Comparative Analysis of Parallel Computational Schemes for   
Solving the Time-consuming Decision Making Problems

В.П. Гергель, Е.А. Козинов

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Abstract. In the present paper, an efficient approach to the parallel solving of the complex multicriterial optimization problems using the heterogeneous computing systems is considered. The complexity of these problems can be very high as the optimized criteria can be multiex tremal and the computing of the criteria values can be time-consuming. In the framework of the proposed approach the multicriterial optimization problem is reduced to solving a series of global optimization problems by means of the convolution of the partial criteria with different sets of parameters. To solve the series of the global optimization problems, the efficient information-statistical global search method is applied. The parallel computations are implemented by solving several global optimization problems simultaneously. This approach provides parallel computing the criteria values with the several optimized parameters. The maximum use of the search information obtained in the course of the computations is an important distinctive feature of the developed parallel computational schemes. The comparative analysis of various methods for parallel computations and numerous numerical experiments are presented in the paper.

Keywords: Decision making, multicriteria optimization, parallel computing, dimensionality

# 1. Введение

Принятие решений (decision making) является присущим практически всем сферам человеческой деятельности. Для формального описания проблем принятия решений предложено целое множество математических постановок, среди которых many classes of optimization problems, including unconstrained optimization, nonlinear programming, global optimization, etc. В наиболее сложных ситуациях выбора широко используются постановки задач многокритериальной оптимизации (МКО). Отличительное свойство задач МКО состоит в возможности задания нескольких критериев, что позволяет более точно формулировать требования к оптимальности выбираемых решений. Тематика МКО является областью активных научных исследований - см., например, монографии [1-6] и обзоры научных и практических результатов в данной области [7-10].

Важной особенностью задач многокритериальной оптимизации является возможная противоречивость частных критериев эффективности, что делает невозможным достижение оптимальных (наилучших) значений одновременно по всем частным критериям. Как следствие, под решением задачи МКО обычно понимается нахождение некоторых компромиссных (efficient, non-dominated) decisions, в которых достигаемые значения по отдельным частным критериям удовлетворяют заданным требованиям к необходимому уровню эффективности.

Данная работа посвящена решению задач МКО, которые используются для описания decision-making problems при проектировании сложных технических объектов и систем. В таких приложениях частные критерии могут иметь сложный *многоэкстремальный* вид, а определение значений критериев и ограничений может потребовать *большого объема вычислений*.

Кроме того, в рамках рассматриваемого подхода допускается возможность корректировки постановок задач МКО при изменении представлений о необходимых требованиях к оптимальности выбираемых решений. Так, в случае избыточности набора частных критериев допускается возможность перевода части критериев в ограничения. Или, наоборот, при недостаточности множества допустимых вариантов возможным является перевод части ограничений в критерии и т.д. Подобные возможности изменения постановок задачи МКО является дополнительным источником повышенной вычислительной сложности процессов поиска оптимальных решений. Как результат, в существующих подходах возможность корректировки постановок задач МКО в ходе вычислений, как правило, не допускается.

Возможность изменения постановок задачи МКО в процессе поиска оптимальных решений является ключевой в рамках рассматриваемого подхода. Реалистичность данного подхода предполагает преодоление **значительной вычислительной сложности** задач принятия решения, что может быть обеспечено за счет использования высокоэффективных методов глобальной оптимизации и полного использования поисковой информации, получаемой в процессе вычислений.

В данной статье представлены результаты выполненных исследований по обобщению постановок задач принятия решений [11-12] и по разработке высокоэффективных методов глобальной оптимизации, использующих всю поисковую информацию, получаемую в процессе вычислений [13-17]. В работах [14-17] представлены разработанные ранее параллельные алгоритмы. Предложенные алгоритмы способны эффективно использовать сотни и тысячи вычислительных ядер [Gergel, V., Kozinov, E. GPU-based parallel computations in multicriterial optimization(Conference Paper) // Communications in Computer and Information Science, V. 965, 2019, P. 88-100]. В рамках данной работы производиться сравнительный анализ реализованных параллельных алгоритмов на современных вычислительных системах. Сравниваются реализации алгоритмов для обшей памяти, распределенной памяти и гетерогенных вычислительных систем. В результате сравнительного анализа подбирается оптимальная конфигурация запуска.

Дальнейшая структура статьи имеет следующий вид. В главе 2 дается постановка задачи многократной многокритериальной оптимизации. В Главе 3 рассматриваются основы разработанного подхода: сведение многокритериальных задач к скалярным задачам оптимизации при помощи минимаксной свертки частных критериев и редукция размерности при использовании разверток Пеано. В главе 4 обосновывается возможность повышения эффективности вычислений на основе повторного использования поисковой информации. В Главе 5 излагается общая схема организации параллельных вычислений, позволяющая максимально использовать вычислительный потенциал современных суперкомпьютерных систем. Глава 6 содержит результаты численных экспериментов, подтверждающих перспективность предлагаемого подхода. В заключении обсуждаются полученные результаты и приводятся возможные основные направления продолжения исследований.

# 2. Постановка задачи многократной многокритериальной оптимизации

Для формального описания процесса поиска эффективных вариантов в сложных задачах принятия решений предлагается следующая обобщенная многоуровневая модель.

1. На верхнем уровне в рамках предлагаемой модели определяется задача принятия решения (ЗПР), для которой должен быть выполнен выбор наилучших значений параметров в соответствии с имеющимися требованиями к оптимальности. В самом общем виде, ЗПР может быть задана с помощью *вектор-функции характеристик*

|  |  |
| --- | --- |
| , , | (1) |

где есть *вектор конструктивных параметров*, а есть *область возможных значений*, которая обычно представляет собой *N*-мерный гиперинтервал

|  |  |
| --- | --- |
| , | (2) |

для заданных векторов и .

Предполагается, что значения характеристик являются неотрицательными и их уменьшение соответствует повышению эффективности выбираемых вариантов. Предполагается также, что характеристики могут быть многоэкстремальными и определение их значений может потребовать достаточно большого объема вычислений.

Кроме того, в рамках рассматриваемого похода предполагается, что характеристики удовлетворяют условию Липщица

|  |  |
| --- | --- |
| , | (3) |

where *Li* is the Lipschitz constant характеристики and  denotes the Euclidean norm in . Важно отметить, что выполнимость условия Липщица соответствует практическим приложениям – при небольших вариациях параметра *y*∈*D* соответствующие изменения значений характеристик являются, как правило, ограниченными.

В целом, модель ЗПР определяется с помощью следующего набора элементов

|  |  |
| --- | --- |
| . | (4) |

Предполагается, что в процессе вычислений данная модель остается постоянной и не изменяется.

1. Требования к оптимальности выбираемых вариантов ЗПР могут быть определены следующим образом. Прежде всего, должны быть выделены характеристики для которых необходимым является достижение минимально-возможных значений. Задание множества индексов таких характеристик определяет *векторный критерий эффективности*

|  |  |
| --- | --- |
| . | (5) |

Для характеристик , не вошедших в векторный критерий эффективности, с множеством индексов должны быть заданы достаточные уровни эффективности с помощью вектора допусков . Наличие допусков позволяет определить *вектор-функцию ограничений*

|  |  |
| --- | --- |
| . | (6) |

Ограничения задают *допустимую область поиска*

|  |  |
| --- | --- |
| . | (7) |

Сформированные подобным образом критерии эффективности и ограничения позволяют сформировать *задачу многокритериальной оптимизации* (*МКО*)

|  |  |
| --- | --- |
| , , | (8) |

определяемую на основе модели *S* из (4) с помощью перечисленных выше элементов

|  |  |
| --- | --- |
| . | (9) |

Предложенная выше схема охватывает многие существующие постановки оптимизационных задач. При и общая постановка становится задачей глобальной оптимизации. При и общая постановка определяет задачу нелинейного программирования. При и общая постановка приводится к задаче многокритериальной оптимизации с ограничениями.

1. Использование постановок задач многокритериальной оптимизации снижает сложность формального описания задач принятия решений – вместо необходимости задания единого «глобального» критерия эффективности в задачах МКО становится возможным определение нескольких частных критериев. Такой подход, безусловно, является более простой задачей для лица, определяющего требования к оптимальности выбираемых решений. Следует отметить также, что многокритериальное задание критериев эффективности соответствует практике выбора решений в разных областях приложений.

Рассмотренная выше схема постановки задач МКО впервые была предложена в [11] и широко использовалась при решении многих прикладных задач принятия решений. Вместе с этим, результаты практического использования подхода показали, что формулировка единственной постановки задач МКО также может оказаться затруднительной при изменении представлений о необходимых требованиях к оптимальности. Так, в случае избыточности набора частных критериев может оказаться целесообразным перевод части критериев в ограничения. Или, наоборот, при получении малой допустимой области может потребоваться ослабление допусков или перевод части ограничений в критерии и т.д.

Как результат, в развитие схемы (1)-(9) в рамках расширенной модели оптимального выбора будет допускаться возможность одновременного формулирования нескольких задач МКО

|  |  |
| --- | --- |
| . | (10) |

Следует отметить, что указанное множество задач в процессе вычислений может изменяться за счет добавления новых или удаления уже существующих задач

|  |  |
| --- | --- |
| . | (11) |

Решение задач множества могут выполняться последовательно либо одновременно в режиме разделения времени или параллельно при наличии нескольких вычислительных устройств. Безусловно, одновременный способ решения задач является более предпочтительным – получаемые результаты в ходе вычислений позволяет оперативно корректировать имеющееся множество задач . В случае же использования параллельных вычислений общее время решения поставленных задач может быть значительно сокращено.

В наиболее простом случае, множество может состоять из одной задачи МКО. После решения задачи МКО текущая постановка может быть изменена на основе анализа получаемых результатов, и процесс вычислений может быть продолжен уже для новой постановки задачи МКО до получения желаемого оптимального решения.

В целом, предложенная модель (1)-(11) процесса поиска оптимальных решений определяет новый класс оптимизационных задач - задач *многократной многокритериальной глобальной оптимизации* (ММГО).

Важно отметить, что возможность применения предлагаемой модели процесса поиска решений основывается на информационной совместимости решаемых задач МКО. При смене постановки задач МКО вся поисковая информация, получаемая в процессе вычислений, может быть сохранена и повторно использована при решении новых сформулированных оптимизационных задач. Данный аспект являются ключевым в рамках разработанного подхода и рассмотрен более подробно в Главе 4.

# 3. Редукция задач многократной многокритериальной глобальной оптимизации

Как уже отмечалось ранее, частные критерии эффективности в задачах МКО обычно являются противоречивыми. Данная особенность задач МКО означает, что получение оптимальных (наилучших) значений одновременно по всем частным критериям не может быть обеспечено. В таких условиях, под решением задачи МКО обычно понимается нахождение отдельных компромиссных (effective, non-dominated) decisions, в которых достигаемые значения по отдельным частным критериям не могут быть улучшены без одновременного ухудшения показателей по каким-либо другим критериям. В предельном случае, может потребоваться нахождение всех эффективных (Парето-оптимальных) вариантов .

В силу высокой актуальности разработано и широко используется достаточно большое множество методов решения задач МКО – см., например, [2-10]. Часть этих алгоритмов обеспечивает получение численных оценок всего множества Парето [3,26-28]. Наряду с полезностью такого подхода (определяются все эффективные решения задачи МКО) использование этих методов часто оказывается затруднительным в силу высокой вычислительной сложности получения оценок множества Парето. Кроме того, оценка всего множества может оказаться избыточной, когда для решения задачи МКО достаточным является получение нескольких отдельных эффективных вариантов (что часто имеет место в практических приложениях). Как результат, более широко используемый подход к решению задач МКО основывается на скаляризации векторного критерия в некоторый общий скалярный показатель эффективности, оптимизация которого может быть выполнена с использованием уже существующих методов оптимизации. В числе таких методов, например, Weighted sum method, Compromise programming method, Weighted min-max method, Goal programming и многие другие алгоритмы – см., например, [2-6].

Общий подход, используемый в данной работе, состоит в сведении решения задачи ММГО к решению последовательности однокритериальных задач глобальной[[1]](#footnote-1) оптимизации:

|  |  |
| --- | --- |
| ,, | (12) |

где есть вектор-функция ограничений из (6), а есть минимаксная свертка частных критериев задачи МКО (5)

|  |  |
| --- | --- |
|  | (13) |

с использованием вектора весовых коэффициентов

|  |  |
| --- | --- |
| *s*. | (14) |

Важным свойством минимаксной свертки является необходимость и достаточность данного подхода для решения задачи МКО: результат минимизации *F*(*λ*,*y*) приводит к получению эффективного варианта[[2]](#footnote-2) задачи МКО и, обратно, любой эффективный вариант задачи МКО может быть получен в результате минимизации *F*(*λ*,*y*) при соответствующих значениях коэффициентов свертки *λi*, 1≤*i*≤*s* – см., например, [4,6,21].

В разработанном подходе предлагается выполнить еще один шаг преобразования решаемых задач *F*(*λ*,*y*) из (12) – провести редукцию размерности с использованием *кривых* (*разверток*) Пеано *y*(*x*), обеспечивающих однозначное отображение отрезка [0,1] на N-мерный гиперкуб D [18, 20]. В результате такой редукции многомерная задача глобальной оптимизации (12) сводится к одномерной задаче:

|  |  |
| --- | --- |
| *F*(*λ*,y(x\*)) = min { *F*(*λ*,y(x)) : x∈[0,1] }. | (15) |

Рассматриваемая схема редукции размерности сопоставляет многомерной задаче с липшицевой минимизируемой функцией *F*(*λ*,y) одномерную задачу, в которой редуцированная функция *F*(*λ*,y(x)) удовлетворяют равномерному условию Гельдера, т.е.

|  |  |
| --- | --- |
| , | (16) |

где константа *H* определяется соотношением , *N* есть размерность задачи оптимизации (1), а *L* есть константа Липшица функции *F*(*λ*,y) из (12). Редукция размерности позволяет применять для решения задач (12) многие хорошо известные и высокоэффективные одномерные алгоритмы глобальной оптимизации (после проведения необходимого обобщения) – см., например, [18, 20,29-35].

Для получения нескольких эффективных вариантов (или оценки всей области Парето) задача (12) должна быть решена для соответствующего набора значений вектора . В этом случае, множество задач МКО из (10), необходимых для решения исходной задачи принятия решения, преобразуется в более широкое множество задач скалярной оптимизации (12)

|  |  |
| --- | --- |
|  | (17) |

для которого каждой задаче может соответствовать несколько задач (12) для различных значениях коэффициентов .

Следует отметить, что все ранее высказанные замечания по множеству задач из (10) справедливы и для множества задач из (17) – задачи множества могут решаться последовательно или одновременно, а состав множества задач может изменяться в ходе вычислений.

# 4. Повышение эффективности вычислений на основе повторного использования поисковой информации

Как уже отмечалось ранее, решение задач МКО *P* из (10) и соответствующих задач глобальной оптимизации *F* из (15) может потребовать большого объема вычислений даже при решении одной отдельной задачи. В случае же необходимости решения последовательности задач из множеств и требуемый объем вычислений может оказаться еще более значительным. И, тем самым, преодоление вычислительной сложности задач принятия решений рассматриваемого класса является необходимым условием возможности практического применения предлагаемого подхода. Возможный подход к решению отмеченной проблемы может состоять в интенсивном использовании всей поисковой информации, получаемой в процессе вычислений.

Численное решение задач глобально-оптимального выбора обычно сводится к последовательному вычислению значений характеристик в точках области поиска [18, 22-25]. Получаемые в результате вычислений данные могут быть представлены в виде *матрицы поисковой информации* (МПИ)

|  |  |
| --- | --- |
|  | (18) |

В результате скаляризации векторного критерия (12) и применения редукции размерности (15) МПИ из (18) может быть преобразована к форме *матрицы состояния поиска* (*МСП*):

|  |  |
| --- | --- |
| , | (19) |

где

* *xi*, 1≤*i*≤*k*, - есть редуцированные точки выполненных итераций глобального поиска, в которых вычислялись значения критериев;
* *zi*, , 1≤*i*≤*k*, - есть значения скалярного критерия и ограничений текущей решаемой задачи оптимизации *F*(*λ*,y(x)) из (15) в точках *xi*, 1≤*i*≤*k*, т.е.

*zi* = ϕ(*xi*) = *F*(*λ*,y(*xi*)), , 1≤*i*≤*k*;

* *li*, 1≤*i*≤*k*, - есть номера итераций глобального поиска, на которых вычислялись точки *xi*, 1≤*i*≤*k*; данные номера используются для запоминания соответствия редуцированных и многомерных точек выполненных итераций, т.е.

*y j*= y(*xi*), j = li, 1≤*i*≤*k*.

Матрица состояния поиска содержит поисковую информацию, приведенную в текущей решаемой скалярной редуцированной задачи (15). Кроме того, поисковая информация в МСП упорядочена по значениям координат точек *xi*, 1≤*i*≤*k*, для более эффективного выполнения алгоритмов глобального поиска; упорядоченное представление точек отражается использованием нижнего индекса, т.е.

*x*1 ≤ *x*2 ≤ … *xk*.

Представление поисковой информации, получаемой в ходе вычислений, в виде матриц и составляет основу для создания эффективных процедур оптимизационного поиска. Наличие подобной информации позволяет осуществлять адаптивный выбор точек выполняемых итераций глобального поиска с учетом результатов всех ранее выполненных вычислений

|  |  |
| --- | --- |
| , | (20) |

где есть правило вычисления точек применяемого алгоритма оптимизации. Подобный адаптивный выбор точек выполняемых итераций поиска может ускорить нахождение эффективных решений, а в случае задач глобальной оптимизации накопление всей поисковой информации и использование правил вида (20) является просто обязательным. Любое сокращение поисковой информации при определении выполняемых точек будет приводить к проведению избыточных итераций глобального поиска.

Особо следует отметить, что наличие МПИ из (18) позволяет привести результаты всех предыдущих вычислений *zi*, 1≤*i*≤*k*, в МСП к значениям очередной решаемой задачи оптимизации *F*(*λ*,y(x)) из (15) для любых новых параметрах постановки задачи МКО *P* из (9) и при любых новых значениях коэффициентов сверткииз (14) на основе имеющихся в ранее вычисленных значений характеристик *wi*, 1≤*i*≤*k*, без каких-либо повторных трудоемких вычислений значений из (1), т.е.

|  |  |
| --- | --- |
| . | (21) |

Тем самым, вся поисковая информация в полном объеме может быть задействована для продолжения вычислений. Эффективное использование данной информации становится важным требованием при выборе методов решения задач *F*(*λ*,y(x)) из (15). Повторное использование поисковой информации может обеспечить постоянное уменьшение объема вычислений для решения каждой последующей оптимизационной задачи вплоть до выполнения всего лишь нескольких ит ераций для поиска очередного эффективного варианта.

Вместе с этим следует отметить, что возможность использования матриц и для организации глобального поиска требует решения вопросов эффективной обработки поисковой информации. В силу возможного большого количества итераций глобального поиска и соответственно больших объема поисковых данных необходимо обеспечить возможность хранения и поддержки упорядоченности данных для МСП значительных размеров с десятками и сотнями миллионов столбцов.

# 5. Параллельные вычисления в задачах многократной многокритериальной глобальной оптимизации

Последовательное решение множества задач из (17) для нахождения нескольких глобально-оптимальных вариантов значительно повышает требуемый объем вычислений для решении задач оптимального выбора. Возможный простой способ ускорения выполняемых вычислений состоит в параллельном решении задач *F*(*λ*,y(x)) множества . При достаточном количестве вычислительных элементов (процессоров) время решения всего множества задач будет определяться временем вычислений для задачи *F*(*λ*,y(x)), решение которой выполнялось наиболее долго. Такой подход достаточно просто реализуем, однако не решает проблему вычислительной сложности при необходимости увеличения множества задач уже в процессе вычислений.

Развитие рассмотренного выше способа организации параллельных вычислений может быть выполнено за счет использования информационной совместимости задач множества – как отмечается в Главе 4, значения характеристик *wi*, 1≤*i*≤*k*, вычисленные при решении какой-либо задачи *F*1(*λ*,y(x)), могут быть преобразованы к значениям любой другой задачи *F*2(*λ*,y(x)). Данный результат позволяет предложить следующую общую схему параллельных вычислений для одновременного решения задач множества .

1. **Распределение задач**. Перед началом вычислений задачи множества из (17) распределяются между вычислительными узлами многопроцессорной системы. Данное распределение может быть достаточно разнообразным - для решения отдельной задачи множества может быть выделено различное количество вычислительных ядер (*q*, *q*≥1) и вычислительных узлов (*p*, *p*≥1). Возможность использования нескольких вычислительных узлов *p>*1 для решения одной и той же задачи обеспечивается при помощи применения множества разверток - см. [13-14]. Если для решения отдельной задачи множества планируется использовать процессорные ядра вычислительного узла не полностью, то в этом случае на одном и том же узле может решаться несколько задач оптимизации в зависимости от количества имеющихся процессоров на вычислительном узле и числа вычислительных ядер в каждом процессоре.
2. **Выбор алгоритмов оптимизации**. Предлагаемая схема параллельных вычислений является общей - для решения задач на каждом вычислительном узле могут применяться различные методы многоэкстремальной оптимизации – см., например, [18, 20,29-35]. Главное требование к выбираемым алгоритмам – методы должны использовать поисковую информацию и для повышения эффективности глобального поиска. И, как уже отмечалось ранее в Главе 3, алгоритмы глобальной оптимизации должны быть обобщены для возможности решения редуцированных одномерных задач глобальной оптимизации вида *F*(*λ*,y(x)) множества . Возможность применения предлагаемого подхода обоснована на примере эффективных алгоритмов глобального поиска, разработанных в рамках информационно-статический теории многоэкстремальной оптимизации – см. [12-18].
3. **Выполнение итераций глобального поиска**. Выполнение итераций глобального поиска для каждой решаемой задачи *F*(*λ*,y(x)) множества осуществляется параллельно. Выполнение каждой итерации на каждом вычислительном узле включает выполнение следующих действий:

а) Выполняется выбор *q*, *q*≥1 точек *xi*, 1≤*i*≤*q*, очередной итерации глобального поиска в соответствии с правилом (20) метода оптимизации; выбор точек осуществляется с учетом имеющейся поисковой информации и ; количество генерируемых точек *q*, *q*≥1 определяется числом вычислительных ядер, используемых для решения задачи *F*(*λ*,y(x));

б) Для каждой выбранной одномерной точки *xi*∈[0,1], 1≤*i*≤*q*, очередной итерации глобального поиска определяется многомерный образ *yi*∈*D*, 1≤*i*≤*q*, в соответствии с отображением *y*(*x*). Далее каждая вычисленный образ *yi*∈*D*, 1≤*i*≤*q*, передается всем используемым вычислительных узлам для исключения многократного выбора одних и тех же точек в области *D* при решении задач множества ;

в) Вычисляются значения характеристик из (1) во всех точках *yi*∈*D*, 1≤*i*≤*q*; для каждой отдельной точки *yi*∈*D*, 1≤*i*≤*q*, данные вычисления выполняются параллельно с использованием разных вычислительных ядер; вычисленные значения характеристик передаются всем используемым вычислительных узлам для включения полученных данных в состав поисковой информации и .

1. **Пополнение поисковой информации**. Перед началом каждой новой итерации глобального поиска проверяется наличие данных, переданных от других вычислительных элементов (процессоров или ядер); получаемые данные должны включаться в состав поисковой информации.

В соответствии с представленной вычислительной схемой параллельных вычислений каждый вычислительный элемент будет содержать одинаковые копии матрицы с поисковой информацией из (18); матрицы из (19) будут иметь различающийся набор точек глобального поиска *xi*, 1≤*i*≤*k*, за счет использования разных разверток *y*(*x*) для редукции размерности и различающиеся значения скалярного критерия и ограничений *zi*, , 1≤*i*≤*k*, в соответствии с решаемыми задачами *F*(*λ*,y(x)) из множества .

В рамках подобной вычислительной схемы становится возможным расширение множества решаемых задач в любой момент вычислений – для решения новой задачи *F*(*λ*,y(x)) достаточно выделить дополнительный вычислительный элемент, продублировать множество и сформировать соответствующую матрицу .

Отметим, что в данной схеме параллельных вычислений отсутствует какой-либо единый управляющий узел. Количество вычислительных узлов может изменяться в ходе глобального поиска, а передача данных может осуществляться в асинхронном режиме (вычислительные узлы обрабатывают получаемые данные только по мере их поступления).

Понятно, что эффект повышения эффективности глобального поиска от использования множеств с поисковой информацией и зависит от применяемых алгоритмов оптимизации. Анализ отдельных элементов рассмотренный выше схемы параллельных вычислений применительно к информационно-статистическим алгоритмам глобальной оптимизации был выполнен в работах [12-17]. Далее в Главе 6 приводятся результаты вычислительных экспериментов для полной экспериментальной оценке эффективности предлагаемой схемы для выбора наилучших вариантов использования вычислительных ресурсов высокопроизводительных суперкомпьютерных систем.

# Результаты вычислительных экспериментов

Вычислительные эксперименты проводились на суперкомпьютере «Лобачевский» Нижегородского государственного университета (операционная система – CentOS 6.4, система управления – SLURM). Один узел суперкомпьютера располагает 2-я процессорами Intel Sandy Bridge E5-2660 2.2 GHz, 64 Gb RAM. Центральный процессор является 8-и ядерным (т.е. всего на узле доступно 16 ядер CPU). Для получения исполняемого программного кода использовался компилятор Intel C++ 17.0.

Выполнение экспериментов было организовано следующим образом. В самом начале приводятся результаты сравнение предлагаемого подхода с рядом других алгоритмов многокритериальной оптимизации. Далее были выполнены эксперименты по выбору оптимальной конфигурации запуска при решении серии шестимерных пятикритериальных задач МКО.

Перед проведением вычислительных экспериментов приведем результаты сравнения предлагаемого подхода с рядом других алгоритмов многокритериальной оптимизации, представленные в [13]. Для сравнения использовалась тестовая двухкритериальная задача, предложенная в [26]

|  |  |
| --- | --- |
|  | (22) |

Под решением задачи МКО понималось построение численной аппроксимации области Парето. Для оценки качества аппроксимации сравнивались полнота и равномерность покрытия области Парето с помощью следующих двух показателей [26,27]:

* The *hypervolume index* (HV), определяемый как объем подобласти значений векторного критерия , доминируемых точками аппроксимации области Парето. Значение этого показателя может быть получено как суммарный объем пересекающих гиперпараллелепипедов, вершинами которых являются точки аппроксимации области Парето и некоторой reference point . В качестве reference point в выполненных экспериментах используется векторная величина

.

Данный показатель характеризует полноту аппроксимации области Парето (большее значение соответствует более полному покрытию области Парето).

* The *distribution uniformity index* (DU) of the points from the Pareto domain approximation, определяемый соотношениями:

|  |  |
| --- | --- |
| *, ,* |  |

где *p* есть количество точек в аппроксимации области Парето *PDA*. Данный показатель характеризует равномерность покрытия области Парето (меньшее значение соответствует более равномерному покрытию области Парето).

В рамках данного эксперимента сравнивались пять алгоритмов многокритериальной оптимизации:

* The Monte-Carlo (MC) method, в котором точки выполняемых испытаний выбирается случайным образом равномерно в области поиска *D*,
* The genetic algorithm SEMO from the PISA library [8,27],
* The Non-uniform coverage (NUC) method [8],
* The bi-objective Lipschitz optimization (BLO) method proposed in [27],
* Последовательный вариант (MAMGS) – разработанный авторами алгоритм.

Для первых трех алгоритмов из приведенного списка использовались результаты экспериментов из [8], для метода BLO результаты экспериментов были приведены в [27]. В полном виде результаты экспериментов из [13] представлены в таблице 1.

Таблица 1. Результаты вычислительных экспериментов из [13] для тестовой задачи (22)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Iterations** | **PDA points** | **HV** | **DU** |
| MC | 500 | 67 | 0.300 | 1.277 |
| SEMO | 500 | 104 | 0.312 | 1.116 |
| NUC | 515 | 29 | 0.306 | 0.210 |
| BLO | 498 | 68 | 0.308 | 0.175 |
| **MAMGS** | **370** | **100** | **0.316** | **0.101** |

Из таблицы 1 видно, что разработанный алгоритм за меньшее число итераций нашел лучшую оценку области Парето как по полноте, так и по равномерности покрытия.

В основной серии экспериментов было решено 30 задач МКО. Каждая задача являлась шестимерной и пятикритериальной, т.е. , . В качестве критериев использовались многоэкстремальные функции, определяемые при помощи генератора GKLS [36]. Данный генератор позволяет сгенерировать многоэкстремальные задачи оптимизации с априорно известными свойствами: количество локальных минимумов, размер их областей притяжения, точку глобального минимума, значение функций в ней и т.д. Пример линий уровней функций, полученных генератором GKLS представлен на рис. 1.

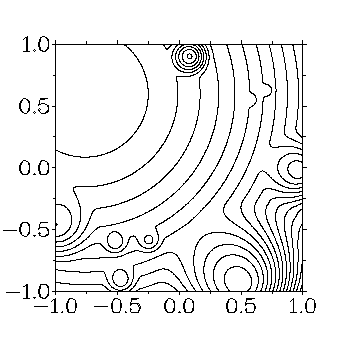
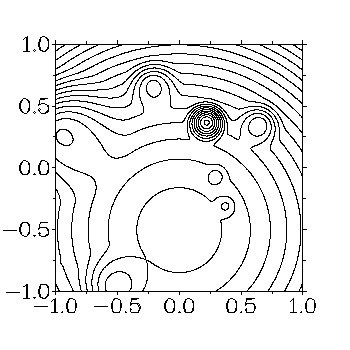
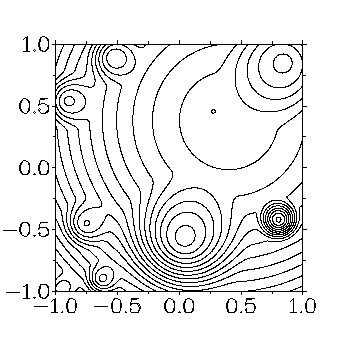


Рис. 1. Линии уровня для трех многоэкстремальных функций семейства   
тестовых задач оптимизации GKLS

Отметим некоторые параметры используемые при проведении вычислительных экспериментов. При решении серии задач МКО во всех приведенных далее результатах экспериментов использовалось условие остановки алгоритма по заданной точности . Для поиска оценки области Парето использовались 100 различных набора сверток из (17). В вычислительных экспериментах использовалось 10 узлов кластера с двумя процессорами содержащими 8 ядер.

Для поиска оптимальной конфигурации запуска в начале были выполнены эксперименты показывающие эффективность использования одного вычислительного узла. Результаты экспериментов приведены в таблице 2.

Таблица 2. Оценка эффективности использования одного узла кластера при решении серии из 30 шестимерных пятикритериальных задач МКО.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Core | Информация повторно: | Итераций | S1 | S2 |
| 1 | не используется | 26 813 722,7 | 1,0 |  |
| 1 | используется | 9 103 069,6 | 2,9 | 1,0 |
| 8 | используется | 1 291 720,0 | 20,8 | 7,0 |
| 16 | используется | 609 169,5 | 44,0 | 14,9 |

В таблице 2 столбец *Core* отображает количество используемых ядер в рамках вычислительного узла. Второй столбец содержит информацию о том – использовалась ли повторна информация при переходе вычислений от одной свертки к другой в серии подзадач (17) или нет. Третий столбец содержит информацию о среднем количестве итераций затрачиваемых на решение одной задачи МКО. Столбец *S1* содержит информацию об общем ускорении полученным при использовании алгоритма поиска в рамках одного узла. Столбец *S2* показывает ускорение параллельного алгоритма относительно последовательной версии эффективно использующей всю накопленную информацию.

Из таблицы 2 видно, что при использовании одного ядра вычислительной системы при сохранении и повторном использовании накопленной информации удалось достичь ускорение почти в три раза. Общее достигнутое ускорение составило 44 раза.

Далее проводились эксперименты по поиску оптимальной конфигурации запуска на вычислительной системе с 10 узлами. При поиске оптимальной конфигурации учитывались следующие варианты решения задач МКО. Во-первых, все задачи МКО могли решаться независимо на разных узлах вычислительной системы. Для организации подобного запуска использовалась очередь задач. Если один из вычислительных узлов освобождался, то из очереди бралась очередная задача для вычислений. Во-вторых, согласно разделу 5 одну задачу МКО возможно решать используя несколько вычислительных узлов. При решении задачи несколькими вычислительными узлами рассматривались различные варианты одновременного решения нескольких подзадач из (17) и использования множественных разверток. Результаты вычислительных экспериментов приведены в таблице 3.

Таблица 3. Оценка эффективности использования десяти узлов кластера при решении серии из 30 шестимерных пятикритериальных задач МКО.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Node | NumMCO | SubTask | Evolvent | Iteration | S3 | S1 |
| 1 | 1 | 1 | 1 | 609 169,5 | 1,0 | 44,0 |
| 10 | 1 | 10 | 1 | 75 045,8 | 8,1 | 357,3 |
| 10 | 1 | 5 | 2 | 77 643,6 | 7,8 | 345,3 |
| 10 | 1 | 1 | 10 | 108 700,0 | 5,6 | 246,7 |
| 10 | 2 | 5 | 1 | 76 819,0 | 7,9 | 349,1 |
| 10 | 2 | 1 | 5 | 81 132,8 | 7,5 | 330,5 |
| **10** | **5** | **2** | **1** | **66 618,6** | **9,1** | **402,5** |
| 10 | 5 | 1 | 2 | 77 995,0 | 7,8 | 343,8 |
| 10 | 10 | 1 | 1 | 73 999,7 | 8,2 | 362,3 |

В таблице 3 столбец *Node* содержит информацию об количестве используемых узлов кластера. Столбец *NumMCO* содержит информацию о количестве параллельно решаемых информационно не совместимых задач МКО содержащихся в серии экспериментов. Столбец *SubTask* содержит информацию о том сколько одновременно решалось подзадач из (17). Столбец *SubTask* содержит информацию о том сколько использовалось разверток при решении каждой из подзадач (17). Столбец *Iteration* содержит информацию о среднем количестве итераций затрачиваемых на решение одной задачи МКО. Столбец *S3* показывает ускорение полученное в результате использования десяти узлов и всех доступных вычислительных ядер относительно одного узла. Столбец *S1* как и в таблице 2 показывает общее достигнутое ускорение.

Из таблицы 3 видно, что в рассматриваемой серии экспериментов наиболее оптимальным оказался запуск в котором параллельно решались пять независимых задач МКО, при этом при решении каждой задачи параллельно решались две подзадачи из (17), а множественные развертки не использовались. Достигаемое при этом ускорение вычислений достигает более чем 400 раз.

# References

1. Parnell, G.S., Driscoll, P.J., Henderson, D.L., editors. (2008) Decision Making in Systems Engineering and Management. Wiley, New Jersey (2nd ed., 2011).
2. Miettinen K. (1999) Nonlinear Multiobjective Optimization. Springer.
3. Marler, R. T., Arora, J. S. (2009). Multi-Objective Optimization: Concepts and Methods for Engineering. VDM Verlag.
4. Ehrgott, M. (2005) Multicriteria Optimization. Springer. (2nd ed., 2010)
5. Collette, Y., Siarry, P. (2011) Multiobjective Optimization: Principles and Case Studies (Decision Engineering). Springer.
6. Pardalos, P.M., Žilinskas, A., Žilinskas, J. (2017) Non-Convex Multi-Objective Optimization. Springer.
7. Figueira,J., Greco, S., Ehrgott, M., editors. (2005). Multiple criteria decision analysis: State of the art surveys. New York, Springer.
8. Hillermeier, C., Jahn, J. (2005). Multiobjective optimization: survey of methods and industrial applications. Surv. Math. Ind. 11, 1–42.
9. Zavadskas, E. K., Turskis, Z., Kildienė, S. (2014). State of art surveys of overviews on MCDM/MADM methods. Technological and Economic Development of Economy 20, 165–179.
10. Cho, J.-H., Wang, Ya., Chen, I.-R., Chan, K.S., Swami, A. (2017) A Survey on Modeling and Optimizing Multi-Objective Systems. IEEE Communications Surveys & Tutorials 19 (3), 1867 – 1901.
11. Strongin, R.G., Gergel, V.P. Markin, D.L. 1988) Multicriterion multiextreme optimization with nonlinear constraints. Lecture Notes in Economics and Mathematical Systems, 351, 120-127.
12. Gergel, V.P., Kozinov, E.A. (2016) Accelerating multicriterial optimization by the intensive exploitation of accumulated search data AIP Conference Proceedings 1776, 090003, DOI: 10.1063/1.4965367
13. Gergel, V.P., Kozinov, E.A. (2018) Efficient multicriterial optimization based on intensive reuse of search information. J Glob Optim. DOI: doi.org/10.1007/s10898-018-0624-3
14. Gergel, V.P., Kozinov, E.A. (2017) Accelerating Parallel Multicriterial Optimization Methods Based on Intensive Using of Search Information. Procedia Computer Science, 108, pp. 1463-1472.
15. Gergel, V., Kozinov, E. (2017) Parallel computing for time-consuming multicriterial optimization problems. Lecture Notes in Computer Science, 10421 446-458. DOI: 10.1007/978-3-319-62932-2\_43
16. Gergel, V., Kozinov, E. (2017) Efficient methods of multicriterial optimization based on the intensive use of search information. Springer Proceedings in Mathematics and Statistics, 197, 27-45. DOI: 10.1007/978-3-319-56829-4\_3
17. Gergel, V., Kozinov, E. (2017) An approach for parallel solving the multicriterial optimization problems with non-convex constraints. Communications in Computer and Information Science, 793, 121-135. DOI: 10.1007/978-3-319-71255-0\_10
18. Strongin, R., Sergeyev, Ya. (2000) Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht (2nd ed. 2013, 3rd ed. 2014).
19. Strongin, R., Gergel, V., Grishagin, V., Barkalov, K. (2013) Parallel computations for global optimization problems. Moscow State University Press. (in Russian)
20. Sergeyev Y.D., Strongin R.G., Lera D. (2013) Introduction to global optimization exploiting space-filling curves. Springer.
21. Eichfelder, G. (2009) Scalarizations for adaptively solving multi-objective optimization problems. Comput. Optim. Appl. 44, 249–273.
22. Floudas, C.A., Pardalos, M.P. (2016). Recent Advances in Global Optimization. Princeton University Press.
23. Locatelli, M., Schoen, F. (2013). Global Optimization: Theory, Algorithms, and Applications. SIAM.
24. Arora, R.K. (2015). Optimization: Algorithms and Applications. CRC Press.
25. Bazaraa, M.S., Sherali, H.D., Shetty, C. M. (2006). Nonlinear Programming: Theory and Algorithms, John Wiley and Sons.
26. Evtushenko, Yu.G., Posypkin, M.A. (2014) A deterministic algorithm for global multi-objective optimization. Optimization Methods and Software,29(5), 1005–1019.
27. Žilinskas, A., Žilinskas, J. (2015) Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems. Commun Nonlinear Sci Numer Simulat, 21, 89–98.
28. Markin, D.L., Strongin, R.G. (1993) Uniform estimates for the set of weakly effective points in multi-extremum multicriterion optimization problems. Computational Mathematics and Mathematical Physics 33(2), 171–179.
29. Piyavskij, S.A. (1972) An algorithm for finding the absolute extremum of a function. Computational Mathematics and Mathematical Physics 12, 57–67.
30. Shubert, B.O. (1972) A sequential method seeking the global maximum of a function. SIAM J. Numer. Anal. 9, 379–388.
31. Breiman, L., Cutler, A. (1993) A deterministic algorithm for global optimization. Math. Program. 58 (1–3), 179–199.
32. Törn, A., Žilinskas, A. (1989) Global Optimization, Springer–Verlag, Lecture Notes in Computer Science, vol. 350.
33. Evtushenko, Y.G. (1985) Numerical Optimization Techniques. Translations Series in Mathematics and Engineering, Springer–Verlag, Berlin.
34. Sergeyev, Y.D. (1998) Global one-dimensional optimization using smooth auxiliary functions. Mathematical Programming 81, 127–146.
35. Gergel, V.P. (1996) A method of using derivatives in the minimization of multiextremum functions. Computational Mathematics and Mathematical Physics 36, 729–742.
36. Gaviano, M., Kvasov, D.E, Lera, D., and Sergeyev, Ya.D. (2003) Software for generation of classes of test functions with known local and global minima for global optimization. ACM Transactions on Mathematical Software 29(4), 469-480.

1. В силу исходных предположений о возможной многоэкстремальности характеристик из (1) [↑](#footnote-ref-1)
2. Точнее, минимизация *F*(*λ*,*y*) может приводить к получению слабо-эффективных вариантов (множество слабо-эффективных вариантов включает в себя область Парето). [↑](#footnote-ref-2)