

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

МЕТОДЫ ОПТИМИЗАЦИИ И НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Методические указания
к выполнению лабораторного практикума
по дисциплине для магистрантов профилей
09.04.02 – “Информационные системы и технологии”
09.04.03 – “Прикладная информатика”
всех форм обучения



Севастополь
СевГУ
2019

УДК 004.1
М54

Р е ц е н з е н т ы:

Е.А. Кожаяев - кандидат техн. наук, доцент кафедры
информационно-вычислительной техники

К.В. Кротов - кандидат техн. наук, доцент кафедры информационных систем

Ответственный за выпуск:

И.П. Шумейко - заведующий кафедрой
информационных систем, к.ф.-м.н., доцент

Составители: Е.Н. Заикина, В.Ю. Карлусов

М54 Методы оптимизации и нелинейное программирование: метод. указания к выполнению лабораторно-вычислительного практикума по дисциплине к выполнению лабораторного практикума по дисциплине для магистрантов профилей 09.04.02 – “Информационные системы и технологии” и 09.04.03 – “Прикладная информатика” всех форм обучения / Разраб. Е.Н. Заикина, В.Ю. Карлусов. – Севастополь: СевГУ, 2019. – 47 с.

Цель методических указаний: обеспечить качественное усвоение теоретических положений курса для наиболее полного овладения математическим и вычислительным аппаратами, применяемыми в оптимизационных методах на основе процедур нелинейного программирования.

УДК 004.1

Методическое пособие рассмотрено и утверждено на заседании кафедры информационных систем, протокол № 10 от 24 мая 2019 г.

Методические рекомендации рассмотрены и рекомендованы к изданию на заседании Учёного совета Института информационных технологий и управления в технических системах, протокол № 06 от 18 июня 2019 г.

Содержание

Введение.....	4
Требования к оформлению и содержанию отчета.....	5
Лабораторная работа № 1. Исследование задач нелинейного программирования средствами пакета MatLab	6
Лабораторная работа № 2. Исследование алгоритма решения задач нелинейного программирования прямыми методами	9
Лабораторная работа № 3. Исследование алгоритма решения задач нелинейного программирования методом наискорейшего спуска	13
Лабораторная работа № 4. Исследование алгоритма решения задачи нелинейного программирования методом сопряжённого градиента	15
Лабораторная работа № 5. Исследование алгоритма решения задачи нелинейного программирования квазиньютоновскими методами	18
Лабораторная работа № 6. Исследование алгоритма решения задач нелинейного программирования методом Зойтендейка	21
Лабораторная работа № 7. Исследование алгоритма решения задач нелинейного программирования методом Розена	25
Лабораторная работа № 8. Исследование алгоритма решения задач нелинейного программирования методом штрафных функций внутренней точки	30
Лабораторная работа № 9. Исследование алгоритма решения задач нелинейного программирования методом штрафных функций внешней точки	33
Заключение.....	35
Библиографический список.....	36
Приложение А. Варианты систем ограничений для изучения алгоритмов решения задач нелинейного программирования ...	37
Приложение Б. Варианты функций цели	44
Приложение В. Результаты решения задачи квадратичного программирования с применением функции quadprog	46

ВВЕДЕНИЕ

Большинство практических задач, встречающихся в научной и производственной деятельности, имеют множество (часто бесконечное) решений и сложное математическое описание. Выбор наилучшего по принятому критерию решения представляется сложной оптимизационной задачей. Немаловажным инструментом научных исследований при решении оптимизационных задач являются методы нелинейного программирования.

В самом общем случае, задача нелинейного программирования (НП-задача) задаётся математической моделью [[1, 5 – 7, 12 – 14]]

$$\begin{cases} f(x_1, x_2, \dots, x_n) \rightarrow \min, \\ g_1(x_1, x_2, \dots, x_n) \geq 0, \\ g_2(x_1, x_2, \dots, x_n) \geq 0, \\ \dots \\ g_m(x_1, x_2, \dots, x_n) \geq 0. \end{cases}$$

где функции $f(X)$ и $g_i(X)$, $i = 1, m$, в общем случае, нелинейные.

Следует отметить, что универсального общеупотребимого метода решения указанных задач не существует.

В методических целях, чтобы обеспечить наглядное восприятие хода решения и контроль вычислительного процесса, сделаем следующие допущения:

- число переменных в модели положим равным двум ($n = 2$);
- в качестве целевой функции $f(X)$ будем использовать полином второй степени;
- функции $g_i(X)$, $i = 1, m$, выберем линейные.

Оные допущения,

- во-первых, нисколько не помешают изучению методов оптимизации;
- во-вторых, сведут общую постановку НП-задачи к задачам квадратичного программирования, для решения которых разработаны программы;
- в-третьих, результаты, полученные с использованием “фирменных” программ могут быть сопоставлены с ответами, являющихся решениями НП-задач, рассчитанных посредством исследуемых в ходе выполнения лабораторных работ алгоритмами.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ И СОДЕРЖАНИЮ ОТЧЁТА

Авторы считают целесообразным оформление отчета в отдельной тетради, которая по окончании изучения дисциплины сдается преподавателю. В этом случае,

- во-первых, результаты выполнения предыдущих работ доступны для анализа;
- во-вторых, наработанный материал может быть с пользой для дела употреблён на этапе подготовки к экзамену либо зачёту.

В общем случае, отчет по лабораторной работе должен содержать:

1. Титульный лист с указанием названия работы и исполнителя её.
2. Цель работы.
3. Математическую модель задачи согласно варианту задания.
4. Результаты предварительных расчетов и аналитических преобразований.
5. Отражение результатов, полученных в ходе вычислений с использованием РС.
6. Выводы, основанные на анализе полученных результатов и результатов предыдущих лабораторных работ.

Вариант задаётся в виде двух чисел:

- первое число, в диапазоне 00 – 99, определяет систему ограничений задачи;
- второе (00 – 49) – задаёт целевую функцию.

Оформленный отчёт защищается в устной форме, в ходе ответов на контрольные вопросы.

ЛАБОРАТОРНАЯ РАБОТА № 1 ИССЛЕДОВАНИЕ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ СРЕДСТВАМИ ПАКЕТА MATLAB

1. ЦЕЛЬ РАБОТЫ

1. Восстановить практические навыки в использовании среды MatLab, начать освоение пакета Optimization toolbox.
2. Восстановить теоретические и практические положения математического анализа применительно к решению задач нелинейного программирования (НП-задач).
3. Подготовить тестовый пример для сопоставления в ходе решения НП-задачи различными методами.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Задача квадратичного программирования, в своей канонической постановке записывается так [7, 11]

$$\begin{aligned} f(X) &= b^T X + \frac{1}{2} X^T C X \rightarrow \max, \\ AX &\leq A_0, X \geq 0, \end{aligned} \tag{1}$$

где C — симметричная отрицательно определённая матрица размерностью $[n \times n]$, b^T — вектор-строка размерностью $[1 \times n]$, A — матрица системы ограничений размерностью $[m \times n]$, A_0 — вектор свободных членов системы ограничений размерностью $[m \times 1]$, n — число переменных, m — число ограничений.

Если задача (1) посвящена определению минимума целевой функции, то матрица C должна быть определена положительно.

Положительную или отрицательную определённость матрицы можно оценить по критерию Сильвестра [4], задающему порядок чередования знаков минорных определителей в зависимости от типа определённости.

Существуют случаи, когда матрица C не определена ни положительно, ни отрицательно. В этом случае функция $f(X)$ в отсутствие ограничений, не будет иметь глобального экстремума.

Переменная часть слагаемого (1) функции цели, в котором задействована матрица C , называется квадратичной формой, и, в неvectorной записи, представима двойной суммой

$$X^T C X = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} \cdot x_i \cdot x_j. \quad (2)$$

Из (2) следует, что диагональные элементы матрицы $c_{i,i}$ являются коэффициентами при квадратах неизвестных, а недиагональные $c_{i,j} = c_{j,i}$ — половине коэффициента в произведении $x_i \cdot x_j$. С учётом множителя в выражении (1), имеем, соответственно, $0,5 \cdot c_{i,i} \cdot x_i^2$ и $c_{i,j} \cdot x_i \cdot x_j$.

Решение задачи (1) можно производить двояко: воспользоваться конструктивной теоремой квадратичного программирования, либо каким-нибудь подходящим по условиям применения методом.

Констатирующая часть теоремы квадратичного программирования выглядит так [7]

$$\begin{aligned} b + C \cdot X_0 - A^T \Lambda + V &= 0, \text{ а)} \\ A_0 - A X_0 - W &= 0, \quad \text{б)} \\ \left. \begin{aligned} V^T X_0 &= 0 \\ W^T \Lambda &= 0 \end{aligned} \right\} &\quad \text{в)} \end{aligned} \quad (3)$$

Компоненты всех векторов Λ , W и V — неотрицательны, а вектора W и V могут быть нулевыми. Условия (3, а) и (3, б) образуют систему из $n + m$ уравнений для $2 \times (n + m)$ неизвестных компонентов X , Λ , V и W . Ограничения (3, в) есть условия дополняющей нежёсткости.

Выражения (3, а и б) являются основанием для построения эквивалентной задачи линейного программирования, в ходе решения которой представляется возможным определить значения элементов всех неизвестных векторов X_0 , Λ , W и V и, после получения оптимальных значений ЗЛП, проверить условия дополняющей нежёсткости и неотрицательности.

1. Условия (3, а) и (3, б) необходимо представить в форме, обеспечивающей положительность элементов столбцов свободных членов

$$\begin{cases} A^T \Lambda - C \cdot X - V = b, \\ AX + W = A_0. \end{cases} \quad (4)$$

2. Поскольку знак в ограничениях (4) — равенство, следует воспользоваться методом искусственного базиса. Необходимо добавить искусственные переменные $\{y_i\}$ и $\{z_i\}$, при этом система ограничений примет вид

$$\begin{cases} A^T \Lambda - C \cdot X - V + Z = b, \\ AX + W + Y = A_0. \end{cases} \quad (5)$$

3. Конструируется псевдоцелевая функция из искусственных переменных

$$\sum_{i=1}^m \mu \cdot y_i + \sum_{j=1}^n \mu \cdot z_j \rightarrow \min. \quad (6)$$

4. Решается эквивалентная ЗЛП с функцией цели (6) при ограничениях (5).

Если в ходе решения ЗЛП векторы Y и Z будут выведены из базиса (достигнут оптимум), а полученные значения X , A , V и W не отрицательны, а так же удовлетворяют (3, в), то компоненты вектора X представляют собой оптимальное решение исходной задачи квадратичного программирования (1).

5. Рассчитывается значение целевой функции.

В пакете Optimization toolbox имеются функции, с помощью которых можно получить решение (1):

- `quadprog` – непосредственное решение (1) методом, на который имеется ссылка в документации MatLab'a;
- `linprog` – решение эквивалентной ЗЛП, построенной с помощью (5) и (6).

и выполнить демонстрацию результатов:

- `plot` – построение одномерных графиков;
- `meshgrid` – разметка поля аргументов для построения объёмного изображения;
- `mesh` – построение поверхности (есть и другие процедуры)
- `contour` – построение линий равного уровня.

3. ХОД РАБОТЫ

1. Получить вариант задания.

2. Пользуясь матричной формой, построить развёрнутую функцию цели.

3. Исследовать поведение целевой функции:

- выпукла (вогнута, с перегибом),
- имеет ли экстремум,
- располагается ли экстремум внутри или ввне области ограничений,
- рассчитать значения независимых переменных, обеспечивающих оптимальное значение целевой функции и само значение.

4. С использованием пакета Optimization toolbox рассчитать координаты решения задачи квадратичного программирования и выполнить необходимые графические построения.

5. Оформить отчет с приложением результатов вычислений, графической информации. Сделать *содержательный* вывод, оценивающий полученные результаты.

6. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая задача является задачей нелинейного программирования?
2. Какая точка называется экстремальной?
3. Чем условный минимум или максимум отличаются от безусловного?
4. Что такое градиент функции?
5. Какая точка называется стационарной?
6. Когда функция называется выпуклой?
7. Когда функция называется вогнутой?
8. Как построить матрицы Гессе для функции?
9. Как оценить выпуклость или вогнутость функции с использованием матрицы Гессе?
10. Для чего применяется Якобиан?
11. Каковы необходимые условия существования экстремальных значений функций?
12. Как найти наибольшее и наименьшее значения функции нескольких переменных в замкнутой ограниченной области?

ЛАБОРАТОРНАЯ РАБОТА № 2 ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ПРЯМЫМИ МЕТОДАМИ

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задачи нелинейного программирования.
2. Освоить аспекты применения функции `fminsearch` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить эвристические алгоритмы решения НП-задач без ограничений.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Решение НП-задач возможно разными методами. Исторически первыми возникли эвристические методы.

В данную группу входят прямые методы решения задачи, не использующие значения градиентов функций [10]. К таковым относятся метод траекторий Хука-Дживса и метод Розенброка. Эти методы применяются к оптимизации нелинейных функций без ограничений.

Идея метода Хука-Дживса состоит в следующем.

- Направления поиска ориентированы, так сказать, по сторонам света или вдоль осей координат в обе стороны.
- Вокруг текущей точки производится поиск направления, в котором функция убывает.
- Если такое направление найдено, то шаг поиска увеличивается, а поиск продолжается в выбранном направлении – устанавливается так называемый тренд поиска, и осуществляется до тех пор, пока функция в этом направлении убывает.
- Если факта убывания функции не обнаружено, то шаг поиска уменьшается.

Таким образом, делается попытка найти «овраг» (при минимизации) функции и двигаться вдоль него к точке минимума.

В задачи максимизации ищется направление возрастания функции и «хребет».

Алгоритм метода показан на рисунке 1.

Входные данные. Начальное значение шага λ , координаты начальной точки X_0 ускоряющий множитель α , точность нахождения решения ε , список возможных направлений поиска $S = \{s_1, s_2, \dots, s_n\}$. Для числа переменных равного двум, это орты: $s_1 = (0, 1)$ и $s_2 = (1, 0)$.

Условие окончания: значение текущего шага укладывается в точность, $\varepsilon \geq \lambda$.

Блок № 1. Координаты начальной точки переприсваивается текущей переменной: $Y = X_0$.

Блок № 2. Организуется цикл перебора направлений поиска $j = 1, n$, тело цикла составляют блоки №№ 3 – 7.

Сравнение в блоке № 3 проверяется факт убывания функции в выбранном направлении: $f(Y + \lambda \times s_j) < f(Y)$. Если функция убывает, выход «да» блока № 3, то текущая точка перемещается в выбранном направлении $Y = Y + \lambda \times s_j$ (блок № 4).

В противном случае, выход «нет» блока № 3, в блоке № 5 проверяется факт убывания функции в направлении, противоположном выбранному: $f(Y - \lambda \times s_j) < f(Y)$.

Если функция не убывает ни в выбранном, ни в противоположном направлениях (выход «нет», блок № 5), то выбирается очередное направление.

Если сравнение в блоке № 5 произошло удачно (выход «да»), то текущая точка решения перемещается в блоке № 6 по закону $Y = Y - \lambda \times s_j$.

Блок № 7 выполняет факультативное действие $\lambda = 2 \times \lambda$, которое увеличивает шаг, ускоряя тем самым поиск. В ряде реализаций алгоритма данное действие отсутствует.

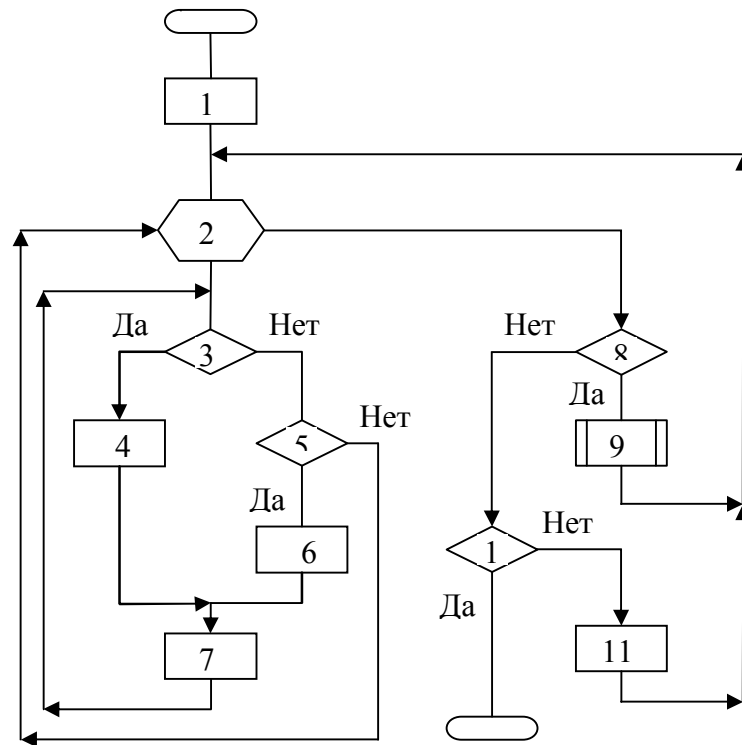


Рисунок 1 – Алгоритм метода Хука-Дживса

Блок № 8 достигается в ходе работы алгоритма, после перебора всех направлений поиска и попадания в ситуацию, когда при заданном значении λ в окрестностях точки X_0 убывания функции не обнаружено.

Блок № 9 выполняет пересчёт координат текущей точки по формуле

$$Y = X_0 + \alpha \times (Y - X_0); X_0 = Y.$$

Блок № 10 осуществляет проверку окончания: $\varepsilon \geq \lambda$? Если «да» то алгоритм заканчивает работу, если «нет», то шаг поиска уменьшается

$$\lambda = \frac{\lambda}{2},$$

после чего начинается поиск вокруг текущей точки X_0 с уменьшенным шагом.

В качестве недостатка отмечается, что при попадании в область размытого локального минимума есть риск остановиться, не достигнув глобального минимума.

3. ХОД РАБОТЫ

1. Реализуйте расчёты по алгоритму Хука-Дживса в среде MatLab для Вашего варианта функции, предусмотрите подсчёт числа итераций.

2. Воспользовавшись функцией `fminsearch` пакета Optimization toolbox среды MatLab выполнить поиск решения.

3. Сравнить между собой по точности и по числу итераций результаты работы алгоритмов Хука-Дживса, `fminsearch` и выполненных в предыдущей лабораторной работе расчётов по нахождению глобального экстремума.

4. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать **содержательный** вывод, оценивающий полученные результаты.

5. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как выглядит в общем виде модель для задачи нелинейного программирования?

2. Какие существуют типы задачи НП, чем они различаются?

3. Как формулируются необходимые условия оптимальности в задачах безусловной оптимизации?

4. Как следует понимать термины «выпуклая» и «вогнутая» функции применительно к задачам нелинейного программирования?

5. Что называется линией уровня функции двух переменных?

6. Почему метод получил ещё название метода траекторий?

7. Что является общей особенностью прямых методов?

8. Критичен ли метод к выбору начальной точки?

9. Почему прямые методы называют ещё и поисковыми?

10. В чём состоят достоинства метода Хука-Дживса?

11. В чём состоят недостатки метода Хука-Дживса?

12. Может ли данный алгоритм заикливаться? Обоснуйте Ваш ответ.

ЛАБОРАТОРНАЯ РАБОТА № 3 ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ МЕТОДОМ НАИСКОРЕЙШЕГО СПУСКА

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задачи нелинейного программирования методом наискорейшего спуска.
2. Освоить аспекты применения функций `fminunc` и `fminbnd` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить алгоритм метода наискорейшего спуска.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Метод *наискорейшего спуска* ещё называется методом *наискорейшего подъёма*.

При решении задачи на минимизацию [2] речь идёт о спуске, а в случае максимизации – о подъёме. В алгоритмах указанное отличие отражается в знаках при выборе направления, которое совпадает с направлением градиента в задачах максимизации и противоположно ему (знак «минус») в задачах минимизации.

Луи Огюст Коши предложил выбирать значение λ_k путём решения задачи одномерной минимизации из условия [3]

$$\lambda_k^* \Rightarrow \min_{\lambda > 0} f(X_k + \lambda s_k), \quad (1)$$

где $s_k = -\nabla f(X_k)$ определяется значением градиента. При этом непосредственно значение λ_k может быть определено из уравнения

$$\frac{\partial f(X_k + \lambda s_k)}{\partial \lambda} = 0 \quad (2)$$

аналитически, либо путём численного решения задачи одномерной минимизации. Алгоритм метода, с точки зрения вычислительной сложности, тривиальный, изображён на рисунке 1.

Если ввести вектор, определяемый как $\vec{P}_k = \vec{X}_{k+1} - \vec{X}_k$, то вектор-градиент $\nabla f(X_{k+1})$ перпендикулярен вектору приращения $\vec{P}_k$, то есть

$\nabla \vec{f}(X_{k+1}) \perp \vec{P}_k$. Поэтому траектория спуска будет иметь, в общем случае, вид зигзага.

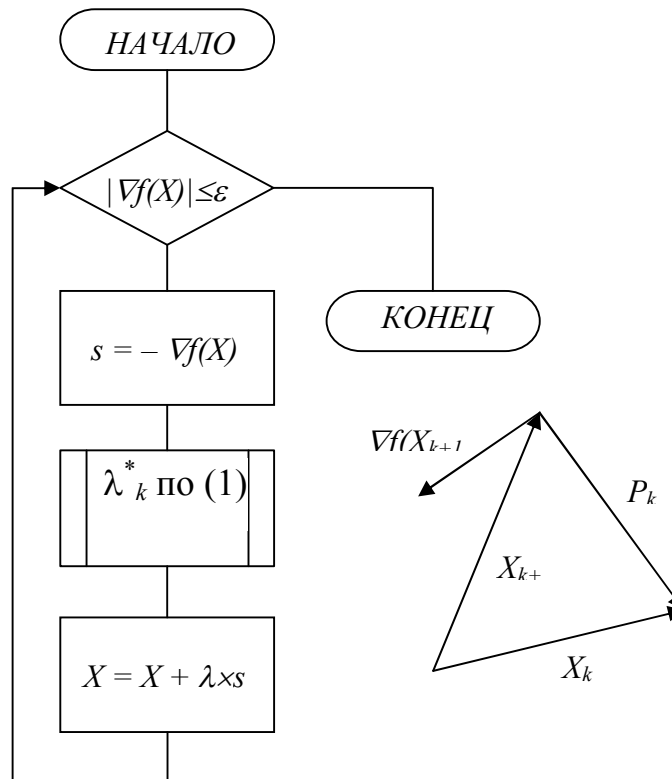


Рисунок 1 – Алгоритм метода наискорейшего спуска

Тут уместно заметить, что метод наискорейшего спуска **не обладает** оптимальными свойствами, **не обеспечивая минимум шагов** при минимуме **вычислений**, если только **направление спуска не совпадает** с направлением на минимум, локальный или глобальный.

Последнее справедливо только для ограниченного класса функций, поэтому метод считается **полношаговым**.

3. ХОД РАБОТЫ

1. Для заданного варианта найти значение градиента функции аналитически.

2. Реализовать две версии алгоритма в части нахождения λ^* :

- с применением аналитического решения (2) и
- с применением численного решения (2) используя алгоритм одномерной минимизации fminbnd.

3. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.

4. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать *содержательный* вывод, оценивающий полученные результаты.

5. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое градиент функции нескольких переменных?
2. Как определить составляющие вектора градиента?
3. Когда будет происходить наискорейший спуск, а когда – полношаговый?
4. Каковы основные свойства градиента функции в некой точке?
5. В чем заключается градиентные методы при поиске минимума функции?
6. Как определяется шаг поиска в градиентных методах?
7. Каковы будут значения градиента в точке глобального минимума?
8. Как адаптировать алгоритм метода наискорейшего спуска для поиска максимального значения функции многих переменных?
9. Как, не меняя программу, реализующую алгоритм наискорейшего спуска, найти точку максимума функции многих переменных?
10. Всегда ли метод наискорейшего спуска найдёт точку глобального минимума при наличии хотя бы одного локального?
11. Может ли данный работать бесконечно?

ЛАБОРАТОРНАЯ РАБОТА № 4 ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ МЕТОДОМ СОПРЯЖЁННОГО ГРАДИЕНТА

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задачи нелинейного программирования методом сопряжённого градиента.
2. Закрепить способы применения функцию `fminunc` и `fminbnd` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить алгоритм метода сопряжённого градиента.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

В общем случае, система линейно-назависимых направлений поиска $S_1, S_2, \dots, S_{n-1}$ называется *сопряжённой по отношению к некоторой положительно определённой матрице Q* если

$$S_i^T Q S_j = 0; \quad i \neq j; \quad i = \overline{1, n-1}, \quad j = \overline{1, n-1}.$$

Направление спуска S_{n+1} будет сопряжено направлению S_n , когда выполняется тождество:

$$S_n^T \nabla^2 f(x_n) S_{n+1} = 0.$$

Как известно из математики [4], для положительно определённой матрицы её собственные вектора образуют систему из $n - 1$ направлений и такая система всегда существует, по крайней мере, одна.

Пусть исходное направление спуска (антиградиент) в некоторой точке X_n есть

$$S_n = -\nabla f(x_n),$$

а сопряжённое направление построено по алгоритму

$$S_{n+1} = -\nabla f(x_{n+1}) + \omega_{n+1} S_n,$$

где ω_{n+1} – весовой коэффициент сопряжения. Этот коэффициент должен обеспечивать условие сопряжённости, то есть выполняться (1).

Коэффициент сопряжения, как показано в [2, 9, 10], равен отношению квадратов норм (длин) векторов текущего и предшествующего градиентов:

$$\omega_{n+1} = \frac{\nabla^T f(x_{n+1}) \cdot \nabla f(x_{n+1})}{\nabla^T f(x_n) \cdot \nabla f(x_n)} = \frac{\|\nabla f(x_{n+1})\|^2}{\|\nabla f(x_n)\|^2}.$$

Алгоритм расчётов, который назван по имени разработчиков алгоритмом Флетчера – Ривса, представлен ниже, на рисунке 1. Шаг λ^* отыскивается по процедуре, как и в методе наискорейшего спуска.

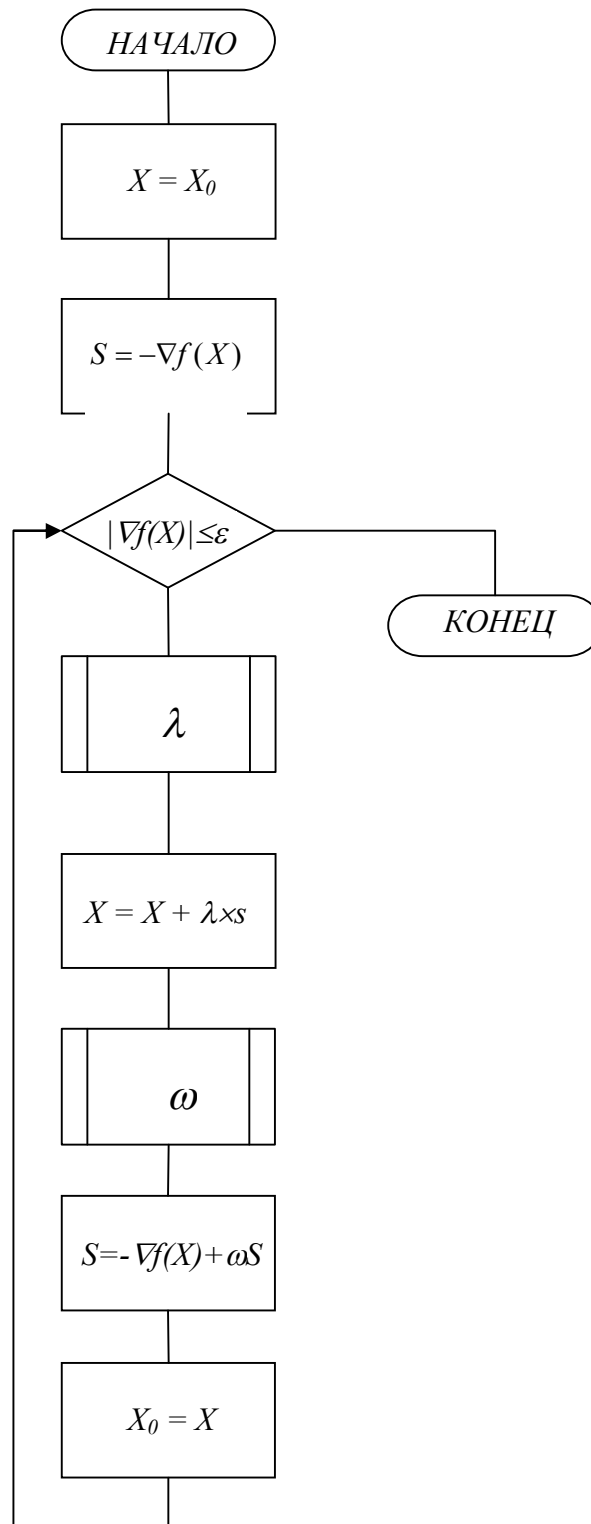


Рисунок 1 – Алгоритм метода Флетчера – Ривса

3. ХОД РАБОТЫ

1. Реализовать две версии алгоритма в части нахождения λ^* :
 - с применением аналитического нахождения шага и

- с применением численного нахождения шага используя алгоритм одномерной минимизации `fminbnd`.

2. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.

3. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать *содержательный* вывод, оценивающий полученные результаты.

4. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое сопряжённые направления?
2. Как называются векторы, если матрица, относительно которых они сопряжены нулевыми с единицами по главной диагонали?
3. Каковы свойства градиента, используемые для отыскания минимальных и максимальных точек?
4. Всегда ли данный алгоритм способен отыскать точку оптимума?
5. Если у алгоритма склонность к заикливанию?
6. Как Вы оцениваете ситуацию отсутствия градиента в некоторой текущей точке решения?
7. Что такое собственные векторы матрицы?
8. Для чего в задачах многомерной оптимизации применяются методы одномерной минимизации?
9. Может ли шаг λ оказаться равным нулю в ходе расчётов по мере спуска?
10. Что такое норма вектора?
11. Как рассчитать норму вектора?

ЛАБОРАТОРНАЯ РАБОТА № 5 ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ КВАЗИНЬЮТОНОВСКИМИ МЕТОДАМИ

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задач нелинейного программирования квазиньютоновскими методами.

2. Закрепить способы применения функций `fminunc` и `fminbnd` пакета `Optimization toolbox` среды `MatLab`.

3. Освоить алгоритмы квазиньютоновских методов решения задач нелинейного программирования.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Квазиньютоновские методы или методы переменной метрики производят построение аппроксимирующей матрицы, являющейся обратной по отношению к матрице Гессе, в процессе спуска к оптимальному значению [2, 7, 8, 10].

Если алгоритм движения точки подчиняется закону

$$x_{n+1} = x_n - \lambda_n D_n \nabla f(x_n),$$

где D_n – аппроксимирующая матрица, λ_n – стабилизирующий множитель, а аппроксимирующую матрицу D_n строят по правилу:

$$H^{-1}(x_n) \approx D_{n+1} = \omega(D_n + \Delta D_n),$$

ω – нормирующий множитель, то приращение ΔD_n , обеспечивающее корректное построение аппроксимации матрицы Гессе $[H(X_k)]^{-1}$, выглядит следующим образом

$$\Delta D_n = \frac{1}{\omega} \cdot \frac{\Delta x_n}{\Delta g_n^T} \cdot \frac{Y^T}{Y} - \frac{D_n \Delta g_n}{\Delta g_n^T} \cdot \frac{Z^T}{Z},$$

где $\Delta g_n = \nabla f(x_{n+1}) - \nabla f(x_n)$, $\Delta x_n = x_{n+1} - x_n$.

Фиктивные добавки – векторы Y и Z , размерности n , решают задачи:

- определмости матричных операций:
- сходимости D_n к $H^{-1}(x_n)$,
- её положительной определённости.

Определение фиктивных добавок – самостоятельная задача, решением которой занимались многие исследователи.

В алгоритме, предложенном Бройденом (Broyden C.G.), положено: $\omega = 1$, $Y = Z = \Delta x_n - D_n \Delta g_n$.

В алгоритме Давидона-Флетчера-Пауэлла (Davidon W.C., Fletcher R., Powell M.J.D) аналогичные добавки определяются как: $\omega = 1$, $Y = \Delta x_n$, $Z = D_n \Delta g_n$.

В обоих случаях, в качестве начального приближения используется единичная матрица I .

Для последнего алгоритма

$$D_{n+1} = D_n + \frac{\Delta x_n}{\Delta g_n^T} \cdot \frac{\Delta x_n^T}{\Delta x} - \frac{D_n \Delta g_n}{\Delta g_n^T} \cdot \frac{(D_n \Delta g_n)^T}{D_n \Delta g_n} = D_n + A_n - B_n$$

или

$$D_{n+1} = I + \sum_{i=1}^n A_i - \sum_{i=1}^n B_i,$$

в котором A_n обеспечивает сходимость D_n к $H^1(x_n)$, а B_n – положительную определённую.

Схема алгоритма является типовой для градиентных методов.

Параметр $\lambda_n^* \Rightarrow \min_{\lambda > 0} f(X_n + \lambda s_n)$, где $s_n = -\nabla f(X_n)$, вычисляется по методу, предложенному Коши для алгоритма наискорейшего спуска.

3. ХОД РАБОТЫ

1. Реализовать две версии алгоритма в части нахождения λ^* :
 - с применением аналитического нахождения шага и
 - с применением численного нахождения шага используя алгоритм одномерной минимизации fminbnd.
2. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.
3. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать **содержательный** вывод, оценивающий полученные результаты.
4. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чём состоят достоинства и недостатки метода Ньютона?
2. В чём предназначение квазиньютоновских методов?
3. Почему методы получили такое название?
4. Для чего необходимо обеспечивать сходимость аппроксимирующей матрицы?
5. Для чего необходимо обеспечивать положительную определённую аппроксимирующей матрицы?
6. Какова цель ввода векторных компонент Y и Z в формулу для расчёта ΔD_{n+1} ?

7. Будет ли алгоритм функционировать, если λ положить константному значению, сэкономив на времени его вычисления?
8. Как по Вашему, насколько критичен алгоритм к выбору начального значения?
9. Почему в реализациях алгоритма нормирующий множитель $\omega = 1$?
10. Что будет, если нормирующий множитель будет не единичен?

ЛАБОРАТОРНАЯ РАБОТА № 6 ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ МЕТОДОМ ЗОЙТЕНДЕЙКА

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задач нелинейного программирования квазиньютоновскими методами.
2. Освоить и закрепить способы применения функции `nonlcon`, `linprog`, `fmincon` и `fminbnd` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить алгоритмы Зойтендейка методов решения задач нелинейного программирования.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Метод Зойтендейка (G. Zoutendijk) [3, 9, 10] базируется на понятии независимого направления.

Вектор с ненулевыми компонентами S называется *возможным направлением* спуска в точке $X \in \mathcal{R}$, если существует $\delta > 0$, такое, что $X + \lambda S \in \mathcal{R}$ для всех $\lambda \in (0, \delta)$ и $f(X + \lambda S) < f(X)$.

По условию задачи с ограничением строится эквивалентная задача линейного программирования.

Существуют три разновидности указанного метода: при линейных ограничениях; при нелинейных ограничениях и улучшенной сходимости при нелинейных ограничениях.

Для квадратичного программирования характерно использование линейных ограничений. Постановка задачи, в этом случае, такова

$$\begin{aligned} & \min f(X) \\ & \begin{cases} AX \leq b, \\ HX = h, \end{cases} \end{aligned}$$

где $f(X)$ – квадратична, $A[m \times n]$, $H[l \times n]$ – матрицы, а $b[m]$ и $h[l]$ – вектора системы линейных ограничений.

Пусть в текущей точке все ограничения со знаком “ $\leq$ ” представимы в виде

$$\begin{cases} A_1 X = b_1, \\ A_2 X < b_2. \end{cases}$$

Матрицы исходной системы подлежат разделению или перестановке по знакам отношений

$$\begin{cases} A^T = \begin{bmatrix} A_1^T & A_2^T \end{bmatrix}, \\ b^T = \begin{bmatrix} b_1^T & b_2^T \end{bmatrix}. \end{cases}$$

Вектор S будет являться направлением спуска в точке X при выполнении условий:

$$\begin{cases} A_1 S \leq 0, \\ HS = 0. \end{cases}$$

На компоненты вектора S налагаются дополнительные ограничения (требования) нормировки:

- на величину элементов

$$-1 \leq s_i \leq 1, j = 1, \bar{n};$$

- на модуль вектора возможного направления S

$$S^T S \leq 1, \|S\|^2 \leq 1,$$

это ограничения объединяет условия $-1 \leq s_j \leq 1, j = 1, \bar{n}$ и

$$\sum_{j=1}^n s_j \leq 1, \sum_{j=1}^n s_j \geq -1;$$

- на величину целевой функции ЗЛП

$$\nabla^T f(X) S \geq -1.$$

Поэтому возможна одна из трёх постановок задачи минимизации, которая может быть решена соответствующим методом линейного программирования:

$$\min \nabla^T f(X)S$$

$$\begin{cases} A_1 S \leq 0, \\ HS = 0, \\ s_j \leq 1, \\ s_j \geq -1, \quad j = 1, m. \end{cases}$$

1)

$$\min \nabla^T f(X)S$$

$$\begin{cases} A_1 S \leq 0, \\ HS = 0, \\ S^T S \leq 1. \end{cases}$$

2)

$$\min \nabla^T f(X)S$$

$$\begin{cases} A_1 S \leq 0, \\ HS = 0, \\ \nabla^T f(X)S \geq -1. \end{cases}$$

3)

Предварительный этап.

Нахождение точки X , удовлетворяющей системе ограничений задачи.

Итерация.

1. Найти множество активных ограничений в текущей точке и композицию матрицы системы ограничений $A^T = [A_1^T \quad A_2^T]$ и решить ЗЛП вида 1, 2 или 3, по желанию.

2. Проверка условия окончания.

Если оптимальное значение целевой функции ЗЛП равно нулю, **то** текущие координаты X определяют точку Куна-Таккера.

3. **Иначе** решить задачу одномерной минимизации Коши: $\min_{0 \leq \lambda \leq \lambda_{\max}} f(X + \lambda S)$, в которой граничное значение параметра λ определяется как

$$\lambda_{\max} = \begin{cases} \min_k \left\{ \frac{\tilde{b}_k}{\tilde{s}_k} \mid \tilde{s}_k > 0 \right\}, & \exists_k \tilde{s}_k > 0, \\ \infty, & \forall_k \tilde{s}_k \leq 0. \end{cases} \quad \text{где } \tilde{b} = b_2 - A_2 X, \quad \tilde{S} = A_2 S.$$

4. Перейти в новую текущую точку $X = X + \lambda \times S$.

Алгоритм не сложен, он представлен на рисунке 1.

3. ХОД РАБОТЫ

1. Реализовать две версии алгоритма в части нахождения λ^* :

- с применением аналитического нахождения шага и
- с применением численного нахождения шага используя алгоритм одномерной минимизации `fminbnd`.

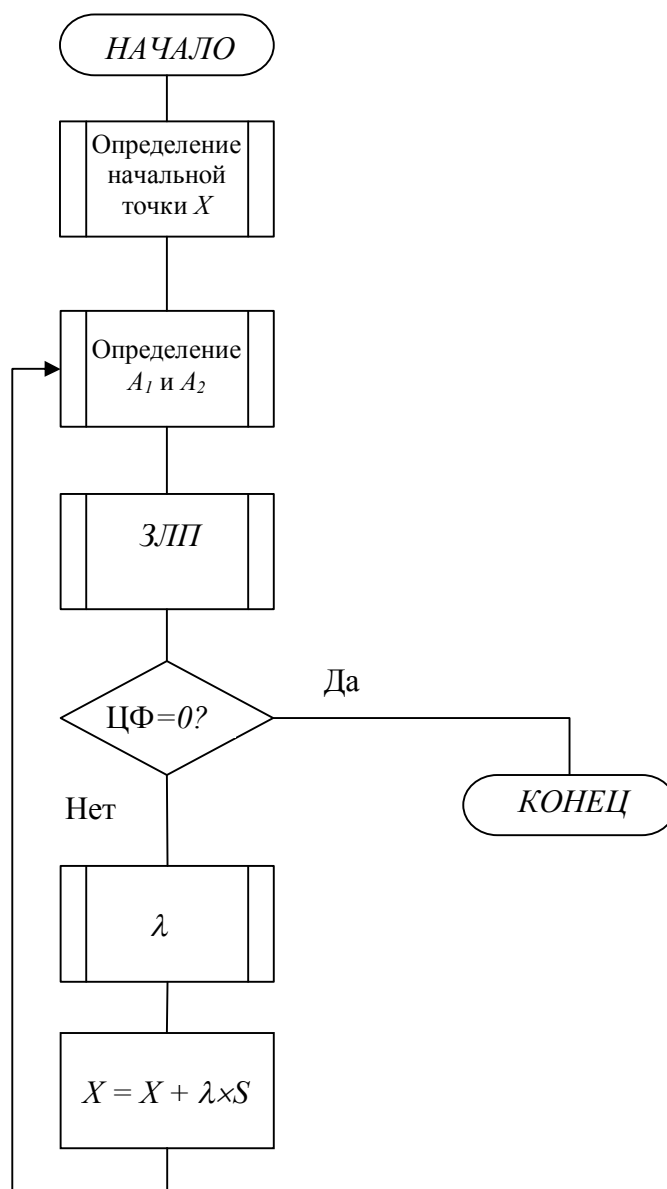


Рисунок 1 – Граф-схема алгоритма Зойтендейка

2. Для решения задачи линейного программирования воспользоваться процедурой `linprog`.

3. Выполнить расчёт тестового примера с использованием `fmincon`.

4. Для разделения ограничений на ограничения равенства и строгие неравенства применить `nonlcon`.

5. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.

6. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать **содержательный** вывод, оценивающий полученные результаты.

7. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какое направление называется направлением возможного спуска?
2. Что такое активное ограничение?
3. О чём говорит неразрешимость построенной эквивалентной задачи линейного программирования?
4. Зачем вычислять параметр λ ?
5. Что реально обозначают условия, используемые в эквивалентной задаче линейного программирования?
6. Что понимается под термином точка Куна-Таккера?
7. Как Вы думаете, возможна ли неразрешимость задачи квадратичного программирования?
8. Что служит причиной неразрешимости задачи квадратичного программирования?
9. Возможна ли безрезультативная работа алгоритма метода?
10. Поменяется ли алгоритм, если из постановки задачи убрать ограничения вида «равенство»?

ЛАБОРАТОРНАЯ РАБОТА № 7 ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ МЕТОДОМ РОЗЕНА

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задач нелинейного программирования квазиньютоновскими методами.
2. Освоить и закрепить способы применения функции `nonlcon`, `linprog`, `fmincon` и `fminbnd` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить алгоритм Розена решения задач нелинейного программирования.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Метод Розена предназначен для решения задач нелинейного программирования с линейной системой ограничений [2, 8, 9]

$$\begin{aligned} \min f(X) \\ \begin{cases} AX \leq b, \\ HX = h, \end{cases} \end{aligned}$$

где $f(X)$ – нелинейная и **дифференцируемая** функция, A – матрица размером $[m \times n]$, H – $[l \times n]$ -мерная матрица, а b – m -мерный вектор, а h – l -мерный вектор системы линейных ограничений.

Розен предложил направление спуска строить по правилу

$$S = -P \times \nabla f(X),$$

где P – матрица проецирования или проецирующая (проектирующая) матрица, которая бы гарантировала сохранение допустимости текущей точки.

Пусть в допустимой точке часть неравенств активна, а другая часть – нет, то допускается представление системы ограничений по признаку активности в виде блочной матрицы:

$$\begin{aligned} A_1 X = b_1, \quad A^T = \begin{bmatrix} A_1^T & A_2^T \end{bmatrix} \\ A_2 X < b_2, \quad b^T = \begin{bmatrix} b_1^T & b_2^T \end{bmatrix} \end{aligned} \quad (1)$$

Матрица проецирования P должна удовлетворять условию $P \times \nabla f(X) \neq 0$ и блокировать возможные направления в сторону активных ограничений. Это будет соблюдаться, когда проектирующая матрица определена как

$$P = I - M^T \times (M \times M^T)^{-1} \times M, \quad (2)$$

где I – единичная матрица, $M^T = [A_1^T H^T]$ – невырождена. Отметим, что выполняется тождество $M \times P = 0$, то есть $A_1 \times P = 0$ и $H \times P = 0$, таким образом, спуск в направлении ограничений, выполняемых как равенство, не производится.

Рассмотрим точку, в которой выполняется условие $P \times \nabla f(X) = 0$. Имеем

$$[I - M^T \times (M \times M^T)^{-1} \times M] \times \nabla f(X) = \nabla f(X) + M^T \times W = 0,$$

где $W^T = [U^T V^T]$, а

$$\nabla f(X) + A_1^T \times U + H^T \times V = 0.$$

Если компоненты вектора U неотрицательны, то X является точкой Куна-Таккера. В противном случае, можно построить новое направление спуска

$$S = -\tilde{P} \times \nabla f(X),$$

где $\tilde{P} = I - \tilde{M}^T \times (\tilde{M} \times \tilde{M}^T)^{-1} \times \tilde{M}$, в котором $\tilde{M}^T = [\tilde{A}_1^T H^T]$, а $\tilde{A}_1$ получено из A_1 путём вычёркивания строк, соответствующих компонентам $u_j < 0$.

Схема алгоритма показана на рисунке 1.

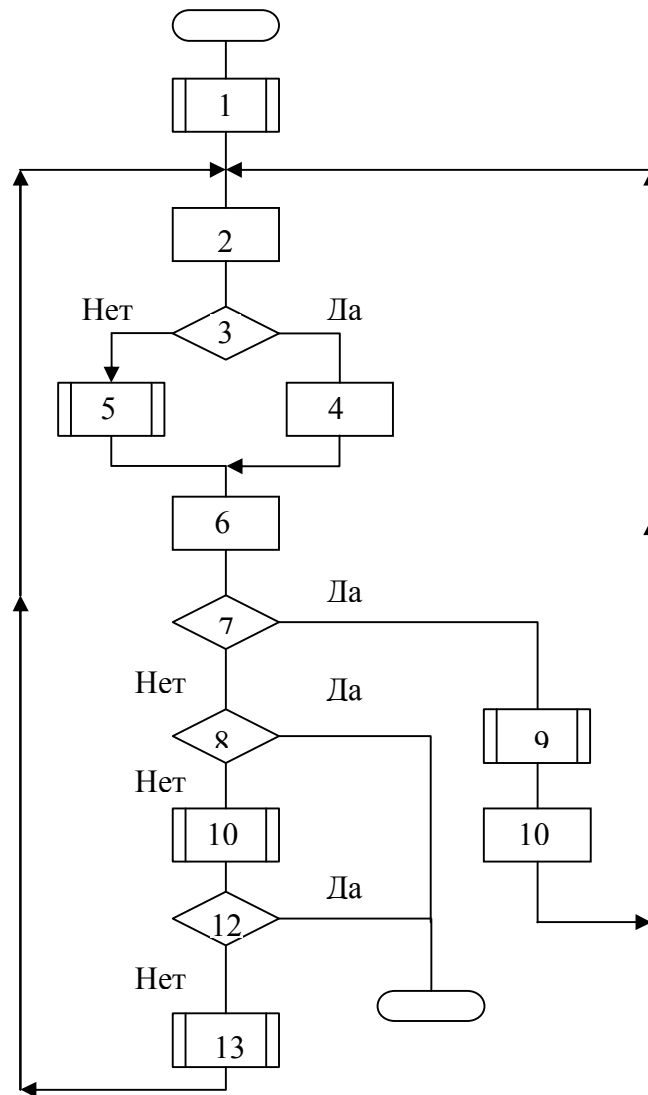


Рисунок 1 – Граф-схема алгоритма Розена

Блок № 1 выполняет выбор начальной точки X , удовлетворяющей системе ограничений задачи, отыскивает активные ограничения и разложение ограничений с неравенствами (1).

В блоке № 2 конструируется матрица $M^T = [A_1^T H^T]$, которая проверяется на равенство нулю всех её элементов условием блока № 3: $M = 0$?, если матрица вся нулевая (выход «да» блока № 3), то проецирующая матрица полагается равной единичной (блок № 4).

В противном случае (выход «нет» блока № 3), выполняется расчёт проецирующей матрицы в блоке № 5 по формуле (2).

После определения элементов проецирующей матрицы, рассчитывается возможное направление спуска $S = -P \times \nabla f(X)$ в блоке № 6.

Блок № 7 выполняет проверку: $S \neq 0?$, то есть, не является ли текущая точка точкой Куна-Таккера.

Если все компоненты вектора возможного направления нулевые (выход «нет» блока № 7), то это, возможно, искомое решение. Чтобы убедиться в этом, проверяется равенство нулю всех компонентов блочной матрицы M (блок № 8, $M = 0?$), и если это так (выход «да» блока № 8), то решение успешно заканчивается.

Если среди компонентов матрицы M есть ненулевые элементы (выход «нет» блока № 8), то ищется разложение $W^T = [U^T V^T]$, блок № 10.

Если компоненты подматрицы U неотрицательны, что проверяется блоком № 12 ($\forall_i u_i \geq 0?$), то X является точкой Куна-Таккера (выход «да»).

Если компоненты U отрицательны (выход «нет» блока № 12), то переопределяется матрицы $A_1 = \tilde{A}_1$ и блочная матрица $\tilde{M}^T = [\tilde{A}_1^T H^T]$ (блок № 13).

Если не все элементы вектора возможного направления нулевые (выход «да» блока № 7), рассчитывается новая текущая точка. Сперва выбирается параметр λ решением задачи одномерной минимизации Коши (блок № 9): $\min_{0 \leq \lambda \leq \lambda_{\max}} f(X + \lambda S)$, в которой граничное значение параметра λ определяется, как и в методе Зойтендейка:

$$\lambda_{\max} = \begin{cases} \min_k \left\{ \frac{\tilde{b}_k}{\tilde{s}_k} \mid \tilde{s}_k > 0 \right\}, & \exists_k \tilde{s}_k > 0, \\ \infty, & \forall_k \tilde{s}_k \leq 0. \end{cases} \quad \text{где } \tilde{b} = b_2 - A_2 X, \quad \tilde{S} = A_2 S.$$

Значение очередной текущей точки рассчитывается в блоке № 11 по рекуррентной формуле $X = X + \lambda \times S$.

3. ХОД РАБОТЫ

1. Реализовать две версии алгоритма в части нахождения λ^* :
 - с применением аналитического нахождения шага и
 - с применением численного нахождения шага используя алгоритм одномерной минимизации `fminbnd`.
2. Для решения задачи линейного программирования воспользоваться процедурой `linprog`.
3. Выполнить расчёт тестового примера с использованием `fmincon`.

4. Для разделения ограничений на ограничения равенства и строгие неравенства применить `nonlcon`.

5. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.

6. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать *содержательный* вывод, оценивающий полученные результаты.

7. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каково назначение проектирующей матрицы?
2. Из каких соображений строится проектирующая матрица?
3. О чём свидетельствует нулевое значение компонентов вектора направления после проецирования?
4. Будет ли получен результат, если начальное приближение недопустимая точка?
5. Можно ли применять сей алгоритм, если все ограничения системы представляются в виде равенств?
6. Каков смысл и геометрическая интерпретация ситуации $P \cdot \nabla f(X_k) = 0$?
7. Каков смысл и геометрическая интерпретация ситуации $P \cdot \nabla f(X_k) = \nabla f(X_k)$?
8. Каков смысл и геометрическая интерпретация ситуации $P \cdot \nabla f(X_k) \neq 0$?
9. Что будет, если в некой точке решения все неравенства окажутся активными?
10. Что будет, если в некой точке решения не будет ни одного активного неравенства?

ЛАБОРАТОРНАЯ РАБОТА № 8

ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ МЕТОДОМ ШТРАФНЫХ ФУНКЦИЙ ВНУТРЕННЕЙ ТОЧКИ

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задач методами штрафных функций.
2. Закрепить способы применения функций `fminunc` и `fseminf` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить алгоритмы методов внутренней точки решения задач нелинейного программирования.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Методы штрафных функций основаны на практике перехода от задачи *условной минимизации* к задаче *безусловной минимизации* путём построения специальной штрафной функции [9, 10].

$$P(X, \rho) = f(X) + \sum_{i=1}^m \rho_i H[h_i(X)] + \sum_{i=m+1}^l \rho_i G[g_i(X)],$$

где $P(X, \rho)$ – штрафная функция; $\rho_i, i = 1, \bar{l}$ – весовые коэффициенты, отражающие значимость соблюдения того или иного ограничения; $H[h_i(X)]$ и $G[g_i(X)]$ – некоторые функционалы.

Функционал $H[h_i(X)]$ должен сделать невыгодным любое отклонение аргумента X от поверхности $h_i(X) = 0$, то есть

$$\lim_{h_i(X) \rightarrow 0} H[h_i(X)] = 0, \quad i = 1, \bar{m}.$$

Поэтому в качестве функционала выбирают чётную степенную функцию

$$H[y] = y^p, \quad p = 2, 4, \dots \text{ либо } H[y] = |y|^p, \quad p = 1, 2, \dots$$

Для метода внутренней точки функционал таков:

$$\lim_{g_i(X) \rightarrow 0^+} G[g_i(X)] = \infty, \quad i = m+1, \bar{l} \text{ для } g_i(X) > 0.$$

Алгоритм (рисунок 1) относится к группе методов внутренней точки, используется для решения задач вида

$$\begin{aligned} & \min f(X); \\ & \begin{cases} g_i(X) \geq 0, & i = 1, \bar{m}; \\ x_j \geq 0, & j = 1, \bar{n}. \end{cases} \end{aligned}$$

По условиям задачи строится штрафная функция вида

$$P(X, r, \Omega) = f(X) + r \cdot \sum_{i=1}^m \omega_i G[g_i(X)]. \quad (3)$$

В формуле (3) использованы: параметр $r > 0$, убывающий по мере вычислений и Ω – вектор положительных весовых коэффициентов, учитывающих важность (значимость) ограничений модели.

Функции $G[g_i(X)]$ выбирается из альтернатив

$$G[Y] = 1/Y \text{ или } G[Y] = -\ln[Y].$$

Легко видеть, что $\lim_{Y \rightarrow 0^+} G[Y] = \infty$, то есть функция барьера, при приближении к нему изнутри области, неограниченно возрастает. Штрафная добавка к функции цели определяется формулой

$$\Delta = r \cdot \sum_{i=1}^m \omega_i G[g_i(X)]$$

Функционирование алгоритма представлено на рисунке 1, в его работе использована уменьшающая константа $0 < \beta < 1$, точность расчётов задаётся константой ε .

Замечания:

- область ограничений должна быть непустой: $X_{\text{opt}} \in \{X: g_i(X) > 0, i = 1, m\}$;
- начальное значение X должно быть допустимой точкой;
- метод критичен по отношению к параметру r : если значение r мало, то будет отыскан оптимум функции, не совпадающий с истинным, а если значение r велико, то возрастёт число итераций;
- аналогично, близкие к единице значения β могут привести к преждевременному прекращению вычислений.

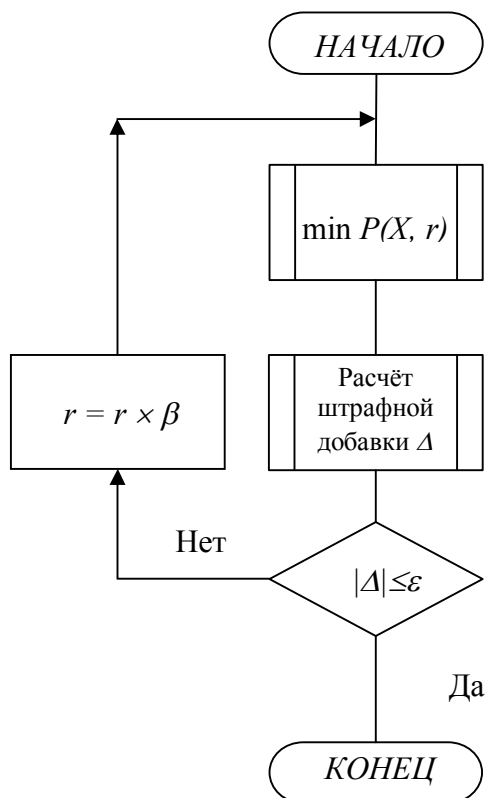


Рисунок 6 – Схема алгоритма МБП

3. ХОД РАБОТЫ

1. По условию варианта составить штрафную функцию.
2. Реализовать алгоритм расчёта, для проведения минимизации на каждом шаге воспользоваться процедурой `fminunc` или `fseminf`.
3. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.
6. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать **содержательный** вывод, оценивающий полученные результаты.
7. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Почему метод называется методом “внутренней точки”?
2. Что вкладывается в понятие функции барьера?

3. Для чего введены множители ω_i в записи штрафной функции?
4. Каков назначение множителя r в записи штрафной функции метода внутренней точки?
5. Почему множитель β следует выбирать меньше единицы?
6. Каким требованиям должен удовлетворять функционал G в методах внутренней точки?
7. Насколько критичен алгоритм метода к выбору начальной точки?
8. Всегда ли задача разрешима данным методом?
9. Что произойдёт, если в области ограничений будет сразу несколько оптимальных решений?
10. Повлияют ли значения множителей ω_i на найденное оптимальное решение?

ЛАБОРАТОРНАЯ РАБОТА № 9. ИССЛЕДОВАНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ МЕТОДОМ ШТРАФНЫХ ФУНКЦИЙ ВНЕШНЕЙ ТОЧКИ

1. ЦЕЛЬ РАБОТЫ

1. Получить практические навыки в решении задач методами штрафных функций.
2. Закрепить способы применения функций `fminunc` и `fseminf` пакета `Optimization toolbox` среды `MatLab`.
3. Освоить алгоритмы методов внешней точки решения задач нелинейного программирования.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Метод внешней точки ориентируется на решение НП-задач в общей постановке [2, 9, 10]

$$\begin{aligned} \min f(X); \\ \begin{cases} h_i(X) = 0, & i = 1, \overline{m}; \\ g_i(X) \geq 0, & i = m+1, \overline{l}. \end{cases} \end{aligned}$$

Штрафная функция описывается выражением

$$P(X, r) = f(X) + rL(X),$$

где r – коэффициент штрафа, а величина штрафа в текущей точке X есть

$$L(X) = \sum_{i=1}^m H[h_i(X)] + \sum_{i=m+1}^l G[g_i(X)].$$

Используются функционалы:

$$G[Y] = [\max \{0; -Y\}]^P, P > 0, \text{ целое,}$$

$$H[Y] = |Y|^P, P > 0, \text{ целое.}$$

Легко видеть, что функционал $G[Y]$ обращается в нуль при попадании текущей точки в область ограничений.

Алгоритм метода полностью дублирует алгоритм метода барьерных поверхностей, рассмотренный в предыдущей лабораторной работе, но, в отличие от него, используется параметр увеличения штрафа β больше единицы.

3. ХОД РАБОТЫ

1. По условию варианта составить штрафную функцию.
2. Реализовать алгоритм расчёта, для проведения минимизации на каждом шаге воспользоваться процедурой `fminunc` или `fseminf`.
3. Сравнить между собой по точности и по числу итераций результаты работы версии алгоритмов и результатов выполнения предыдущих лабораторных работ.
6. Оформить отчет с приложением результатов вычислений, необходимой графической информации. Сделать **содержательный** вывод, оценивающий полученные результаты.
7. Защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Почему метод называется методом “внешней точки”?
2. За счёт чего будет происходить минимизация при возрастании r ?
3. Каким требованиям должен удовлетворять функционал G в методах внешней точки?

4. Каким требованиям должен удовлетворять функционал H в методах штрафных функций?
5. Почему множитель β следует выбирать много больше единицы?
6. Каков назначение множителя r в записи штрафной функции метода внешней точки?
7. Обязательно ли начальное приближение является недопустимой точкой?
8. Что произойдёт, если область ограничений пуста?
9. Всегда ли достижимо оптимальное решение при использовании указанного метода?
10. Может ли целевая функция в исходной постановке задачи определяться кусочно-непрерывно?

ЗАКЛЮЧЕНИЕ

“Аппарат нелинейного программирования оказывается весьма полезным в автоматическом регулировании, космических исследованиях, строительной механике и многих других областях человеческой деятельности” [12]

“Специалист, желающий решить свою практическую задачу с помощью математических методов оптимизации, должен располагать некоторым запасом реализуемых моделей, среди которых может выбирать ... адекватную модель” [9].

“... а математику уже затем учить надо, что она ум в порядок приводит” [М.В. Ломоносов].

Заручившись поддержкой сонма столь выдающихся авторитетов, повествование сие завершаем, в пику персонажу М.Е. Салтыкова-Щедрина, архистратигу Перехват-Залихватскому, коей въехал в г. Глупов на белом коне, сжёг гимназию и упразднил науки. На Святой Руси имя подобным деятелям – легион. А “...дорога из города Глупова в город Умнов лежит через город Буянов, а не через манную кашу...”. Михаил Евграфович знал, о чём писал.

*Е.Н. Заикина
В.Ю. Карлусов*

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Банди Б. Методы оптимизации: Вводный курс [Текст]: / Б. Банди. - М.: Радио и Связь, 1988. – 128 с.
2. Бейко И.В. Методы и алгоритмы решения задач оптимизации. [Текст]: / И.В. Бейко, Б.Н. Бублик, П.Н. Зинько. - Киев: Вища школа, 1983. - 512 с.
3. Введение в нелинейное программирование [Текст]: / Под. ред. К.-Х. Эльстера. – М.: Наука, 1985. – 263 с.
4. Гантмахер Ф.Р. Теория матриц / Ф.Р. Гантмахер, М.: Наука, ГРФМЛ, 1988. – 552 с.
5. Горлач, Б.А. Исследование операций [Электронный ресурс] : учебное пособие / Б.А. Горлач. — Электрон. дан. — Санкт-Петербург : Лань, 2013. — 448 с. — Режим доступа: <https://e.lanbook.com/book/4865>. — Загл. с экрана.
6. Есипов, Б.А. Методы исследования операций [Электронный ресурс] : учебное пособие / Б.А. Есипов. — Электрон. дан. — Санкт-Петербург : Лань, 2013. — 304 с. — Режим доступа: <https://e.lanbook.com/book/68467>. — Загл. с экрана.
7. Зайченко Ю.П. Исследование операций. [Текст]: учебное пособие. / Ю.П. Зайченко. -Киев: Вища школа, 1979. -392 с.
8. Кюнц Г.П. Нелинейное программирование [Текст]: / Г.П. Кюнц, В. Крелле. - М.: Советское радио, 1965. - 303 с.
9. Фиакко А.В. Нелинейное программирование: Методы последовательной безусловной оптимизации [Текст]:/ А.В. Фиакко, Г.П. Мак-Кормик –М.: Мир, 1972. – 240 с.
10. Химмельблау Д. Прикладное нелинейное программирование [Текст]: / Д. Химмельблау. - М.: Мир, 1975. – 534 с.

Электронные издания

11. Ашманов, С.А. Теория оптимизации в задачах и упражнениях [Электронный ресурс] : учебное пособие / С.А. Ашманов, А.В. Тимохов. — Электрон. дан. — Санкт-Петербург : Лань, 2012. — 448 с. — Режим доступа: <https://e.lanbook.com/book/3799>. — Загл. с экрана.
12. Болдырев, Ю. Я. Вариационное исчисление и методы оптимизации : учеб. пособие для вузов / Ю. Я. Болдырев. — Москва : Издательство Юрайт, 2019. — 240 с. — (Серия : Университеты России). — ISBN 978-5-534-01707-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/book/variacionnoe-ischislenie-i-metody-optimizacii-438267> (дата обращения: 06.04.2019).
13. Гончаров, В. А. Методы оптимизации : учеб. пособие для бакалавриата и магистратуры / В. А. Гончаров. — Москва : Издательство Юрайт, 2019. — 191 с. — (Серия : Бакалавр и магистр. Академический курс).

- ISBN 978-5-9916-3642-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/book/metody-optimizacii-425157> (дата обращения: 06.04.2019).
14. Кудрявцев, К. Я. Методы оптимизации : учеб. пособие для вузов / К. Я. Кудрявцев, А. М. Прудников. — 2-е изд. — Москва : Издательство Юрайт, 2019. — 140 с. — (Серия : Университеты России). — ISBN 978-5-534-08523-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/book/metody-optimizacii-442329> (дата обращения: 06.04.2019).
 15. Сухарев, А. Г. Методы оптимизации : учебник и практикум для бакалавриата и магистратуры / А. Г. Сухарев, А. В. Тимохов, В. В. Федоров. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 367 с. — (Серия : Бакалавр и магистр. Академический курс). — ISBN 978-5-9916-3859-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/book/metody-optimizacii-444155> (дата обращения: 06.04.2019).
 16. Токарев, В. В. Методы оптимизации : учеб. пособие для бакалавриата и магистратуры / В. В. Токарев. — Москва : Издательство Юрайт, 2019. — 440 с. — (Серия : Бакалавр и магистр. Академический курс). — ISBN 978-5-534-04712-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/book/metody-optimizacii-438843> (дата обращения: 06.04.2019).
 17. Методы оптимизации: теория и алгоритмы : учеб. пособие для академического бакалавриата / А. А. Черняк, Ж. А. Черняк, Ю. М. Метельский, С. А. Богданович. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 357 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-04103-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/book/metody-optimizacii-teoriya-i-algoritmy-438378> (дата обращения: 06.04.2019).
 18. Колбин, В.В. Специальные методы оптимизации [Электронный ресурс] / В.В. Колбин. — Электрон. дан. — Санкт-Петербург : Лань, 2014. — 384 с. — Режим доступа: <https://e.lanbook.com/book/41015>. — Загл. с экрана.

ПРИЛОЖЕНИЕ А
(обязательное)

ВАРИАНТЫ СИСТЕМ ОГРАНИЧЕНИЙ ДЛЯ ИЗУЧЕНИЯ АЛГОРИТМОВ
РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

$$00. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 6 \cdot x_1 + 3 \cdot x_2 \leq 18; \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 2 \cdot x_1 + 2 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$02. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 5 \cdot x_1 + 3 \cdot x_2 \geq 15; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$04. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 6 \cdot x_1 + 3 \cdot x_2 \leq 18; \\ 7 \cdot x_1 + 2 \cdot x_2 \geq 14; \\ 3 \cdot x_1 + 3 \cdot x_2 \geq 9; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$06. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 5 \cdot x_1 + 2 \cdot x_2 \leq 10; \\ 2 \cdot x_1 + 4 \cdot x_2 \leq 8; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$08. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 5 \cdot x_1 + 6 \cdot x_2 \leq 30; \\ 4 \cdot x_1 + 2 \cdot x_2 \leq 8; \\ 5 \cdot x_1 + 3 \cdot x_2 \geq 15; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$10. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$01. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 7 \cdot x_1 + 5 \cdot x_2 \leq 35; \\ 3 \cdot x_1 + 6 \cdot x_2 \leq 18; \\ 7 \cdot x_1 + 2 \cdot x_2 \geq 14; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$03. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 7 \cdot x_1 + 6 \cdot x_2 \leq 42; \\ 4 \cdot x_1 + 6 \cdot x_2 \leq 24; \\ 5 \cdot x_1 + 3 \cdot x_2 \geq 15; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$05. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 1 \cdot x_1 + 6 \cdot x_2 \leq 6; \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 2 \cdot x_1 + 2 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$07. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 5 \cdot x_1 + 7 \cdot x_2 \leq 35; \\ 7 \cdot x_1 + 2 \cdot x_2 \leq 14; \\ 3 \cdot x_1 + 4 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$09. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 6 \cdot x_1 + 5 \cdot x_2 \leq 30; \\ 2 \cdot x_1 + 6 \cdot x_2 \leq 12; \\ 5 \cdot x_1 + 2 \cdot x_2 \geq 10; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$11. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$12. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$14. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 2 \cdot x_2 \leq 6; \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$16. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 2 \cdot x_2 \leq 10; \\ 2 \cdot x_1 + 6 \cdot x_2 \leq 12; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 1; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$18. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 3 \cdot x_1 + 3 \cdot x_2 \leq 9; \\ 7 \cdot x_1 + 1 \cdot x_2 \leq 7; \\ 1 \cdot x_1 + 7 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$20. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$22. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 3 \cdot x_2 \leq 18; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$24. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$26. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$13. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 3 \cdot x_2 \leq 12; \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 5 \cdot x_1 + 1 \cdot x_2 \geq 5; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$15. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 7 \cdot x_2 \leq 7; \\ 6 \cdot x_1 + 3 \cdot x_2 \leq 18; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 1; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$17. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 7 \cdot x_2 \leq 28; \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 3 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$19. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 2 \cdot x_1 + 1 \cdot x_2 \leq 6; \\ 1 \cdot x_1 + 3 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$21. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$23. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 3 \cdot x_1 + 6 \cdot x_2 \leq 18; \\ 4 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$25. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 3 \cdot x_1 + 2 \cdot x_2 \leq 6; \\ 3 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$27. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 5 \cdot x_1 + 4 \cdot x_2 \leq 20; \\ 4 \cdot x_1 + 2 \cdot x_2 \geq 8; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$28. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 6 \cdot x_2 \leq 42; \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 5 \cdot x_1 + 2 \cdot x_2 \geq 10; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$30. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 7 \cdot x_2 \leq 7; \\ 6 \cdot x_1 + 2 \cdot x_2 \leq 12; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$32. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 3 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$34. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 1 \cdot x_2 \leq 5; \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 1 \cdot x_1 + 2 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$36. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 3 \cdot x_1 + 6 \cdot x_2 \leq 18; \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 4 \cdot x_1 + 2 \cdot x_2 \geq 8; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$38. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 7 \cdot x_2 \leq 7; \\ 3 \cdot x_1 + 4 \cdot x_2 \leq 12; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$40. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 4 \cdot x_1 + 4 \cdot x_2 \leq 16; \\ 5 \cdot x_1 + 1 \cdot x_2 \geq 5; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$42. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 5 \cdot x_1 + 1 \cdot x_2 \geq 5; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$29. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 4 \cdot x_2 \leq 16; \\ 7 \cdot x_1 + 1 \cdot x_2 \leq 7; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$31. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 2 \cdot x_2 \leq 12; \\ 3 \cdot x_1 + 4 \cdot x_2 \leq 12; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$33. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 12 \cdot x_1 + 14 \cdot x_2 \leq 168; \\ 4 \cdot x_1 + 5 \cdot x_2 \geq 20; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$35. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 6 \cdot x_2 \leq 6; \\ 3 \cdot x_1 + 3 \cdot x_2 \leq 9; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$37. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 4 \cdot x_1 + 6 \cdot x_2 \leq 24; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$39. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 4 \cdot x_2 \leq 20; \\ 4 \cdot x_1 + 6 \cdot x_2 \leq 24; \\ 6 \cdot x_1 + 2 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$41. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 7 \cdot x_2 \leq 35; \\ 8 \cdot x_1 + 3 \cdot x_2 \leq 24; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$43. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 2 \cdot x_2 \leq 6; \\ 5 \cdot x_1 + 2 \cdot x_2 \leq 10; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$44. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 2 \cdot x_1 + 3 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$46. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 7 \cdot x_1 + 2 \cdot x_2 \geq 14; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$48. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 3 \cdot x_1 + 6 \cdot x_2 \leq 18; \\ 5 \cdot x_1 + 1 \cdot x_2 \geq 5; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$50. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 2 \cdot x_2 \leq 12; \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$52. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 4 \cdot x_2 \leq 16; \\ 2 \cdot x_1 + 5 \cdot x_2 \leq 10; \\ 5 \cdot x_1 + 2 \cdot x_2 \geq 10; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$54. \quad \begin{cases} f(x_1, x_2) = \rightarrow \min \\ 5 \cdot x_1 + 4 \cdot x_2 \leq 20; \\ 4 \cdot x_1 + 6 \cdot x_2 \leq 24; \\ 2 \cdot x_1 + 7 \cdot x_2 \geq 14; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$56. \quad \begin{cases} f(x_1, x_2) = \rightarrow \min \\ 7 \cdot x_1 + 2 \cdot x_2 \leq 14; \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 2 \cdot x_1 + 3 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$58. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$45. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 7 \cdot x_2 \leq 7; \\ 4 \cdot x_1 + 4 \cdot x_2 \leq 16; \\ 6 \cdot x_1 + 2 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$47. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 2 \cdot x_1 + 3 \cdot x_2 \leq 12; \\ 6 \cdot x_1 + 1 \cdot x_2 \leq 6; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$49. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 2 \cdot x_1 + 6 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$51. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 8 \cdot x_1 + 3 \cdot x_2 \leq 24; \\ 7 \cdot x_1 + 7 \cdot x_2 \leq 49; \\ 4 \cdot x_1 + 2 \cdot x_2 \geq 8; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$53. \quad \begin{cases} f(x_1, x_2) = \rightarrow \min \\ 7 \cdot x_1 + 2 \cdot x_2 \leq 14; \\ 2 \cdot x_1 + 6 \cdot x_2 \leq 12; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 1; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$55. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 3 \cdot x_1 + 3 \cdot x_2 \leq 9; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$57. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 7 \cdot x_1 + 2 \cdot x_2 \leq 14; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$59. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 7 \cdot x_2 \leq 42; \\ 7 \cdot x_1 + 5 \cdot x_2 \leq 35; \\ 7 \cdot x_1 + 3 \cdot x_2 \geq 21; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$60. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$62. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 6 \cdot x_2 \leq 36; \\ 5 \cdot x_1 + 7 \cdot x_2 \leq 35; \\ 7 \cdot x_1 + 3 \cdot x_2 \geq 21; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$64. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 2 \cdot x_1 + 3 \cdot x_2 \leq 12; \\ 5 \cdot x_1 + 4 \cdot x_2 \leq 20; \\ 3 \cdot x_1 + 5 \cdot x_2 \geq 15; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$66. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 2 \cdot x_2 \leq 14; \\ 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$68. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 4 \cdot x_2 \leq 20; \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$70. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 6 \cdot x_2 \leq 30; \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$72. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 1 \cdot x_1 + 5 \cdot x_2 \leq 5; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$74. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 6 \cdot x_1 + 6 \cdot x_2 \leq 36; \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 2 \cdot x_1 + 6 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$61. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 3 \cdot x_1 + 3 \cdot x_2 \geq 9; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$63. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 3 \cdot x_1 + 6 \cdot x_2 \leq 18; \\ 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$65. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 6 \cdot x_2 \leq 6; \\ 4 \cdot x_1 + 2 \cdot x_2 \leq 8; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$67. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 4 \cdot x_1 + 3 \cdot x_2 \leq 12; \\ 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 5 \cdot x_1 + 1 \cdot x_2 \geq 5; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$69. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 1 \cdot x_2 \leq 4; \\ 5 \cdot x_1 + 1 \cdot x_2 \leq 5; \\ 1 \cdot x_1 + 6 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$71. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 5 \cdot x_1 + 6 \cdot x_2 \leq 30; \\ 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$73. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 2 \cdot x_1 + 5 \cdot x_2 \leq 10; \\ 6 \cdot x_1 + 1 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

$$75. \quad \begin{cases} f(x_1, x_2) \rightarrow \min \\ 1 \cdot x_1 + 7 \cdot x_2 \leq 7; \\ 2 \cdot x_1 + 5 \cdot x_2 \leq 10; \\ 3 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{cases}$$

76. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 1 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
77. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 1 \cdot x_1 + 7 \cdot x_2 \leq 7; \\ 4 \cdot x_1 + 5 \cdot x_2 \leq 20; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
78. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 4 \cdot x_1 + 3 \cdot x_2 \leq 12; \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 5; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
79. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 6 \cdot x_1 + 5 \cdot x_2 \leq 30; \\ 1 \cdot x_1 + 6 \cdot x_2 \leq 6; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
80. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 6 \cdot x_1 + 3 \cdot x_2 \geq 18; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
81. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 5 \cdot x_1 + 6 \cdot x_2 \leq 30; \\ 6 \cdot x_1 + 3 \cdot x_2 \leq 18; \\ 2 \cdot x_1 + 7 \cdot x_2 \geq 14; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
82. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 2 \cdot x_1 + 5 \cdot x_2 \leq 10; \\ 2 \cdot x_1 + 3 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
83. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 6 \cdot x_1 + 6 \cdot x_2 \leq 36; \\ 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 5 \cdot x_1 + 2 \cdot x_2 \geq 10; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
84. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 1 \cdot x_1 + 1 \cdot x_2 \leq 6; \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 1 \cdot x_1 + 3 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
85. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 3 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
86. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
87. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 3 \cdot x_1 + 5 \cdot x_2 \leq 15; \\ 4 \cdot x_1 + 3 \cdot x_2 \leq 12; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
88. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 7 \cdot x_1 + 5 \cdot x_2 \leq 35; \\ 1 \cdot x_1 + 4 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
89. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 2 \cdot x_1 + 7 \cdot x_2 \leq 14; \\ 5 \cdot x_1 + 3 \cdot x_2 \leq 15; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
90. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 2 \cdot x_1 + 5 \cdot x_2 \leq 10; \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 25; \\ 3 \cdot x_1 + 1 \cdot x_2 \geq 3; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$
91. $f(x_1, x_2) \rightarrow \min$
 $\left\{ \begin{array}{l} 5 \cdot x_1 + 6 \cdot x_2 \leq 30; \\ 2 \cdot x_1 + 1 \cdot x_2 \leq 6; \\ 2 \cdot x_1 + 2 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$

$$92. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 5 \cdot x_1 + 7 \cdot x_2 \leq 35; \\ 6 \cdot x_1 + 3 \cdot x_2 \leq 18; \\ 7 \cdot x_1 + 1 \cdot x_2 \geq 7; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$94. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 7 \cdot x_1 + 4 \cdot x_2 \leq 28; \\ 5 \cdot x_1 + 7 \cdot x_2 \leq 35; \\ 3 \cdot x_1 + 5 \cdot x_2 \geq 15; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$96. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 7 \cdot x_1 + 3 \cdot x_2 \leq 21; \\ 5 \cdot x_1 + 6 \cdot x_2 \leq 30; \\ 2 \cdot x_1 + 1 \cdot x_2 \geq 2; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$98. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 6 \cdot x_1 + 2 \cdot x_2 \leq 12; \\ 2 \cdot x_1 + 4 \cdot x_2 \leq 8; \\ 4 \cdot x_1 + 1 \cdot x_2 \geq 4; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$93. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 3 \cdot x_1 + 7 \cdot x_2 \leq 21; \\ 6 \cdot x_1 + 4 \cdot x_2 \leq 24; \\ 4 \cdot x_1 + 3 \cdot x_2 \geq 12; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$95. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 1 \cdot x_1 + 6 \cdot x_2 \leq 6; \\ 5 \cdot x_1 + 4 \cdot x_2 \leq 20; \\ 3 \cdot x_1 + 2 \cdot x_2 \geq 6; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$97. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 5 \cdot x_1 + 5 \cdot x_2 \leq 25; \\ 3 \cdot x_1 + 4 \cdot x_2 \leq 12; \\ 7 \cdot x_1 + 2 \cdot x_2 \geq 14; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

$$99. \quad f(x_1, x_2) \rightarrow \min$$

$$\left\{ \begin{array}{l} 6 \cdot x_1 + 5 \cdot x_2 \leq 30; \\ 3 \cdot x_1 + 6 \cdot x_2 \leq 18; \\ 5 \cdot x_1 + 3 \cdot x_2 \geq 15; \\ x_1 \geq 0; x_2 \geq 0. \end{array} \right.$$

ПРИЛОЖЕНИЕ Б
(обязательное)
ВАРИАНТЫ ФУНКЦИЙ ЦЕЛИ

Функция цели задаётся в виде элементов матрицы C и вектора b ,
используемых в записи $f(X) = b^T X + \frac{1}{2} X^T C X$

№	c_{11}	c_{22}	$c_{12} = c_{21}$	b_1	b_2
(1)	(2)	(3)	(4)	(5)	(6)
0.	— 0.6	— 0.6	0.7	0.4	0.55
1.	— 0.35	— 0.35	0.65	0.45	— 0.3
2.	— 0,1	— 0,1	0,3	— 0,1	— 0,2
3.	— 0.7	— 0.7	0.6	0.7	0.35
4.	— 0.45	— 0.45	0.55	— 0.65	0.4
5.	— 0,2	— 0,2	0,4	— 0,1	2,0
6.	— 0.8	— 0.8	0.3	0.6	0.45
7.	— 0.65	— 0.65	0.35	0.55	— 0.7
8.	— 0.5	— 0.5	0.4	0.3	— 0.65
9.	— 0.55	— 0.55	0.45	— 0.4	0.6
10.	— 0,5	— 0,5	0,3	— 0,5	0,5
11.	— 0.45	— 0.45	0.7	0.35	— 0.55
12.	— 0.35	— 0.35	0.65	— 0.3	0.3
13.	— 0.4	— 0.4	0.6	— 0.6	0.55
14.	— 0.3	— 0.3	0.55	— 0.65	0.6
15.	— 0,1	— 0,1	0,8	1,0	2,5
16.	— 0.65	— 0.6	0.3	— 0.7	0.55
17.	— 0.65	— 0.65	0.35	— 0.75	0.3
18.	— 0.8	— 0.8	0.4	— 0.8	0.35
19.	— 0.55	— 0.55	0.45	0.85	— 0.4
20.	— 0.8	— 0.8	0.7	— 0.5	0.45
21.	— 1,0	— 1,0	0,8	1,0	2,5
22.	— 0.85	— 0.85	0.65	0.7	0.7
23.	— 0.5	— 0.5	0.6	— 0.65	0.65
24.	— 0.55	— 0.55	0.55	— 0.6	0.6
25.	— 0.75	— 0.75	0.3	— 0.55	0.50
26.	— 0.45	— 0.45	0.35	0.3	0.7

(1)	(2)	(3)	(4)	(5)	(6)
27.	— 0,8	— 0,8	1,2	0,8	0,8
28.	— 0.38	— 0.38	0.4	— 0.35	— 0.65
29.	— 0.88	— 0.88	0.45	0.4	0.6
30.	— 0.63	— 0.63	0.7	0.45	— 0.55
31.	— 0.67	— 0.67	0.65	— 0.7	0.3
32.	— 0.47	— 0.47	0.6	0.65	— 0.75
33.	— 1,25	— 1,25	1,75	— 1,5	— 0,5
34.	— 0.75	— 0.75	0.55	— 0.8	0.8
35.	— 0.83	— 0.83	0.3	0.85	0.85
36.	— 0.65	— 0.65	0.35	— 0.5	0.5
37.	— 0.53	— 0.53	0.4	0.55	— 0.45
38.	— 0.55	— 0.55	0.45	0.45	0.25
39.	— 0,27	— 0,27	0,8	— 0,5	0,5
40.	— 0.35	— 0.35	0.7	— 0.4	0.20
41.	— 0.42	— 0.42	0.65	— 0.85	0.15
42.	— 0,75	— 0,75	0,8	— 2,25	2,5
43.	— 0.35	— 0.35	0.6	— 0.3	0.4
44.	— 0.45	— 0.45	0.55	0.6	— 0.75
45.	— 0,25	— 0,25	0,75	— 1,0	0,5
46.	— 0.8	— 0.8	0.3	0.65	— 0.35
47.	— 0.61	— 0.61	0.6	— 0.7	0.5
48.	— 0.32	0.32	0.5	— 0.5	0.7
49.	— 0.85	— 0.85	0.35	0.25	0.4

Приложение В
(обязательное)
РЕЗУЛЬТАТЫ РЕШЕНИЯ ЗАДАЧИ КВАДРАТИЧНОГО
ПРОГРАММИРОВАНИЯ С ПРИМЕНЕНИЕМ ФУНКЦИИ `quadprog`

Матричные компоненты условия задачи

$$A = \begin{pmatrix} 2 & 5 \\ 3 & 2 \\ 1 & 1 \end{pmatrix}, \quad A_0 = \begin{pmatrix} 10 \\ 6 \\ 4 \end{pmatrix}, \quad b^T = (2 \quad 3), \quad C = \begin{pmatrix} -\frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{2} \end{pmatrix}.$$

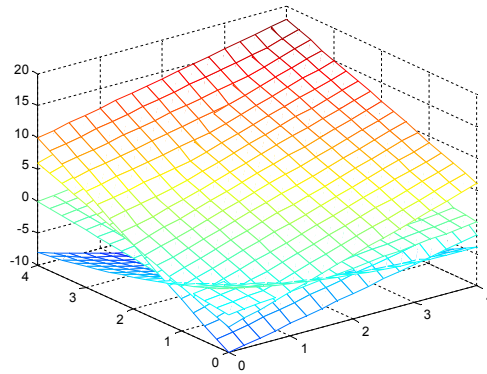


Рисунок В.1 – Пространственная картина

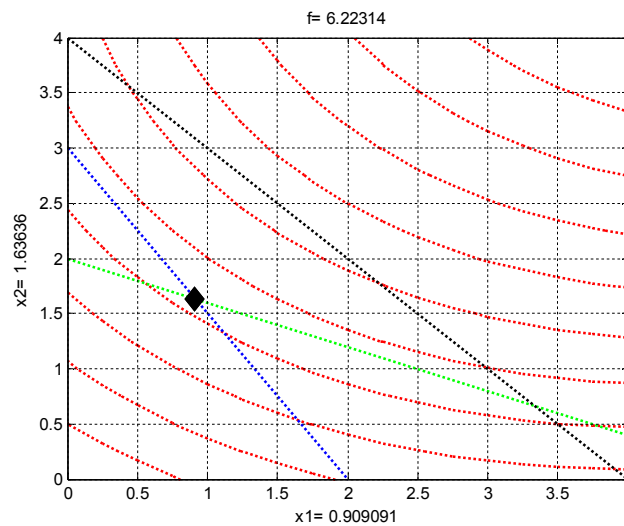


Рисунок В.2 – Линии равного уровня и точка оптимального решения

МЕТОДЫ ОПТИМИЗАЦИИ И НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

*Методические указания
к выполнению лабораторного практикума*

Составители: **Заикина** Елена Николаевна,
Карлусов Вадим Юрьевич

Издается в авторской редакции

Изд. № 221/19. Объем 3 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»