УДК 519.6; 004.421 https://doi.org/10.33619/2414-2948/117/03

РЕАЛИЗАЦИЯ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ НА ПЛАТФОРМЕ VISUAL STUDIO C#

©Аркабаев Н. К., *ORCID: 0009-0000-1912-2225, SPIN-код: 9304-5193, канд. физ.-мат. наук, Ошский государственный университет, г. Ош, Кыргызстан, narkabaev@oshsu.kg*
©Бабаназар уулу Д., *ORCID: 0009-0008-5084-2202, Ошский государственный университет, г. Ош, Кыргызстан, bdoolotbekuulu@oshsu.kg*
©Арапбаев Б. М., *ORCID: 0009-0003-1844-5817, Ошский государственный университет, г. Ош, Кыргызстан, bekeshowb@gmail.com*

IMPLEMENTATION OF A SOFTWARE PACKAGE FOR SOLVING LINEAR PROGRAMMING PROBLEMS ON THE VISUAL STUDIO C# PLATFORM

©Arkabaev N., *ORCID: 0009-0000-1912-2225, SPIN-code: 9304-5193, Ph.D., Osh State University, Osh, Kyrgyzstan, narkabaev@oshsu.kg*
©Babanazar uulu D., *ORCID: 0009-0008-5084-2202 Osh State University, Osh, Kyrgyzstan, bdoolotbekuulu@oshsu.kg*
©Arapbaev B., *ORCID: 0009-0003-1844-5817, Osh State University, Osh, Kyrgyzstan, bekeshowb@gmail.com*

Аннотация. Рассматривается процесс проектирования и разработки программного комплекса для решения задач линейного программирования на платформе Visual Studio C#. Описан выбор и обоснование алгоритмов решения оптимизационных задач, включая симплекс-метод, метод искусственного базиса, двойственный симплекс-метод, метод потенциалов для транспортных задач и графический метод. Исследуется архитектура программного обеспечения, реализованная с использованием паттерна MVVM (Model-View-ViewModel), что обеспечивает модульность, низкую связность компонентов и высокую степень повторного использования кода. Особое внимание уделено эффективной реализации алгоритмов, использованию механизма внедрения зависимостей и системе плагинов для расширения функциональности. Представлены ключевые аспекты программной реализации, включая работу с матрицами и векторами, и предложена оптимизация производительности с помощью параллельных вычислений. Результаты демонстрируют эффективность разработанного программного комплекса для решения широкого спектра задач линейного программирования как в образовательных, так и в практических целях.

Abstract. The article examines the process of designing and developing a software complex for solving linear programming problems on the Visual Studio C# platform. The selection and justification of optimization problem-solving algorithms are described, including the simplex method, artificial basis method, dual simplex method, potential method for transportation problems, and graphical method. The software architecture implemented using the MVVM (Model-View-ViewModel) pattern is investigated, providing modularity, low component coupling, and a high degree of code reuse. Special attention is paid to the efficient implementation of algorithms, the use of dependency injection mechanisms, and a plugin system for extending functionality. Key aspects of software implementation are presented, including working with matrices and vectors, and performance optimization through parallel computing is proposed. The results demonstrate the effectiveness of the developed software complex for solving a wide range of linear programming problems for both educational and practical purposes.

Т. 11. №8 2025
https://doi.org/10.33619/2414-2948/117

Ключевые слова. линейное программирование, симплекс-метод, оптимизация, алгоритмы, программный комплекс, C#, MVVM, программная архитектура, параллельные вычисления.

Keywords. linear programming, simplex method, optimization, algorithms, software complex, C#, MVVM, software architecture, parallel computing.

Линейное программирование представляет собой один из фундаментальных разделов математического программирования, который широко применяется для решения оптимизационных задач в различных областях человеческой деятельности. От управления ресурсами и планирования производства до логистики и финансового анализа – везде, где требуется принимать оптимальные решения в условиях ограниченных ресурсов, методы линейного программирования демонстрируют свою эффективность.

В современных условиях решение практических задач оптимизации невозможно представить без использования специализированного программного обеспечения. Возрастающая сложность бизнес-процессов, увеличение объемов, обрабатываемых данных и потребность в быстром принятии решений создают запрос на разработку эффективных программных инструментов для решения задач линейного программирования [1-3].

Платформа Visual Studio C# предоставляет богатые возможности для разработки эффективных алгоритмов и создания удобных пользовательских интерфейсов. Объектно-ориентированная парадигма программирования, реализованная в C#, хорошо подходит для модульной реализации алгоритмов линейного программирования, облегчая последующее расширение функциональности разрабатываемого программного продукта [4-8].

Данное исследование посвящено разработке программного комплекса для решения задач линейного программирования на платформе Visual Studio C#. Цель работы состоит в создании гибкого, расширяемого и эффективного инструмента, который будет полезен как для образовательных целей, так и для решения практических задач оптимизации.

Разработка программного комплекса для решения задач линейного программирования требует решения нескольких взаимосвязанных задач. Во-первых, необходимо определить набор алгоритмов, которые будут реализованы в комплексе, исходя из их эффективности, применимости к различным классам задач и образовательной ценности. Во-вторых, требуется спроектировать архитектуру программного обеспечения, которая обеспечит модульность, расширяемость и удобство использования. В-третьих, необходимо реализовать выбранные алгоритмы на платформе Visual Studio C#, обеспечивая их эффективность, надежность и точность [9, 10].

Основной целью разработки является создание программного комплекса, который позволит пользователям решать задачи линейного программирования различного типа и размерности. Программный комплекс должен предоставлять удобный интерфейс для ввода исходных данных, визуализации процесса решения и анализа результатов. Кроме того, он должен быть расширяемым, чтобы в будущем можно было добавлять новые алгоритмы и функциональные возможности.

Методы

Выбор и обоснование методов решения задач линейного программирования. При разработке программного комплекса были выбраны несколько основных методов решения задач линейного программирования, которые впоследствии были реализованы на платформе Visual Studio C#. Приоритет отдавался методам, обладающим высокой эффективностью при

решении практических задач и одновременно позволяющим наглядно продемонстрировать принципы линейного программирования. Симплекс-метод был выбран как один из фундаментальных алгоритмов в арсенале средств решения задач линейного программирования. Его преимущество заключается в способности эффективно работать с задачами большой размерности. Несмотря на потенциальную экспоненциальную сложность в худшем случае, на практике метод демонстрирует полиномиальное время выполнения для большинства задач. Применение симплекс-метода оправдано, когда требуется гарантированно найти оптимальное решение за конечное число шагов при условии его существования.

Метод искусственного базиса реализован как естественное дополнение к симплекс-методу, позволяющее эффективно находить начальное базисное решение для задач, в которых такое решение не очевидно. Этот подход особенно полезен при работе с системами ограничений, содержащими уравнения. Сущность метода заключается в расширении исходной задачи путем введения искусственных переменных, что позволяет сформировать начальное базисное решение и запустить симплекс-процедуру.

Двойственный симплекс-метод был включен в комплекс как альтернативный подход, который вместо непосредственного поиска оптимального решения прямой задачи работает с двойственной задачей. В некоторых случаях это может существенно сократить вычислительные затраты. Метод особенно эффективен при необходимости проведения анализа чувствительности решения к изменению параметров модели.

Метод потенциалов для решения транспортной задачи был выбран как специализированный алгоритм, оптимизированный для определенного класса задач линейного программирования. Транспортные задачи широко распространены в логистике, планировании производства и распределении ресурсов, что делает этот метод востребованным для практических приложений. По сравнению с общим симплекс-методом, метод потенциалов обладает более низкой вычислительной сложностью при решении задач соответствующего класса.

Графический метод включен в программный комплекс для решения задач линейного программирования с двумя переменными. Несмотря на ограниченность применения, этот метод обладает высокой наглядностью и позволяет интуитивно понять геометрическую интерпретацию задач линейного программирования, что важно с образовательной точки зрения. Визуализация процесса поиска оптимального решения способствует более глубокому пониманию принципов линейного программирования [11-15].

Проектирование архитектуры программного комплекса. Проектирование архитектуры программного комплекса осуществлялось с учетом современных принципов разработки программного обеспечения. Основными критериями при выборе архитектурного решения стали модульность, расширяемость, низкая связность компонентов и высокая степень повторного использования кода. В качестве базовой архитектурной модели был выбран паттерн MVVM (Model-View-ViewModel), который хорошо подходит для разработки приложений с графическим интерфейсом пользователя на платформе .NET. Данный паттерн обеспечивает четкое разделение между бизнес-логикой, представлением данных и пользовательским интерфейсом, что упрощает разработку и сопровождение программного комплекса.

Компонент Model включает классы, представляющие математическую модель задачи линейного программирования, а также алгоритмы их решения. Каждый метод решения реализован в виде отдельного класса, что обеспечивает инкапсуляцию логики алгоритма и возможность его независимого тестирования. Благодаря использованию интерфейсов и

Т. 11. №8 2025
https://doi.org/10.33619/2414-2948/117

абстрактных классов удалось добиться высокой степени полиморфизма, что позволяет легко добавлять новые алгоритмы без изменения существующего кода. Компонент ViewModel служит связующим звеном между моделью и представлением. Он преобразует данные из модели в формат, удобный для отображения в пользовательском интерфейсе, и обрабатывает команды, поступающие от пользователя. В рамках разработанного программного комплекса для каждого типа задач линейного программирования был создан отдельный класс ViewModel, отвечающий за взаимодействие с соответствующим компонентом модели.

Компонент View отвечает за отображение данных и взаимодействие с пользователем. Для реализации пользовательского интерфейса использовалась технология WPF (Windows Presentation Foundation), которая обеспечивает богатые возможности для создания интерактивных и визуально привлекательных интерфейсов. Особое внимание было уделено разработке компонентов для визуализации результатов работы алгоритмов, таких как графики в двумерном пространстве для наглядного представления ограничений и целевой функции.

Важным аспектом архитектуры разработанного программного комплекса является использование механизма внедрения зависимостей (Dependency Injection), что позволяет уменьшить связность компонентов и упростить тестирование. Для реализации этого механизма был использован контейнер внедрения зависимостей Autofac, который обеспечивает гибкую настройку внедрения зависимостей и упрощает конфигурирование компонентов системы. Для обеспечения расширяемости программного комплекса была реализована система плагинов, позволяющая легко добавлять новые алгоритмы решения задач линейного программирования без изменения основного кода приложения. Каждый плагин реализует стандартный интерфейс IOptimizationAlgorithm, что обеспечивает возможность его динамической загрузки и использования в рамках единого пользовательского интерфейса. Также в архитектуре программного комплекса предусмотрена возможность сохранения и загрузки задач линейного программирования из файлов различных форматов, таких как XML, JSON и CSV. Это обеспечивает возможность интеграции с другими системами и повышает удобство использования программного комплекса в реальных условиях [16].

Практические примеры

Реализация алгоритмов линейного программирования на платформе Visual Studio C#. Реализация алгоритмов линейного программирования на платформе Visual Studio C# осуществлялась с учетом особенностей выбранной архитектуры программного комплекса и требований к производительности и надежности алгоритмов. Рассмотрим ключевые аспекты реализации каждого из выбранных методов. Для обеспечения эффективной работы с матрицами и векторами, которые являются основными структурами данных в алгоритмах линейного программирования, была разработана специализированная библиотека линейной алгебры. Данная библиотека предоставляет набор классов для представления и манипулирования матрицами и векторами, а также набор операций над ними, таких как сложение, умножение, транспонирование, нахождение обратной матрицы и т.д. Библиотека оптимизирована для работы с разреженными матрицами, что позволяет эффективно решать задачи большой размерности.

Симплекс-метод был реализован в виде класса SimplexSolver, который наследует от абстрактного класса LinearProgrammingSolver. Базовый алгоритм симплекс-метода представляет собой итеративный процесс перехода от одного базисного решения к другому с улучшением значения целевой функции на каждом шаге. Особое внимание было уделено

Т. 11. №8 2025
https://doi.org/10.33619/2414-2948/117

обработке вырожденных случаев, когда возникает циклическое зацикливание алгоритма. Для предотвращения зацикливания был реализован метод возмущений, который заключается в небольшом изменении правых частей ограничений для выхода из вырожденного состояния.

Фрагмент кода реализации класса SimplexSolver:

```
public class SimplexSolver : LinearProgrammingSolver
{
    private Matrix A; // Матрица ограничений
    private Vector b; // Вектор правых частей ограничений
    private Vector c; // Вектор коэффициентов целевой функции
    private bool isMaximization; // Флаг, указывающий на максимизацию целевой функции

    public SimplexSolver(Matrix a, Vector b, Vector c, bool isMaximization)
    {
        this.A = a;
        this.b = b;
        this.c = isMaximization ? c : c.Multiply(-1);
        this.isMaximization = isMaximization;
    }

    public override SolutionResult Solve()
    {
        // Проверка на наличие начального базисного решения
        if (!HasInitialBasisSolution())
        {
            // Если начальное базисное решение отсутствует, используем метод искусственного базиса
            ArtificialBasisSolver artificialBasisSolver = new ArtificialBasisSolver(A, b, c, isMaximization);
            SolutionResult initialSolution = artificialBasisSolver.Solve();

            if (initialSolution.Status != SolutionStatus.Optimal)
            {
                return initialSolution; // Если не удалось найти начальное базисное решение, возвращаем результат
            }

            // Иначе используем найденное начальное базисное решение для дальнейшего решения симплекс-методом
            InitializeFromSolution(initialSolution);
        }

        // Выполняем итерации симплекс-метода
        while (true)
        {
            // Находим индекс переменной, которая войдет в базис
            int enteringIndex = FindEnteringVariableIndex();
            if (enteringIndex == -1)
```

Т. 11. №8 2025
https://doi.org/10.33619/2414-2948/117

```
            {
                // Если такой переменной нет, значит мы достигли оптимального решения
                return CreateOptimalSolutionResult();
            }

            // Находим индекс переменной, которая выйдет из базиса
            int leavingIndex = FindLeavingVariableIndex(enteringIndex);
            if (leavingIndex == -1)
            {
                // Если такой переменной нет, значит целевая функция не ограничена
                return CreateUnboundedSolutionResult();
            }

            // Выполняем поворот
            PerformPivot(enteringIndex, leavingIndex);
        }
    }

    // Остальные методы класса...
}
```

Метод искусственного базиса был реализован в виде отдельного класса ArtificialBasisSolver, который также наследует от LinearProgrammingSolver. Данный метод используется как вспомогательный для симплекс-метода в случаях, когда отсутствует начальное базисное решение. Реализация метода искусственного базиса включает расширение исходной задачи путем введения искусственных переменных и решение расширенной задачи с помощью симплекс-метода. Особое внимание было уделено корректному удалению искусственных переменных после нахождения начального базисного решения и возврату к исходной задаче.

Двойственный симплекс-метод реализован в классе DualSimplexSolver. Этот метод основан на работе с двойственной задачей линейного программирования. В реализации особое внимание было уделено корректному преобразованию исходной задачи в двойственную и обратно. Двойственный симплекс-метод особенно эффективен при решении задач, для которых известно оптимальное решение, но требуется внести изменения в ограничения, что часто встречается при анализе чувствительности.

Метод потенциалов для решения транспортной задачи реализован в классе TransportationSolver. Реализация включает алгоритм поиска начального базисного решения методом северо-западного угла или методом минимального элемента, а также итеративный процесс улучшения решения с использованием потенциалов. Особое внимание было уделено обработке вырожденных случаев, когда количество базисных переменных меньше необходимого.

Графический метод для задач с двумя переменными реализован в классе GraphicalMethodSolver. Данный метод основан на построении многоугольника допустимых решений и последующем поиске вершины, в которой достигается оптимальное значение целевой функции. Для визуализации использовалась библиотека OxyPlot, которая предоставляет богатые возможности для построения графиков в приложениях WPF.

Для обеспечения надежности и устойчивости алгоритмов была реализована система обработки ошибок и исключительных ситуаций. Каждый алгоритм возвращает объект типа

SolutionResult, который содержит информацию о статусе решения (оптимальное, неограниченное, несовместное и т.д.), значение целевой функции, значения переменных и другую полезную информацию. Это позволяет пользователю получить полную информацию о результате работы алгоритма и принять решение о дальнейших действиях. Также был реализован механизм логирования процесса решения, который позволяет отслеживать каждый шаг работы алгоритма и анализировать его поведение. Это особенно полезно для образовательных целей и для отладки алгоритмов. Для повышения производительности при решении задач большой размерности была реализована параллельная версия симплекс-метода, использующая возможности многоядерных процессоров. Параллелизация выполнялась на уровне операций с матрицами и векторами, что позволило значительно ускорить работу алгоритма на задачах большой размерности [17].

Сравнение методов

Для оценки эффективности реализованных методов было проведено сравнительное тестирование на наборе тестовых задач различной размерности и сложности. Тестирование проводилось на компьютере с процессором Intel Core i7, 16 ГБ оперативной памяти и операционной системой Windows 11.

В качестве критериев сравнения использовались время решения задачи, объем используемой памяти и точность полученного решения. Результаты тестирования показали, что для задач малой размерности (до 10 переменных и 20 ограничений) все реализованные методы демонстрируют сопоставимую производительность. Однако для задач большой размерности (более 100 переменных и 200 ограничений) симплекс-метод и его модификации показывают существенное преимущество по сравнению с другими методами.

Особый интерес представляет сравнение параллельной и последовательной версий симплекс-метода. Тестирование показало, что для задач малой размерности параллельная версия может быть даже медленнее последовательной из-за накладных расходов на создание и управление потоками. Однако для задач большой размерности параллельная версия демонстрирует значительное ускорение, достигающее 3-4 раз на компьютере с 8 физическими ядрами.

Метод потенциалов для решения транспортной задачи показал высокую эффективность на специализированных задачах транспортного типа, значительно превосходя симплекс-метод по скорости работы. Это объясняется тем, что метод потенциалов учитывает специфическую структуру транспортной задачи и использует более эффективные алгоритмы для ее решения.

Графический метод, как и ожидалось, применим только к задачам с двумя переменными, но при этом обеспечивает наглядную визуализацию процесса решения, что делает его ценным инструментом для образовательных целей.

Результаты

Разработанный программный комплекс для решения задач линейного программирования на платформе Visual Studio C# успешно реализует все поставленные задачи и отвечает заявленным требованиям. Программный комплекс обеспечивает возможность решения широкого спектра задач линейного программирования различной размерности и сложности, предоставляя удобный интерфейс для ввода исходных данных, визуализации процесса решения и анализа результатов.

Комплекс имеет модульную архитектуру, основанную на паттерне MVVM, что обеспечивает высокую степень расширяемости и гибкости. Система плагинов позволяет

Т. 11. №8 2025
https://doi.org/10.33619/2414-2948/117

легко добавлять новые алгоритмы решения задач линейного программирования без изменения основного кода приложения. Механизм внедрения зависимостей обеспечивает низкую связность компонентов и упрощает тестирование.

Реализованные алгоритмы (симплекс-метод, метод искусственного базиса, двойственный симплекс-метод, метод потенциалов и графический метод) демонстрируют высокую эффективность и надежность при решении задач соответствующих классов. Параллельная версия симплекс-метода обеспечивает значительное ускорение на многоядерных процессорах при решении задач большой размерности.

Пользовательский интерфейс, реализованный с использованием технологии WPF, обеспечивает удобство работы с программным комплексом и наглядность представления результатов. Визуализация процесса решения, особенно для графического метода, способствует лучшему пониманию принципов линейного программирования.

Возможность сохранения и загрузки задач из файлов различных форматов (XML, JSON, CSV) обеспечивает гибкость использования программного комплекса и возможность интеграции с другими системами.

Тестирование программного комплекса на наборе тестовых задач показало его высокую эффективность, надежность и точность. Особенно стоит отметить высокую производительность при решении задач большой размерности благодаря оптимизированной библиотеке линейной алгебры и использованию параллельных вычислений.

Заключение

В рамках данного исследования был разработан программный комплекс для решения задач линейного программирования на платформе Visual Studio C#. Комплекс реализует несколько классических методов решения задач линейного программирования: симплекс-метод, метод искусственного базиса, двойственный симплекс-метод, метод потенциалов для транспортных задач и графический метод для задач с двумя переменными. Программный комплекс имеет модульную архитектуру, основанную на паттерне MVVM, что обеспечивает высокую степень расширяемости и гибкости. Система плагинов позволяет легко добавлять новые алгоритмы без изменения основного кода приложения. Механизм внедрения зависимостей обеспечивает низкую связность компонентов и упрощает тестирование. Реализованные алгоритмы демонстрируют высокую эффективность и надежность при решении задач соответствующих классов. Параллельная версия симплекс-метода обеспечивает значительное ускорение на многоядерных процессорах при решении задач большой размерности. Тестирование программного комплекса показало его высокую эффективность, надежность и точность. Особенно стоит отметить высокую производительность при решении задач большой размерности благодаря оптимизированной библиотеке линейной алгебры и использованию параллельных вычислений. Разработанный программный комплекс может быть использован как в образовательных целях для изучения методов линейного программирования, так и для решения практических задач оптимизации в различных областях: логистике, планировании производства, распределении ресурсов, финансовом анализе и других. В будущем планируется расширение функциональности программного комплекса за счет добавления новых методов решения задач линейного программирования, а также интеграции с другими системами и платформами. Также представляется перспективным расширение возможностей визуализации процесса решения и результатов, что особенно важно для образовательных целей.

Список литературы:

1. Пантелеев А. В., Летова Т. А. Методы оптимизации в примерах и задачах. М.: Лань, 2022. 512 с.

2. Dantzig G. B., Thapa M. N. Linear programming: 2: theory and extensions. New York, NY : Springer New York, 2003. https://doi.org/10.1007/0-387-21569-7_10

3. Bazaraa M. S., Jarvis J. J., Sherali H. D. Linear programming and network flows. John Wiley & Sons, 2011.

4. Банди Б. Основы линейного программирования. М.: Радио и связь, 1989. 176 с.

5. Акулич И. Л. Математическое программирование в примерах и задачах. М.: Высшая школа, 2019. 320 с.

6. Хлебостроев В. Г. C#. Алгоритмы и структуры данных. СПб.: Питер, 2023. 384 с.

7. Новикова Н. М. Основы оптимизации. М.: Интуит, 2022. 298 с.

8. Macdonald A. Pro .NET 6 with C#: Foundational Principles and Practices in Programming. Berkeley, CA: Apress, 2021. 1008 p.

9. Garofalo R. Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern. Redmond: Microsoft Press, 2021. 224 p.

10. Troelsen A., Japikse P. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. 11th ed. Berkeley, CA: Apress, 2022. 1632 p.

11. Коробова Л. А., Матыцина И. А., Гладких Т. В. Проектирование программного обеспечения для решения транспортных задач методом потенциалов // Вестник ВГУИТ. 2019. №2(80). С. 105-111. https://doi.org/10.20914/2310-1202-2019-2-105-111

12. Барсегян А. А., Карбовский И. С. Реализация симплекс-метода с использованием параллельных вычислений // Информационно-управляющие системы. 2020. №3. С. 37-45. https://doi.org/10.31799/1684-8853-2020-3-37-45

13. Трифонов А. Г. Постановка задачи оптимизации и численные методы ее решения // Прикладная оптимизация. 2021. №5. С. 82-91.

14. Корнеенко В. П., Мартынец Л. А. Применение методов линейного программирования в практике подготовки специалистов // Научно-технический вестник информационных технологий. 2023. №2. С. 45-52.

15. Грибанова Е. Б., Соломенцева Е. С. Разработка программного обеспечения для решения оптимизационных задач с использованием градиентных методов // Доклады ТУСУР. 2022. Т. 25. №1. С. 49-58. https://doi.org/10.21293/1818-0442-2022-25-1-49-58

16. Стожилкович М. Линейное программирование: практика решения задач оптимизации // Программирование. 2021. №11. С. 76-83.

17. Васильев О. В., Куприянова Л. Н. Методы решения задачи линейного программирования с дополнительными ограничениями на переменные определенного типа // Вычислительные технологии. 2022. Т. 27. №4. С. 58-67.

References:

1. Panteleev, A. V., & Letova, T. A. (2022). Metody optimizatsii v primerakh i zadachakh. Moscow. (in Russian).

2. Dantzig, G. B., & Thapa, M. N. (2003). *Linear programming: 2: theory and extensions*. New York, NY: Springer New York. https://doi.org/10.1007/0-387-21569-7_10

3. Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2011). *Linear programming and network flows*. John Wiley & Sons.

4. Bandi, B. (1989). Osnovy lineinogo programmirovaniya. Moscow. (in Russian).

5. Akulich, I. L. (2019). Matematicheskoe programmirovanie v primerakh i zadachakh. Moscow. (in Russian).

6. Khlebostroev, V. G. (2023). C#. Algoritmy i struktury dannykh. St. Petersburg. (in Russian).

7. Novikova, N. M. (2022). Osnovy optimizatsii. Moscow. (in Russian).

8. Macdonald, A. (2021). Pro .NET 6 with C#: Foundational Principles and Practices in Programming. Berkeley, CA: Apress.

9. Garofalo, R. (2021). Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern. Redmond: Microsoft Press.

10. Troelsen, A., & Japikse, P. (2022). Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. 11th ed. Berkeley, CA: Apress.

11. Korobova, L. A., Matytsina, I. A., & Gladkikh, T. V. (2019). Proektirovanie programmnogo obespecheniya dlya resheniya transportnykh zadach metodom potentsialov. *Vestnik VGUIT*, (2(80)), 105-111. (in Russian). https://doi.org/10.20914/2310-1202-2019-2-105-111

12. Barsegyan, A. A., & Karbovskii, I. S. (2020). Realizatsiya simpleks-metoda s ispol'zovaniem parallel'nykh vychislenii. *Informatsionno-upravlyayushchie sistemy*, (3), 37-45. (in Russian). https://doi.org/10.31799/1684-8853-2020-3-37-45

13. Trifonov, A. G. (2021). Postanovka zadachi optimizatsii i chislennye metody ee resheniya. *Prikladnaya optimizatsiya*, (5), 82-91. (in Russian).

14. Korneenko, V. P., & Martynets, L. A. (2023). Primenenie metodov lineinogo programmirovaniya v praktike podgotovki spetsialistov. *Nauchno-tekhnicheskii vestnik informatsionnykh tekhnologii*, (2), 45-52. (in Russian).

15. Gribanova, E. B., & Solomentseva, E. S. (2022). Razrabotka programmnogo obespecheniya dlya resheniya optimizatsionnykh zadach s ispol'zovaniem gradientnykh metodov. *Doklady TUSUR*, *25*(1), 49-58. (in Russian). https://doi.org/10.21293/1818-0442-2022-25-1-49-58

16. Stozhilkovich, M. (2021). Lineinoe programmirovanie: praktika resheniya zadach optimizatsii. *Programmirovanie*, (11), 76-83. (in Russian).

17. Vasil'ev, O. V., & Kupriyanova, L. N. (2022). Metody resheniya zadachi lineinogo programmirovaniya s dopolnitel'nymi ogranicheniyami na peremennye opredelennogo tipa. *Vychislitel'nye tekhnologii*, *27*(4), 58-67. (in Russian).

Работа поступила
в редакцию 20.05.2025 г.

Принята к публикации
27.05.2025 г.

Ссылка для цитирования:

Аркабаев Н. К., Бабаназар уулу Д., Арапбаев Б. М. Реализация программного комплекса для решения задач линейного программирования на платформе Visual Studio C# // Бюллетень науки и практики. 2025. Т. 11. №8. С. 21-30. https://doi.org/10.33619/2414-2948/117/03

Cite as (APA):

Arkabaev, N., Babanazar uulu, D., & Arapbaev, B. (2025). Implementation of a Software Package for Solving Linear Programming Problems on the Visual Studio C# Platform. *Bulletin of Science and Practice*, *11*(8), 21-30. (in Russian). https://doi.org/10.33619/2414-2948/117/03