коалиционной манипулируемости обобщенных скоринговых правил, был выбран альтернативный подход. Было решено сделать запрос на установку на суперкомпьютерном кластере дополнительных программных модулей, а именно фреймворка .net core [13]. Данный подход позволил упростить и ускорить процесс переработки существующих программ для использования на суперкомпьютерном кластере.

3. Программная реализация и результаты вычислительных экспериментов

3.1 Расчет оценки манипулируемости для числа альтернатив более 5

Разработанная версия расчетной программы реализует подход к распараллеливанию на уровне данных (параллелизм данных). В рамках решения задачи определения манипулируемости профиля предпочтений для группы участников происходит создание «пула объектов» [14] класса std::thread библиотеки STL [15], которые последовательно отбирают из набора (1 миллион) профилей первый необработанный и после завершения работы возвращаются в «пул».

Данный подход показал превосходство над наивным подходом, когда полный набор профилей разделяется на равные доли между всеми вычислительными потоками. Он позволил устранить проблему простоя отдельных потоков (вычислительных ядер), который возникал из-за особенностей алгоритма вычисления степени манипулируемости правила агрегирования (NK-индекса). Алгоритм останавливается при нахождении первой модификации профиля, допускающей успешное манипулирование, и позволил довести средний уровень загрузки вычислительных ядер до 98%.

Также при реализации алгоритма удалось реализовать подход без блокировок, т.к. каждый поток сохраняет результат лишь в ассоциированную с ним (связь организуется по числовому идентификатору) ячейку памяти.

Данный подход хорошо показал себя при расчете до 30 участников для 3–7 альтернатив. На рис. 1 и далее приведены примеры расчетов для Правила относительного большинства (классической процедуры агрегирования, при использовании которой выбирается альтернатива, которая занимает первое место в предпочтениях максимального количества участников) и модели расширенных предпочтений Лексимин [16].

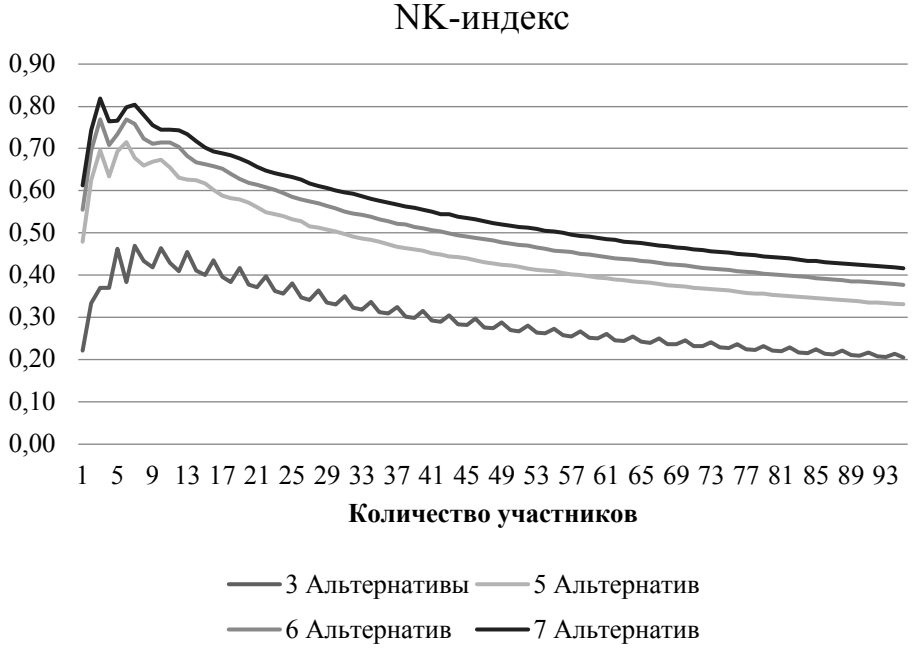

Рис. 1. График зависимости степени манипулируемости процедуры агрегирования (NK-индекс от числа участников для различного числа альтернатив) на примере Правила относительного большинства и расширенных предпочтений Лексимин

Благодаря примененных подходам к распараллеливанию алгоритмов, в рамках данной работы был впервые построен сравнительный график зависимости NK-индекса от числа участников для числа альтернатив более 5 (на примере Правила относительного большинства и расширенных предпочтений Лексимин). График представлен на рис. 2.

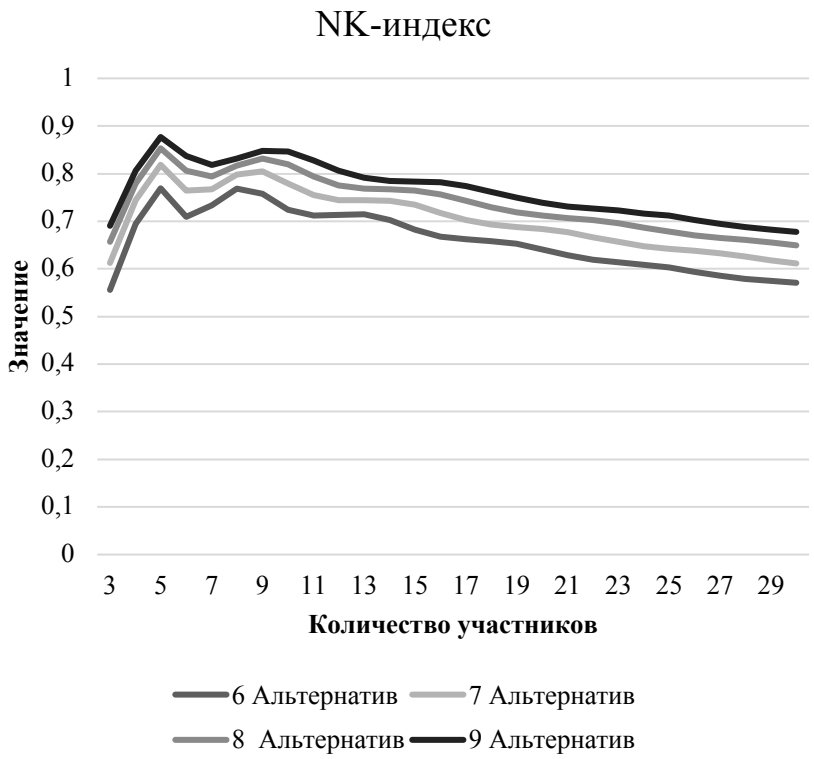

Рис. 2. График зависимости степени манипулируемости процедуры агрегирования (NK-индекс от числа участников для различного числа альтернатив) на примере Правила относительного большинства и расширенных предпочтений Лексимин до 9 альтернатив

Однако организация программы, при которой каждый задействованный вычислительный узел (32 вычислительных ядра) производил расчет для одного заданного числа участников, привело к тому, что каждый последующий расчет занимает больше времени, чем предыдущий (см. рис. 3).

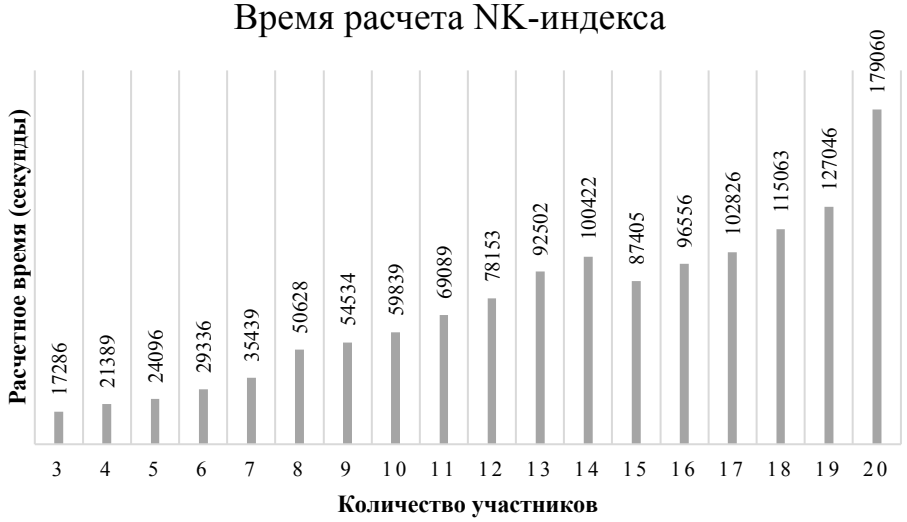

Рис. 3. График зависимости времени расчета на одном узле (32 ядра) для Правила относительного большинства и 9 альтернатив от количества участников

Время расчетов также существенно возрастает с ростом числа альтернатив, что связано с увеличением размерности пространства профилей, когда для каждого из 1 миллиона профилей, производится перебор $(m!)^n - 1$ вариаций, где m – количество альтернатив, а n – количество

участников. В табл. 1 и на рис. 4 приведены данные зависимости времени расчета NK-индекса для 25 участников, Правило относительного большинства, расширенное предпочтение Лексимин.

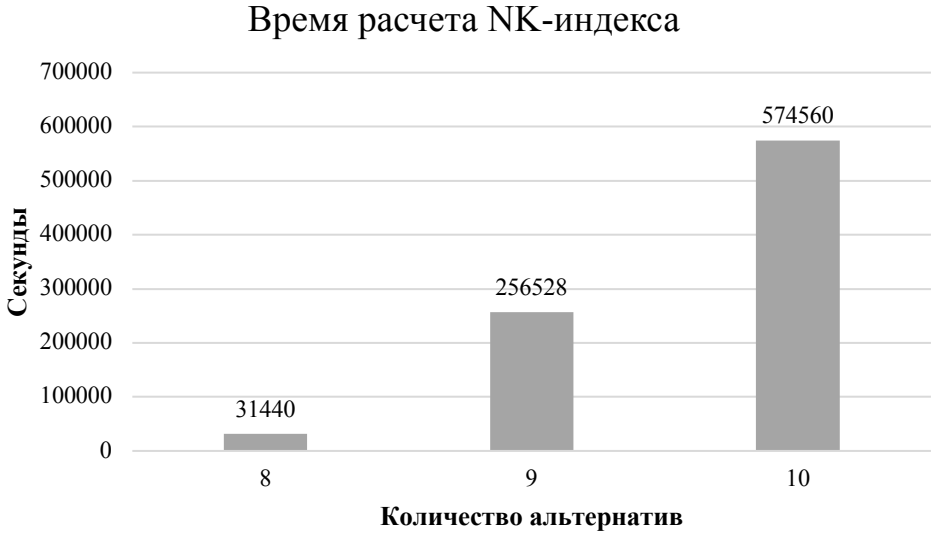

Рис. 4. График зависимости времени расчета на одном узле (32 ядра) для Правила относительного большинства и 25 участников

Таблица 1. Зависимость времени расчета на одном узле (32 ядра) для Правила относительного большинства и 25 участников от числа альтернатив.
*Для 10 участников приведена оценка времени, т.к. расчет завершился по таймауту

Количество альтернатив	Время расчета NK-индекса	Время расчета NK-индекса (секунды)
8	8ч 44м 00с	31440
9	2д 23ч 15м 28с	256528
10	Более 3д 7ч 48м 00с	Более 287280 (574560*)

Таким образом, даже для относительно небольшого числа участников 32 ядра узла вычислительного кластера затрачивают дни машинного времени для расчета индекса манипулируемости на 1 миллионе профилей. Данное ограничение не позволяет использовать существующий подход для выполнения расчетов для числа альтернатив более 9 и числа участников более 30 за приемлемое время.

Следует отметить, что основные временные затраты при работе алгоритма приходятся на перебор большого количества вариаций исходного вектора профиля предпочтений, каждая из которых подвергается проверке на манипулируемость. Представляется целесообразными перейти от распараллеливания на уровне данных к распараллеливанию на уровне алгоритма (параллелизм инструкций). Альтернативным, и более эффективным подходом, представляется использование распараллеливание на GPU с использованием технологии CUDA [17].

3.2 Расчет оценки коалиционной манипулируемости обобщенных скоринговых правил

При решении данной задачи расчет оценки манипулируемости для обобщенных скоринговых правил подсчета означает вычисление NK-индекса не 1, 2 или нескольких, а 6089 вариантов скоринговых процедур, возникающих в результате вариации параметров. Для каждой такой процедуры необходимо рассчитать результат для каждого из профилей предпочтений, а также для возможных вариаций каждого профиля каждым из участников с целью получения наилучшего для себя результата агрегации. При этом требования к ресурсам возрастают с ростом числа участников, поскольку помимо увеличения длины профиля предпочтений увеличивается и

число возможных коалиций. Все это приводит к тому, что задача оценки коалиционной манипулируемости обобщенных правил подсчета очков требовательна как к вычислительным ресурсам компьютера, так и к ресурсам памяти.

Поскольку для вероятностной модели Impartial Culture общее количество профилей составляет $(m!)^n$, что для случая 3 альтернатив и 40 участников дало бы порядка $13*10^{30}$ профилей, что сделало бы невозможным расчеты на современных компьютерах, мы использовали метод генерации всех возможных профилей для вероятностной модели Impartial Anonymous Culture с дальнейшим пересчетом вероятностей профилей и результатов в Impartial Culture (были использованы алгоритмы и формулы из [18]). Общее количество профилей в Impartial Anonymous Culture оценивается как $C^n_{m!+n-1}$, для случая m=3 альтернатив и n=40 участников это дает $C^n_{m!+n-1}$ = 1221759 профилей и занимает 193 мегабайта на жестком диске, тогда как в случае n = 100 полное количество профилей = $96,5*10^6$ и занимает 15,5 гигабайт.

Расчет для одного множества обобщенных скоринговых правил (включает 6089 различных правил) даже для 20 участников на персональном компьютере в однопоточной реализации алгоритма занял около 30 часов, что, учитывая очень быстрый рост алгоритмической сложности сделал невозможным расчет за разумное время случая, например, 40 участников. Все вышеперечисленное указывает на необходимость создания параллельной реализации алгоритма и ее запуска на суперкомпьютере.

Программа использует параллелизм данных в том смысле, что исходный массив профилей предпочтений распределяется между несколькими потоками программы, каждый из которых выполняется на отдельном вычислительном ядре в пределах одного узла кластера, что также соответствует модели параллелизма с общей памятью.

Однако, как и в случае необобщенных скоринговых правил, простое разделение исходного числа профилей поровну между потоками не подходит для задачи оценки манипулируемости, поскольку расчет завершается при нахождении первой вариации профиля, приводящей к выигрышу для данного агента и выбранного набора параметров. Это может привести к простою отдельных вычислительных ядер, поскольку они могут завершить свою работу раньше остальных, а специфика выполнения задач на кластере не позволяет повторно использовать такие ядра до полного завершения расчета. Для решения этой проблемы необходимо организовать обработку массива данных с использованием "пула объектов" потоков.

Для решения этой задачи на языке C# мы использовали метод Parallel.ForEach с опцией разбиения «Без буферизации», реализованный в библиотеке .Net Task Parallel Library (TPL) [19]. Отключение буферизации при разбиении массива обрабатываемых данных (списка векторов – профилей предпочтений) необходимо для устранения возможной утечки памяти, связанной с тем, что механизм разбиения, используемый по умолчанию для структуры данных списка, может при некоторых условиях неограниченно выделять память.

Еще одна важная проблема, которую было необходимо решить при распараллеливании алгоритма, была минимизация необходимости синхронизации потоков. В результате приведенного преобразования нам удалось достичь 88% загрузки вычислительных ядер в процессе вычислений.

Программа использует параллелизм задач в том смысле, что применяется пакетный режим суперкомпьютерного кластера. Для ее работы мы подготовили файл скрипта для команды **sbatch**, в котором запрашиваются различные ресурсы, подготавливается среда, а затем осуществляется запуск расчетной программы для диапазона значений числа участников.

5. Заключение

В результате данной работы были разработаны параллельные версии расчетных программ для оценки манипулируемости скоринговых правил. Программа для расчета степени манипулируемости для 10 скоринговых правил перенесена на язык программирования C++, оптимизирована и распараллелена с использованием возможностей библиотеки STL и Intel oneAPI Toolkit, что сделало возможными расчеты для случая более 5 альтернатив.

В рамках данной работы был впервые построен сравнительный график зависимости NK-индекса от числа участников для числа альтернатив более 5 (на примере Правила большинства и расширенного предпочтения Лексимин).

Произведены расчеты для отдельных скоринговых правил для числа альтернатив до 7 и числа участников до 100, а также для числа альтернатив до 9 для числа участников до 30. Произведены расчеты оценки коалиционной манипулируемости обобщенных скоринговых правил для числа участников до 50.

Результаты расчетов доступны в виде интерактивных графиков на сайте Manipulability of Social Choice Rules Project (в разработке, URL: https://demo.moscowlab.info).

Несмотря на достигнутые результаты, в ходе работы мы столкнулись с проблемой, состоящей в том, что с ростом числа участников и альтернатив вычислительная сложность задачи возрастает настолько, что подход с использованием параллелизма данных становится неэффективным. Например, расчет оценки коалиционной манипулируемости для 100 участников для одного семейства обобщенных скоринговых правил не смог завершиться в течение 7 дней (расчет выполнялся на 6 вычислительных ядрах) и был прекращен по таймауту. Аналогично расчет оценки индивидуальной манипулируемости для Правила относительного большинства для случая 10 альтернатив не смог завершиться за приемлемое время даже для малого числа участников.

Представляются перспективными доработка подхода с использованием параллелизма задач для распределения вычислений для заданного числа агентов между узлами (с использованием модели распределенной памяти), а также дальнейшая доработка алгоритма для использования параллелизма инструкций, в том числе с использованием технологии CUDA.

6. Благодарности

Исследование поддержано Международным центром анализа и выбора решений НИУ ВШЭ.
Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ [12].

Литература

1. Gibbard A. Manipulation of voting schemes // Econometrica. 1973. Vol. 41. P. 587–601. DOI: 10.2307/1914083.
2. Satterthwaite M. Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions // Journal of Economic Theory. 1975. Vol. 10. P. 187–217. DOI: 10.1016/0022-0531(75)90050-2.
3. Chamberlin J.R. An investigation into the relative manipulability of four voting systems // Behavioral Science. 1985. Vol. 30, no. 4. P. 195–203. DOI: 10.1002/bs.3830300404.
4. Diss M., Tsvelikhovskiy B.: Manipulable outcomes within the class of scoring voting rules // Math. Soc. Sci. 2021. Vol. 111. P. 11–18. DOI: 10.1016/j.mathsocsci.2021.02.002.
5. Favardin P., Lepelley D. Some Further Results on the Manipulability of Social Choice Rules // Soc Choice Welfare. 2006. Vol. 26. P. 485–509. DOI: 10.1007/s00355-006-0106-2.
6. Aleskerov F., Kurbanov E. Degree of manipulability of social choice procedures // Alkan A. et al. (eds.) Current Trends in Economics. Studies in Economic Theory. Berlin Heidelberg, N.Y.: Springer, 1999. Vol. 8. P. 13–27.
7. Keller O., Hassidim A., Hazon N. New Approximations for Coalitional Manipulation in Scoring Rules // Journal of Artificial Intelligence Research. 2019. Vol. 64 . P. 109–145. DOI: 10.1613/jair.1.11335.
8. Nitzan S. The vulnerability of point-voting schemes to preference variation and strategic manipulation // Public Choice. 1985. Vol. 47. P. 349–370. DOI: 10.1007/BF00127531.
9. Kelly J. Almost all social choice rules are highly manipulable, but few aren't // Social Choice and Welfare. 1993. Vol. 10. P. 161–175. DOI: 10.1007/BF00183344.

10. Иванов А.А. Эффективные вычислительные схемы расчета манипулируемости процедур агрегирования // Информационные технологии и вычислительные системы. 2020. № 2. С. 38–50.
11. Aleskerov F., Karabekyan D., Sanver R., Yakuba V. On manipulability of positional voting rules // SERIEs: Journal of the Spanish Economic Association. 2011. Vol. 2, no. 4. P. 431–446. DOI: 10.1007/s13209-011-0050-y.
12. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021. Vol. 1740, no. 1. P. 012050. DOI: 10.1088/1742-6596/1740/1/012050.
13. Microsoft .Net Core Framework. URL: https://learn.microsoft.com/ru-ru/dotnet/fundamentals/ (дата обращения: 29.01.2025)
14. Freeman A. The Object Pool Pattern. 2015. P. 137–156. DOI: 10.1007/978-1-4842-0394-1_7.
15. Meyers S. Effective STL: 50 specific ways to improve your use of the standard template library. GBR: Addison-Wesley Longman Ltd., 2001.
16. Barbera S, Bossert W, Pattanaik P. Ranking Sets of Objects // S. Barbera, P.J. Hammond, C. Seidl (eds.): Handbook of Utility Theory, vol. 2. Boston: Kluwer Academic Publishers, 2004.
17. Nickolls J., Buck I., Garland M., Skadron K. Scalable Parallel Programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for? // Queue. 2008. Vol. 6, no. 2. P. 40–53. DOI: 10.1145/1365490.1365500.
18. Иванов А.А. Алгоритмы расчета точных значений индексов манипулируемости для случая трех альтернатив // Журнал Новой экономической ассоциации. 2022. Т. 5, № 57. С. 14–23. DOI: 10.31737/2221-2264-2022-57-5-1.
19. Task Parallel Library. URL: https://learn.microsoft.com/ru-ru/dotnet/standard/parallel-programming/task-parallel-library-tpl (дата обращения: 29.01.2025)