УДК 519.893

Косолап А.И.

ОПТИМИЗАЦИЯ В КОНЕЧНОМЕРНОМ ЕВКЛИДОВОМ ПРОСТРАНСТВЕ

ГВУЗ «Украинский государственный химико-технологический университет», г. Днепр, Украина

В данной работе оптимизационные модели в евклидовом пространстве разделены на четыре классы сложности. Для решения задач первых двух классов сложности разработаны эффективные алгоритмы. Это прямо-двойственные методы внутренней точки. Дискретные и комбинаторные оптимизационные задачи третьего класса сложности рекомендуется преобразовывать к четвертому классу сложности с непрерывным изменением переменных. Для задач третьего и четвертого класса сложности в настоящее время не разработаны эффективные алгоритмы за исключением узкого класса задач, которые являются унимодальными. Общая задача оптимизации формулируется как минимум (максимум) целевой функции при наличии ограничений. Сложность задачи зависит от структуры целевой функции и ее ограничений. Если функции, определяющие оптимизационную модель являются квадратичными или полиномиальными, то для получения оценок решений в таких задачах может быть использовано полуопределенное программирования. Для задач полуопределенной оптимизации разработаны эффективные методы. Иногда достаточно разработать алгоритм без построения математической модели. Такой пример видим при сортировке массива чисел. Разработаны эффективные алгоритмы для решения этой задачи. В работе для задач сортировки построена оптимизационная модель, и она совпадает с моделью задачи о назначениях. Из этого следует, что задача сортировки является унимодальной. Для решения мультимодальных задач эффективные алгоритмы не разработаны. В работе предлагается простой и эффективный алгоритм для оптимального распределения ресурсов в многопроцессорных системах. Эта задача является мультимодальной. В общем случае, для решения мультимодальных задач предлагается метод точной квадратичной регуляризации. Этот метод доказал свою сравнительную эффективность при решении многих тестовых задач различной размерности.

Ключевые слова: евклидовое пространство, оптимизация, унимодальные задачи, мультимодальные задачи, классы сложности, численные методы.

DOI: 10.32434/2521-6406-2020-1-7-20-28

Постановка проблемы

В последние годы оптимизация находит широкое применение в любой сфере человеческой деятельности. Это экономика, финансы, технологические производства, управление, проектирование, информатика и многие другие области. Оптимизация, как выбор наилучших решений, необходима везде. Сейчас оптимизацию начали относить к междисциплинарной области исследований. Она включает две составляющие: построение оптимизационных моделей систем и их численное решение. Существуют классы оптимизационных моделей и исследуемую систему можно описать той или иной оптимизационной моделью. Большинство моделей включают целевую функцию, зависящую от многих переменных. Необходимо найти значение этих переменных, при которых целевая функция будет принимать экстремальное (минимальное или максимальное) значение. Обычно на изменение переменных накладываются ограничения, которые задаются неравенствами или равенствами по отношению к некоторым функциям. Эти функции непрерывны и, как правило, дифференцируемые. Дифференцируемость упрощает построение эффективных численных алгоритмов поиска экстремальной точки. Например, очень многие системы могут быть описаны оптимизационными моделями с квадратичными функциями, которые бесконечно раз дифференцируемые. Это упрощает построение эффективных алгоритмов.

Оптимизационные модели можно разделить на классы по их сложности. Самыми простыми являются линейные модели. Такие модели могут быть решены практически для любой размерности пространства (размерность равна числу переменных модели). Следующим классом являются выпуклые оптимизационные модели, в которых целевая функция является выпуклой функцией при условии ее минимизации и допустимое множество изменения переменных моделей также выпуклое. Для этого класса построены эффективные полиномиальные алгоритмы. Такие модели также могут быть решены практически для любых размерностей задачи. Следующий класс дискретных моделей, в которых переменные могут принимать только дискретные значения. Например, для многих комбинаторных задач оптимизационные модели содержат булевы переменные, которые могут принимать только два значения нуль или единица. Эта простота значительно усложняет поиск экстремальной точки. За исключением небольшого класса задач, эти модели можно отнести к третьему классу, для которого не разработаны эффективные алгоритмы поиска экстремальной точки. Существующие алгоритмы имеют экспоненциальную сложность. Это означает, что время решения таких задач будет расти экспоненциально при увеличении размерности задачи. Наконец, четвертый класс образуют невыпуклые оптимизационные модели. В этих моделях некоторые из функций могут быть невыпуклыми. Это порождает многоэкстремальность моделей, иногда их называют мультимодальными моделями. Существующие методы для решения таких задач позволяют находить только точки локального экстремума. Они не позволяют ответить на вопрос: найдено лучшее решение или только локально лучшее? Оптимизационная модель может содержать только выпуклые функции и относится к четвертому классу сложности. Это будет тогда, когда максимизируют выпуклую функцию или ограничения имеют вид равенств. Четвертый класс является наиболее большим и большинство реальных систем описываются моделями этого класса. Поэтому актуальной является задача разработки эффективных алгоритмов для третьего и четвертого класса моделей.

Цель статьи

Целью настоящей работы является разработка эффективных алгоритмов для задач оптимизации третьего и четвертого класса сложности в конечномерном евклидовом пространстве.

Анализ последних исследований и публикаций

В настоящее время оптимизация является достаточно обширной областью исследований. Энциклопедия по оптимизации насчитывает 4646 стр. [1]. Укажем лучшие книги по оптимизации задач первых двух классов сложности [2,3]. Для этих задач разработано программное обеспечение, которое входит, практически во все математические пакеты, например, MathLab, Maple и другие [4,5]. Задачи дискретной и комбинаторной оптимизации рассматриваются в книгах [6,7]. В последние годы для решения этих задач используется полуопределенное программирование [8—10], которое преобразует дискретные задачи к квадратичным. Для задач четвертого класса сложности также разработано большое число алгоритмов [11,12]. В последние годы преобладают генетические и эволюционные алгоритмы, основанные на случайном поиске. С увеличением размерности задачи эффективность таких алгоритмов сильно снижается. Конкуренцию этим алгоритмам составляет метод точной квадратичной регуляризации [13].

Изложение основного материала

Любой набор всех переменных модели (их фиксированные значения) определяет точку в многомерном пространстве. Для того, чтобы множество точек было пространством в нем задают метрику (расстояние между точками). Различные метрики определяют разные пространства. Естественной метрикой является евклидова, которая определяет расстояние между двумя точками как длину отрезка, соединяющего эти точки. Большинство оптимизационных моделей рассматривают в евклидовом пространстве. Структура таких моделей является более простой.

После построения оптимизационной модели выбирается алгоритм для ее решения. Необходимо выбрать такой алгоритм, чтобы решить задачу за минимальное компьютерное время. Это не простая задача, так как число разработанных алгоритмов для решения оптимизационных задач очень большое. Иногда проще не строить математическую модель, а сразу находить эффективный алгоритм решения задачи. В этом случае говорится о компьютерном моделировании, так как алгоритм реализуется в виде компьютерной программы. Рассмотрим пример упорядочения массива n чисел a_i по возрастанию. Разработаны простые и эффективные алгоритмы такого упорядочения без построения оптимизационной модели. Построим теперь для этой задачи оптимизационную модель. Имеем:

$$\sum_{j=1}^{n} a_j x_{ij} \le \sum_{j=1}^{n} a_j x_{i+1j}, \quad i = 1,\ldots, n-1;$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \sum_{j=1}^{n} x_{ij} = 1, \; x = 0 \vee 1.$$

Здесь переменные x принимают только булевы значения. Заметим, что сумма

$$\sum_{i=1}^{n-1} (a_{i+1} - a_i) = a_n - a_1$$

принимает максимальное значение, если последовательность чисел a_i упорядочена по возрастанию. Однако она будет максимальной и для других упорядочений, которые не меняют a_1 и a_n. Поэтому рассмотрим следующую сумму

$$\sum_{i=1}^{n-1} (a_{i+1} - a_1).$$

Эта сумма будет максимальной на упорядоченной по возрастанию последовательности чисел a_i. Действительно, любая перестановка чисел приведет к уменьшению, по крайней мере, одного слагаемого рассмотренной сумы. Это приводит к следующей оптимизационной задаче

$$\max\{\sum_{i=1}^{n-1} \left(\sum_{j=1}^{n} a_j x_{i+1j} - \sum_{j=1}^{n} a_j x_{1j} \right) | \sum_{i=1}^{n} x_{ij} = 1,$$

$$\sum_{j=1}^{n} x_{ij} = 1, x = 0 \vee 1\}. \qquad (1)$$

Задача (1) совпадает с задачей о назначениях

$$\max\{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \,|\, \sum_{i=1}^{n} x_{ij} = 1,$$

$$\sum_{j=1}^{n} x_{ij} = 1, x = 0 \vee 1\}.$$

Для решения задачи (1) разработаны алгоритмы с числом операций порядка $O(n^3)$. В то время как для сортировки n чисел разработаны алгоритмы с числом итераций порядка $O(n\log(n))$. Таким образом, задача сортировки чисел является частным случаем задачи о назначениях. Соответствующие коэффициенты c_{ij} для задачи (1) будут линейно зависимы.

Приведем еще один пример задачи, для которой непосредственное построение алгоритма является более предпочтительным. Рассмотрим задачу распределения заданий в многопроцессорной вычислительной системе. Такая система будет работать эффективно при равномерной загрузке всех процессоров. Будем предполагать, что имеется m процессоров и n заданий, а также время выполнения каждого задания t_j, $j=1,\ldots,n$. Оптимизационная модель этой задачи будет иметь вид:

$$\min\{\sum_{i=1}^{n} \left(\sum_{j=1}^{m} t_j x_{ij} \right)^2 | \sum_{j=1}^{m} x_{ij} = 1,$$

$$i = 1,\ldots,n, \; x = 0 \vee 1\}, \qquad (2)$$

где переменные принимают только булевы значения

$$x_{ij} = \begin{cases} 1, & \text{если i-е задание} \\ & \text{выполняется j-м процессором;} \\ 0, & \text{в противном случае.} \end{cases}$$

Целевая функция этой задачи равна сумме квадратов времен работы каждого процессора. Так как время обработки всех заданий постоянно, то разбиение этого времени на сумму квадратов слагаемых будет минимальным, если эти слагаемые равны. Это соответствует равномерной загрузке процессоров. Задача (2) содержит булевы переменные. Такие задачи решаются методом ветвей и границ. Однако даже для небольших размерностей m=6, n=24 время решения исчислялось часами, и при этом не всегда решение было точным.

Были рассмотрены другие модели этой задачи. Эта комбинаторная задача преобразуется к квадратичной мультимодальной задаче

$$\min\{\sum_{i=1}^{n} \left(\sum_{j=1}^{m} t_j x_{ij} \right)^2 | \sum_{j=1}^{m} x_{ij} = t_j,$$

$$i = 1,\ldots,n, \; \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij}^2 \ge \sum_{j=1}^{m} t_j^2\}. \qquad (3)$$

В этой модели переменные x_{ij} равны времени выполнения i-го задания на j-м процессоре. Последнее ограничение исключает возможность выполнения одного задания разными процессорами. Действительно, если разбить время выполнения какого-то задания на части, то сумма квадратов этих частей уменьшится и последнее ограничение задачи (3) будет нарушено. Задача (3) мультимодальная, что связано с последним невыпуклым ограничением. Для ее решения использовался эволюционный поиск, и он показал лучшие результаты по сравнению с методом ветвей и границ для задачи (2) как по времени решения, так и по его точности. Еще лучшие результаты были получены при использовании метода точной квадратичной регуляризации для решения задачи (3).

Далее, для задачи (3) был разработан простой алгоритм распределения заданий по процессорам, эффективность которого превзошла все ожидания. Этот алгоритм включает следующую последовательность шагов.

Шаг 1. Упорядочим последовательность времен t_j по убыванию, разобьём ее на части по m заданий в каждой части (последняя часть может быть неполной) и пронумеруем процессоры.

Шаг 2. Будем назначать каждые m заданий части процессорам в порядке возрастания их номеров для нечетной части заданий и в порядке убывания номеров процессоров для четной части заданий.

Шаг 3. После распределения всех заданий определяем процессоры с максимальной и минимальной загрузкой. Посредством обмена заданиями минимизируем разность времен их работы. Этот процесс повторяем до тех пор, пока уравнивание станет невозможным.

Программная реализация этого алгоритма показала лучшие результаты при решении многих задач. Преимущество было как по равномерности загрузки процессоров, так и по времени решения, которое осуществлялось за доли секунды.

Можно привести и другие примеры задач, когда непосредственная разработка алгоритма дает лучшие результаты. Несколько таких алгоритмов разработано для сетевых задач. Недостатком непосредственной разработки алгоритмов является то, что такие алгоритмы часто эвристические и не гарантируют получение наилучшего решения. Построение таких алгоритмов сложнее построения оптимизационной модели. Поэтому для оптимизационных моделей разработано большое число методов и программ, готовых для решения конкретной задачи. Причем число таких методов и программ постоянно растет.

Остановимся на методах решения оптимизационных задач. В общем случае эти задачи можно записать в виде

$$\min\{f_0(x) \mid f_i(x) \le 0,\ i = 1,\ldots,m,\ x \in E^n\},$$

где x – искомый n-мерный вектор (точка) евклидового пространства E^n.

Часть этих методов, которые излагаются в университетских курсах, уже давно не используются в практических приложениях. Например, методы проекции градиента или приведенного градиента малоэффективны. Такие методы как градиентного спуска или Франка-Вулфа были разработаны очень давно, но в последние годы модифицируются. Рассмотрим наиболее эффективные методы решения перечисленных выше классов моделей. Для линейных моделей вне конкуренции остаются симплекс-метод и прямо-двойственный метод внутренней точки. Задачи выпуклой оптимизации разбивают на два подкласса безусловной и условной оптимизации. Для безусловной оптимизации наиболее эффективными являются методы сопряженных направлений и квазиньютоновские методы. Для условной оптимизации лучшие результаты показывают прямо-двойственные методы внутренней точки. Для класса дискретных оптимизационных задач лучшим является метод ветвей и границ. На наш взгляд задачи этого класса следует преобразовывать к задачам четвертого класса с непрерывными переменными. Такое преобразование осуществить очень просто. Например, булевы переменные можно заменить квадратичным неравенством

$$\sum_{i=1}^{n} x_i (1 - x_i) \le 0,\ 0 \le x \le 1,$$

которому удовлетворяют только булевы переменные. Пример другого преобразования показан выше, при преобразовании задачи (2) к задаче (3).

Оптимизационные задачи четвертого класса представляют наибольший интерес, так как большинство прикладных задач относится к этому классу. Достаточно большим подклассом является класс общих квадратичных задач, которые формулируются следующим образом:

$$\min\{x^T A_0 x + b_0^T x \mid x^T A_i x + b_i^T x + c_i \le 0,\ i = 1,\ldots,m\}. \quad (4)$$

Здесь неизвестная переменная $x \in E^n$ – точка n-мерного евклидового пространства, все матрицы A_i – симметричные. Если все матрицы A_i – положительно полуопределенные, то задача (4) становится выпуклой. Существует большое число прикладных задач, которые могут быть описаны моделью (4), в частности, рассмотренная выше задача (3) является квадратичной. Задача (4) может иметь большое число локальных минимумов, например, 2^n. Эффективность решения задачи (4) локальным методом, например, прямо-двойственным методом внутренней точки, зависит от выбора начальной точки. Если эта точка выбрана вблизи точки глобального минимума, то задача (4) будет решена. Однако проверить это практически невозможно. Тем не менее, первыми методами для решения задач четвертого класса и задач (4), в частности, были методы мультистарта. В этих методах выбиралось большое число начальных точек, покрывающих допустимую область задачи, и для каждой такой точки решалась задача локальным методом. Среди найденных решений выбиралось наилучшее. Однако такой метод в n-мерном пространстве является неэффективным. Для многих начальных точек будем получать одно и то же решение, возможность покрыть достаточно плотно допустимое множество уже для 10-мерного пространства невозможно. Практические задачи требуют решения в пространстве размерности сотен и десятки тысяч переменных. Кроме того, допустимая область задачи часто имеет сложную структуру и поиск даже допустимой точки в таких задачах является проблемным. В дальнейшем метод мультистарта был модифицирован и последовательность начальных точек выбиралась случайным образом в соответствии с некоторым алгоритмом. Распространение получили генетические, эволюционные алгоритмы и другие подобные алгоритмы, которые используют результаты предыдущих итераций для поиска новых начальных точек. Для проверки эффективности этих алгоритмов было разработано большое число (больше тысячи) сложных мультимодальных тестовых задач. Большинство этих задач имеют известную точку глобального минимума. Для некоторых таких мультимодальных задач новые методы показали хорошие результаты. Однако для многих задач эти результаты были далеки от оптимальных. Можно приспособить эти методы для одних тестовых задач, но они не будут работать для других. Несмотря на продолжающиеся исследования в этой области прогресс в этом направлении маловероятен.

Особенностью мультимодальных задач является то, что для них не существует подклассов с простыми решениями. Одним из общих подходов для решения задач (4) является полуопределенная оптимизация. Задачу (4) можно записать в виде:

$$\min\{Q_0 \bullet X \mid Q_i \bullet X \le 0,\ i = 1,\ldots,m,\ X \succeq 0\}, \tag{5}$$

где неизвестной является переменная полуопределенная матрица X ранга единица. Обозначение $C \bullet X$ означает скалярное произведение двух матриц

$$C \bullet X = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}\,.$$

Все матрицы Q_i – симметричные и выражаются через параметры задачи (4) следующим образом:

$$X = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix},\quad Q_0 = \begin{pmatrix} 0 & \dfrac{b_0^T}{2} \\ \dfrac{b_0}{2} & A_0 \end{pmatrix},\quad Q_i = \begin{pmatrix} c_i & \dfrac{b_i^T}{2} \\ \dfrac{b_i}{2} & A_i \end{pmatrix}.$$

Если опустить условие того, что ранг матрицы равен единице, то задача (5) станет выпуклой. В этом случае, ее решение будет определять решение задачи (4), если ранг оптимальной матрицы X равен единица. В противном случае, будет получена нижняя оценка решения задачи (4). В работе [8] показано как эта нижняя оценка может быть улучшена. В частности, решение задачи (5) может быть использовано в качестве начальной точки для решения задачи (4) локальным методом. Для решения задачи (5) разработаны эффективные методы, это – прямо-двойственный метод внутренней точки и полуопределенный симплекс-метод. Решение задачи (5) будет достигаться на границе допустимой области, что обусловлено линейной целевой функцией по отношению матрицы X. Это означает, что решение задачи (5) будет достигаться на границе допустимой области, в частности на границе полуопределенного конуса $X \succeq 0$. Образующими этого конуса являются матрицы ранга единица. Поэтому, если решение достигается на одной из этих образующих, то получаем точное решение. Вычислительные эксперименты показывают, что во многих случаях получаем нижние оценки далекие от оптимальных. При переходе от задачи (4) к задаче (5) увеличивается размерность задачи. Например, простые ограничения $x_i \geq 0$ для каждого i необходимо заменить матричным ограничением. Если размерность задачи (4) n=100, то в задачу (5) необходимо добавить 100 матриц размера $(n+1)\times(n+1)$. Тем не менее, в последние годы для решения задач (5) используется преимущественно полуопределенная оптимизация. Она не ограничивается задачами (5) и используется также для полиномиальных задач.

Одной из проблем в разработке эффективных методов для решения мультимодальных задач является сложность структуры ограничений. Например, невыпуклая область

$$\{x | 4(x_2+1)\cos(x_1)+3(x_1+1)\cos(x_2)\leq 4\},$$

изображенная на рис. 1, является достаточно сложной для поиска оптимальной точки.

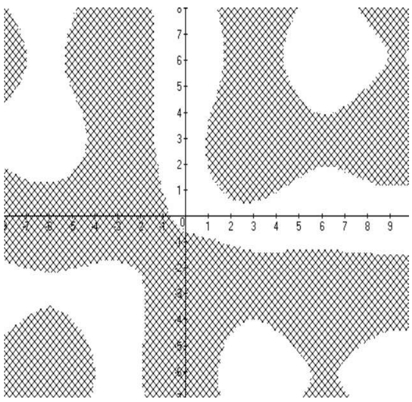

Рис.1. Невыпуклая допустимая область

Однако после квадратичной регуляризации

$$\{x \,|\, 4(x_2+1)\cos(x_1)+3(x_1+1)\cos(x_2)+$$
$$+10(x_1^2+x_2^2)\leq 100\}$$

она становится выпуклой (рис. 2).

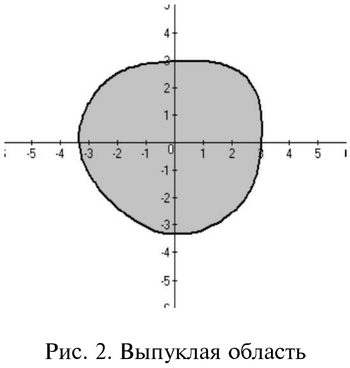

Рис. 2. Выпуклая область

Допустимую точку в выпуклой области найти очень легко, что нельзя сказать о предыдущей области.

Квадратичная регуляризация используется для преобразования задач оптимизации четвертого класса к виду:

$$\max\{\| x\|^2 | x\in S\}, \tag{6}$$

где $\|x\|^2$ – квадрат евклидовой нормы вектора, а S – выпуклое множество. Для задачи (5) преобразованная задача будет иметь вид:

$$\max\{| x \|^2 | \, x^T A_0 x + b_0^T x + s + (r-1)\| x \|^2 \leq d,$$

$$x^T A_i x + b_i^T x + c_i + r\| x \|^2 \leq d,\ i=1,...,m\}, \tag{7}$$

где параметр r>0 выбирается таким, чтобы допустимое множество задачи (7) было выпуклым, а параметр s таким, чтобы первое ограничение задачи (7) в точке максимума было активным. Норма вектора $\|x\|$ содержит n+1 переменную. Задача (7) содержит еще одну новую переменную d, значение которой находим методом дихотомии. Последовательно увеличивая значение d, для каждого его значения решаем задачу (7) прямо-двойственным методом внутренней точки. Найденную точку x проверяем на выполнимость условия $r\|x\|^2=d$. Если это условие выполняется с заданной точностью и значение d минимально, то решение задачи (7) совпадает с решением задачи (5).

Задача (6) в общем случае мультимодальная, но для некоторых выпуклых множеств S она преобразуется к унимодальной. Это будет тогда, когда множество S – правильный многогранник, прямоугольный параллелепипед, вписанный в шар многогранник, либо в каждой точке выпуклой поверхности S ее кривизна больше кривизны шара с центром в начале координат. Сегодня метод точной квадратичной регуляризации является лучшим выбором при решении мультимодальных задач, что подтверждается многочисленными сравнительными экспериментами.

Рассмотрим практику решения мультимодальных задач методом точной квадратичной регуляризации на примере задачи (5). Выбираем значение параметра r>0 для которого все матрицы $A_i+(r-1)I$ будут положительно определенные. Для этого достаточно, чтобы они были с преобладающей главной диагональю. Параметр s должен удовлетворять условию $s i\|x^*\|^2-f_0(x^*)$, где x^* – решение задачи (5). Так как решение x^* задачи (5) неизвестно, то необходимо произвести оценку значения s, которая будет уточняться в ходе решения задачи. Далее воспользуемся программой локальной оптимизации. Такие программы есть в математических пакетах, например MathLab или Maple. В открытом доступе в Internet есть программа OpenSolver, являющаяся надстройкой Excel. Для начала решения задачи (7) потребуется минимальное значение переменной d, которое находим, решая задачу выпуклой оптимизации заданной программой

$$\min\{d \mid x^T A_0 x + b_0^T x + s + (r-1)\|x\|^2 \le d,$$
$$x^T A_i x + b_i^T x + c_i + r\|x\|^2 \le d,$$
$$i = 1,\dots,m,\ r\|x\|^2 \le d\}.$$

Если в найденном решении x^0, d_0 последнее ограничение является активным, т.е. $r\|x^0\|^2=d_0$, то x^0 – решение задачи (5) при условии, что параметр s удовлетворяет приведенным выше условиям. Если же $r\|x^0\|^2<d_0$, то значение d необходимо последовательно увеличивать с определенным шагом до достижения условия $r\|x^k\|^2=d_k$, где x^k – решение задачи (7) при $d=d_k$. Как правило, с ростом значения d последовательность $f_0(x^k)$ убывает. Ее возрастание возможно, в таких случаях необходимо уменьшить шаг изменения переменной d. Такая процедура изменения d называется дихотомией. Процесс изменения d и решение задачи (7) локальным поиском продолжается до выполнения условия $r\|x^k\|^2=d_k$ с заданной точностью. Эффективность рассмотренного алгоритма можно увеличить посредством смещения допустимой области задачи вдоль биссектрисы положительного ортанта

$$\min\{f_0(x-h) \mid f_i(x-h) \le 0,\ i=1,\dots,m,\ x \in E^n\},$$

где h>0. При таком смещении кривизна поверхности выпуклого множества S не меняется, а кривизна шара убывает, что приводит к уменьшению модальности задачи.

Эффективность рассмотренного метода точной квадратичной регуляризации подтверждается многочисленными сравнительными экспериментами на решении сложных тестовых задач. Приведем пример решения задачи Rana в 100-мерном евклидовом пространстве:

$$\min\Bigg\{\sum_{i=1}^{n-1}\Big[\left(x_{i+1}+1\right)\cos\left(\sqrt{|x_{i+1}-x_i+1|}\right)\times$$
$$\times\sin\left(\sqrt{|x_{i+1}+x_i+1|}\right)+x_i\cos\left(\sqrt{|x_{i+1}+x_i+1|}\right)\times$$
$$\times\sin\left(\sqrt{|x_{i+1}-x_i+1|}\right)\Big] \,\Big|\, -520 \le x \le 520\Bigg\}.$$

На этой задаче в течении десятков лет проверяются все новые алгоритмы. Лучший результат для этой функции $f(x^*)$=-41047,18, без указания точки минимума x^*, приведен в работе [14]. В тоже время метод точной квадратичной регуляризации позволил найти значительно лучшее значение минимума этой функции $f(x^*)$=-50855,784.

Выводы

В данной работе оптимизационные задачи разбиты на четыре класса сложности. Для первых двух классов разработаны эффективные методы поиска экстремума. Дискретные задачи рекомендуется преобразовывать к непрерывным квадратичным задачам. Для решения задач четвертого класса сложности предлагается метод точной квадратичной регуляризации. Для некоторых мультимодальных задач, например, задач оптимального распределения ресурсов в многопроцессорных системах в данной работе построены простые и эффективные алгоритмы без построения оптимизационной модели.

СПИСОК ЛИТЕРАТУРЫ

1. Encyclopedia of Optimization / Editors C.A. Floudas, P.M. Pardalos. – Springer, 2009. – 4646 p.

2. *Luenberger D.G., Ye Y.* Linear and nonlinear programming. – Springer, 2008. – 546 p. https://doi.org/10.1007/978-0-387-74503-9

3. *Nocedal J., Wright S.J.* Numerical optimization. – Springer, 2006. – 685 p.

4. *Ye Y.* Optimization in Practice with MATLAB for Engendering students and Professionals. – Cambridge University Press. – 2015. – 469 p.

5. *Venkataraman P.* Applied Optimization with MATLAB Programming. Second Edition. – John Wiley & Sons, INC, Hoboken, 2015. – 524 p.

6. *Ye Y.* Semidefinite programming. – Stanford University, 2003. – 161 p.

7. *Ding Y.* On Efficient Semidefinite Relaxations for Quadratically Constrained Quadratic Programming. – Waterloo: Ontario, Canada. – 2007. – 68 p.

8. *Косолап А.И., Перетятько А.С.* Полуопределенное программирование и его приложения. – Днепр: ПГАСА, 2018. – 148 с.

9. *Korte B., Vygen J.* Combinatorial Optimization Theory and Algorithms. Third Edition. – Springer, 2006. – 595 p.

10. *Chen D.S., Batson R.G., Dang Y.* Applied Integer Programming. Modeling and Solution. – John Wiley & Sons, INC, Hoboken. – 2010. – 462 p.
https://doi.org/10.1002/9781118166000

11. *Horst R., Tuy H.* Global Optimization: Deterministic Approaches; 3rd ed., Berlin: Springer–Verlag, 1996. – 727 p.
https://doi.org/10.1007/978-3-662-03199-5

12. *Kenneth V.P., Storn R.M., Lampinen J.A.* Differential Evolution. A Practical Approach to Global Optimization. – Berlin Heidelberg: Springer-Verlag, 2005. – 542 p.

13. *Косолап А.И.* Глобальная оптимизация. Метод точной квадратичной регуляризации. – Д.: ПГАСА, 2015. – 164 с.

14. *Jamil M., Yang X.S.* A literature survey of benchmark functions for global optimization problems // Int. J. Math. Model Numer. Optim. – Vol. 4. – № 2. – 2013 – P.150-194.
https://doi.org/10.1504/IJMMNO.2013.055204

Поступила в редакцию 30.06.2020

ОПТИМІЗАЦІЯ В СКІНЧЕННОВИМІРНОМУ ЕВКЛІДОВОМУ ПРОСТОРІ

Косолап А.І.

У даній роботі оптимізаційні моделі в евклідовому просторі розділені на чотири класи складності. Для розв'язування задач перших двох класів складності розроблені ефективні алгоритми. Це прямо-двоїсті методи внутрішньої точки. Дискретні і комбінаторні оптимізаційні задачі третього класу складності рекомендується перетворювати до четвертого класу складності з безперервними змінними. Для задач третього і четвертого класу складності в даний час не розроблені ефективні алгоритми за винятком вузького класу задач, які є унімодальними. Загальна задача оптимізації формулюється як мінімум (максимум) цільової функції при наявності обмежень. Складність такої задачі залежить від структури цільової функції і її обмежень. Якщо функції, що визначають оптимізаційну модель є квадратичними або поліноміальними, то для отримання оцінок розв'язків в таких задачах може бути використано напіввизначене програмування. Для задач напіввизначеної оптимізації розроблені ефективні методи. Іноді досить розробити алгоритм без побудови математичної моделі. Такий приклад бачимо при сортуванні масиву чисел. Розроблено ефективні алгоритми для розв'язування цієї задачі. В роботі для задач сортування побудована оптимізаційна модель, і вона співпадає з моделлю задачі про призначення. З цього випливає, що задача сортування чисел є унімодальною. Для розвязування мультимодальних задач ефективні алгоритми не розроблені. У роботі пропонується простий і ефективний алгоритм, для оптимального розподілу ресурсів в багатопроцесорних системах. Ця задача є мультимодальною. У загальному випадку, для розв'язування мультимодальних задач пропонується метод точної квадратичної регуляризації. Цей метод довів свою порівняльну ефективність при розв'язуванні багатьох тестових задач різної розмірності.

Ключові слова: Евклідовий простір, оптимізація, унімодальна задача, мультимодальаі задача, класи складності, чисельні методи.

OPTIMIZATION IN A FINITE-DIMENSIONAL EUCLIDE SPACE

Kosolap A.I.

Ukrainian State University of Chemical Technology, Dnipro, Ukraine

In this paper, optimization models in Euclidean space are divided into four complexity classes. Effective algorithms have been developed to solve the problems of the first two classes of complexity. These are the primal-dual interior-point methods. Discrete and combinatorial optimization problems of the third complexity class are recommended to be converted to the fourth complexity class with continuous change of variables. Effective algorithms have not been developed for problems of the third and fourth complexity classes, with the exception of a narrow class of problems that are unimodal. The general optimization problem is formulated as a minimum (maximum) objective function in the presence of constraints. The complexity of the problem depends on the structure of the objective function and its feasible region. If the functions that determine the optimization model are quadratic or polynomial, then semidefinite programming can be used to obtain estimates of solutions in such problems. Effective methods have been developed for semidefinite optimization problems. Sometimes it's enough to develop an algorithm without building a mathematical model. We see such an example when sorting an array of numbers. Effective algorithms have been developed to solve this problem. In the work for sorting problems, an optimization model is constructed, and it coincides with the model of the assignment problem. It follows from this that the sorting problem is unimodal. Effective algorithms have not been developed to solve multimodal problems. The paper proposes a simple and effective algorithm for the optimal allocation of resources in multiprocessor systems. This problem is multimodal. In the general case, for solving multimodal problems, a method of exact quadratic regularization is proposed. This method has proven its comparative effectiveness in solving many test problems of various dimensions.

Keywords: Euclidean space, optimization, unimodal problems, multimodal problems, complexity classes, numerical methods.

REFERENCES

1. Encyclopedia of Optimization. Editors C.A. Floudas, P.M. Pardalos, Springer, 2009, 4646 p.

2. Luenberger D.G., Ye Y. Linear and nonlinear programming. Springer, 2008, 546 p.

https://doi.org/10.1007/978-0-387-74503-9

3. Nocedal J., Wright S.J. Numerical optimization. Springer, 2006, 685 p.

4. Ye Y. Optimization in Practice with MATLAB for Engendering students and Professionals. Cambridge University Press, 2015, 469 p.

5. Venkataraman P. Applied Optimization with MATLAB Programming. Second Edition, John Wiley & Sons, INC, Hoboken, 2015, 524 p.

6. Ye Y. Semidefinite programming. Stanford University, 2003, 161 p.

7. Ding Y. On Efficient Semidefinite Relaxations for Quadratically Constrained Quadratic Programming. Waterloo: Ontario, Canada, 2007, 68 p.

8. Kosolap A.I., Peretiatiко A.S. *Poluopredelennoe programmirovanie i ego prilozheniya* [Semi-Defined programming and its applications]. Dnepr: PGASA, 2018, 148 p. (in Russian).

9. Korte B., Vygen J. Combinatorial Optimization Theory and Algorithms. Third Edition. Springer, 2006, 595 p.

10. Chen D.-S., Batson R.G., Dang Y. Applied Integer Programming. Modeling and Solution. John Wiley & Sons, INC, Hoboken, 2010, 462 p. https://doi.org/10.1002/9781118166000

11. Horst R., Tuy H. Global Optimization: Deterministic Approaches; 3rd ed., Berlin: Springer-Verlag, 1996, 727 p. https://doi.org/10.1007/978-3-662-03199-5

12. Kenneth V.P., Storn R.M., Lampinen J.A. Differential Evolution. A Practical Approach to Global Optimization. Berlin Heidelberg: Springer-Verlag, 2005, 542 p.

13. Kosolap A.I. *Globalinaya optimizatsyia. Metod tochnoj кvadratichnoj reguliarizatsii* [Global optimization. The method of exact quadratic regularization]. Dnepr: PGASA, 2015, 164 p. (in Russian).

14. Jamil M, Yang X.S. A literature survey of benchmark functions for global optimization problems. Int. J. Math. Model Numer. Optim., 2013, vol. 4., no. 2, pp.150-194. https://doi.org/10.1504/IJMMNO.2013.055204