Интернет-журнал «Мир науки» / World of Science. Pedagogy and psychology https://mir-nauki.com

2018, №3, Том 6 / 2018, No 3, Vol 6 https://mir-nauki.com/issue-3-2018.html

URL статьи: https://mir-nauki.com/PDF/56PDMN318.pdf

Статья поступила в редакцию 06.06.2018; опубликована 27.07.2018

Ссылка для цитирования этой статьи:
Окишев С.В. Проблема создания и использования генераторов и решателей математических задач // Интернет-журнал «Мир науки», 2018 №3, https://mir-nauki.com/PDF/56PDMN318.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ.

For citation:
Okishev S.V. (2018). The problem of creating and using generators and solvers of mathematical exercises. *World of Science. Pedagogy and psychology*, [online] 3(6). Available at: https://mir-nauki.com/PDF/56PDMN318.pdf (in Russian)

УДК 372.851

ГРНТИ 28.23.25

Окишев Сергей Владимирович
ФГБОУ ВПО «Омский государственный университет путей сообщения», Омск, Россия
Кандидат технических наук, доцент
E-mail: okishev59@mail.ru
РИНЦ: http://elibrary.ru/author_profile.asp?id=687885

Проблема создания и использования генераторов и решателей математических задач

Аннотация. Генераторы и решатели математических задач создаются и используются в высшем образовании уже несколько десятилетий. Они стали основой систем компьютерной математики и комплексов дистанционного обучения. Автором представлено краткое описание основных направлений деятельности по созданию генераторов и связанных с ней технологических, воспитательных и социальных проблем. Генерация карточек контрольных работ для традиционного обучения во многом отличается от создания генератора клонов задач для электронного учебника или генерации тестов в контролирующих системах аттестации обучающихся. Предлагаемые другими исследователями классификации генераторов носят, как правило, глобальный характер и основаны на различиях шаблонов формул от систем искусственного интеллекта или от абстрактных исчислений на базе теории алгоритмов. Автор сделал попытку классифицировать простейшие генераторы математических задач по характеру и последовательности выполняемого построения, не претендуя на полноту такой классификации. В статье выдвигается идея генератора как творческого настольного инструмента преподавателя-энтузиаста. Таким генераторам не требуется штамповать тысячи задач в секунду, но требуется «очеловеченное» построение задач под контролем квалифицированного методиста. Генератор не должен заменять преподавателя при создании задач. Он должен только помочь преподавателю в этом своими графическими построениями и рутинными вычислениями. При таком подходе скорость построения задач все равно увеличивается в несколько раз по сравнению с ручной разработкой, качество задач оказывается выше, чем при ручном построении и сохраняется непосредственный визуальный контроль за каждой созданной задачей.

Ключевые слова: генератор; решатель; задача; алгоритм; математика; студент; преподаватель; проблема; карточка; контрольная работа; тест; программное обеспечение; язык программирования

Введение

Генераторы и решатели математических задач создаются и используются в высшем образовании уже несколько десятилетий. Они стали основой компьютерной математики и комплексов дистанционного обучения, столь популярных в настоящее время. При традиционном обучении преподаватели многих вузов создают программы автоматической генерации вариантов контрольных работ и типовых расчетов по математическим дисциплинам. Первой и основной целью статьи является краткое описание главных направлений создания генераторов с попутной формулировкой возникающих в образовательной среде проблем технологического, воспитательного и социального плана. Второй целью публикации является попытка классификации простейших генераторов математических задач по характеру и последовательности выполняемых построений. Автор не претендует на полноту подобной классификации. Третья цель статьи – сформулировать подход к генератору как к творческому настольному инструменту преподавателя-энтузиаста, позволяющему создавать небольшие по объему качественные наборы задач для традиционного обучения математике. Автор формулирует требования к такому генератору, отличные от общепринятых (быстродействие, дешевизна базового программного обеспечения (ПО), совместимость отдельных частей ПО, возможность применения дистанционных технологий и так далее).

Генераторы и решатели

В начале исследования полезно сформулировать основные рассматриваемые понятия.

Ключевое понятие данной статьи – понятие генератора. Большинство современных авторов не формулируют в своих работах понятие генератора, а описывают сложные технологии или программные комплексы, в которых генераторы (и решатели) являются лишь встроенными элементами продвинутых технологий. Определения генератора имеются в явном виде в работах В.В. Кручинина [1, 2], но они являются недостаточно универсальными. Наиболее близкими к идеалу представляются определения из исследований И.А. Посова [3, 4]. В автореферате [4] своей диссертации Илья Александрович Посов предложил следующее определение: «Под генератором понимается программный модуль, результатом работы которого является условие и ответ к некоторому заданию». Автор статьи будет понимать под генератором математических задач программу, получающую в автоматическом режиме условия и ответы таких задач, либо создающую такие задачи в автоматизированном режиме диалога с преподавателем.

Обоснованием необходимости создания генераторов задач некоторые авторы [5, 6, 7] считают проблему «списывания», которой придают слишком большое значение. Бороться с этой проблемой они предполагают исключительно «технически», штампуя сотни и тысячи неповторяющихся вариантов задач и контрольных работ. Автор совершенно не согласен с подобным подходом. Обучающиеся давно перешли от технологий срисовывания решений у соседа к технологиям использования сотовой связи и Интернета. Борьба со «списыванием» предполагает разнообразные меры в первую очередь организационного и воспитательного характера. Во-первых, не следует выдавать контрольные работы «на дом». Выдаваемые для домашнего выполнения задания следует считать лишь тренировочными вариантами для студентов, и не начислять за них больших баллов в рейтинг. Во-вторых, после выдачи группе вариантов контрольной работы не следует уходить из аудитории на кафедру и «путешествовать» по Интернету. Подобное поведение преподавателей побуждает студентов к нечестному выполнению работ. При воспитательных беседах со студентами следует объяснить им, что «списывание» является разновидностью коррупции (получения незаконными способами благ) и недопустимо в правовом государстве. Для многих молодых людей важным

56PDMN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

является понятие престижа, поэтому стоит намекнуть им, что поведение с получением результатов решения задач со стороны характерно для «лузеров», «аутсайдеров» и понижает их престиж до самой низкой отметки. Следует предусмотреть и меры наказания студентов, пытающихся (несмотря на строгий контроль) заниматься списыванием. Возможно введение системы «желтых» и «красных» карточек (как в футболе). Получение красной карточки может означать для студента невозможность переписать последующие контрольные работы на более высокую оценку, отказ преподавателя от индивидуальных консультаций таких студентов или более строгий прием экзамена. Многолетний педагогический опыт автора показывает, что уже в первом семестре удается полностью ликвидировать «списывание», и в дальнейшем эта проблема более не возникает. Таким образом, более подходящим обоснованием необходимости разработки генераторов может служить «обновление наборов типовых расчетов, задач для проведения контрольных и самостоятельных работ», как справедливо отмечено в статье [8].

Под решателем математических задач обычно понимается ПО, предназначенное для автоматического решения таких задач. На вход решателю поступают краткие описания задач в определенном формате, на выходе получаются решения (ответы). Например, системы компьютерной математики MathCAD, MatLab, Mathematica являются, по сути, интегрированными решателями математических задач. В современной образовательной среде ушли в прошлое попытки создания универсального решателя задач (General Problem Solver). Разрабатываются решатели под конкретную учебную дисциплину или даже под конкретное учебно-методическое пособие. К сожалению, понятие решателя имеет и другое толкование. Решателями именуют себя группы специалистов, готовых в online-режиме выполнять за нерадивых студентов и школьников задания по различным дисциплинам, в первую очередь по математике. По сути дела, такая деятельность является преступной, хотя и не подпадает под статьи уголовных законов. Кроме того, в Интернете предлагается выполнение курсовых и дипломных работ, рефератов и контрольных работ по вполне доступным ценам. Возникает новая социальная проблема. С одной стороны, нагромождаются все более совершенные системы генерации заданий и компьютерного тестирования, а с другой – совершенствуются средства борьбы с этими системами с помощью современных технологий. В этой усложняющейся борьбе почти не остается места тому, что априори необходимо: изучению студентом соответствующей математической дисциплины, формированию тех самых знаний, умений и навыков, которые прописаны в образовательных программах.

Интересен вопрос о соотношении между генераторами и решателями задач. Очевидно, что решатель, как правило, не требует генерации заданий. Условия задачи в заданном формате просто вводятся в него пользователем. Считать решатель обязательной частью генератора также неверно. Вполне возможны генераторы задач без встроенного решателя. Таковыми являются генераторы задач по шаблонам, в которых через набор переменных «записаны» как исходные формулировки задач, так и получаемые формулы ответов. При подстановке конкретных значений переменных получаются сразу как постановка задачи, так и ответ без каких-либо процедур решения. Пример простейшего генератора задач без решателя описан в работе [9]. Генерировались системы линейных алгебраических уравнений (СЛАУ) с квадратной матрицей коэффициентов. Для генерации задавалось решение системы в виде набора целых чисел. Матрица коэффициентов СЛАУ генерировалась случайным образом по специальному алгоритму, после чего свободные члены вычислялись как суммы левых частей системы. При этом отсутствовал этап решения СЛАУ каким-либо алгоритмом. Таким образом, в предложенном генераторе решатель отсутствует. Более сложен вопрос комбинирования генераторов и решателей при создании электронных учебников [7] или компьютерных тестирующих систем [10]. Организация диалога с обучающимся (или поэтапного контроля выполнения тестовых заданий) требует наличия не только ответов к задачам, но и пошаговых решений задач. Поэтому решатели используются здесь внутри разветвленной схемы диалога

неоднократно. Вначале же, разумеется, должен выполнить свои функции генератор задания или теста. Из-за того, что в большинстве случаев генераторы включают в себя и решатели задач, большинство авторов не упоминают о решателях вообще. Часть авторов подчеркивает это единство использованием дефиса: генераторы-решатели математических задач.

История и основные направления разработки генераторов

Программы-генераторы возникли как одно из направлений использования компьютеров в организации учебного процесса. В 60-е и 70-е годы XX века ЭВМ представлялась устройством решения вычислительных задач, а студентов в основном обучали языкам программирования и численным методам. С появлением и развитием в 80-х годах персональных компьютеров ЭВМ стала представляться обучающим устройством для демонстрации алгоритмов и иллюстрации изучаемых дисциплин. Первые «персональные» компьютеры (например, типов БК или «Кедр») были весьма примитивны, и преподавателю приходилось думать, как можно сделать хоть что-то с помощью такого оборудования. Компьютеры типа «Ямаха» с цветным дисплеем породили массу методик обучения школьников и студентов рисованию и созданию игровых программ с цветной графикой, не улучшая обучение основным дисциплинам. Обучающие и демонстрирующие программы получили свое развитие еще на базе больших машин серии ЕС ЭВМ. В 1985 году автор был на длительной стажировке в киевском Институте Кибернетики и присутствовал на многочисленных лабораторных занятиях студентов по «Методам оптимизации и исследованию операций». Уже тогда преподавателями были реализованы демонстрационные программы, наглядно показывающие на экранах терминалов изучаемые алгоритмы (симплекс-метод, метод потенциалов, построение критического пути).

Отдельное направление создания генераторов задач возникло где-то на рубеже 80-х и 90-х годов XX века. Преподаватели многих вузов стали пытаться автоматизировать процесс составления типовых задач, в первую очередь – по математике. Автор статьи был в числе этих «первопроходцев», но его разработки [11, 12] не получили должного развития. Здесь следует обрисовать огромную технологическую проблему, помешавшую первому поколению программ-генераторов стать настоящим инструментом работы преподавателей. Проблема эта – примерно 15-летнее отставание компьютерной техники в СССР (а далее – и в России) от мирового уровня. В конце 80-х годов программное обеспечение кандидатских и докторских диссертаций часто представляло собой огромные колоды перфокарт, которые сам диссертант набивал на перфораторе, за невозможностью использовать вспомогательный персонал. В начале 90-х годов персональные компьютеры в периферийных вузах исчислялись единицами, и добиться работы на них было нереально. Например, на нашей кафедре первый персональный компьютер появился лишь в конце 90-х и использовался в основном для оформления документации. Несколько облегчали положение домашние компьютеры преподавателей. В начале 90-х они имели уже систему MS DOS, надстроенную NORTON-коммандером, и неплохие версии языка BASIC. Вот на этом-то обеспечении и создавались первые программы-генераторы. А мир в это время уже пользовался системой WINDOWS, современными языками программирования и Интернетом! Естественно, что внедрение новых операционных систем и языков программирования «поставило крест» на первом поколении генераторов. Наиболее упорные сторонники прежних технологий создавали целые системы генераторов задач, таких как система генераторов Карнаухов-Commander, описываемая в статье [13] в 2011 году. Большинство разработчиков, тем не менее, перешли на более высокие уровни осмысления проблемы и ее реализации.

В начале XXI века в отечественной научно-педагогической среде оформилось само понятие генератора, возникли первые классификации генераторов, анализировались подходы к

56PDMN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

построению таких программ. Можно выделить несколько направлений этой деятельности. Первое направление – это создание наборов задач и карточек контрольных работ для традиционного обучения [14, 15, 16, 17]. Автор статьи относит себя именно к этому направлению. Следует заметить, что цель автоматического обновления вариантов контрольных работ и экзаменационных билетов, формулируемая представителями направления, является не столь актуальной. Содержание классических математических дисциплин: «Алгебры», «Геометрии», «Математического анализа», «Теории вероятностей» остается стабильным в течение многих десятилетий. На рубеже веков автором данной статьи был разработан цикл контрольных работ по указанным дисциплинам [18]. Часть задач была сгенерирована на компьютере, но значительно большая часть составлялась творчески вручную. Каждая контрольная работа имела 100 и более вариантов. Работая на потоке, имеющем примерно 40 студентов, автор использует ежегодно лишь 50 вариантов каждой контрольной работы. На следующий год используются вторые 50 вариантов. Студенты не имеют возможности сфотографировать варианты и не знают, какой именно номер достанется им на очередной работе из-за перемешивания вариантов внутри текущего списка. В итоге создание подпольной базы решенных вариантов становится бессмысленной задачей. Таким образом, одни и те же варианты контрольной могут успешно использоваться десятки лет. При оформлении экзаменационных билетов в них следует указать просто тему задачи (задание) без конкретных данных. Данные задач выдаются преподавателем на экзамене из заранее заготовленных генератором наборов, но сами-то эти наборы достаточно создать один раз! Большое разнообразие задач экзамена и большой объем наборов задач у экзаменатора не позволяют студентам «отслеживать» решения. Повторная выдача той же самой задачи может состояться через 7-8 лет. Важным, на взгляд автора, является лишь обновление задач типовых расчетов, выдаваемых на дом. Такие задания следует обновлять ежегодно, но и это обновление не решает проблему. Вновь генерируемым задачам студенты успешно могут противопоставить обращение через Интернет к экспертам-решателям, которые выполнят за них работу качественно и в срок. О социальных и технологических проблемах первого направления речь пойдет также в отдельном разделе.

Второе направление считает своей целью создание специальных языков и систем разработки генераторов и распространение построенных генераторов через Интернет среди российских преподавателей [4, 19]. При активном создании и использовании генераторов такой подход перспективен. Разрабатывая свои первые программы-генераторы на языке QUICK BASIC, автор затрачивал на создание, отладку и интерфейс очередного генератора примерно три месяца. Таким образом, в год удавалось создавать не более трех-четырех генераторов, каждый из которых был способен порождать лишь одну задачу, пусть и с разнообразными версиями. Наличие языка разработки генераторов позволяет ускорить и частично автоматизировать процесс. Возможность получить готовые генераторы задач через Интернет также можно считать большим подарком для преподавателей. Проблема состоит в том, что многие из преподавателей не нуждаются в таком подарке и не сумеют им качественно воспользоваться.

Третье направление использует генераторы как вспомогательное оборудование при разработке кадровых компьютерных учебных программ (КУП) или при создании электронных учебников [7, 20]. В данном направлении происходит переход от собственно генераторов задач к программам самоподготовки студентов, имеющим встроенные генераторы и решатели. В рамках направления формулируется ряд важных свойств генераторов [1]. Мощность генератора – это общее количество различных задач, которые может породить генератор. Семантическое расстояние между порожденными задачами должно обеспечивать существенное различие формулировок задач и их ответов. Управление сложностью позволяет в рамках программы-генератора порождать более или менее сложные задания по одной и той же теме. Указанные

свойства обеспечивают при каждом входе студента в обучающую программу новые версии стандартных заданий, а также настройку электронного учебника на уровень знаний студента, обнаруживаемый в процессе диалога с ним. Не со всеми положениями сторонников этого направления автор согласен. Важным достижением считается, например, устранение человека-преподавателя от процесса обучения. Здесь, по сути, все ставится с ног на голову. Квалифицированным преподавателем считается тот, кто может качественно обслуживать компьютерную систему, загружая ее адекватным наполнением. Гордиться подобными «достижениями» может разве что системный программист, никогда в жизни не занимавшийся реальной преподавательской деятельностью. Кстати, создание ярких, сложных и малополезных компьютерных презентаций не может считаться таковой деятельностью.

Четвертое направление сосредоточилось на создании компьютерных тестирующих систем (КТС) и компьютерных систем обучения (КСО) [10, 21]. При этом в последних двух направлениях зачастую генерируются уже не столько математические задачи, сколько разнообразные тесты, в том числе текстовые, графические или логические. В работе [21] предложены многочисленные критерии оценки качества генератора тестов. Часть из этих критериев может быть применена и к более простым генераторам задач. Рассматриваются вопросы мощности КТС (интегрированного понятия, в которое в качестве составляющей входит и мощность генераторов, используемых системой). Подвергнуты критике старые системы создания компьютерных тестов, например, популярная система дистанционного обучения MOODLE. В целом направление можно охарактеризовать как поднявшееся на качественно новый уровень дистанционное обучение.

Генераторы карточек контрольных работ и наборов математических задач

Работа многочисленных преподавателей по созданию систем генераторов математических задач привела к некоторому общему пониманию сути реализуемого процесса. Основой подхода является создание программируемых (алгоритмических) генераторов. При этом необходимыми оказались три основных элемента ПО. Первым таким элементом выступает некоторая система компьютерной математики или «продвинутый» калькулятор. Расточительно каждый раз программировать самому алгоритм решения математической задачи, когда эта проблема уже успешно решена в универсальных системах. Вторым элементом служит некоторый специальный язык разработки генераторов. Системы компьютерной математики – это лишь решатели задач. Порождение условий задач, а также процессы формирования вариантов контрольных работ, должны выполняться собственно в генераторе. Кроме того, алгоритмы многих специальных задач не реализованы в системах компьютерной математики и должны создаваться заново разработчиками генераторов. Наконец, результаты работы должны быть выданы в текстовой и графической форме и распечатаны на карточках. С этой целью применяются языки разметки, наиболее популярным из которых в настоящее время является LaTeX. Однако и в языке разметки могут отсутствовать многие необходимые символы или возможность изображения сложных графических структур. Эти составляющие системы генераторов приходится разрабатывать вручную. Кроме того, необходимо состыковывать и увязывать в единое целое все три вышеперечисленных элемента ПО. Таким образом, создание системы генераторов представляет собой сложную проблему. Решить такую проблему способен только преподаватель, являющийся одновременно программистом высокого уровня. Большинству преподавателей, являющихся лишь пользователями компьютера, эта задача не по плечу.

Особенности разработчиков генераторов породили и особое виденье ими проблем генерации. Частично они осознают, конечно, своеобразие проблемы. Например, в статье [22] по поводу генераторов отмечается, что «для большинства преподавателей требуется не какое-

либо сверхсовременное программное обеспечение, на освоение которого будет потрачено немало времени и сил, а нечто весьма простое и функциональное, легкое в освоении и применении». Затем в той же статье описываются критерии выбора преподавателя: цена, доступность, простота, инструкции на русском языке, широта функционала, возможность расширения функционала. Из этого описания становится ясно, что под преподавателем понимается преподаватель-программист, а под выбором – выбор компонент исходной «триады» для последующей разработки систем генераторов. Но далеко не все преподаватели таковы. Здесь мы сталкиваемся с огромной социальной проблемой невостребованности передовых разработок (как программных, так и методических) в преподавательской среде.

Множество преподавателей можно условно разбить на три основных группы. Первую составляют немногочисленные преподаватели-виртуозы, работающие в рамках традиционных образовательных технологий и не понимающие, зачем вводить разнообразные новшества, лишь ухудшающие учебный процесс. Преподавателей этого типа отличает качественное чтение лекций с использованием доски, способность к построению на ней сложных картинок, актерская импровизация, тщательная подготовка практических занятий дома с разработкой своеобразных нестандартных задач, авторские находки в преподавании многих тем, резко отличающиеся от классических учебников. К сожалению, количество таких преподавателей постоянно уменьшается в процессе смены поколений. Вторую группу составляют преподаватели, активно применяющие современные технологии (и к месту, и не к месту) и пытающиеся решить проблемы обучения через компьютер. В этой постепенно увеличивающейся группе преподаватели-программисты, являющиеся разработчиками генераторов, образуют небольшую изолированную «элиту». Наиболее крупную группу составляют в настоящее время преподаватели-исполнители, заботящиеся в первую очередь о сохранении своего рабочего места, доходов и о формальном выполнении своих обязанностей. Такие преподаватели могут быть в числе первых при заполнении кафедральной документации или даже при написании научных статей, но в учебном процессе они участвуют «по минимуму», отрабатывая полученные деньги. Они не заинтересованы в реальных знаниях студентов, но заинтересованы в достаточно высоком формальном рейтинге этих студентов, который постоянно завышают. Некоторые из таких преподавателей умудряются договариваться со студентами и отменяют значительную часть аудиторных занятий «за ненадобностью». В условиях ослабленного контроля за ходом учебного процесса студенты и без занятий получат требуемые оценки, а преподаватель – свои деньги. Эта же группа порождает и коррупционную составляющую, выявляемую вновь и вновь в преподавательской среде.

Предположим, что некоторый преподаватель-программист разработал набор генераторов задач и предлагает его в качестве подарка сотрудникам своей кафедры. Какова будет их реакция? Преподаватель-виртуоз, узнав, что генерируемые задачи носят «шаблонный» характер, откажется от генератора, заявив, что у него уже имеются гораздо лучшие задачи, а при необходимости он разработает новые и без генератора. Преподаватель-компьютерщик заявит, что в любой момент найдет в Интернете какие угодно задачи в любом количестве и продолжит готовить презентации, глоссарии и тесты. Преподаватель-исполнитель после продолжительной с ним беседы заявит разработчику генератора следующее: «Ну ладно, сгенерируй мне по сто задач по этим трем темам и положи на стол». От него вряд ли можно ожидать чего-то иного при его потребительском отношении к жизни. Таким образом, колоссальная по объему и сложности работа создателя генераторов, описываемая, например, в работах [23, 24], проваливается в никуда, и такой «сценарий» достаточно распространен.

У разработчиков генераторов задач для традиционного обучения могут возникнуть и другие проблемы, например, технологического плана. В статье [16] описывается генерация задач по теме «Неопределенный интеграл». В качестве системы компьютерной математики для

создаваемого генератора был выбрана система MatLab. Этот выбор был обусловлен жесткими требованиями администрации вуза. Лицензия стоила больших денег, и повсеместное использование приобретенного ПО было необходимо. Вообще-то задачи на неопределенные интегралы замечательно генерируются по шаблонам, которые можно взять из таблиц интегралов либо легко разработать самостоятельно. Неудачный же выбор системы компьютерной математики привел к крупным проблемам. К ним относились:

1. непривычная для России «англоязычная» запись математических формул;
2. применение в ответах гиперболических функций вместо экспонент;
3. комплексные решения при нахождении действительных интегралов;
4. неверная запись математических операций (нарушение порядка действий).

Указанные недостатки свели практическую ценность разработки к нулю. Разработчикам пришлось больше бороться с недостатками выбранного решателя, чем строить задачи на интегралы. Эта борьба состояла в массовой замене неверных математических выражений заново конструируемыми формулами. В процессе исправления сгенерированных задач и их ответов использовались традиционные «шаблоны». Применялись также процедуры редактирования формул.

Классификация генераторов

Прежде чем предлагать свою классификацию генераторов, приведем в качестве примеров две классификации подходов к генерации, имеющие общетеоретический характер. В статье Ильи Александровича Посова [3] предлагается следующая классификация технологий создания генераторов задач:

1. программируемые генераторы, основа которых – программирование алгоритма решения математической задачи;
2. параметрические генераторы, представляющие собой текст со вставками, в которые следует поместить случайные числа;
3. предметно-ориентированные генераторы, построенные на основе готовой универсальной процедуры, основанной на базе знаний и редакторе генераторов;
4. выбирающие генераторы, которые производят выбор фрагментов задач из подготовленных заранее баз фрагментов и их «склеивание» в формулировку очередной задачи.

При этом наиболее мощной считается первая технология, и для облегчения работы по созданию программируемых генераторов предлагается разработка специального языка создания генераторов. Автор в целом солидарен с этой позицией.

В работе [21] предложена классификация методов генерации тестовых заданий, имеющая «глобальный» характер. Эта классификация используется для сравнения методов генерации. Различаются следующие подходы:

1. методы хранимых и параметризированных шаблонов;
2. метод множественных генераторов (библиотека программируемых генераторов);
3. метод графа знаний и обучающего кластера;
4. метод генераций на основе онтологий (анализ динамики работы студента с ПО);
5. метод нейронных сетей (обучаемый искусственный интеллект генератора);

6. метод на основе формальных исчислений (например, исчислений Э. Поста).

Автор статьи предлагает классификацию генераторов задач по особенностям устройства самого алгоритма, реализующего такую генерацию. При этом можно выделить следующие типы алгоритмов генерации:

1. генерация условия задачи, а затем решение ее некоторым алгоритмом;
2. одновременная генерация условия задачи и ее ответа;
3. генерация ответа задачи, а затем генерация задач с данным ответом;
4. генерация «ядра» задачи с получением из него условий и ответа как следствий;
5. итерационное построение задачи в виде усложняющейся структуры с пересчетом текущего ответа;
6. построение задачи путем отбрасывания части условий из избыточной системы условий, а затем решение полученной задачи;
7. комбинирование задачи из нескольких частей, получаемых из наборов-банков этих частей, с последующим решением;
8. построение системы заданий с ветвящейся логической структурой (гиперзадача);
9. системы задач, реализуемые в виде процесса с переходами в разные состояния.

Первый тип генерации является классическим. К примеру, традиционная программа-решатель реализует ввод условия задачи вместо его генерации, а затем работает некоторый (численный или аналитический) метод решения задачи. В дисциплине «Алгебра» по первому типу может быть устроен генератор задач на вычисление определителей. Сначала происходит генерация определителя заданного порядка. Затем происходит вычисление определителя, скажем, методом Гаусса и устранение возможных погрешностей, так как учебные задачи на определители традиционно целочисленные.

Второй тип генерации соответствует применению шаблонов. При этом вместо параметров подставляются в формулы конкретные числа и составляются требуемые коэффициенты. Конечно, требуется предусмотреть сокращение дробей и прочие «мелочи», но сам подход выглядит очевидным. Таким способом удобно генерировать задачи на неопределенные интегралы из «Математического анализа».

Третий тип генерации «раскручивает» задачу от получаемого в начале ответа. При этом возможно получить несколько различных задач с одним и тем же ответом. Ясно, что таких «родственных» задач должно быть в общем наборе немного. Решатель при указанном способе генерации отсутствует. Пример алгоритма данного типа, порождающего задачи на решение СЛАУ, был разобран в начале статьи.

Четвертый тип генерации характерен для задач с четко выраженным математическим «ядром». Например, для многочлена характерным является набор его корней, являющийся «ядром» проблемы. Но в задаче может потребоваться не сам многочлен. Например, это может быть характеристический многочлен матрицы, а в итоге должны быть получены семейства собственных векторов. При этом должна быть сгенерирована исходная матрица (условие задачи) и найдены некоторым алгоритмом ее собственные векторы (ответ). То же самое «ядро» порождает другие условия и ответ, если речь идет о генерации задач на однородные линейные дифференциальные уравнения высокого порядка с постоянными коэффициентами. Здесь условием задачи является уравнение (по сути, видоизмененная запись многочлена), а ответом служит параметрическая формула, получаемая по «ядру».

56PDMN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

Пятый тип генерации предложен в работах В.А. Болотюка и Л.А. Болотюк. Например, в статье [24] приводятся изображения сгенерированных карточек контрольных работ по дискретной математике, где с помощью рекурсивного алгоритма рассматриваемого типа построены комбинаторная схема на расчет числа вариантов и задача на составление матрицы смежности графа. Специальная библиотека подпрограмм, разработанная авторами, позволяет строить сложные графические изображения к задачам. Аналогично можно генерировать, постепенно надстраивая графическую структуру и пересчитывая ответ, задачи на кратчайшие пути, задачи на критические сроки выполнения проектов и транспортные задачи оптимизации стоимости перевозок.

Шестой тип генерации рассмотрим на характерном примере. Во многих задачах на вычисление определенных и двойных интегралов, на классическую и графическую оптимизацию функций двух переменных требуется генерировать замкнутую область на плоскости, задаваемую системой ограничений. Задание заведомо избыточной системы ограничений, а затем случайное или упорядоченное удаление части из них позволяет получать на одном и том же исходном наборе ограничений различные плоские области. Сгенерированная область является «ядром» нескольких математических задач, к которому могут быть добавлены дополнительные записи, порождающие условие задачи конкретного вида. Решение задачи (получение ответа) происходит алгоритмически.

Седьмой тип генерации, или генератор-компилятор, работает с базой фрагментов задач и «собирает» из случайно выбранных фрагментов итоговую задачу, которая затем решается некоторым алгоритмом. Сама база фрагментов задач предварительно заполняется вспомогательными генераторами фрагментов. Предположим, что преподаватель желает составить самостоятельную работу на решение СЛАУ с помощью обратной матрицы. При этом он не хочет «мучить» студентов сложностями в алгоритме обращения и собирается использовать только матрицы с определителями ±1, ±2 и ±3. Преподаватель запускает вспомогательный генератор квадратных матриц с установкой заданного условия на ее определитель. При этом большинство сгенерированных матриц отбрасывается, а «годные» сохраняются в базе фрагментов. Затем произвольным образом заполняется база правых частей СЛАУ. Теперь можно запускать основной генератор-решатель матричного способа и готовить наборы карточек для самостоятельной работы.

Восьмой и девятый типы генерации относятся скорее к методам генерации взаимосвязанных систем задач и подсказаны автору статьями Л.А. Ашкинази. Например, в статье [25] предложен способ организации контрольной работы в виде единой «гиперзадачи» с условными переходами к следующим этапам контрольной, зависящими от результатов предыдущих этапов. Правда, и сам автор описываемой статьи называет такое построение «искусственным» и предлагает его лишь как еще одну версию генерации. Второй подход из работы [25] вообще относится к физическим задачам и использует изменяющееся состояние физического процесса для организации логических переходов между заданиями создаваемой контрольной работы.

Предложенный список типов алгоритмов генерации можно расширять и далее, по мере накопления новых идей и методов.

Генератор как рабочий инструмент преподавателя

Изложенная позиция автора по вопросам генерации задач ясна. Задачи он собирается применять на традиционных учебных занятиях, не нуждаясь ни в огромных наборах вариантов, ни в их постоянном обновлении. Генератор должен помочь ему при необходимости быстро и качественно создать небольшие наборы задач (не более 50-100 единиц) по конкретной теме

математической дисциплины. Поэтому и требования к генератору становятся иными. Скорее всего, это будет программа авторской разработки, без дополнительной покупки нестандартного лицензионного ПО. Программа эта должна генерировать задачи в режиме КУП под постоянным контролем со стороны преподавателя. Хотя некоторые процедуры отбрасывания негодных задач могут быть реализованы как автоматические, гораздо важнее оценка качества задачи самим преподавателем, для чего необходима графическая визуализация задачи и ее решения на экране и выдача вспомогательных числовых параметров, характеризующих «качество» построенного варианта. Кроме того, сами процессы генерации (например, множество используемых чисел и вероятности их появления) следует организовать так, чтобы частота появления «неудачных» задач была невелика: ведь каждую допустимую задачу преподаватель будет просматривать вручную! Первым проблемным моментом предлагаемого генератора является необходимость построения на экране весьма сложных графических изображений и рисования специальных символов. Здесь будет необходимо создавать целую систему или библиотеку подпрограмм экранной графики. Вторым проблемным моментом такого генератора является построение и распечатка карточек заданий. Возможно, что автору генератора придется повозиться с ручной обрезкой кадров-изображений и вставкой их в единый красочный шаблон варианта в виде рисунков. Это потребует значительных затрат времени, зато и карточки вариантов можно будет получить более качественные. Собственно, деятельность преподавателя состоит не в экономии времени, а в повышении качества выполняемой им работы и ее отдельных фрагментов.

Примером-иллюстрацией данного подхода к генерации задач может служить генератор задач на кривые второго порядка для дисциплины «Геометрия». Типовая задача на исследование и построение кривой имеет следующую формулировку: «По данному общему уравнению кривой второго порядка определить тип кривой, получить ее каноническое уравнение и построить кривую на плоскости». Генератор таких задач очень прост по устройству и именно поэтому приводится в статье в качестве примера. Во-первых, простота генератора заключается в отсутствии какого-либо решателя и использовании «пересчитываемых друг в друга» шаблонов канонического и общего уравнений кривой. Во-вторых, графическое изображение кривой второго порядка любого типа легко реализуется с помощью ее параметрических уравнений (кривая, как траектория движения точки).

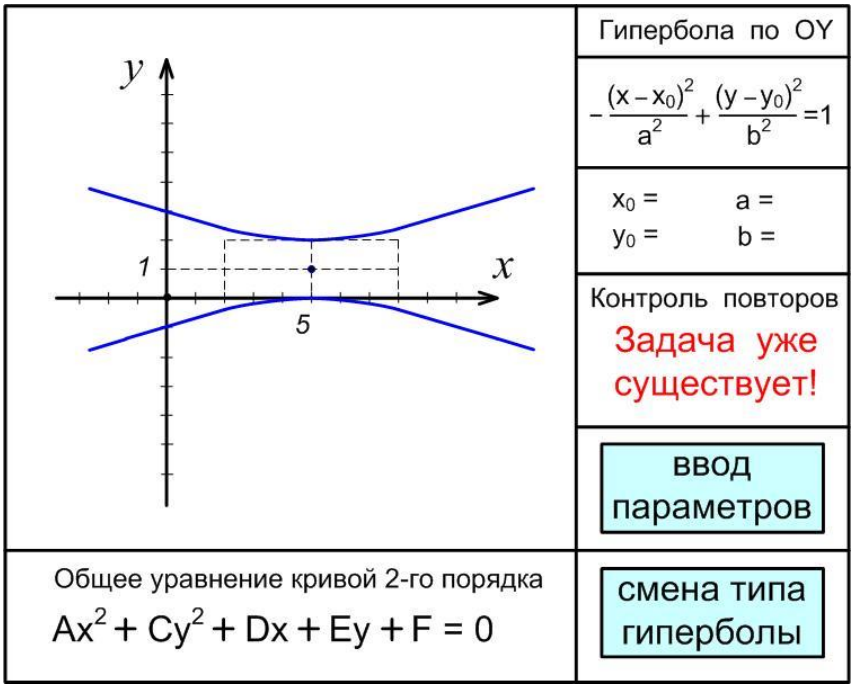

Рисунок. *Окно основного режима генерации (разработано автором)*

56PDMN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

Устройство предлагаемого генератора как КУП предполагает наличие предварительных графических меню выбора режима работы, основного окна-кадра, где происходит генерация и анализ полученной задачи, и графических окон для формулировки условия задачи и прорисовки ее ответа. Предположим, что в первом меню выбран гиперболический тип кривой, а во втором – гипербола с фокусами, расположенными на оси ординат. Тогда окно основного режима генерации соответствующих гипербол примет вид, схематически изображенный на рисунке. Для понимания общего смысла задачи у всех параметров и коэффициентов сохранены буквенные обозначения.

Автор считает, что не следует генерировать параметры кривой x_0, y_0, a, b случайным образом. Координаты центра кривой и размеры ее полуосей должен вводить сам преподаватель. При этом автоматически заполняется шаблон канонического уравнения (справа), прорисовывается сама кривая, и находятся коэффициенты общего уравнения (внизу). В специальных массивах генератор должен сохранять информацию о предыдущих сгенерированных за время сеанса гиперболах и при совпадении данных выдавать сообщение о повторной генерации задачи. Отбраковка задач будет происходить человеком по следующим не вполне строгим критериям (на основе предоставленной генератором графической и числовой информации):

1. большое удаление кривой от начала координат;
2. неудачная форма кривой (например, ее излишняя «вытянутость»);
3. неудачные коэффициенты общего уравнения (большие, нулевые, одинаковые);
4. задача уже есть в создаваемом наборе (подсказка-дополнение).

Сгенерировав достаточное количество гипербол указанного типа (не более 15-20 задач), пользователь вернется в меню гиперболического типа и выберет режим «гипербола с фокусами на оси абсцисс» или «пара пересекающихся прямых» и продолжит свою работу. Более сложные генераторы математических задач будут иметь свои особенности, но в целом схема работы с ними будет аналогичной.

Итоги

В статье проблема создания и использования генераторов математических задач рассмотрена во взаимодействии с сопутствующими технологическими, социальными и воспитательными проблемами. По мнению автора, лишь анализируя это взаимодействие, можно выбрать правильные пути создания и использования генераторов в каждом конкретном случае. Предложена классификация генераторов алгоритмического типа и авторское виденье генератора как инструмента автоматизированной творческой деятельности преподавателя по созданию методического обеспечения традиционных математических дисциплин.

56PDMN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

ЛИТЕРАТУРА

1. Борисов С.И., Кручинин В.В. Применение генераторов в компьютерных технологиях обучения // Интеграция образования. 2004. №4(37). С. 116-121.
2. Кручинин В.В., Магазинников Л.И., Морозова Ю.В. Модели и алгоритмы компьютерных самостоятельных работ на основе генерации тестовых заданий // Известия Томского политехнического университета. 2006. Т.309. №8. С. 258-263.
3. Посов И.А. Программирование генераторов задач // Компьютерные инструменты в образовании. 2010. №3. С. 19-31.
4. Посов И.А. Автоматизация процесса разработки и использования многовариантных учебных заданий: автореф. дис. … канд. пед. наук. Санкт-Петербург, 2012. 19 с.
5. Посов И.А. Обзор генераторов и методов генерации учебных заданий // Образовательные технологии и общество. 2014. Т.17. №4. С. 593-609.
6. Посов И.А. Автоматическая генерация задач // Компьютерные инструменты в образовании. 2007. №1. С. 54-62.
7. Левинская М.А. Автоматизированная генерация заданий по математике для контроля знаний учащихся // Образовательные технологии и общество. 2002. Т.5. №4. С. 214-221.
8. Болотюк В.А., Болотюк Л.А. Об опыте использования графического калькулятора HP 50g для разработки карточек с задачами по высшей математике // Мир науки. 2016. Т.4. №6. С. 5.
9. Окишев С.В., Ашаева П.А., Гельманова Е.А. Программа-генератор систем линейных алгебраических уравнений с варьируемыми вероятностями элементов // Информационные технологии: актуальные проблемы подготовки специалистов с учетом реализации требований ФГОС: Материалы пятой Всероссийской научно-методической конференции. Омский автобронетанковый инженерный институт. Омск, 2018. С. 267-271.
10. Сергушичева А.П., Швецов А.Н. Алгоритм реализации метода автоматической генерации тестовых заданий // Алгоритмы, методы и системы обработки данных. 2006. №11. С. 118-125.
11. Окишев С.В., Гателюк О.В. О генерации на компьютере типовых заданий по дисциплине «Высшая математика» для втуза // Математика в вузе: Труды Международной научно-методической конференции. Петербургский гос. у-нт путей сообщения. СПб, 1998. С. 172.
12. Гателюк О.В., Окишев С.В. Компьютерное составление задач по курсу высшей математики // Железнодорожный транспорт Сибири: проблемы и перспективы: Материалы межвузовской научно-технической конференции, посвященной 160-летию отечественных железных дорог и 100-летию железнодорожного образования в Сибири. Омский гос. у-нт путей сообщения. Омск, 1998. С. 133-134.
13. Карнаухов В.М. Компьютерные генераторы контрольных работ в преподавании математики // Природообустройство. 2011. №3. С. 105-110.
14. Болотюк В.А., Болотюк Л.А. Роль генераторов и решателей задач в преподавании высшей математики // Интернет-журнал Науковедение. 2013. №6(19). С. 173.

56PDMN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

15. Болотюк В.А., Болотюк Л.А. Об автоматизации процесса составления экзаменационных и тестовых материалов по высшей математике // Международный научный институт «Educatio». 2015. №10-1. С. 52-56.
16. Макарова И.Д., Макаров С.Е. Генерирование заданий по математике для студентов технических вузов в среде MATLAB // Актуальные проблемы преподавания математики в техническом вузе. 2017. №5. С. 79-84.
17. Коновалов Я.Ю., Соболев С.К. Методические аспекты компьютерного генерирования заданий по математике // Наука и образование: научное издание МГТУ имени Н.Э. Баумана. 2016. №7. С. 285-295.
18. Окишев С.В. Принципы организации цикла контрольных работ по дисциплине «Высшая математика» для студентов железнодорожных специальностей // Новые технологии – железнодорожному транспорту: подготовка специалистов, организация перевозочного процесса, эксплуатация технических средств: Сборник трудов с международным участием. Омский гос. у-нт путей сообщения. Омск, 2000. Ч.1. С. 44-45.
19. Зорин Ю.А., Посов И.А. Инструментальные системы построения и получения многовариантных тестовых заданий // Компьютерные инструменты в образовании. 2014. №1. С. 14-25.
20. Кручинин В.В. Концептуальные модели компьютерных учебных программ // Вестник Томского государственного педагогического университета. 2005. №7. С. 138-143.
21. Швецов А.Н., Сергушичева А.П. Опыт применения метода автоматической генерации тестовых заданий // Образовательные технологии и общество. 2017. Т.20. №4. С. 318-333.
22. Болотюк, В.А., Болотюк Л.А. О применении HOMELISP в процессе обучения математики // Интернет-журнал Науковедение. 2015. Т.7. №5(30). С. 175.
23. Болотюк, В.А., Болотюк Л.А. HOMELISP – инструмент для разработки генераторов и решателей задач // Актуальные проблемы преподавания математики в техническом вузе. 2015. №3. С. 27-33.
24. Болотюк, В.А., Болотюк Л.А. Использование графического калькулятора для разработки генераторов карточек с задачами по высшей математике // Сборники конференций НИЦ Социосфера. 2016. №45. С. 22-26.
25. Ашкинази Л.А., Гришкина М.П. Генератор задач по физике // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2006. №7. С. 207-208.

Okishev Sergey Vladimirovich
Omsk state transport university, Omsk, Russia
E-mail: okishev59@mail.ru

The problem of creating and using generators and solvers of mathematical exercises

Abstract. Generators and solvers of mathematical problems have been created and used in higher education for several decades. They became the basis of computer mathematics and distance learning systems. The author presents a brief description of the main activities for the creation of generators and related technological, educational and social problems. The generation of test cards for traditional training is much different from the creation of a clone generator for an electronic textbook or the generation of tests in controlling systems of attestation of students. The classifications of generators offered by other researchers are, as a rule, global in nature and are based on differences in the patterns of formulas from artificial intelligence systems or from abstract calculi based on the theory of algorithms. The author made an attempt to classify the simplest generators of mathematical problems by the nature and sequence of the construction performed, without claiming for completeness of such classification. In the article the idea of the generator as a creative desktop tool of an enthusiastic teacher is put forward. Such generators do not need to stamp thousands of tasks per second, but it requires "humanized" task construction under the supervision of a qualified methodologist. The generator should not replace the teacher when creating tasks. He should only help the teacher in this with his graphical constructs and routine calculations. With this approach, the speed of building tasks still increases several times compared to manual development, the quality of tasks is higher than with manual construction and direct visual control of each created task is preserved.

Keywords: generator; solver; exercise; algorithm; mathematics; student; teacher; problem; card; control work; test; software; programming language