Вестник ВГУ. Серия: Экономика и управление. 2025. № 3. С. 3–11.
Proceedings of Voronezh State University. Series: Economics and Management. (3), 3–11.

Вестник Воронежского государственного университета
Серия: Экономика и управление

Математические и инструментальные методы экономики

Научная статья
УДК 519.863
DOI: https://doi.org/10.17308/econ.2025.3/13261
JEL: C61

Алгоритмы серийной диктатуры распределения студентов по учебным профилям в университете

Е. В. Глазунова[1]✉, И. А. Зелин[2]

[1, 2] Санкт-Петербургский государственный экономический университет, ул. Садовая, 21, 191023, Санкт-Петербург, Российская Федерация

Предмет. Алгоритмы серийной диктатуры – механизмы распределения, в котором агенты последовательно, в заранее определенном порядке, выбирают наиболее предпочтительные для себя варианты. Данные алгоритмы исследуются в рамках задачи распределения студентов по учебным профилям.
Цель. Статья посвящена алгоритмам серийной диктатуры поиска распределения студентов по учебным профилям в университете, где студенты в порядке своей успеваемости выбирают наиболее предпочтительные для себя места в учебных профилях среди оставшихся.
Метод. Предлагаются два точных алгоритма, использующие методы динамического программирования, а также жадный алгоритм, являющийся более эффективным в терминах временной сложности.
Результаты. Осуществлена апробация на реальных данных СПбГЭУ в рамках распределения студентов по учебным профилям направления «Экономика», а также выполнены массовые расчеты на синтетических данных. Проведен анализ связи скоррелированности предпочтений учебных профилей по студентам с количеством конфликтов зависти для описанных алгоритмов.
Выводы. Разработанные алгоритмы расширяют инструментарий для организаторов учебного процесса, позволяя использовать не только методы целочисленного программирования (которые на данный момент используются в СПбГЭУ), но и алгоритмы серийной диктатуры. Предложенные алгоритмы являются эффективными (псевдополиномиальными или полиномиальными) в терминах вычислительной сложности. При наличии сильно «раскоррелированных» предпочтений (90 %) профилей доля конфликтов зависти не превышает 30 %.

Ключевые слова: двусторонние рынки, динамическое программирование, жадные алгоритмы.

Для цитирования: Глазунова, Е. В., & Зелин, И. А. (2025). Алгоритмы серийной диктатуры распределения студентов по учебным профилям в университете. *Вестник Воронежского государственного университета. Серия: Экономика и управление*, (3), 3–11. DOI: https://doi.org/10.17308/econ.2025.3/13261

Proceedings of Voronezh State University
Series: Economics and Management

Mathematical and Quantitative Methods

Original article
UDC 519.863
DOI: https://doi.org/10.17308/econ.2025.3/13261
JEL: C61

Serial dictatorship algorithms for assigning students to academic tracks

E. V. Glazunova[1✉], I. A. Zelin[2]

[1, 2] Saint Petersburg State University of Economics, 21 Sadovaya str., 191023, St. Petersburg, Russian Federation

Object. Serial dictatorship algorithms are allocation mechanisms where agents sequentially, in a predetermined order, select their most preferred available. These algorithms are studied within the problem of allocating students to academic tracks.
Purpose. This article investigates serial dictatorship algorithms for assigning students to academic tracks in universities, where students sequentially choose their most preferred available slots in academic tracks based on their academic performance (GPA) ranking.
Methodology. We proposed two algorithms that use dynamic programming methods, along with a greedy algorithm that offers greater efficiency in terms of time complexity.
Results. The algorithms were tested using real data from Saint Petersburg State University of Economics (UNECON) for student allocation to academic tracks in the "Economics" program, alongside large-scale computations on synthetic data. We analyzed the relationship between correlation patterns in student preferences for academic tracks and the number of envy conflicts for the described algorithms.
Conclusions. The developed algorithms expand the toolkit available to academic administrators, enabling the use of not only integer programming methods (currently employed at UNECON) but also serial dictatorship algorithms. The proposed algorithms are efficient (pseudopolynomial or polynomial) in terms of computational complexity. With strongly "decorrelated" preferences (90 %), the proportion of envy conflicts does not exceed 30 %.

Key words: student allocation to academic tracks, two-sided markets, dynamic programming, greedy algorithms.

For citation: Glazunova, E. V., & Zelin, I. A. (2025). Serial dictatorship algorithms for assigning students to academic tracks. *Proceedings of Voronezh State University. Series: Economics and Management*, (3), 3–11. DOI: https://doi.org/10.17308/econ.2025.3/13261

Введение

Одной из задач проекта «Умный университет» (Васильев и др., 2024) является задача о распределении студентов по учебным профилям в университете. В данной задаче необходимо найти назначение студентам учебных профилей так, чтобы для каждого учебного профиля можно было сформировать несколько учебных групп в заданных пределах, а также чтобы размер каждой группы тоже был в заданных пределах. При этом каждый студент имеет предпочтения по множеству учебных профилей, а каждый профиль имеет предпочтения по множеству студентов – рейтинг на основе средневзвешенного балла успеваемости студентов. Назначение студентам учебных профилей должно быть таким, чтобы предпочтения сторон были учтены.

Данную задачу можно рассматривать как задачу двусторонних рынков, где одна сторона – студенты, вторая – профили. Это задача один-на-много, где агенту одной стороны может быть назначен не более чем один агент противоположной стороны, а агенту другой стороны – несколько агентов противоположной стороны. Однако отличительной чертой данной задачи является то, что у стороны, которой может быть назначено несколько агентов противоположной стороны, есть не только максимальная квота, но и минимальная – минимальный и максимальный размер учебных групп, более того, таких групп может быть открыто несколько (в пределах ограничений) и каждая учебная группа должна удовлетворять минимальной и максимальной квоте.

В работе Ю. М. Васильева и др. (2024) представлена задача целочисленного программирования, позволяющая решить данную задачу. Данная задача сформулирована в предположении, что предпочтения профилей по множеству студентов различны. Однако на практике вместо средневзвешенного балла часто используется средний балл, и предпочтения всех профилей по множеству студентов совпадают с общим рейтингом студентов, составленному по успеваемости студентов. В этом случае задача может быть упрощена, и вместо целочисленного программирования может использоваться алгоритм серийной диктатуры (Satterthwaite & Sonnenschein, 1981), где первый студент рейтинга выбирает из всего пространства решений наилучшие для себя решения, а последующие студенты выбирают наилучшие для себя решения с учетом выбора предыдущих студентов.

Обзор предшествующих работ

Как было сказано выше, рассматриваемая задача может быть представлена в виде задачи о поиске распределения на двустороннем рынке (Manlove, 2013). Примерами таких рынков являются задача о распределении студентов по колледжам, распределение выпускников по рабочим местам, сотрудников по проектам или пациентов больницы по палатам (Boehmer & Heeger, 2022). Из них наиболее близкой задачей, рассматриваемой в данной работе, является задача распределения студентов по колледжам, которая является задачей вида один-на-много, где каждому студенту может быть назначен не более чем один колледж, а каждому колледжу – несколько студентов в пределах максимальной квоты (Gale & Shapley, 2013; Roth & Sotomayor, 1990). Ключевым понятием для данной задачи является понятие стабильности. Распределение называется стабильным, если оно индивидуально рационально, т. е. не существует такого назначения студенту колледжа, что этому студенту предпочтительнее не быть распределенным в колледж вовсе, чем быть распределенным в данный колледж, или же колледж предпочитает не взять студента вовсе, чем взять данного студента; а также не существует блокирующих пар, т. е. пар вида «студент – колледж», которые не назначены друг другу в распределении, однако предпочитают друг друга своим назначениям. Стабильное распределение в таком случае может быть найдено с помощью алгоритма отложенного принятия (Gale & Shapley, 2013).

Однако на практике могут возникать задачи, в которых требуется найти решение, удовлетворяющее не только максимальным квотам, но и минимальным.

Одной из таких моделей является модель поиска распределения, где, говоря в терминах задачи распределения студентов по колледжам, каждый колледж должен быть или открыт, и тогда количество студентов, назначенных этому колледжу, должно быть в пределах от минимальной до максимальной квоты, или этот колледж должен быть закрыт, что означает, что ему не назначаются студенты вовсе (Biró et al., 2010). В работе Ágoston et al.

(2016) представлена задача целочисленного программирования, позволяющая решить данную задачу.

Альтернативной моделью является модель, где каждому колледжу должно быть назначено студентов в пределах от минимальной до максимальной квоты (Hamada et al., 2011). В такой задаче не для любых исходных данных существует допустимое решение, однако проверить, существует ли допустимое решение, можно за полиномиальное время (Gabow, 1983). Для решения данной задачи часто используется алгоритм ACDA (Artificial Cap Deferred Acceptance), который искусственно занижает максимальные квоты колледжам, гарантируя, чтобы минимальные квоты были достигнуты путем запуска алгоритма отложенного принятия (Hamada et al., 2017). В работе Fragiadakis et al. (2015) представлены алгоритмы ESDA, который работает по модифицированной процедуре отложенного принятия, где места в школах разделяются на два типа: регулярные места, равные минимальной квоте школы, и расширенные места, равные разнице между максимальной и минимальной квотами. Студенты подают заявления сначала на регулярные, а затем на расширенные места согласно своим предпочтениям. Механизм ограничивает назначение расширенных мест таким образом, чтобы обеспечить полное заполнение регулярных мест и, как следствие, выполнение всех минимальных квот. В работе Hamada et al. (2016) также исследуется задача поиска распределения с минимальными квотами, и было показано, что задача о поиске распределения с минимальным числом блокирующих пар NP-трудная.

Методология исследования

Формальная постановка задачи имеет следующий вид. Дано множество студентов $S, |S| = n$, подлежащих распределению на учебные профили, также дано множество учебных профилей $C, |C| = m$. Для каждого учебного профиля $c \in C$ определено минимальное количество групп $\underline{g_c}$, которое должно быть открыто на профиле, и максимальное количество групп $\overline{g_c}$, которое может быть открыто на профиле; также определены минимальный и максимальный размер учебной группы $\underline{p_c}$ и $\overline{p_c}$ соответственно. Для каждого студента $s \in S$ определено полные строгие отношение предпочтений по множеству профилей P_s. Учебные профили также формируют полные строгие предпочтения по множеству студентов на основе средневзвешенной успеваемости студентов. Однако часто вместо средневзвешенного балла для расчета рейтинга студентов используется средний балл, и таким образом рейтинги – предпочтения учебных профилей по студентам – совпадают и обозначаются как P_C. В этом случае могут использоваться алгоритмы серийной диктатуры для поиска распределения студентов по учебным профилям.

Распределение $\mu \subset S \times C$ и $z_c \in Z_{\geq 0}, \forall c \in C$ – множество пар вида «студент – учебный профиль», а также количество открытых групп для каждого профиля z_c. Распределение считается допустимым, если:

1) $\left|\{(s,c) | (s,c) \in \mu\}\right| = 1, \forall s \in S$ – каждому студенту назначен ровно один учебный профиль;

2) $\underline{g_c} \leq z_c \leq \overline{g_c}, \forall c \in C$ – число открытых учебных групп на каждом профиле в заданных пределах;

3) $\underline{g_c} \leq z_c \leq \overline{g_c}, \forall c \in C$ – число студентов, назначенных на учебный профиль в заданных пределах.

Рассматриваемая задача является обобщением задачи о поиске распределения студентов по колледжам с минимальными квотами (Biró et al., 2010), для которой известно, что стабильного распределения, т. е. распределения без блокирующих пар и блокирующих коалиций, может не существовать. Поэтому к решению могут быть предъявлены более слабые требования, а именно отсутствие конфликтов зависти и нерасточительности.

Конфликт зависти в распределении μ означает существование такой пары «студент – учебный профиль» $(s,c) \in S \times C \setminus \mu$, где c – профиль, открытый в результате распределения ($z_c > 0$), что:

1) $cP_s\mu(s)$ – профиль конфликта строго предпочтительнее профиля $\mu(s)$ – профиля, назначенного студенту s в распределении μ;

2) $\exists s' \in \mu(c) : sP_C s'$ – студент из пары конфликта строго предпочтительнее хотя бы одного из студентов $\mu(c)$ – множества назначенных студентов профилю μ в распределении μ.

Распределение называется нерасточительным, если не существует такой пары «студент – учебный профиль» $(s,c) \in S \times C \setminus \mu$, где c – профиль, открытый в результате распределения $(z_c > 0)$, что:

1) $cP_s\mu(s)$ – профиль конфликта строго предпочтительнее профиля $\mu(s)$ – профиля, назначенного студенту s в распределении ;

2) $|\mu(c)| < \overline{p_c} \times z_c$ и $|\mu(\mu(s))| > \underline{p}_{\mu(s)} \times z_{\mu(s)}$ – на профиле c существуют свободные места, а при исключении студента s из профиля $\mu(s)$ у профиля $\mu(s)$ не нарушится ограничение на минимальный размер каждой группы.

Задача о проверке существования допустимого распределения схожа с задачей о размене монет (Окулов, 2017), которая может решаться методами динамического программирования (Powell, 1955). Идея в том, что купюра, которую необходимо разменять – количество студентов, а номиналы монет – допустимые размеры профилей. Используя эту идею, был разработан алгоритм серийной диктатуры распределения студентов на профили на основе динамического программирования, который является модификацией алгоритма решения задачи о размене.

Результаты исследования

Введем необходимые обозначения. Рассматривая упорядоченный набор учебных профилей C, обозначим как N упорядоченное по профилям объединение упорядоченных по возрастанию множеств всех допустимых размеров профиля c. Для заданного c его подмножество может быть описано как $N_c = g \in \left[\underline{g_c} : \overline{g_c}\right] \left\{ (p \times g, c), p \in \left[\underline{p_c} : \overline{p_c}\right] \right\}$. Обозначим как $N = c \in CN_c$, пусть i-й элемент упорядоченного множества N обозначается как N_i.

Пусть минимальное число студентов, которое должно быть назначено на все профили до текущего рассматриваемого профиля, обозначается как LB_c. $isFirst_i$ – параметр, принимающий значение 1, если $i = 1$ или $N_{i,2} \neq N_{i-1,2}$, и 0 в обратном случае.

$$lastPos_i = \begin{cases} 0, N_{i,2} = C_1, \\ \max\{j | j \le i, N_{j,2} = S_{i,2}\}, \text{иначе.} \end{cases}$$

Для каждого студента в порядке рейтинга P_C создадим таблицу A размера $(|N| + 1, n + 1)$. Инициализируем первую строку: первый элемент положим равным нулю, остальные – $+\infty$. Значения в строках ниже посчитаем итеративно по формуле

$$A_{i,j} = \begin{cases} \min\left(A_{i-1,j-N_{i-1,1}}, P_s\left(N_{i-1,2}\right)\right), & j - N_{i-1,1} - LB_{i-1} \ge 1 \wedge \\ & isFirst_{i-1} = 1 \wedge N_{i-1,1} \neq 0 \wedge \\ & A_{lastPos_{i-1}, j-N_{i-1,1}} \neq +\infty, \\ A_{i-1,j-N_{i-1,1}}, & j - N_{i-1,1} - LB_{i-1} \ge 1 \wedge \\ & isFirst_{i-1} = 1 \wedge N_{i-1,1} \neq 0 \\ \min\left\{A_{i-1,j}, P_s\left(N_{i-1,2}\right), A_{lastPos_{i-1}, j-N_{i-1,1}}\right\}, & \\ & j - N_{i-1,1} - LB_{i-1} \ge 1 \wedge \\ & isFirst_{i-1} \neq 1 \wedge \\ & A_{lastPos_{i-1}, j-N_{i-1,1}} \neq +\infty, \\ +\infty, & j - N_{i-1,1} - LB_{i-1} < 1 \wedge \\ & isFirst_{i-1} = 1 \\ A_{i-1,j}, & \text{в обратном случае.} \end{cases}$$

где $P_s\left(N_{i-1,2}\right)$ означает позицию профиля $N_{i-1,2}$ в списке предпочтений студента s.

Пусть в результате предыдущего шага студент s назначен на профиль c, тогда необходимо пересчитать $N_c := \{i \in N_c : i - 1 \ge 0\}$. Затем необходимо обновить множество N, удалить из множества студентов студента s, уменьшить n на единицу и переопределить введенные ранее параметры, назначая на наиболее предпочтительный допустимый профиль следующего студента в рейтинге, если множество оставшихся студентов не пустое, иначе распределение найдено. После того как найдено назначение студентов на профили, остается определить $z_c = \left\lceil \mu(c) \setminus \overline{p_c} \right\rceil, \forall c \in C$ – количество создаваемых групп на каждом профиле. Далее необходимо распределить студентов по группам учебного профиля, делать это можно исходя из различных критериев (например, чтобы все группы имели как можно более похожий средний балл).

Смысл таблицы A заключается в том, что в ней хранится позиция в списке предпочтений наиболее предпочтительного профиля для рассматриваемого студента среди текущих рассмотренных профилей, на который можно распределить студента с учетом гарантии существования распределения всех студентов по профилям.

Таблица A пересчитывается по первой строке формулы (1), если студентов достаточно для «размена» (т. е. можно гарантировать существование распределения студентов по профилям), и это первая «монета» профиля (первый допустимый размер профиля), которая не равна нулю (т. е. на этом шаге можем распределить студента на этот профиль), и при этом $A_{lastPos_{i-1},j-N_{i-1,1}} \neq +\infty$, что говорит о существовании решения, где студент назначается в текущий рассматриваемый профиль, то берется профиль максимального приоритета между всеми предыдущими рассмотренными профилями и текущими.

Таблица A пересчитывается по второй строке формулы (1), если выполнены все условия строки 1, кроме того, что может быть открыт текущий рассматриваемый профиль – решения не существует, в этой ячейке таблицы будет записано $A_{i-1,j-N_{i-1,1}} = +\infty$.

Таблица A пересчитывается по третьей строке формулы (1), если студентов достаточно для «размена», но текущая «монета» не первая для рассматриваемого профиля, и при этом $A_{lastPos_{i-1},j-S_{i-1,1}}$ не равен $+\infty$, что значит, что можно добавить текущую «монету» к «размену», берется минимум между: предыдущим разменом без текущей «монеты» (текущего размера профиля); профилем, соответствующему текущей «монете», а также минимальному профилю по предпочтению из «размена» без текущего профиля, с учетом того, что в «размене» участвует текущая «монета».

Таблица A пересчитывается по четвертой строке формулы (1), если студентов недостаточно «для размена», а «монета» является первой из набора профиля – это значит, что «размена» нет, отсюда ответ $+\infty$.

Таблица A пересчитывается по пятой строке формулы (1), если не можем использовать текущую «монету» в «размене», берется последний «размен» без взятия текущей «монеты» в «размен».

Вычислительная сложность данного алгоритма является псевдополиномиальной (Дроздов, 2000) и определяется как $O\left(n^3 \times N\right)$.

Несмотря на то, что алгоритм на базе динамического программирования является на практике довольно эффективным, могут быть разработаны более эффективные алгоритмы в терминах временной сложности, например жадные алгоритмы (Cormen et al., 2009).

В основе представленных далее подходов лежит следующая интуитивная идея. Студенты в порядке рейтинга P_C подают заявки на профили в соответствии с приоритетами. Удовлетворить заявку, если не исчерпана верхняя квота на профиле, можно двумя способами:

1) если уже открыт набор в группу соответствующего профиля и не исчерпан максимальный размер каждой группы профиля, студент может быть определен туда:

2) если такой группы нет, но есть достаточное число студентов в эксцессе и студентов сверх минимального размера групп в других группах того же профиля, с помощью которых можно поддержать минимальный размер новой группы на профиле, допускается открытие новой группы профиля и данный студент не требуется для поддержки других минимальных квот, то можно открыть новую группу и определить студента туда.

Эксцессом здесь называется разность между числом студентов, еще не распределенных на профили, и суммой нижних квот уже открытых групп, еще не покрытых распределенными студентами. Мы утверждаем также, что если ни один из способов удовлетворения заявки ни для одного из профилей из предпочтений студента невозможен, то допустимого распределения не существует.

Асимптотическая сложность алгоритма оценивается в $O\left(n \times m\right)$: каждый студент подает заявку не более чем на m профилей, процедуры обработки заявки и предобработки данных для профиля можно считать атомарными.

Рассмотрим подробнее выполнение свойств отсутствия конфликтов зависти и нерасточительности. Нерасточительность в обоих алгоритмах, как и в алгоритме MSDA (Fragiadakis et al., 2015), инвариантна: на каждом шаге гарантируется, что студент отправится на менее предпочтительный профиль при наличии мест на более предпочтительном профиле только в том случае, если его участие действительно необходимо для поддержки минимального размера группы, введенной изначально или возникшей для удовлетворения предпочтений более приоритетного студента. Этот же инвариант обеспечивает справедливость при условии строгой упорядоченности общего рейтинга студентов.

Обсуждение результатов

Описанные алгоритмы были реализованы на языке программирования Python (Rossum, 1995). Для тестирования использовались реальные анонимизированные данные Санкт-Петербургского государственного экономического университета, а также сгенерировано большое число рандомизированных наборов данных.

В таблице представлено сравнение времени счета описанных алгоритмов, а также подхода на базе целочисленного программирования (Васильев и др., 2024) на реальных данных СПбГЭУ для 316 студентов направления «Экономика», подлежащих распределению на семь профилей.

Таблица

Время работы алгоритмов

Алгоритм	Время счета, сек.
Целочисленное программирование	156
Динамическое программирование	43
Жадный алгоритм	<1

Также были проведены массовые расчеты на синтетических данных для сравнения времени счета представленных алгоритмов. На рис. 1 представлена зависимость времени счета представленных алгоритмов, где ДП – подход на базе динамического программирования, а жадный – подход на основе жадного алгоритма. Оба алгоритма демонстрируют возможность их применения на практике, однако жадный алгоритм имеет значительное преимущество с точки зрения временных затрат, являясь при этом точным алгоритмом.

Описанные алгоритмы гарантируют отсутствие конфликтов зависти в случае, если предпочтения профилей по студентам одинаковы. Однако, если предпочтения профилей будут различны, а поиск распределения будет происходить на основе общего рейтинга, рассчитанного по среднему баллу студентов, то как много конфликтов зависти это породит? Для ответа на этот вопрос были проведены вычислительные эксперименты на синтетических данных. Была определена зависимость между уровнем раскоррелированности (единица минус коэффициент корреляции) предпочтений профилей с общим рейтингом, определенным по среднему баллу студентов, и долей студентов, участвующих в конфликтах зависти.

Предпочтения профилей генерировались в соответствии с Shimada et al. (2020). На рис. 2

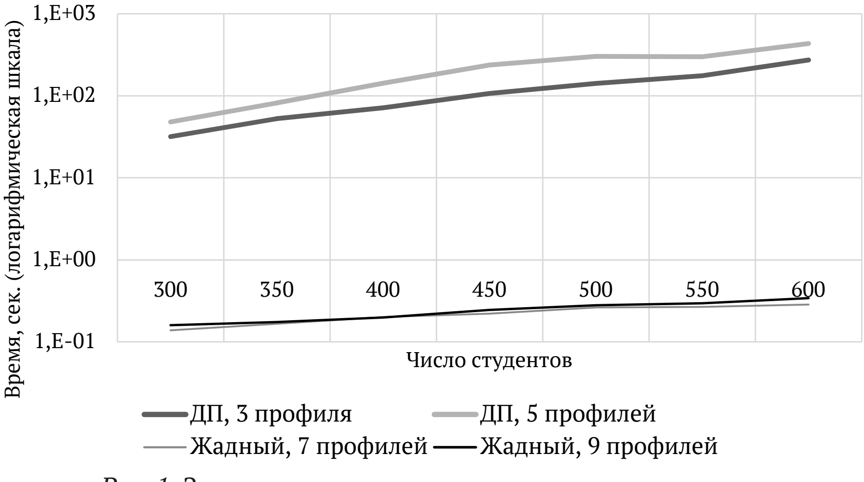

Рис. 1. Зависимость времени счета от размера задачи

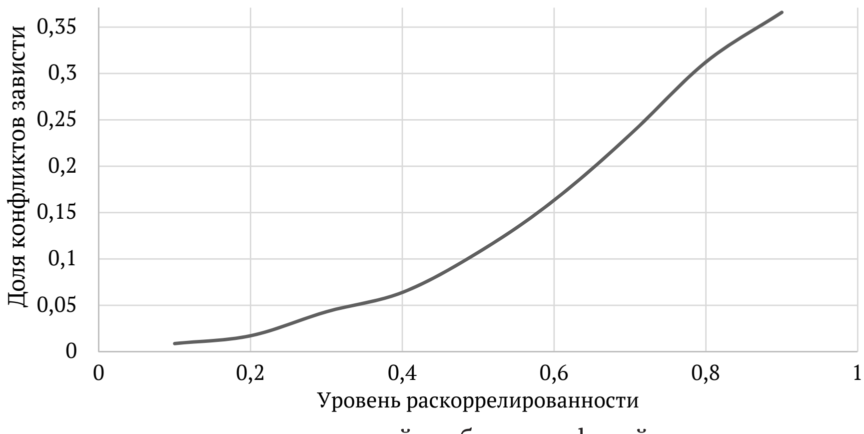

Рис. 2. Связь раскоррелированности предпочтений учебных профилей и количество конфликтов зависти

представлена зависимость уровня раскоррелированности предпочтений учебных профилей и количества студентов, участвующих в конфликтах зависти. Даже при уровне раскоррелированности предпочтений в 90 % в среднем наблюдается не более 35 % студентов, участвующих в конфликтах зависти.

Заключение

В статье представлены алгоритмы серийной диктатуры для решения задачи распределения студентов по учебным профилям, т. е. алгоритмы, где студенты в порядке своего рейтинга выбирают наилучший допустимый для себя профиль с учетом выбора предыдущих студентов в рейтинге.

Для описанных алгоритмов была определена асимптотическая вычислительная сложность, при этом жадный алгоритм оказывается на два порядка эффективнее, что также демонстрируется массовыми вычислительными экспериментами, однако оба алгоритма применимы на практике с точки зрения временных затрат и оказываются эффективнее целочисленного программирования. При этом в силу большей эффективности жадного алгоритма (в части временных затрат на поиск решения) рекомендуется использовать именно его.

Вычислительные эксперименты показывают, что описанные алгоритмы применимы к данной задаче не только в том случае, если предпочтения учебных профилей совпадают и определяются по среднему баллу студентов (в этом случае гарантируется отсутствие конфликтов зависти и нерасточительность распределения), но и в случае, если предпочтения профилей определяются на основе средневзвешенного балла, где весовые коэффициенты профилей отличаются: тогда даже при высоком уровне раскоррелированности предпочтений количество студентов, участвующих в конфликтах зависти, не превышает в среднем 35 %.

Направлением дальнейшего исследования является разработка алгоритмов формирования учебных групп профилей согласно различным критериям, например максимизация «похожести» разбиения на группы на предыдущее разбиение студентов по группам и максимизация «похожести» среднего балла в учебных группах одного профиля.

Авторский вклад

Глазунова Е. В.: разработка алгоритма динамического программирования, подготовка текста статьи, руководство проектом.

Зелин И. А.: разработка жадного алгоритма, программная реализация описанных алгоритмов, проведение массовых расчетов.

Contribution of the authors

Glazunova E. V.: development of the dynamic programming algorithm, manuscript preparation, project management.

Zelin I. A.: development of a greedy algorithm, software implementation of described algorithms, conducting large-scale computations.

Конфликт интересов

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Conflict of Interest

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

Библиографический список

1. Васильев, Ю. М., Глазунова, Е. В., & Фридман, Г. М. (2024). Умный университет: справедливое распределение студентов по учебным профилям. *Вестник Воронежского государственного университета. Серия: Экономика и управление*, (1), 16–24. [Vasiliev, Yu. M., Glazunova, E. V., & Friedman, G. M. (2024). “Smart University” Project: fair matching of students to academic trajectories. *Proceedings of Voronezh State University. Series: Economics and Management*, (1), 16–24. (In Russian).] DOI: 10.17308/econ.2024.1/11838

2. Дроздов, С. Н. (2000). *Комбинаторные задачи и элементы теории вычислительной сложности*. Таганрог: Изд-во ТРТУ. [Drozdov, S. N. (2000). *Combinatorial Problems and Elements of Computational Complexity Theory*. Taganrog: TRTU Publishing House. (In Russian).]

3. Окулов, С. М. (2017). *Динамическое программирование*. Москва: БИНОМ. Лаборатория знаний. [Okulov, S. M. (2017). Dynamic Programming. Moscow: BINOM. Laboratory of Knowledge Publishing House. (In Russian).]

4. Ágoston, K. C., Biró, P., & McBride, I. (2016). Integer programming methods for special college admissions problems. *Journal of Combinatorial Optimization*, (32), 1371–1399. DOI: 10.1007/s10878-016-0085-x

5. Biró, P., Fleiner, T., Irving, R. W., & Manlove, D. F. (2010). The College Admissions problem with lower and common quotas. *Theoretical Computer Science*, *411*(34–36), 3136–3153. DOI: 10.1016/j.tcs.2010.05.005

6. Boehmer, N., & Heeger, K. (2022). A Fine-Grained View on Stable Many-To-One Matching Problems with Lower and Upper Quotas. *ACM Transactions on Economics and Computation*, *10*(2), 1–53. DOI: 10.1145/354660

7. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. Cambridge: The MIT Press.

8. Fragiadakis, D., Iwasaki, A., Troyan, P., Ueda, S., & Yokoo, M. (2015). Strategyproof matching with minimum quotas. *ACM Transactions on Economics and Computation*, *4*(1), 1–40. DOI: 10.1145/2841226

9. Gabow, H. N. (1983). An Efficient Reduction Technique for Degree-Constrained Subgraph and Bidirected Network Flow Problems. *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, 448–456. DOI: 10.1145/800061.808776

10. Gale, D., & Shapley, L. S. (2013). College admissions and the stability of marriage. *American Mathematical Monthly*, *120*(5), 386–391. DOI: 10.4169/amer.math.monthly.120.05.386

11. Hamada, K., Iwama, K., & Miyazaki, S. (2016). The hospitals/residents problem with lower quotas. *Algorithmica*, *74*, 440–465. DOI: 10.1007/s00453-014-9951-z

12. Hamada, K., Iwama, K., & Miyazaki, S. (2011). *The Hospitals/Residents Problem with Quota Lower Bounds*. In: Demetrescu, C., & Halldórsson, M. M. (eds.). Lecture Notes in Computer Science, 6942. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-23719-5_16

13. Hamada, N., Hsu, C. L., Kurata, R., Suzuki, T., Ueda, S., & Yokoo, M. (2017). Strategy-proof school choice mechanisms with minimum quotas and initial endowments. *Artificial Intelligence*, (249), 47–71. DOI:10.1016/j.artint.2017.04.006

14. Manlove, D. (2013). Algorithmics of Matching Under Preferences. *World Scientific*. DOI:10.1142/8591

15. Powell, W. B. (1955). *Approximate dynamic programming: solving the curses of dimensionality*. Princeton University. The Department of Operations Research and Financial Engineering Princeton.

16. Rossum, G. V. (1995). *Python tutorial, Technical Report CS-R9526*. Amsterdam: Centrum voor Wiskunde en Informatica (CWI).

17. Roth, A. E., & Sotomayor, M. A. O. (1990). *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis.* Cambridge University Press.

18. Satterthwaite, M. A., & Sonnenschein, H. (1981). Strategy-Proof Allocation Mechanisms at Differentiable Points. *The Review of Economic Studies*, *48*(4), 587–597. DOI:10.2307/2297198540

19. Shimada, N., Yamazaki, N., & Takano, Y. (2020). Multi-objective Optimization Models for Many-to-one Matching Problems. *Journal of Information Processing*, (28), 406-412. DOI: 10.2197/ipsjjip.28.406

Глазунова Екатерина Валерьевна, ассистент, Санкт-Петербургский государственный экономический университет, Санкт-Петербург, Российская Федерация

E-mail: katarina.glazunova97@inbox.ru

ORCID ID: 0000-0001-9936-8961

Зелин Иван Алексеевич, студент, Санкт-Петербургский государственный экономический университет, Санкт-Петербург, Российская Федерация

E-mail: ivan.zelin@study.thws.de

ORCID ID: 0009-0009-2150-0219

Ekaterina V. Glazunova, Assist. Prof., Saint Petersburg State University of Economics, Saint Petersburg, Russian Federation

E-mail: katarina.glazunova97@inbox.ru

ORCID ID: 0000-0001-9936-8961

Ivan A. Zelin, student, Saint Petersburg State University of Economics, Saint Petersburg, Russian Federation

E-mail: ivan.zelin@study.thws.de

ORCID ID: 0009-0009-2150-0219

Получена 04.07.2025
Получена в доработанном виде 19.09.2025
Одобрена 26.09.2025

Received 04.07.2025
Revisited 19.09.2025
Accepted 26.09.2025