УДК 330.45
JEL C61

АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ СТУДЕНТОВ ПО УЧЕБНЫМ ГРУППАМ

Глазунова Екатерина Валерьевна, аналитик, ассистент

Санкт-Петербургский государственный экономический университет, наб. канала Грибоедова, 30-32, литер А, Санкт-Петербург, Россия, 191023; e-mail: katarina.glazunova97@inbox.ru

Предмет: методы формирования академических групп после процесса распределения студентов по учебным профилям с целью максимизации одного из четырех критериев оптимизации. *Цель*: разработка алгоритмов распределения студентов по учебным группам профиля с точки зрения четырех критериев оптимизации: максимизация схожести суммарного и среднего балла в учебных группах одного профиля, максимизация схожести успеваемости студентов внутри каждой учебной группы и максимизация схожести распределения по группам с предыдущим. *Дизайн исследования*: задачи формулируются как модификации задач о разбиении множества чисел, для оценки схожести успеваемости студентов внутри учебных групп используется индекс Джини, решается задача о разбиении графа на квази-клики с максимальной плотностью. *Результаты*: для каждой задачи сформулирована задача смешанного программирования, а также жадные алгоритмы. Математические модели и алгоритмы апробированы на данных СПбГЭУ при распределении студентов направления «Экономика».

Ключевые слова: задача о разбиении множества чисел, квази-клика, индекс Джини, смешанное программирование, жадный алгоритм.

DOI: 10.17308/meps/2078-9017/2025/9/24-39

Введение

Студенты, поступая на определенное направление подготовки, например экономика, после первого или второго года назначаются на учебные профили, например «финансы и кредит» или «экономика предприятий и организаций». Студенты определяют свои предпочтения по всему множеству учебных профилей, далее с учетом этих предпочтений, а также успеваемости студенты распределяются на учебные профили. В Санкт-Петербургском государственном экономическом университете этот процесс происходит с помощью решения задачи целочисленного программирования [1].

После того как студенты распределены по учебным профилям, необходимо сформировать учебные группы студентов, в составе которых они будут обучаться. Предлагается четыре критерия оптимизации, согласно которым можно формировать учебные группы:

1. максимизация схожести суммарного балла в различных группах одного профиля;

2. максимизация схожести среднего балла в различных группах одного профиля;

3. максимизация схожести успеваемости студентов внутри каждой учебной группы, то есть в каждой учебной группе должны находиться студены примерно одной успеваемости;

4. максимизация схожести распределения по группам с предыдущим, то есть если пара студентов училась в одной группе ранее, то желательно, чтобы эта пара студентов училась в одной группе после распределения на профили.

Первая задача является расширением задачи о разбиении множества чисел (англ. multiway number partitioning), где множество чисел необходимо разбить на несколько подмножеств так, чтобы сумма чисел в подмножествах была бы наиболее близка друг к другу. Впервые данная задача была предложена в 1969 году для оптимизации назначения задач идентичным машинам [8]. Данная задача является NP-трудной [6], поэтому решение таких задач точными методами на полномасштабных реальных данных невозможно за адекватное время. Поэтому в данной работе рассматривается не только формулировка задачи смешанного программирования [2], но и жадный алгоритм [4].

Вторая задача является модификацией первой задачи, где необходимо множество чисел разбить на несколько подмножеств так, чтобы среднее значение чисел в подмножествах было бы наиболее близко друг к другу.

Для решения третьей задачи предлагается использовать индекс Джини [7] для оценки неравенства успеваемости студентов.

Рассмотрим четвертую задачу. Построим граф, где вершины соответствуют множеству студентов, а наличие ребра означает, что студенты учились ранее в одной группе. Очевидно, такой граф состоит из одной или нескольких компонент связности, где каждая компонента связности является кликой, то есть полным подграфом. Идеально разбиение графа на клики, где каждая клика являлась бы учебной группой. В работах [3, 5] рассматривается задача о разбиении графа на клики. Однако зачастую такое разбиение на клики невозможно, поэтому вторая задача – задача о разбиении графа на квази-клики с максимальной плотностью. Необходимо разделить граф на заранее определенное число квази-клик или γ-клик [11], то есть подграфов, плотность которых не меньше γ, где плотность графа определяется по формуле (1):

$$d(G) = 2 \times |E|/(|V| \times (|V| - 1)) \qquad (1)$$

Для схожей задачи – поиска разбиения графа на минимальное число квази-клик, плотность которых не меньше заданной, применяются точные алгоритмы [12], но поскольку задача NP-трудная [12], то применяются также рандомизированные жадные алгоритмы [11], а также алгоритмы локального поиска [13].

В данной работе предлагаются задачи смешанного программирования для решения задачи о разбиении множества студентов профиля на учебные группы с точки зрения четырех критериев оптимизации, описанных выше, также предлагаются жадные эвристики, позволяющие получить допустимое решение данных задач за разумное время.

Методы и результаты исследования

1. Максимизация схожести суммарного балла в учебных группах одного профиля

Формальная постановка задачи о формировании множества учебных групп студентов при максимизации схожести суммарного балла в группах имеет следующий вид.

Дано некоторое множество студентов S, подлежащее разбиению на группы. Каждый студент $s \in S$ характеризуется средним баллом своей учебной успеваемости $score_s$. Дано количество учебных групп K, которые необходимо сформировать, а также минимальный $\underline{q}$ и максимальный $\overline{q}$ размеры групп.

Сформулируем задачу смешанного программирования, как модификацию задачи о разбиении множества чисел.

Введем следующие переменные в задаче:

- $x_{sk}, \forall s \in S, k \in \{1, \dots, K\}$ – бинарная переменная, которая принимает значение 1, если студент s назначен в учебную группу под номером k;
- γ – вещественная переменная, означающая минимальный суммарный балл среди всех учебных групп.

Тогда, чтобы максимизировать схожесть суммарного балла в учебных группах, необходимо максимизировать значение γ, целевая функция имеет вид:

$$\gamma \to \max. \tag{2}$$

Возможны и другие подходы к формированию целевой функции: минимизировать максимальный суммарный балл среди всех групп или минимизировать разницу между максимальным суммарным баллом среди всех групп и минимальным. Если происходит разбиение на две группы, то эти целевые функции эквивалентны, иначе они дают различные результаты [10].

Также необходимо учесть следующие ограничения.

Каждый студент должен быть распределен в ровно одну учебную группу:

$$\sum_{k \in \{1,\dots,K\}} x_{sk} = 1 \qquad \forall s \in S \tag{3}$$

Размер каждой учебной группы должен быть в заданных пределах:

$$\underline{q} \leq \sum_{s \in S} x_{sk} \leq \overline{q} \qquad \forall k \in K \tag{4}$$

Значение переменной γ меньше или равно суммарного балла студентов, назначенных в каждую группу:

$$\gamma \leq \sum_{s \in S} score_s \cdot x_{sk} \qquad \forall k \in K \tag{5}$$

Естественные ограничения:

$$\begin{aligned} x_{sk} \in \{0,1\} \qquad & \forall s \in S, \forall k \in \{1, \dots, K\} \\ \gamma \geq 0 \end{aligned} \tag{6}$$

Сформулируем жадный алгоритм решения данной задачи. Идея алгоритма базируется на жадном алгоритме для решения задачи о разбиении множества чисел [9], но с учетом дополнительных ограничений на размер группы.

Сформируем K пустых множеств, в которых будем хранить студентов, назначаемых в каждую учебную группу. Отсортируем множество студентов по убыванию их суммарного балла. Далее будем повторять следующий процесс для каждого студента: выберем среди множества формируемых учебных групп те, назначение в которые текущего студента не приведет к потери допустимости решения с точки зрения размера групп, то есть, если назначить студента в текущую группу, то оставшихся студентов будет достаточно, чтобы заполнить остальные группы до минимального размера; далее среди выбранных учебных групп выберем ту группу, в которой минимальный суммарный балл студентов, назначим в эту группу текущего студента.

Если же требуется сформировать учебные группы, которые бы отличались минимально друг от друга не по суммарному баллу студентов в группе, а по среднему баллу в группе, то задачу следует переформулировать, добавив в нее дополнительные переменный и ограничения, что является эвристическим показателем того, что вычислительно задача становится более сложной. Однако с учетом ограничения на размер группы и того, что границы размеров групп неширокие, можно отметить, что максимизация похожести суммарного балла является эвристикой для оптимизации критерия максимизация похожести среднего балла в группах.

2. Максимизация схожести среднего балла в учебных группах одного профиля

Сформулируем задачу смешанного программирования.

Введем следующие переменные в задаче:

- $x_{sk}, \forall s \in S, k \in \{1, \dots, K\}$ – бинарная переменная, которая принимает значение 1, если студент s назначен в учебную группу под номером k;
- z_{kl}, $\forall k \in \{1, \dots, K\}$, $\forall l \in \{\underline{q}, \dots, \overline{q}\}$ – бинарная переменная, принима-

ющая значение 1, если k-ая группа имеет размер l, и 0 в обратном случае;

- γ – вещественная переменная, означающая минимальный средний балл среди всех учебных групп.

Аналогично предыдущей задаче целевая функция примет вид:

$$\gamma \to \max \tag{7}$$

При следующих ограничениях.

Каждый студент должен быть распределен в ровно одну учебную группу:

$$\sum_{k\in\{1,\dots,K\}} x_{sk} = 1 \qquad \forall s \in S \tag{8}$$

Определение размера учебной группы – количество студентов, назначенных в группу равно размеру группы:

$$\sum_{v\in V} x_{vk} = \sum_{l\in\{\underline{q},\dots,\overline{q}\}} l \cdot z_{kl} \qquad \forall k \in \{1,\dots,K\} \tag{9}$$

Каждой учебной группе соответствует ровно один размер:

$$\sum_{l\in\{\underline{q},\dots,\overline{q}\}} z_{kl} = 1 \qquad \forall k \in \{1,\dots,K\} \tag{10}$$

Значение переменной γ меньше или равно среднему баллу студентов, назначенных в каждую группу:

$$\gamma \le \left(\sum_{s\in S} score_s \cdot x_{sk}\right)/l + M \cdot (1 - z_{kl}) \qquad \begin{array}{c} \forall k \in K, \\ \forall l \in \{\underline{q},\dots,\overline{q}\} \end{array} \tag{11}$$

где M – достаточно большое число.

Естественные ограничения:

$$\begin{gathered} x_{sk} \in \{0,1\} \quad \forall s \in S, \forall k \in \{1,\dots,K\} \\ z_{kl} \in \{0,1\} \quad \forall k \in K, \forall l \in \{\underline{q},\dots,\overline{q}\} \\ \gamma \ge 0 \end{gathered} \tag{12}$$

Жадный алгоритм для решения задачи о максимизации схожести среднего балла имеет следующий вид. Сформируем K пустых множеств, где будут храниться студенты, назначенные в каждую группу. Сформируем $N \times K$ временных решений, где N – число студентов, то есть каждого студента назначим в каждую учебную группу, исключим из рассмотрения те решения, которые приводят к недопустимому решению в дальнейшем (аналогично первому алгоритму), далее среди всех временных решений выберем в качестве текущего решения то, что имеет наилучшее значение целевой функции (7). Исключим из множества студентов назначенного на данном шаге студента. Будем повторять процесс до тех пор, пока множество студентов непусто.

Альтернативным критерием оптимизации может выступать максимизация схожести успеваемости студентов внутри каждой учебной группы. То есть студенты одной учебной группы должны иметь наиболее похожую успеваемость.

3. Максимизация схожести успеваемости студентов внутри учебных групп

В экономике для измерения неравенства используется индекс Джини [7], рассчитываемый по следующей формуле:

$$GINI = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2 \cdot n^2 \cdot \overline{x}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2 \cdot n \cdot \sum_{i=1}^{n} x_i} \tag{13}$$

где x_i – значение дохода i-го экономического агента; n – число агентов.

В данной задаче вместо измерения экономического неравенства индекс Джини будет использоваться для измерения неравенства успеваемости студентов внутри каждой учебной группы. Исходные данные задачи аналогичны предыдущей задаче.

Задача смешанного программирования примет следующий вид.

Введем переменные задачи:

- $x_{sk}, \forall s \in S, k \in \{1, \dots, K\}$ – бинарная переменная, которая принимает значение 1, если студент s назначен в учебную группу под номером k;
- $z_{kl}, \forall k \in \{1, \dots, K\}, \forall l \in \{\underline{q}, \dots, \overline{q}\}$ – бинарная переменная, принимающая значение 1, если k-ая группа имеет размер l, и 0 в обратном случае;
- $o_{ksc}, \forall k \in \{1, \dots, K\}, \forall s, c \in S: s \neq c$ – бинарная переменная, принимающая значение 1, если пара студентов s, c назначены в одну группу k, и 0 в обратном случае;
- $d_{ksc}, \forall k \in \{1, \dots, K\}, \forall s, c \in S: s \neq c$ – вещественная переменная, принимающая значение абсолютного отклонения успеваемости студента s от успеваемости студента c, деленной на среднюю успеваемость группы, если они оба назначены в эту группу k;
- $y_{sk}, \forall s \in S, k \in \{1, \dots, K\}$ – вещественная переменная, является фиктивной, необходима для линеаризации ограничений;
- γ – вещественная переменная, означающая максимальный индекс Джини среди всех учебных групп.

Целевая функция примет вид:

$$\gamma \to \min \tag{14}$$

При следующих ограничениях.

Каждый студент назначен ровно в одну группу:

$$\sum_{k\in\{1,\ldots,K\}} x_{sk} = 1 \qquad \forall s \in S \qquad (15)$$

Связь размера группы и переменных типа z:

$$\sum_{s\in S} x_{sk} = \sum_{l\in\{\underline{q},\ldots,\overline{q}\}} l \cdot z_{kl} \qquad \forall k \in \{1,\ldots,K\} \qquad (16)$$

Каждая учебная группа имеет ровно один размер:

$$\sum_{l\in\{\underline{q},\ldots,\overline{q}\}} z_{kl} = 1 \qquad \forall k \in \{1,\ldots,K\} \qquad (17)$$

Связь переменных типа o и x – пара студентов s, c учатся в одной группе k, если они оба назначены в эту группу:

$$o_{ksc} \geq x_{sk} + x_{ck} - 1 \qquad \begin{matrix} \forall k \in \{1,\ldots,K\} \\ \forall s, c \in S : s \neq c \end{matrix} \qquad (18)$$

Определение абсолютного отклонения успеваемости пары студентов, деленной на среднюю успеваемость группы:

$$\begin{matrix} d_{ksc} \geq score_s - score_c - M \cdot (1 - o_{ksc}) & \forall k \in \{1,\ldots,K\} \\ d_{ksc} \geq score_c - score_s - M \cdot (1 - o_{ksc}) & \forall s, c \in S : s \neq c \end{matrix} \qquad (19)$$

где M – достаточно большое число.

Определение максимального значения индекса Джини среди всех групп:

$$\gamma \geq \frac{\sum_{s\in S} \sum_{c\in S: s\neq c} d_{ksc}}{2 \cdot l \cdot \sum_{s\in S} score_s x_{sk}} - M \cdot (1 - z_{kl}) \qquad \begin{matrix} \forall k \in \{1,\ldots,K\} \\ \forall l \in \{\underline{q},\ldots,\overline{q}\} \end{matrix} \qquad (20)$$

Такое ограничение является нелинейным, поэтому умножим обе части на $2 \cdot l \cdot \sum_{s\in S} score_s \cdot x_{sk}$:

$$\gamma \cdot 2 \cdot l \cdot \sum_{s\in S} score_s \cdot x_{sk} \geq\geq \sum_{s\in S} \sum_{\substack{c\in S: \\ s\neq c}} d_{ksc} - M \cdot (1 - z_{kl}) \qquad \begin{matrix} \forall k \in \{1,\ldots,K\} \\ \forall l \in \{\underline{q},\ldots,\overline{q}\} \end{matrix} \qquad (21)$$

Такое ограничение все еще является нелинейным, поэтому введем переменную $y_{sk}, \forall s \in S, \forall k \in \{1,\ldots,K\}$ для линеаризации перемножения γ и x_{sk}, тогда получим следующие ограничения:

$$2 \cdot l \cdot \sum_{s\in S} score_s \cdot y_{sk} \geq \sum_{s\in S} \sum_{\substack{c\in S: \\ s\neq c}} d_{ksc} - M \cdot (1 - z_{kl}) \qquad \begin{matrix} \forall k \in \{1,\ldots,K\} \\ \forall l \in \{\underline{q},\ldots,\overline{q}\} \end{matrix} \qquad (22)$$

$$\begin{matrix} y_{sk} \geq \gamma - M \cdot (1 - x_{sk}) & \\ y_{sk} \leq \gamma + M \cdot (1 - x_{sk}) & \forall s \in S \\ y_{sk} \geq -M \cdot x_{sk} & \forall k \in \{1,\ldots,K\} \\ y_{sk} \leq M \cdot x_{sk} & \end{matrix} \qquad (23)$$

Естественные ограничения:

$$\begin{aligned}
&x_{sk} \in \{0,1\} && \forall s \in S, \forall k \in \{1, \dots, K\} \\
&z_{kl} \in \{0,1\} && \forall k \in \{1, \dots, K\}, \forall l \in \{\underline{q}, \dots, \overline{q}\} \\
&o_{ksc} \in \{0,1\} && \forall k \in \{1, \dots, K\}, \forall s, c \in S: s \neq c \\
&y_{sk} \geq 0 && \forall s \in S, \forall k \in \{1, \dots, K\} \\
&d_{ksc} \geq 0 && \forall k \in \{1, \dots, K\}, \forall s, c \in S: s \neq c \\
&\gamma \geq 0 &&
\end{aligned} \tag{24}$$

Жадный алгоритм для решения задачи о максимизации схожести среднего балла внутри одной группы аналогичен жадному алгоритму для предыдущей задачи, но с учетом целевой функции минимизации максимального значения индекса Джини среди всех учебных групп.

4. Максимизация схожести распределения по группам с предыдущим

Сформулируем задачу смешанного программирования для решения задачи поиска разбиения студентов на учебные группы так, чтобы искомое разбиение было максимально похоже на предыдущее. Математическая модель базируется на идее определения размера каждой квази-клики посредством введения соответствующей бинарной переменной, аналогично тому, как это реализовано в модели «Formulation using quasi-clique size decompositions» [11].

Формальная постановка задачи имеет следующий вид. Так же дано множество студентов S, подлежащих разбиению на группы. Пусть студенты множества S ранее учились в N учебных группах, для каждого студента $s \in S$ известен номер его предыдущей учебной группы $g_s \in \{1, \dots, N\}$. Кроме того, так же, как и в предыдущей задаче, определены количество учебных групп K, которые необходимо сформировать, а также минимальный $\underline{q}$ и максимальный $\overline{q}$ размеры групп.

Введем в рассмотрение граф $G = (V, E)$, где множество вершин равняется множеству студентов S $(V = S)$, а $E = \{(u, v) | u \in V, v \in V, g_u = g_v\}$, то есть пара вершин соединена ребром, если соответствующие им студенты ранее учились в одной группе.

Введем в задачу следующие переменные:

- $x_{vk}, \forall v \in V$, $\forall k \in \{1, \dots, K\}$ – бинарная переменная, принимающая значение 1, если студент v назначен в группу k,и 0 в обратном случае;
- z_{kl}, $\forall k \in \{1, \dots, K\}$, $\forall l \in \{\underline{q}, \dots, \overline{q}\}$ – бинарная переменная, принимающая значение 1, если k-ая группа имеет размер l, и 0 в обратном случае;
- $o_{kuv}, \forall k \in \{1, \dots, K\}$, $\forall (u, v) \in E$ – бинарная переменная, принимающая значение 1, если пара студентов, которые учились ранее, назначены в одну группу k, и 0 в обратном случае;

- γ – вещественная переменная, означающая минимальную плотность среди всех формируемых квази-клик.

Целевая функция имеет вид:

$$\gamma \to \max \tag{25}$$

При следующих ограничениях.

Каждый студент распределен ровно в одну группу:

$$\sum_{k\in\{1,\dots,K\}} x_{vk} = 1 \qquad \forall v \in V \tag{26}$$

Связь размера группы и переменных типа z:

$$\sum_{v\in V} x_{vk} = \sum_{l\in\{\underline{q},\dots,\overline{q}\}} l\cdot z_{kl} \qquad \forall k \in \{1,\dots,K\} \tag{27}$$

Каждая учебная группа имеет ровно один размер:

$$\sum_{l\in\{\underline{q},\dots,\overline{q}\}} z_{kl} = 1 \qquad \forall k \in \{1,\dots,K\} \tag{28}$$

Связь переменных типа o и x – пара студентов u, v учатся в одной группе k, если они оба назначены в эту группу:

$$\begin{aligned} o_{kuv} &\le x_{uk} && \forall k \in \{1,\dots,K\} \\ o_{kuv} &\le x_{vk} && (u,v) \in E \end{aligned} \tag{29}$$

Плотность каждой формируемой квази-клики (которым соответствует учебная группа) не меньше γ:

$$\sum_{(u,v)\in E} o_{kuv} \ge \gamma\cdot\frac{l\cdot(l-1)}{2} - M\cdot(1-z_{kl}) \qquad \begin{aligned}&\forall k \in \{1,\dots,K\}\\ &\forall l \in \{\underline{q},\dots,\overline{q}\}\end{aligned} \tag{30}$$

где M – достаточно большое число.

Естественные ограничения:

$$\begin{aligned} &x_{uk} \in \{0,1\} && \forall u \in V, \forall k \in \{1,\dots,K\} \\ &z_{kl} \in \{0,1\} && \forall k \in \{1,\dots,K\}, \forall l \in \{\underline{q},\dots,\overline{q}\} \\ &o_{kuv} \in \{0,1\} && \forall k \in \{1,\dots,K\}, \forall (u,v) \in E \\ &\gamma \ge 0 \end{aligned} \tag{31}$$

Сформулируем жадный алгоритм для решения задачи разбиения графа на K квази-клик, максимизируя минимальную плотность среди всех квази-клик с учетом того, что каждая компонента связности графа – клика.

Сформируем K пустых множества, куда будем назначать вершины созданного графа G, то есть назначать студентов группам. Среди всех клик графа G выберем максимальную клику. Далее для каждого множества, соответствующего учебной группе, определим максимальный размер «подклики», то есть максимальное количество студентов, которое можно назначить в текущую группу от рассматриваемой клики так, чтобы гарантировать допустимость искомого решения с точки зрения размеров групп. Для каждой учебной группы максимальное количество студентов, которое можно назначить в текущую учебную группу, определяется как минимум среди размера рассматриваемой клики, разницей между максимальным размером группы $\overline{q}$ и количеством студентов, уже назначенных в эту группу, а также разницей между оставшимся числом вершин в графе G и суммарным количеством студентов, необходимых для заполнения остальных групп до минимального размера. Так для каждой учебной группы определен размер «подклики» (подграфа заданного размера), теперь временно назначим «подклику» соответствующего размера в каждую учебную группу. Среди всех назначений выберем то, которое обеспечивает максимальную плотность сформированной квази-клики. Удалим из графа G вершины, назначенной группе «подклики».

5. Числовые результаты

Для описанных выше задач были проведены расчеты на данных распределения студентов направления «Экономика» Санкт-Петербургского государственного экономического университета, среди всех сформированных учебных профилей были отобраны те, где необходимо создать более одной учебной группы. В таблице 1 представлено сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести суммарного балла групп. При проведении расчетов для решения задачи смешанного программирования использовался оптимизатор COIN-OR CBC 2.10.3[1], для задач смешанного программирования было введено временное ограничение на поиск решения в 600 с. Весь исходный код реализованных алгоритмов доступен на github[2].

Таблица 1

Сравнение смешанного программирования и жадного алгоритма
для задачи максимизации схожести суммарного балла групп

Набор данных	Смешанное программирование		Жадный алгоритм	
	Значение ЦФ	Время счета	Значение ЦФ	Время счета
2 группы Мин. балл 1711 Сред. балл 2504.24 Макс. балл 2709 Ст. отклонение 166.18	72623	<1 с	72511	<1 с
5 групп Мин. балл 1370 Сред. балл 2311.84 Макс. балл 2704 Ст. отклонение 259.68	75774	600 с	74681.0	<1 с

[1] COIN-OR CBC 2.10.3. Доступно: https://github.com/coin-or/Cbc.
[2] Исходный код проекта. Доступно: https://github.com/katarina74/dividingStudentsIntoGroups.

Окончание табл. 1

Набор данных	Смешанное программирование		Жадный алгоритм	
	Значение ЦФ	Время счета	Значение ЦФ	Время счета
2 группы Мин. балл 1238 Сред. балл 2022.74 Макс. балл 2594 Ст. отклонение 332.47	44500	600 с	44452.74	<1 с
2 группы Мин. балл 1531 Сред. балл 2178.98 Макс. балл 2604 Ст. отклонение 204.18	70816.98	600 с	70051.98	<1 с

В таблице 2 представлено сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести среднего балла групп.

Таблица 2

Сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести среднего балла групп

Набор данных	Смешанное программирование		Жадный алгоритм	
	Значение ЦФ	Время счета	Значение ЦФ	Время счета
2 группы Мин. балл 1711 Сред. балл 2504.24 Макс. балл 2709 Ст. отклонение 166.18	2504.24	600 с	2421.28	<1 с
5 групп Мин. балл 1370 Сред. балл 2311.84 Макс. балл 2704 Ст. отклонение 259.68	2310.64	600 с	2255.51	5.44 с
2 группы Мин. балл 1238 Сред. балл 2022.74 Макс. балл 2594 Ст. отклонение 332.47	2022.73	600 с	1946.26	< 1 с
2 группы Мин. балл 1531 Сред. балл 2178.98 Макс. балл 2604 Ст. отклонение 204.18	2178.97	600 с	2173.18	<1 с

В таблице 3 представлено сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести успеваемости студентов внутри групп. Прочерк означает, что за отведенное время не удалось найти допустимое решение.

Таблица 3

Сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести успеваемости студентов внутри групп

Набор данных	Смешанное программирование		Жадный алгоритм	
	Значение ЦФ	Время счета	Значение ЦФ	Время счета
2 группы Мин. балл 1711 Сред. балл 2504.24 Макс. балл 2709 Ст. отклонение 166.18	-	-	0.033	<1 с
5 групп Мин. балл 1370 Сред. балл 2311.84 Макс. балл 2704 Ст. отклонение 259.68	0.1272	600 с	0.075	8.88 с
2 группы Мин. балл 1238 Сред. балл 2022.74 Макс. балл 2594 Ст. отклонение 332.47	0.102	600 с	0.094	<1с
2 группы Мин. балл 1531 Сред. балл 2178.98 Макс. балл 2604 Ст. отклонение 204.18	0.054	600 с	0.051	< 1 с

В таблице 4 представлено сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести распределения по группам на предыдущее.

Таблица 4

Сравнение смешанного программирования и жадного алгоритма для задачи максимизации схожести распределения по группам на предыдущее

Набор данных	Смешанное программирование		Жадный алгоритм	
	Значение ЦФ	Время счета	Значение ЦФ	Время счета
2 формируемые группы 10 групп в исходном распределении	0.38	86.10	0.38	< 1 с
5 формируемых групп 18 групп в исходном распределении	0.18	600 с	0.33	< 1 с
2 формируемые группы 13 групп в исходном распределении	0.19	2.48 с	0.15	< 1 с
2 формируемые группы 13 групп в исходном распределении	0.17	9.73 с	0.16	< 1 с

Таким образом, подход на основе смешанного программирования лишь в некоторых случаях позволяет найти оптимальное решение за отведенное время. В то время как жадные алгоритмы работают сравнительно быстрее, они не всегда позволяют получить оптимальное решение. Однако они демонстрируют высокую эффективность с точки зрения качества найденных решений.

Заключение

В работе рассматривается задача поиска разделения студентов на учебные группы с точки зрения четырех критериев оптимизации:

1. максимизация схожести суммарного балла различных групп одного профиля;

2. максимизация схожести среднего балла различных групп одного профиля;

3. максимизация схожести успеваемости студентов внутри каждой учебной группы;

4. максимизация схожести разбиения студентов по группам на предыдущее.

Выделяя преимущества и недостатки каждого метода, можно отметить следующее. Найти решение по первому критерию оптимизации на практике проще с точки зрения временных затрат, чем по второму критерию. Кроме того, с учетом того, что на размеры групп накладываются неширокие границы, поиск решения по первому критерию обеспечивает приближение к оптимуму по второму критерию. В то же время поиск решения по второму критерию позволит гарантированно найти оптимальное решение. Данный критерий является наиболее предпочтительным при поиске разбиения студентов на группы, так как является наиболее просто интерпретируемым и понятным. Преподавателям проще готовить материал для обучения: легче определить темп и сложность материала. Кроме того, данный критерий обеспечивает справедливую и конкурентную среду внутри учебных групп. Разбиение на сильные и слабые группы, согласно третьему критерию, позволяет преподавателям давать углубленный материал сильным группам и более простой материал слабым группам, однако разбиение по такому критерию может привести к снижению мотивации в слабых группах и отсутствию возможности слабым студентам учиться у более сильных. Четвертый критерий позволяет минимизировать стресс студентов от изменения сложившихся коллективов и потери социальных связей. Данный критерий стоит использовать, если известно, что в ранее созданных учебных группах были сплоченные коллективы, или если важно сохранить социальные связи для проектной работы студентов, в остальных случаях максимизация данного критерия не гарантирует качество учебного процесса. Таким образом, наиболее легко интерпретируемым и гарантирующим качество учебного процесса является второй критерий, или первый критерий как его приближение.

Было проведено сравнение подходов на основе смешанного программирования и жадных алгоритмов на реальных данных распределения студентов направления «Экономика» по учебным профилям. Подход на основе смешанного программирования зачастую не позволяет найти оптимальное решение за отведенное время, в то время как жадные алгоритмы демонстрируют эффективность как с точки зрения временных затрат, так и качества решения, хотя и зачастую не позволяют найти оптимальное решение.

Список источников

1. Васильев Ю.М., Глазунова Е.В. и Фридман Г.М. Умный университет: справедливое распределение студентов по учебным профилям // *Вестник ВГУ. Серия: Экономика и управление*, 2024, no. 5, с. 16-24.
2. Мину М. *Математическое программирование. Теория и алгоритмы*. Москва, Наука, 1990. 485 с.
3. Brimberg J., Janićijević S., Mladenović N. et al. Solving the clique partitioning problem as a maximally diverse grouping problem // *Optim Lett,* 2017, no. 11, pp. 1123-1135.
4. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. I*ntroduction to Algorithms, third edition.* The MIT Press Cambridge, Massachusetts London, England, 2009. 1312 p.
5. De Amorim, S.G., Barthélemy JP. & Ribeiro C.C. Clustering and clique partitioning: Simulated annealing and tabu search approaches // *Journal of Classification*, 1992, no. 9, pp. 17-41.
6. Garey M.R., Johnson D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company. 1979. 238 p.
7. Gini C. On the Measure of Concentration with Special Reference to Income and Statistics // *Colorado College Publication, General Series*, 1936, no. 208, pp. 73-79.
8. Graham R.L. Bounds on multiprocessing timing anomalies // *SIAM Journal on Applied Mathematics*, 1969, no. 17(2), pp. 416-429.
9. Korf R. Multi-Way Number Partitioning // *IJCAI International Joint Conference on Artificial Intelligence*, 2009, pp. 538-543.
10. Korf R. Objective functions for Multi-Way number partitioning // *Proceedings of the International Symposium on Combinatorial Search*, 2010, no. 1(1), pp. 71-72.
11. Melo R., Ribeiro C., Riveaux M.J. The minimum quasi-clique partitioning problem: Complexity, formulations, and a computational study // *Information Sciences*, 2022, no. 612.
12. Ribeiro C., Riveaux M.J. An exact algorithm for the maximum quasi-clique problem // *International Transactions in Operational Research*, 2019, no. 26.
13. Zhou Q., Zhu T., Wu Q., Jiang Z., Wang W. An efficient iterated local search for the minimum quasi-clique partitioning problem // *Computers & Operations Research*, 2025, no. 179.

ALGORITHMS FOR DIVIDING STUDENTS INTO STUDY GROUPS

Glazunova Ekaterina Valerevna, analyst, assistant

Saint Petersburg State University of Economics, Sadovaya str., 21, St. Petersburg, Russia, 191023; e-mail: katarina.glazunova97@inbox.ru

Importance: After the first or second year of study, students in a specific field, such as «Economics», are assigned to specialized tracks, for example, «Finance and Credit» or «Enterprise and Organizational Economics». Once students are distributed across these tracks, it becomes necessary to form study groups. *Purpose*: To develop algorithms for assigning students to study groups within a track based on two optimization criteria: Maximizing the «similarity» of average academic scores across groups within the same track. Maximizing the «similarity» of group distribution to the previous one. *Research design:* Development of algorithms for distributing students into specialized study groups based on four optimization criteria: maximizing the «similarity» of the total and average scores within groups of the same profile, maximizing the «similarity» of academic performance among students within each group, and maximizing the «similarity» of distribute on compared to previous groupings. *Results*: For each problem, mixed-integer programming models were formulated, along with greedy heuristic algorithms. The mathematical models and algorithms were tested on real-world data from Saint Petersburg State University of Economics when distributing students in the «Economics» program.

Keywords: multiway number partitioning, quasi-clique partition problem, Gini index, mixed-integer programming, greedy algorithm.

References

1. Vasil'ev Ju.M., E.V. Glazunova i G.M. Fridman. «Umnyj universitet: spravedlivoe raspredelenie studentov po uchebnym profiljam» [«Smart University» Project: fair matching of students to academic trajectories]. *Vestnik VGU. Serija: Jekonomika i upravlenie,* 2024, no. 5, pp. 16-24. (In Russ.)

2. Minu M. *Matematicheskoe programmirovanie. Teorija i algoritmy*. [Mathematical programming. Theory and algorithms]. Moscow, Nauka, 1990. 485 p. (In Russ.)

3. Brimberg J., Janićijević S., Mladenović N. et al. Solving the clique partitioning problem as a maximally diverse grouping problem. *Optim Lett*, 2017, no. 11, pp. 1123-1135. (In Eng.)

4. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, third edition. *The MIT Press Cambridge*, Massachusetts London, England, 2009. 1312 p. (In Eng.)

5. De Amorim S.G., Barthélemy JP. & Ribeiro C.C. Clustering and clique partitioning: Simulated annealing and tabu search approaches. *Journal of Classification*, 1992, no. 9, pp. 17-41. (In Eng.)

6. Garey M.R., Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. *W. H. Freeman and Company*, 1979. 238 p. (In Eng.)

7. Gini C. On the Measure of Concentration with Special Reference to Income and Statistics. Colorado College

Publication, General Series, 1936, no. 208, pp. 73-79. (In Eng.)

8. Graham R.L. Bounds on multiprocessing timing anomalies. SIAM *Journal on Applied Mathematics*, 1969, no. 17(2), pp. 416-429. (In Eng.)

9. Korf R. Multi-Way Number Partitioning. *IJCAI International Joint Conference on Artificial Intelligence*, 2009, pp. 538-543. (In Eng.)

10. Korf R. Objective functions for Multi-Way number partitioning. *Proceedings of the International Symposium on Combinatorial Search*, 2010, no. 1(1), pp. 71-72. (In Eng.)

11. Melo R., Ribeiro C., Riveaux M. J. The minimum quasi-clique partitioning problem: Complexity, formulations, and a computational study. *Information Sciences*, 2022, no. 612. (In Eng.)

12. Ribeiro C., Riveaux M.J. An exact algorithm for the maximum quasi-clique problem. *International Transactions in Operational Research,* 2019, no. 26. (In Eng.)

13. Zhou Q., Zhu T., Wu Q., Jiang Z., Wang W. An efficient iterated local search for the minimum quasi-clique partitioning problem. *Computers & Operations Research*, 2025, no. 179. (In Eng.)