

УДК: 004.9:378.1:519.854.33

DOI: 1030987/article\_5bfd98c4da31b1.20111431

Д.А. Коростелёв, С.Н. Зимин

## МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ХОДЕ НАБОРА АБИТУРИЕНТОВ В ВУЗ

*В статье рассмотрена проблема повышения эффективности набора абитуриентов. Приведено формальное описание задачи максимального заполнения бюджетных мест и максимизация среднего балла абитуриентов, поступающих на эти места, а также математического и программного обеспечения поддержки принятия решения, позволяющее решать эту задачу. Приведены рекомендации по использованию представленного математического и программного обеспечения.*

**Ключевые слова:** набор абитуриентов в вуз, поддержка принятия решений, математическое и программное обеспечение, бинарное программирование.

D.A. Korostelyov, S.N. Zimin

## THE SOFTWARE OF DECISION SUPPORT MAKING DURING ENROLLMENT OF ENTRANTS IN HIGHER EDUCATION INSTITUTION

*The article deals with the problem of increasing the efficiency of recruitment of applicants. A formal description of the task of maximally filling budget places and maximizing the average score of applicants entering these places, as well as the decision support software, allows to solve this problem. The recommendations on the use of the presented software are given.*

**Key words:** admission to university, decision support, software, binary programming.

### Введение

Перед многими современными вузами РФ стоит задача качественного набора абитуриентов. Для этих целей вузы постоянно проводят профориентационную работу, осуществляют планирование набора абитуриентов и организуют приёмную кампанию в рамках действующего законодательства РФ [1]. В ходе приёмных кампаний в последние годы большинство вузов сталкивается с проблемами, связанными с набор абитуриентов, в частности с заполнением бюджетных мест. Вызвано это демографическими проблемами в РФ в 90-х годах XX века, высокой конкуренцией между вузами, а также порой недостаточно эффективной профориентационной работой [2].

Ежегодно для организации набора студентов и приёма документов поступающих, а также проведения вступительных испытаний и зачисления в состав студентов лиц, прошедших по конкурсу, в вузах организуется приёмная комиссия. Приёмная комиссия, в свою очередь, обеспечивает соблюдение прав граждан на образование, установленных Конституцией Российской Федерации, гласность и открытость проведения всех процедур приёма.

Целью приёмной комиссии является организация процесса формирования контингента студентов из числа наиболее способных и подготовленных лиц. Для достижения этой цели приёмная комиссия взаимодействует с подразделениями вуза, разрабатывает предложения и проводит согласованные организационные мероприятия, направленные на обеспечение набора в вуз [3]. Одними из ключевых задач приёмной комиссии являются максимальное заполнение бюджетных мест и максимизация среднего балла абитуриентов, поступающих на эти места.

Эффективность решения этих задач отчасти зависит от действий сотрудников приёмной комиссии, которые на основе своего опыта и указаний руководства способны

посоветовать сменить абитуриентам приоритеты выбираемых направлений в своих заявлениях. К сожалению, эти рекомендации ничем, кроме интуиции и опыта сотрудников приёмной комиссии не подкреплены, а значит, в этом случае нельзя говорить о том, что найдены оптимальные решения для формирования рекомендаций абитуриентам. Повысить объективность формирования подобных рекомендаций можно достичь за счёт применения соответствующего математического и программного обеспечения поддержки принятия решений. В настоящее время не существует специализированных программных систем, решающих указанные задачи. Поэтому разработка таких систем поддержки принятия решений для сотрудников приёмной комиссии в ходе набора абитуриентов, а также разработка соответствующего математического обеспечения, является актуальной проблемой, решение которой рассматривается в данной работе.

## 1. Постановка задачи

Формально задачу максимального заполнения бюджетных мест и максимизация среднего балла абитуриентов, поступающих на эти места можно описать следующим образом.

Имеется  $n$  направлений подготовки образующих множество  $N=\{N_{ij}\}$ . Будем считать, что  $b_i(t)$  – это количество бюджетных мест на  $i$ -е направление подготовки в момент времени  $t$ ,  $M(t)$  – множество всех подавших заявления абитуриентов к моменту времени  $t$ , а  $M_i(t)$  – множество абитуриентов, подавших заявления на  $i$ -е направление подготовки к моменту времени  $t$ . Тогда количество баллов абитуриента  $j$  в момент времени  $t$  ( $1 \leq j \leq |M(t)|$ ) на направление подготовки с номером  $i$  определяется величиной  $s_{ij}(t)$ . Для разных направлений подготовки эта величина может отличаться, т.к. в расчете суммарного балла учитываются результаты ЕГЭ по разным дисциплинам. Если абитуриент не подавал заявление на направление подготовки  $i$ , то будем полагать  $s_{ij}(t)=-1$ .

Т.к. в ходе приемной кампании абитуриенты не сразу приносят информацию о своих личных достижениях, а также результаты по ЕГЭ становятся известны не сразу (к тому же некоторые категории абитуриенты сдают ЕГЭ в вузе), то величина  $s_{ij}(t)$  меняется до момента времени  $t_0$  – конца приема документов. Для  $t \geq t_0$  величину  $s_{ij}(t)$  можно считать постоянной.

Каждый абитуриент может подать заявление максимум на 3 направления подготовки в одном вузе [3]. Направления подготовки, на которое подал  $j$ -й абитуриент заявление в момент времени  $t$ , обозначим как множество –  $T_j(t)=\{T_{j,k}(t)\}$ , где  $1 \leq k \leq |T_{j,k}(t)|$ ,  $1 \leq T_{j,k}(t) \leq n$ . При  $t \geq t_0$  – элементы множества  $T_j(t)$  не меняются и равны  $T_j(t_0)$  (однако может меняться их порядок), направление  $T_{j,l}$  считается для абитуриента приоритетным. Абитуриент становится рекомендованным к зачислению только при условии, что он подал оригинал документов, а само зачисление осуществляется в два этапа:

- 1) в момент времени  $t_1$  – до заполнения 80% на бюджетные места,  $B_{1,i} = [0.8 \cdot B_i]$  – количество бюджетных мест на  $i$ -ю специальность на 1-м этапе;
- 2) в момент времени  $t_2$  – до заполнения 100% на бюджетные места,  $B_{2,i} = B_i - B_{1,i}$  – количество бюджетных мест на  $i$ -ю специальность на 2-м этапе.

Обозначим соответствующие множества абитуриентов:  $V_{1,i}(t)$  – рекомендованные к зачислению абитуриенты на 1-м этапе по  $i$ -му направлению подготовки, а  $V_{2,i}(t)$  – на 2-м этапе по  $i$ -му направлению подготовки:

$$V_1(t) = \cup_i V_{1,i}(t);$$

$$V_2(t) = \cup_i V_{2,i}(t).$$

В первую очередь большинство приемных комиссий стремится заполнить бюджетные места, т.е. сделать так, чтобы для любого  $i$  ( $1 \leq i \leq n$ ) выполнялось условие:

$$|V_{1,i}(t)| + |V_{2,i}(t)| = B_i.$$

К сожалению, достичь этой цели не всегда возможно. В этом случае стараются минимизировать число не востребуемых бюджетных мест:

$$\sum_{i=1}^N B_i - |V_{1,i}(t)| - |V_{2,i}(t)| \rightarrow \min,$$

с учетом ограничения:

$$|V_{1,i}(t)| + |V_{2,i}(t)| \leq B_i.$$

## 2. Математические методы и алгоритмы решения задачи оптимального набора абитуриентов

В интервал времени  $[t_0; t_1]$  абитуриенты могут приносить оригиналы документов, забирать документы, а также менять приоритет направлений подготовки. Основной задачей приемной комиссии в этот период становится содействие абитуриентам в определении основного для себя направления, т.к. на разные направления разный конкурс и разные шансы поступления. В этот период абитуриентов, подавших оригиналы документов (обозначим их как множество  $M'(t)$ ), можно разделить на 2 подмножества:

- 1)  $G_1(t) = V_1(t)$  – абитуриенты, попавшие в список рекомендованных к зачислению и не желающих менять приоритет направлений;
- 2)  $G_2(t) = M'(t) \setminus V_1(t)$  – абитуриенты, не попавшие в список рекомендованных к зачислению и абитуриенты, попавшие в список рекомендованных к зачислению и допускающих смену приоритета направлений.

В случае, когда остаются не востребуемые бюджетные места, приемная комиссия старается рекомендовать абитуриентам из множества  $G_2(t)$  сменить приоритетное направление подготовки. Может существовать множество вариантов формирования таких рекомендаций  $R_l(t) = \{V'_{1,r}(t)\}$ , где  $V'_{1,r}(t) = \{V'_{1,r,i}(t)\}$ , а  $V'_{1,r,i}(t)$  – список абитуриентов, которым рекомендуется выбрать в качестве приоритетного  $i$ -е направления подготовки в момент времени  $t$ . Среди элементов множества  $R_l$  приемная комиссия стремится выбрать вариант с минимальным значением не востребуемых бюджетных мест:

$$\sum_{i=1}^N B_{1,i} - |V_{1,i}(t)| - |V'_{1,r,i}(t)| \rightarrow \min; \quad (1)$$

$$|V_{1,i}(t)| + |V'_{1,r,i}(t)| \leq B_{1,i}.$$

При этом если таких вариантов несколько, то выбирается тот, в котором сумма баллов абитуриентов множества  $V'_{1,r}(t)$  максимальна, т.е.:

$$V'_{1,r}(t): \sum_{i=1}^n \sum_{j=1}^{|V'_{1,r,i}(t)|} S_{i,V'_{1,r,i,j}(t)} \rightarrow \max.$$

Эта же задача может быть записана следующим образом. Имеется целевая функция  $f_t(x)$ , которую необходимо максимизировать:

$$f_t(x) = \sum_{j=1}^{|G_2(t)|} \sum_{i=2}^{|T_{G_2,j}(t)(t)|} x_{i,j} \cdot S_{i,T_{G_2,j}(t)(t)},$$

где  $x_{i,j}=1$  – определяет, что абитуриенту  $G_{2,j}(t)$  рекомендуется выбрать специальность  $T_{G_2,j}(t),i(t)$ .

Также имеем ограничения:

$$0 \leq x_{i,j},$$

$$\sum_{i=1}^{|T_{G_2,j}(t)|} x_{T_{G_2,j}(t),i}(t).j \leq 1,$$

$$\sum x_{i,j} \leq B_{1,i}; G_{2,j}(t) = i$$

Отметим, что максимизация функции  $f_i(x)$  также максимизирует количество заполненных бюджетных мест, т.е. значение формулы (1) имеет минимально возможное значение. Это обусловлено тем, что при наличии бюджетных мест и кандидатов на них, значение функции  $f_i(x)$  увеличивается, если этого кандидата включить в рекомендации, т.к. сумма его баллов  $> 0$ .

Таким образом, исходная задача сводится к задаче линейного программирования, а для её решения могут быть применены соответствующие хорошо известные методы: симплекс-метод, метод внутренних точек и др. [4, 5]. При этом среди оптимальных решений этой задачи необходимо выбирать только целочисленные, а значит можно применять более эффективные методы – методы целочисленного программирования (как точные, например, метод Гомори [6], так и эвристические [5]). Однако если учесть, что  $\forall x_{i,j}: x_{i,j} \in \{0; 1\}$ , то получаем задачу бинарного программирования, для решения которой могут использоваться различные методы: например, аддитивный алгоритм Балаша [7], или методы динамического программирования. При этом задача бинарного программирования относится к NP-полным задачам, а значит её решение при большой размерности (в случае с абитуриентами – размерность может достигать нескольких сотен переменных) с помощью ЭВМ может занять существенное время. В этом случае необходимо снизить размерность задачи.

Заметим, что в реальной практике работы приемной комиссии множество  $G_2(t)$  может быть разделено на несколько непересекающихся подмножеств – компонент связности, в которых все абитуриенты одного подмножества подали заявления на направления подготовки, которые не имеют пересечений с направлениями подготовки других подмножеств  $G_2(t)$ . Обычно, компоненты связности образуются на базе факультетов или группы кафедр.

Для реализации данного подхода отобразим множества абитуриентов  $M = \{M_1, M_2, \dots, M_k\}$ , где  $k$  – количество абитуриентов, и направлений подготовки  $N = \{N_1, N_2, \dots, N_n\}$  на

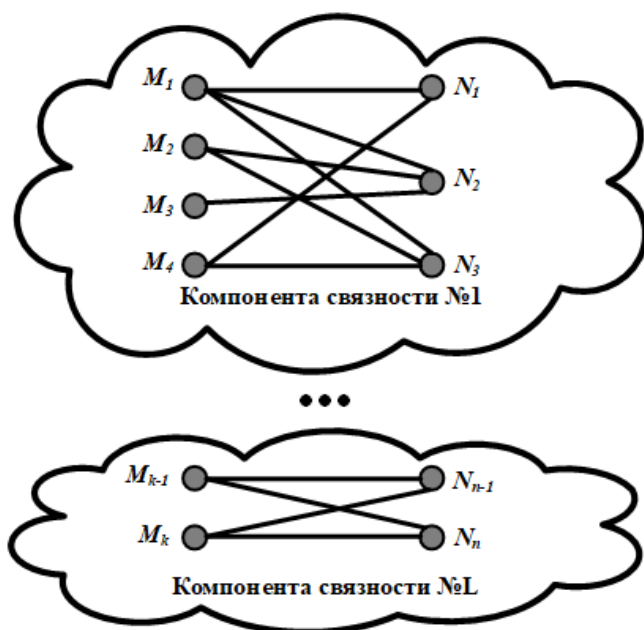


Рис. 1. Пример компонент связности на графе абитуриентов и направлений

вершины графа  $H = \langle C, E \rangle$ , где  $C$  – множество вершин графа,  $C = M \cup N$ ,  $E$  – связи между вершинами. Если абитуриент  $M_j$  подал заявление на направление подготовки  $N_i$ , то включаем эту связь во множество рёбер  $E$  графа  $H$ . В этом случае замкнутые группы абитуриентов и направлений подготовки, в пределах которых абитуриенты могут менять приоритеты в своих заявлениях, образуют отдельные компоненты связности в графе  $H$  (рис. 1). Для поиска компонентов связности на графе применяются соответствующие алгоритмы [8].

Выделение компонент связности позволяет снизить размерность исходной задачи и перейти к независимому решению нескольких задач бинарного программирования меньшей размерности.

Аналогично решается эта же задача в ходе 2-го этапа набора.

### 3. Пример применения разработанного алгоритма

Рассмотрим применение разработанного алгоритма на следующем примере. Пусть имеется 3 направления подготовки ( $n=3$ ): «Направление №1», «Направление №2» и «Направление №3». На каждое из направлений после завершения 1-го этапа набора (момент времени  $t_1 < t_e \leq t_2$ ) осталось по одному бюджетному месту (т.е.:  $B_{2,1}=B_{2,2}=B_{2,3}=1$ ). Имеется 4 абитуриента ( $m=4$ ), подавших заявления на некоторые из этих направлений с определенным приоритетом (табл. 1) и все они не возражают против смены приоритета направлений (т.е.  $G_2(t_e)=\{1, 2, 3, 4\}$ ). Положим, что на направления подготовки одинаковый набор дисциплин, поэтому у абитуриентов будет одинаковый суммарный балл для каждого из них по результатам ЕГЭ (табл. 1).

Таблица 1 Заявления и баллы абитуриентов на соответствующие направления подготовки

| Абитуриенты               | 1-е направление (приоритетное) | 2-е направление | 3-е направление |
|---------------------------|--------------------------------|-----------------|-----------------|
| Абитуриент №1(167 баллов) | «Направление 1»                | «Направление 2» | «Направление 3» |
| Абитуриент №2(182 балла)  | «Направление 1»                | «Направление 3» |                 |
| Абитуриент №3(164 балла)  | «Направление 1»                | «Направление 3» |                 |
| Абитуриент №4(171 балла)  | «Направление 3»                |                 |                 |

Если не предпринимать никаких действий, то получится, что поступят только «Абитуриент №2» и «Абитуриент №4» с суммарным баллом  $182+171=353$ . При этом останется незаполненным одно бюджетное место на «Направлении 3».

Попробуем решить задачу оптимизации распределения приоритетов в заявлениях абитуриентов с помощью разработанного метода. Для данных, приведённых в табл. 1, имеем:  $M_1(t_e)=\{1, 2, 3, 4\}; M_2(t_e)=\{1\}; M_3(t_e)=\{1, 2, 3, 4\}; M_4(t_e)=\{4\}; s_{1,1}(t_e) = s_{2,1}(t_e) = s_{3,1}(t_e) = 167;$   $s_{1,2}(t_e) = s_{3,2}(t_e) = 182; s_{2,2}(t_e) = s_{4,2}(t_e) = -1; s_{1,3}(t_e) = s_{3,3}(t_e) = 164; s_{2,3}(t_e) = s_{4,3}(t_e) = -1;$   $s_{1,4}(t_e) = s_{2,4}(t_e) = s_{3,4}(t_e) = -1; s_{4,4}(t_e) = 171; T_1=\{1, 2, 3\}; T_2=\{1, 3\}; T_3=\{1, 3\}; T_4=\{3\}.$

Имеем следующие переменные, значения которых необходимо определить:

- «Абитуриент №1»:  $x_{1,1}, x_{2,1}, x_{3,1}$ ;
- «Абитуриент №2»:  $x_{1,2}, x_{3,2}$ ;
- «Абитуриент №3»:  $x_{1,3}, x_{3,3}$ ;
- «Абитуриент №4»:  $x_{3,4}$ .

Ограничения:

$$\left\{ \begin{array}{l} 0 \leq x_{1,1}, x_{2,1}, x_{3,1}, x_{1,2}, x_{3,2}, x_{1,3}, x_{3,3}, x_{3,4}; \\ x_{1,1} + x_{2,1} + x_{3,1} \leq 1; \\ x_{1,2} + x_{3,2} \leq 1; \\ x_{1,3} + x_{3,3} \leq 1; \\ x_{3,4} \leq 1; \\ x_{1,1} + x_{1,2} + x_{1,3} \leq 1; \\ x_{2,1} \leq 1; \\ x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} \leq 1. \end{array} \right.$$

Целевая функция:

$$f_{t_e}(x) = 167x_{1,1} + 167x_{2,1} + 167x_{3,1} + 182x_{1,2} + 182x_{3,1} + 164x_{1,3} + 164x_{3,3} + 171x_{3,4} \rightarrow \max$$

Решив эту задачу линейного программирования соответствующим методом [4-7] получим:  $x_{1,1}=0; x_{2,1}=1; x_{3,1}=0; x_{1,2}=1; x_{3,1}=0; x_{1,3}=0; x_{3,3}=0; x_{3,4}=1$ . Это означает, что «Абитуриенту

№1» необходимо в качестве приоритетного направления выбрать «Направление 2», «Абитуриенту 2» – «Направление 1», «Абитуриенту 4» – «Направление 3». При таком раскладе «Абитуриент 3» не попадает в список зачисляемых, поэтому для него рекомендаций нет. Такое решение позволяет закрыть все оставшиеся бюджетные места, а суммарный балл зачисляемых абитуриентов увеличился до  $167+182+171=520$ .

#### 4. Программная реализация математических методов и алгоритмов

Рассмотрим вопросы разработки программного обеспечения, реализующего предложенные математические методы и алгоритмы, с учетом необходимости интеграции его в структуру информационных ресурсов вуза. Для примера будем рассматривать информационные ресурсы Брянского государственного технического университета, в котором работают авторы статьи.

В настоящее время для абитуриентов Брянского государственного технического университета имеется специализированный сайт «БГТУ-Абитуриент», на котором имеется информация о заявлениях абитуриентов на направления подготовки в течение всего периода работы приёмной комиссии за 2016-й и 2017-й год. Сведения о заявлениях абитуриентов и количестве мест для поступающих представлены в виде соответствующих файлов специальной структуры, расположенных на сайте в соответствующем каталоге, а также на веб-страницах самого сайта [9]. Для удобства автоматической обработки этих сведений более эффективным является использование файлов специализированной структуры.

С учетом описанных особенностей информационной системы, с которой необходимо интегрироваться, была разработана схема интеграции программного обеспечения поддержки принятия решений в ходе набора абитуриентов в вуз с веб-сайтом «БГТУ-Абитуриент» (рис. 2).

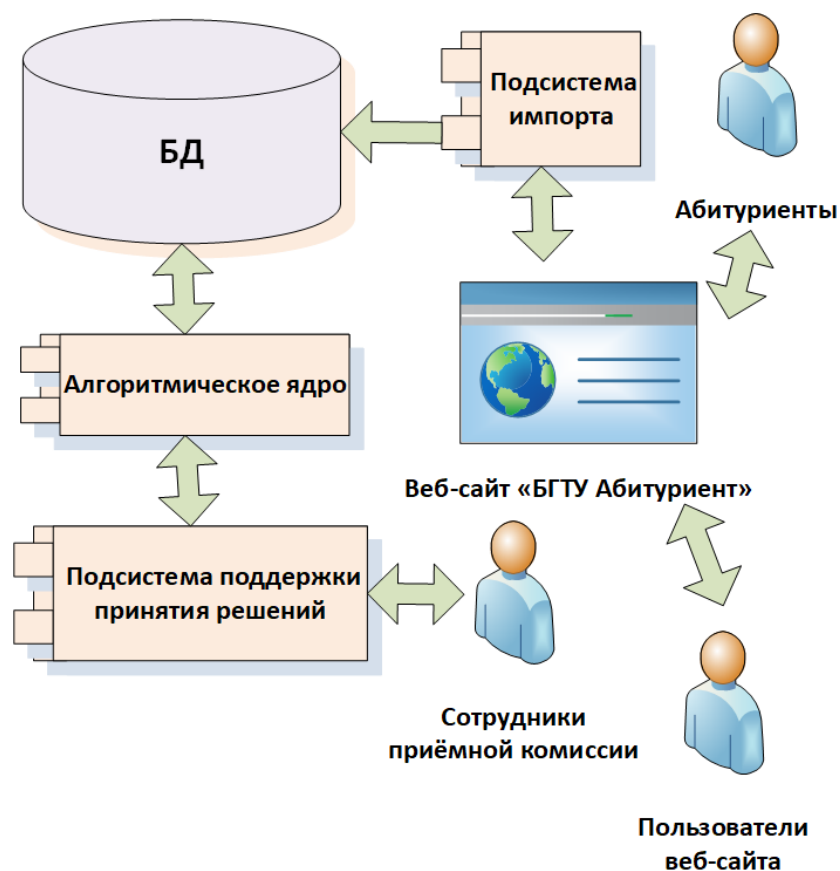


Рис. 2. Схема интеграции программного обеспечения поддержки принятия решений в ходе набора абитуриентов в вуз с веб-сайтом «Абитуриенту БГТУ»

Ключевое место в архитектуре программного обеспечения занимает база данных, которой заполняется в автоматическом режиме с помощью специальной подсистемы импорта. Разработанная база данных приведена к третьей нормальной форме и содержит восемнадцать таблиц (рис. 3).

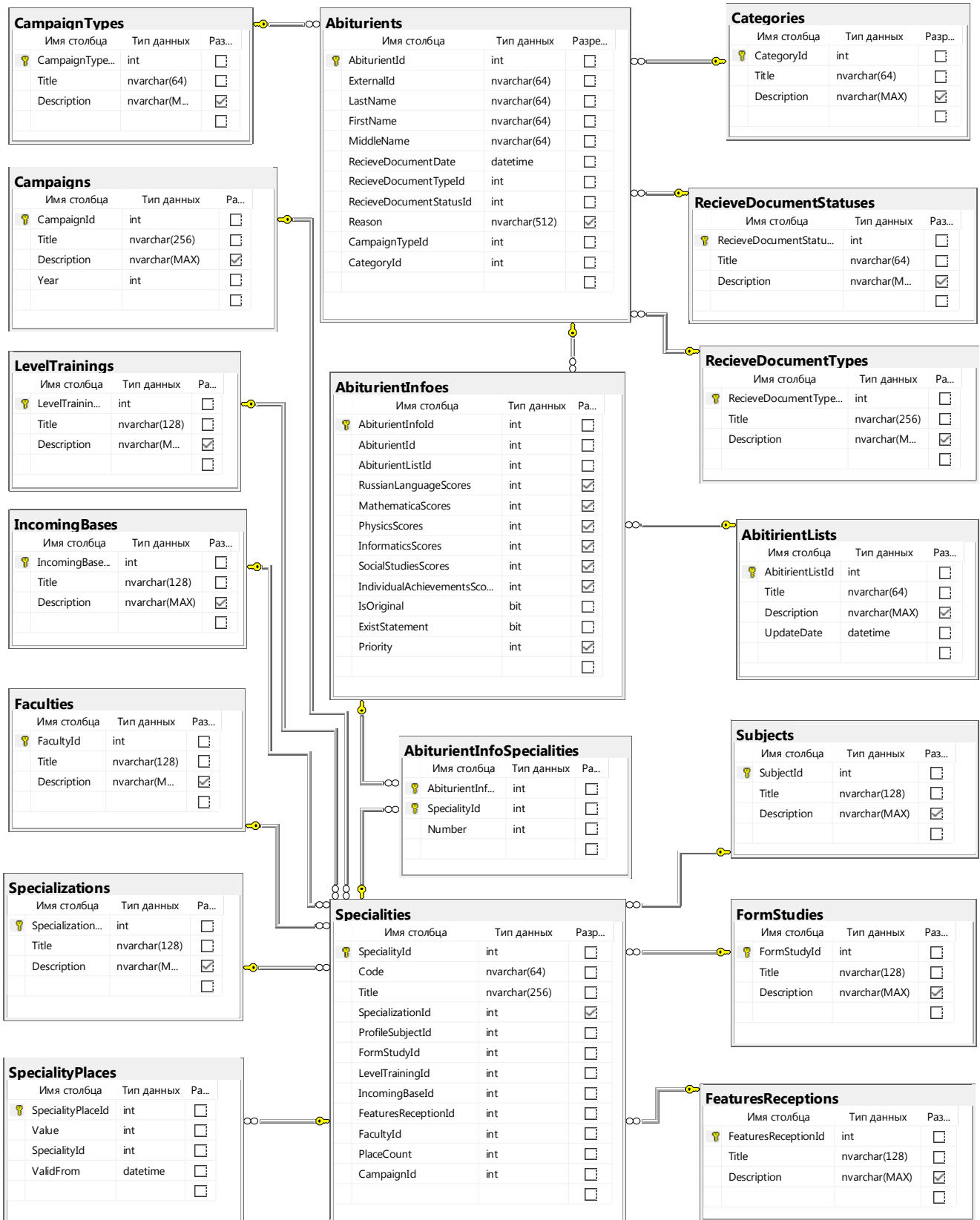


Рис. 3. Модель базы данных хранения сведений об абитуриентах для программного обеспечения поддержки принятия решений в ходе набора абитуриентов в вуз

- 1) «*Specialities*» – содержит данные о специальностях и направлениях подготовки. Является источником сведений для множества  $N$ .
- 2) «*SpecialityPlaces*» – содержит информацию о доступных местах для поступающих. Является источником сведений для функции  $b_i(t)$ .
- 3) «*AbiturientLists*» – содержит данные о списках абитуриентов. Является источником сведений о возможных значениях моментов времени  $t$ .
- 4) «*AbiturientInfoes*» – содержит данные об абитуриентах в конкретный момент времени  $t$ . Является источником сведений для множества  $M(t)$  и  $s_{ij}(t)$ .
- 5) «*AbiturientInfoSpecialities*» – содержит данные о поданных заявлениях абитуриентов на направления подготовки. Является источником сведений для множеств  $M_i(t)$  и  $T_j(t)$ .
- 6) **Справочные таблицы:** «Abiturients» (общие сведения об абитуриентах), «Campaigns» (приёмные кампании), «CampaignTypes» (типы приёмных кампаний), «Categories» (категории набора абитуриентов), «Faculties» (факультеты), «FeaturesReceptions» (особенности приёма), «FormStudies» (формы обучения), «IncomingBases» (основание поступления), «LevelTrainings» (уровень подготовки), «RecieveDocumentStatuses» (статус получения документов), «RecieveDocumentTypes» (тип получения документов), «Specializations» (специализации), «Subjects» (дисциплины).

Формирование оптимальных (с точки зрения приёмной комиссии) вариантов распределения поступающих абитуриентов осуществляется алгоритмическим ядром на основе сведений из приведённой базы данных и передаются в подсистему принятия решений, с которой взаимодействуют секретари приёмной комиссии.

### Заключение

Стоит отметить, что получаемые от системы рекомендации секретарь приёмной комиссии не всегда сможет выполнить, т.к. существуют субъективные факторы, препятствующие этому. Например, абитуриенты, исходя из своих убеждений, могут отказаться от смены приоритетов направлений подготовки в заявлении. Или же могут быть проблемы с личным присутствием студента в приёмной комиссии для подачи обновленного заявления. Учесть эти особенности можно, если добавить соответствующие ограничения при формировании множеств  $M_i(t)$  и  $T_j(t)$  за счет фиксации приоритетного направления подготовки для «проблемных» абитуриентов как единственного.

Также стоит отметить, что ситуация с заявлениями студентов меняется динамически, поэтому формирование рекомендаций следует осуществлять постоянно (например, несколько раз в сутки). Программная реализация предложенных алгоритмов позволит их формировать в автоматическом режиме, что обеспечить секретарям приёмной комиссии оперативную поддержку принятия решений при формировании рекомендаций абитуриентам изменить приоритет направлений подготовки. Эти же рекомендации могут быть выданы и напрямую абитуриентам, чтобы они оперативно и самостоятельно смогли предпринять необходимые меры по смене приоритета направлений подготовки в своих заявлениях.

С помощью разработанных математического и программного обеспечения можно повысить как средний бал студентов, зачисленных в вузы, так и эффективность самой приёмной кампании в целом.



Список литературы:

1. Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 07.03.2018, с изм. от 07.03.2018) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 07.03.2018). Режим доступа: <http://docs.cntd.ru/document/556716692>.
2. Официальная статистика. Население. Демография. Естественное движение населения. Режим доступа: [http://www.gks.ru/wps/wcm/connect/rosstat\\_main/rosstat/ru/statistics/population/demography/](http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/population/demography/).
3. Приказ Минобрнауки России от 14.10.2015 N 1147 (ред. от 11.01.2018) «Об утверждении Порядка приема на обучение по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры». Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_188408/](http://www.consultant.ru/document/cons_doc_LAW_188408/).
4. Математическое программирование: учебное пособие. 5-е изд., стереотип / В.Г. Карманов. М.: Физматлит, 2004. 264 с.
5. Karmarkar N.A. New polynomial-time algorithm for linear programming // *Combinatorica*. 1984. V. 4. N 4. P. 373-395.
6. Gomory R.E. Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem / 50 Years of Integer Programming 1958-2008 // Jünger M. et al. (eds). Springer, Berlin, Heidelberg. P. 77-103.
7. Balas E., Zemel E. An algorithm for Large 0-1 Knapsack Problems // *Operations Research*. 1980. V. 28. N 5. P. 1130-1154.
8. Свами М., Тхуласираман К., Графы, сети и алгоритмы. М.: Мир, 1984. 454 с.
9. Шкаберин В.А., Подвесовский А.Г., Азарченков А.А., Коростелев Д.А., Трубаков А.О. Особенности проектирования визуального интерфейса для web-сайта «БГТУ-абитуриент» // *Вестник Брянского государственного технического университета*. 2017. №1 (54). С. 185-191.

Сведения об авторах:

**Коростелёв Дмитрий Александрович**

к.т.н., доцент, доцент кафедры «Информатика и программное обеспечение»,  
Брянский государственный технический университет,  
тел.: (4832)56-09-84, +7-906-699-48-57,  
E-mail: [nigm85@mail.ru](mailto:nigm85@mail.ru)

**Зимин Сергей Николаевич**

к.т.н., старший преподаватель кафедры «Информатика и программное обеспечение»,  
Брянский государственный технический университет,  
тел.: (4832)56-09-84, +7-910-337-23-57,  
E-mail: [wmailzsn@gmail.com](mailto:wmailzsn@gmail.com)

References:

1. Federal Law No. 273-FZ of December 29, 2012 (Ed. 07.03.2018, amend. 07.03.2018) «On Education in the Russian Federation» (with amendments and additions, entered into force on 07.03. 2018). Available at: <http://docs.cntd.ru/document/556716692>. [in Russian language]
2. Official statistics. Population. Demography. Natural movement of the population. Available at: [http://www.gks.ru/wps/wcm/connect/rosstat\\_main/rosstat/ru/statistics/population/demography/](http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/population/demography/). [in Russian language]
3. Order of the Ministry of Education and Science of Russia from 14.10.2015 N 1147 (Edited on 11.01.2018) «On the approval of the procedure for admission to study on educational programs of higher education - undergraduate programs, specialty programs, master's programs». Available at: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_188408/](http://www.consultant.ru/document/cons_doc_LAW_188408/). [in Russian language]
4. Karmanov V.G. (2004). *Mathematical Programming: A Tutorial. The stereotype*. Moscow: Fizmatlit. [in Russian language]
5. Karmarkar N.A. (1984). New polynomial-time algorithm for linear programming. *Combinatorica*. Vol. 4, (4), pp. 373-395.[in Russian language]
6. Jünger M. (Ed.), Gomory R.E. (2008). *Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem. 50 Years of Integer Programming 1958-2008*. Springer, Berlin, Heidelberg, pp. 77-103.[in Russian language]
7. Balas E., Zemel E. (1980). An algorithm for Large 0-1 Knapsack Problems. *Operations Research*. Vol. 28, (5), pp. 1130-1154.[in Russian language]
8. Swami M., Thulasaraman K. (1984). *Graphs, networks and algorithms*. Moscow: Mir. [in Russian language]
9. Shkaberin V.A., Podvesovsky A.G., Azarchenkov A.A., Korostelev D.A., Trubakov A.O. (2017). Features of designing a visual interface for the web-site «BSTU-entrant». *Bulletin of Bryansk State Technical University*, 54(1), pp. 185-191.[in Russian language]

*Статья поступила в редколлегию 28.04.18.*

*Рецензент: д.т.н., доцент Брянского государственного технического университета  
Аверченков А.В.*

*Статья принята к публикации 07.09.18*

Information about authors:

**Korostelyov Dmitry Alexandrovich**

Ph.D., associate professor, associate professor of the department «Informatics and software»  
Bryansk State Technical University,  
tel.: (4832) 56-09-84, + 7-906-699-48-57,  
E-mail: [nigm85@mail.ru](mailto:nigm85@mail.ru)

**Zimin Sergey Nikolaevich**

Ph.D., senior lecturer of the department «Informatics and software»  
Bryansk State Technical University,  
tel.: (4832) 56-09-84, + 7-910-337-23-57,  
E-mail: [wmailzsn@gmail.com](mailto:wmailzsn@gmail.com)