САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Петрович Юрий Михайлович

Выпускная квалификационная работа

Автоматизация оптимального распределения транспортных ресурсов предприятия

Уровень образования: бакалавриат
Направление 02.03.02 «Фундаментальная информатика и информационные технологии»
Основная образовательная программа СВ.5003.2018 «Программирование и информационные технологии»

Научный руководитель:
Кандидат физико-математических наук,
Доцент кафедры механики управляемого движения
Потоцкая Ирина Юрьевна

Рецензент:
Пупышев Михаил Юрьевич

Санкт-Петербург
2021

Содержание

Введение

В современном мире любое производство, бизнес или любой другой вид деятельности сталкивается с проблемой оптимизации своей работы с целью минимизации затрат трудовых, финансовых и других ресурсов.

Одной из важнейших задач предприятий, занимающихся каким-либо производством, доставкой грузов или перевозкой пассажиров, является оптимальное распределение имеющихся транспортных ресурсов по тем или иным маршрутам с учетом грузоподъемности, вместимости груза или пассажиров, а также иных факторов. Необходимо не только обеспечить каждый маршрут достаточным количеством транспортных средств для перевозки нужного объема груза, но и сделать это так, чтобы расходы организации были снижены до минимума.

В качестве примера предприятий, сталкивающихся с такой задачей регулярно, можно привести организации, занимающиеся вывозом мусора из крупных городов. Разработка алгоритмов и систем управления вывозом твердых бытовых отходов из крупных мегаполисов является актуальной как по причине увеличения количества этих отходов, так и увеличения трафика в городах. В каждом городе существуют специальные организации, ответственные за сбор и вывоз бытовых отходов на полигоны или специализированные предприятия по их переработке. В данный момент ведется разработка системы «Чистый город», целью создания которой является динамическое составление маршрутов и оптимальное распределение по ним транспортных ресурсов. Ниже приведена схема работы вышеописанной системы [1]:

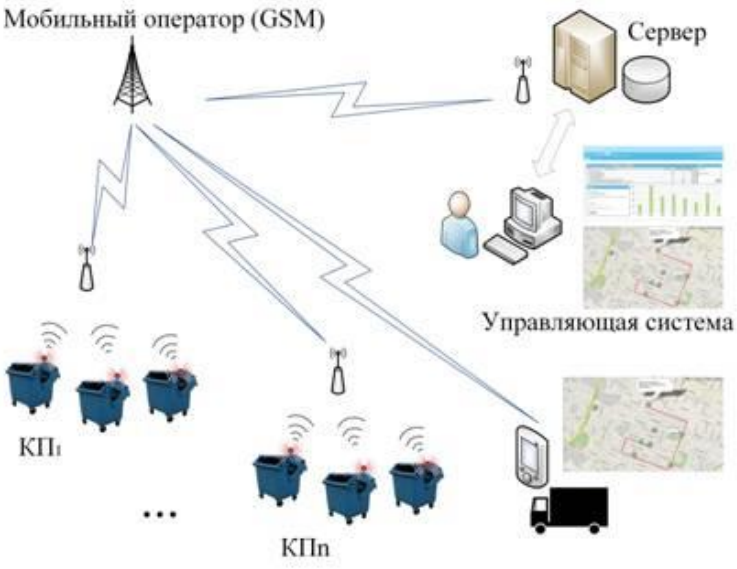

Рисунок 1 – Схема работы системы «Чистый город»

Понятно, что существует еще множество приложений задачи оптимизации распределения транспорта к различным предметным областям.

В данной выпускной работе будет рассмотрела проблема нахождения оптимального распределения транспортных средств по маршрутам. Способы её решения будут заключатся в различных подходах к решению задач целочисленного линейного программирования.

Постановка задачи

Цель работы – разработка алгоритма оптимального распределения транспортных ресурсов предприятия на основе методов целочисленного линейного программирования. Далее поставленные задачи описаны более подробно:

1. Построить математическую модель оптимизируемого процесса, используя инструменты целочисленного линейного программирования (далее – ЦЛП).
2. Ознакомиться с проблемой решения задач ЦЛП, а также существующими методами решения.
3. Выбрать наиболее подходящий метод решения для программной реализации, исходя из особенностей построенной математической модели.
4. Реализовать выбранный подход в виде программы с пользовательским интерфейсом.
5. Провести анализ работы созданной системы, а также эффективности используемого алгоритма.

Обзор литературы

В приложенной литературе описывается постановка задачи ЦЛП, различные подходы к ее решению, а также проблемы, связанные с решением данного класса задач.

Приложение [1] рассказывает об устройстве и назначении системы «Умный город», которая была рассмотрела для более глубокого понимания решаемой проблемы.

Далее следуют пособия [2] и [3], в которых описывается формальная постановка решаемой задачи, а также описание существующих подходов к ее решению. Данные ресурсы стали базой для понимания остальной использованной литературы.

В статье [4] приведено описания метода Гомори решения класса задач ЦЛП, описанного самим создателем метода – Ральфом Гомори.

Диссертация [5] и книга [6] использовались для выбора конкретного алгоритма и для его последующей реализации. Кроме этого, в [6] приведены результаты различных исследований и экспериментов, связанных алгоритмами решения задач ЦЛП. Также были прочитаны статьи [7], [8] и [9], в которых приведены исследования алгоритмов Гомори, информация, взятая из них повлияла на выбор реализуемого алгоритма.

1. Изучение проблемы решения задач ЦЛП

Начнем с того, что задача ЦЛП является NP-трудной задачей (то есть она принадлежит к классу задач, которые на данный момент нельзя решить за полиномиальное время), поэтому все существующие на данный момент решения весьма ограничены в применимости и имеют существенные недостатки. В стандартном виде задача представляется так:

$$c^T x \to max, \tag{1}$$

При условиях:

$$\begin{gathered} Ax = b, \\ x \geq 0, \\ x \in Z^n, \end{gathered} \tag{2}$$

Где c, b – векторы, а A – матрица.

Далее рассмотрены основные методы решения рассматриваемого класса задач.

1.1. Методы ветвей и границ

Для начала рассматривается семейство методов, называемое методами ветвей и границ, основанных на разбиении множества допустимых решений на подмножества (ветвление), а также на оценивании целевой функции на этих подмножествах (вычислении границ).

Общая схема метода заключается в последовательном разбиении допустимого множества на ряд новых, что приводит к порождению меньших по размерности задач, каждая из которых также может быть разбита на меньшие. Для того, чтобы избежать перебора «неперспективных» подзадач используются два типа оценок – нижние границы для значений целевой функции на подмножестве допустимых решений и верхние границы для оптимального значения целевой функции.

Данный метод является конечным при конечности множества допустимых решений. Из-за излишнего перебора данные методы показывают существенный рост времени вычислений при росте размера задач.

1.2. Методы отсечений

Рассмотрим теперь семейство методов, основанных на другом подходе к решению задач ЦЛП. Идея метода отсечений заключается в следующем:

1) Исходная задача $(L^{\text{ц}}, C)$ будет решаться с помощью многоэтапного процесса, причем на r-м этапе будет решатся вспомогательная задача линейного программирования (L_r, C). Здесь $L_0 = L$.
2) Множество целочисленных точек одинаково для всех многогранников, а значит если оптимальный план задачи (L_r, C) удовлетворяет условию целочисленности, то он окажется также и оптимальным планом исходной задачи. Если же план не является целочисленным, то он не будет являться планом задачи (L_{r+1}, C).
3) Переход от r-го этапа решения задачи к $(r+1)$-му этапу в случае нецелочисленности плана $X(L_r, C)$ осуществляется с помощью правильного отсечения:

$$\alpha_r X \leq \beta_r$$

Добавление которого к линейным условиям задачи (L_r, C) превращает многогранник L_r в L_{r+1}. Правильным отсечением здесь называется новое условие, которому удовлетворяют все целочисленные планы исходной задачи, но не удовлетворяет план задачи (L_r, C).

Указанные идеи можно реализовать лишь в случае указания способа построения правильных отсечений, обеспечивающих конечность процесса решения.

Кроме того, появляется важная проблема чрезмерного увеличения количества правильных отсечений в случае, если процесс окажется достаточно продолжительным.

Для решения поставленной в данной работе задачи был выбран алгоритм, являющийся исторически первым, для которого построение

правильных ограничений было описано алгоритмически и доказана конечность работы. Указанный алгоритм носит название первого алгоритма Гомори.

Отметим, что всего существует три алгоритма Гомори, все они рассматривались в качества кандидатов для реализации.

Третий алгоритм Гомори было решено не использовать по причине его требований к целочисленности всех коэффициентов задачи, что существенно сужает возможности реализуемой системы.

Второй алгоритм решает расширенный класс задач, а именно – частично целочисленные задачи (в том числе и полностью целочисленные). Он отличается от первого алгоритма лишь способом построения правильного отсечения, но имеет существенный недостаток в контексте рассматриваемой в данной работе задачи. Проблема заключается в том, что при построении отсечения по второму алгоритму используется операция деления, что приводит к усугублению проблемы вычислительных неточностей (далее в работе будет указано как эта проблема решалась при реализации). Этот алгоритм был реализован, но подобрать оптимальные условия округления так, чтобы не нарушить корректность работы алгоритма не удалось.

Далее будет описана схема работы первого алгоритма Гомори в приложении к рассматриваемой задаче нахождения оптимального распределения транспортных ресурсов по маршрутам, а также улучшения алгоритма, указанного его автором и дающего преимущество в виде ограничения на количество неравенств и переменных в задаче.

2. Реализация метода в приложении к рассматриваемой задаче

В данном разделе будет описана постановка задачи, а также детально описана работа первого алгоритма Гомори, выбранного для реализации.

2.1. Входные данные задачи

Рассмотрим данные, которые пользователь приложения сможет вводить для последующего решения задачи:

- Таблица со списком маршрутов и их характеристик, вида:
  «Название маршрута – протяженность – Характеристика 1 – Характеристика 2 - …»
- Таблица со списком транспортных средств и их характеристик, вида:
  «Типа ТС (название) – расходы на использование в день – расход на горючее (на км) – Характеристика 1 – Характеристика 2 - …»

Характеристики транспортных средств удовлетворяют потребности маршрутов, указанных в полях их характеристик с соответствующими номерами.

Ограничениями задачи будут являться:

- Количество транспортных средств каждого типа.
- Удовлетворение всех потребностей маршрутов, указанных в графах характеристик.
- Количество задействованных транспортных средств каждого типа – целые числа.

Целевой функцией будут являться суммарные расходы на обслуживание и использование транспортных средств.

Задача – минимизировать целевую функцию.

В следующем разделе будет приведена математическая постановка данной задачи.

2.2. Математическая постановка задачи

Ранее были указаны входные данные задачи, далее они описаны математически и рассмотрены более детально.

Из предоставленных таблиц с информацией о транспортных средствах и маршрутах получены следующие данные:

- n – Суммарное количество маршрутов, по которым необходимо будет распределить транспортные ресурсы.
- m – Суммарное количество различных типов транспортных средств, доступных для развертывания на маршрутах.
- a_t – Количество доступных транспортных средств типа t.
- r_j – Протяженность j-го маршрута в километрах.
- $c_{t_{km}}$ – Расходы на использование типа транспорта t на 1 километр пути.
- c_{t_d} – Расходы на использование типа транспорта t в течение одного дня.
- g_r^k – k-е требование r-го маршрута.
- d_t^k – k-я характеристика типа транспорта t, удовлетворяющая соответствующие требования g^k.

Обозначим суммарное количество характеристик транспортных средств и соответствующих требований маршрутов как z.

Искомые переменные $t_k^i -$ число транспортных средств k-го типа на i-м маршруте (Всего $n * m$ переменных).

Опишем имеющиеся ограничения:

1. $\sum_{i=1}^{n} t_k^i \leq a_k,\ k = \overline{1,m}$ – ограничения на количество единиц транспорта каждого типа.
2. $\sum_{k=1}^{m} t_k^i * d_k^p \geq g_i^p, i = \overline{1,n}, p = \overline{1,z}$ – ограничения, связанные с удовлетворением всех потребностей p на каждом i-м маршруте.

Опишем целевую функцию нашей задачи:

- $P = \sum_{k=1}^{m} \sum_{i=1}^{n} t_k^i * c_k^i \rightarrow min,$

  где

  $c_k^i = \sum_{k=1}^{m} \sum_{i=1}^{n} round\left(c_{d_k} + r_i * c_{km_k}\right)$ – цена использования k-го транспортного средства на i-м маршруте,

  $round -$ функция округления.

Потребность в округлении коэффициентов целевой функции будет объяснена далее, пока что отметим, что подобное округление, за редким исключением, не нарушит оптимальности найденного решения.

Распишем итоговый вид нашей задачи:

$$P = \sum_{k=1}^{m} \sum_{i=1}^{n} t_k^i * c_k^i \rightarrow min,$$

при ограничениях:

$$\sum_{i=1}^{n} t_k^i \leq a_k, \qquad k = \overline{1,m}$$

$$\sum_{k=1}^{m} t_k^i * d_k^p \geq g_i^p, i = \overline{1,n};\ p = \overline{1,z}$$

$$t_k^i - \text{целые.}$$

Далее приведем задачу к стандартному виду, добавив новые переменные и перейдя к поиску максимума функции:

$$P = \sum_{k=1}^{m} \sum_{i=1}^{n} t_k^i * -c_k^i \rightarrow max, \tag{1}$$

при ограничениях:

$$x_k = a_k - \sum_{j=1}^{n} t_k^j, k = \overline{1,m} \tag{2}$$

$$x_{m+ip} = -g_i^p + \sum_{k=1}^{m} t_k^i * d_k^p, i = \overline{1,n}; p = \overline{1,z} \tag{3}$$

$$t_k^i, x_j - \text{целые } \forall i,k,j. \tag{4}$$

В итоге пришли к задаче с числом переменных, равным:

$$m * n + m + nz,$$

и числом уравнений равным:

$$m + nz.$$

Далее перейдем к описанию реализованного алгоритма и его приложению к рассматриваемой задаче (1) – (4).

2.3. Первый алгоритм Гомори

Выше была описана математическая постановка (1) – (4) решаемой задачи об оптимальном распределении транспортных ресурсов. Задача была приведена к стандартному виду.

Для реализации был выбран первый алгоритм Ральфа Гомори, описанный им в статье [4].

Данный алгоритм работает на основе симплекс-метода, потому запишем данные нашей задачи в виде симплекс-таблицы, после чего опишем дальнейшие преобразования.

Симплекс-таблица, построенная по данным задачи, будет иметь следующий вид:

	B	$-t_1^1$	$-t_2^1$	…	$-t_m^1$	$-t_1^2$	…	$-t_m^2$	…	$-t_m^n$
P	0	c_1^1	c_2^1		c_m^1	c_1^2		c_m^2		c_m^n
t_1^1	0	-1	0		0	0		0		0
t_2^1	0	0	-1		0	0		0		0
…										
t_m^1	0	0	0		-1	0		0		0
t_2^2	0	0	0		0	-1		0		0
…										
t_m^2	0	0	0		0	0		-1		0
…										
t_m^n	0	0	0		0	0		0		-1
x_1	a_1	1	0		0	1		0		0
x_2	a_2	0	1		0	0		0		0
…										
x_m	a_m	0	0		1	0		1		1

x_{m+1}	$-g_1^1$	$-d_1^1$	$-d_2^1$		$-d_m^1$	0		0		0
x_{m+2}	$-g_2^1$	0	0		0	$-d_1^1$		$-d_m^1$		0
…										
x_{m+nz}	$-g_n^z$	0	0		0	0		0		$-d_m^z$

Таблица 1. Симплекс-таблица

Отметим, что изображенная симплекс-таблица является лексикографически нормальной (далее l-нормальной), так как любой столбец таблицы R_i является лексикографически положительным, то есть:

$$R_i \succ 0, \forall i$$

Столбец B является расширенным лексикографически положительным псевдопланом (далее l-псевдоплан) в силу l-нормальности таблицы.

Далее, для удобства перепишем все t_k^i в виде x_i, нумеруя их в порядке появления в симплекс-таблице (от 1 до n*m), остальные индексы увеличим на соответствующее значение.

Для начала решим нашу задачу без условия целочисленности при помощи лексикографической модификации метода последовательного уточнения оценок (лексикографического двойственного симплекс-метода, далее обозначается как l-метод).

Опишем общую, r-ю итерацию метода:

1. Имеется l-псевдоплан X^r и соответствующие ему T_r (соответствующая симплекс-таблица), N_r (множество индексов небазисных переменных), B_r (множество индексов базисных переменных).
2. Производится проверка, является ли таблица T_r допустимой (т.е. выполнено ли условие $x_{i0}^r \geq 0, \forall\, i > 0$). Если является, то X^r – l-оптимальный план. Если нет, то выбирается выводимая из базиса переменная x_k, это делается по следующему правилу:
$$k = \min\{i \mid i = \overline{1, m*n+m+nz};\ x_{i0} < 0\}$$
3. Затем ищем переменную x_i, вводимую в базис по правилу:

$$\frac{R_l}{|x_{kl}|} = lex\min\left\{\frac{R_j}{|x_{kj}|}\middle| j \in N_r;\ x_{kj} < 0\right\}$$

4. Если среди чисел $x_{kj}, (j \in N_r)$ нет отрицательных, то задача неразрешима. Если такие числа есть, то производим преобразование симплекс-таблицы по следующим формулам:

$$R_k^* = -\frac{R_l}{x_{kl}},$$

$$R_j^* = R_j - \frac{x_{kj}}{x_{kl}} R_l \text{ для всех } j \in (N \setminus \{l\}) \cup \{0\},$$

Получаем новый l-псевдоплан X^{r+1} и T_{r+1}, B_{r+1}, N_{r+1}. Тут

$$B_{r+1} = (B_r \cup \{l\}) \setminus \{k\},$$

$$N_{r+1} = (N_r \cup \{k\}) \setminus \{l\}.$$

После решения задачи был получен l-оптимальный план задачи (обозначим его $X(\overline{L_r, C})$), если он целочисленный, то он также является оптимальным планом исходной задачи.

Если найденный план не оптимален, то необходимо перейти к большой итерации алгоритма Гомори, опишем ее.

Рассмотрим r-ю большую итерацию:

1. Пусть план $X(\overline{L_r, C})$, полученный после одной из больших итераций или в результате первоначального применения l-метода к исходной задаче, не удовлетворяет условию целочисленности. Отметим, что целевая функция и переменные уже выражены через небазисные переменные в симплекс-таблице.
2. Выберем наименьшую по номеру строку, которой соответствует нецелочисленная компонента

$$k = \min\{i | i \in \{0, 1, \dots, n\};\ x_{i0}^r - \text{не целое}\}$$

и построим соответствующее правильное отсечение

$$x_{m*n+m+nz+r+1} = -\{x_{k0}^r\} + \sum_{j \in N_r} (-\{x_{kj}^r\}) * (-x_j)),$$

$$x_{m*n+m+nz+r+1} \geq 0$$

$$x_{m*n+m+nz+r+1} - \text{целое},$$

здесь

$$\{x\} - \text{взятие дробной части числа}$$

$$\{x\} = x - [x].$$

3. Припишем соответствующую строку к таблице T_r снизу. Получим недопустимую по последней строке l-нормальную таблицу, к которой применим вышеописанный l-метод и в случае нецелочисленности найденного решения вернемся к пункту 1.

Отметим, что после выведения из базиса добавленной на большой итерации алгоритма переменной, соответствующую строку можно вычеркнуть из симплекс-таблицы, а после введения в базис переменной $x_l (l \geq m * n + m + nz)$, соответствующая строка не восстанавливается.

Если в процессе решения будет получена симплекс-таблица, которой будет соответствовать неразрешимая задача линейного программирования, то и исходная задача неразрешима.

Стоит отметить, что указанное замечание о вычеркивании строки с добавленным ограничением играет очень важную роль, ограничивая размеры рассматриваемой на каждом этапе симплекс-таблицы. В приложении к решаемой задаче об оптимальном распределении транспортных ресурсов, размеры симплекс-таблицы не будут превышать $(m * n)$ по столбцам (то есть количество столбцов постоянно), и $(m * n + m + n * z + 1)$ по строкам. В реализуемых системах указанное замечание часто забывается, что приводит к очень существенному увеличению размеров таблицы и, как следствие, к замедлению работы алгоритма.

Доказательство конечности реализованного алгоритма приведено в [6]. Отметим условия, при которых конечность доказана:

1) Гарантирована целочисленность целевой функции (именно поэтому коэффициенты были округлены ранее) и строка, соответствующая ей

учитывается при выборе строки для построения правильного отсечения;

2) По крайней мере одно из следующих утверждений верно:
    a. Целевая функция ограничена снизу на L_0 (что является верным в рассматриваемой задаче, оценки снизу для максимизируемой функции можно получить, назначив все доступные транспортные средства на самые “дорогие” позиции, то есть на те, у которых самые большие коэффициенты в целевой функции);
    b. Задача $(L_0^{\text{ц}}, C)$ имеет хотя бы один план X' (что не гарантировано в рассматриваемой задаче, как и в большинстве прикладных задач).

Таким образом конечность первого алгоритма Гомори для рассматриваемого класса задач можно считать доказанной.

Далее будет проведен анализ работы реализованного алгоритма при различных размерах входных данных.

3. Анализ работы реализованного алгоритма

Для анализа работы реализованного алгоритма генерировались наборы по 2000 задач различных размеров, вычислялось среднее время решения задачи (или получения вывода о ее неразрешимости). Кроме того, измерялась доля задач, решенных менее чем за 1000 больших итераций, так как при определенных данных алгоритм может не найти оптимального решения и за несколько тысяч итераций.

Отметим отдельно одну из проблем реализации алгоритмов решения задач ЦЛП. Эта проблема заключается в округлении чисел и неточности компьютерных вычислений, из-за чего целое в “человеческом” понимании число может быть ошибочно принято за дробное, и алгоритм продолжит работу.

При реализации описанная проблема решалась использованием модуля «Fraction» языка Python, реализующего вычисления в дробях. Точность вычисления была ограничена 10 знаками после запятой (максимальное значение знаменателя дроби - 1e10).

Для анализа работы алгоритма генерировались задачи с разным количеством переменных и ограничений, ниже приведена таблица со следующими данными: количество транспортных маршрутов в задаче, количество различных типов транспортных средств, количество различных характеристик маршрутов и единиц транспорта, суммарное количество переменных в задаче, среднее время решения задачи, количество сгенерированных задач, а также процент задач, решенных менее чем за 1000 больших итераций.

Таблица 1. Результаты тестов

Транспорт	2	2	2	3	2	3	3	4	4	5	5
Маршруты	2	2	3	2	3	3	3	3	4	4	5
Характеристики	1	2	1	2	2	1	2	2	2	2	2
Количество переменных	8	10	11	13	14	15	18	22	28	33	40
Среднее время(сек.)	0.037	0.1	0.22	0.26	0.53	0.57	1.43	2.31	4.61	7.57	8.33
Решенных Задач (%)	98%	95%	91%	91%	84.7%	87.6%	74%	69%	43%	58%	45%

Как видно из приведенной таблицы, увеличение количества переменных приводит к существенному росту времени решения задач. Это связано с тем, что с увеличением количества ограничений и переменных, а в рассматриваемой в данной работе число ограничений напрямую связано с числом переменных, растет и количество больших итераций Гомори, необходимых для получения оптимального плана задачи.

Важно отметить, что среднее время решения задач, указанное в таблице выше учитывает лишь решенные задачи, таким образом, реальное время решения задач соответствующих размеров может быть существенно больше, и шанс этого тем больше, чем меньше процент задач, которые удалось решить за количество итераций, меньшее установленного порога. Для последних 2-х наборов испытаний порог был повышен до 1500 больших итераций Гомори, для остальных он был равен 1000 итераций.

Кроме указанных выше экспериментов были проведены дополнительные, целью которых было определить порог количества переменных, при которых алгоритм отработает за число итераций, меньшее 3000. Решение удалось найти для 79 переменных, вычисления заняли примерно 5 минут. Заметим, что это не означает, что задачи большего размера нельзя решить данным алгоритмом.

4. Обзор реализованной системы

В данном разделе приведен краткий обзор разработанной системы, а также доступный в ней функционал.

Начнем с окна транспортных средств, которое видит пользователь при запуске системы. Его можно наблюдать на Рисунке ниже.

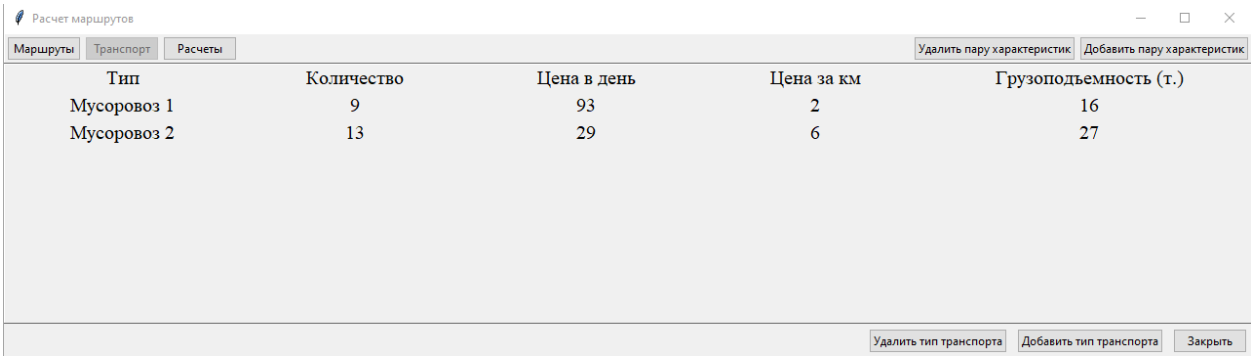

Рисунок 2. Окно транспортных средств

В данном окне пользователь может наблюдать список доступных к распределению транспортных средств, добавлять новые типы транспорта, удалять уже существующие, а также добавлять новые пары характеристик.

В нижней панели можно увидеть кнопки добавления нового типа транспорта, а также удаления уже существующего, а также кнопку закрытия программы. Важно отметить, что все введенные пользователем данные, а также проведенные расчеты сохраняются после закрытия программы и будут доступны во время следующего запуска приложения.

В верхней панели окна можно увидеть кнопки перехода между страницами маршрутов, транспортных средств и расчетов, все они выполнены в виде таблиц и будут продемонстрированы далее. Также в верхней панели присутствуют кнопки удаления и добавления пар характеристик.

Было принято решение добавлять характеристики парами следующего вида: «Требование маршрута – Характеристика транспорта, удовлетворяющая

связанное требование». Это сделано для того, чтобы логически связать характеристики для последующего решения задачи.

Перейдем к окну со списком маршрутов, во многом схожему с окном транспорта. Данное окно можно увидеть на рисунке ниже.

Рисунок 3. Окно маршрутов

Указанное окно обладает тем же функционалом, что и окно транспортных средств.

На следующем рисунке можно наблюдать последнее окно – окно расчетов, ради которого и создавалось приложение.

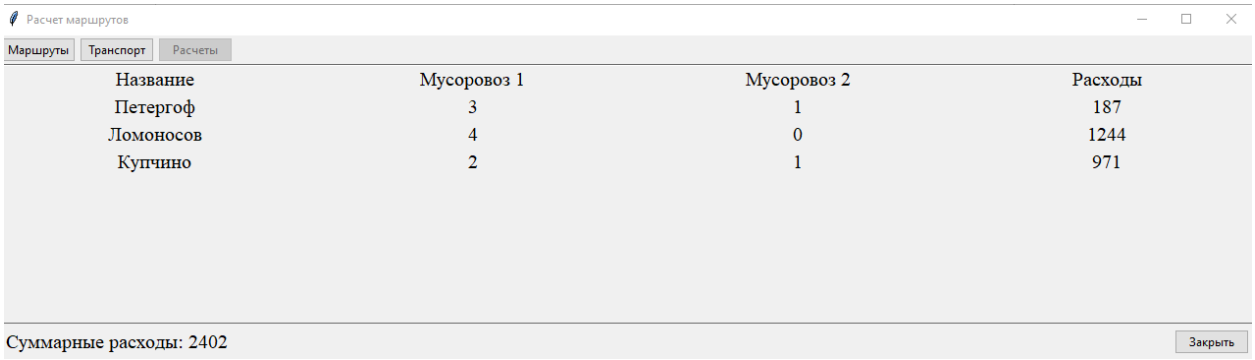

Рисунок 4. Окно расчетов

В данном окне пользователь может наблюдать план оптимального распределения транспортных средств по маршрутам, для каждого маршрута указаны минимальные расходы на удовлетворение всех его потребностей, а в нижнем левом углу можно наблюдать суммарные расходы для всех

маршрутов, они указаны в целых числах, из-за чего возможны небольшие отклонения от введенных пользователем сумм. Как отмечалось ранее, все расходы округляются до целых чисел в силу требований используемого алгоритма.

Перейдем к выводам по проделанной работе.

Выводы

1. Изучив существующие методы решения задач целочисленного программирования, можно подобрать ряд алгоритмов, подходящий для каждого конкретного класса задач.

2. Проведенные исследования показали, что первый алгоритм Гомори работает достаточно хорошо для задач поиска оптимального распределения транспортных ресурсов по маршрутам, рассматриваемых в данной работе. Результаты аналогичны полученным в исследованиях [7] – [9], а в некоторых случаях и превосходят их, в качестве критериев сравнения использовалось среднее количество итераций, необходимых для поиска решения, а также максимальный размер задачи, которую удалось решить.

3. Реализованная система показала себя хорошо для решения задач небольшого размера, что уже способно облегчить работу, выполняемую сотрудниками организаций, так как решение пусть и небольших NP-трудных задач способно существенно повысить эффективность производств или иных структур. Решение же задач большего размера является актуальной проблемой, из-за большого количества операций, выполняемого недопустимо долго при существующих мощностях, а также ошибок округления, которые также накапливаются с ростом количества итераций.

4. Отметим, что существует множество модификаций алгоритмов, рассмотренных в данной задаче, но согласно исследованиям, приведенным в книге [6], все они показывают сравнимые результаты и не приводят к их улучшению.

5. Интересен факт того, что реализованный алгоритм работает для рассматриваемого класса задач существенно эффективнее, чем в общем случае, описанном в [8].

Заключение

В процессе выполнения выпускной квалификационной работы были решены следующие задачи:

1. Изучена литература по теме исследования. Выбраны инструменты для математической формализации поставленной задачи - методы ЦЛП.
2. Построена математическая модель, формализующая задачу оптимального распределения транспортных ресурсов предприятия.
3. Проведен сравнительный анализ большого количества алгоритмов для решения задач ЦЛП. Выбран алгоритм для дальнейшей программной реализации.
4. Реализована система, решающая поставленную задачу для небольших предприятий. Разработан простой пользовательский интерфейс, уже позволяющий использовать эту систему в практических целях.
5. Проведен анализ работы системы, её оптимизация.

Список использованных источников

[1] О подходе к управлению сбором бытовых отходов с помощью гибридной интеллектуальной системы проекта “умный город” / Долинина О.Н., Печенкин В.В. – 2017.

[2] Исследование операций / Костюкова О.И. – 2003г.

[3] Методы дискретной оптимизации / Тюхтина А.А. – 2014г.

[4] Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem / Ralph E. Gomory – 2010г.

[5] Решение двух классов дискретных задач исследования операций / Шалбузов К.Д. – 2015г.

[6] Дискретное программирование / Корбут А.А., Финкельштейн Ю.Ю. Москва, издательство «Наука» – 1969г.

[7] Fixed-cost transportation problems / Balinski M.L., NavalRes. Log. Quart. – 1965г.

[8] Faces of an integer polyhedron / Gomory R.E., Proc. Nac. Acad. U.S.A. – 1967г.

[9] Industrial scheduling / Muth J.F., Thompson G.L., Englewood Cliffs, New Jersey, Prentice Hall – 1963г.