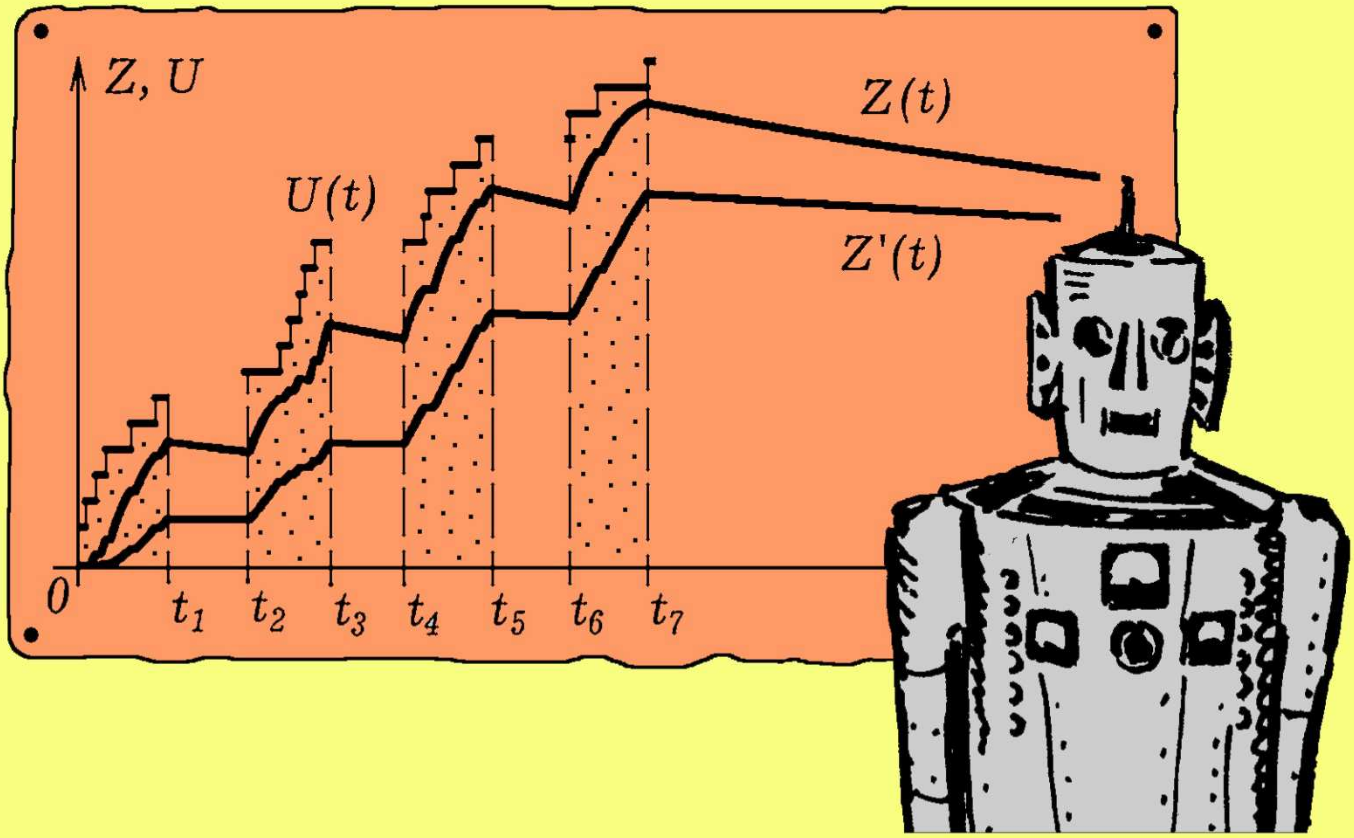
Р.В. Майер

КИБЕРНЕТИЧЕСКАЯ ПЕДАГОГИКА



имитационное моделирование процесса обучения

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ГЛАЗОВСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ
имени В. Г. Короленко»

Р. В. Майер

КИБЕРНЕТИЧЕСКАЯ ПЕДАГОГИКА: ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ОБУЧЕНИЯ

Монография

Глазов
ГГПИ
2014

УДК 37.02
ББК 32.81
М14

М14 ***Майер Р. В.***

Кибернетическая педагогика: имитационное моделирование процесса обучения: монография. – Глазов: Глазов. гос. пед. ин-т, 2014. – 141 с.

Рецензенты:
Ю. А. Сауров, доктор педагогических наук, профессор кафедры физики и методики обучения физике ВятГГУ, профессор, член-корреспондент РАО
В. А. Саранин, доктор физико-математических наук, профессор кафедры физики и дидактики физики ГГПИ

ISBN 978-5-93008-176-3

Монография посвящена проблеме исследования дидактических систем методом имитационного моделирования. В ней процесс обучения рассмотрен с позиций информационно-кибернетического моделирования, проанализированы различные дискретные и непрерывные модели системы «ученик – учитель», получены графики, описывающие динамику изменения уровня знаний среднестатистического ученика, с помощью имитационных моделей изучены различные ситуации, возникающие в процессе обучения. Рассмотрены варианты решения оптимизационной задачи обучения, проанализированы результаты. В книге приведены тексты более 30 программ на языке Pascal.

Книга предназначена для ученых и работников образования, интересующихся проблемами кибернетической педагогики и использованием математических и компьютерных моделей для анализа дидактических систем.

ISBN 978-5-93008-176-3
УДК 37.02
ББК 32.81

Введение

Истина слишком сложна; нам же дано
постичь лишь приближение к ней.
Джон фон Нейман

В нашей стране практически каждый человек учился, учится или будет учиться в общеобразовательной школе, среднем или высшем профессиональном учебном заведении. Государство тратит немалые средства на финансирование системы образования, поэтому **проблема повышения эффективности процесса обучения является актуальной**. Его оптимизация требует не только совершенствования содержания и методики изучения отдельных предметов, но и разработки теоретических основ дидактики с привлечением как гуманитарных (психология), так и точных наук (математика, кибернетика). В настоящее время получил распространение так называемый **информационно-кибернетический подход** к анализу учебного процесса, основанный на рассмотрении системы «учитель – ученик» с точки зрения теории управления. Возник и развивается новый раздел педагогики – **кибернетическая педагогика**.

Кибернетикой называют науку об управлении сложными техническими, биологическими и социальными системами, которые способны воспринимать, хранить и обрабатывать информацию. С точки зрения кибернетической педагогики процессы обучения и воспитания могут быть сведены к управлению развитием различных качеств личности учащихся с помощью целенаправленных и согласованных воздействий со стороны учителя и родителей. Цель обучения состоит в передаче учащимся совокупности знаний, в формировании умений и навыков, развитии у них способностей наблюдать, размышлять и эффективно взаимодействовать с окружающим миром. Перечислим **основные направления кибернетической педагогики** [12; 16; 36]:

1. Анализ педагогической системы с точки зрения связей управления и информационных потоков, которыми обмениваются управляющая и управляемая подсистемы.

2. Оптимизация процесса обучения, нахождение таких форм и методов организации учебного процесса, при которых функционирование системы образования было бы наиболее эффективным, то есть при наименьших затратах приносило бы максимальную пользу.

3. Практическое использование электронных устройств и автоматизированных обучающих систем для управления процессом обучения и тестирования; программированное обучение.

Среди современных методов исследования педагогических систем особое положение занимают методы математического и имитационного (компьютерного) моделирования. Их сущность состоит в том, что реальная педагогическая система заменяется абстрактной моделью, – некоторым идеализированным объектом, который имеет наиболее существенные свойства изучаемой системы. При этом исследуется поведение модели с помощью **математических методов** [11; 16] и путем **компьютерной имитации** [14; 44; 48]. Последнее означает создание компьютерной программы, которая ведет себя подобно системе «**учитель – учащиеся**» и проведение серии экспериментов при различных параметрах, начальных условиях и внешних воздействиях. Высокое быстродействие современных ЭВМ позволяет обрабатывать большие объемы информации и достаточно быстро осуществлять компьютерную имитацию. Изменяя начальные данные и параметры модели, можно исследовать пути развития системы, определить ее состояние в конце обучения. В этом состоит преимущество данного подхода по сравнению с методом качественного анализа. В некоторых случаях используют **мультиагентное моделирование**, при котором каждый учащийся заменяется программным агентом, функционирующим независимо от других агентов [10]. Для получения статистически значимых результатов используют **метод статистических испытаний** [47; 54]. Он состоит в многократной (более 10 000) реализации исследуемого процесса и подсчете числа различных исходов с последующим вычислением среднего арифметического, среднего квадратического отклонения, изучением характера распределения.

Сформулируем **основную задачу имитационного моделирования процесса обучения**: зная параметры учащихся, характеристики используемых методов и учебную программу (распределение учебной информации), определить уровень знаний (сформированности навыка) у учащихся в конце обучения. Также может быть решена **оптимизационная задача**: найти распределение учебного материала, уровень требований учителя, длительности занятий, при которых уровень знаний учащихся в конце обучения достигнет заданного значения, а сам процесс обучения будет удовлетворять наложенным на него ограничениям.

Основная **идея исследования** состоит в том, что метод имитационного моделирования действительно имеет смысл использовать для изучения дидактических систем, так как он позволяет проанализировать процесс обучения, выявить его особенности, установить связь между уровнем знаний учащихся в конце обучения, распределением учебной информации и параметрами ученика, помогает наметить пути оптимизации обучения.

Методологической основой настоящего исследования являются идеи Ж. Пиаже, Дж. Брунера, Л. С. Выготского, П. Я. Гальперина, Ю. К. Бабанского, В. Л. Матросова, И. Я. Лернера, В. М. Монахова, Л. В. Занкова, Д. Б. Эльконина, В. В. Давыдова, С. Л. Рубинштейна, М. Н. Скаткина, Л. М. Фридмана (педагогическая психология), Торндайка, О. Зельца, К. Дункера, Грино, Найта, Линдсея, Норманна, В. Кёлера, О. К.Тихомирова (проблема решения задачи), Д. Пойа, А. В. Хуторского (эвристическое обучение), А. Пуанкаре, Ж. Адамара, С. И. Шапиро (психология математического творчества), Н. Винера, К. Шеннона, Ф. Розенблатта, А. Н. Колмогорова, В. М. Глушкова, Д. А. Поспелова, И. Р. Пригожина (кибернетика, теория информации, синергетика), Р. Аткинсона, Г. Бауэра, О. Г. Гохмана, Л. Б. Ительсона, Э. Кротерса, Л. П. Леонтьева, В. В. Майера, Д. А. Новикова, Ф. С. Робертса (математическое моделирование обучения), Б. Скиннера, Н. Краудера, Л. Б. Ительсона, С. И. Архангельского, В. П. Беспалько, Е. И. Машбица, В. Е. Фирстова, В. С. Аванесова, И. В. Роберт (кибернетический подход в педагогике, программированное обучение и автоматизированные обучающие системы). Большое влияние оказала математическая модель дидактического переходного процесса, предложенная В. В. Майером [42, с 129–142].

Автор понимает, что рассмотренные в настоящей работе компьютерные модели, конечно же, не позволяют выработать рекомендации повышения эффективности обучения в том или ином конкретные случае. Однако они дополняют качественные рассуждения, делают их более объективными и обоснованными, могут быть использованы тогда, когда проведение педагогического эксперимента неправомерно или приводит к отрицательным результатам. Логичность и формализованность, воспроизводимость и конкретность получающихся выводов выгодно отличает метод имитационного моделирования от «метода качественных рассуждений». Придет время, когда в учебник педагогики наравне с качественными моделями войдут некоторые математические формулы и графики, полученные как результат компьютерной имитации.

Автор Р. В. Майер

Глава 1.
КИБЕРНЕТИЧЕСКИЙ ПОДХОД К ПРОЦЕССУ ОБУЧЕНИЯ И РЕШЕНИЮ ЗАДАЧ

В настоящее время большое распространение получили кибернетические методы исследования. Кибернетика – наука об управлении сложными системами (устройствами или организмами) и их сообществами. В этой главе рассмотрены принципы кибернетической педагогики, проанализированы различные подходы к проблеме информационно-кибернетического моделирования дидактической системы «ученик – учитель» и различные модели мыслительной деятельности человека.

1.1. Кибернетические принципы функционирования дидактической системы

Обучение состоит в передаче знаний или формировании навыков решения определенного класса задач. Основная цель заключается в приобретении умений успешного взаимодействия с окружающей средой. Кибернетическая педагогика рассматривает процесс обучения с позиций общей теории управления; в ее основе лежит системный подход. Как правило, **дидактическая система состоит из ученика и учителя**; роль учителя выполняет ЭВМ или человек, который сообщает полезную информацию, находит и исправляет ошибки, стимулирует работу ученика. Также возможно **самообучение** или обучение без учителя, при котором ученик самостоятельно изучает тот или иной вопрос и стимулирует свою деятельность.

Известно, что управление, т. е. целенаправленное изменение объекта (ученика), возможно, когда сформулирована цель управления, существуют канал сбора информации о состоянии среды и объекта, канал воздействия на объект и способ управления, позволяющий, учитывая информацию о состоянии объекта и среды, достичь поставленной цели. Кибернетическое подход предполагает анализ структуры системы управления, выявление прямых и обратных связей, установление информационных потоков. Рассмотрим основные принципы кибернетики [46] применительно к дидактическим системам:

Принцип разнообразия: управляющая система должна иметь большее разнообразие, чем разнообразие управляемой системы. Чтобы учитель имел возможность изменять свое состояние и поведение в ответ на изменение состояния ученика, он должен быть устроен сложнее, иметь большее число «внутренних состояний». В противном случае он не сможет осуществлять управление его деятельностью и правильно реагировать на изменение ситуации. Вместо термина «разнообразие» можно использовать «сложность». Из этого принципа следует, что увеличение сложности или разнообразия знаний учащегося требует повышения сложности знаний учителя и используемых методов обучения. Если разнообразие методов учителя меньше некоторого минимума, то он не сможет эффективно управлять деятельностью ученика. Понятно, что увеличение сложности управляемой подсистемы (ученика) должно сопровождаться увеличением сложности управляющей подсистемы (учителя).

Принцип целостности (или эмерджентности): свойства системы не сводятся к сумме свойств ее отдельных элементов, а зависят от ее структуры. У. Эшби показал, что «чем больше система и чем больше различия в размерах между частью и целым, тем выше вероятность того, что свойства целого могут сильно отличаться от свойств частей» [46]. Поэтому при моделировании системы обучения следует учитывать взаимосвязи между элементами. Знания учителя и ученика, содержание учебника, методы обучения, дидактическая система «учитель – ученик», вся система образования отвечают принципу целостности.

Принцип внешнего дополнения: «любая система управления нуждается в «черном ящике» – определенных резервах, с помощью которых компенсируются неучтенные воздействия внешней и внутренней среды» [46]. Иными словами, управление большой системой требует корректировки управляющих сигналов, которые следуют из теоретической модели. Их можно рассматривать как сигналы некоторого воображаемого «черного ящика», находящегося между системой управления и объектом управления.

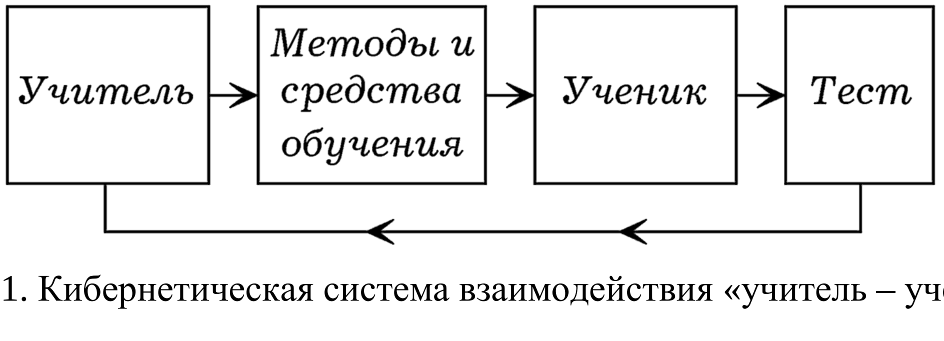


Рис 1.1. Кибернетическая система взаимодействия «учитель – ученик»

Принцип обратной связи: чтобы система могла адаптироваться к изменениям состояния объекта и внешним воздействиям, необходимо наличие канала обратной связи, по которому передается информация о состоянии объекта. При обучении обратная связь реализуется при общении учителя с учащимися, наблюдении за их деятельностью на уроке, в процессе анализа результатов устного или письменного опроса, тестирования, самостоятельных, контрольных работ и т. д. (рис. 1.1).

Принцип декомпозиции и иерархии управления: управляемый объект можно рассматривать как систему, состоящую из относительно независимых друг от друга подсистем, между которыми имеется определенная субординация. Например, ученик выполняет указания учителя, который подчиняется завучу, тот подчиняется директору, который, в свою очередь, подчиняется отделу образования и т. д.

Принцип активного самодвижения, обусловленного регулярным воспроизведением маловероятных состояний элементов, подсистем или самоуправляемой системы в целом, происходящего за счет притока энергии извне [51]. При обучении уменьшается неопределенность знаний учащихся, то есть система в целом переходит в более упорядоченное состояние с меньшей энтропией за счет энергии внешней среды.

Принцип целеполагания и целеосуществления: функционирование любой кибернетической системы направлено на достижение некоторой цели, минимизации некоторой целевой функции при заданных ограничениях. В процессе обучения учитель стремится увеличить количество знаний учащихся при фиксированной продолжительности занятий так, чтобы оно соответствовало предъявляемым требованиям. Целеосуществление требует сопоставления полученных результатов с целеположенными и корректировки функционирования системы [51, с. 10–16].

При анализе процесса обучения имеет смысл использовать информационно-кибернетический подход еще и потому, что с развитием информационно-коммуникационных технологий широкое распространение получили персональные ЭВМ и другие кибернетические устройства. Они в зависимости от заложенного в них программного обеспечения способны сообщать учащимся учебную информацию (в текстовом, графическом, звуковом виде), задавать вопросы и оценивать правильность ответов, осуществлять управление их учебной деятельностью.

1.2. Ученик как вероятностный автомат

Под обучением в самом широком смысле будем понимать процесс выработки у обучаемого (человека, животного или машины) определенных реакций на внешние раздражители (сигналы) путем многократных воздействий и подкрепления со стороны «учителя» (человека или машины). Чтобы сформировать у учащегося навык (умение), необходимо: 1) обеспечить понимание производимых элементарных операций; 2) организовать многократное выполнение учащимся соответствующей последовательности действий, каждый раз «поощряя» и «наказывая» учащегося за правильно или неправильно выполненные операции. **При самообучении отсутствует внешняя корректировка ответа ученика, то есть это обучение без поощрения** или наказания. Дополнительная информация о правильности или неправильности действий ученика (его реакции на внешние раздражители) не сообщается.

При обсуждении этих вопросов следует помнить, что немаловажную роль в работе мозга играют случайные процессы. Поведение человека (учителя или ученика) может быть промоделировано с помощью дискретного устройства – **вероятностного автомата**. Алгоритм функционирования такого автомата удобно задать в виде стохастического графа – совокупности вершин, соединенных стрелками, которые соответствуют переходам из одного состояния в другое. На работу вероятностного автомата (ВА) влияет фактор случайности; при поступлении на вход того или иного сигнала ВА переходит в другое состояние с заданной вероятностью p_{ij}. Эти вероятности переходов образуют стохастическую матрицу, от которой зависит реакция ВА на входные сигналы.

Если автомат не обучен, то все элементы этой матрицы равны, то есть он выбирает каждую следующую операцию совершенно произвольно и после ее выполнения сравнивает свои действия с эталоном (учителем). Учитель подтверждает правильность выбора операции (то есть «поощряет»), или сообщает, что выбор сделан неверно («наказывает»), подсказывая, какую операцию следовало бы выбрать. Все это приводит к тому, что вероятность правильных переходов увеличивается, стремясь к 1, а вероятность неправильных – уменьшается, приближаясь к 0. В конце обучения автомат практически без ошибок выполняет требуемую последовательность операций: $O_1 \rightarrow O_2 \rightarrow \dots \rightarrow O_N$.

1.3. Нейросеть и ее обучение

Согласно нейрофизиологической теории **мозг** представляет собой **совокупность связанных между собой нервных клеток – нейронов**, каждая из которых может находиться в возбужденном и невозбужденном состояниях. Нейроны соединены между собой и образуют сложные нейронно-сетевые структуры, определяющие мыслительную деятельность человека [2; 32].

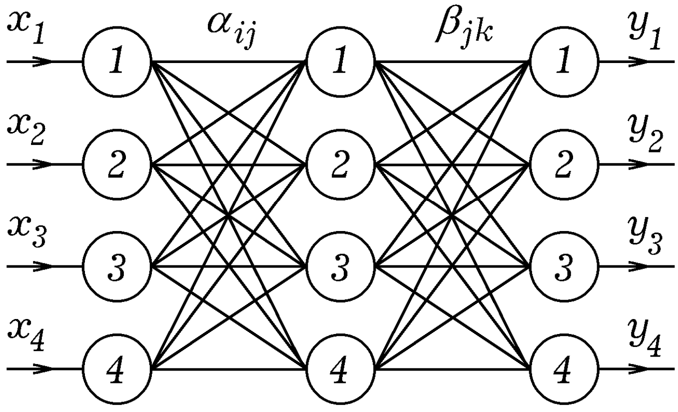


Рис. 1.2. Схема трехслойной нейросети

Представьте себе робота с электронной «нервной системой», состоящей из связанных между собой электронных блоков, моделирующих нейроны (рис. 1.2). На вход поступают сигналы x_i, с выхода снимаются сигналы y_j. Каждый нейрон – это автомат с n входами x_1, x_2, ..., x_n и одним выходом y, который характеризуется порогом Θ и весами ω_1, ω_2, …, ω_n. Состояние его выхода в момент $t+1$ зависит от входов в моменты t. На выходе $y=1$ в момент $t+1$ тогда, когда сумма всех весов возбужденных входов в момент t превышает порог срабатывания: $\omega_1 x_1 + \omega_2 x_2 + ... + \omega_n x_n > \Theta$; в противном случае $y = 0$. Если $\omega_i > 0$, то вход возбуждающий, а если $\omega_i < 0$, то вход тормозящий. Нейросеть состоит из множества нейронов, соединенных так, что выход одного после разветвления присоединяется к входам других нейронов, причем каждый вход соединен не более чем с одним выходом.

На результат работы нейросети влияют межнейронные коэффициенты α_{ij}, β_{jk}. **Обучение нейросети приводит к изменению коэффициентов межнейронных связей и к образованию новых связей.** При этом устанавливаются ассоциации между предъявляемым образом O и уже знакомыми образами O_1, O_2, ..., O_N; информация, которую запомнила нейросеть, распределяется по всем образующим ее нейронам. Новое понятие запоминается легче, если человек помнит другие похожие на него понятия и устанавливает связи между ними [2].

Нейросетевая модель мозга позволяет проиллюстрировать эффективность ассоциативной памяти. Допустим, имеются два слова «abc» и «def», составленные из букв алфавита: A = {a, b, c, d, e, f}. Для распознавания этих слов достаточно построить нейросеть с шестью входами x_1, x_2, ..., x_6, содержащую два нейрона N_1 и N_2 (рис. 1.3.1). Они изображены кружками, внутри которых указан порог срабатывания $\Theta = 2{,}9$. **Возбуждающие связи** с весами 1 изображены непрерывной линией, а **тормозящие** с весами –1 – пунктиром. Тогда при поступлении на вход вектора (1,1,1,0,0,0), соответствующего слову «abc», состояние выходов y_1 и y_2 таково: (1,0). Слову «def» отвечает вектор (0,0,0,1, 1,1), при этом на выходах нейросети появляется (0,1). Любые другие комбинации букв (ab, bcd, defa) не распознаются нейросетью, на ее выходах (0,0) [50].

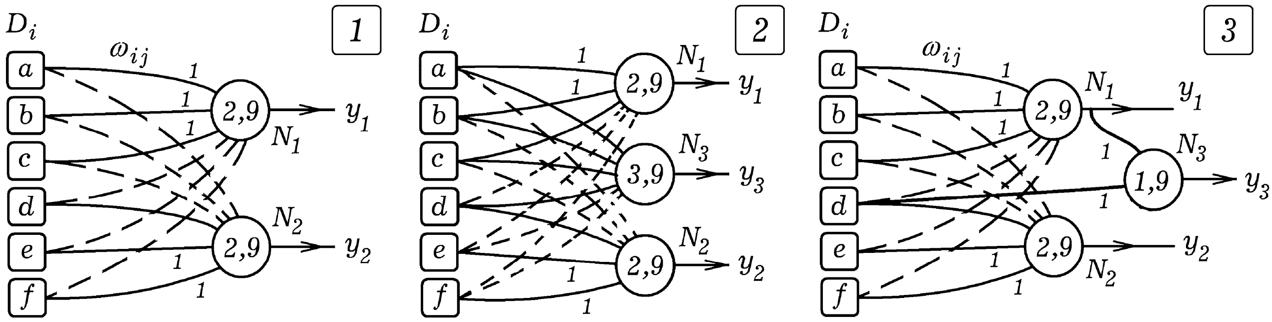


Рис. 1.3. Нейросетевая модель ассоциативной памяти

Допустим, что необходимо «научить» эту нейросеть, кроме слов «abc» и «def», узнавать слово «abcd». Эта задача может быть решена двумя способами. Например, можно добавить к существующим двум третий нейрон, который непосредственно контактирует с входами D_i $i = 1$, 2, …, 6. Чтобы нейросеть узнавала три слова «abc», «def», «abcd», необходимо добавить нейрон N_3 с $\Theta = 3{,}9$ и установить шесть новых связей (четыре возбуждающие и две тормозящие с весом –1) (рис. 1.3.2).

Другой способ состоит в том, что добавляемый нейрон N_3 связан выходами нейронов N_1 и N_2 или одним из них. Чтобы нейросеть различала слово «abcd», достаточно добавить нейрон N_3 с $\Theta = 1{,}9$, имеющий две возбуждающие связи с весами 1 (рис. 1.3.3). В данном случае, если на вход подать (1,1,1,1,0,0), что соответствует слову «abcd», то на выходах появится 1,1,0.

Нейросеть по-прежнему распознает слова «abc» и «def», все остальные варианты приводят к тому, что на ее выходах получаются (0,0,0). Понятно, что второй способ эффективнее первого: чтобы нейросеть приобрела новую способность узнавать объект «abcd», достаточно добавить меньшее количество связей.

Согласно **принципу экономии мышления** вероятнее всего реализуется такой способ мышления, который требует меньшего количества психических усилий. Поэтому человек, помнящий слова «abc» и «def», слово «abcd» запоминает как сочетание букв, отличающиеся от слова «abc» наличием четвертой буквы «d». В этом случае мы не учитывали последовательность букв в слове, хотя это тоже имеет большое значение.

Все эти рассуждения легко распространить и на объекты другой природы: физические тела, графические изображения, геометрические фигуры и т. д. Каждый объект имеет какие-то свойства, их наличие, различные сочетания и последовательности регистрируются входными нейронами D_i. Если человек запомнил и научился распознавать объекты O_1, O_2, …, O_n, то чтобы запомнить и узнавать новый объект O_{n+1}, ему следует найти самый похожий объект и выявить отличия [2; 8; 53]. Допустим, имеется объект O_k, обладающий качествами «a, b, c», в то время как новый объект O_{n+1} обладает качествами «a, b, c, d». Человеку проще запомнить объект O_{n+1}, обладающий такими же качествами, что и объект O_k, но отличающийся наличием качества «d». Если не устанавливать ассоциативные связи с известными уже образами, то потребуется создание большего числа связей, что менее эффективно. Чем больше связей требуется для вызова того или иного образа, тем менее надежна система.

1.4. Система обучения автомата как модель обучения человека

Обучение человека часто сводится к формированию умения решать задачи двух типов: 1) задачи, требующие выполнения жесткой последовательности действий и логических методов решения; 2) задачи, связанные с творческой деятельностью, требующие совершения интуитивных скачков, узнавания объектов, применения эвристических методов решения. При решении многих учебных задач присутствуются обе составляющие. Прочитав условие задачи, ученик относит ее к тому или иному классу, высказывает догадки, делает допущения и только потом решает ее, используя соответствующий алгоритм.

В связи с этим существуют два пути обучения людей и машин (ЭВМ, роботов и т. д.): 1) сообщение алгоритма решения задачи; 2) обучение на примерах. В первом случае учитель сообщает ученику жесткий алгоритм решения какой-то задачи, например, сборки технологического узла. В случае с ЭВМ такое обучение сводится к загрузке в ее память компьютерной программы, исполнение которой и позволит решить задачу. При обучении на примерах ученик (человек или ЭВМ) учится распознавать образы, то есть правильно классифицировать предъявляемые ему объекты. Он усваивает ограниченное число примеров, например, различные изображения букв «А, Б, В», запоминая их. После этого он распознает новые объекты (новые изображения этих букв), которые не предъявлялись в процессе обучения.

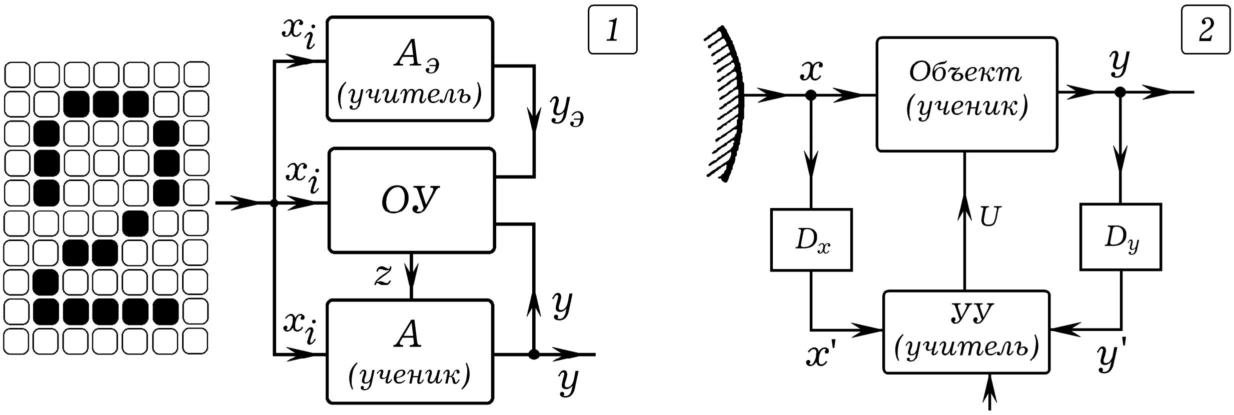


Рис. 1.4. Кибернетическая система обучения автомата (нейросети)

Кибернетическая система обучения на примерах автомата или нейросети хорошо известна [9; 46]. Допустим, имеется рецепторное поле (матрица оптодатчиков), эталонный автомат $A_Э$, обучаемый автомат A, обучающее устройство $ОУ$ (рис. 1.4.1). С целью обучения автоматам A и $A_Э$ путем воздействия на рецепторное поле предъявляются $k < n$ объектов, случайно выбранных из множества $X = \{ x_1,\ x_2,\ \dots,\ x_n \}$. Эталонный автомат $A_Э$ (машина, человек) указывает обучаемому автомату A, к какому классу относится каждый из k предъявленных объектов. Обучающее устройство ОУ сравнивает реакции $y_Э$ эталонного автомата $A_Э$ с реакциями y обучаемого автомата A и с помощью сигналов z изменяет его внутреннее состояние так, чтобы его реакции возможно чаще совпадали с реакциями $A_Э$. Затем автоматам предъявляется экзаменационная последовательность объектов x_{k1}, x_{k2}, …, x_{ks}. Если число ошибок автомата не превышает допустимый уровень, обучение закончено. Эффективность обучения зависит: 1) от того, какие именно характеристики подаются на вход автомата A, как они кодируются; 2) от числа возможных состояний автомата; 3) от алгоритма работы обучающего устройства. В качестве обучаемого устройства может быть использован вероятностный автомат (ВА) или нейросеть. В книге [43, с. 135] представлена похожая кибернетическая схема обучения (рис. 1.4.2), состоящая из среды, двух датчиков, управляющего устройства (учитель) и объекта управления (ученика).

В диссертации [52] при рассмотрении проблемы создания обучающей экспертной системы проанализирован следующий подход к моделированию процесса обучения. Допустим, уровень знаний ученика необходимо повысить от S' до S. Процесс обучения сводится к следующей последовательности итераций: $S' = S_0'$; $S_1' = S_0' \cup \Delta S_0'$, $S_2' = S_1' \cup \Delta S_1'$; …; $S_n' = S_{n-1}' \cup \Delta S_{n-1}' = S$. При этом на каждом i-ом шаге обучения актуальный уровень знаний S_i' обучаемого увеличивается за счет усвоения знаний зоны ближайшего развития $\Delta S_i'$, где $i = 0,1,2,\ldots,(n-1)$. Система «учитель – ученик» может быть промоделирована с помощью двух конечных автоматов A и A', задаваемых массивами входной и выходной информации, множеством внутренних состояний, функциями переходов и выходов. Они моделируют управляющую подсистему (учитель) и управляемую подсистему (ученик).

В. Е. Фирстов пишет: «Процесс обучения в данной модели происходит следующим образом. Пусть перед A' поставлен вопрос $z_0 \in Z$, ответ на который требует от ученика с уровнем знаний S'_0 в процессе обдумывания привлечения знаний зоны $\Delta S'_0$ и формирования ответа $a_1 \in Z'$, поступающего на вход автомата A. Ответ анализируется учителем, принимающим резолюцию $s_1 \in S$, которая переводит A в состояние $s_2 = f(s_1; a_1) \in S$ и формулирует следующий вопрос $z_1 = g(s_1; a_1) \in Z$. Далее описанный процесс аналогичным образом приводит к вопросу z_2 и, таким образом, происходит «освоение» зоны потенциального развития $\Delta S'_0$, затем $\Delta S'_1$ и т. д., т. е. автомат A реализует «обучение» A' с уровня S' до уровня S» [52].

1.5. Кибернетическая система учебного процесса

Как сказал У. Р. Эшби, кибернетика – «наука о том, как надо управлять очень сложной системой, чтобы в итоге она вела себя желательным для нас образом». **Основная задача кибернетической педагогики** состоит в выявлении принципов и способов эффективного управления учебным процессом, при котором минимальные затраты времени (усилий, денег) позволяют достичь требуемого уровня знаний учащихся. Решение этой проблемы требует построения абстрактной кибернетической системы учебного процесса, состоящей из множества взаимосвязанных объектов, участвующих в информационном обмене. Создание такой качественной модели позволяет осуществить математическое моделирование, а затем перейти к имитации на ЭВМ.

Построим кибернетическую систему учебного процесса (рис. 1.5). Она должна включать в себя абстрактные модели учителя, учеников и их родителей, способных воспринимать, запоминать, перерабатывать и обмениваться информацией. Сначала абстрагируемся от стохастического характера поведения перечисленных выше объектов и будем считать их детерминированными автоматами с большим числом внутренних состояний. В простейшем случае учитель моделируется автоматом, задаваемым двойкой $< P, A >$, где P – программа курса, A – алгоритм работы. Программа курса характеризуется множеством $\{\upsilon_1, \upsilon_2, \ldots, \upsilon_N\}$ из N вопросов (тем), их сложностью S_i и временем их изучения t_i. Модель ученика задается четверкой $< \alpha, \gamma, U, Z >$, где α – коэффициент научения, γ – коэффициент забывания ученика, U – уровень его притязаний из интервала [0; 1], пропорциональный оценке, на которую учащийся претендует, $Z = \{Z_1, Z_2, ..., Z_N\}$ – знания ученика. Будем считать, что Z_i – уровень знаний i -й темы, который лежит в интервале [0; 1] и равен вероятности правильного выполнения теста по данной теме. Модель родителя – воображаемый автомат, задаваемый двойкой $< V, W >$, где W – уровень притязаний родителя, V – возможность родителя оказать психологическое воздействие на своего ребенка и повысить уровень его притязаний U [27].

В процессе обучения учитель воздействует на учеников, передавая им учебную информацию и осуществляя текущий контроль (вопросы, тестирование). Учащиеся также воздействуют на учителя, сообщая, что им понятно или непонятно, задавая вопросы и выполняя задания текущего теста. Так возникает **первый замкнутый контур управления**. Учитель, видя реакцию учеников, может очень быстро (в течение урока) на нее реагировать: отвечать на вопросы, обращать внимание учащихся на их ошибки, помогать им их исправлять.

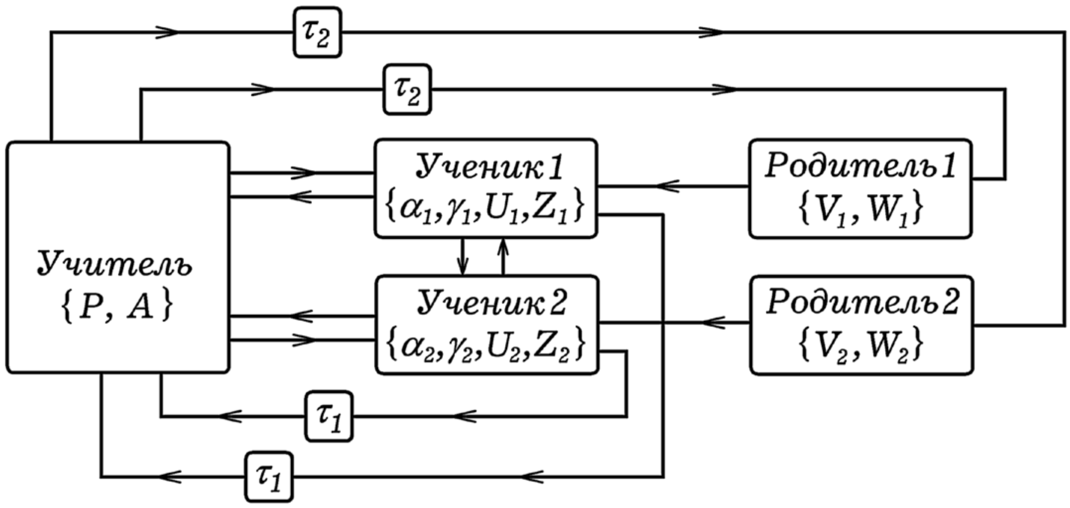


Рис. 1.5. Учебный процесс как кибернетическая система

В конце изучения темы учитель проводит контрольную работу, результаты которой также позволяют оценить уровень знаний учащихся и выбрать дальнейшую стратегию обучения: либо приступить к изучению новой темы, либо повторить изучение тех вопросов, которые были усвоены недостаточно хорошо. Это **второй замкнутый контур управления**. Он содержит элемент задержки, поэтому сигнал от учащегося приходит с запаздыванием на время τ_1 (несколько дней).

В случае, когда учитель видит, что учащийся плохо работает, он сообщает об этом родителям. Если успехи ребенка не устраивают родителя ($Z < W$), и тот имеет возможность воздействовать на ребенка (V достаточно велико), то он повышает мотивацию учащегося к обучению, увеличивая его параметр U. Это **третий замкнутый контур управления**. Он также содержит элемент задержки на время τ_2 (1–2 недели).

Можно усложнить систему, введя в нее новые элементы, например, директора школы, который контролирует работу учителя и результаты обучения, сопоставляя их с требуемым уровнем. При этом получится **четвертый замкнутый контур управления** (на рис. 1.5 он не изображен) [27].

Надо понимать, что в ряде случаев кибернетическая система управления претерпевает изменения. Например, в роли учителя может выступать компьютер с обучающей программой или подключенный через Интернет к тому или иному образовательному ресурсу. Роль родителей, повышающих мотивацию учащегося, может играть учитель, который проводит с ними воспитательную беседу, убеждает в необходимости ответственного отношения к учебе и т. д.

На основе кибернетического подхода может быть создана имитационная модель учебного процесса. Например, в статье [17] предложена модель оптимального управления процессом обучения. Он включает в себя: 1) кортеж $X =< X_1, X_2,... >$, параметры которого описывают обучаемого (психологические качества, компетентность и т. д.); 2) кортеж $Y =< Y_1, Y_2,... >$, характеризующий учебно-методические материалы и учебную программу (уровень абстракции, объем, структура и т. д.); 3) кортеж $Z =< Z_1, Z_2,.... >$, параметры которого описывают образовательные ресурсы (методические приемы, средства наглядности и т. д.). При моделировании контролируются уровень компетентности обучаемого $K(X,Y,Z)$ и время, требуемое для обучения $T(X,Y,Z)$. При этом решаются два вида задач оптимизации обучения: 1) «максимизировать уровень компетентности обучаемого $K(X,Y,Z)$ при ограничении времени на процесс обучения $T(X,Y,Z)$»; 2) «минимизировать время на процесс обучения $T(X,Y,Z)$ без потери качества овладения обучаемым компетенций $K(X, Y, Z)$ заданной программой обучения» [17].

В статье [10] предлагается мультиагентная модель процесса обучения Learning, состоящая из обучаемых агентов «Ученик», накапливающих знания, обучающего агента «Учитель», передающего знания обучаемому агенту и оценивающего степень их накопления, и объектного блока «Среда обучения». Имитационное моделирование позволило получить графики накопления знаний агентами в фазе активного обучения и изучить изменения эффективности накопления знания агентами в фазах активного и самостоятельного обучения. В работе [48] моделируется автоматизированное обучение с помощью взвешенных ориентированных графов. Известны и другие компьютерные модели обучения [7; 14; 15; 41; 43; 44].

1.6. Модели решения учебных задач

Существуют различные подходы к проблеме решения задачи [1; 2; 8; 53; 56]. Учебной задачей в самом широком смысле называется любое задание, которое получает учащийся от учителя с целью обучения. Несколько сузим это понятие, исключив творческие задания (написание сочинения или рисование картины), так как творческая деятельность плохо формализуется, и эти задания могут быть выполнены огромным числом различных способов. В результате останутся задачи, решение которых требует последовательного выполнения ограниченного числа операций в определенном порядке (решить уравнение, нарисовать график, заполнить таблицу, собрать электрическую цепь).

Чтобы решить задачу, школьнику следует: 1) определить тип задачи; 2) правильно выбрать алгоритм решения (он может быть не самым оптимальным, но все равно должен приводить к результату); 3) правильно выполнить все операции. К **задачам первого типа** будем относить те задачи, для решения которых учащийся использует алгоритмический подход. Например, нахождение корней квадратного уравнения школьником, который знает все необходимые формулы и последовательность действий (алгоритм решения). К **задачам второго типа** отнесем те задачи, алгоритм решения которых неизвестен, и человек вынужден применять метод перебора, эвристический метод, интуитивные рассуждения. Часто учебная задача для данного школьника частично является задачей первого типа, а частично – второго типа.

При самостоятельном решении задач основную роль играет замкнутая цепь управления, реализуемая в сознании учащегося. Учащемуся сообщают условия задачи, исходные данные и дают задание, что необходимо найти. При этом возможны два варианта: 1. Учащийся может убедиться в правильности своего ответа. Например, он решает уравнение и, найдя его корни, может путем подстановки убедиться в правильности решения. 2. Учащийся не может проверить правильность своего решения, ему не с чем сравнить полученный результат. При этом ему сложнее найти ошибку.

Далеко не всегда решение задачи логически следует из условия. В некоторых случаях учащийся должен выдвинуть гипотезу, которая никак не вытекает из имеющихся у него данных, сделать какое-то предположение, догадаться, совершить интуитивный скачок. Например, при решении ряда геометрических задач приходится делать дополнительные построения, при решении физических задач – чем-то пренебрегать, заменяя физические объекты их идеализированными моделями, и т. д. В чем-то учащийся становится похожим на ученого, которому тоже изначально не известен путь решения задачи. Как этот процесс может быть промоделирован на компьютере?

Рассмотрим известную аналогию между решением задачи и исследованием поверхности земли. Можно создать компьютерную программу, моделирующую движение исследователя при поиске пути к цели (рис. 1.6.1). Пусть в некоторой области О поверхности находится группа людей, которым нужно добраться до цели R. Область О хорошо изучена и окрашена в белый цвет. Поверхность неровная, где-то трудно проходимые болота Б, где-то горы и непреодолимые препятствия П. Один из людей высказывает предположение, что достичь цели R можно, двигаясь в направлении A какое-то время t_1 (то есть совершив N_1 шагов). Он начинает проверять свою гипотезу и совершает N_1 шагов в направлении A. Можно предусмотреть случайные отклонения от выбранного направления движения, но в среднем человек смещается в направлении A. Если при этом он не достигает цели R, то он возвращается обратно в О, сообщая людям о результатах своего путешествия, которые наносятся на карту. Точки поверхности, по которым прошел человек, становятся известны всем людям, и они ходят по ним без особого труда. Через некоторое время человек высказывает предположение, что для достижения цели необходимо из точки A уже разведанного пространства совершить N_2 шагов в направлении B. Он начинает проверять свою гипотезу, совершая шаги в направлении B. Но впереди непроходимое болото. Совершив N_2 шагов, он останавливается и возвращается обратно. Результаты своего путешествия он наносит на карту.

Этот алгоритм многократно повторяется. Каждый раз человек случайно выбирает известную ему точку разведанной поверхности, направление движения и количество шагов N (оно может увеличиваться с течением времени или изменяться случайно). С течением времени увеличивается белая область, соответствующая исследованной части поверхности, а черная часть (неизвестное) уменьшается. Наконец наступает момент, когда человек достигает цели R.

Понятно, что если цель R окружена трудно проходимым болотом или находится на высокой возвышенности, то достичь ее на данном этапе развития техники невозможно. Совершенно аналогично, если задача слишком сложная, например, надо вычислить интеграл, а школьник владеет только арифметическими действиями, то он не сможет ее решить на данном этапе своего развития.

При объяснении решения задачи учитель как бы показывает ученику путь из О в R. Можно представить муравья, движущегося по правильному пути, оставляющего феромоновый след, который со временем испаряется. Если ученик вовремя не повторит ход рассуждений (не пройдет по тому же пути), то след исчезнет, и ученик не вспомнит, как решается задача.

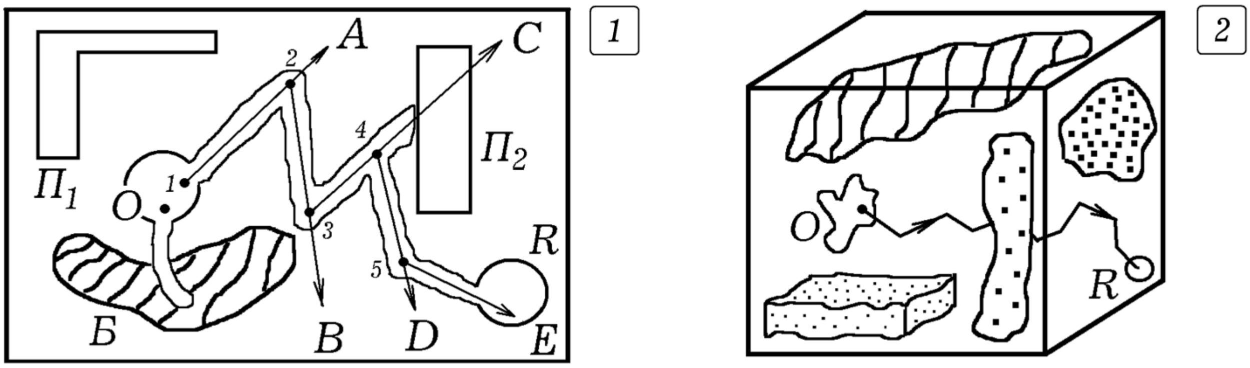


Рис. 1.6. Решение сложной задачи как поиск цели

Эта модель может быть расширена и распространена на s-мерное пространство, в котором каждая точка имеет s координат. Область О соответствует известному, а вдали имеется область R, которую необходимо достичь (рис. 1.6.2). В этом s-мерном пространстве также имеются трудно непроходимые области, которые могут быть заданы случайно. Из области О периодически выходит точка, которая движется в случайном направлении, а затем возвращается обратно. При моделировании на ЭВМ задача дискретизируется, то есть непрерывное пространство заменяется сеткой, ячейки которой имеют разную проходимость. Если проходимость некоторых отрезков сетки мала, то получаем задачу о поиске выхода из лабиринта или задачу о нахождении наикратчайшего пути между двумя вершинами сложного графа.

При решении задачи, требующей интуитивного скачка, человек выдвигает всевозможные идеи и проверяет их правильность. Опять воспользуемся аналогией с перемещением по поверхности, траекторию движения по которой можно закодировать символами: N – 100 шагов на север, S – 100 шагов на юг, W, E – 100 шагов на запад или восток. Допустим, правильный путь такой: NNWWWNEE. Компьютер случайным образом создает «слова», состоящие из 1, 2, 3... букв алфавита $\{N, S, W, E\}$. Каждое слово соответствует некоторому пути и является гипотезой, нуждающейся в проверке. Эта проверка может состоять в моделировании движения точки по поверхности из начального положения О или в сопоставлении случайным образом сгенерированного слова с правильным словом. Сопоставление начинается слева направо и продолжается до тех пор, пока не будет найдена ошибка. На каждое сравнение отводится 1 условная единица времени (УЕВ). Можно представить себе человека, который пытается пройти по пути NNWSNE и на четвертом шаге обнаруживает ошибку. После этого он выдвигает новую гипотезу и начинает ее снова проверять. Наконец наступает момент, когда сгенерированное слово приводит к цели.

1.7. Мыслительный процесс с точки зрения теории катастроф

Процессы обучения и понимания, вообще говоря, дискретны хотя бы потому, что человеку приходится оперировать отдельными знаками, идеями, теориями. Усмотрение способа решения задачи, усвоение отдельной идеи или теории происходят в результате последовательности скачков или переходных процессов. С точки зрения теории катастроф эти «скачкообразные изменения, возникающие в виде внезапного ответа на плавное изменение внешних условий», называются **катастрофами** [3, с. 8].

Состояние системы «учитель – ученик» будем характеризовать тремя величинами: $x =$ «уровень знаний», $y =$ «воздействие учителя», $z =$ «уровень понимания теории». При изучении новой теории, усвоении новых мыслей происходит скачок в ее понимании. На рис. 1.7.1 и 1.7.2 с разных ракурсов изображена **поверхность со сборкой из двух складок**, соответствующая зависимости уровня понимания теории (или некоторой идеи) от воздействия учителя и уровня знаний ученика. Если ученик обладает достаточным количеством знаний, то в результате воздействия учителя, старающегося привести его к некоторой мысли, он плавно переходит к новому уровню понимания теории (путь $1' \to 2'$). В случае, когда уровень знаний учащегося невысок, он движется по пути $1 \to 2 \to 3 \to 4$, проходящему через складку, и совершает качественный скачок на новый уровень понимания теории.

Воздействие учителя, неподкрепленное знаниями учащегося, также приводит к катастрофе – резкому переходу на новый уровень. Это сопровождается тем, что ученик приходит к пониманию новой мысли не в результате маленьких шажков, каждый из которых логически обусловлен. Понимание перескакивает через ряд важных операций, и новая мысль усваивается как догма. Пусть ученик А не усвоил, что называется функцией синус, а учитель требует от него усвоить, что $\sin(\pi/6) = 1/2$. Ученик А вынужден отнестись к этой истине как к догме, совершая переход $1 \to 2 \to 3 \to 4$, содержащий скачок $2 \to 3$. Другой ученик В понял, что такое функция синус и как определить синус угла с помощью тригонометрического круга. Поэтому он находит ответ на вопрос в результате последовательности осознанных действий $1' \to 2'$ [1; 56].

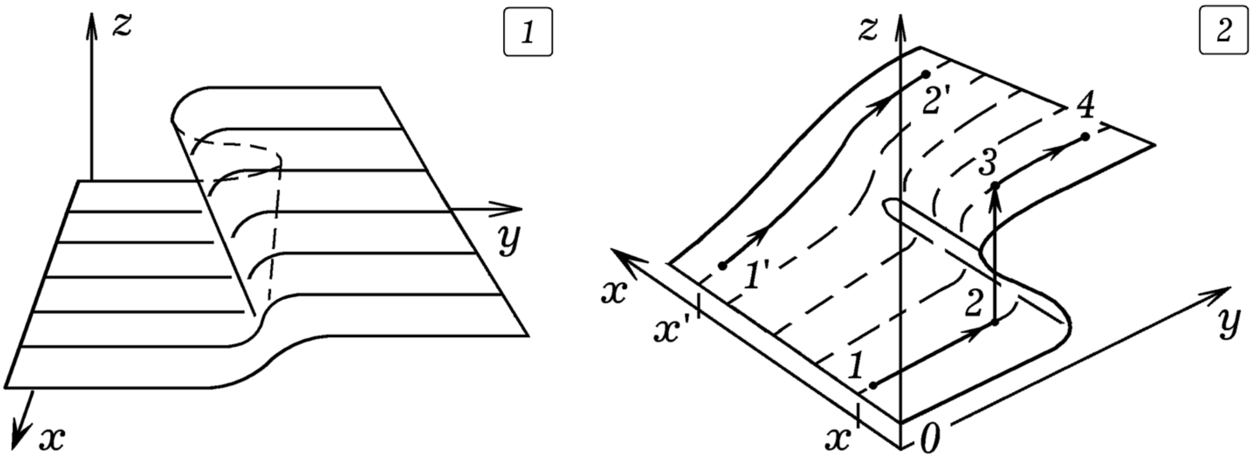


Рис. 1.7. Поверхность со сборкой из двух складок

В процессе изучения сложных математических или физических теорий ученик вынужден осуществлять последовательность логических рассуждений, переходя от одной мысли к другой [2]. При этом учитель «ведет» ученика от мысли 1 к мысли 2, затем к мысли 3 и т. д. Можно представить себе поверхность с углублениями, в одном из которых находится шарик. Когда внимание учащегося переключается с мысли 1 на мысль 2, шарик как бы перекатывается из первого углубления во второе. Переход от сложной мысли к простой соответствует «спуску» шарика по «нисходящей лестнице» (рис. 1.8.1) и происходит самопроизвольно или как результат небольших усилий ученика. Движение от простой мысли к сложной аналогично перемещению шарика по «восходящей лестнице» (рис. 1.8.2); оно возможно в случае, когда на шарик действует сила F, соответствующая воздействию учителя или волевым усилиям ученика. Если обобщить эти рассуждения, то получим искривленную поверхность (рис. 1.8.3), состоящую из нескольких желобов. Координата Z отвечает уровню знаний ученика, i – номеру идеи, а S – субъективной сложности идей. Чтобы учесть влияние случайных факторов, можно представить, что эта поверхность вибрирует, или на шарик действует хаотически изменяющаяся сила.

Если у ученика знаний Z_1 по данной теме немного, то переход от идеи 1 к идее 5 требует определенных усилий со стороны ученика и учителя. При объяснении шарик как бы перекатывается, взбираясь вверх по «восходящей лестнице». Если знаний Z_2 достаточно, то ученик сам без посторонней помощи может легко проделать переход 1–2–3–4–5. Возможны промежуточные варианты, когда учитель должен лишь немножко «подталкивать» учащегося [2].

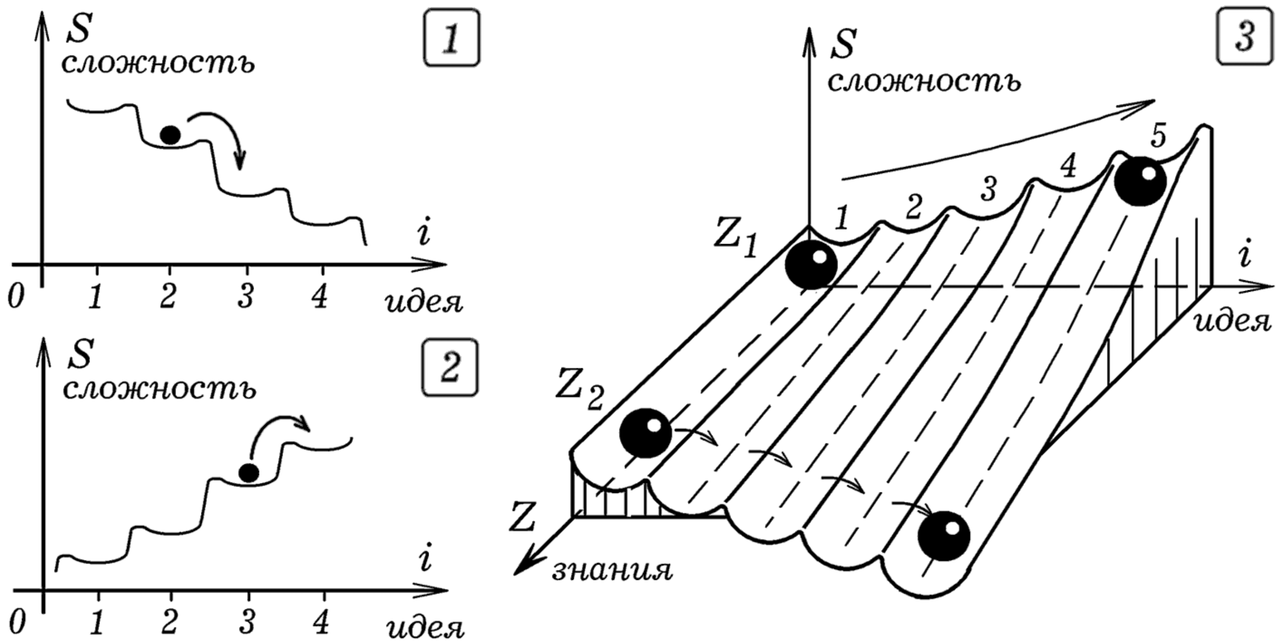


Рис. 1.8. Переход от одной мысли к другой в зависимости от знаний

1.8. Другие аналогии и модели мыслительной деятельности

Психологи утверждают, что появление новых гипотез происходит как результат «слепой вариации» исходных данных, структуры и параметров системы и «естественного отбора» новых идей. Пока задача не решена, человек на основе имеющейся информации не имеет возможности определить априори, какая гипотеза правильнее. Выдвигаются всевозможные гипотезы, которые могут быть проверены, отброшены или приняты. Этот принцип положен в основу **усилителя мыслительных способностей**, который был предложен У. Р. Эшби в середине XX века. Он состоит из генератора шума, преобразователя, блока отбора, блока управления и клапана. Преобразователь формирует различные случайные варианты объектов отбора, например, последовательности символов. В блоке отбора из генерированных вариантов выбираются те, которые соответствуют заданным критериям. Если сгенерированная последовательность символов соответствует критериям, блок управления открывает клапан и пропускает ее на вход следующего каскада усилителя [50, с. 97]. Установлено, что: 1) время решения задачи тем меньше, чем чаще человек вырабатывает гипотезы и быстрее их проверяет (число интеллектуальных действий в единицу времени – известный критерий креативности); 2) количество шагов, предлагаемых учащимся для решения задачи, не должно быть меньше минимального числа шагов, требующихся для решения задачи.

Оказывается, что **мозг ведет себя подобно микрочастице, находящейся в потенциальной яме**. Аналогом координаты является объем знаний, скорость и направление мыслей – аналог импульса микрочастицы. Мозг, как и микрочастица, находится в непрерывном движении. **Решение задачи аналогично прохождению микрочастицы через потенциальный барьер.** Если задача трудная, а знаний мало (потенциальный барьер высок, энергия частицы мала), то вероятность ее решения (преодоления потенциального барьера частицей) невелика. После решения задачи состояние мозга изменяется – человек начинает думать о чем-то другом, приобретает новые интеллектуальные умения, которыми не обладал до ее решения.

В процессе измерения микроскопическая система взаимодействует с измерительным прибором, и ее состояние изменяется. Аналогично и **при определении уровня знаний и других характеристик мозга происходит изменение его состояния**. При прохождении теста, содержащего достаточно трудные задания, человек чему-то учится, у него появляются новые мысли и т. д. Если учащемуся предложить простые задания, то состояние его мозга не изменится, но и оценить уровень знаний не удастся. Выпускник школы легко и безошибочно решит арифметические примеры за первый класс. При этом он ничему не научится, оценить его знания не удастся. Если тест содержит слишком сложные задания, то учащийся с ними не справится и тоже ничему не научится. В оптимальном случае тест должен состоять из последовательности задач, сложность которых постепенно нарастает. Это позволит выявить задачи, которые учащийся может решить, и тем самым оценить его уровень знаний.

Обучение людей не во всем похоже на «обучение» вычислительных машин, вероятностных автоматов, нейросетей. ЭВМ в отличие от человека: 1) обладает памятью, способной хранить информацию сколь угодно долго; 2) может быстро передавать информацию (компьютерную программу и данные) другим ЭВМ непосредственно или через некоторую базовую станцию. Поэтому процесс «обучения» в ряде случаев сводится к загрузке информации и занимает мало времени. Кроме того, ЭВМ не забывает полученные знания.

Иначе обстоят дела с электронно-механическим устройством (роботом), который обучается работать в некоторой новой для него среде. Понятно, что он должен быть достаточно «умным», а его «мозг» сложным, иметь некоторые начальные навыки к обучению. Оперируя различными объектами, робот может научиться их распознавать и правильно использовать. При обучении методом проб и ошибок учитель должен оценивать правильность выполнения каждой операции или последовательность операций. При самообучении робот должен сам «понимать» цель, к которой он стремится, самостоятельно оценивать правильность решения задачи.

Глава 2.
ДИСКРЕТНАЯ МОДЕЛЬ ФОРМИРОВАНИЯ НАВЫКА РЕШЕНИЯ ЗАДАЧИ

В настоящей главе обсуждаются дискретные модели обучения решению учебных задач. Используется автоматный подход, при котором ученик рассматривается как вероятностный автомат, а его обучение сводится к изменению вероятностей правильных и неправильных переходов из одного состояния в другое. Приводится математическая теория обучения, анализируются различные стратегии взаимодействия учителя с учащимся, методом статистических испытаний определяются характеристики обучения.

2.1. Алгоритмический и вероятностный подходы к деятельности ученика

Эффективный метод исследования процесса обучения состоит в построении структурно-алгоритмической модели учебной деятельности, что может быть осуществлено на основе принципов, сформулированных в [49, с. 38]. Структурно-алгоритмический подход предусматривает рассмотрение любой деятельности как системы взаимосвязанных операций (элементарных действий), которая приводит к достижению поставленной цели. Ее удобно изображать в виде графа деятельности, представляющего собой совокупность вершин, соединенных дугами, который соответствует определенной последовательности выполнения некоторого множества операций.

Очень часто решение стандартной задачи заключается в определении ее типа (распознавание образов) и реализации того или иного алгоритма, приводящего к результату. Ученик, хорошо решающий задачи, прочитав условие, сможет назвать тему и перечислить формулы, которые позволят получить правильный ответ. Школьник, решая последовательность однотипных задач или проводя серию измерений, работает по жесткому алгоритму, выполняя конечный набор действий в определенном порядке (рис. 2.1). Поэтому важной частью обучения является сообщение ученику алгоритма решения типовых задач изучаемой дисциплины. Например, необходимо научить ребенка складывать и вычитать целые числа с помощью счетных палочек, то есть решать примеры вида $S = x \pm y$. Для этого его обучают работать по алгоритму, изображенному на рис. 2.2. Если ученик получает пример «5 + 3 = ?», то он сначала кладет 5 палочек, затем, видя знак сложения «+», к ним докладывает еще 3 палочки, а затем считает все палочки и сообщает результат 8. При решении задачи «6 – 4 = ?» ребенок кладет 6 палочек, затем, видя знак вычитания «–», убирает 4 палочки и, сосчитав оставшиеся, сообщает результат 2.

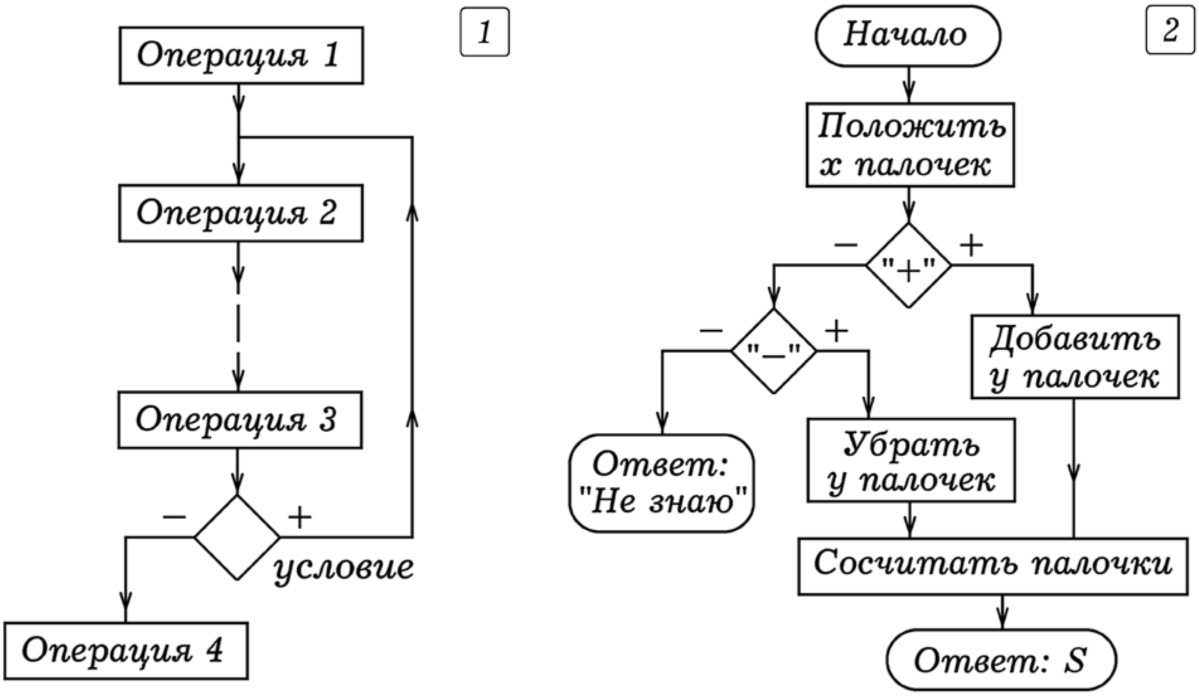


Рис. 2.1. Алгоритмический подход к учебной деятельности ученика

Алгоритмический подход к изучению деятельности ученика состоит в анализе алгоритмов решения учебных задач различного типа. При этом считается, что ученик – детерминированный автомат, работающий в соответствии с заложенной в него программой. Его альтернативой является **вероятностный подход**, согласно которому ученик ведет себя как вероятностный автомат (ВА), выполняющий последовательность действий в зависимости от входной информации и своего внутреннего состояния.

Можно предположить, что если учащийся совершенно необучен, то он выбирает каждую следующую операцию совершенно случайно и после ее выполнения сравнивает свои действия с эталоном (учителем). Учитель подтверждает правильность выбора операции или сообщает, что выбор сделан неверно, подсказывая, какую операцию следовало бы выбрать. Так происходит обучение, в результате которого в сознании учащегося устанавливаются связи между отдельными операциями. Вследствие забывания уровень знаний ученика со временем уменьшается, причем скорость уменьшения знаний пропорциональна их количеству.

ВА, моделирующий ученика, удобно задать в виде стохастического графа – совокупности вершин, соединенных стрелками, которые соответствуют переходам от одной операции к другой. Вероятности переходов можно представить в виде матрицы вероятностей. Если эта матрица будет состоять только из 0 и 1, то она уже будет соответствовать детерминированному автомату. Поэтому вероятностный подход включает в себя детерминированный как частный случай. В дальнейшем ВА, моделирующий ученика, будем называть абстрактной моделью ученика (АМУ) или просто учеником. Для изучения учебной деятельности с помощью детерминированного и вероятностного подходов удобно использовать программный способ синтеза модели [22; 23; 28].

2.2. Математическая теория обучения дискретной модели ученика

Пусть с целью формирования определенного навыка ученик совершает серию из большого числа N однотипных действий. Если ученик необучен, то вероятности выбора любого из m действий равны. При этом вероятность правильного выбора действия $p = 1/m$, а вероятность ошибочного выбора $q = 1 - p = (m-1)/m$. При получении входной информации, подтверждающей правильность или неправильность выбора, происходит подкрепление, вероятность совершения правильного выбора p возрастает на величину $\alpha(1-p) = \alpha q$, а вероятность ошибки q уменьшается на такую же величину, где α – коэффициент научения $(0 < \alpha << 1)$. Уровень сформированности навыка (знаний) ученика будем оценивать по формуле: $Z = (p - 1/m)/(1 - 1/m) = (mp-1)/(m-1)$. Если ученик совсем не обучен, $p = 1/m$, $Z = 0$; если ученик хорошо обучен и всегда совершает правильный выбор операции, то $p = 1$, $Z = 1$. При больших значения m можно считать, что $Z \approx p$. Вследствие забывания уровень знаний за время Δt уменьшается на $\beta \cdot Z \Delta t$, где β – коэффициент забывания $(0 < \beta << 1)$.

Теорема 1. Если после выполнения каждого k-го действия сообщать ученику правильный выбор, увеличивая тем самым его вероятность p на величину $\alpha(1-p)=\alpha q$, то уровень сформированности навыка (знаний) будет увеличиваться по закону:

$$Z(t)=1-(1-Z_0)\exp(-\alpha nt/k).$$

Доказательство. За время dt ученик совершает ndt действий, при этом каждый из $(n/k)dt$ раз учитель сообщает ему, какое действие правильное, в результате чего вероятность p увеличивается на $\alpha(1-p)=\alpha q$. Приращение dp за время dt равно $dp=(\alpha n/k)(1-p)dt$. Имеем:

$$\int_{p_0}^{p}\frac{dp}{p-1}=-\int_0^t\frac{\alpha n}{k}dt,\quad \ln\frac{p-1}{p_0-1}=-\frac{\alpha n}{k}t,\quad p=1-(1-p_0)\exp(-\alpha nt/k).$$

Учитывая, что при больших m $Z\approx p$, получаем доказываемое уравнение.

Теорема 2. Если каждый k-й раз после совершения правильного действия сообщать ученику об этом, увеличивая тем самым вероятность правильного выбора p на величину $\alpha(1-p)=\alpha q$, то уровень сформированности навыка (знаний) будет расти по логистическому закону:

$$\frac{dZ}{dt}=\frac{\alpha n}{k}\cdot\left(1-\frac{1}{m}\right)\cdot\left(Z+\frac{1}{m-1}\right)\cdot(1-Z).$$

Доказательство. За время dt совершается $npdt$ правильных действий, при этом каждый из $(np/k)dt$ раз сообщается ученику, какое действие правильное, в результате чего вероятность p увеличивается на $\alpha(1-p)=\alpha q$. Приращение вероятности правильного выбора dp за время dt равно $dp=(\alpha np/k)(1-p)dt$. Так как $p=((m-1)Z+1)/m$, то $dp=dZ(m-1)/m$. Получаем:

$$\frac{m-1}{m}\cdot\frac{dZ}{dt}=\frac{\alpha n}{k}\cdot\frac{(m-1)Z+1}{m}\cdot\left(1-\frac{(m-1)Z+1}{m}\right).$$

Отсюда следует доказываемое уравнение. При большом числе возможных операций m получаем логистическое уравнение $dZ/dt=AZ(1-Z)$.

Теорема 3. Если каждый k-й раз при совершении неправильного действия сообщать ученику правильный выбор, увеличивая тем самым соответствующую вероятность p на величину $\alpha(1-p)=\alpha q$, то уровень сформированности навыка (знаний) будет увеличиваться по закону:

$$Z(t) = 1 - \frac{1}{\frac{1}{1-p_0} + \frac{\alpha n t}{k}}.$$

Доказательство. За время dt совершается $n(1-p)dt$ неправильных действий, при этом каждый из $(n(1-p)/k)dt$ раз сообщается ученику, какое действие правильное, в результате чего вероятность p увеличивается на $\alpha(1-p) = \alpha q$. Приращение вероятности правильного выбора dp за время dt равно $dp = (\alpha n/k)(1-p)^2 dt$. Разделяя переменные, получаем:

$$\frac{dp}{(1-p)^2} = \frac{\alpha n}{k} dt, \qquad \int_{p_0}^{p} \frac{dp}{(1-p)^2} = \int_0^t \frac{\alpha n}{k} dt.$$

При взятии интеграла следует воспользоваться подстановкой $p = \sin^2 x$:

$$\int \frac{d \sin^2 x}{\cos^4 x} = \int \frac{2 \sin x dx}{\cos^3 x} = -2\int \frac{d \cos x}{\cos^3 x} = \frac{1}{\cos^2 x} = \frac{1}{1-p}.$$

В результате имеем $p = 1 - 1/(1/(1-p_0) + \alpha n t/k)$. Переходя к Z, получаем доказываемое уравнение.

Теорема 4. Вследствие забывания уровень знаний ученика при отсутствии обучения уменьшается по экспоненциальному закону:

$$Z(t) = Z_0 \exp(-\beta \cdot t).$$

Доказательство. Скорость снижения уровня знаний пропорциональна его величине Z. За время dt приращение знаний dZ составляет $dZ = -\beta \cdot Z \cdot dt$. Интегрируя, получим экспоненциальную зависимость.

2.3. Компьютерное моделирование обучения дискретной модели ученика

Пусть ученик осваивает определенную последовательность операций: операция 1, затем операция 2, после этого операция 3, затем снова операция 1 и т. д. Используется метод проб и ошибок. Формирование этого навыка у человека происходит аналогично обучению вероятностного автомата (ВА) с тремя состояниями, которые соответствуют операциям 1, 2 и 3. До обучения ВА случайным образом выполняет различные операции, совершая при этом ошибки. Алгоритм его работы имеет вид стохастического (вероятностного) графа – совокупности вершин, соединенных стрелками, которые соответствуют переходам от одной операции к другой (рис. 2.2.1). Вероятности p_{ij} перехода от i-й операции к j-й образуют двумерную матрицу вероятностей. Если автомат не обучен, то вероятности всех переходов равны:

$$P = \begin{pmatrix} 0{,}33 & 0{,}33 & 0{,}33 \\ 0{,}33 & 0{,}33 & 0{,}33 \\ 0{,}33 & 0{,}33 & 0{,}33 \end{pmatrix},$$

то есть он выбирает следующую операцию совершенно произвольно.

За работой обучаемого ВА следит учитель, знающий правильную последовательность действий и функционирующий как детерминированный автомат по следующему жесткому алгоритму. Если ученик совершил правильное действие, то его «поощряют» высокой оценкой 1, в результате чего вероятность повторения этого действия увеличивается, а остальных – уменьшается. В случае ошибки ученика «наказывают» оценкой 0, что приводит к уменьшению вероятности ее повторения. В результате обучения вероятность правильных переходов p_{12}, p_{23}, p_{31} возрастает, стремясь к 1, а вероятность остальных ошибочных действий – уменьшается, приближаясь к 0. Матрица вероятностей стремится к виду:

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

В конце обучения автомат практически без ошибок выполняет требуемую последовательность $1 \to 2 \to 3 \to 1 \to 2 \to 3 \to 1 \ldots$

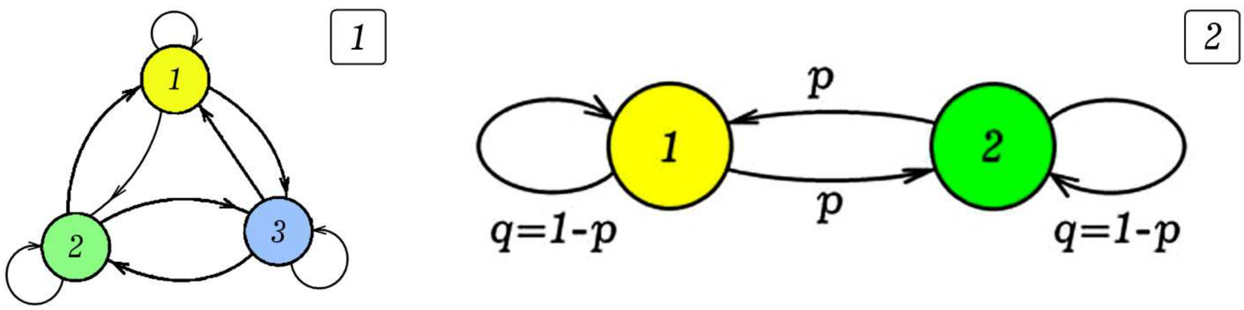


Рис. 2.2. Диаграмма Мура автомата с двумя состояниями

Рассмотрим ВА с двумя внутренними состояниями [19; 20], соответствующими операциям 1 и 2 (рис. 2.2). Будем считать, что автомат обучен, когда из состояния 1 он переходит в состояние 2, а из состояния 2 – в состояние 1 и т. д.: $1 \to 2 \to 1 \to 2 \to 1 \to ...$ Переходы $1 \to 1$ и $2 \to 2$ являются ошибочными. Вероятность правильного действия обозначим через p, тогда вероятность ошибки равна $q = 1 - p$. Можно изучать работу вероятностного автомата с 3 или 4 состояниями, но и в этом случае на любом шаге t один из переходов будет правильным, а остальные – неправильными.

Для моделирования процесса обучения используется программа ПР–2.1 или ПР–2.2. Изначально автомат не обучен, вероятность правильного действия мала ($p = 0{,}01$). Программа содержит цикл, в котором выбор каждой операции осуществляется с помощью генератора случайных чисел. Если случайное число x из интервала [0; 1] меньше p, то ученик совершает правильное действие, если нет, то делает ошибку. Обучение с подкреплением приводит к изменению матрицы вероятностей: вероятность правильного выбора p увеличивается на αq, где α – коэффициент научения ($0 < \alpha < 1$), а вероятность ошибки уменьшается на ту же величину: **`p:=p+a*q; q:=q-a*q;`**. Уровень сформированности навыка равен вероятности p выбора правильной операции.

При обучении человека часть информации забывается. Чтобы это учесть, на каждом временном шаге будем уменьшать вероятность правильного действия p на $\gamma \cdot p$, где γ – коэффициент забывания ($0 < \gamma < 1$), и на такую же величину увеличивать вероятность ошибки q: **`p:=p–g*p; q:=q+g*p;`**. Результаты работы программы представлены на рис. 2.3. При этом могут быть проанализированы следующие ситуации:

1. Обучение с «поощрением»: при выполнении правильного действия ученика «поощряют», пересчитывая вероятности p и q. Так как сначала ученик ошибается гораздо чаще ($q > p$), то при малых t обучение происходит медленно (рис. 2.3.1). Зато по мере увеличения «знаний» вероятность совершения правильного действия растет. Акты обучения происходят все чаще, вероятность p увеличивается почти до 1 . Программа ПР–2.1 должна содержать оператор:

```
If (x<p)and(t<4000) then
      begin p:=p+a*q; q:=q-a*q; end;.
```

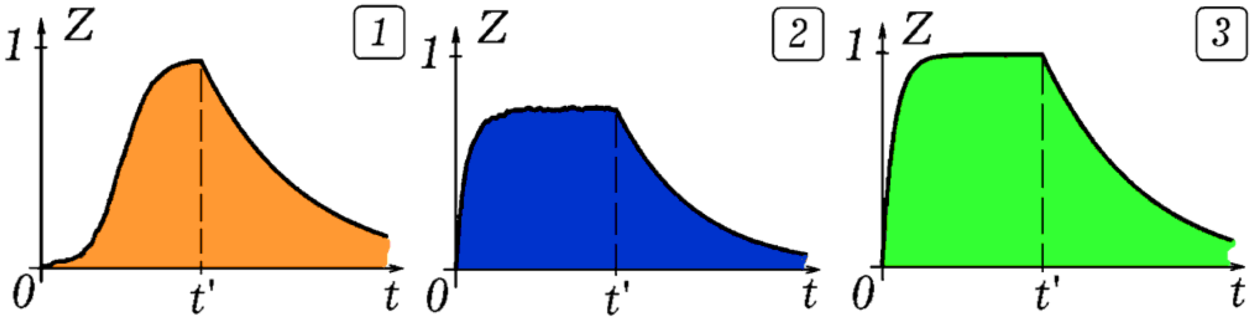


Рис. 2.3. Зависимости уровня знаний ученика от времени

2. Обучение с «наказанием»: в случае ошибки ученика «наказывают», подсказывая ему правильный ответ, что приводит к росту p и уменьшению q. Сначала ученик ошибается часто, поэтому уровень его знаний быстро растет, вероятность ошибки q падает (рис. 2.3.2). Акты обучения происходят все реже и реже, уровень знаний за счет забывания не достигает 1. Программа ПР–2.1 должна содержать условный оператор:

```
If (x>p)and(t<4000) then
     begin p:=p+a*q; q:=q-a*q; end;.
```

3. Обучение с «поощрением» и «наказанием»: при правильном ответе ученика «поощряют», а при неправильном «наказывают», подсказывая правильный ответ. В обоих случаях вероятность правильного действия p растет, а вероятность ошибки q снижается. Так как при любом действии учащегося его учат, то уровень знаний быстро растет и достигает 1 (рис. 2.3.3). Программа ПР–2.1 должна содержать оператор:

```
If t<4000 then begin p:=p+a*q; q:=q-a*q; end;.
```

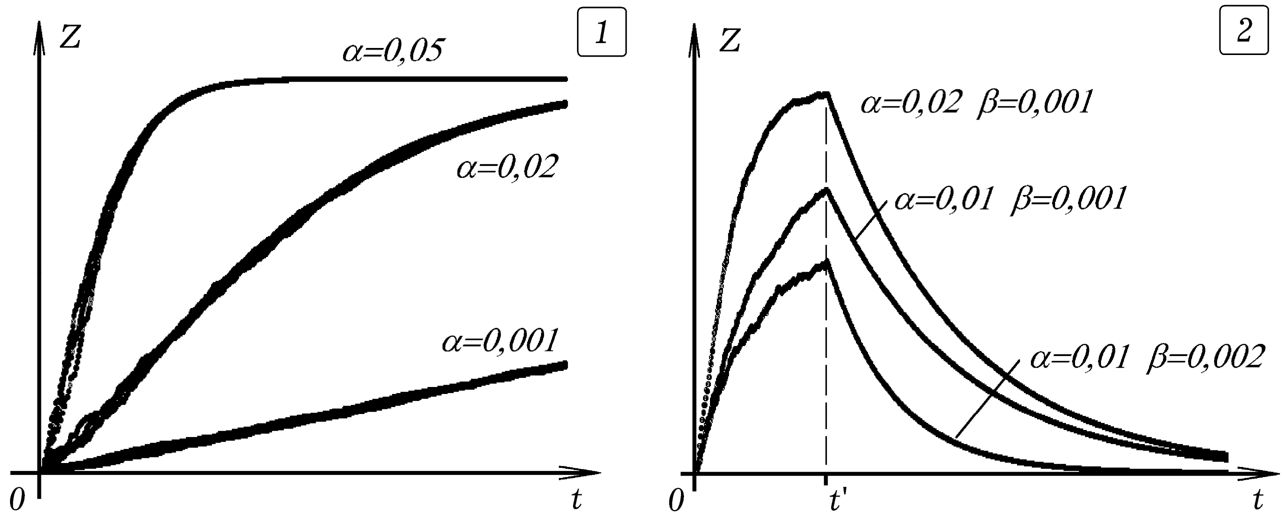


Рис. 2.4. Зависимость уровня знаний от времени

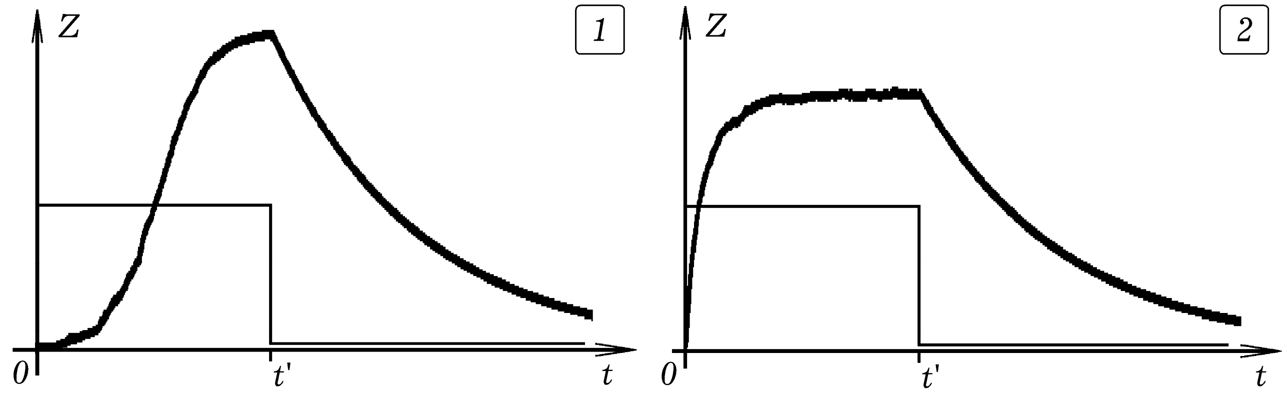


Рис. 2.5. Изменение знаний при обучении и забывании

Результаты использования программы ПР–2.2 представлены на рис. 2.4.1. При малом α уровень знаний растет пропорционально времени (количеству выполненных операций), а при большом α быстро достигает насыщения и остается неизменным. На рис. 2.4.2 и 2.5 изображены кривые зависимостей $Z = Z(t)$ в случае, когда в течение некоторого времени t' осуществлялось обучение, а затем оно прекратилось. Видно, что к концу обучения уровень знаний достигает максимума, а затем убывает вследствие забывания. Графики, представленные на рис. 2.5, соответствуют ситуациям, когда учащегося только «поощряли» за правильные ответы (рис. 2.5.1) и только «наказывали» за неправильные ответы (рис. 2.5.2). Решение этой задачи для ученика, деятельность которого моделируется ВА с четырьмя состояниями, дает аналогичные результаты. Общее число выполненных операций 500–2000.

2.4. Различные стратегии взаимодействия учителя и ученика: моделирование на ЭВМ

Пусть ученик должен механически запомнить последовательность выполнения каких-либо действий, например, научиться считать от 0 до 9 на русском или иностранном языке, выучить алфавит, последовательность каких-то не связанных друг с другом слов, чисел и т. д. К этой ситуации можно отнести случай, когда запоминание немеханическое, элементы усваиваемой информации (операции, действия) связаны между собой логическими связями, но степенью связи мы пренебрегаем или считаем, что в среднем она несколько повышает быстроту усвоения информации, не изменяя характера этого процесса.

Итак, учащийся пытается усвоить выполнение определенной последовательности операций $O_0 \to O_1 \to O_2 \to \ldots \to O_9$, приводящей к решению некоторой учебной задачи. При этом процесс обучения состоит из двух этапов. На первом этапе обучаемый 5–10 раз выполняет последовательность операций $O_0 \to O_1 \to O_2 \to \ldots \to O_9$ вместе с учителем (компьютером, учебником), например, вслух читает алфавит. Каждый раз, когда учащийся совершает правильный переход от операции O_i к O_{i+1}, он учится с коэффициентом научения α_1. Это будем учитывать так: сначала вероятность правильного перехода $p_{i,i+1}$ увеличим на $\alpha_1(1-p_{i,i+1})$, после чего осуществим нормирование: вероятность всех переходов $p_{i,j}$ пересчитаем таким образом, чтобы их сумма была точно равна 1. Для нахождения нормированных вероятностей используется формула:

$$p_{i,j}^{норм} = \frac{p_{i,j}}{p_{i,0}+p_{i,1}+p_{i,2}+\ldots+p_{i,9}}, \quad j = 0,1,2,\ldots 9.$$

На втором этапе обучения реализуется метод проб и ошибок. Ученик по памяти пытается воспроизвести запоминаемую последовательность операций, а учитель как-то реагирует на ответы учащегося: «поощряет» правильные, «наказывает» или исправляет неверные действия и т. д. На рис. 2.6 представлен алгоритм функционирования системы «учитель – учащийся». В случае правильного ответа учащегося учитель «поощряет» его (говорит «Да» или молчит), при этом школьник обучается с коэффициентом научения α_2. В случае ошибочного действия $O_i \to O_k$, $k \neq i+1$ учитель выбирает одну из следующих четырех стратегий реагирования [27].

Стратегия 1: «Неверно, повторите еще раз ту же операцию». При этом он «наказывает» ученика с коэффициентом научения α_3. Это значит, что вероятность выбранного неправильного перехода p_{ik} уменьшается на $\alpha_3 p_{ik}$, а затем осуществляется нормирование всех вероятностей p_{ij} ($j = 1,2,..,N$). После этого учащийся снова пытается выбрать правильную операцию O_{i+1}.

Стратегия 2: «Неверно. Правильно так: O_{i+1}. Повторите еще раз ту же операцию». При этом увеличивается вероятность правильного перехода $p_{i,i+1}$ на $a_3(1-p_{i,i+1})$ и нормируются остальные вероятности p_{ij} ($j = 1,2,..,9$). Ученик продолжает решение задачи с операции O_i.

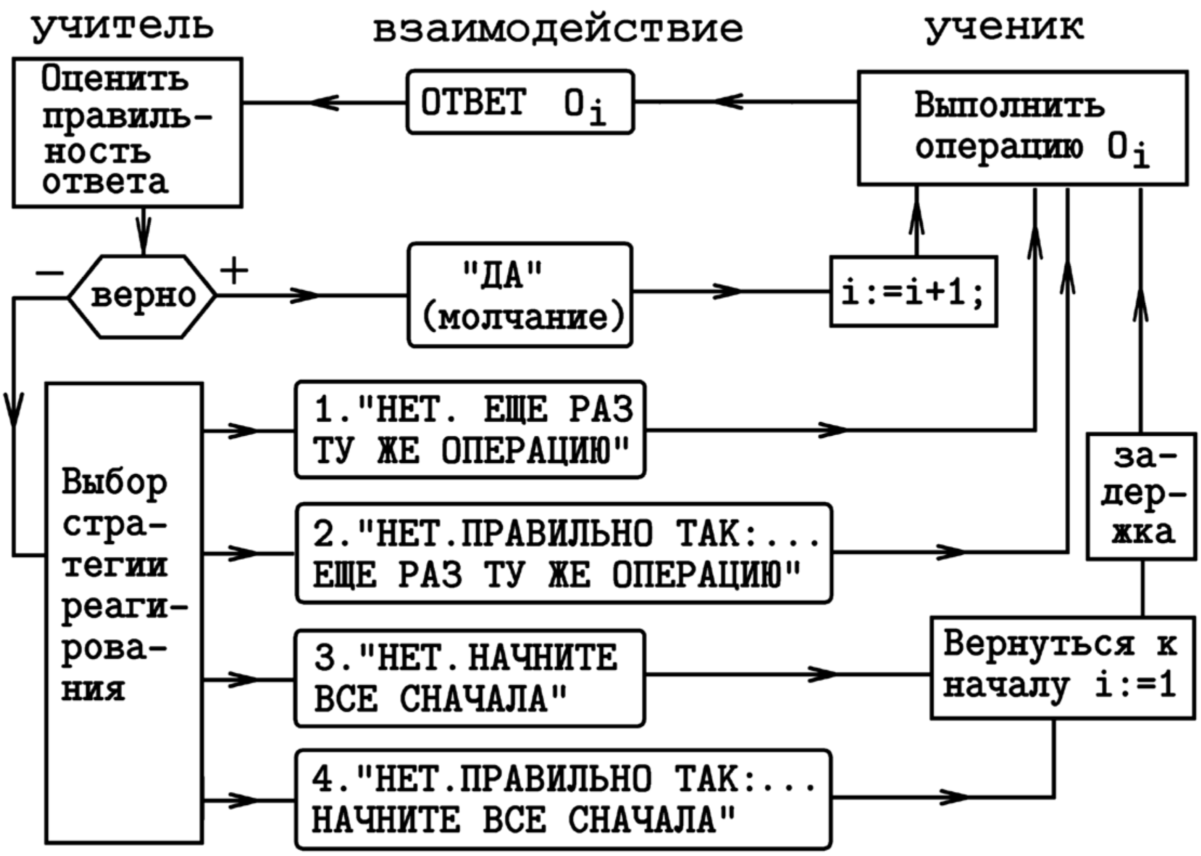


Рис. 2.6. Взаимодействие между учителем и учащимся

Стратегия 3: «Неверно. Повторите всю последовательность действий ***с начал***а (с операции O_1)». Учащегося наказывают с коэффициентом обучения a_3. При этом вероятность выбранного неправильного перехода p_{ik} уменьшается на $a_3 p_{ik}$, после чего осуществляется нормирование всех вероятностей p_{ij} ($j = 0,1,..,9$). Затем ученик начинает решать задачу с самого начала.

Стратегия 4: «Неверно. Правильно так: O_{i+1}. Повторите всю последовательность действий сначала (с операции O_1)». При этом увеличивается вероятность правильного перехода $p_{i,i+1}$ на $a_3(1 - p_{i,i+1})$ и нормируются остальные вероятности p_{ij} ($j = 0,1,..,9$). Ученик возвращается к началу задачи.

Важным вопросом является проблема оценки результатов обучения ученика. В качестве показателей успешности обучения выбраны: 1) уровень знаний (или уровень сформированность навыка), равный среднему арифметическому вероятностей всех правильных переходов $p_{cp} = (p_{01} + p_{12} + p_{23} + ... + p_{89})/9$; 2) вероятность правильного решения задачи (выполнения всей последовательности операций), равная произведению вероятностей правильных переходов: $p_{зад} = p_{01} p_{12} p_{23} p_{34} ... p_{89}$.

Компьютерная программа ПР–2.3, моделирующая анализируемые ситуации, приведена в приложении. Результаты моделирования представлены в таблице 2.1 и на рис. 2.7. В нашем случае всего было 10 операций O_i, им соответствовало 9 правильных переходов: $O_0 \to O_1 \to O_2 \to \ldots \to O_9$. Было задано $\alpha_1 = 0{,}1$ $\alpha_2 = \alpha_3 = 0{,}2$. Число повторов в предварительном обучении равно $k_1 = 5$. Каждый раз, когда учитель показывает правильную последовательность операций, вероятность правильных переходов при этом возрастает. После предварительного обучения (1 этап) уровень знаний был $p_{ср} = 0{,}47$, а вероятность правильного решения задачи $p_{зад} = 0{,}001$.

После этого моделировалось **обучение методом проб и ошибок** (2 этап). Решать задачу ученик начинает с операции O_0, счетчик операций ***N_o*** увеличивается на 1. ПЭВМ выбирает случайное число x из интервала [0; 1] и методом выбора по жребию разыгрывает следующий номер операции, выбираемой учеником. Если ученик совершает правильный переход, то есть $O_i - O_{i-1} = 1$ (после O_4 выбрана O_5), то учитель хвалит учащегося, подтверждая правильность выбора. При этом вероятность правильно совершенного перехода увеличивается на $\Delta p = \alpha_2 (1 - p[o[i-1], o[i]])$, а затем нормируются вероятности p_{ij} ($j = 0,1,..,9$) так, чтобы их сумма была равна 1.

Если ученик совершил неправильный переход ($O_i - O_{i-1} \neq 1$), то учитель наказывает ученика. Если при этом он не подсказывает правильный выбор, то вероятность неверно совершенного перехода уменьшается на $\Delta p = \alpha_3 p[o[i-1], o[i]]$, после чего вероятности p_{ij} нормируются. В случае, когда учитель подсказывает правильный ответ, то используется другой алгоритм: вероятность неверно совершенного перехода уменьшается на Δp, а вероятность правильного перехода от O_{i-1} к $O_{i-1} + 1$ увеличивается на Δp. Сумма вероятностей всех переходов остается равной 1.

Применяется **метод статистических испытаний**. Программа делает 200 циклов (испытаний) и каждый раз вычисляет общее число ответов N_1, число ошибочных ответов N_2, общее время обучения $t = (N_1 - N_2)\Delta t_1 + N_2 \Delta t_2$, которое не должно превзойти заданное значение $t_{\max} = 800$ условных единиц времени (УЕВ). Когда это происходит, программа выходит из цикла, заканчивается данное испытание, результаты выводятся на экран ПЭВМ. В нашем случае $\Delta t_1 = 1$ УЕВ, $\Delta t_2 = 2$ УЕВ, то есть на ошибочный ответ и его исправление затрачивается в 2 раза больше времени, чем на правильный ответ.

Таблица 2.1

Результаты моделирования

Стратегия учителя	Число ответов, $N_{от}$	Число ошибок, $N_{ош}$	Время обучения, t	$p_{ср}$	$p_{зад}$
Стратегия 1	521	286	807	0,84	0,20
Стратегия 2	649	157	806	0,97	0,78
Стратегия 3	698	105	804	0,93	0,50
Стратегия 4	727	78	805	0,96	0,70

Из таблицы 2.1 видно, что при заданных параметрах модели наиболее эффективной является стратегия 2 («Нет. Правильно так. Повторите еще раз ту же операцию») и стратегия 4 («Нет. Правильно так. Повторите все сначала»). Эти стратегии поведения учителя предполагают подсказку учащимся правильного выбора операции. При использовании учителем стратегии 4 учащийся дает максимальное количество ответов при минимальном числе ошибок. Стратегия 1 («Нет. Повторите еще раз ту же операцию») является самой неэффективной.

Стратегии 3 и 4, предусматривающие возврат учащегося к началу выполнения всех действий, приводят к тому, что он чаще выполняет первые операции O_1, O_2, O_3, O_4 и реже последние O_6, O_7, O_8, O_9. Поэтому после второго этапа обучения вероятности переходов p_{01}, p_{12}, p_{23}, p_{34} достаточно высоки, в то время как вероятности p_{56}, p_{67}, p_{78}, p_{89} малы, что приводит к низкой вероятности $p_{зад}$ решения всей задачи. Стратегии 1 и 2 не требуют возврата учащегося к началу задачи, т. е. после ошибки он продолжает выполнять действия с того места, где он совершил ошибку. Поэтому вероятности правильных переходов после второго этапа обучения примерно одинаковы. Стратегия 2 эффективнее стратегии 1, т. к. при ее использовании учитель подсказывает правильный выбор операции [27].

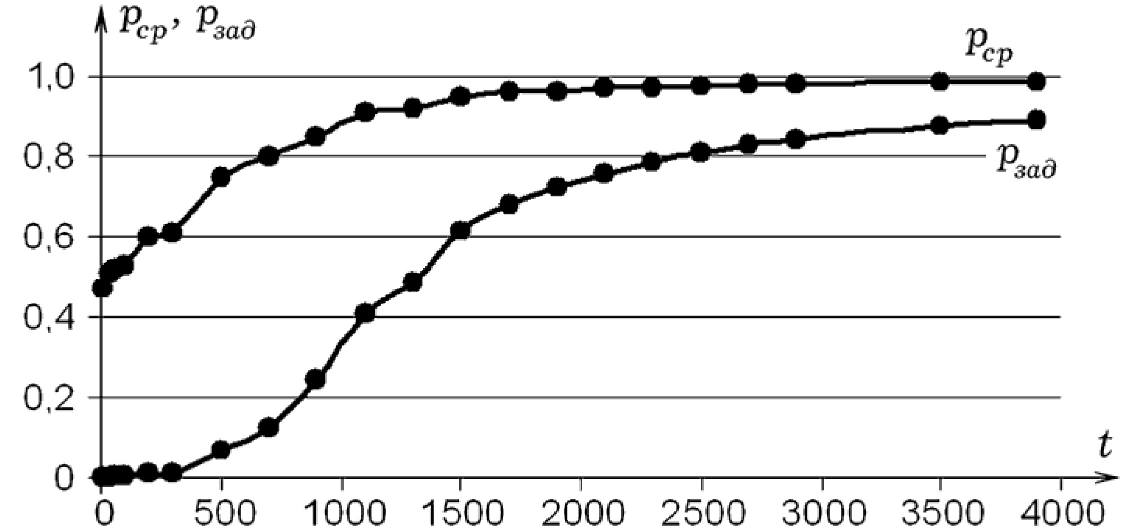


Рис. 2.7. Типичная зависимость $p_{ср}$ и $p_{зад}$ от времени обучения

Для изучения зависимости уровня знаний $p_{ср}$ и вероятности $p_{зад}$ решения задачи от времени обучения методом проб и ошибок был проведен вычислительный эксперимент, в котором задавалось время обучения t и определялись средние значения $p_{ср}$ и $p_{зад}$ каждый раз для 100 испытаний. При этом использовалась стратегия 1. Получающиеся графики представлены на рис. 2.7. Видно, что в процессе обучения кривая научения растет по логистическому закону от 0,47 (уровень после предварительного обучения) до 1. Вероятность правильного выполнения задачи $p_{зад}$ сначала невелика, затем также возрастает, стремясь к 1. При использовании других стратегий характер изменения $p_{ср}$ и $p_{зад}$ такой же, время формирования навыка меньше.

2.5. Решение сложных задач с обучением

При решении сложной задачи учащийся ведет себя как вероятностный автомат (ВА), осуществляющий ту или иную последовательность операций из некоторого множества $O = \{1,\ 2,\ ...,\ N\}$. Пусть решение всегда начинается с операции 1; последовательность $1 \to 2 \to 3 \to \ldots \to N$ является правильной. Вероятность выбора операции 2 после 1 обозначим p_1, выбора $(i+1)$-й операции после i-й – p_i. Решение задачи аналогично поиску выхода из лабиринта. Все операции ВА совершает безошибочно; если же он допускает ошибку, значит, выполняет какую-то другую операцию. Вероятность решения задачи с первой попытки равна произведению $P = p_1 p_2 p_3 ... p_N$. Если алгоритм решения известен, то $p_1 = p_2 = ... = 1$, и ВА ведет себя как детерминированный автомат, достигая результата за минимальное число шагов.

В начале обучения вероятности p_i правильного выполнения $(i+1)$-й операции после i -й операции малы и равны 0,1. При правильном выборе первой операции он переходит ко второй и т. д. На каждый шаг затрачивается одинаковое время Δt. Допустим, он ошибся, но не заметил этого и продолжает двигаться по неправильному пути. Ученик совершает определенное количество шагов k (пусть $k = N + 2$) и, придя к неверному результату, обращается к учителю. Учитель в течение времени $2\Delta t$ проверяет решение и находит число j первых правильно выполненных операций. Он сообщает ученику, что операции 1, 2, 3, ..., j выполнены правильно, и подсказывает $(j+1)$-ю операцию. В результате этого подкрепления и подсказки соответствующие вероятности p_i $(i = 1, 2, ..., j, j+1)$ правильных переходов увеличиваются на $\alpha(1 - p_i)$. После этого ученик либо возвращается к операции 1 (стратегия 1), либо пытается закончить решение задачи, выполняя $j+1$, $j+2$ и последующие операции (стратегия 2). В случае ошибки он снова обращается к учителю.

После того, как задача решена правильно, происходит подкрепление, и вероятность p_i $(i = 1, 2, ..., N)$ выбора правильных действий увеличивается на $\alpha(1 - p_i)$. Вычисляется вероятность решения задачи $P = p_1 p_2 p_3 ... p_N$, характеризующая степень обученности ученика. Затем он приступает к решению следующей задачи того же типа, и все повторяется.

Используется программа ПР–2.4. Номер выполняемой операции сохраняется в массиве O[i]. При правильном решении O[1] = 1, O[2] = 2, ... O[N] = = N. Если допущена ошибка в выборе k-й операции, то O[k] = –1. На экране строятся графики зависимостей номера выполняемой операции и общей вероятности решения задачи от времени $P(t)$. Если решение задачи не потребовало вмешательства учителя, то первый график имеет вид возрастающей прямой, идущей от первой к N -й операции. При наличии ошибок в решении этот график будет прерываться. Графиком зависимости вероятности правильного решения от времени $P(t)$ является логистическая S-кривая. Чтобы промоделировать ситуацию, в которой учащийся после обнаружения ошибки не продолжает решать задачу, а возвращается к первой операции, необходимо раскомментировать оператор {`If j<N then j:=1;`}.

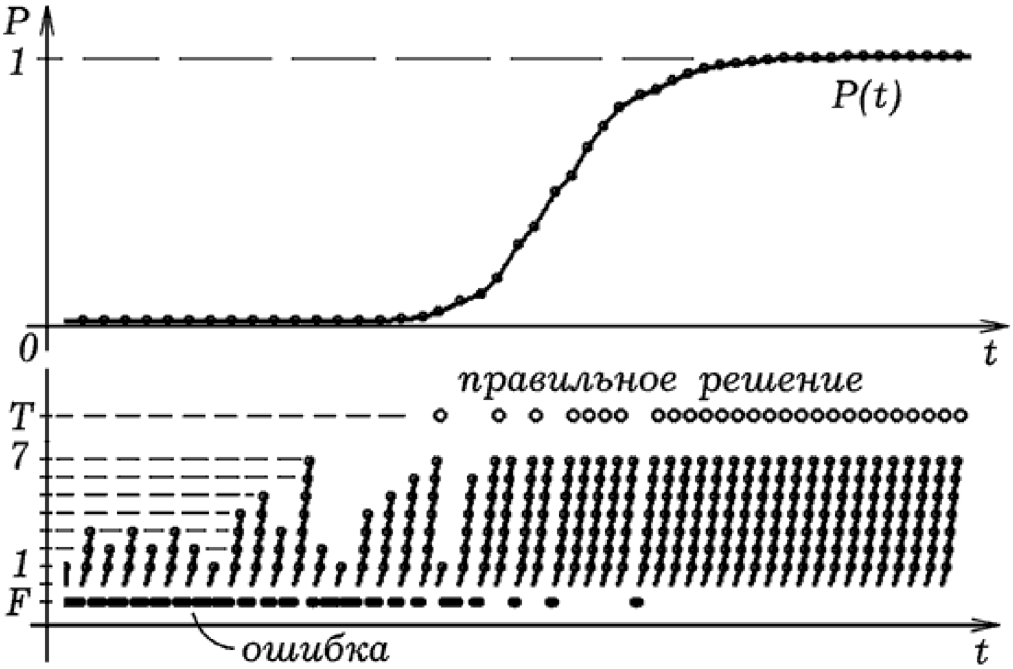


Рис. 2.8. Моделирование решения сложной задачи: стратегия 1

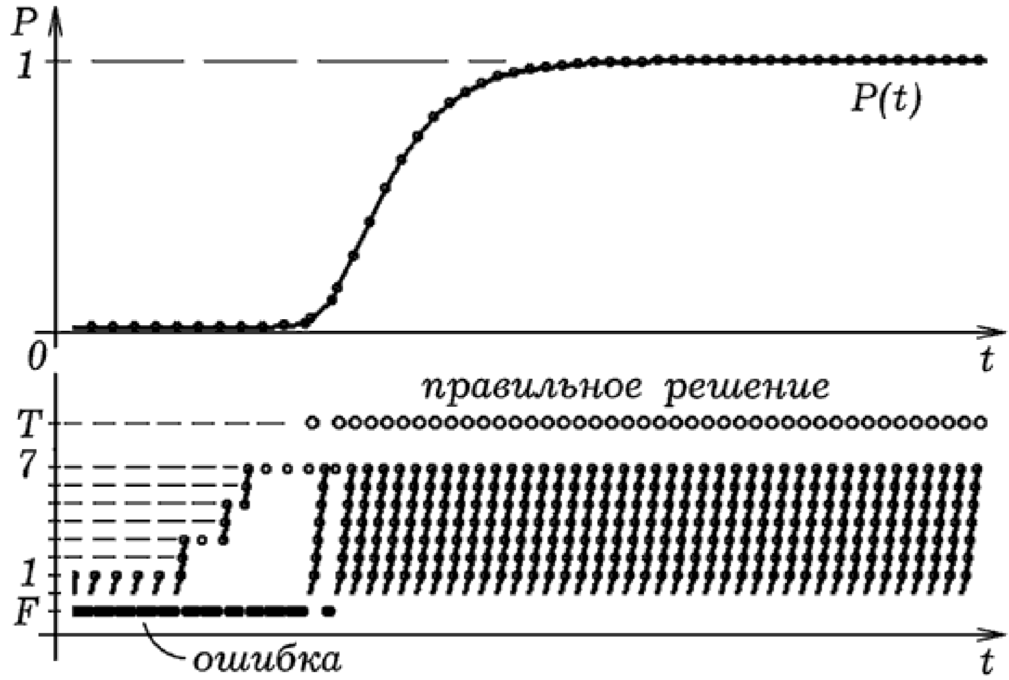


Рис. 2.9. Моделирование решения сложной задачи: стратегия 2

Рассмотрим типичные результаты имитационного моделирования решения сложной задачи для двух стратегий: 1) учащийся после совершения ошибки и подсказки учителя возвращается к началу решения задачи (рис 2.8); 2) учащийся после совершения ошибки и подсказки учителя продолжает решать данную задачу (рис. 2.9). При запуске программа рисует ступенчатую кривую, показывающую, как изменяется номер выполняемой учащимся операции, и строит график зависимости вероятности решения задачи данного типа от времени $P = P(t)$. В случае ошибки учащегося ПЭВМ ставит точку на уровне F (FALSE). Если задача решена правильно и до конца, то ПЭВМ ставит точку на уровне T (TRUE).

Видно, что в обоих случаях формирование навыка происходит в соответствии с логистической функцией, графиком зависимости $P(t)$ является S-кривая. Во втором случае (рис. 2.9) формирование навыка происходит заметно быстрее, чем в первом (рис. 2.8). Это объясняется тем, что каждый раз, возвращаясь к началу решения (стратегия 1), ученик сначала учится выполнять операции 1, 2, 3 и не может сразу приступить к выполнению операций 4, 5, 6, 7. Во втором случае учащийся усваивает все операции одновременно.

2.6. Решение сложных задач без обучения

Имеется задача, для решения которой следует выполнить N операций в заданном порядке $O_1 \rightarrow O_2 \rightarrow \ldots \rightarrow O_N$. Предположим, что учащийся не знает, как она решается, но ему известно, что для этого требуется не более L шагов. При решении задачи учащийся как бы движется по некоторому пути в лабиринте, каждый раз делая выбор: выполнить новый шаг в том или ином направлении или начать все сначала. По мере увеличения числа выполненных шагов желание учащегося идти по выбранному пути уменьшается, так как он не знает, правильно ли он выбрал направление движения или ошибся в самом начале. Он выполняет L шагов в некотором направлении (проходит несколько узлов лабиринта) и, если ему не удается получить ответ (выйти из лабиринта), возвращается к началу пути. Затем повторяет все снова и так k раз. С каждой новой попыткой вероятность того, что учащийся бросит решать задачу, увеличивается. Упорство учащегося характеризуется глубиной поиска L и числом предпринятых попыток k. Будем исходить из того, что учащийся практически не обучается: с самого начала все операции он выполняет правильно, и для решения задачи ему необходимо найти правильный путь движения, что возможно сделать лишь методом проб и ошибок [53].

Автомат, моделирующий деятельность учащегося, перед каждым шагом должен «решить», следует выполнять новое действие или лучше вернуться к началу O_1. Логично предположить, что с каждым i *-ым* шагом вероятность того, что ВА будет продолжать идти по выбранному пути, уменьшается по экспоненциальному закону: $P = \exp(-ai)$. Перед каждой новой попыткой решить задачу ВА должен сделать выбор: продолжать решение или отказаться от него. Вероятность каждой следующей попытки тоже уменьшается по закону $Q = \exp(-bj)$, где j – номер попытки. Справедливо утверждение: ВА обязательно решит задачу, если: 1) ВА умеет выполнять все необходимые операции O_1, O_2, ..., O_N; 2) в «памяти» ВА имеется путь от первой O_1 до конечной операции O_N с ненулевыми вероятностями переходов p_i; 3) число шагов L, предпринимаемых ВА в каждой попытке, больше длины решения N; 4) ВА делает бесконечно большое число попыток; 5) ВА умеет отличать правильный ответ от неправильного. Рассмотрим алгоритм программы, моделирующей деятельность ученика при решении задачи:

```
i:=1; op[1]:=1; k:=1; shag:=0; R:=true; ПЕЧАТЬ(i,'  ',op[i]);
ПОВТОРЯТЬ {–
  shag:=shag+1; i:=i+1; x:=RND(100)/100;
  ЕСЛИ x<p[op[i–1]] ТО op[i]:=op[i–1]+1 ИНАЧЕ {m: op1:=RND(30)/10+1;
  ЕСЛИ(op1=op[i–1]+1)ИЛИ(op1=op[i–1]) ТО ИДТИ К m
                                                        ИНАЧЕ op[i]:=op1;}
  ЕСЛИ op[i]–op[i–1]=1 ТО ПЕЧАТЬ(' ВЕРНО') ИНАЧЕ {
                                           ПЕЧАТЬ (' НЕВЕРНО'); R:=false; }
  Q :=exp(–0.08*shag); x:=RND(100)/100;
  ЕСЛИ x > Q ТО [shag:=0; op[i]:=1; R:=true;
                          k:=k+1; ПЕЧАТЬ('попытка ',k);]    —}
ПОКА ((op[i]=6)И(R=true))ИЛИ(i>150);
ЕСЛИ (op[i]=6)И(R=true) ТО ПЕЧАТЬ('ЗАДАЧА РЕШЕНА');
ПЕЧАТЬ('ЧИСЛО ШАГОВ ',i,');
```

Используется программа ПР–2.5. Элементы массива op[i] хранят номер операции, выбранной на i -ом шаге с момента $t = 0$ получения задачи. Номер шага по выбранному пути решения задачи (попытка k) хранится в счетчике shag. ВА, моделирующем ученика, может допустить ошибку и продолжать двигаться в неверном направлении. Так продолжается, пока ВА не примет решение начать новую попытку и вернется к решению задачи. Даже двигаясь в правильном направлении, ВА может остановиться и вернуться к началу задачи. При этом число i выполненных шагов с момента $t = 0$ остается равным некоторому целому числу, а переменная shag принимает значение 1. Если все операции выполнены правильно, то ВА «понимает», что решил задачу. Результаты моделирования представлены на рис. 2.10.1. По вертикали откладывается номер операции, последняя попытка соответствует правильному пути $O_1 \to O_2 \to \ldots \to O_6$.

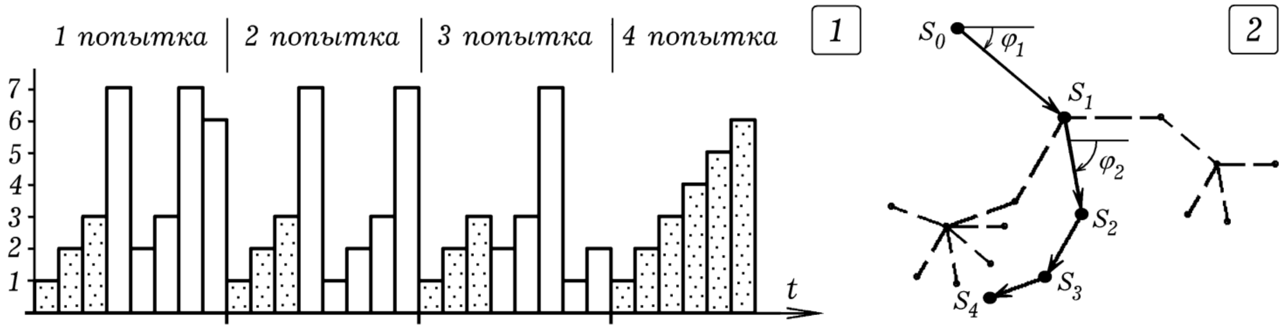


Рис. 2.10. Результаты моделирования решения задачи без обучения

Чтобы построить граф решения (рис. 2.10.2), от начальной точки S_0, соответствующей начальному состоянию, откладывается вектор S_0S_1 длиной $L_1 = 1$ под углом $\varphi_i = j\Delta\varphi$ к горизонтали, где j – номер операции. Затем от точки S_1 откладывается вектор S_1S_2 длиной $L_2 = 1/2$ под углом $\varphi_i = j\Delta\varphi$ к горизонтали и т. д. В результате перебора всех различных сочетаний и последовательностей операций получается граф, имеющий фрактальную структуру. Правильному решению задачи соответствует некоторый путь из состояния S_0 в состояние S_m, где m – число операций. Понятно, что при заданных параметрах модели (p_i, a и b) число шагов N, совершаемых ВА для решения задачи, является случайной величиной. Чтобы определить среднее значение N, использовался метод статистических испытаний. В программу был добавлен цикл, в котором 100–500 раз запускается ВА, решающий задачу. При этом подсчитывалось общее число шагов, число отказов от решения, после чего результаты усреднялись (программа ПР–2.6).

Аналогичным образом определяется среднее время решения задачи данного типа. Создается компьютерная модель вероятностного автомата (ВА), который моделирует движение учащегося по некоторому пути в лабиринте. В одном случае ВА выполняет L шагов в некотором направлении (проходит несколько узлов лабиринта) и, если ему не удается получить ответ (выйти из лабиринта), возвращается к началу пути. Затем повторяет все снова и так 500–1000 раз. В другом случае ВА, моделирующий деятельность учащегося, перед каждым шагом «решает», следует выполнять новое действие или лучше вернуться к началу – операции 1. Во всех случаях ВА не обучается, при фиксированных вероятностях p_i время (число шагов N) решения задачи – случайная величина. Подсчитывается общее число шагов, число отказов от решения, после чего результаты усредняются. По результатам можно построить график зависимости времени решения задачи от вероятности правильного выполнения операции p_i.

2.7. Зависимость времени решения задачи от вероятности выполнения операции

Допустим, ученик решает учебную задачу, заключающуюся в выполнении последовательности операций $O_1 \to O_2 \to O_3 \to O_4 \to O_5$. Это может быть повторение алфавита или формирование навыка сборки того или иного технологического узла. Средняя длительность и вероятность правильного выполнения каждой операции задаются матрицами: $\tau_i = (1{,}6 \pm 0{,}5;\ 2{,}7 \pm 0{,}5;\ 1{,}4 \pm 0{,}5;\ 3{,}8 \pm 0{,}5;\ 2{,}6 \pm 0{,}5)$ и $p_i = (0{,}6;\ 0{,}7;\ 0{,}4;\ 0{,}8;\ 0{,}6)$. То есть для операции O_1 время выполнения τ_1 – случайная величина с равномерным законом распределения в интервале $1{,}6 \pm 0{,}5$, а вероятность правильного выполнения $p_1 = 0{,}6$. Если операция выполнена неверно, то она повторяется снова и снова, пока не будет выполнена правильно. Необходимо вычислить среднее время решения задачи и ее среднее квадратическое отклонение (СКО).

Рис. 2.11. Распределение времени решения задачи как случайной величины

Будем использовать метод статистических испытаний. Рассмотрим программу ПР–2.7, содержащую цикл, в котором моделируется однократное выполнение всех операций, приводящих к правильному решению задачи. Правильность той или иной операции моделируется с помощью генератора случайных чисел. В случае правильного выполнения операции номер выполняемой операции увеличивается на 1. Если операция выполнена неверно, то учащийся всегда (!) понимает это и повторно пытается сделать ее правильно. При этом определяется общее время i-й реализации процесса решения задачи. Затем осуществляется следующая $(i+1)$-я реализация и так 2000–5000 раз. В результате вычисляется суммарное время выполнения всех M реализаций, среднее значение и СКО времени τ_p одной реализации. Программа также позволяет понять характер распределения величины τ_p и после доработки получить соответствующую гистограмму. В нашем конкретном случае получается результат, представленный на рис. 2.11.1. При правильном выполнении всех операций с первого раза время решения задачи минимально и составляет $\tau_1+\tau_2+\tau_3+\tau_4+\tau_5=12{,}1$ УЕВ. Среднее время решения и его СКО составляют 18,4 УЕВ и 5,05 УЕВ. Чтобы рассчитать СКО, необходимо сначала найти среднее время, а затем присвоить это значение переменной tsr и еще раз запустить программу.

В некоторых случаях можно считать, что все операции имеют одинаковую длительность и вероятность выполнения. На рис. 2.11.2 представлено распределение времени решения задачи как случайной величины при $\tau_i=1{,}0\pm0{,}1$ УЕВ и $p_i=0{,}6$ $(i=1{,}2{,}...{,}5)$. Среднее время $8{,}31$ УЕВ при СКО $2{,}36$.

Программа ПР–2.8 позволяет получить распределение времени решения задачи как случайной величины (рис. 2.12) в случае, когда время и вероятность выполнения отдельной операции одинаковы и равны $\tau_i=1$ УЕВ, $p_i=0{,}1,\ 0{,}2,\ \ldots,\ 0{,}8$ $(i=1,\ldots,\ 5)$. Видно, что при увеличении p_i среднее значение величины приближается к его минимальному значению 5 УЕВ, а СКО (разброс) уменьшается. К аналогичному результату можно прийти, анализируя график зависимости среднего времени решения задачи из пяти операций от вероятности выполнения каждой операции $p_i=p$ (рис 2.13). Будем приближенно считать, что при увеличении вероятности p среднее время решения задачи уменьшается по закону $t=t_{\min}+b/p$ $(0<b<1)$. Эту зависимость можно аппроксимировать иначе: $p = \exp(-b(t - t_{\min}))$ или $t = t_{\min} - \ln p / b$. Полученные закономерности помогут построить имитационную модель обучения.

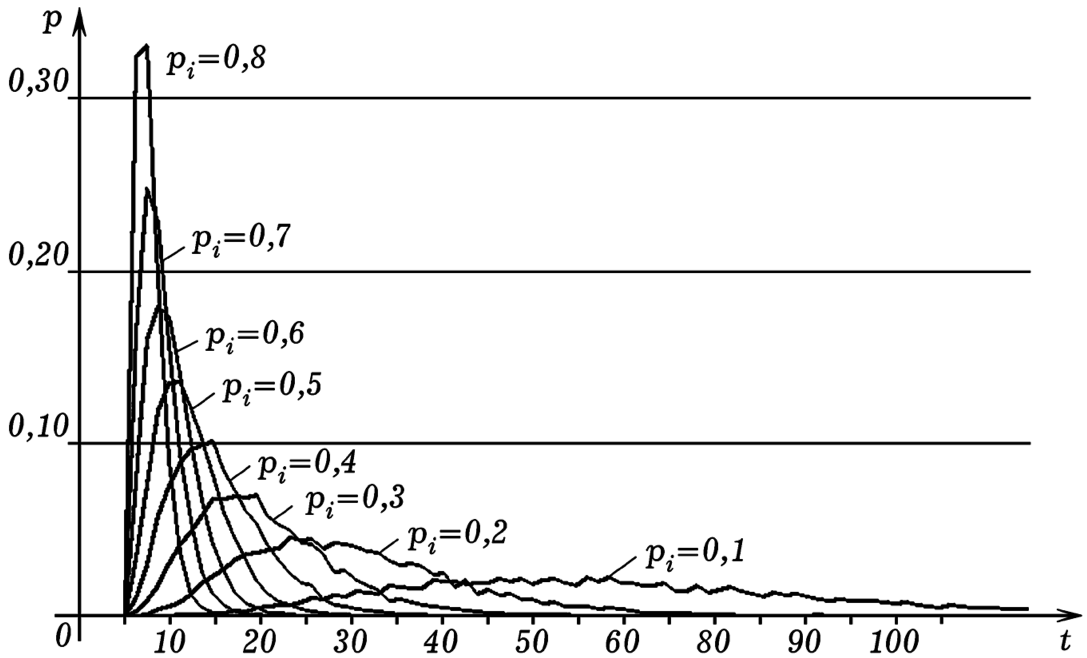


Рис. 2.12. Распределение времени τ решения задачи при различных p_i

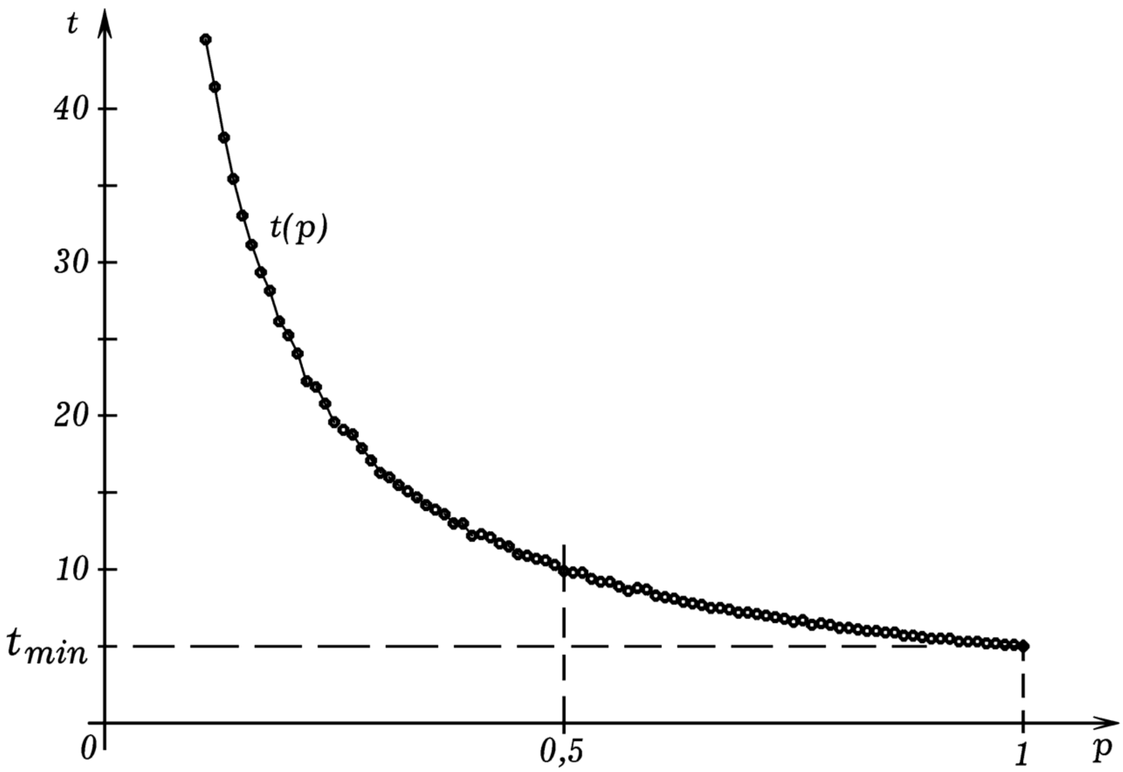


Рис. 2.13. Зависимость времени решения задачи от вероятности p

2.8. Приложение к главе 2

```
uses crt, graph;                                           {ПР-2.1}
var t,Gd,Gm : integer; x,p,q,a,g : real;           {Free Pascal}
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
Randomize; line(0,450,640,450); line(10,0,10,480);
p:=0.01; q:=1-p; a:=0.003; g:=0.0004;
Repeat
  inc(t); x:=random(1000)/1000;
  If (x>p)and(t<4000) then begin
    p:=p+a*q; q:=q-a*q; end;
    p:=p-g*p; q:=q+g*p; {zabivanie}
   Circle(10+round(t/15),450-round(400*p),2);
until (t>10000)or(KeyPressed);
Repeat until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                          {ПР-2.2}
const N=2;  NN=2000;
var x : array[0..NN+1] of integer; p : array[1..2,1..2] of real;
EC,DV,MV,k,i,j : integer; alpha,beta,s,rnd : real;
Procedure Uchenik;
Begin  rnd:=random(1000)/1000; inc(i);
  If rnd<p[j,1] then x[i]:=1 else x[i]:=2; end;
Procedure Obuchenie;
Label met;
Begin
  If k>500 then goto met;
  If (x[i-1]=1)and(x[i]=2) then  begin
   p[1,2]:=p[1,2]+alpha*(1-p[1,2]); p[1,1]:=(1-alpha)*p[1,1];
end;
  If (x[i-1]=2)and(x[i]=1) then  begin
  p[2,1]:=p[2,1]+alpha*(1-p[2,1]);
  p[2,2]:=(1-alpha)*p[2,2];  end; met:
end;
Procedure Zabyvan;
begin
  p[1,2]:=p[1,2]-beta*(p[1,2]-p[1,1]);
  p[1,1]:=p[1,1]+beta*(p[1,2]-p[1,1]);
  p[2,1]:=p[2,1]-beta*(p[2,1]-p[2,2]);
  p[2,2]:=p[2,2]+beta*(p[2,1]-p[2,2]); end;
Procedure Raschet;
begin
For i:=1 to 2 do For j:=1 to 2 do p[i,j]:=0.5;
For k:=1 to NN-5 do begin
  Uchenik; Obuchenie; Zabyvan; j:=x[i]; delay(1);
  circle(10+round(0.3*i),450-round(800*((p[1,2]-0.5))),1); end;
end;
BEGIN
  DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
  Randomize; Line(0,450,640,450);
  beta:=0.001; alpha:=0.01; Raschet;
```

```
  beta:=0.002; alpha:=0.01; Raschet;
  beta:=0.001; alpha:=0.02; Raschet;
  Repeat until Keypressed; CloseGraph;
END.

uses crt;                                                    {ПР-2.3}
const  N=10; K_is=100; a1=0.1; a2=0.2; a3=0.2;
slovo:array[1..10]of integer=(0,1,2,3,4,5,6,7,8,9);
var kol_isp,time,N_o,N_pr,N_osh,k,l,i,i1,i2,j: integer;
p: array[0..N,0..N] of real; o: array [-1..N]of integer;
N1,N2,Z1,Z2,P_sr,P_zad,dp,x,sum_p,sum,tt : real;
Label mm;
Procedure Pechat;
begin  {raspechatka massiva p}
 For i1:=0 to 9 do begin writeln;
 For i2:=0 to 9 do write(' ',p[i1,i2]:2:3);
end; writeln; end;
Procedure Normir; {normirovanie p}
begin sum:=0; For j:=0 to N-1 do sum:=sum+p[o[i-1],j];
  For j:=0 to N-1 do p[o[i-1],j]:=p[o[i-1],j]/sum; end;
BEGIN clrscr;
Repeat k:=0;
 For i:=0 to N do For j:=0 to N do p[i,j]:=1/N;
 Repeat inc(k); randomize; i2:=0;
{1. PREDVARITELNOE OBUCHENIE}
  For i:=2 to 10 do begin i1:=slovo[i];
  p[i2,i1]:=p[i2,i1]+a1*(1-p[i2,i1]);
  For j:=0 to N-1 do begin
  If j<>i1 then p[i2,j]:=p[i2,j]-a1*(1-p[i2,i1])/(N-1); end;
  i2:=i1; end;
until (k=5)or(Keypressed); {Pechat;}
{2. OBUCHENIE METODOM PROB I OSHIBOK}
l:=0; N_osh:=0; N_o:=0;
Repeat inc(l); o[0]:=0;
For i:=0 to 8 do begin inc(N_o); sum_p:=0; x:=random(100)/100;
for j:=0 to N-1 do begin sum_p:=sum_p+p[i,j];
If x>sum_p then o[i]:=j+1; end; {uchenik vibiraet opraciu}
If (o[i]-o[i-1]=1)and(i<9) then begin inc(N_pr); {verno}
p[o[i-1],o[i]]:=p[o[i-1],o[i]]+a2*(1-p[o[i-1],o[i]]);
  Normir; end;
 (*If o[i]-o[i-1]<>1 then begin inc(N_osh); {neverno}
  p[o[i-1],o[i-1]+1]:=p[o[i-1],o[i-1]+1]
                            +a3*p[o[i-1],o[i-1]+1];
  Normir; goto mm; {o[i]:=o[i-1];} end;
end; mm: {2 i 4 strategii} *)
If o[i]-o[i-1]<>1 then begin inc(N_osh); {neverno}
  p[o[i-1],o[i]]:=p[o[i-1],o[i]]-a3*p[o[i-1],o[i]];
  Normir; {goto mm;} o[i]:=o[i-1]; end;
end; mm:  {1 i 3 strategii}
until (N_o+N_osh>800)or(Keypressed); {Pechat;}
time:=N_o+N_osh; inc(kol_isp);
P_sr:=(p[0,1]+p[1,2]+p[2,3]+p[3,4]
             +p[4,5]+p[5,6]+p[6,7]+p[7,8]+p[8,9])/9;
```

```
P_zad:=p[0,1]*p[1,2]*p[2,3]*p[3,4]*p[4,5]
             *p[5,6]*p[6,7]*p[7,8]*p[8,9];
writeln(N_o,' Kol_osh ',N_osh,' Vrem ',time,
             ' P_sr ',P_sr,' P_zad ',P_zad);
N1:=N1+N_o; N2:=N2+N_osh; tt:=tt+time; Z1:=Z1+P_sr; Z2:=Z2+P_zad;
until (Keypressed)or(kol_isp=K_is);
writeln('Kol_otv ',N1/K_is,' Kol_osh ',N2/K_is,
        ' Vremya ',tt/K_is);
writeln(' P_sr ',Z1/K_is,' P_zad ',Z2/K_is); readkey;
END.
```

```
uses crt, graph;                                            {ПР-2.4}
const N=8; Mt=1; a=0.13;
var x,y,pp: real;
i,j,k,t,d,r,u,h,l,M,flag,dt,DM,DV,MV:integer;
O: array[0..50] of integer; p: array[-1..50] of real;
BEGIN x:=24; M:=200; DV:=Detect;
InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
For i:=1 to N do p[i]:=0.05;
For d:=1 to 50 do begin O[1]:=1; j:=1; dt:=1;
  Repeat i:=j; k:=0; flag:=1; { -- reshenie zadachi -- }
    Repeat circle(Mt*t,450-10*O[i],1); inc(k);
      If r=O[i]-1 then line(Mt*t,450-10*O[i],Mt*u,450-10*r);
      r:=O[i]; u:=t; inc(i); t:=t+dt; x:=x+157.3+random(87);
      If x>M then Repeat x:=x-M; until x<M; y:=x/M; delay(30);
      If y<p[O[i-1]] then O[i]:=O[i-1]+1 else
        begin flag:=0; O[i]:=-1; circle(Mt*t,450+10,1); end;
    until (Keypressed)or(k>=N+2)or((O[i]>=N)and(flag=1));
    circle(Mt*t,450-round(250),2); { = PROVERKA = }
    j:=1; h:=1; For l:=1 to N do
      If (O[l+1]=O[l]+1)and(h=1) then inc(j) else h:=0;
    O[1]:=1; pp:=1;
    For l:=1 to j+1 do p[l]:=p[l]+a*(1-p[l]); pp:=1;
    For l:=1 to N do pp:=pp*p[l];
    circle(Mt*t,450-round(150*pp),2); t:=t+2*dt;
{  If j<N then j:=1;}
  until (Keypressed)or(j=N);
If j=N then circle(Mt*(t-1),450-12*N,3); j:=1;
end; readkey; CloseGraph;
END.
```

```
uses crt,graph;                                              {ПР-2.5}
const p: array [1..6]of real= (0.6,0.5,0.4,0.7,0.6,0);
var zz,dv,mV,kk,i,k,shag,op1:integer;sluch,ver,S:real;
R:boolean; x,y,op:array[1..1501]of integer; Label m;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');randomize;
i:=1; op[1]:=1; k:=1; x[1]:=320; y[1]:=40; shag:=1;
R:=true; circle(x[1],y[1],3);
Repeat inc(shag); inc(i); sluch:=random(100)/100;
  If sluch<p[op[i-1]] then op[i]:=op[i-1]+1 else
      begin m:
        op1:=round(random(50)/10)+1;
        If (op1=op[i-1]+1)or(op1=op[i-1])
```

```
              then goto m else op[i]:=op1;
      end;
  If op[i]-op[i-1]=1 then begin {write(' verno');} end
                                        else R:=false;
  ver:=exp(-0.1*shag); sluch:=random(100)/100;
  If shag>7 then begin shag:=1; op[i]:=1;
  R:=true; inc(k);x[1]:=320; y[1]:=40;
  end else begin zz:=op[i];
  x[shag]:=x[shag-1]+round(250/shag*cos(2*3.14*zz/15));
  y[shag]:=y[shag-1]+round(250/shag*sin(2*3.14*zz/15));
  line(x[shag-1],y[shag-1],x[shag],y[shag]);
  circle(x[shag],y[shag],2);
  rectangle(5*i,450-10,5*(i-1),450-10*op[i]); delay(10);
  end;
until ((op[i]=5)and(R=true))or(i>1500)or(Keypressed);
inc(kk); S:=S+i; shag:=1; x[1]:=320; y[1]:=40;
Repeat until KeyPressed; Closegraph;
END.
```

```
uses crt;                                               {ПР–2.6}
const p: array [1..5]of real=(0.8,0.85,0.7,0.6,0);
var i,k,k_isp,shag,op1:integer; x,ver,S:real;
R:boolean; op:array[1..9000]of integer; Label m;
BEGIN clrscr; randomize; k_isp:=1; i:=1;
Repeat op[1]:=1; k:=1; shag:=0; R:=true;
Repeat inc(shag); inc(i); x:=random(100)/100;
 If x<p[op[i-1]] then op[i]:=op[i-1]+1 else begin m:
 op1:=round(random(30)/10)+1;
 If (op1=op[i-1]+1)or(op1=op[i-1]) then goto m
                         else op[i]:=op1; end;
 If op[i]-op[i-1]<>1 then R:=false;
{ writeln('ISP ',k_isp,' | ',i,' SHAG ',shag,
       ' OPERACIYA ', op[i],' PUT ',R); delay(50);}
 ver:=exp(-0.1*shag); x:=random(100)/100;
 If x>ver then begin shag:=0; op[i]:=1;
 R:=true; inc(k);
 writeln('VSE SNACHALA. Popitka ',k); delay(50); end;
until ((op[i]=5)and(R=true))or(Keypressed);
If (op[i]=5)and(R=true) then begin
 Writeln('ZADACHA RESHENA ');
 Writeln('VSEGO SHAGOV= ',i,' ISPITANIE ',k_isp,
 'DLINA PUTI',i/k_isp); delay(50); inc(k_isp); end;
until (k_isp>=100)or(Keypressed); Readkey;
END.
```

```
uses crt, graph;                                        {ПР-2.7}
const M=10000;
{p_op:array[1..5]of real=(0.8,0.7,0.4,0.8,0.6);
t_op:array[1..5]of real=(1.6,2.7,1.4,3.8,2.6); {zad-1}
{p_op:array[1..5]of real=(0.8,0.7,0.4,0.8,0.6);
t_op:array[1..5]of real=(1,1,1,1,1); {zsd-2}
p_op:array[1..5]of real=(0.6,0.6,0.6,0.6,0.6);
t_op:array[1..5]of real=(1,1,1,1,1);
```

```
Var x,D,t,pr,S,tsr,SP:real;
 i,j,n,DV,MV,k:integer;
 nn : array[1..60]of integer;
BEGIN clrscr; k:=0; tsr:=8.31; D:=0;
Repeat n:=0; t:=0; inc(k);
  Repeat t:=t+t_op[n+1]+random(200)/1000-0.1;
    x:=random(1000)/1000;
    If x<p_op[n+1] then inc(n);
  until (n=5)or(KeyPressed);
  S:=S+t; D:=D+sqr(tsr-t);
  For i:=1 to 60 do
    If (t>=i)and(t<i+1) then inc(nn[i]);
until (KeyPressed)or(k>M);
For i:=1 to 20 do writeln(i,' ',nn[i]);
writeln(S/M,' SKO=',sqrt(D/(M-1)));
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For i:=1 to 60 do begin
 rectangle(10*i,450-round(nn[i]/5),10*i+5,450);
 rectangle(10*i+1,451-round(nn[i]/5),10*i+6,450); end;
line(0,450-round(500/5),700,450-round(500/5));
Repeat until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                              {ПР-2.8}
const M=10000;
t_op:array[1..5]of real=(1,1,1,1,1);
Var x,D,t,pr,S,tsr,SP,p_op:real;
 i,j,n,DV,MV,k:integer;
 nn : array[0..101]of integer;
BEGIN
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For j:=1 to 8 do begin p_op:=0.1*j;
k:=0;
For i:=1 to 100 do nn[i]:=0;
Repeat n:=0; t:=0; inc(k);
  Repeat t:=t+t_op[n+1];
    x:=random(1000)/1000;
    If x<p_op then inc(n);
  until (n=5)or(KeyPressed);
  For i:=1 to 100 do
    If (t>=i)and(t<i+1) then inc(nn[i]);
until (KeyPressed)or(k>M);
For i:=1 to 100 do begin
circle(8*i,470-round(nn[i]/8),1);
line(8*i,470-round(nn[i]/8),8*(i-1),470-round(nn[i-1]/8));
line(8*i+1,471-round(nn[i]/8),8*(i-1)+1,471-round(nn[i-1]/8));
end; end;
line(0,470-round(1000/8),700,470-round(1000/8));
line(0,470-round(2000/8),700,470-round(2000/8));
line(0,470,700,470);
Repeat until KeyPressed; CloseGraph;
END.
```

Глава 3.
НЕПРЕРЫВНАЯ МОДЕЛЬ ОБУЧЕНИЯ

Одна из важных проблем кибернетической педагогики состоит в следующем: как, зная параметры ученика, его исходный уровень знаний и воздействие, оказываемое учителем, предсказать знания ученика в последующие моменты времени. Метод имитационного моделирования позволяет создать компьютерную программу, симулирующую поведение системы «учитель – ученик», и исследовать влияние ее параметров на результат обучения.

3.1. Скорость увеличения знаний ученика

Допустим, имеется n учеников, каждый из которых характеризуется набором параметров α_i, β_i, γ_i … ($i=1,2,...,n$), и m учителей, владеющих методами M_1, M_2, M_3 и т. д. **Основная задача дидактики** состоит в том, чтобы так организовать учебный процесс, то есть выбрать методы и распределить изучаемый материал в течение заданного промежутка времени, чтобы в конце обучения учащиеся справились с системой тестов $T=\{T_1, T_2,\ldots\}$. Обозначим через U уровень требований учителя. Сформулируем **закон дидактики**: скорость увеличения знаний Z пропорциональна прилагаемым усилиям $F(t)$ ученика, эффективности методики обучения η, коэффициентам усвоения α и понимания Π: $dZ/dt=\alpha\Pi\eta F(t)$. Можно считать, что прилагаемые усилия ученика F пропорциональны разности между уровнем требований учителя и знаниями учащихся $U-Z$ [21; 24].

Рассмотрим известную аналогию. Конденсатор емкостью C (аналог ученика) подсоединяют к источнику (аналог учителя) с ЭДС E через резистор R. Пусть в начальный момент времени конденсатор заряжен до напряжения U (аналог знаний ученика Z). Тогда через резистор потечет ток $i=dq/dt=$ $=(E-U)/R$. Так как $q=CU$, то $dU/dt=(E-U)/RC$. Скорость увеличения напряжения на конденсаторе (быстрота обучения ученика) пропорциональна разности между ЭДС источника E (требованиями учителя) и напряжением U (знаниями ученика).

Усилия F, прилагаемые учеником, характеризуют интенсивность его мыслительной деятельности и пропорциональны мотивации M. Именно **противоречие между требуемым и имеющимся уровнями знаний ученика** является движущей силой его учебной деятельности. Чем больше ученик знает, тем легче он устанавливает ассоциативные связи и быстрее усваивает новые знания. Мотивация M к обучению зависит от уровня U требований учителя и уровня знаний Z ученика. Величина U равна количеству знаний, сообщенных учителем с целью их усвоения учеником. Не будем различать внешнюю мотивацию, обусловленную требованиями учителя, и внутреннюю мотивацию, вызванную собственным желанием ученика освоить соответствующую дисциплину. Чем больше M (значит, F) и Z ученика, тем легче он устанавливает ассоциативные связи и быстрее усваивает новые знания. Психологи установили, что если уровень предъявляемых требований невысок, то мотивация к обучению отсутствует, учащийся не прилагает усилий. Если уровень требований слишком высок ($U - Z > C$), то прикладываемые усилия F снижаются (график F_1, рис. 3.1). В идеале учитель должен правильно оценивать состояние ученика и предъявлять ему такие требования, при которых его мотивация максимальна. Такой режим обучения, при котором требования U оптимальным образом согласуются с уровнем знаний Z данного ученика, его мотивация максимальна, будем называть **согласованным**.

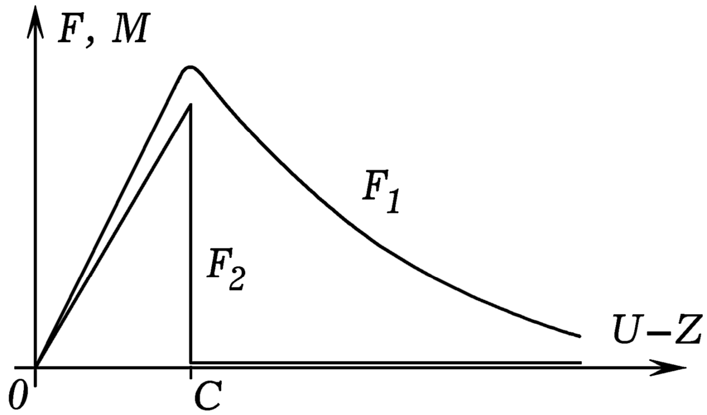


Рис. 3.1. Зависимость усилий от уровня требований и знаний ученика

3.2. Однокомпонентная модель обучения

Проблема имитационного моделирования процесса обучения на основе численного решения дифференциальных уравнений неоднократно обсуждалась различными исследователями [3; 14; 33; 37; 39; 41]. Сначала будем исходить из того, что учебный материал однороден, то есть состоит из равных по сложности элементов. Например, учащийся в течение года пытается запомнить пятьсот слов иностранного языка или изучить факты, составляющие основу курса химии и т. д. В этом случае мы получим однокомпонентную модель обучения, результат которого характеризуется уровнем знаний учащихся Z. Позже будет предложена многокомпонентная модель обучения, учитывающая то, что разные элементы учебного материала усваиваются и забываются неодинаково. Сформулируем принципы, которые следует положить в основу нашей **однокомпонентной модели обучения**:

1. Сообщаемая учащимся информация (знания) является совокупностью равноправных несвязанных между собой элементов, число которых пропорционально ее количеству. Все элементы учебного материала одинаково легко запоминаются и с одинаковой скоростью забываются.

2. Процесс обучения сводится к усвоению знаний и забыванию. Скорость изменения количества знаний равна сумме скорости усвоения ($dZ_у / dt > 0$) и скорости забывания ($dZ_з / dt < 0$):

$$\frac{dZ}{dt} = \frac{dZ_у}{dt} + \frac{dZ_з}{dt}.$$

3. Время усвоения одного элемента учебного материала много меньше длительности обучения. При этом учащийся усваивает всю сообщаемую ему информацию: $dZ_у = dI$, $dZ_у / dt = dI / dt$.

4. Скорость увеличения знаний пропорциональна произведению уровня знаний учащегося Z в степени b $(0 \le b \le 1)$ и мотивации M: $dZ / dt = \alpha M Z^b$. Чем больше ученик знает, тем легче он усваивает новые знания из-за образующихся ассоциативных связей с имеющимися. С другой стороны, чем ниже мотивация учащегося, тем меньше усилий F он затрачивает, тем ниже скорость увеличения знаний. В случае, когда прирост знаний ΔZ много меньше их общего количества Z (обучение в течение одного или нескольких занятий), то можно считать что Z остается постоянным и $b = 0$.

5. Усилия ученика F (мотивация M) прямо пропорциональны разности между уровнем предъявляемых требований U и уровнем знаний Z: $F = M = k(U - Z)$. В случае, когда $U - Z$ превышает некоторый предел C, ученик перестает прикладывать усилия: $F = 0$ (график F_2 на рис. 3.1).

6. Скорость забывания пропорциональна количеству имеющихся у учащегося знаний: $dZ/dt = -\gamma \cdot Z$**.**

Учитывая перечисленные выше соображения, получаем, что скорость увеличения знаний выражается уравнением:

$$\frac{dZ}{dt} = \alpha F Z^b - \gamma \cdot Z ,$$

где α и γ – коэффициенты научения и забывания конкретного ученика, а b зависит от влияния имеющихся знаний на усвоение новой информации.

Известно, что прилагаемые усилия F зависят от уровня U требований учителя (количества знаний, которые необходимо усвоить) и уровня знаний Z ученика: $F = F(U,Z)$. Зависимость F от $U - Z$ изображается кривой F_2 на рис 3.1. Чем больше F и знания Z учащегося, тем легче он устанавливает ассоциативные связи и быстрее усваивает новые знания. Итак, скорость изменения знаний ученика:

$$\frac{dZ}{dt} = \begin{cases} \alpha Z^b (U - Z) - \gamma \cdot Z, & U \le Z + C, \\ -\gamma \cdot Z, & U > Z + C. \end{cases}$$

Когда Z мало, скорость роста уровня знаний невысока из-за отсутствия возможности образования ассоциативных связей. По мере увеличения Z она растет, но при $Z \to U$ уменьшается за счет снижения усилий F (мотивации M). Если U превышает Z на величину, большую критического значения C, то ученик перестает учиться.

Особый интерес представляет собой случай, когда скорость усваивания учебной информации постоянна ($\upsilon = dI/dt = dZ_y/dt = const$). При этом:

$$\frac{dZ}{dt} = \upsilon - \gamma \cdot Z , \quad \int_0^t \frac{dZ}{\upsilon - \gamma \cdot Z} = \int_0^t dt , \quad Z(t) = \frac{\upsilon}{\gamma}\left(1 - \exp(-\gamma \cdot t)\right) + Z_0 \exp(-\gamma \cdot t) .$$

3.3. Зависимость результата обучения от параметров ученика и требований учителя

Кибернетическая педагогика исходит из того, что система «учитель – ученик» является самоадаптирующейся. В ней регулируемая величина (уровень знаний учащегося Z) стремится достичь заданной величины (уровня требований учителя U). Если $Z < U$, то учащиеся увеличивают свои знания так, чтобы они соответствовали уровню требований. Создадим компьютерную модель, которая, исходя из параметров ученика, его исходного уровня знаний и «траектории обучения», позволяла бы предсказать знания ученика в последующие моменты времени. Запишем дифференциальное уравнение $dZ/dt = \alpha Z(U-Z) - \gamma \cdot Z$ в конечных разностях: $\Delta Z = \alpha Z(U-Z)\Delta t - \gamma \cdot Z\Delta t$. Используется программа ПР–3.1 (Приложение к главе 3). Она содержит цикл по времени t, в котором вычисляется скорость увеличения знаний, определяется уровень знаний в следующий дискретный момент времени $t+1$, строится соответствующая точка графика, после чего все повторяется снова. Упрощенная блок-схема алгоритма представлена на рис. 3.2.

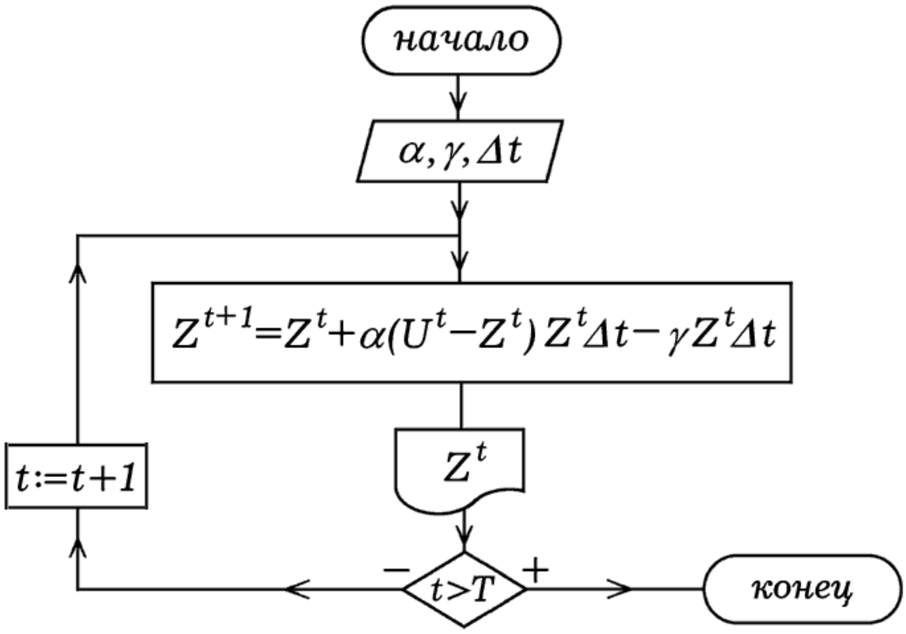


Рис. 3.2. Алгоритм имитационного моделирования процесса обучения

Ситуация 1. Два учащихся с различными коэффициентами научения α_1, α_2 и одинаковыми коэффициентами забывания γ одновременно проходят обучение у одного учителя, который в течение одного промежутка времени t' предъявляет один и тот же уровень требований U. Необходимо промоделировать изменение знаний учащихся $Z_1(t)$ и $Z_2(t)$.

Используется программа ПР–3.1. Результаты моделирования обучения двух учеников с различными коэффициентами научения α в случае, когда обучение проводилось в течение времени t', а затем прекратилось, представлены на рис. 3.3.1. Видно, что уровень знаний учащихся при обучении растет, достигает максимума, а после прекращения обучения уменьшается по экспоненциальному закону. Пунктиром показан вид графиков в случае, если бы обучение не прекращалось. Учащийся с коэффициентом научения $\alpha = 0{,}03$ за время t' не успевает достичь максимального уровня знаний.

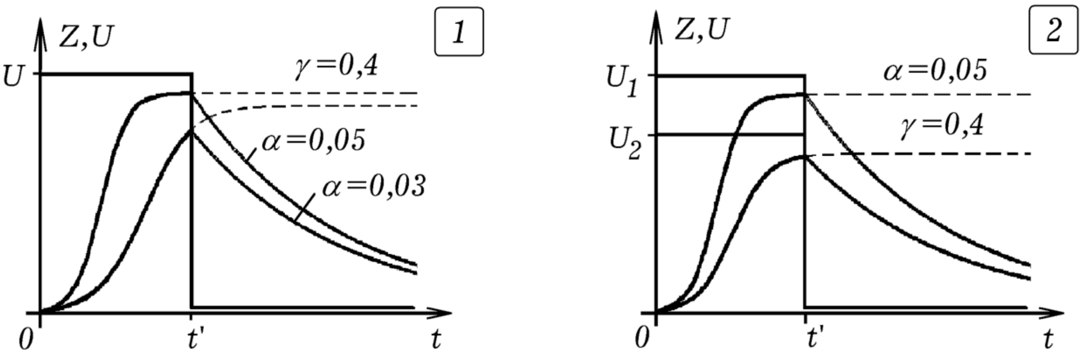


Рис. 3.3. Результаты моделирования обучения

Ситуация 2. Два примерно одинаковых учащихся (с равными α и γ) в течение одного и того же промежутка времени t' проходят обучение. Учитель предъявляет к учащимся различные требования U_1 и U_2, при которых мотивация к обучению не пропадает. Необходимо получить графики $Z_1(t)$ и $Z_2(t)$.

Результаты решения задачи представлены на рис. 3.3.2. Видно, что тот учащийся, которому предъявляются более высокие требования U_1, учится быстрее и достигает более высоких знаний. Уровень знаний в обоих случаях достигает максимального значения, которое зависит от требуемого уровня U. Существенно то, что уровень требований не настолько высок, чтобы пропала мотивация к обучению.

Ситуация 3. Два учащихся с различными коэффициентами научения $\alpha_1 = 0{,}05$ и $\alpha_2 = 0{,}03$ изучают некоторый курс, причем уровень требований возрастает так, что второй учащийся не успевает его «догнать» и теряет мотивацию к обучению. Получим графики $Z_1(t)$ и $Z_2(t)$.

Предположим, что уровень требований растет по закону $U = 0{,}0002t^2$. Имитационное моделирование дает результаты, которые приведены на рис. 3.4.1. Сначала оба учащихся отвечают требованиям учителя (интервал $[0; t_1]$). Уровень требований растет все быстрее и быстрее, поэтому в момент t_1 учащийся 2 с низким α отстает так сильно, что его мотивация M падает до 0, в то время как учащийся 1 продолжает соответствовать требованиям учителя. В момент t_2 учитель «отрывается» от обоих учеников, предъявляет слишком высокие требования, и учащиеся перестают учиться. Учитель, заметив снижение мотивации, должен принять меры и уменьшить уровень требований U.

Ситуация 4. Уровни требований учителя, соответствующие оценкам 5 и 4, растут по заданным законам $U_4(t)$ и $U_5(t)$. Коэффициенты научения α забывания γ ученика известны. Промоделируем ситуацию, когда сначала учащийся претендует на оценку 5, но уровень требований растет слишком быстро, и учащийся вынужден снизить уровень своих притязаний до оценки 4.

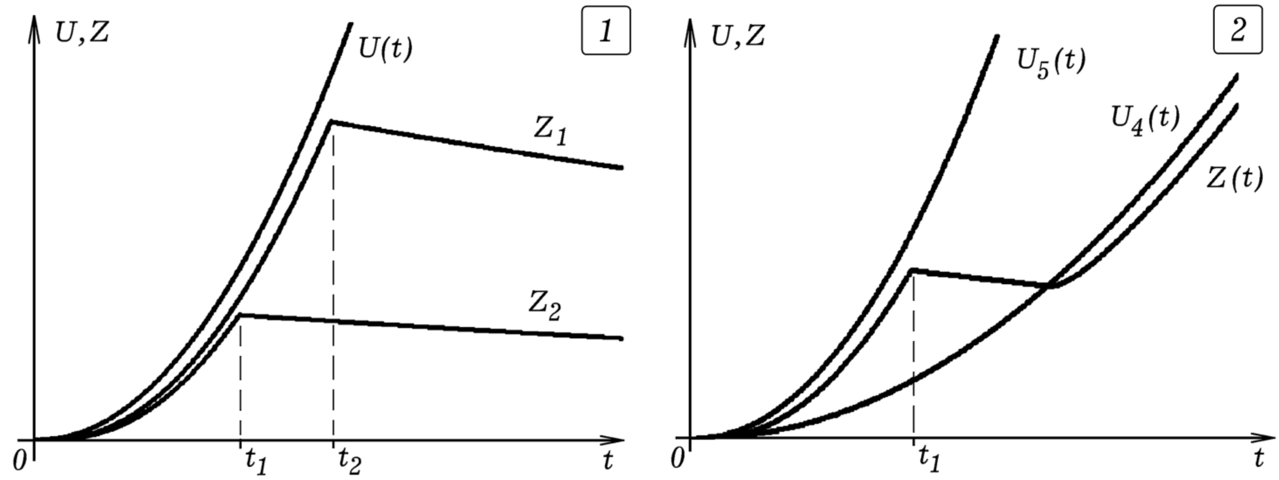


Рис. 3.4. Уровень требований учителя плавно увеличивается

Проанализируем результаты моделирования (рис. 3.4.2). В момент t_1 разрыв между знаниями ученика и требованиями учителя превышает критическое значение, и ученик «перестает бороться» за оценку 5. Уровень требований, соответствующий оценке 4, растет медленнее, поэтому учащийся успевает за ним.

Ситуация 5. Два ученика с различными коэффициентами научения $\alpha_1 = 0{,}05$ и $\alpha_2 = 0{,}03$ изучают некоторый курс, причем уровень требований возрастает по закону: $U = kt^2 + C$. При этом сначала второй, а затем первый ученики не успевают его «догнать» и теряют мотивацию к обучению. Учитель, оценивая результаты их работы, снижает уровень требований, «подхватывая» обоих учеников и снова плавно повышая уровень требований. Ученики какое-то время учатся, затем снова отстают от учителя, теряя мотивацию к обучению. Учитель, обнаружив это, снова резко снижает уровень предъявляемых требований, создавая при этом условия для их успешной работы, и т. д.

Используется программа ПР–3.2, результаты моделирования приведены на рис. 3.5. В моменты времени t_1, t_3 и t_5 учитель, увидев снижение мотивации обоих учеников, резко снижает уровень требований так, что ученики снова чувствуют интерес к обучению. В моменты t_2, t_4 и t_6 первый ученик снова начинает учиться, его знания начинают увеличиваться. В процессе обучения уровни знаний $Z_1(t)$ и $Z_2(t)$ учащихся в среднем растут с некоторой скоростью. Максимальная скорость увеличения знаний учеников зависит от их способности усваивать новую информацию (коэффициента научения α), максимальной мотивации к обучению и способности учителя реагировать на результаты работы учеников.

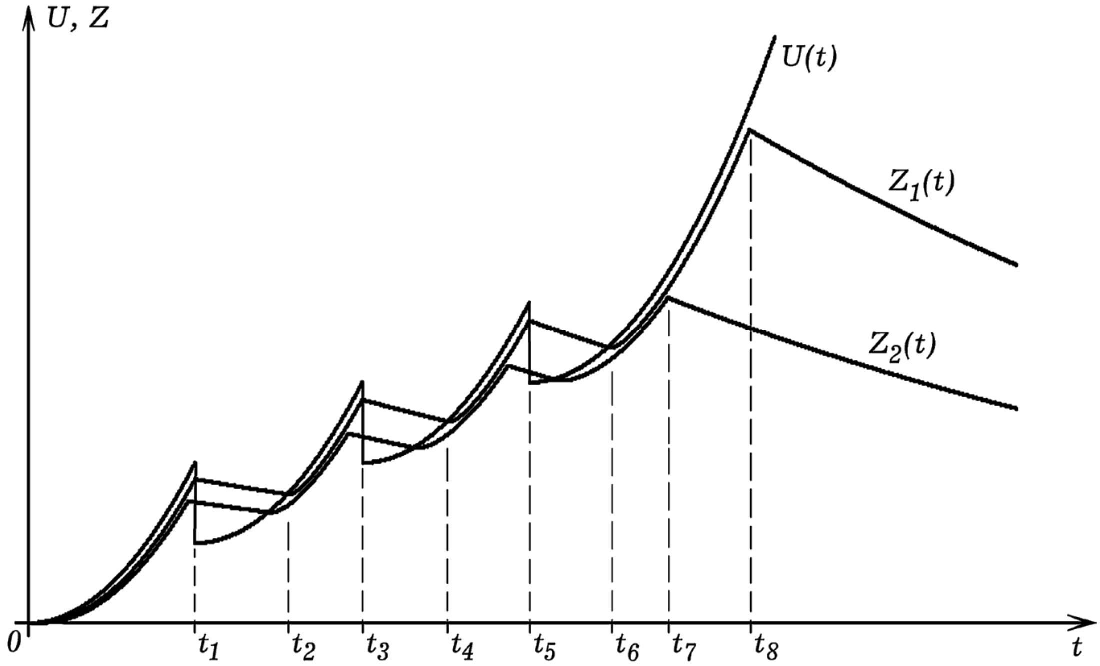


Рис. 3.5. Учитель периодически снижает уровень требований U

Ситуация 6. Промоделируйте предыдущую ситуацию для одного ученика, сделав так, чтобы программа самостоятельно выбирала моменты времени, когда учителю необходимо снизить уровень требований. Необходимо учесть, что учитель с запаздыванием τ реагирует на результаты работы ученика.

Система «учитель – ученик» является самоадаптирующейся. Промоделировать «автоматическую подстройку» учителя под ученика можно с программы ПР–3.3 (рис. 3.6). «Отрыв» ученика от требований учителя происходит в моменты t_1, t_4 и т. д. Учитель с запаздыванием τ обнаруживает, что учащийся не усваивает сообщаемую ему информацию и снижает уровень предъявляемых требований до величины $U_0 = 0{,}9Z$ (моменты t_2, t_5), а затем снова увеличивает U. Средняя скорость обучения не зависит от закона увеличения $U(t)$, а определяется и коэффициентом научения слабого ученика. Анализ представленных на рис. 3.5 и 3.6 графиков позволяет предположить возможность согласованной работы учителя и ученика, при которой средняя по времени мотивация ученика M , значит, и усилия F, прилагаемые им, стабилизируются на достаточно высоком уровне.

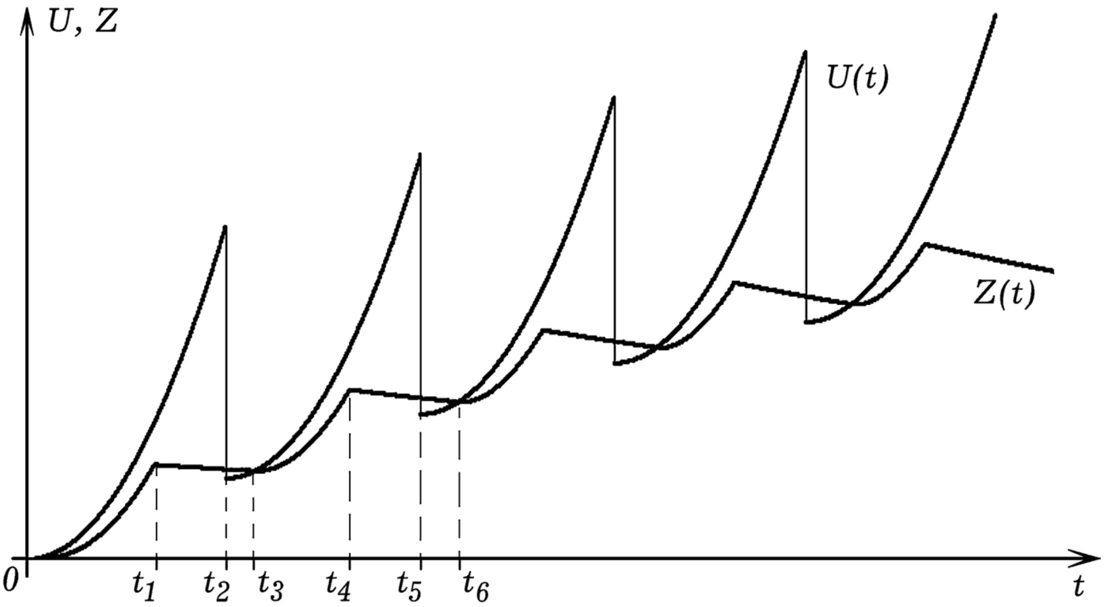


Рис. 3.6. Моделирование самоадаптирующейся системы «учитель – ученик»

Метод компьютерного моделирования позволяет исследовать и более сложные модели обучения. Например, можно считать, что мотивация учащихся к учебной деятельности связана с уровнем требований следующим образом:

$$M = \begin{cases} (U - Z), \quad если \quad Z < U < Z + C, \\ C\exp[-\beta(U - Z - C)], \quad если \quad U \geq Z + C. \end{cases}$$

То есть если $U - Z < C$, то $M = U - Z$, а в случае, когда , $U \geq Z + C$ мотивация плавно снижается по экспоненте $M = C\exp[-\beta(U - Z - C)]$.

Используя компьютерную модель обучения, можно обосновать известный принцип «от простого к сложному». Допустим, сначала изучается сложная тема, а затем простая, то есть сначала уровень требований учителя высокий, а затем низкий ($U_1 > U_2$). Если U_1 очень сильно превосходит уровень знаний Z ученика, то мотивация к обучению снижается, и уровень знаний не растет (ученик просто не может усвоить материал). Если же ученик усвоил сложную тему, то при изучении второй более простой темы скорость роста знаний невысока за счет того, что уровень требований U незначительно превосходит уровень знаний Z ученика, и он не затрачивает много усилий.

На рис. 3.7 показано, как ведет себя рассмотренная выше модель обучения (программа ПР–3.4), когда уровень U предъявляемых требований (уровень сложности задания) скачкообразно увеличивается. Сначала учащимся предлагают сравнительно простые задания, а когда они их освоят, то задания предлагаются сложнее, затем еще сложнее и т. д. Чтобы уровень знаний рос, необходимо обеспечить не очень большой разрыв между Z и U. Слишком резкое увеличение уровня требований (сложности изучаемого материала) приводит к снижению мотивации и уменьшению уровня знания вследствие забывания (рис. 3.7.2). Если сначала предложить сложные задания (уровень требований U высок), а затем простые, то обучение происходить не будет. Итак, для оптимизации учебного процесса **необходимо подбирать уровень требований таким образом (сложность предлагаемых учащимся заданий), чтобы сохранялась высокая мотивация к обучению**.

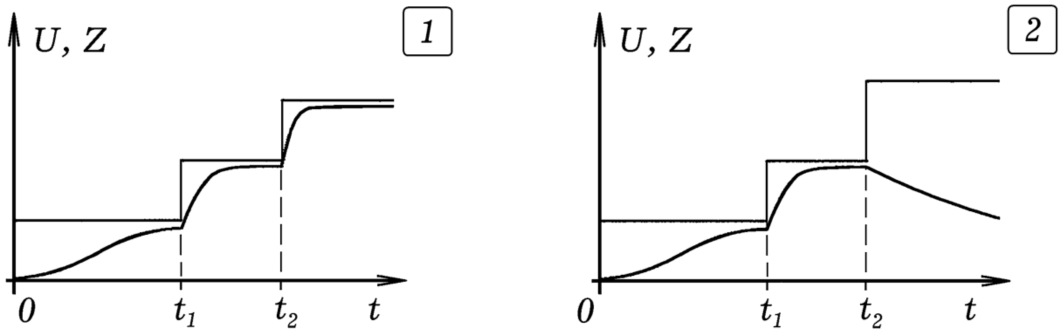


Рис. 3.7. Обучение при скачкообразном повышении уровня требований

3.4. Результаты имитационного моделирования изучения несвязанных между собой тем

С помощью имитационного моделирования проанализируем несколько ситуаций, связанных с изучением тем различной сложности, которые не связаны между собой так, что изучение одной темы не влияет на усвоение другой.

Ситуация 1. Группа учащихся изучает дисциплину из четырех независимых тем. Каждая тема заканчивается контрольной работой, а в конце курса предусмотрен экзамен. Уровень требований по каждой теме растет по закону: $U_i = 0{,}05\tau$; длительности изучения каждой темы заданы: T={90; 200; 120; 210}. Получим зависимость знаний среднестатистического ученика от времени.

Используется программа ПР–3.5. Результаты компьютерного моделирования представлены на рис. 3.8.1. Видно, что при изучении каждой темы уровень требований растет пропорционально времени. На каждом следующем занятии внутри одной темы учитель требует овладения новыми знаниями и сохранения знаний, полученных ранее. Изучение темы заканчивается проведением контрольной работы, после чего учитель вспоминает о ней только на экзамене. Во время изучения i -й темы увеличивается количество знаний Z_i по этой теме, а после окончания Z_i уменьшается по экспоненциальному закону из-за забывания. При подготовке к экзамену учащиеся вспоминают материал по всем темам, их знания увеличиваются. После экзамена уровни знаний Z_i уменьшаются вследствие забывания.

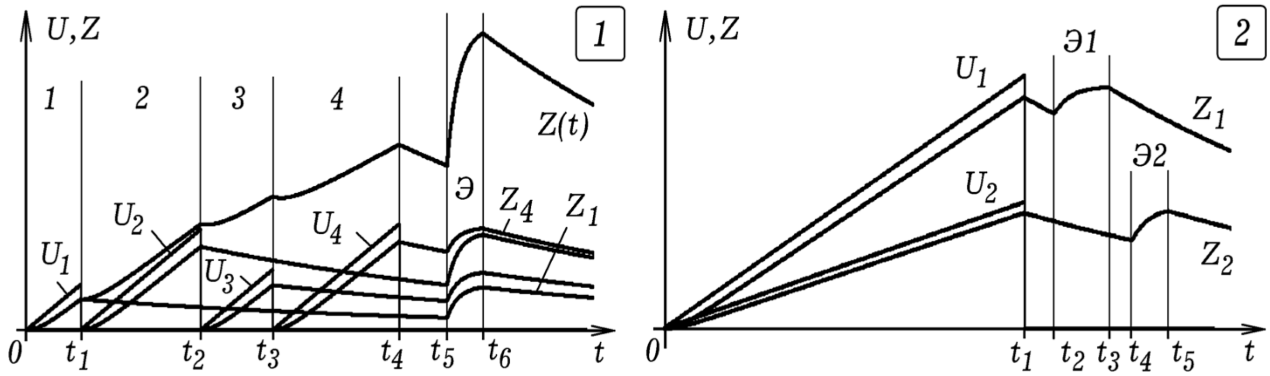


Рис. 3.8. Изучение курса, заканчивающегося экзаменами

Ситуация 2. Ученик в течение нескольких месяцев одновременно изучает два курса. Учитель на каждом последующем занятии требует знания всего предыдущего материала, причем уровень требований растет пропорционально времени t . В конце семестра предусмотрены экзамены по всем вопросам [22].

Как показывает имитационное моделирование (программа ПР–3.6), количество знаний Z_1 и Z_2 изменяется так. В течение семестра (от 0 до t_1) уровень знаний монотонно растет, затем несколько уменьшается (рис. 3.8.2). При подготовке к экзаменам (от t_2 до t_3 и от t_4 до t_5) уровень соответствующих знаний снова возрастает, а после сдачи экзамена – убывает.

Ситуация 3. Учащийся изучает тему, состоящую из шести параграфов. Учитель требует, чтобы к концу обучения учащийся усвоил весь учебный материал. Пусть к концу первой недели учитель выдвигает требования U_1: выучить 15 иностранных слов. К концу второй недели требования увеличиваются до уровня U_2: следует выучить новые 20 слов и не забыть ранее выученные 15 слов и т. д. Получим зависимость уровня знаний ученика от времени.

Понятно, что обучение будет результативным, если уровень требований U в каждый момент времени превышает уровень знаний учащегося Z на величину, меньшую C. Используется программа ПР–3.7. Получающиеся графики представлены на рис. 3.9. После окончания обучения уровень требований снижается до 0, количество знаний учащегося уменьшается по экспоненциальному закону вследствие забывания.

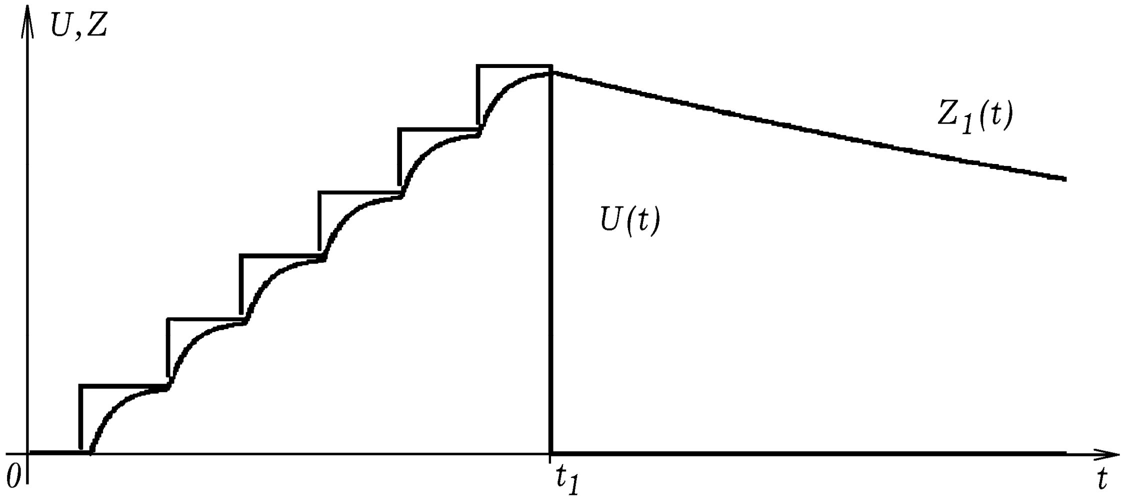


Рис. 3.9. Уровень требований скачкообразно растет, затем резко падает

Ситуация 4. При изучении некоторой темы учащиеся на восьми уроках получают знания двух типов: 1) знания Зн–1, которые после изучения используются на последующих уроках; 2) знания Зн–2, которые изучаются один раз и больше не используются. Уровни требований, равные количеству знаний, которые необходимо усвоить на каждом уроке, известны: U_1= (30, 60, 90,120, 150), U_2 = (70, 70, 70, 70, 70). Создадим имитационную модель обучения [22].

Учитель в течение i -й недели сообщает U_{1i} знаний Зн–1 и U_{2i} знаний Зн–2, требуя их полного усвоения, а в конце проводит контрольную работу по этому материалу. Так же проходят $(i+1)$-я неделя и т. д. Уровень требований U_{1i} по знаниям Зн–1 с каждой неделей скачкообразно увеличивается: ученик должен помнить информацию, полученную на предыдущих уроках и изучаемую в течение текущей недели. Уровень требований U_{2i}, касающийся знаний Зн–2, по мере изучения курса остается постоянным.

Используется программа ПР–3.8. Результаты моделирования представлены на рис. 3.10.1. Видно, что уровень знаний Зн–1 по мере изучения курса возрастает, в то время как уровень знаний Зн–2 остается практически постоянным. С некоторого момента для знаний Зн–2 наступает динамическое равновесие: среднее количество знаний, приобретаемых учеником за достаточно большое время, оказывается равным количеству знаний, забываемых им за то же время.

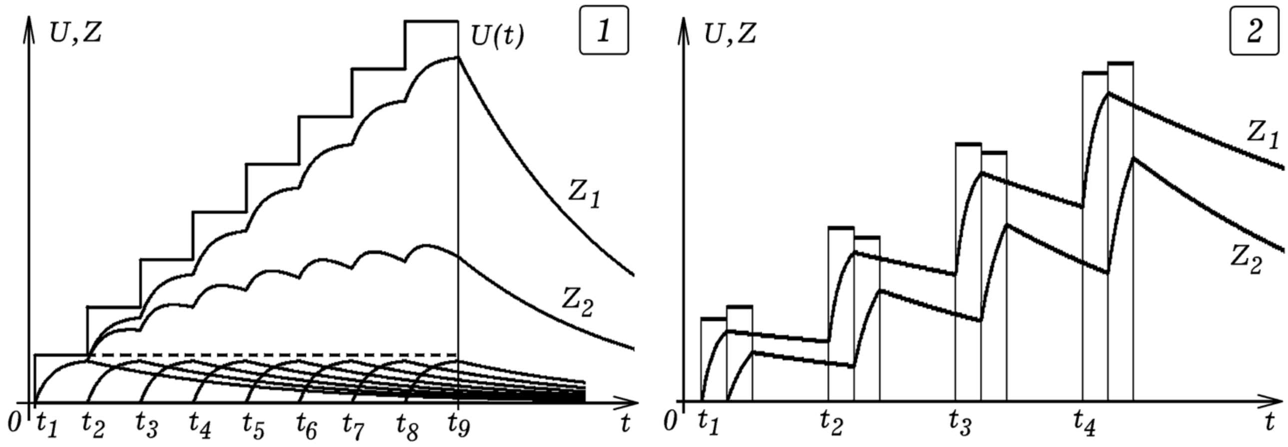


Рис. 3.10. Имитационное моделирование процесса обучения

Ситуация 5. Учащийся в течение четырех недель посещает уроки по предметам 1 и 2 (физика и литература), следующие друг за другом раз в неделю. Известно количество знаний U_i, которое должен усвоить ученик, а его коэффициенты научения α и забывания γ по предмету 1 равны 0,025 и 0,0005, а по предмету 2 – 0,012 и 0,001. Необходимо исследовать изменение уровня знаний учащегося по мере изучения обоих курсов [22].

Используется программа ПР–3.9. Результаты моделирования представлены на рис. 3.10.2. В цикле по времени отдельно вычисляются Z_1 и Z_2, результаты выводятся на экран в виде графиков. Видно, как во время занятий растет уровень знаний по предмету 1 и предмету 2. В перерывах между занятиями уровень знаний снижается вследствие забывания.

3.5. Изучение нескольких тем, имеющих различную сложность

Теперь рассмотрим ситуацию, когда элементы учебного материала не являются независимыми и равноправными. Некоторые из вопросов могут иметь большую сложность, а их изучение может требовать усвоения каких-то других вопросов. Допустим, учитель должен обеспечить изучение учащимися курса, состоящего из n тем. При изучении каждой темы уровень требований нарастает; учащиеся готовятся к контрольной работе, состоящей из нескольких заданий. В конце курса проводится экзамен. В общем случае изучаемые темы имеют различную сложность S и между собой связаны: усвоение i-й темы уменьшает трудность понимания j-й темы. Задано время T_k ($k = 1,2,..., n$), отводимое на каждую тему. Построим компьютерную модель процесса обучения.

Учтем, что **трудность** (или **субъективная сложность** S_r) изучаемой r-й темы курса может зависеть от уровня знаний Z_k учащимся k-й темы. Пусть эта зависимость выражается так: $S_r = a + b\exp(-cZ_k)$, где a, b, $c > 0$, $a + b \le 1$, $b < a$. Сложность темы лежит в интервале $[0, 1]$. Минимальная сложность $S = 0$ соответствует очень простой теме, максимальная $S = 1$ – теме, которую ученик не может понять в принципе (требуется очень большое время). Получается уравнение [22]:

$$\frac{dZ_{ij}}{dt} = \alpha_j (1 - S_{ij}) Z_{ij} (U_i - Z_{ij}) - \gamma_j Z_{ij};\ Z_{ij} > 0;\ U_i > Z_{ij};\ \alpha_j,\ \gamma_j > 0,$$

где Z_{ij} – уровень знаний j-ым учеником i-й темы, U_i – предъявляемый уровень требований, то есть количество знаний, сообщенных учителем, которые должен усвоить ученик. В конечных разностях получаем:

$$Z_{ij}^{t+1} = Z_{ij}^t + (\alpha_j (1 - S_{ij}) Z_{ij}^t (U_i - Z_{ij}^t) - \gamma_j Z_{ij}^t) \Delta t.$$

После изучения темы учитель проводит контрольную работу (тест) из m задач $K = \{z_1(1),\ z_2(1,2),\ \ldots,\ z_m(4,5)\}$. Если для решения задачи $z_k(i)$ достаточно знаний i-й темы, то результат (или вероятность ее решения R_k) равен уровню усвоения учащимся этой темы: $R_k = Z_i / U_i$. Если k-я задача $z_k(r, s)$ комбинированного типа и требует знаний r-й и s-й темы, то по закону умножения вероятностей $R_k = (Z_r / U_r)(Z_s / U_s)$.

Ситуация 1. Студент изучает курс, состоящий из пяти не связанных между собой тем, которые излагаются последовательно друг за другом: 1, 2, 3, 4, 5. Сложности S_i тем и отводимое на изучение время T_i заданы двумя матрицами: S = (0,3; 0,1; 0,4; 0,7; 0,2) и T = (1,2; 1,7; 1,5; 1,8; 2,4). В конце курса проводится контрольная работа из пяти задач: $K = \{z_1(1),\ z_2(1,2),\ z_3(2,3),\ z_4(3,4),\ z_5(3,5)\}$. Создадим имитационную модель этого процесса [22].

Программа ПР–3.10, моделирующая процесс обучения, должна содержать цикл по времени t, в котором вычисляется скорость увеличения знаний, определяется уровень знаний в следующий момент времени $t+1$, результаты выводятся на экран, после чего все повторяется снова. Используемая компьютерная программа содержит цикл по времени, в котором пересчитываются значения Z[i] для каждой i -й темы ($i=1,2,...,5$). В программе определяется суммарный уровень знаний. Результаты выводятся на экран в графическом виде (рис. 3.11.1). Оценка за контрольную работу рассчитывается по формуле: $R=(Z_1+Z_1Z_2+Z_2Z_3+Z_3Z_4+Z_3Z_5)/5$. На рис. 3.11.1 оценка R соответствует отрезку в правой части графика. При других значениях S_i и T_i получаются графики, представленные на рис. 3.11.2.

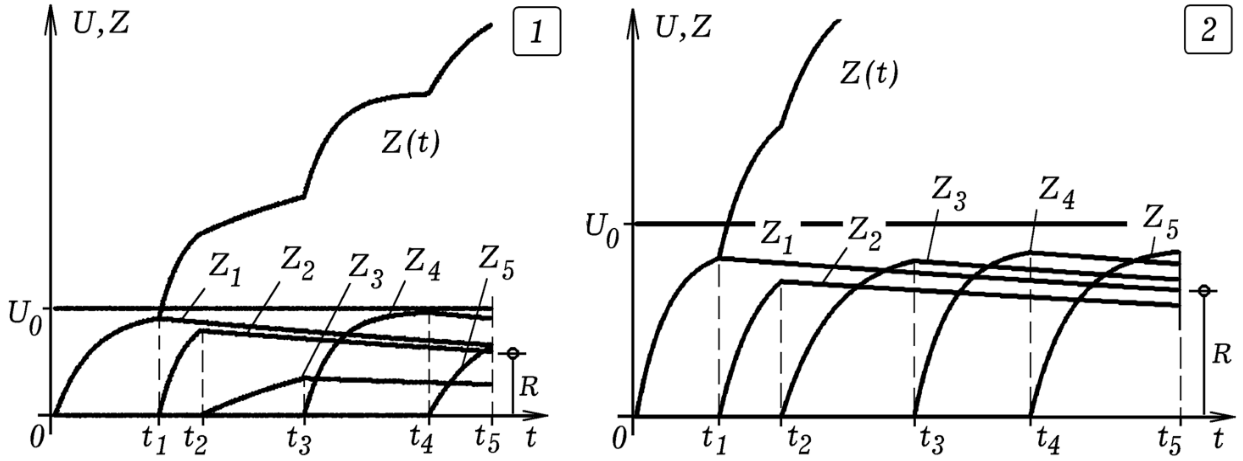


Рис. 3.11. Моделирование изучения тем, имеющих различную сложность

3.6. Учет изменения работоспособности ученика в течение дня

Скорость увеличения знаний ученика пропорциональная его коэффициенту научения α, работоспособности r, приложенным усилиям F (мотивации M) и уровню знаний Z в степени β ($0 \le \beta \le 1$): $dZ/dt = r\alpha F Z^b - \gamma \cdot Z$, где r и γ – коэффициент работоспособности, зависящий от степени усталости ученика, и коэффициент забывания. Величина r по мере совершения учеником работы P сначала равна r_0 ($0 < r_0 \le 1$), а затем плавно снижается до 0 по закону: $r = r_0/(1+\exp(k_1(P-P_0)))$. Здесь P_0 – работа, совершаемая учеником на занятии (решение задач, выполнение заданий), при которой его работоспособность уменьшается от $r_0 = 1$ до $r = 0{,}5$. При обучении уровень требований учителя (сообщаемые им знания) больше уровня знаний ученика ($U > Z$), и учебная работа, совершенная учеником (число выполненных заданий), зависит от приложенных усилий (интенсивности мыслительной деятельности) и длительности обучения. Усилия ученика F пропорциональны его мотивации или разности между уровнем требований U учителя и количеством знаний Z:

$$F = U - Z, \quad \Delta P = k_2 F \Delta t = k_2 (U - Z)\Delta t, \quad P = k_2 \sum_{i=1}^{N} F_i \Delta t.$$

Здесь N – число элементарных промежутков времени, на которое разбит урок. Если уровень предъявляемых требований мал ($U \le Z$), то есть ученик на уроке занят решением простых для него задач, то затрачиваемые им усилия пропорциональны времени: $P = kt$. Это позволяет учесть появление у ученика усталости и снижения работоспособности даже в случае, когда он выполняет простые задания длительное время. В перерывах между занятиями ученики отдыхают, работоспособность восстанавливается по экспоненциальному закону:

$$dr / dt = k_3 (r_{\max} - r), \quad r(t) = r_{\max} - (r_{\max} - r_0)\exp(-k_3 (t - t_0)),$$

где $r_0 = r(t_0)$ – работоспособность в момент начала отдыха t_0, где $r_{\max}$ – максимальная работоспособность ученика в данное время t учебного дня. Она плавно снижается по закону $r_{\max} = \exp(-k_4 t)$. Скорость увеличения знаний при прочих равных условиях тем выше, чем меньше субъективная сложность (трудность понимания) S изучаемого материала: $dZ / dt = r(1 - S)\alpha F Z^b$. Сложность учебного материала S лежит в интервале от 0 до 1 и в общем случае зависит от уровня изучения других вопросов. Итак, получается модель [30]:

Во время обучения ($U > Z$): $$\frac{dZ}{dt} = \frac{(1 - S)\alpha F Z^b}{1 + \exp(k(P - P_0))} - \gamma \cdot Z,$$

Во время перерыва ($U = 0$): $$dZ / dt = -\gamma \cdot Z.$$

Пусть учитель так организует процесс обучения в течение дня, что ученики работают с максимальным напряжением $F = const$. Прирост знаний существенно меньше общего количества знаний ученика Z, поэтому $b = 0$. Проводится 5 уроков одинаковой длительности $T_u = t_1 = t_2 - t'_1 = \ldots = t_5 - t'_4$, разделенных перерывами длительностью $T_u = t'_1 - t_1 = t'_2 - t_2 = \ldots = t'_4 - t_4$. Результаты имитационного моделирования представлены на рис. 3.12 (программа ПР–3.11). Параметры модели подобраны так, чтобы ее поведение соответствовало реальной ситуации. В интервале от 0 до t_5 коэффициент работоспособности r совершает колебания относительно плавно уменьшающегося значения. При уменьшении длительности перерывов между занятиями учащиеся не успевают восстановить свою работоспособность, результаты обучения снижаются [30].

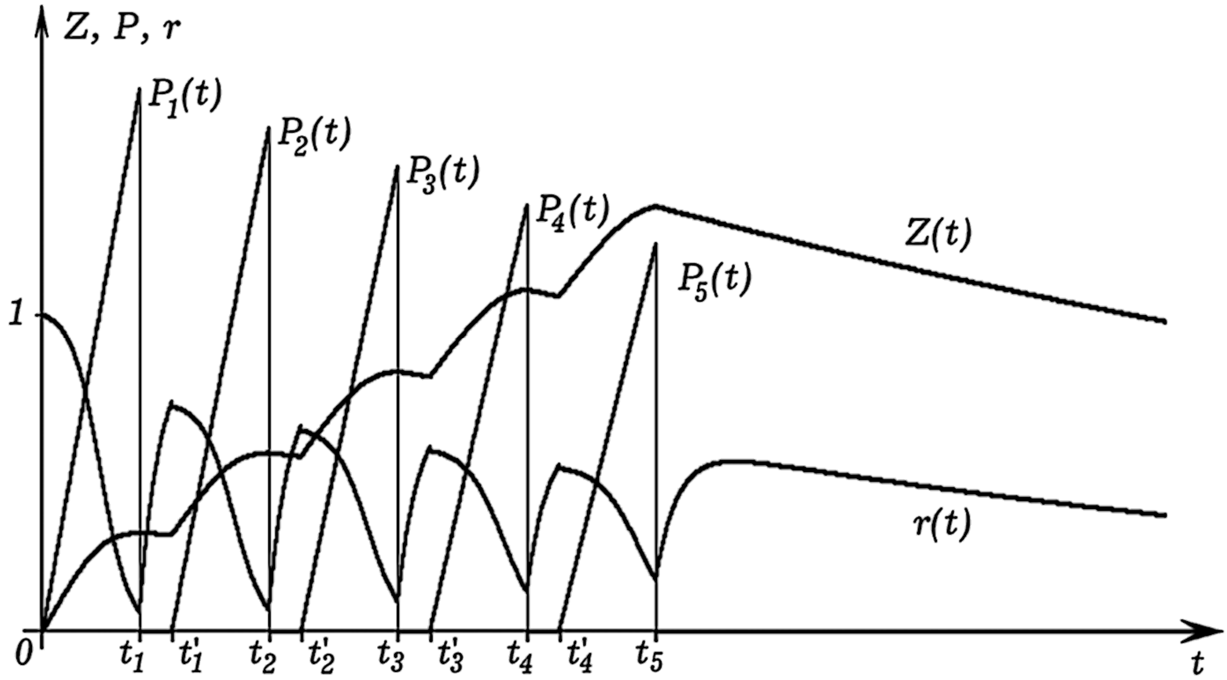


Рис. 3.12. Модель, учитывающая изменения работоспособности ученика

3.7. Многокомпонентная модель обучения

Как известно, процесс усвоения и запоминания сообщаемой информации состоит в установлении ассоциативных связей между новыми и имеющимися знаниями. В результате приобретенные знания становятся более прочными и забываются значительно медленнее. Предлагаемая многокомпонентная модель обучения выражается системой уравнений [24; 29]:

$$dZ_1 / dt = k\alpha_1 (U - Z) Z^b - k\alpha_2 Z_1 - \gamma_1 Z_1,$$
$$dZ_2 / dt = k\alpha_2 Z_1 - k\alpha_3 Z_2 - \gamma_2 Z_2,$$
$$dZ_3 / dt = k\alpha_3 Z_2 - k\alpha_4 Z_3 - \gamma_3 Z_3,$$
$$dZ_4 / dt = k\alpha_4 Z_3 - \gamma_4 Z_4,$$

где U – уровень требований, предъявляемый учителем, равный сообщаемым знаниям Z_0, Z – суммарные знания, Z_1 – самые «непрочные» знания первой категории с высоким коэффициентом забывания γ_1, а Z_4 – самые «прочные» знания четвертой категории с низким γ_4 ($\gamma_4 < \gamma_3 < \gamma_2 < \gamma_1$). Коэффициенты усвоения α_i характеризуют быстроту перехода знаний $(i-1)$-й категории в знания i -й категории. Пока происходит обучение, $k=1$, а когда оно прекращается, $k=0$. Если прирост знаний ученика существенно меньше их общего количества, то $b=0$. Коэффициент забывания $\gamma = 1/\tau$, где τ – время, в течение которого количество знаний i -й категории уменьшается в $e = 2{,}72...$ раза. Результат обучения характеризуется суммарным уровнем приобретенных знаний $Z = Z_1 + Z_2 + Z_3 + Z_4$ и коэффициентом «прочности» $\mathrm{Pr} = (Z_2/4 + Z_3/2 + Z_4)/Z$. При изучении одной темы сначала растет уровень знаний Z, затем происходит увеличение доли «прочных» знаний Z_4 и повышается прочность Pr.

Ситуация 1. Учитель проводит три урока, уровень требований $U(t)$ в течение урока растет пропорционально времени или остается постоянным. Необходимо проанализировать процесс обучения ученика с помощью двухкомпонентной и четырехкомпонентной моделей.

Двухкомпонентная модель обучения выражается уравнениями:

$$dZ_1/dt = k\alpha_1(U - Z) - k\alpha_2 Z_1 - \gamma_1 Z_1,$$

$$dZ_2/dt = k\alpha_2 Z_1 - \gamma_2 Z_2,\ Z = Z_1 + Z_2.$$

Используется программа ПР–3.12; результаты моделирования приведены на рис. 3.13.1. Учитель проводит три урока, в течение которых уровень требований растет пропорционально времени: $U = a(t - t_0) + b$. Видно, что во время перерывов и после обучения уровень «непрочных» знаний Z_1 быстро уменьшается, а прочные знания Z_2 забываются существенно медленнее. При использовании четырехкомпонентной модели обучения (программа ПР–3.13) получаются похожие результаты (рис. 3.13.2). Уровень требований $U(t)$ в течение занятия остается постоянным.

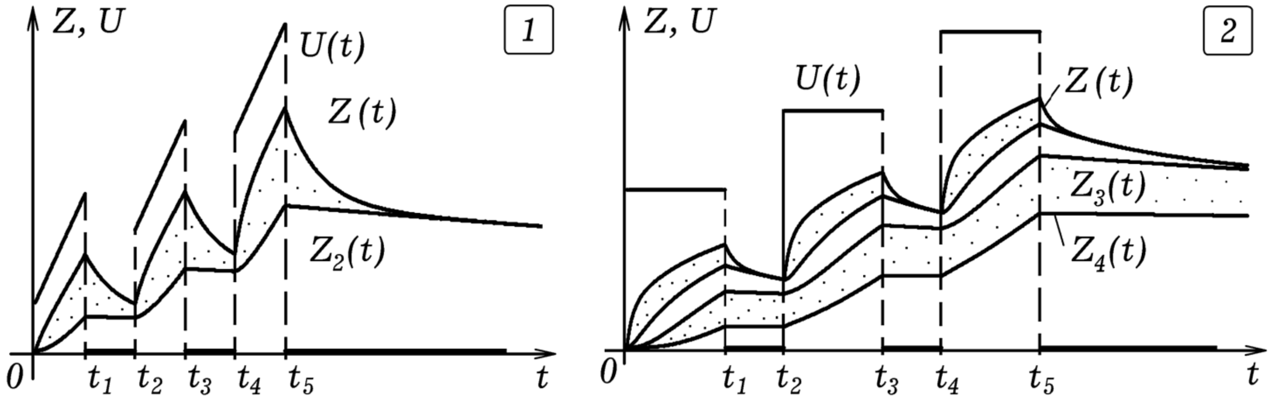


Рис. 3.13. Двухкомпонентная и четырехкомпонентная модели обучения

Ситуация 2. Учитель должен обучить ученика решать 10 задач возрастающей сложности $\theta_i = i\Delta\theta$, которая равна количеству знаний Z, требующихся для решения i -й задачи. Учитель располагает задачи в порядке возрастания сложности и задает их ученику через равные промежутки времени Δt. Если ученик не решил i -ю задачу, то учитель обучает его в течение времени Δt, а затем снова предлагает эту же или аналогичную задачу той же сложности θ_i. Если уровень знаний ученика Z больше θ_i, то ученик, вероятнее всего, решит задачу в течение Δt. При этом Z не увеличится, но часть «непрочных» знаний становится «прочными». После этого учитель предъявляет ему (i +1)-ю задачу с более высоким уровнем сложности θ_{i+1}. Если у ученика знаний недостаточно, то с большой вероятностью он не сможет решить задачу сразу. Учитель в течение времени Δt объясняет материал, либо ученик занимается по учебнику. Уровень требований $U = \theta_i$, Z_1 и Z_2 растет. Затем ученик снова пробует решить задачу. Занятия длительностью $T_З >> \Delta t$ чередуются переменами продолжительностью $T_П >> \Delta t$. Промоделируем эту ситуацию [24; 29].

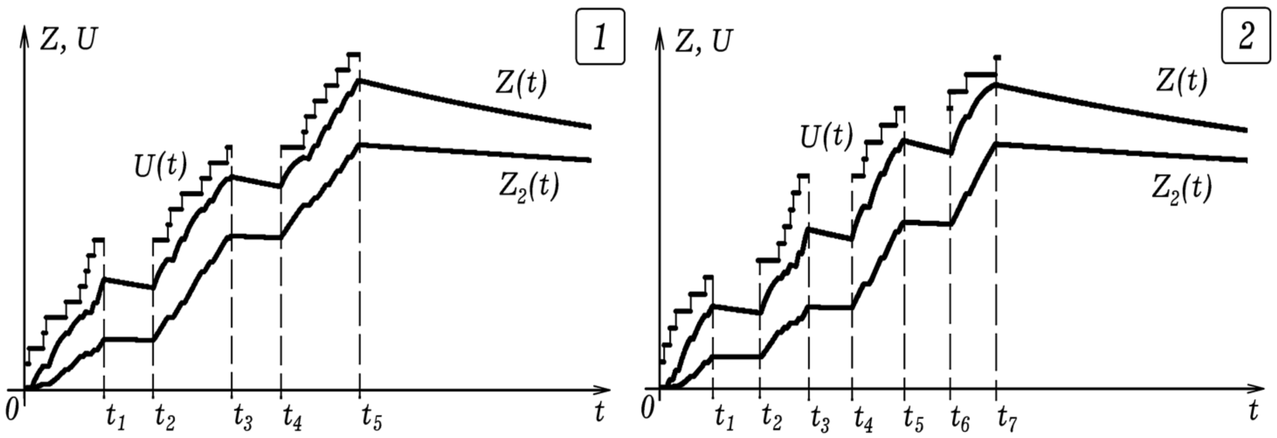


Рис. 3.15. Модель обучения путем решения задач возрастающей сложности

Используется программа ПР–3.14, в ней решение задачи рассматривается как случайный процесс, вероятность которого вычисляется по формуле Роша: $p_i = 1/(1+\exp(-\lambda(Z(t)-\theta_i))$. Результаты имитационного моделирования обучения на трех и четырех занятиях представлены на рис. 3.15. Ступенчатая линия $U(t) = \theta(t)$ показывает, как меняется сложность решаемых задач (уровень предъявляемых требований); графики $Z(t) = Z_1 + Z_2$ и $Z_2(t)$ характеризуют динамику роста «непрочных» и «прочных» знаний. Полученные кривые похожи на графики на рис. 3.13.1, когда требования U в течение урока растут пропорционально времени.

3.8. Обобщенная модель обучения

Рассмотрим многокомпонентную модель обучения, учитывающую различную сложность изучаемых тем и изменение работоспособности ученика в течение учебного дня [30]. Пусть Z – суммарные знания ученика, Z_1 – самые «непрочные» знания первой категории с высоким коэффициентом забывания γ_1, Z_2 – знания второй категории с меньшим коэффициентом забывания γ_2, …, а Z_n – самые «прочные» знания n-й категории с низким γ_n ($\gamma_1 > \gamma_2 > ... > \gamma_n$). Коэффициенты усвоения α_i характеризуют быстроту перехода знаний $(i-1)$-й категории в более прочные знания i-й категории. Пока происходит обучение, $o=1$, а когда оно прекращается, $o=0$. Коэффициент забывания $\gamma=1/\tau$, где τ – время уменьшения знаний в $e=2{,}72...$ раза. Обучение характеризуется не только количеством приобретенных знаний $Z = Z_1 + Z_2 + ... + Z_n$, но и коэффициентом «прочности»:

$$\mathrm{Pr} = (Z_2 / 2^{n-2} + ... + Z_{n-1} / 2 + Z_n) / Z .$$

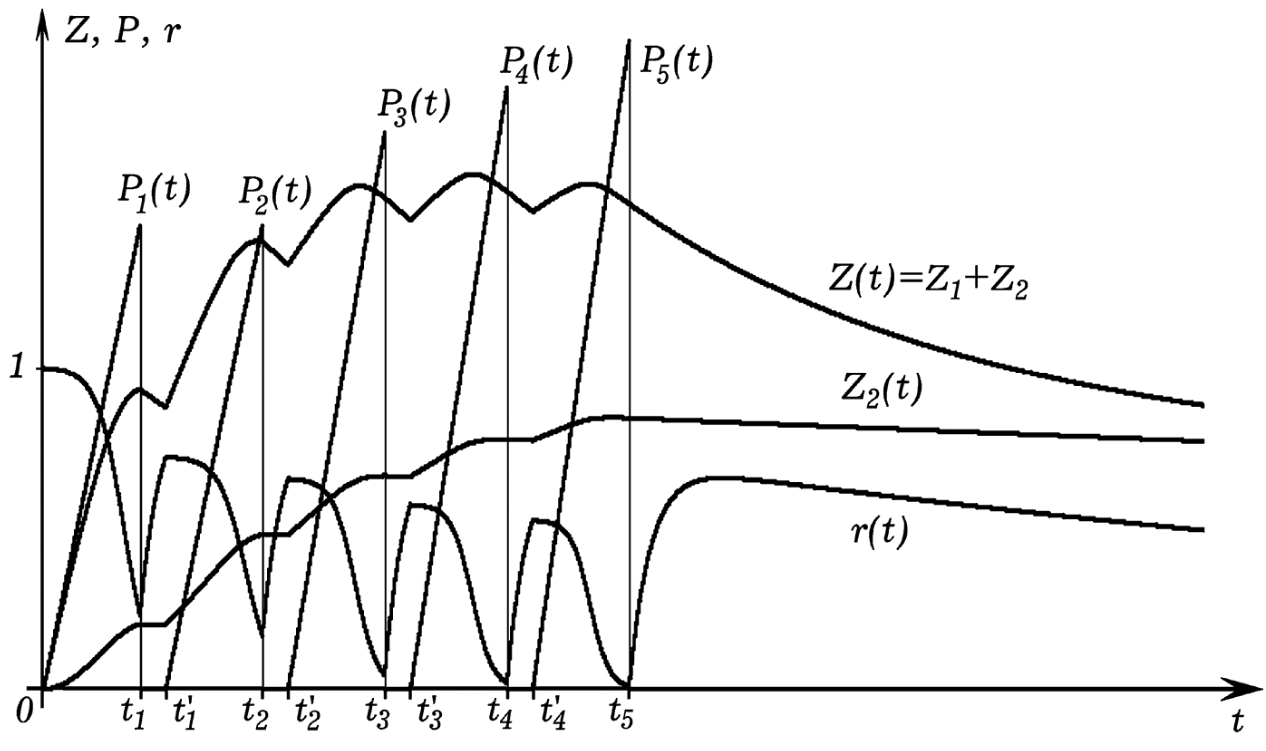


Рис. 3.16. Двухкомпонентная модель, учитывающая изменения работоспособности ученика

При изучении одной темы сначала растет уровень знаний Z, затем происходит увеличение доли прочных знаний Z_n и повышается прочность $\Pr$. Автором предложена **обобщенная модель обучения**, не имеющая аналогов в известной ему литературе. Пусть начальная работоспособность ученика $r_0 = 1$. В любой момент времени $Z(t) = Z_1(t) + ... + Z_n(t)$. Во время обучения ($o = 1$):

$$F = U - Z > 0, \quad r = r_0 / (1 + \exp(k_1(P - P_0))), \quad P = k_2 \int_{t_0}^{t} (1 + S)(U - Z) dt,$$

$$dZ_1 / dt = r(1 - S)(\alpha_1 F Z^b - \alpha_2 Z_1) - \gamma_1 Z_1,$$
$$dZ_2 / dt = r(1 - S)(\alpha_2 Z_1 - \alpha_3 Z_2) - \gamma_2 Z_2,$$
$$\ldots, \; dZ_n / dt = r(1 - S)\alpha_n Z_{n-1} - \gamma_n Z_n.$$

Время перерыва ($o = 0$): $U = 0$, $dr/dt = k_3(r_{\max} - r)$, $r_{\max} = \exp(-k_4 t)$, $dZ_1/dt = -\gamma_1 Z_1$, $dZ_2/dt = -\gamma_2 Z_2$, …, $dZ_n/dt = -\gamma_n Z_n$.

Используется программа ПР–3.15, результаты моделирования приведены на рис. 3.16. «Прочные» знания Z_2 в процессе обучения растут, а затем практически не забываются. «Непрочные» знания $Z_1 = Z - Z_2$ забываются существенно быстрее. Работоспособность ученика во время урока плавно снижается, а во время перерывов повышается до величины, которая постепенно уменьшается в течение дня из-за накапливающейся усталости [30].

Итак, метод имитационного моделирования позволяет проанализировать различные ситуации, встречающиеся в педагогической практике, и изучить влияние тех или иных факторов на результат обучения.

3.9. Приложение к главе 3

```
uses crt, graph;                                           {ПР-3.1}
var DV,MV: integer; U,Z,dt,t,a,g: real;              {Free Pascal}
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
a:=0.03; g:=0.1; dt:=0.01; U:=40; Z:=2; line(0,450,640,450);
Repeat t:=t+dt; If t<10 then U:=40 else U:=Z;
Z:=Z+a*(U-Z)*Z*dt-g*Z*dt;
circle(10+round(t*20),450-round(10*Z),1);
circle(10+round(t*20),450-round(10*U),1);
```

```
until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                        {ПР-3.2}
const  a1=0.05; a=0.03; g=0.0005; dt=0.02; T1=300; Mt=3; Mz=8;
var i,i1,i2,j,DV,MV : integer; M,t,dZ,Z,U,Z1,M1: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,470,640,470); U:=0.2;
Repeat t:=t+dt;
If t<2*T1 then U:=0.0001*sqr(t);
If t>400 then U:=8+0.0001*sqr(t-400);
If t>800 then U:=16+0.0001*sqr(t-800);
If t>1200 then U:=24+0.0001*sqr(t-1200);
If (U-Z<2.5)and(U>Z) then M:=U-Z else M:=0;
If (U-Z1<2.5)and(U>Z1) then M1:=U-Z1 else M1:=0;
Z:=Z+a*M*dt-g*Z*dt; Z1:=Z1+a1*M1*dt-g*Z1*dt;
circle(10+round(t/Mt),470-round(Mz*Z),1);
circle(10+round(t/Mt),470-round(Mz*Z1),1);
circle(10+round(t/Mt),470-round(Mz*U),1);
until KeyPressed; ReadKey; CloseGraph;
END.
```

```
uses crt, graph;                                        {ПР-3.3}
const  a=0.03; g=0.0005; dt=0.005; T1=300; Mt=0.5; Mz=15;
var uch,i,i1,i2,j,DV,MV : integer;
U0, M,t,tau,t0,dZ,Z,U,Z1,M1: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,470,640,470); U0:=0;
Repeat t:=t+dt; U:=U0+0.0002*sqr(t-t0);
If (U-Z<2)and(U>Z-0.1) then
     begin uch:=1; M:=abs(U-Z) end else M:=0;
If U<Z then M:=0.01;
If (M=0)and(uch=1) then begin tau:=100; uch:=0; end;
tau:=tau-dt;
If (tau<0)and(uch=0)then begin U0:=0.9*Z; t0:=t; uch:=1; end;
Z:=Z+a*M*dt-g*Z*dt;
circle(10+round(t*Mt),471-round(Mz*Z),1);
circle(10+round(t*Mt),470-round(Mz*Z),1);
circle(10+round(t*Mt),471-round(Mz*U),1);
circle(10+round(t*Mt),470-round(Mz*U),1);
until (KeyPressed)or(t>5000);
Repeat until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                              {ПР-3.4}
const  a1=0.05; a=0.03; g=0.003; dt=0.01; T1=300;
var i,i1,i2,j,DV,MV : integer; M,t,dZ,Z,U,Z1,M1: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,450,640,450); U:=0.2;
Repeat t:=t+dt; {U:=U+0.005*dt;}
If t<200 then U:=4;
If (t>=200)and(t<400) then U:=8;
If (t>=400)and(t<600) then U:=15;
If (U-Z<5)and(U>Z) then M:=U-Z else {M:=0;}M:=5/exp(U-Z-5);
Z:=Z+a*M*dt-g*Z*dt;
circle(10+round(t),450-round(20*Z),1);
circle(10+round(t),450-round(20*Z1),1);
circle(10+round(t),450-round(20*U),1);
until KeyPressed; ReadKey; CloseGraph;
END.
```

```
uses crt, graph;                                               {ПР-3.5}
const g=0.0015; dt=0.03; T1=90; T2=200; T3=120; T4=210;
t_1=700; t_2=760; N=4;
var i,i1,i2,j,DV,MV: integer; a,t,Sum: real;
M,Z,U,tt: array[0..5] of real;
Procedure Raschet;
Begin
If (t>tt[i-1])and(t<tt[i]) then U[i]:=0.05*(t-tt[i-1])
  else U[i]:=0;
If (U[i]-Z[i]<2)and(U[i]>Z[i]) then M[i]:=U[i]-Z[i]
  else M[i]:=0;
If (t>t_1)and(t<t_2)and(Z[i]<0.05*(tt[i]-tt[i-1]))
                  then M[i]:=0.05*(tt[i]-tt[i-1])-Z[i];
If (t>t_1)and(t<t_2) then a:=0.05;
Z[i]:=Z[i]+a*M[i]*dt-g*Z[i]*dt; Sum:=Sum+Z[i]; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,450,640,450); a:=0.035; tt[1]:=T1;
tt[2]:=tt[1]+T2; tt[3]:=tt[2]+T3; tt[4]:=tt[3]+T4;
Repeat t:=t+dt; Sum:=0; For i:=1 to N do Raschet;
For i:=1 to N do begin
circle(10+round(t/1.5),450-round(10*Sum),1);
circle(10+round(t/1.5),450-round(10*Z[i]),1);
circle(10+round(t/1.5),450-round(10*U[i]),1); end;
until KeyPressed; ReadKey; CloseGraph;
END.
```

```
uses crt, graph;                                        {ПР-3.6}
const  a=0.035; g=0.0015; dt=0.03; T1=600;
t_1=650; t_2=740; t_3=780; t_4=840; N=2; U_ex=12;
var i,i1,i2,j,DV,MV : integer; t: real;
M,Z,U,b,tt: array[0..5] of real;
Procedure Raschet;
Begin If t<T1 then U[i]:=b[i]*t else U[i]:=0;
If (U[i]-Z[i]<3)and(U[i]>Z[i])
  then M[i]:=U[i]-Z[i] else M[i]:=0;
If (t>t_1)and(t<t_2)and(Z[1]<b[1]*T1)
  then M[1]:=b[1]*(T1)-Z[1];
If (t>t_3)and(t<t_4)and(Z[2]<b[2]*T1)
  then M[2]:=b[2]*(T1)-Z[2];
Z[i]:=Z[i]+a*M[i]*dt-g*Z[i]*dt; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,450,640,450); b[1]:=0.02; b[2]:=0.01;
Repeat t:=t+dt; For i:=1 to N do begin Raschet; end;
For i:=1 to N do begin
 circle(10+round(t/1.5),450-round(20*Z[i]),1);
 circle(10+round(t/1.5),450-round(20*U[i]),1); end;
until KeyPressed; ReadKey; CloseGraph;
END.
```

```
uses crt, graph;                                       {ПР-3.7}
const a=0.035; g=0.0005; dt=0.01; T1=300; T2=100;
var i,i1,i2,j,DV,MV: integer; M,Z,U,t: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,450,640,450);  U:=0.2;
Repeat t:=t+dt; {U:=U+0.005*dt;} U:=1.9;
If t>T2 then U:=3.8; if t>2*T2 then U:=5.6;
If t>3*T2 then U:=7.4; if t>4*T2 then U:=9.2;
If t>5*T2 then U:=11; if t>6*T2 then U:=0;
If (U>Z)and(U-Z<2) then M:=U-Z else M:=0;
Z:=Z+a*M*dt-g*Z*dt;
circle(10+round(t/3),450-round(20*Z),1);
circle(10+round(t/3),450-round(20*U),1);
until KeyPressed; Readkey; CloseGraph;
END.
```

```
uses crt, graph;                                         {ПР-3.8}
const  a=0.035; g=0.005; dt=0.03; T1=100;
var i,i1,i2,j,k,DV,MV : integer; t,ZZ,UU: real;
M1,M2,Z1,Z2,U1,U2,tt: array[0..10] of real;
```

```
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
U1[1]:=30;U1[2]:=60;U1[3]:=90;U1[4]:=120;U1[5]:=150;
U2[1]:=70;U2[2]:=70;U2[3]:=70;U2[4]:=70;U2[5]:=70;
line(0,450,640,450); tt[1]:=T1;
Repeat t:=t+dt; i:=1;
k:=round(t) div T1; U1[1]:=k*50+50;
If t>800 then U1[1]:=0;
If U1[i]>Z1[i] then M1[i]:=U1[i]-Z1[i] else M1[i]:=0;
Z1[i]:=Z1[i]+a*M1[i]*dt-g*Z1[i]*dt;
circle(10+round(t/1.8),450-round(1*Z1[i]),1);
circle(10+round(t/1.8),450-round(1*U1[i]),1);
For j:=1 to 10 do U2[j]:=0; U2[k+1]:=50; ZZ:=0;
For i:=1 to 8 do begin
If U2[i]>Z2[i] then M2[i]:=U2[i]-Z2[i] else M2[i]:=0;
Z2[i]:=Z2[i]+a*M2[i]*dt-g*Z2[i]*dt;
circle(10+round(t/1.8),450-round(1*Z2[i]),1);
circle(10+round(t/1.8),450-round(1*U2[i]),1);
ZZ:=ZZ+Z2[i]; end; circle(10+round(t/1.8),450-round(ZZ),1);
until (KeyPressed)or(t>1500); ReadKey; CloseGraph;
END.
```

```
uses crt, graph;                                             {ПР-3.9}
type Uroven = array[1..20] of integer;
const aF=0.025; aL=0.012; gF=0.0005; gL=0.001; dt=0.03; T1=80;
UF: Uroven=(35,0,0,0,0,50,0,0,0,0,90,0,0,0,0,120,0,0,0,0);
UL: Uroven=(0,40,0,0,0,0,60,0,0,0,0,100,0,0,0,0,130,0,0,0);
var i,i1,i2,j,k,DV,MV: integer; t,ZZ,UU,ZF,ZL: real;
MF,ML,U1,U2,tt: array[1..20] of real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,450,640,450);
Repeat t:=t+dt;
i:=round(t/T1)+1; If i>18 then i:=18;
If UF[i]>ZF then MF[i]:=UF[i]-ZF else MF[i]:=0;
ZF:=ZF+aF*MF[i]*dt-gF*ZF*dt;
circle(round(t/3),450-round(2*ZF),1);
circle(round(t/3),450-round(2*UF[i]),1);
if UL[i]>ZL then ML[i]:=UL[i]-ZL else ML[i]:=0;
ZL:=ZL+aL*ML[i]*dt-gL*ZL*dt;
circle(round(t/3),450-round(2*ZL),1);
circle(round(t/3),450-round(2*UL[i]),1);
until (KeyPressed)or(t>60000); CloseGraph;
END.
```

```
uses crt, graph;                                          {ПР-3.10}
var i,j,k,DV,MV: integer;
a,g,U,Zn,dt,t1,t,Zad1,Zad2,Zad3,Zad4,Zad5,Rez: real;
Z,S,time: array[0..6]of real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
a:=0.5; g:=0.01; dt:=0.01; U:=1;
time[1]:=5; time[2]:=4; time[3]:=6; time[4]:=3; time[5]:=4;
S[1]:=0.5; S[2]:=0.1; S[3]:=0.3; S[4]:=0.2; S[5]:=0.4;
line(0,450,640,450);
For i:=1 to 5 do begin t:=0; t1:=t1+time[i-1]; Zn:=Zn+Z[i-1];
Repeat t:=t+dt; Z[i]:=Z[i]+a*(1-S[i])*(U-Z[i])*dt;
For k:=1 to 5 do Z[k]:=Z[k]-g*Z[k]*dt;
circle(10+round((t1+t)*20),450-round(100*(Zn+Z[i])),1);
For k:=1 to 5 do circle(10+round((t1+t)*20),
                   450-round(100*(Z[k])),1);
circle(10+round((t1+t)*20),450-round(100),1);
until (t>time[i])or(KeyPressed); end;
Zad1:=Z[1]; Zad2:=Z[1]*Z[2]; Zad3:=Z[2]*Z[3];
Zad4:=Z[3]*Z[4]; Zad5:=Z[3]*Z[5];
Rez:=(Zad1+Zad2+Zad3+Zad4+Zad5)/5;
circle(10+round(550),450-round(100*Rez),2);
Repeat until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                         {ПР-3.11}
const  a=0.05; g=0.0002; dt=0.01; Tu=300; Tp=25{50}; Mt=5; Mz=4;
var i,i1,i2,j,DV,MV : integer; O,M,t,dZ,Z,U,Z1,M1,S,P,r,r0: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,470,640,470); M:=3; r0:=1;
Repeat t:=t+dt;
If t<Tu then begin O:=1; S:=0.4; end;
If t>Tu then begin O:=0; end;
If t>Tu+Tp then begin O:=1; S:=0.3; end;
If t>2*Tu+Tp then begin O:=0; end;
If t>2*Tu+2*Tp then begin O:=1; S:=0.2; end;
If t>3*Tu+2*Tp then begin O:=0; end;
If t>3*Tu+3*Tp then begin O:=1; S:=0.1; end;
If t>4*Tu+3*Tp then begin O:=0; end;
If t>4*Tu+4*Tp then begin O:=1; S:=0; end;
If t>5*Tu+4*Tp then begin O:=0; end;
If O=1 then begin P:=P+0.2*(1+S)*(M+1)*dt;
r:=r0/(1+exp(0.02*(P-200)));
Z:=Z+O*r*(1-S)*a*M*dt-g*Z*dt; end else
```

```
begin P:=0; r:=r+0.015*(exp(-0.0003*t)-r)*dt; Z:=Z-g*Z*dt; r0:=r;
end;
circle(20+round(t/Mt),470-round(Mz*Z),1);
circle(20+round(t/Mt),471-round(Mz*Z),1);
circle(20+round(t/Mt),470-round(200*r),1);
circle(20+round(t/Mt),471-round(200*r),1);
circle(20+round(t/Mt),470-round(1*P),1);
circle(20+round(t/Mt),471-round(1*P),1);
until KeyPressed; ReadKey; CloseGraph;
END.
```

```
uses crt, graph;                                       {ПР-3.12}
const a1=0.01; a2=0.003; g1=0.005; g2=0.0001;
dt=0.01; Mt=0.2; Mz=2.9;
Var t,U,Z1,Z2,Z3,Z4,Z,Pr,S,k: real; DV,MV: integer;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+dt; U:=0; k:=0;
If (t<300) then begin U:=0.15*t+20; k:=1; end;
If (t>600)and(t<900) then begin U:=0.15*(t-600)+50; k:=1; end;
If (t>1200)and(t<1500) then begin
                 U:=0.15*(t-1200)+90; k:=1; end;
Z:=Z1+Z2; Pr:=Z2/(Z+0.001);
Z1:=Z1+k*a1*(U-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt;
circle(10+round(Mt*t),450-round(Mz*Z),1);
circle(10+round(Mt*t),450-round(Mz*(Z2)),1);
circle(10+round(Mt*t),450-round(Mz*(U)),1);
circle(10+round(Mt*t),450-round(150*Pr),1);
until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                        {ПР-3.13}
const a1=0.01; a2=0.0075; a3=0.005; a4=0.0025; g1=0.02;
g2=0.002; g3=0.0002; g4=0.00002; dt=0.01; Mt=0.25; Mz=5;
Var t,U,Z1,Z2,Z3,Z4,Z,Pr,S,k:real;
DV,MV:integer;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+dt; U:=0; k:=0;
If (t<500) then begin U:=40; k:=1; end;
If (t>800)and(t<1300) then begin U:=60; k:=1; end;
If (t>1600)and(t<2100) then begin U:=80; k:=1; end;
Z:=Z1+Z2+Z3+Z4; Z1:=Z1+k*a1*(U-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt-k*a3*Z2*dt;
```

```
Z3:=Z3+k*a3*Z2*dt-g3*Z3*dt-k*a4*Z3*dt;
Z4:=Z4+k*a4*Z3*dt-g4*Z4*dt; Pr:=(Z2/4+Z3/2+Z4)/(Z+0.001);
S:=S+k*(U-Z)*dt; circle(10+round(Mt*t),450-round(Mz*U),1);
circle(10+round(Mt*t),450-round(Mz*Z),1);
circle(10+round(Mt*t),450-round(Mz*(Z2+Z3+Z4)),1);
circle(10+round(Mt*t),450-round(Mz*(Z3+Z4)),1);
circle(10+round(Mt*t),450-round(Mz*Z4),1);
{circle(10+round(Mt*t),450-round(150*Pr),1);
circle(10+round(Mt*t),450-round(S/100),1);
circle(10+round(Mt*t),450-round(150),1);}
until KeyPressed; CloseGraph;
END.
```

```
uses crt,graph;                                      {ПР-3.14}
const a1=0.005; a2=0.002; g=0.0002; dt=0.2;
b=0.03; Mt=0.1; N=195;
var DV,MV,i,k,r:integer; Z,Z1,Z2,U,t,p,x,Tu,Tp:real;
 Z0: array[1..N]of integer; usl:boolean;
Procedure Test;
begin r:=1; p:=1/(1+exp(-b*(Z-Z0[i]))); x:=random(100)/100;
If (x<p{Z>Z0[i]/2})and(usl) then
begin i:=i+1; circle(round(Mt*t),10,2); r:=0; end
else circle(round(Mt*t),30,2); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');randomize;
For i:=1 to N do Z0[i]:=10+15*i;
i:=1; t:=dt; Z1:=0; Z2:=0; Tu:=600; Tp:=600;
Repeat t:=t+dt; inc(k); {If t<4300 then U:=r*(Z+40) else r:=0;}
usl:=(t<Tu)or((t>Tu+Tp)and(t<2*Tu+Tp))or((t>2*Tu+2*Tp)and
(t<3*Tu+2*Tp))or((t>3*Tu+3*Tp)and(t<4*Tu+3*Tp));
If usl then U:=r*(Z+40) else r:=0; Z:=Z1+Z2;
If U>Z then Z1:=Z1+a1*r*(U-Z)*dt-g*Z1*dt-a2*r*Z1*dt else
  Z1:=Z1-g*Z1*dt;
If Z1>0 then Z2:=Z2+a2*r*Z1*dt-g*Z2*dt/10 else
  Z2:=Z2-g*Z2*dt/10;
If k mod 100=0 then Test; If i>N then i:=N; If k=1000 then k:=0;
circle(round(Mt*t),450-round(Z0[i]),1);
circle(round(Mt*t),450-round(Z1+Z2),2);
circle(round(Mt*t),450-round(Z2),2);
circle(round(Mt*t),450-round(U),1);
until (Keypressed); Closegraph;
END.
```

```
uses crt, graph;                                     {ПР-3.15}
const a1=0.06; a2=0.002; g1=0.001;
```

```
g2=5E-5; dt=0.01; Tu=300; Tp=100; Mt=5; Mz=4; Mr=200;
var o,DV,MV : integer; F,Z,Z1,Z2,S,P,t,r,r0: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,470,640,470); F:=3; r0:=1; Z:=0;
Repeat t:=t+dt; If t<Tu then O:=1; If t>Tu then O:=0;
If t>Tu+Tp then begin O:=1; S:=0; end;
If t>2*Tu+Tp then begin O:=0; end;
If t>2*Tu+2*Tp then begin O:=1; S:=0.2; end;
If t>3*Tu+2*Tp then begin O:=0; end;
If t>3*Tu+3*Tp then begin O:=1; S:=0.3; end;
If t>4*Tu+3*Tp then begin O:=0; end; If t>4*Tu+4*Tp then
begin O:=1; S:=0.4; end; If t>5*Tu+4*Tp then begin O:=0; end;
Z:=Z1+Z2; If O=1 then begin P:=P+0.2*(1+S)*F*dt;
r:=r0/(1+exp(0.03*(P-200)));
Z1:=Z1+r*(1-S)*(a1*F-a2*Z1)*dt-g1*Z1*dt;
Z2:=Z2+a2*r*(1-S)*Z1*dt-g2*Z2*dt; end else begin
P:=0; r:=r+0.015*(exp(-2E-4*t)-r)*dt;
Z1:=Z1-g1*Z1*dt; Z2:=Z2-g2*Z2*dt; r0:=r; end;
circle(10+round(t/Mt),470-round(Mz*Z),1);
circle(10+round(t/Mt),470-round(Mz*Z2),1);
circle(10+round(t/Mt),470-round(Mr*r),1);
circle(10+round(t/Mt),470-round(P*1.2),1);
until KeyPressed; CloseGraph;
END.
```

Глава 4.
ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАК МЕТОД ПЕДАГОГИЧЕСКОГО ИССЛЕДОВАНИЯ

Один из методов исследования сложных систем заключается в создании имитационной модели и изучении ее функционирования при различных условиях. Его применение в дидактических исследованиях позволяет создавать имитационные модели, соответствующие конкретной ситуации, и определять, как зависит результат обучения от параметров системы, начальных условий и внешних воздействий. Этот подход имеет определенные преимущества по сравнению с «методом качественных объяснений»: компьютерные модели логичны, получающиеся результаты обладают статистической устойчивостью, более объективны и жестко связаны наложенными на модель условиями и ограничениями.

4.1. Моделирование сложных ситуаций, связанных с обучением

Рассмотрим несколько ситуаций, возникающих при обучении, построим имитационную модель и проанализируем получающиеся результаты. Это поможет определить круг проблем, которые могут быть решены методом имитационного моделирования.

Ситуация 1. Обучение с помощью ЭВМ, вопросы одного уровня сложности. Учащийся изучил 10 вопросов так, что вероятность правильного ответа на каждый из них 0,1. В момент $t = 0$ осуществляется тестирование на компьютере. Персональная ЭВМ задает 10 разных вопросов примерно одного уровня сложности. Например, учащийся должен перевести 10 слов с иностранного языка. Ответ на каждый вопрос требует 1 условной единицы времени (УЕВ). Компьютер оценивает работу ученика и в зависимости от правильности ответа выдает соответствующую надпись на экране. После окончания тестирования учащийся изучает те вопросы, на которые он дал неправильные ответы, затрачивая время, пропорциональное числу изучаемых вопросов. После окончания обучения снова осуществляется тестирование на компьютере. Компьютер указывает номера неправильных ответов и снова предоставляет время для обучения и т. д. Так продолжается до тех пор, пока доля правильно выполненных заданий не достигнет заданного значения $K = 0{,}8$. Затем следует перерыв в обучении, после которого учащийся снова начинает работать по той же методике: он изучает следующие 10 вопросов, тестируется на ЭВМ, повторяет вопросы, на которые дал неверные ответы, снова тестируется на ЭВМ и т. д.

При создании имитационной модели учтем следующее. В случае правильного ответа на вопрос теста учащийся получает подкрепление. При этом вероятность правильного ответа повышается в соответствии с законом: $p_i' = p_i + \alpha_1(1 - p_i)$. На изучение одного вопроса затрачивается t_1 условных единиц времени (УЕВ), при этом вероятность правильного ответа повышается в соответствии с законом $p_i' = p_i + \alpha_2(1 - p_i)$. Используется программа ПР–4.1. Результаты моделирования роста знаний при обучении ученика в течение трех занятий представлены на рис. 4.1. В течение первого урока (от 0 до t_1) уровень знаний повышается до 8, в течение второго (от t_2 до t_3) – до 16, в течение третьего (от t_4 до t_5) – до 24 условных единиц. Каждая точка на графике $Z(t)$ соответствует результату тестирования. Видно, что в начале каждого занятия время между тестами больше, чем в конце. Это объясняется тем, что когда уровень знаний мал, учащийся допускает больше ошибок в тесте и, следовательно, тратит больше времени на изучение тех вопросов, в которых он ошибся.

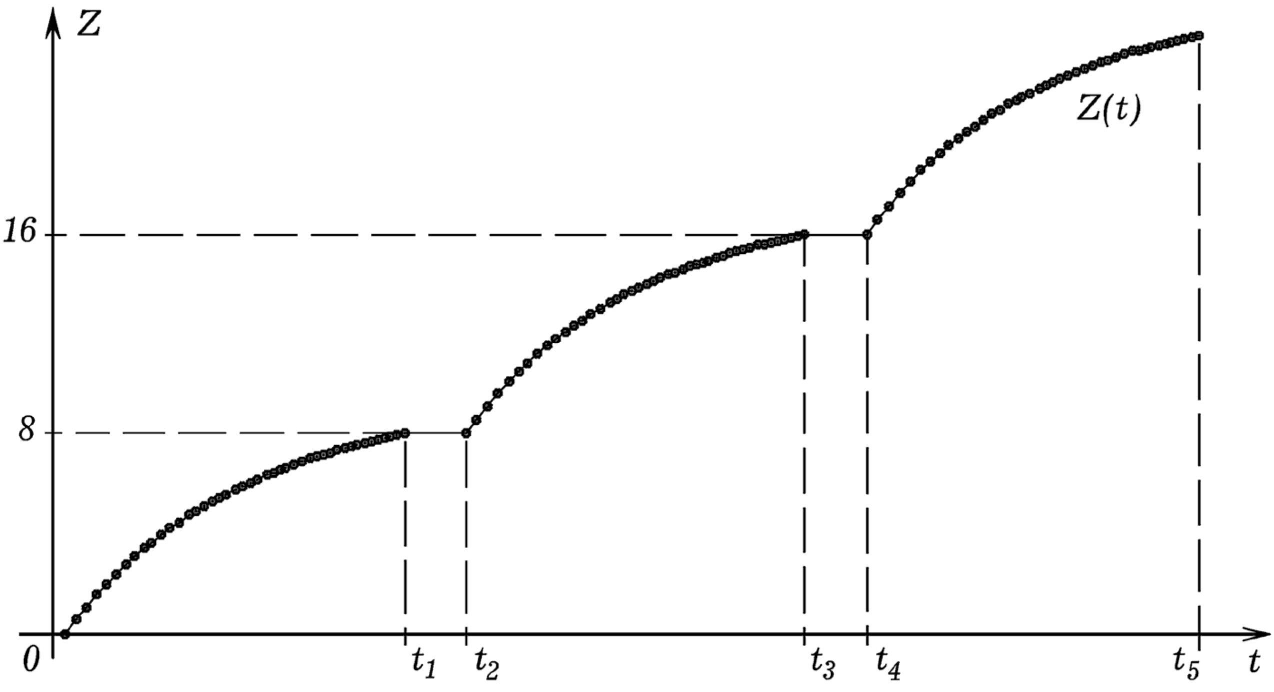


Рис. 4.1. Повышение уровня знаний после трех занятий обучения

Ситуация 2. Изучение вопросов разного уровня сложности, непрерывная модель. Пусть группа учащихся в момент $t=0$ имеет начальный уровень знаний Z_0. Учащиеся изучают первую тему по следующей методике. Сначала они проходят тестирование на компьютере, который предлагает 10 вопросов с возрастающим уровнем сложности $b_1=1$, $b_2=2$, $b_3=3$, …, $b_{10}=10$. Вероятность правильного ответа учащегося на вопрос, имеющий сложность b_i, определяется формулой Раша:

$$p_i = \frac{1}{1+\exp(\alpha(Z-b_i))}.$$

Считается, что группа учащихся справилась с i-ым заданием, если $p_i > 0{,}70-0{,}80$, то есть 70–80 учеников из 100 ответили верно. В противном случае учащиеся не справились с этим заданием теста. Если среднее число правильных ответов по всему тесту меньше $K_{\min}$ = 7 или 8, то считается, что учащиеся не справились с заданиями теста, и они начинают повторно изучать те вопросы, на которые дали неверные ответы. Изучение одного вопроса требует t_1 УЕВ и приводит к повышению уровня знаний на ΔZ, так как скорость усвоения информации остается постоянной. После этого снова группа учащихся тестируется и определяется число правильных ответов. Если оно меньше $K_{\min}$, то учащиеся снова изучают вопросы, с которыми не справились. Так продолжается до тех пор, пока число правильных ответов не достигнет $K_{\min}$ (7–8 из 10).

После занятия З1 начинается перерыв в обучении П1, который длится $T_p=500$ УЕВ. В течение перерыва учащиеся теряют часть знаний вследствие забывания. Уровень знаний уменьшается по экспоненциальному закону с коэффициентом забывания γ. Во время второго занятия З2 учащимся предлагается тест из 10 вопросов, сложность которых уже выше и равна $b_1=10$, $b_2=11$, $b_3=12$, …, $b_{10}=19$. После тестирования учитель проводит занятие, рассматривая только те вопросы, на которые большинству учащихся не удалось правильно ответить. После этого снова проводится тестирование, затем снова обучение, так до тех пор, пока число правильных ответов на вопросы теста не превысит $K_{\min}$. После этого следует второй перерыв П2 в обучении, в течение которого часть информации забывается, и начинается занятие З3. Учащимся предлагается тест из вопросов, сложность которых $b_1=20$, $b_2=21$, $b_3=22$, …, $b_{10}=29$. Обучение осуществляется по той же методике.

Используется программа ПР–4.2 (приложение), результат имитационного моделирования – на рис. 4.2. Каждому тестированию из 10 вопросов соответствует одна точка на графике. Видно, что в течение занятий З1, З2 и З3 средний уровень знаний учеников растет, а во время перерывов П1, П2 и П3 – уменьшается. Во время каждого занятия результаты тестирования увеличиваются от 0 до $K_{\min}$.

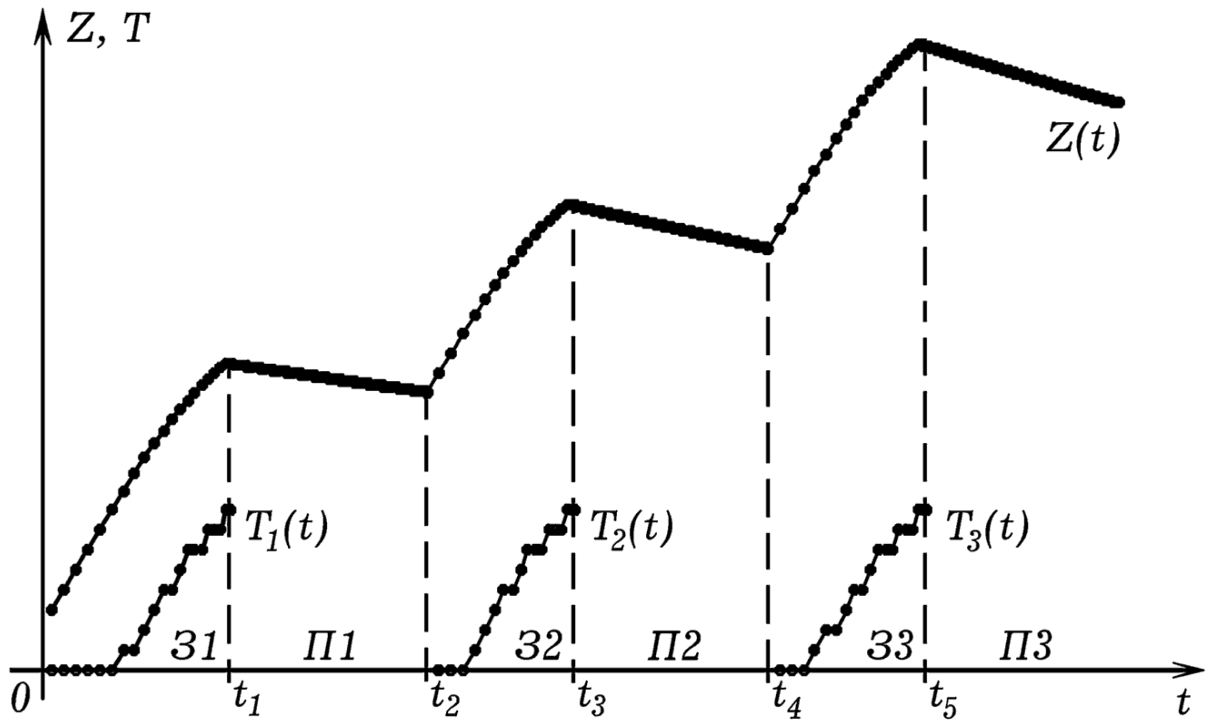


Рис. 4.2. Увеличение количества знаний $Z(t)$
и результаты тестирования $T(t)$

Ситуация 3. Обучение решению задач с учителем, ученик не всегда может оценить правильность решения. Ученик решает задачу, требующую выполнения 10 операций. Вероятность правильного выполнения i -й операции равна p_i. При этом возможны варианты: 1) ученик выполняет операцию правильно (вероятность p_i); 2) ученик выполняет операцию неверно и понимает это (вероятность $q_i = k(1 - p_i)$, $k < 1$); 3) ученик выполняет операцию неверно, но считает, что она выполнена правильно (вероятность $1 - p_i - q_i$). Если операция выполнена верно, или учащийся ошибся, но ему кажется, что он все сделал верно (варианты 1 и 3), то он переходит к следующей операции, продолжая решать задачу. Если хотя бы одна операция выполнена неверно, то задача решена неправильно. В случае, когда учащийся осознает, что допустил ошибку, он обращается к учебнику и в течение времени t_1 изучает материал.

После этого он повторяет свою попытку выполнить эту операцию. Так продолжается до тех пор, пока он не выполнит все 10 операций.

Если задача решена неверно, то учитель объясняет, какие операции выполнены неправильно. Вероятность их правильного выполнения увеличивается по закону: $p_i' = p_i + \alpha(1 - p_i)$. На объяснение того, как следует правильно выполнять каждую операцию, затрачивается время t_2 УЕВ. После этого учитель дает новую задачу того же типа, требующую выполнения той же последовательности операций. Уровень знаний учащегося находится, как сумма всех вероятностей: $Z = p_1 + p_2 + p_3 + ... + p_{10}$.

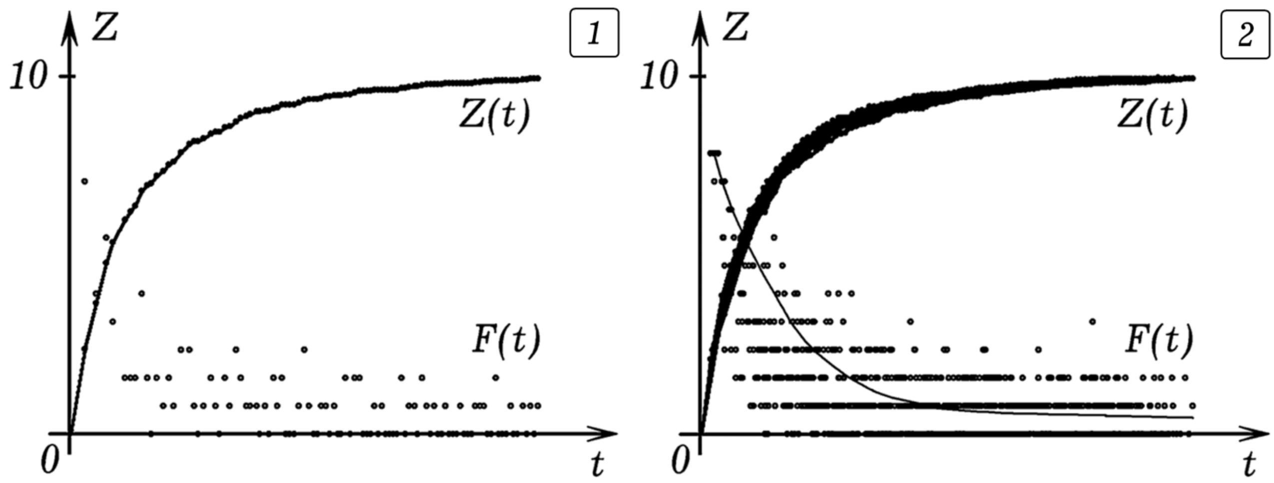


Рис. 4.3. Знания ученика Z и усилия учителя F как функция времени t

Для имитационного моделирования используется программа ПР–4.3. В случае с одним учащимся получается график, изображенный на рис. 4.3.1. Видно, что уровень знаний растет по закону $Z(t) = 10\,(1 - \exp(-b_1 t))$, а усилия учителя уменьшаются, так как ученик ошибается реже. Результаты моделирования для группы из 10 учащихся представлены на рис. 4.3.2. В данном случае усилия учителя описываются дискретной функцией $F(t)$, среднее значение которой можно аппроксимировать так: $F = F_0 \exp(-b_2 t)$.

4.2. Проблема оптимизации процесса обучения: дискретная модель

Особый интерес представляет **проблема нахождения оптимального пути изучения той или иной совокупности вопросов различной сложности** [23]. Допустим, учащийся изучает $N = 3$ темы, и по каждой из тем учитель может решить различное число задач. Время обучения ограничено, общее количество задач, которое решает учащийся, $M = 200$. Будем использовать автоматный подход к моделированию обучения, заменяя учащегося вероятностным автоматом, который с вероятностью p решает задачу правильно, а с вероятностью $(1-p)$ – нет. На решение одной задачи затрачивается время Δt, при этом за счет размышлений и изучения учебной литературы вероятность решения учащимся задачи данного типа повышается на $\Delta p = \alpha(1-p)$, где α – коэффициент научения. Уровень знаний Z_i i-й темы равен вероятности p_i решения задачи по i-й теме. Сложность задач i-й зависит от уровня овладения предыдущих тем. Будем считать, что коэффициент научения α_i, соответствующий i-й теме, с ростом уровня знаний Z_j ($j < i$) увеличивается (например, $\alpha_3 = aZ_1 + 0{,}1$). В конце обучения проводится тестирование, его результат равен $T = V_1 Z_1 + V_2 Z_2 + V_3 Z_3$, где V_1, V_2, V_3 – коэффициенты важности, характеризующие степень значимости изученных тем. Необходимо найти оптимальную последовательность решения задач (функцию $U(t)$), при которой результат тестирования T в конце обучения максимален. Будем использовать метод имитационного моделирования. В программе ПР–4.4 случайно перебираются различные пути изучения этих тем, каждый раз выбирается тот, при котором T больше. Проанализируем следующие ситуации:

Ситуация 1. Коэффициенты усвоения второй и третьей тем зависят от уровней знаний Z_1 и Z_2 следующим образом: $\alpha_2 = 0{,}5\alpha_1 Z_1$, $\alpha_3 = \alpha_1 Z_2$. Все темы имеют одинаковую важность: $V_1 = V_2 = V_3 = 1$. Результаты оптимизации при $\alpha_1 = 0{,}025$ представлены на рис. 4.4.1, а при $\alpha_1 = 0{,}05$ – рис. 4.4.2. В интервале $[0; t_1]$ следует решать задачи из темы 1, в интервале $[t_1; t_2]$ – задачи из темы 2, а в интервале $[t_2; t_3]$ – задачи из темы 3. Во втором случае коэффициенты усвоения в два раза больше, поэтому и результаты тестирования после выполнения того же числа задач выше ($T_1 = 1{,}91$, $T_2 = 2{,}74$, $T_{\max} = 3$).

Ситуация 2. Коэффициенты усвоения тем: $\alpha_1 = 0{,}025$, $\alpha_2 = \alpha_1 Z_1$, $\alpha_3 = \alpha_1(Z_1 + Z_2)$. При тестировании проверяются знания только третьей темы: $V_1 = V_2 = 0$, $V_3 = 1$. Результаты оптимизации $U(t)$ представлены на рис. 4.4.3. Оценка за тест $T_1 = 0{,}97$ при $T_{\max} = 1$.

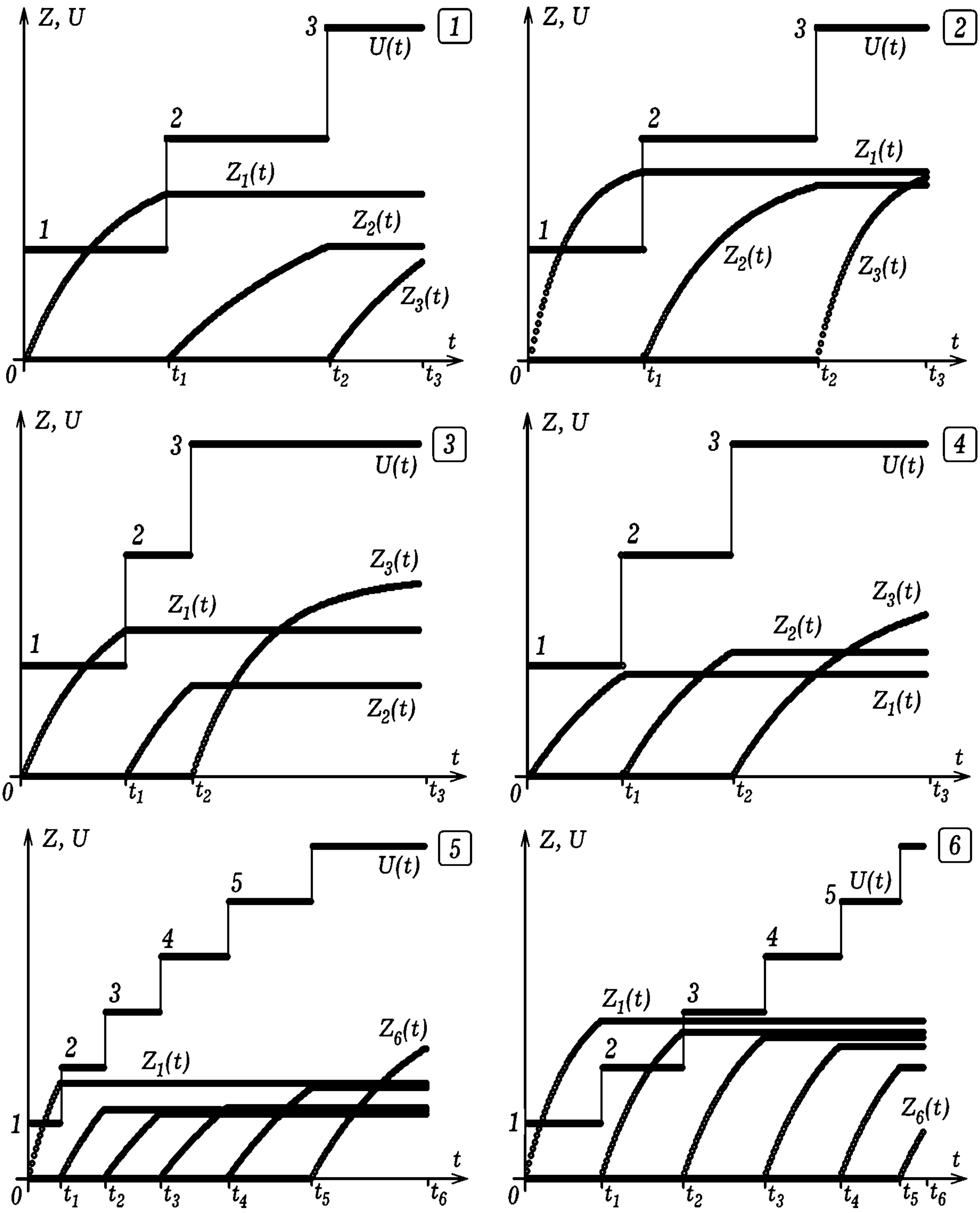


Рис. 4.4. Результаты оптимизации процесса обучения

Ситуация 3. Коэффициенты усвоения трех тем заданы так: $\alpha_1 = 0{,}015$, $\alpha_2 = 0{,}015 Z_1 + 0{,}01$, $\alpha_3 = 0{,}015(Z_1 + Z_2)$, а их коэффициенты важности: $V_1 = 0$, $V_2 = 1$, $V_3 = 5$. Результаты оптимизации представлены на рис. 4.4.4. Оценка за тест $T = 4{,}69$, $T_{\max} = 6$.

Ситуация 4. Учащийся изучает 6 тем. Коэффициент усвоения i -й темы пропорционален уровню знаний предыдущей $(i-1)$-й темы: $\alpha_1 = 0{,}04$, $\alpha_2 = 0{,}04 Z_1$, $\alpha_i = 0{,}04 Z_{i-1}$. Тест проверяет только знания 6-й темы; коэффициенты важности равны: $V_1 = V_2 = ... = V_5 = 0$, $V_6 = 1$. Используется программа ПР–4.5, результаты оптимизации – на рис. 4.4.5. Оценка за тест $T = 0{,}67$, $T_{\max} = 1$.

Ситуация 5. Учащийся изучает 6 тем. Коэффициенты усвоения i -й темы пропорциональны уровню знаний предыдущей $(i-1)$-й темы и задаются так: $\alpha_1 = 0{,}04$, $\alpha_2 = 0{,}04 Z_1$, $\alpha_i = 0{,}04 Z_{i-1}$. Тест проверяет знания всех тем; коэффициенты важности равны: $V_1 = V_2 = ... = V_6 = 1$. Результаты оптимизации представлены на рис. 4.4.6. Оценка за тест $T = 3{,}67$ при $T_{\max} = 6$.

Из результатов моделирования следуют выводы [23]: 1. Если изучение i -й темы приводит к увеличению коэффициента усвоения j-й темы, то сначала следует изучать i -ю тему, а затем j-ю. 2. Если в выходном тесте в большей степени представлены задачи i -й темы (то есть они имеют большую важность), то большую часть учебного времени следует решать задачи i -й темы. Все это приведет к повышению эффективности обучения. Функция $U(t)$ соответствует уровню требований учителя.

4.3. Оптимизация обучения: непрерывная модель

Обучение будет наиболее эффективным, когда уровень требований учителя превышает знания учащегося на максимально возможную величину, при которой у учащегося не пропадает мотивация к учебной деятельности (усилия F максимальны). Выше такой режим обучения был назван **согласованным**. Для нахождения эффективного пути обучения, соответствующего минимальным затратам энергии учителя и учащегося, в качестве целевой функции рассматриваемой оптимизационной задачи возьмем функционал:

$$P = \int_1^2 k(U - Z)dt \approx \sum_{j=1}^{n} k(U_j - Z_j)\Delta t .$$

Разность $U - Z$ характеризует интенсивность умственной деятельности (прилагаемые усилия F), а величина P пропорциональна работе, совершенной учеником. Нагрузка должна быть равномерно распределена по всем занятиям и не должна превышать критическое значение $P_{\max}$, чтобы не было переутомления. Поэтому для каждого урока продолжительностью T_u нужно вычислять затраты энергии (или совершенную учеником работу) $P_i = k(U - Z)\Delta t$ и сравнивать их с пороговым значением $P_{\max}$.

Ситуация 2. В режиме согласованного обучения проводятся три занятия, начинающиеся в фиксированные моменты времени 0, t_2, t_4. Необходимо определить длительность занятий $T_u = t_1 = t_3 - t_2 ...$, при которой уровень знаний после обучения в момент t' будет равен Z' или выше.

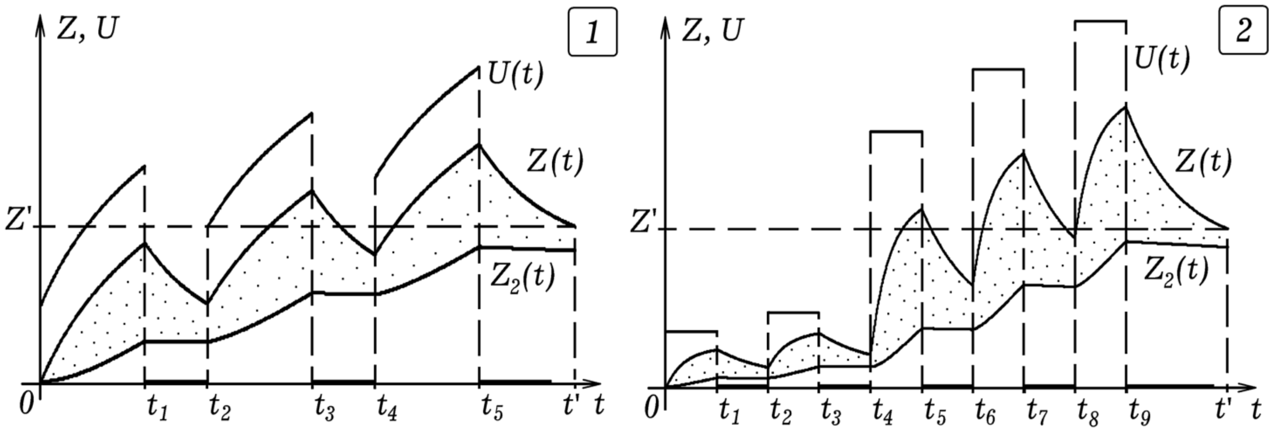


Рис. 4.5. Нахождение оптимальной организации обучения

Для решения этой оптимизационной задачи используется программа ПР–4.6, содержащая цикл, в котором случайным образом изменяется длительность урока T_u, затем пересчитывается количество знаний у ученика Z и выясняется, приблизилось Z к требуемому значению Z' или нет. Если да, то изменения T_u принимаются, если нет, то отвергаются, и все повторяется снова. Результаты решения этой задачи представлены на рис. 4.5.1.

Ситуация 3. Проводится пять занятий, начинающихся в моменты 0, t_2, t_4, t_6, t_8, которые имеют фиксированную длительность $T_u = t_1 = t_3 - t_2 ...$ Уровни требований, предъявляемые учителем, U_1, U_2, U_3, U_4, U_5 могут изменяться. Необходимо подобрать такие U_i, чтобы уровень знаний учащегося в момент t' достиг заданного значения Z'.

Используется программа ПР–4.7. На каждом шаге программа подбирает U_i $(i = 1,2,..,5)$ так, чтобы совершаемая учеником работа в течение урока не превышала порогового значения, а уровень знаний в момент t' был не ниже

Z'. Если эти условия выполняются, то новое значение U_i принимается, а если нет, то отвергается. Результаты моделирования представлены на рис. 4.5.2.

4.4. Изучение вопросов, связанных генетической связью: вероятностный и детерминированный подходы

Проблема итеративного обучения, заключающегося в формировании знаний и навыков в результате многократного повторения определенной последовательности действий, достаточно подробно исследована, например, в [3; 17; 32]. Особый интерес представляет собой процесс обучения, при котором изучаемые элементы учебного материала (ЭУМ) связаны генетической связью. Например, запоминание однокоренных или похожих слов иностранного языка, овладение теорией, являющейся развитием ранее рассмотренных теорий, вывод следствий из теорем и т. д. Будем считать, что два объекта ЭУМ 1 и ЭУМ 2 связаны генетической связью, если один объект получается из второго путем добавления или удаления элементов. Допустим, ученик сначала изучил теорию A, затем, расширив свои знания, освоил теорию B, которая включает в себя теорию A как частный случай, после этого изучил теорию C, включающую в себя B, и т. д. Например, ученик научился считать (ЭУМ 1), складывать целые числа (ЭУМ 2), складывать дроби (ЭУМ 3) и т. д. Очевидно, что этот случай нельзя свести к простому итеративному обучению за счет многократного повторения.

При обучении в мозгу человека возникают новые психические образования, появляются нервные центры, соединенные между собой связями. Некоторые связи становятся сильнее, другие, наоборот, исчезают. **Дискретной моделью ученика** (ДМУ) может служить созданный на базе ЭВМ вероятностный автомат переменной структуры, у которого в процессе обучения увеличивается число внутренних состояний, соответствующих изучаемым понятиям, и изменяются вероятности переходов (то есть устанавливаются нужные и исчезают ненужные связи). Пусть ДМУ изучает элемент учебного материала (ЭУМ) A, соответствующий слову «a_», затем ЭУМ AB (слово «ab_»), являющийся развитием ЭУМ A, после этого – ЭУМ ABC (слово «abc_»), который получается их предыдущего ЭУМ AB, и т. д. Сначала ДМУ не обучена и имеет большое число ($N = 1000$) внутренних состояний, связанных между собой связями с вероятностями перехода $p_{ij} = 1/N$. Последнее состояние будем считать конечным $N_{stop} = N$, в него автомат попадает из любого другого состояния, считав символ «_», означающий конец слова. Необученный автомат при этом должен выдать ответ: «не понял», а обученный – значение входного слова.

Обучение должно происходить так. Сначала ДМУ находится в состоянии $q = 0$. На его вход поступает слово «ab_». Автомат, считав букву «а», совершает переход $0 \rightarrow 1$ в какое-то состояние, которому он автоматически присваивает номер $q = 1$ (рис. 4.6). При этом вероятность p_{01} перехода $0 \rightarrow 1$ увеличивается на $\Delta p = \alpha(1 - p_{01})$. Затем он считывает букву «b» и автоматически совершает переход в состояние, которому он присваивает номер $q = 2$, вероятность p_{12} перехода $1 \rightarrow 2$ увеличивается на $\Delta p = \alpha(1 - p_{12})$. Когда ДМУ прочитывает последнюю букву «_», он переходит в последнее состояние $N_{stop} = N$, вероятность этого перехода $2 \rightarrow N$ увеличивается на $\Delta p = \alpha(1 - p_{2N})$. Переменной zn присваивается значение слова «ab_», сообщаемое учителем (например, «reakciya2») [25].

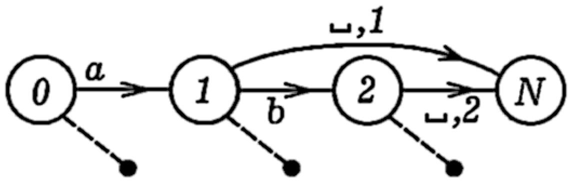


Рис. 4.6. Обучение дискретной модели ученика

Пусть ДМУ обучается слову «a_», которому соответствует значение «reakciya1». При этом автомат совершает переходы $0 \rightarrow 1$ и $1 \rightarrow N$ (рис. 4.6), их вероятности увеличиваются. Если на вход автомата поступает слово «abc_», а он уже усвоил слово «ab_», то из состояния $q = 2$ автомат, получив на вход «с», не переходит в состояние N_{stop}. Он создает другой переход в новое состояние $q = 3$ (на рис. 4.6 не изображено). При поступлении символа «_» автомат из состояния 3 переходит в состояние N_{stop}, запоминая значение слова «abc_» (zn:= «reakciya3»). Аналогично происходит обучение словам acb_ и adea_, имеющим другие значения. На всех этапах обучения «учитель» контролирует работу «ученика», подкрепляя его действия: если ДМУ совершил правильный переход, то соответствующая вероятность увеличивается, а если нет – уменьшается. После многократного обучения ДМУ приобретет способность распознавать слова, поступающие на вход.

Теперь несколько упростим задачу. Пусть автомат ДМУ не самостоятельно создает новые состояния и связи между ними, а прислушивается к «учителю», сообщающему, в какое состояние и по какому переходу следует перейти после поступления на вход нового символа. Вместо многократного повторения одних и тех же операций, «поощрений» и «наказаний» с последующим пересчетом вероятностей переходов добавим в текст программы условный оператор, обеспечивающий переход автомата из одного состояния в другое. Это соответствует ситуации, когда учитель просто сообщает учащемуся последовательность действий или алгоритм, ведущий к решению задач данного типа. Программа ПР–4.8 моделирует работу детерминированного автомата с заданным числом внутренних состояний. В нее можно добавлять условные операторы и таким образом создавать новые связи между различными состояниями ДМУ.

Ситуация 1. ДМУ изучает объекты (формирует навыки) в следующей последовательности: 1) объект «abcd_»; 2) объект «a_»; 3) объект «ab_»; 4) объект «abc_». После обучения на каждый объект ДМУ реагирует соответствующим образом.

В данном случае учащийся сначала изучает всю теорию ABCD (или осваивает полную последовательность действий), а затем разбирает частные случаи: теорию A, сложную теорию AB и еще более сложную теорию ABC. Используется программа ПР–4.8. Для изучения объекта «abcd» необходимо создать пять новых связей с помощью условных операторов:

```
If (q=0)and(y='a')then begin q:=1; znn:='OK'; goto m1; end;
If (q=1)and(y='b')then begin q:=2; znn:='OK'; goto m1; end;
If (q=2)and(y='c')then begin q:=3; znn:='OK'; goto m1; end;
If (q=3)and(y='d')then begin q:=4; znn:='OK'; goto m1; end;
If (q=4)and(y='_')then begin q:=9;znn:=zn[4]; goto m1; end;
```

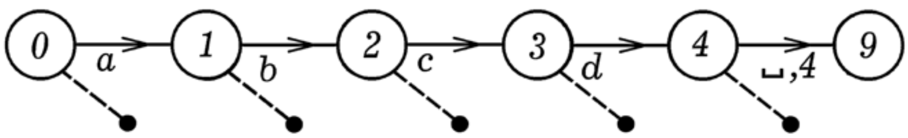


Рис. 4.7. Граф обучения, соответствующий ситуации 2

Чтобы теперь обучить автомат словам «a_», «ab_», «abc_», достаточно к существующим добавить три новые связи с помощью операторов:

```
If (q=1)and(y='_') then begin q:=9; znn:=zn[1]; goto m1; end;
If (q=2)and(y='_') then begin q:=9; znn:=zn[2]; goto m1; end;
```

```
If (q=3)and(y='_') then begin q:=9; znn:=zn[3]; goto m1; end;
```

Если теперь после запуска программы вводить слова «abcd_», «a_», «ab_», «abc_», то созданная ДМУ будет реагировать правильно (рис. 4.8). Все это моделирует изучение запоминания генетически связанных объектов [25].

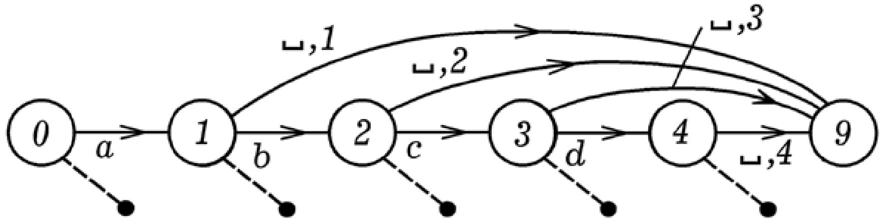


Рис. 4.8. Граф обучения, соответствующий ситуациям 1 и 2

Ситуация 2. Пусть ДМУ изучает объекты (формирует навыки) в следующей последовательности: 1) объект «a_»; 2) объект «ab_»; 3) объект «abc_»; 4) объект «abcd_»; 5) объект «cdbc_». Промоделируем на компьютере функционирование детерминированного автомата с фиксированным числом состояний, который можно будет научить различать эти объекты путем добавления условных операторов (новых связей).

Работа этого автомата поясняется рис. 4.9. Сначала автомат находится в состоянии $q = 0$. Чтобы «научить» его правильно реагировать на слово «a_», необходимо организовать переходы (0, «a») $\to$ (1, «OK») и (1, «_»)$\to$ $\to$ (9, «reakciya1»), в результате чего на экране монитора должна появиться надпись «reakciya1», которую можно считать значением слова «a_». Для этого используются следующие два оператора условного перехода:

```
If (q=0)and(y='a') then begin q:=1; znn:='OK';  goto m1; end;
If (q=1)and(y='_') then begin q:=9; znn:=zn[1]; goto m1; end;
```

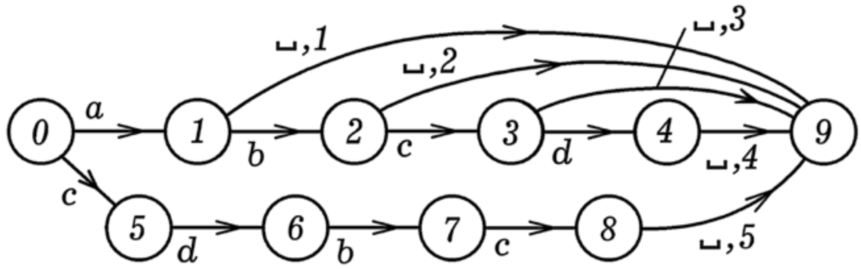


Рис. 4.9. Граф обучения новому слову «cdbc_»

Чтобы автомат, «изучивший» объект «a_», «научить» правильно реагировать на объект «ab_», также достаточно создать два перехода (1, «b») $\to$ $\to$ (2, «OK») и (2, «_»)$\to$ (9, «reakciya2»). В результате последнего перехода на экране должно появиться слово «reakciya2». Для этого нужно еще два оператора:

```
If (q=1)and(y='b') then begin q:=2; znn:='OK'; goto m1; end;
If (q=2)and(y='_') then begin q:=9; znn:=zn[2]; goto m1; end;
```

Аналогичным образом осуществляется «обучение» автомата словам «abc_» и «abcd_»: каждый раз достаточно добавлять по два оператора условного перехода, соответствующих созданию двух новых связей. Чтобы проверить распознавание компьютером того или иного объекта, необходимо присвоить переменной m номер предъявляемого слова. На экран будет выведена последовательность состояний, которую проходит автомат, и его реакция.

С объектом «cdbc_» ситуация принципиально иная: он не может быть получен из ранее изученных объектов путем добавления одного или двух элементов. Слово «cdbc_» не ассоциируется ни с одним из ранее изученных слов; чтобы «научить» автомат правильно на него реагировать, необходимо создать пять новых связей. В компьютерную программу, моделирующую такой автомат, следует добавить пять операторов:

```
If (q=0)and(y='c') then begin q:=5; znn:='OK'; goto m1; end;
If (q=5)and(y='d') then begin q:=6; znn:='OK'; goto m1; end;
If (q=6)and(y='b') then begin q:=7; znn:='OK'; goto m1; end;
If (q=7)and(y='c') then begin q:=8; znn:='OK'; goto m1; end;
If (q=8)and(y='_') then begin q:=9;znn:=zn[5]; goto m1; end;
```

Из **принципа экономии** следует, что **в сложных системах с большей вероятностью реализуются состояния и процессы, требующие минимальных затрат энергии** (ресурсов, сил, времени). В данном случае такой системой является мозг человека. Запоминание новых знаний (формирование умений и навыков), основанное на создании ассоциаций с уже имеющимися знаниями, требует создания меньшего числа связей, значит, меньших затрат энергии. Поэтому оно осуществляется легче и быстрее, чем при механическом заучивании.

4.5. Изучение вопросов, связанных генетической связью: результаты имитационного моделирования

Проанализируем проблему изучения генетически связанных ЭУМ с помощью непрерывной модели ученика. Пусть учитель в течение первого урока решает n_1 учебных задач типа 1 на знание теории А, затем после перемены – n_2 задач типа 2 на знание теории В. После второй перемены – n_3 задач типа 3 на знание теории С. После третьей перемены – n_4 задач типа 1–2, требующих знаний теории А и В, затем после перемены – n_5 задач типа 1–2–3, требующих знаний теорий А, В и С, и т. д. в соответствии с программой обучения ПО–1. А в следующий раз учитель работает по программе ПО–2 и на всех 5 уроках сразу решает задачи типа 1–2–3, одновременно формируя знание тем А, В и С.

Пусть при решении каждой задачи типа k вероятность правильного решения $Z_k = p_k$ увеличивается на $\alpha(1-p_k)$, где α – коэффициент научения. Чем выше уровень знаний $Z_k = p_k$, тем меньше времени t_k затрачивается на решение следующей задачи типа k. Будем считать, что: $t_k = t_0 - \ln(p_k)/\beta$, где t_0 – минимальное время, затрачиваемое учеником, имеющим максимальный уровень знаний $Z_k = p_k = 1$, на решение задачи типа k.

Урок заканчивается, ученик несколько дней не занимается изучением данного предмета, теряя при этом приобретенные знания вследствие забывания: $p_i^{t+1} = p_i^t(1-\gamma\Delta t)$, $i = 1,2,3$. На втором уроке учитель расширяет теоретическую модель от А до АВ и решает n_2 задач типа 1–2, в которых используются знания теорий А и В. При каждом решении задачи повышаются уровни знаний $Z_1 = p_1$ и $Z_2 = p_2$, затрачивается время $t_1 = t_0 - \ln(p_1)/\beta$ и $t_2 = t_0 - \ln(p_2)/\beta$. Если у учащегося сформированы знания теории С на уровне $Z_3 = p_3$, которая не используется при решении задач типа 1–2, то эти знания уменьшаются вследствие забывания по закону: $p_3^{t+1} = p_3^t(1-\gamma\cdot\Delta t)$.

Эта ситуация моделируется с помощью программы ПР–4.9. В ней перечислены все типы решаемых задач: zadacha[1]:='a'; zadacha[2]:='b'; zadacha[3]:='ab'; zadacha[4]:='c'; zadacha[5]:='bc'; zadacha[6]:='abc'; zadacha [7]:='_';. Последняя zadacha[7] соответствует перерыву между уроками. Программа обучения ПО1={a, 6; p, 18; ab, 8; p, 18; abc, 6; p, 18; abc, 6; p, 18} задана в виде: z:=1; chislo:=6; Raschet; z:=7; chislo:=18; Raschet; z:=3; chislo:=8; Raschet; z:=7; chislo:=18; Raschet и т. д. Это означает, что задачи типа 1 ('a') будут решены 6 раз, после чего наступит перерыв длительностью 18 условных единиц времени. Затем решаются задачи типа 1–2 ('ab’) 8 раз, снова перерыв, после этого задачи типа 1–2–3 («abc») 6 раз и т. д. Процедура Raschet содержит цикл по времени, в котором пересчитываются уровни знаний $Z_i = p_i$, $i = 1,2,3,...$ в процессе решения каждой задачи. При этом учитывается уменьшение времени решения задачи с ростом знаний учащегося, а также уменьшение неиспользуемых знаний вследствие забывания [25].

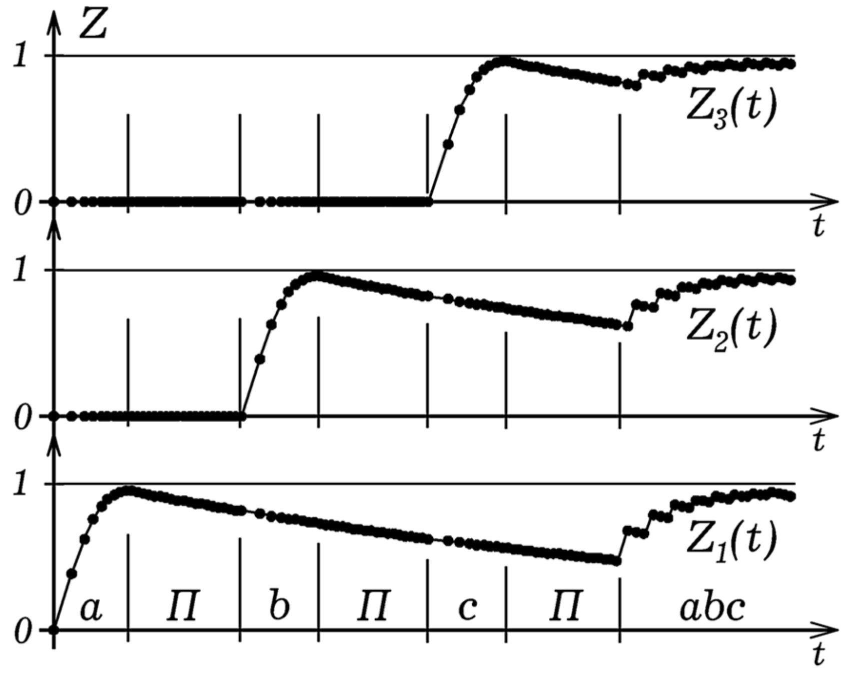


Рис. 4.10. Моделирование изучения генетически связанных теорий

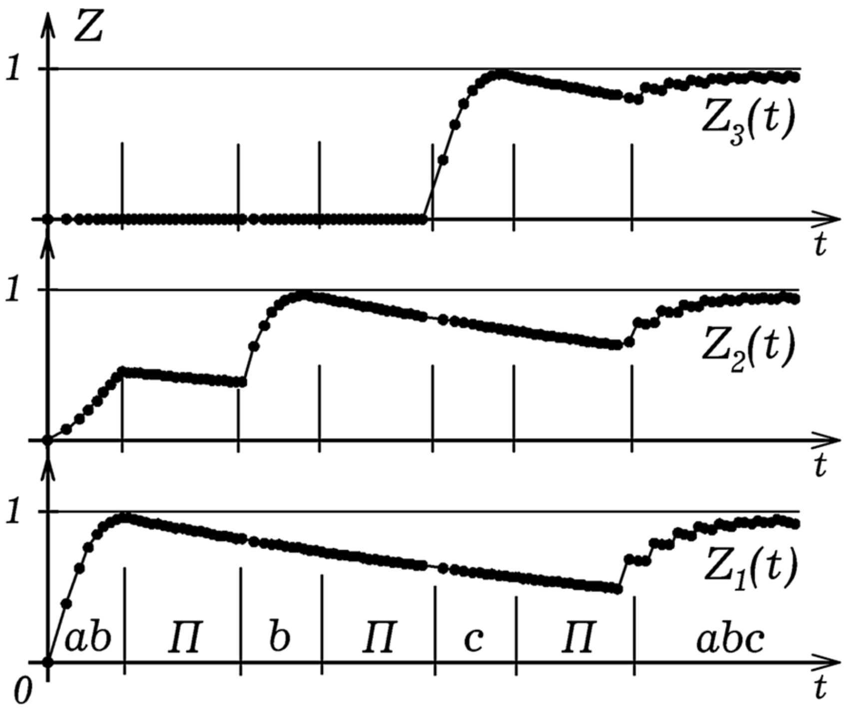


Рис. 4.11. Результаты моделирования изучения связанных теорий

Ситуация 1. Учитель работает по программе: ПО1 = {a, 6; p, 18; b, 8; p, 18; c, 6; p, 18; abc, 6; p, 18}, то есть на первом уроке решает задачи типа A, на втором – типа B, на третьем – типа C, а на четвертом – типа ABC. Для моделирования этой ситуации в программу ПР–4.9 необходимо вставить операторы:

```
z:=1; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=2; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=4; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=6; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
```

Результаты вычислений представлены на рис. 4.10. Видно, что на первом уроке растет Z_1, а затем снижается; на втором уроке происходит увеличение Z_2, а затем снижается. На третьем уроке растет Z_3. Во время четвертого урока, на котором решаются задачи типа ABC, происходит увеличение Z_1, Z_2 и Z_3.

Ситуация 2. Учитель работает по программе: ПО2 = {ab, 6; p, 18; b, 8; p, 18; c, 6; p, 18; abc, 6; p, 18}, то есть на первом уроке решает задачи типа AB, на втором – типа B, на третьем – типа C, а на четвертом – типа ABC. Результаты моделирования приведены на рис. 4.11.

4.6. Изучение вопросов, связанных генетической связью: многокомпонентная модель обучения

Рассмотренная выше модель слишком примитивна и не учитывает ряд важных факторов. Если бы процесс обучения был подобен наполнению резервуара жидкостью из различных сосудов и описывался бы линейным уравнением, то было бы все равно, в какой последовательности осуществляется изучение теорий A, B и C. На самом же деле это не так. Психологи установили следующее:

1. При обучении человек не может удержать в оперативной памяти слишком большое количество новой информации (обычно 5–9 блоков). 2. При многократном повторении одних и тех же действий (в том числе интеллектуальных), при решении подобных задач прочность знаний растет, и после обучения учащийся их практически не забывает. Поэтому не надо обучать всем буквам алфавита за один урок. Необходимо алфавит разбить на порции по 3–4 буквы и на каждом уроке давать новую порцию информации. В то же время если человек научился читать или считать, а затем выполнил достаточно

большое количество упражнений, то знания становятся настолько прочными, что и после большого перерыва он сможет вспомнить изученный материал. 3. Обучение приводит к повышению интеллектуального уровня учащихся, то есть увеличению коэффициента научения. Чем больше учащийся знает, тем легче он усваивает новую информацию, устанавливая ассоциативные связи с уже имеющимися знаниями. 4. Если изучаемый материал слишком сложен и/или длительность занятия достаточно велика, то ученик начинает испытывать усталость, что приводит к снижению коэффициента научения.

Будем использовать многокомпонентную модель обучения [27; 28], считая, что при многократном повторении часть информации хорошо усваивается, на ее основе формируются навыки (или более прочные знания) с низким коэффициентом забывания. Математически это можно выразить так [25]:

$$z_i^{t+1} = z_i^t + a_1(1 - p_i^t) - a_2 z_i^t - \gamma_1 z_i^t, \quad n_i^{t+1} = n_i^t + a_2 z_i^t - \gamma_2 n_i^t,$$

$$p_i^{t+1} = z_i^{t+1} + n_i^{t+1}.$$

Здесь p_i^t – вероятность правильного решения задачи по i-й теме, которая складывается из знаний первой категории z_i^t, забываемых быстро, и знаний второй категории или навыков n_i^t, которые забываются существенно медленнее. За один акт обучения знания первой категории в количестве $a_2 z_i^t$ становятся знаниями второй категории. Коэффициенты забывания γ_1 и $\gamma_2 < \gamma_1$ определяют быстроту уменьшения знаний первой и второй категории. Коэффициент научения a тем больше, чем больше средний уровень обучения $p_{ср}$ (количество знаний), чем меньше длина задачи $L_з$ и усталость $K_{уст}$ учащегося. Будем считать, что:

$$a = \frac{A}{(2 - p_{ср})L_з(1 + K_{уст})}.$$

Используется компьютерная программа ПР–4.10. Результаты моделирования процесса обучения по учебной программе ПОЗ = {a, 6; p, 20; a, 9; p, 20; b, 6; p, 20; b, 9; p, 20; c, 6; p, 20; c, 9; p, 20; abc, 10; p, 20; abc, 10; p, 20} представлены на рис. 4.12. Большие точки показывают изменение суммарных знаний (вероятности p) с течением времени, а маленькие точки – изменение знаний второй категории (навыков n).

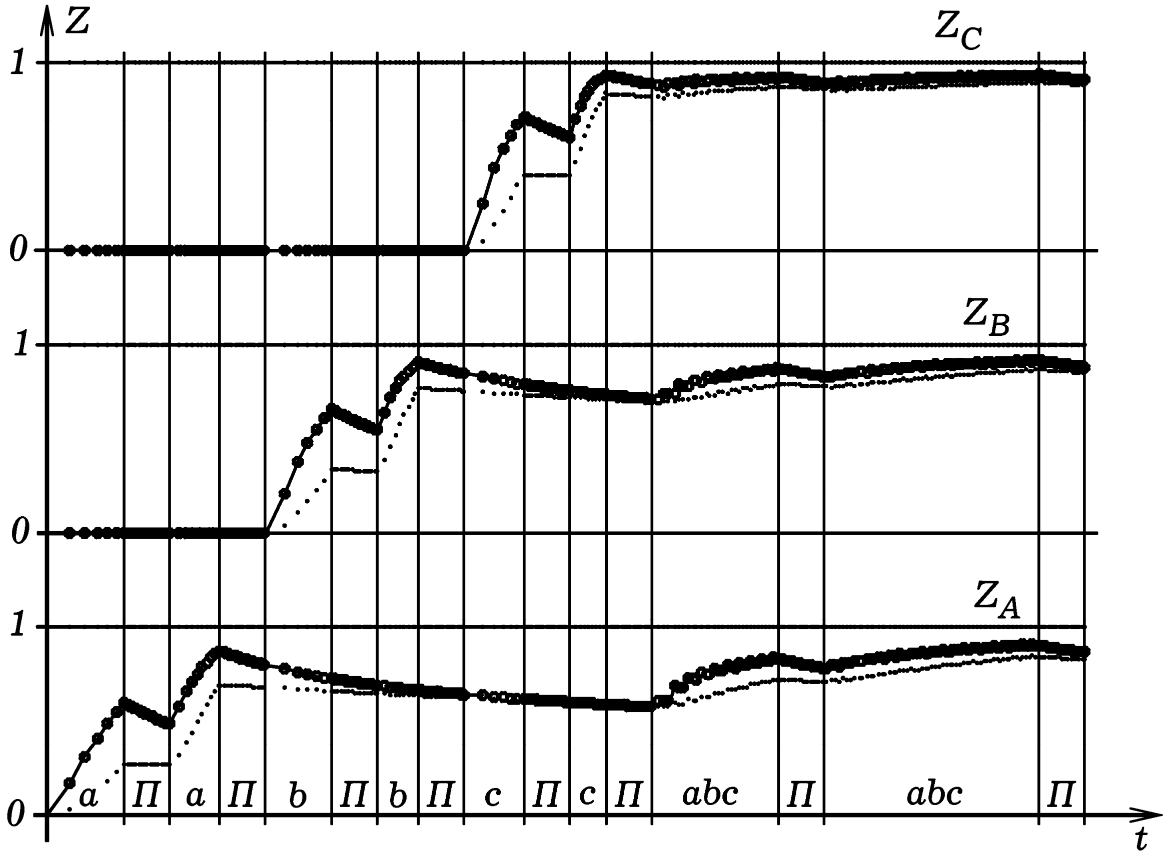


Рис. 4.12. Результаты использования многокомпонентной модели обучения

4.7. Приложение к главе 4

```
{$N+}Uses crt, graph;                                          {ПР-4.1}
const  N=10; al1=0.02; al2=0.06; T_p=100;               {Free Pascal}
var p,otv : array[1..10]of real;
Gd,Gm,t,i,k:integer; sl,al,sum,d:real;
S11,sss,tt : array[1..500]of real; Label mm;
Procedure Test;
begin  sum:=0;
For i:=1 to 10 do begin sl:=random(100)/100; t:=t+1;
If sl>p[i] then otv[i]:=0 else otv[i]:=1;
write(otv[i]:2:2,' '); sum:=sum+p[i];
end; writeln(sum,' ',t); sss[k]:=sum; tt[k]:=t; S11[k]:=sum+d;
end;
BEGIN clrscr; Randomize;
Repeat inc(k); Test;
For i:=1 to 10 do begin
If otv[i]=1 then p[i]:=p[i]+al1*(1-p[i]);
If otv[i]=0 then begin p[i]:=p[i]+al2*(1-p[i]); t:=t+1; end; end;
If sum>8 then begin t:=t+T_p; d:=sum+d;
For i:=1 to 10 do p[i]:=0; end;
until (KeyPressed)or(k>300); {readkey;}
Gd:=Detect; InitGraph(Gd,Gm,'c:\bp\bgi'); line(0,450,640,450);
```

```
For k:=1 to 150 do begin
circle(10+round(tt[k]/3),450-round(15*s11[k]),1);
circle(10+round(tt[k]/3),450-round(15*s11[k]),2); end;
Repeat until (KeyPressed); CloseGraph;
END.
```

```
{$N+} uses crt, graph;                                   {ПР-4.2}
const N=50; dt=0.002; a=0.2; tob=2; dZ=0.1; g=0.0002;
var ttt,i,j,k,l,t,DV,MV : integer;
otv,p,b: array[1..10] of single; Z,sl,sum:single;
rez,Zn: array[1..100]of single; tt: array[1..100]of integer;
Label mm,m1,m2;
Procedure Test;
begin sum:=0;
For i:=1 to 10 do p[i]:=1/(1+exp(-a*(Z-b[i])));
For i:=1 to 10 do begin t:=t+1;
If p[i]>0.8 then otv[i]:=1 else otv[i]:=0;
{write(otv[i]:2:1,' ');} sum:=sum+otv[i]; end;
{writeln(sum:2:1,' ',Z:2:2,' ',t);} Zn[k]:=Z; tt[k]:=t;
rez[k]:=sum; end;
Procedure Draw;
Begin circle(10+round(t/5),450-round(rez[k]*10),2);
circle(10+round(t/5),450-round(Z*10),2); end;
BEGIN  Randomize;
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
For i:=1 to 10 do b[i]:=i; Z:=2;
Repeat inc(k); Test;
For i:=1 to 10 do If otv[i]=0 then begin Z:=Z+dZ; t:=t+tob; end;
Draw;
until (KeyPressed)or(sum>7);
For ttt:=1 to 500 do begin t:=t+1; Z:=Z-g*Z; Draw; end;
writeln('pppp');
For i:=1 to 10 do b[i]:=8+1*i;
Repeat inc(k); Test;
For i:=1 to 10 do if otv[i]=0 then begin Z:=Z+dZ; t:=t+tob; end;
Draw;
until (KeyPressed)or(sum>7);
For i:=1 to 10 do b[i]:=16+1*i;
For ttt:=1 to 500 do begin t:=t+1; Z:=Z-g*Z; Draw; end;
writeln('pppp');
Repeat inc(k); Test;
For i:=1 to 10 do if otv[i]=0 then begin Z:=Z+dZ; t:=t+tob; end;
Draw;
until (KeyPressed)or(sum>7);
For ttt:=1 to 500 do begin inc(t); Z:=Z-g*Z; Draw; end;
Line(0,450,640,450); {readkey;}
{For k:=1 to 50 do
circle(10+round(tt[k]/3),450-round(Zn[k]*10),2);}
Repeat until KeyPressed;
CloseGraph;
END.
```

```
uses crt,graph;                                          {ПР-4.3}
const  N=10; K_is=100; a1=0.1; a2=0.2; dt1=1; dt2=1.5;
var kk,k,l,i,i1,i2,j,DV,MV,Uch: integer; S,sl,q,t,ttt,SSS:real;
verno,pnv,p: array[0..10] of real;
N1,N2,Z1,Z2,P_sr,P_zad,dp,x,sum_p,sum,tt : real;
Label m1;
BEGIN Randomize; {clrscr;}
DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
Repeat t:=0; SSS:=0; ttt:=0;
For i:=1 to 10 do p[i]:=0.0;
Repeat For i:=1 to 10 do begin { Reshaet zadachu } m1:
sl:=random(100)/100; q:=(1-p[i])/3; t:=t+dt1;
If sl<p[i] then verno[i]:=1 else verno[i]:=0;
If sl>1-q then pnv[i]:=1 else pnv[i]:=0;
{writeln(i,' ',verno[i]:2:1,' ',pnv[i]:2:1,' ',p[i]);}
If pnv[i]=1 then begin p[i]:=p[i]+a1*(1-p[i]); t:=t+dt2;
goto m1; end;
end; {Proverka uchitelem, obuchenie} Uch:=0;
For i:=1 to 10 do
If verno[i]=0 then begin p[i]:=p[i]+a2*(1-p[i]); t:=t+dt2;
inc(Uch); end;
S:=0; For i:=1 to 10 do S:=S+p[i]; {writeln(' ',S/10:2:1);}
circle(10+round(t/2.6),450-round(40*S),1);
line(10+round(t/2.6),450-round(40*S),
     10+round(ttt/2.6),450-round(40*SSS));
ttt:=t; SSS:=S; line(0,450,640,450);
circle(10+round(t/2.6),450-round(30*Uch),1);
until (KeyPressed)or(S>9.5); inc(kk);
until kk>10;
Readkey; CloseGraph;
END.
```

```
uses crt, graph;                                        {ПР-4.4}
const  dt=0.1; N=3; M=200; Mz=180; a=0.015;
var flag,i,j,DV,MV : integer;
al,Z,Z1,Z2,Z3,Zk,M1,vagn: real;
p: array[0..N] of real; ZZ: string;
n1,nomer: array[0..M] of word;
Procedure Schet;
begin For j:=1 to N do p[j]:=0;
 For i:=1 to M do begin
  Z1:=p[1]; Z2:=p[2]; Z3:=p[3]; al:=a;
  If nomer[i]=2 then al:=a*Z1+0.01;
  If nomer[i]=3 then al:=a*(Z1+Z2);
  p[nomer[i]]:=p[nomer[i]]+al*(1-p[nomer[i]]);
  If flag=1 then begin circle(10+2*i,400-round(Mz*Z1),2);
  circle(10+2*i,400-round(Mz*Z2),2);
  circle(10+2*i,400-round(Mz*Z3),2); end;
 end; Z:=0; vagn:=0;
 For j:=1 to N do begin
  If j=2 then vagn:=1;
  If j=3 then vagn:=5;
```

```
  Z:=Z+vagn*p[j]; end;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
Repeat Schet; Zk:=Z;
 For i:=1 to M do n1[i]:=nomer[i];
 For i:=1 to 3 do begin j:=random(M+1);
 nomer[j]:=round(random(350)/100); end; Schet;
 If Z<Zk then For i:=1 to M do nomer[i]:=n1[i];
 If Z>Zk then begin cleardevice; flag:=1; Schet;
 flag:=0; line(0,400,640,400); line(10,0,10,400);
 For j:=1 to M do circle(10+2*j,400-100*nomer[j],2);
 circle(600,400-round(Z*40),2); Zk:=Z; Str(Z,ZZ);
 OutTextXY(20,10,ZZ); end;
until (KeyPressed); CloseGraph;
END.
```

```
uses crt, graph;                                              {ПР-4.5}
const  dt=0.1; N=6; M=200; Mz=180; a=0.04;
var flag,i,j,DV,MV : integer; al,Z,Z1,Z2,Z3,Zk,M1,vagn: real;
p: array[0..N+1] of real; n1,nomer: array[0..M+1] of word;
ZZ: string;
Procedure Schet;
begin For j:=1 to N do p[j]:=0;
For i:=1 to M do begin al:=a;
  If nomer[i]>1 then al:=a*p[nomer[i]-1];
  p[nomer[i]]:=p[nomer[i]]+al*(1-p[nomer[i]]);
  If flag=1 then For j:=1 to N do
       circle(10+2*i,400-round(Mz*p[j]),2); end; Z:=0;
For j:=1 to N do begin vagn:=1; Z:=Z+vagn*p[j]; end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
Repeat Schet; Zk:=Z;  For i:=1 to M do n1[i]:=nomer[i];
   For i:=1 to 3 do begin j:=random(M+1);
   nomer[j]:=round(random(650)/100); end; Schet;
   If Z<Zk then For i:=1 to M do nomer[i]:=n1[i];
   If Z>Zk then begin cleardevice; flag:=1; Schet;
   flag:=0; line(0,400,640,400); line(10,0,10,400);
   For j:=1 to M do circle(10+2*j,400-50*nomer[j],2);
   circle(600,400-round(Z*10),2); Zk:=Z; Str(Z,ZZ);
   OutTextXY(20,10,ZZ); end;
until KeyPressed; CloseGraph;
END.
```

```
uses crt, graph;                                              {ПР-4.6}
const a1=0.01; a2=0.002; N=3; Ur=90;
g1=0.005; g2=0.0001; dt=0.8; Mt=0.35; Mz=2;
var Dl,Dl1,min,UU,kk1,kk2,bb1,bb2,k1,k2,b1,b2,t,
Z1,Z2,Z3,Z4,Z,Zn,Zn2,Pr,Sum,g,S1,k:real; i,kk:integer;
S,U,U1: array[0..N] of real; F:string; DV,MV:integer;
Procedure Raschet;
begin  t:=0; Sum:=0; Z1:=0; Z2:=0; Zn:=0;
```

```
For i:=1 to N do S[i]:=0;
Repeat t:=t+dt; k:=0; i:=round(t/10);
If t<Dl then begin
  UU:=U[1]; k:=1; S[1]:=S[1]+k*abs(UU-Z)*dt; end;
If (t>500)and(t<500+Dl) then begin
  UU:=U[2]; k:=1; S[2]:=S[2]+k*abs(UU-Z)*dt; end;
If (t>1000)and(t<1000+Dl) then begin
  UU:=U[3]; k:=1; S[3]:=S[3]+k*abs(UU-Z)*dt; end;
If UU>=Z then Z1:=Z1+k*a1*(UU-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
If UU<Z then Z1:=Z1-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; Z:=Z1+Z2;
Sum:=Sum+k*abs(UU-Z)*dt;
until (t>1600)or(KeyPressed); Zn:=Z; Zn2:=Z2; end;
Procedure Draw;
begin t:=0;  Z1:=0; Z2:=0; Z:=0; cleardevice;
Repeat t:=t+dt; k:=0;
If t<Dl then begin UU:=U[1]; k:=1; end;
If (t>500)and(t<500+Dl) then begin UU:=U[2]; k:=1; end;
If (t>1000)and(t<1000+Dl) then begin UU:=U[3]; k:=1; end;
If UU>=Z then Z1:=Z1+k*a1*(UU-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
If UU<Z then Z1:=Z1-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; Z:=Z1+Z2;
circle(10+round(Mt*t),450-round(Mz*Z),1);
circle(10+round(Mt*t),450-round(Mz*Ur),1);
circle(10+round(Mt*t),450-round(Mz*(Z2)),1);
circle(10+round(Mt*t),450-round(Mz*(UU)),1);
until (t>1600)or(KeyPressed); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Randomize; min:=10000000; Dl:=300;
For i:=1 to N do U[i]:=70*i;
{U[1]:=58; U[2]:=97; U[3]:=100; U[4]:=170; U[5]:=210;}
Repeat
Dl1:=Dl; Raschet; S1:=Sum;
if Sum<min then min:=Sum;
Dl:=Dl+random(160)/10-8; if Dl>300 then Dl:=300; Raschet;
For i:=1 to N do if S[i]>150*Dl then g:=1;
If (g=1)or(Sum>min)or(Zn<Ur)or(Zn2<0.6*Ur) then begin
  Dl:=Dl1; end; g:=0;
If (g=0)and(Sum<=min)and(Zn>Ur)and(Zn2>0.6*Ur) then begin
Draw; circle(10,round(min/200),3);
str(Sum,F);  OutTextXY(20,20,F);
str(U[1],F); OutTextXY(20,30,F);
str(U[2],F); OutTextXY(20,40,F);
```

```
str(Dl,F);   OutTextXY(20,50,F); {delay(65000); }
end; {str(Sum,F); Draw; OutTextXY(20,20,F); delay(5);}
until Keypressed; CloseGraph;
END.

uses crt, graph;                                         {ПР-4.7}
const a1=0.01; a2=0.002; N=5; Ur=90; g1=0.005; g2=0.0001;
dt=0.8; Mt=0.3; Mz=2.1;
var min,UU,t,Z1,Z2,Z,Zn,Zn2,Pr,Sum,S1,g: real;
i,k,kk,DV,MV:integer; S,U,U1: array[0..N] of real; F:string;
Procedure Raschet;
begin t:=0; Sum:=0; Z1:=0; Z2:=0; Zn:=0;
For i:=1 to N do S[i]:=0;
Repeat t:=t+dt; k:=0; i:=round(t/10);
If t<200 then begin UU:=U[1]; k:=1;
  S[1]:=S[1]+k*abs(UU-Z)*dt; end;
If (t>400)and(t<600) then begin
  UU:=U[2]; k:=1; S[2]:=S[2]+k*abs(UU-Z)*dt; end;
If (t>800)and(t<1000) then begin
  UU:=U[3]; k:=1; S[3]:=S[3]+k*abs(UU-Z)*dt;end;
If (t>1200)and(t<1400) then begin
  UU:=U[4]; k:=1; S[4]:=S[4]+k*abs(UU-Z)*dt;end;
If (t>1600)and(t<1800) then begin
  UU:=U[5]; k:=1; S[5]:=S[5]+k*abs(UU-Z)*dt;end;
If UU>=Z then Z1:=Z1+k*a1*(UU-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
If UU<Z then Z1:=Z1-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; Z:=Z1+Z2;
Sum:=Sum+k*abs(UU-Z)*dt;
until (t>2200)or(KeyPressed); Zn:=Z; Zn2:=Z2; end;
Procedure Draw;
begin t:=0;  Z1:=0; Z2:=0; Z:=0; cleardevice;
Repeat t:=t+dt; k:=0;
If t<200 then begin UU:=U[1]; k:=1; end;
If (t>400)and(t<600) then begin UU:=U[2]; k:=1; end;
If (t>800)and(t<1000) then begin UU:=U[3]; k:=1; end;
If (t>1200)and(t<1400) then begin UU:=U[4]; k:=1; end;
If (t>1600)and(t<1800) then begin UU:=U[5]; k:=1; end;
If UU>=Z then Z1:=Z1+k*a1*(UU-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
If UU<Z then Z1:=Z1-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; Z:=Z1+Z2;
circle(10+round(Mt*t),450-round(Mz*Z),1);
circle(10+round(Mt*t),450-round(Mz*Ur),1);
circle(10+round(Mt*t),450-round(Mz*(Z2)),1);
```

```
circle(10+round(Mt*t),450-round(Mz*(UU)),1);
until (t>2200)or(KeyPressed); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Randomize; min:=1E+7;
For i:=1 to N do U[i]:=20+40*i;
Repeat
For i:=1 to N do U1[i]:=U[i]; Raschet; S1:=Sum;
If Sum<min then min:=Sum;
For i:=1 to N do begin U[i]:=U[i]+random(800)/100-4;
If U[i]<0 then U[i]:=0; end; Raschet;
If (Sum>min)or(Zn<Ur)or(Zn2<0.6*Ur) then
For i:=1 to N do U[i]:=U1[i];
g:=0; For i:=1 to N do if S[i]>75*200 then g:=1;
If (g=0)and(Sum<=min)and(Zn>Ur)and(Zn2>0.6*Ur) then begin
Draw; circle(10,round(min/200),3);
str(Sum,F); OutTextXY(20,20,F);
str(U[1],F); OutTextXY(20,30,F);
str(U[2],F); OutTextXY(20,40,F);
str(U[3],F); OutTextXY(20,50,F);
str(U[4],F); OutTextXY(20,60,F);
str(U[5],F); OutTextXY(20,70,F); end;
until Keypressed; CloseGraph;
END.
```

```
uses crt;                                                    {ПР-4.8}
var m,q,i,j,k: integer;
y: string; slovo: array[1..5] of string;
zn:array[1..5] of string; znn: string; Label m1,m2;
BEGIN clrscr;
slovo[1]:='a_';    zn[1]:='reakciya1';
slovo[2]:='ab_';   zn[2]:='reakciya2';
slovo[3]:='abc_';  zn[3]:='reakciya3';
slovo[4]:='abcd_'; zn[4]:='reakciya4';
slovo[5]:='cdbc_'; zn[5]:='reakciya5'; m:=5; q:=0;
For j:=1 to length(slovo[m]) do begin y:=copy(slovo[m],j,1);
If (q=0)and(y='a') then begin q:=1; znn:='OK';  goto m1; end;
If (q=1)and(y='_') then begin q:=9; znn:=zn[1]; goto m1; end;
{ obuchenie slovu 2 }
If (q=1)and(y='b') then begin q:=2; znn:='OK';  goto m1; end;
if (q=2)and(y='_') then begin q:=9; znn:=zn[2]; goto m1; end;
{ obuchenie slovu 3 }
if (q=2)and(y='c') then begin q:=3; znn:='OK';  goto m1; end;
if (q=3)and(y='_') then begin q:=9; znn:=zn[3]; goto m1; end;
{ obuchenie slovu 4 }
if (q=3)and(y='d') then begin q:=4; znn:='OK'; goto m1; end;
if (q=4)and(y='_') then begin q:=9; znn:=zn[4]; goto m1; end;
```

```
{ obuchenie slovu 5 }
if (q=0)and(y='c') then begin q:=5; znn:='OK'; goto m1; end;
if (q=5)and(y='d') then begin q:=6; znn:='OK'; goto m1; end;
if (q=6)and(y='b') then begin q:=7; znn:='OK'; goto m1; end;
if (q=7)and(y='c') then begin q:=8; znn:='OK'; goto m1; end;
if (q=8)and(y='_') then begin q:=9; znn:=zn[5]; goto m1; end;
znn:='neponimau'; writeln(' ',q,' zz: ',znn);  goto m2;
m1: writeln(' ',q,' zz: ',znn);
end; m2: readkey;
END.
```

```
uses crt, graph;                                           {ПР-4.9}
const  alfa=0.4; beta=0.1; dt=0.03;
var i,j,j1,k,z,chislo,DV,MV: integer;
zadacha : array[0..10] of string; y: string;
g,t,t0,t1: real; p: array[1..5]of real;
Procedure Raschet;
begin k:=0;
Repeat
  For i:=1 to length(zadacha[z]) do begin
    y:=copy(zadacha[z],i,1);
    If y='a' then j:=1;   If y='b' then j:=2;
    If y='c' then j:=3;   If y='_' then j:=4; { peremena }
    p[j]:=p[j]+alfa*(1-p[j]);
    If p[j]<=1 then t:=t+t0-ln(p[j])/beta; t1:=0; p[4]:=1;
    If j=4 then g:=0.003 else g:=0.001;
    Repeat For j1:=1 to 3 do begin
    If (j1<>j)and(p[j1]>0) then p[j1]:=p[j1]-g*p[j1]*dt;
    end; t1:=t1+dt;
    until t1>t0-ln(p[j])/beta;
    circle(10+round(1.2*t),450-round(p[1]*100),2);
    circle(10+round(1.2*t),300-round(p[2]*100),2);
    circle(10+round(1.2*t),150-round(p[3]*100),2);
  end; inc(k);
until k>=chislo;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(10,0,10,480); line(0,450,640,450); t0:=3;
zadacha[1]:='a';  zadacha[2]:='b';
zadacha[3]:='ab'; zadacha[4]:='c';
zadacha[5]:='bc'; zadacha[6]:='abc'; zadacha[7]:='_';
{--- Programma obucheniya ---}
z:=1; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=3; chislo:=8; Raschet; z:=7; chislo:=18; Raschet;
z:=6; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=6; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
readkey;  CloseGraph;
END.
```

```
uses crt, graph;                                           {ПР-4.10}
const dt=0.001; g1=0.01; g2=0.0005; beta=0.1; Mt=0.6;
var i,j,r,s,k,chislo,DV,MV : integer;
zadacha : array[0..10] of string; y: string;
a1,a2,t,t0,t1,Ust,SP: real; pp,p,z,n: array[1..5]of real;
Procedure Raschet;
begin k:=0;
Repeat For i:=1 to length(zadacha[s]) do begin
    y:=copy(zadacha[s],i,1);
    If y='a' then j:=1; If y='b' then j:=2;
    If y='c' then j:=3; If y='_' then j:=4; { peremena }
    SP:=(p[1]+p[2]+p[3])/3;
    a1:=0.3/(2-SP)/length(zadacha[s])/(Ust+1);
    a2:=0.3/(2-SP)/length(zadacha[s])/(Ust+1);
    If j<4 then begin Ust:=Ust+(p[j]-pp[j]);
    z[j]:=z[j]+a1*(1-p[j])-a2*z[j];
    n[j]:=n[j]+a2*z[j]; end else Ust:=0;
    pp[j]:=p[j]; p[j]:=z[j]+n[j]; p[4]:=1;
    If (p[j]<=1)and(p[j]>0)then t:=t+t0-ln(p[j])/beta; t1:=0;
    Repeat For r:=1 to 3 do begin
      If (r<>j)and(z[r]>0) then z[r]:=z[r]-g1*z[r]*dt;
      If (r<>j)and(n[r]>0) then n[r]:=n[r]-g2*n[r]*dt;
      p[r]:=z[r]+n[r]; end; t1:=t1+dt;
    until (t1>t0-ln(p[j])/beta)or(KeyPressed);
    circle(10+round(Mt*t),450-round(p[1]*100),2);
    circle(10+round(Mt*t),450-round(n[1]*100),1);
    circle(10+round(Mt*t),300-round(p[2]*100),2);
    circle(10+round(Mt*t),300-round(n[2]*100),1);
    circle(10+round(Mt*t),150-round(p[3]*100),2);
    circle(10+round(Mt*t),150-round(n[3]*100),1);
end; inc(k); until (k>=chislo)or(KeyPressed);
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(10,0,10,480); line(0,450,640,450); t0:=2;
For j:=1 to 3 do p[j]:=0.1;
zadacha[1]:='a';  zadacha[2]:='b';
zadacha[3]:='c'; zadacha[4]:='ab';
zadacha[5]:='bc'; zadacha[6]:='abc'; zadacha[7]:='_';
{--- Programma obucheniya ---}
s:=1; chislo:=6; Raschet; s:=7; chislo:=20; Raschet;
s:=1; chislo:=9; Raschet; s:=7; chislo:=20; Raschet;
s:=2; chislo:=6; Raschet; s:=7; chislo:=20; Raschet;
s:=2; chislo:=9; Raschet; s:=7; chislo:=20; Raschet;
s:=3; chislo:=6; Raschet; s:=7; chislo:=20; Raschet;
s:=3; chislo:=9; Raschet; s:=7; chislo:=20; Raschet;
s:=6; chislo:=10; Raschet; s:=7; chislo:=20; Raschet;
s:=6; chislo:=20; Raschet; s:=7; chislo:=20; Raschet;
readkey;  CloseGraph;
END.
```

Глава 5.
КИБЕРНЕТИЧЕСКАЯ ПЕДАГОГИКА
И ПРОЦЕСС ОБУЧЕНИЯ

В настоящей главе рассматривается метод согласования математический модели обучения с распределением учебной информации и результатами тестирования, анализируется алгоритм деятельности учащегося при выполнении учебного задания, обсуждается проблема использования информационных технологий в процессе обучения.

5.1. Согласование модели обучения с результатами тестирования

Компьютерные модели, рассмотренные в предыдущих главах, позволяют изучить динамику увеличения знаний некоторой абстрактной модели ученика. Чтобы они описывали среднестатистического учащегося и соответствовали реальной ситуации, необходимо задать определенные числовые значения параметрам абстрактной модели ученика (коэффициентам научения и забывания), определить скорость поступления учебной информации, уровень требований в течение обучения и т. д. Некоторые из этих значений могут быть получены из психологических экспериментов, другие – как результат согласования имитационной модели процесса обучения с результатами тестирования.

Пример согласования модели с результатами тестирования рассмотрен в монографии «Исследование процесса формирования системы эмпирических знаний по физике» [31; 32]. При анализе формирования эмпирических знаний все факты (ЭУМ), изучаемые в школе, были разделены на три категории: 1) факты, которые могут быть установлены экспериментально в повседневной жизни; 2) факты, которые могут быть установлены экспериментально в условиях обучения, но которые не устанавливаются в повседневной жизни; 3) факты, которые невозможно установить в условиях обучения. Было выдвинуто предположение, что факты разных категорий забываются с различной скоростью.

Разобьем учебный процесс на интервалы длительностью τ и будем считать, что внутри каждого такого интервала учебный материал распределен равномерно, то есть скорость поступления информации к учащемуся остается постоянной: $\upsilon = dI/dt = const$, причем вся информация усваивается $dZ_y = dI$. Так как быстрота увеличения знаний учащегося складывается из скорости обучения υ и скорости забывания $-\gamma \cdot Z$, то $dZ/dt = \upsilon - \gamma \cdot Z$. Считая, что в момент начала обучения t_0 количество знаний ученика $Z(t_0) = Z_0$, получаем:

$$\int_{Z_0}^{Z} \frac{dZ}{Z - \upsilon/\gamma} = -\gamma \int_{t_0}^{t} dt .$$

Отсюда следует, что количество знаний ученика в момент времени t равно:

$$Z(t) = \frac{\upsilon}{\gamma}\left[1 - e^{-\gamma(t-t_0)}\right] + Z_0 e^{-\gamma(t-t_0)}.$$

Пусть в начальный момент времени количество знаний учащегося равно нулю и $\tau = 1$ год. Количество знаний учащегося в конце $(i+1)$-го учебного года:

$$Z_{i+1} = Z_i e^{-\gamma\tau} + \frac{\upsilon_{i+1}}{\gamma}\left[1 - e^{-\gamma \cdot \tau}\right],$$

где Z_i – уровень знаний в конце i-го года, υ_{i+1} – скорость поступления знаний в $(i+1)$-го году. Это уравнение позволяет последовательно вычислить количество эмпирических знаний учащихся в конце 1, 2, …, 11 года обучения.

Так как количество знаний фактов j-го учебного года равно сумме знаний, усвоенных в 1, 2, …, i, … , j-ом классах и частично забытых в течение $(j-1)$, $(j-2)$, …, $(j-i)$, … , 1, 0 лет соответственно, то имеем [31; 26]:

$$Z_j = \sum_{i=1}^{j} \Delta Z_i e^{-\gamma(j-i)\tau} = \sum_{i=1}^{j} \frac{\upsilon_i}{\gamma}(1 - e^{-\gamma \cdot \tau}) e^{-\gamma(j-i)\tau},$$

где $\Delta Z_i = (\upsilon_i / \gamma)(1 - e^{-\gamma\tau}) e^{-\gamma(j-i)\tau}$ – знания, приобретенные в i-ом классе, $\tau = 1$ год – время обучения в одном классе.

Использование данной модели для исследования процесса формирования системы эмпирических знаний требует учета зависимости времени забывания от категории фактов. Считая, что коэффициенты забывания фактов первой, второй и третьей категорий соответственно равны γ_1, γ_2, γ_3, а их скорости поступления υ_{i1}, υ_{i2}, υ_{i3}, где $i =$ 1, 2, …, 11 – номер класса, после преобразований получаем:

$$Z_j = \sum_{k=1}^{3} Z_{jk} = \sum_{k=1}^{3}\sum_{i=1}^{j} \frac{\upsilon_{ik}}{\gamma_k}(1-\exp(-\gamma_k\tau))\exp(-\gamma_k(j-i)\tau).$$

Здесь Z_{ik} – количество знаний учащихся, соответствующее фактам k-й категории в конце j-го класса. Введем коэффициент сформированности эмпирических знаний K_j как отношение знаний фактов Z_j в j-ом классе к общему количеству эмпирической информации: $K'_j = Z_j / I_j$. При этом количество сообщенной информации равно:

$$I_j = \sum_{k=1}^{3}\sum_{i=1}^{j} \upsilon_{ik}\tau.$$

Таблица 5.1

Количество фактов 1, 2 и 3 категорий
в различных разделах школьного курса физики

Класс i	Категория факта	Механика	Молекулярная физика	Электродинамика	Оптика	Квантовая физика
7 кл.	1 2 3	32 13 1	13 0 0	0 0 0	0 0 0	0 0 0
8 кл.	1 2 3	1 0 0	13 13 0	12 32 3	9 4 0	0 0 0
9 кл.	1 2 3	27 23 1	0 0 0	0 0 0	0 0 0	0 0 0
10 кл.	1 2 3	2 2 0	9 13 3	16 42 8	0 0 0	0 0 0
11 кл.	1 2 3	2 2 1	0 0 0	2 28 8	3 13 6	2 13 37

В результате контент-анализа стандартных учебников природоведения, физической географии и физики [28; 31] были определены значения скоростей поступления эмпирических знаний в разных классах по различным разделам физики для фактов 1, 2 и 3 категорий в единицах измерения факт./год. При этом считалось, что количество эмпирических знаний пропорционально числу рисунков, несущих эмпирическую информацию. Результаты для школьных учебников физики представлены в табл. 5.1.

С целью определения коэффициентов забывания в 1997–1999 гг. было проведено тестирование около 100 студентов 1 курса Глазовского государственного педагогического института, заключавшееся в установлении уровня знаний данными студентами 50 учебных фактов (по 10 из каждого раздела физики). Это позволило оценить коэффициент сформированности эмпирических знаний по фактам различных категорий как отношение числа заданных вопросов N к числу правильных ответов n: $K = n / N$. Полученные результаты – во втором столбце табл. 5.2.

Таблица 5.2

Согласование модели с результатами тестирования

		ТЕСТ	МОДЕЛЬ			
			1	2	3	4
Уровень	K_1	0,72	0,30	0,50	0,73	0,73
знаний	K_2	0,35	0,43	0,62	0,17	0,35
фактов	K_3	0,19	0,60	0,19	0,32	0,19
Коэффи-	γ_1	—	0,38	0,20	0,090	0,090
циенты	γ_2	—	0,38	0,20	0,95	0,49
забывания	γ_3	—	0,38	1,48	0,95	1,5
Критерий S		—	0,36	0,12	0,049	0,00016

Задача согласования математической модели с результатами тестирования сводится к определению таких значений γ_1, γ_2, γ_3, при которых коэффициенты сформированности эмпирических знаний K'_k для фактов различных категорий k = 1, 2, 3, предсказываемые моделью, максимально близки к соответствующим значениям K_k, полученным при тестировании. Для этого нами использовался метод наименьших квадратов, заключающийся в минимизации суммы квадратов разностей $(K_k - K'_k)^2$, где k =1, 2, 3:

$$S = \sum_{k=1}^{3} (K_k - K'_k)^2 = \min .$$

Для оптимизации параметров γ_1, γ_2, γ_3 и моделирования формирования эмпирических знаний создан пакет программ, который включал в себя:

1. Массив данных о распределении фактов в школьном курсе физики, а также данные, полученные в результате тестирования. Из них следует, что коэффициенты сформированности у учащихся знаний фактов первой, второй и третьей категорий соответственно равны $K_1 = 0{,}72$, $K_2 = 0{,}35$, $K_3 = 0{,}19$.

2. Подпрограмма, позволяющая, исходя из данных о распределении фактов в курсе физики и значений γ_1, γ_2, γ_3, вычислить теоретические значения уровней сформированности фактуальных знаний, касающиеся фактов различных категорий K'_k (k = 1, 2, 3), для момента времени t = 11,5 лет.

3. Подпрограмма, определяющая значение суммы S квадратов разностей между K_k и K'_k.

4. Программа, минимизирующая сумму S, то есть определяющая значения γ_k (k = 1, 2, 3), при которых величина S минимальна. Она осуществляет подгонку модели к результатам тестирования.

5. Программа, осуществляющая имитационное моделирование процесса формирования у учащихся знаний фактов, относящихся к первой, второй и третьей категориям или различным разделам физики.

5.2. Результаты имитационного моделирования формирования эмпирических знаний по физике

Чтобы убедиться в необходимости деления фактов на три категории, нами проведены расчеты для трех случаев: 1) факты всех трех категорий имеют одинаковые коэффициенты забывания: $\gamma_1 = \gamma_2 = \gamma_3$; 2) факты первой и второй категорий имеют одинаковые коэффициенты забывания $\gamma_1 = \gamma_2$, отличные от коэффициента забывания фактов третьей категории γ_3; 3) факты второй и третьей категорий имеют одинаковые коэффициенты забывания $\gamma_2 = \gamma_3$, отличные от коэффициента забывания фактов первой категории γ_1; 4) коэффициенты забывания фактов γ_1, γ_2, γ_3 первой, второй и третьей категорий различны. Критерием близости результатов моделирования и тестирования является сумма S, которая при их совпадении обращается в нуль. Значения S

представлены в третьем, четвертом, пятом и шестом столбцах табл. 5.2 соответственно.

Видно, что четвертая модель, учитывающая различие коэффициентов забывания фактов первой, второй и третьей категорий, наиболее точно соответствует результатам тестирования. Итак, искомые коэффициенты забывания равны $\gamma_1 = 0{,}090$ лет$^{-1}$, $\gamma_2 = 0{,}49$ лет$^{-1}$, $\gamma_3 = 1{,}5$ лет$^{-1}$. Интересно, что небольшие вариации момента времени t проведения контрольного тестирования приводят к тому, что программа выдает практически те же значения γ_1, γ_2 при немного другом γ_3. Так, если задать $t =$ 11,75 лет, то $\gamma_1 =$ 0,085 лет$^{-1}$, $\gamma_2 = 0{,}43$ лет$^{-1}$, $\gamma_3 = 1{,}2$ лет$^{-1}$. Это связано с тем, что факты третьей категории изучаются в основном в 10–11 классах и достаточно быстро забываются.

Получению описанных выше результатов предшествовали попытки решения данной задачи другим способом [31; 26]. Он отличается тем, что количество эмпирической информации в учебнике оценивалось не по количеству фактов, а по числу рисунков, на которых изображены схемы экспериментов или их результаты. Кроме того, согласование модели с результатами тестирования осуществлялось путем сравнения уровня сформированности эмпирических знаний по механике, молекулярной физике и термодинамике, электродинамике, оптике и квантовой физике с соответствующими значениями, предсказываемыми моделью. Полученные при этом результаты таковы: $\gamma_1 \approx$ 0,06 лет$^{-1}$, $\gamma_2 \approx$ 0,3 лет$^{-1}$, $\gamma_3 \approx$ 2 лет$^{-1}$.

Границы применимости предложенной модели определяются погрешностью введенных в модель данных, а также влиянием целого ряда неучтенных и неконтролируемых факторов. Например, при моделировании предполагалось, что изучение физики осуществляется без перерывов, скорость поступления информации, касающейся фактов определенной категории, в течение года постоянна, не учитывалась подготовка учащихся к выпускным и вступительным экзаменам и т. д. Учет этих факторов требует существенного усложнения модели, введения новых переменных, для оценки которых потребовалось бы более обширное тестирование.

Согласно результатам моделирования наиболее быстро забываются факты третьей категории, изучаемые на чисто умозрительном уровне. Их коэффициент забывания $\gamma_3 \approx 1{,}5$ лет$^{-1}$, период забывания половины информации $T_3 = \ln 2 / \gamma_3 = 0{,}46$ лет. Факты, изучаемые с опорой на систему учебного эксперимента, забываются несколько медленнее: $\gamma_2 \approx 0{,}49$ лет$^{-1}$, $T_2 = \ln 2 / \gamma_2 = 1{,}4$ года. Факты первой категории, которые учащийся может экспериментально установить в повседневной жизни, забываются еще медленнее или практически не забываются: $\gamma_1 \approx 0{,}090$ лет$^{-1}$, $T_1 = \ln 2 / \gamma_1 = 7{,}7$ лет.

Таблица 5.3

Количество эмпирических знаний учащихся, предсказываемое имитационной моделью

Год	По категориям фактов			По разделам физики					Общее число фактов
	1	2	3	1	2	3	4	5	
7	43	10	2,1	42	13	0,0	0,0	0,0	55
8	73	45	2,0	36	34	38	12	0,0	120
9	92	46	0,98	75	28	26	9,8	0,0	139
10	110	73	6,0	65	44	72	8,4	0,0	190
11	109	89	28	59	35	78	24	31	227
12	100	55	6,5	50	29	53	16	12	161
13	92	34	1,5	44	25	39	12	6,5	127
14	84	21	0,33	39	22	30	10	4,1	105
15	77	13	0,08	35	19	24	8,2	2,9	89
16	69	7,8	0,02	31	17	20	7,0	2,1	78

Полученные значения γ_1, γ_2, γ_3 позволяют построить графики зависимостей количества эмпирических знаний от времени. В табл. 5.3 представлены результаты вычислений, а на рис. 5.1 и 5.2 изображены соответствующие им кривые для фактов различных категорий, всего курса физики в целом и для его различных разделов. Кривые на рис. 5.1 получены в результате определения количества эмпирических знаний с учетом подсчета числа рисунков, несущих эмпирическую информацию, в учебниках физики, природоведения и физической географии.

Графики на рис. 5.2 и 5.3 получены на основе данных, взятых из табл. 5.1. На рис. 5.2 показаны изменения эмпирических знаний учащихся в целом, а также знаний фактов первой, второй и третьей категорий с течением времени. На рис. 5.3.1 и 5.3.2 показано, как изменяется количество эмпирических знаний по разделам физики: по механике (кривая 1), молекулярной физике и термодинамике (кривая 2), электродинамике (кривая 3), оптике (кривая 4), квантовой физике (кривая 5). Провалы в графиках связаны с забыванием в промежутках между первым и вторым изучением данного раздела физики, происходящими на первой и второй ступени обучения. Видно, что эмпирические знания по различным разделам физики забываются с разной скоростью. Например, уровень сформированности эмпирических знаний по механике, состоящих преимущественно из фактов первой категории, уменьшается медленно (рис. 5.3.1, кривая 1), в то время как факты квантовой физики, в основном относящиеся к третьей категории, забываются быстрее (рис. 5.3.2, кривая 5).

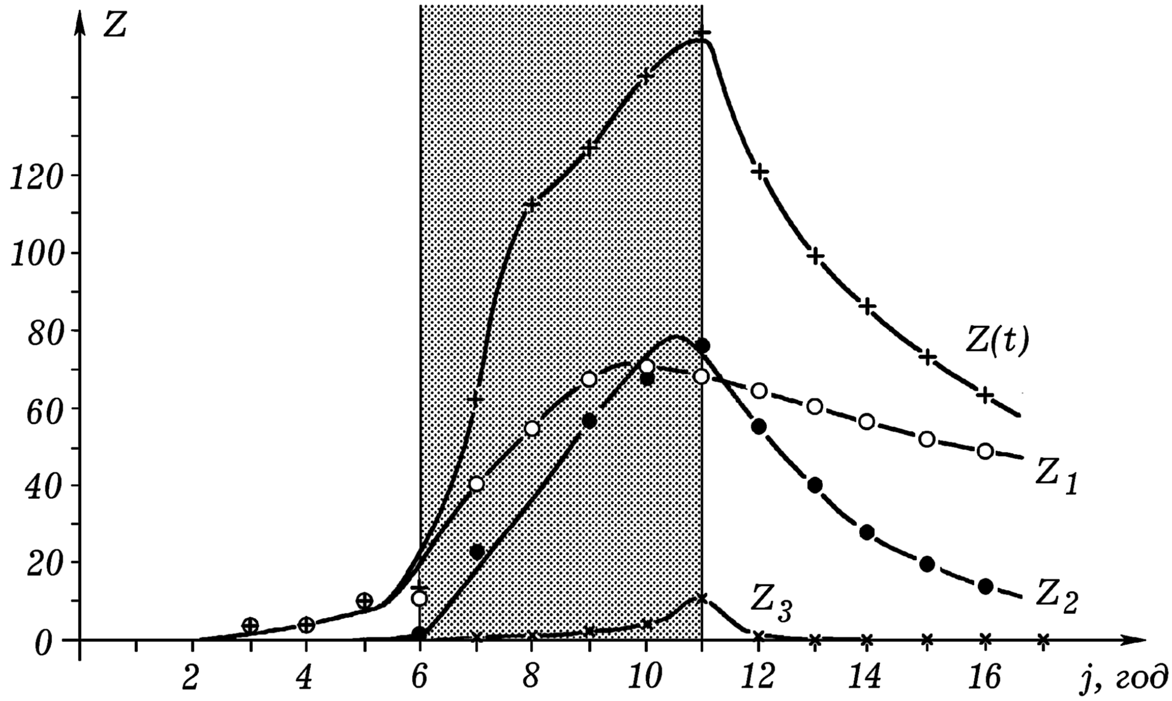


Рис. 5.1. Зависимость знаний фактов различных категорий от времени

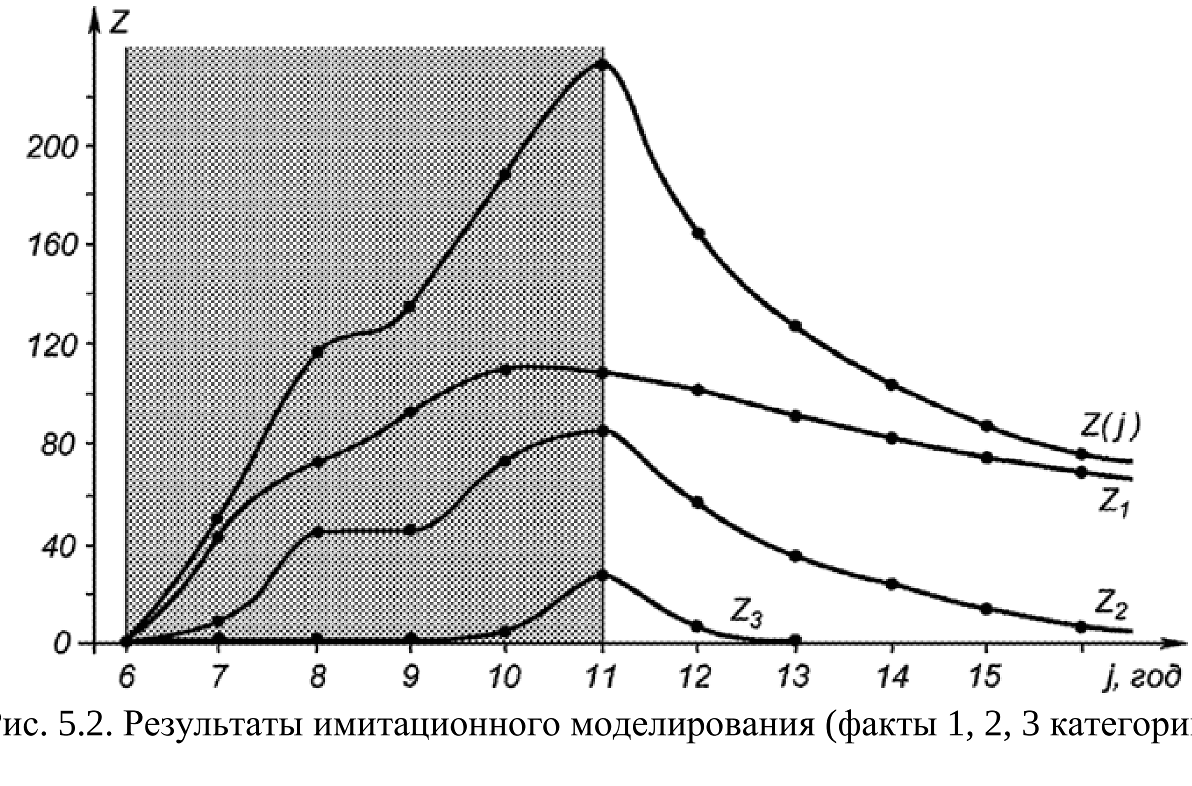


Рис. 5.2. Результаты имитационного моделирования (факты 1, 2, 3 категории)

Из рис. 5.1 и 5.2 следует, что уровень знаний фактов первой категории плавно возрастает до некоторого значения, а затем остается практически неизменным. Уровни знаний фактов второй и третьей категорий плавно возрастают, в конце обучения достигают максимума, а затем экспоненциально уменьшаются. Соответствие результатов моделирования педагогическому опыту подтверждает истинность исходных положений модели. Это прежде всего относится к гипотезе о целесообразности деления фактов на три категории, что позволяет учесть их неодинаковость с дидактической точки зрения, обусловленную встречаемостью некоторых фактов в повседневной жизни и возможностью их экспериментального установления на уроке.

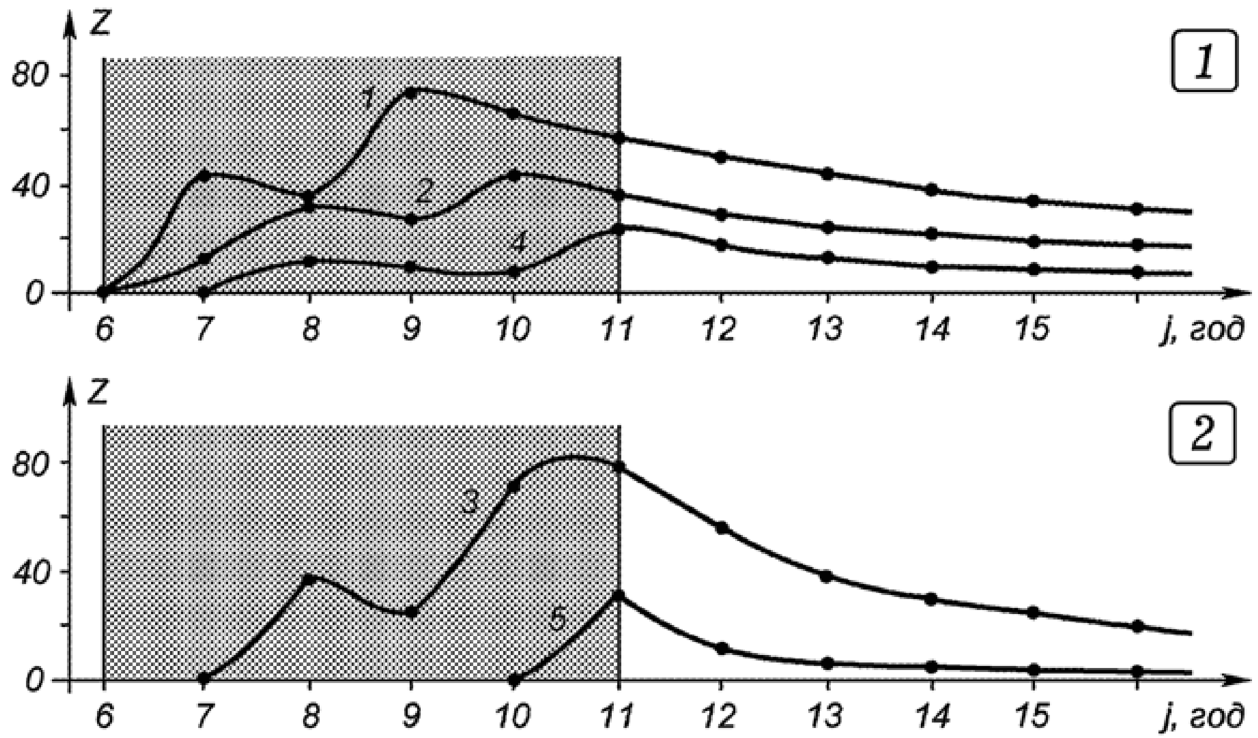


Рис. 5.3. Изменение знаний фактов со временем (по разделам физики)

Анализ полученных кривых, а также результатов тестирования учащихся позволяет сформулировать следующие закономерности формирования у учащихся системы эмпирических знаний [31; 26]:

1. Уровень знаний фактов первой категории, входящих в повседневный опыт учащихся, по мере их изучения возрастает, а после окончания обучения остается практически неизменным.

2. Уровни знаний фактов второй и третьей категорий, не входящих в повседневную деятельность учащихся, после изучения уменьшаются вследствие забывания.

3. Скорость забывания фактов второй и третьей категории тем больше, чем в меньшей степени их изучение опирается на деятельность учащихся, связанную с наблюдением и выполнением учебных опытов, их умозрительным изучением.

5.3. Изучение деятельности учащегося при проведении физического эксперимента

Во второй главе обсуждалось решение учебных задач алгоритмического типа, сводящихся к выполнению последовательности действий по определенному алгоритму. К таким задачам относятся выполнение арифметических действий, решение квадратных уравнений, сборка и разборка технологического узла, поиск выхода из лабиринта, выполнение лабораторной работы по физике и т. д. С целью изучения деятельности учащихся при решении учебной задачи имеет смысл осуществить регистрацию действий учащихся при проведении реального опыта (лабораторной работы) с помощью некоторого записывающего устройства, например, магнитофона.

Как правило, результатом выполнения учащимся или студентом лабораторной работы является письменный отчет, в котором приводятся данные, полученные при измерении и вычислении, экспериментальные и теоретические кривые, вывод по работе. Однако отчет не позволяет проанализировать деятельность человека в процессе выполнения опыта. Была поставлена задача разработать прибор, регистрирующий деятельность ученика при выполнении лабораторной работы «Изучение разряда конденсатора». Выбор именно этой работы обусловлен тем, что в процессе ее выполнения: 1) состояние экспериментальной установки изменяется дискретно, а не непрерывно; 2) испытуемый нажимает на различные переключатели, что можно легко зарегистрировать.

Разработанный и изготовленный нами прибор для регистрации действий учеников при выполнении лабораторной работы [31; 26] состоит из задающего генератора, модулирующего генератора, ключевого элемента и цепи для изучения заряда конденсатора. Цепь для заряда и разряда конденсатора изображена и собрана на лицевой панели прибора, электронная часть размещена внутри непрозрачного корпуса так, что испытуемый не знает о ее существовании. Установка также содержит электронно-цифровой секундомер, позволяющий измерить время заряда, и магнитофон в режиме «Запись», вход которого соединен с электронной частью прибора.

Испытуемый (школьник, студент) должен был изучить инструкцию лабораторной работы, выполнить серию опытов, построить экспериментальный график изменения напряжения на конденсаторе от времени. В процессе выполнения работы он замыкает и размыкает два переключателя, отвечающих за заряд и разряд конденсатора и измерение времени электронным секундомером. При этом изменяется частота задающего генератора и включается модуляция сигнала, идущего на магнитофон. При воспроизведении магнитофонной записи можно установить, сколько времени в том или ином состоянии находилась экспериментальная установка, тем самым восстановить ход действий студента.

На рис. 5.4 представлен один из типичных графиков, показывающий последовательность изменения состояний установки при проведении опыта одним из студентов. Его анализ позволяет выделить следующие два этапа проведения самостоятельного эксперимента: 1) предварительное взаимодействие с экспериментальной установкой, в процессе которого испытуемый приобретает знания, требуемые для выработки алгоритма, и умения, необходимые для его реализации; 2) собственно выполнение эксперимента по найденному алгоритму.

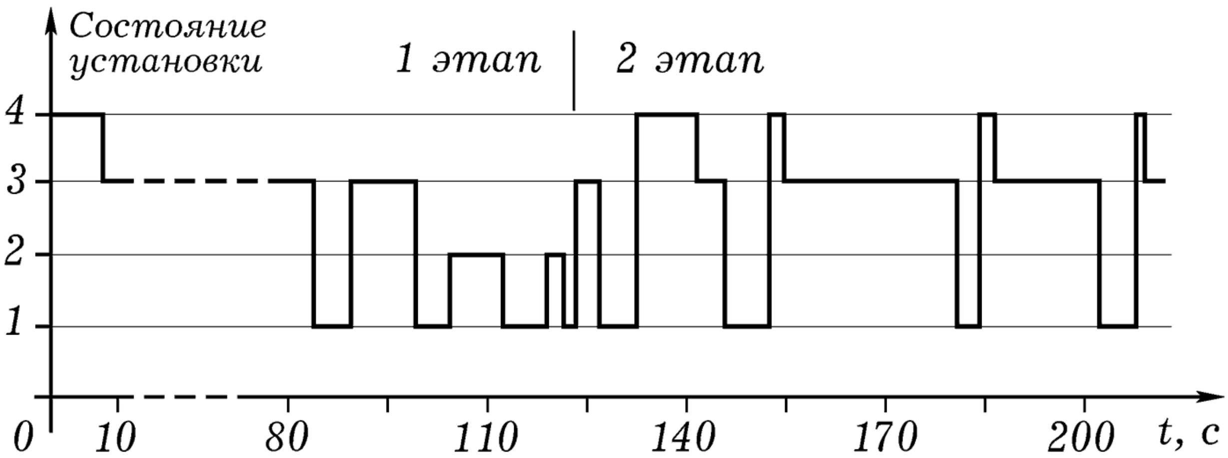


Рис. 5.4. Изменения состояния установки при выполнении опыта

Оба этих этапа хорошо видны на рис. 5.4. Примерно до 120 с длился первый этап, в процессе которого испытуемый выполнял различные операции в неправильной последовательности, наблюдая за их результатами. В конце первого этапа студент понял, как следует проводить эксперимент, дальше его действия приобретают упорядоченный характер 3 – 1 – 4 – 3 – 1 – 4 и т. д. На рисунке представлена только часть графика, так как последующие операции испытуемого не содержат ошибок.

Более глубокий анализ порядка выполнения студентом учебного физического эксперимента показал, что в данной ситуации можно выделить следующие виды операций: 1) простая операция в произвольный момент времени; 2) одновременное выполнение двух или более простых операций в произвольный момент времени; 3) выполнение операции в момент начала или окончания физического процесса; 4) считывание информации с индикатора прибора.

Испытуемые, как правило, начинают проведение опыта с освоения простых действий, состоящих из включения одного из переключателей, запуска секундомера и т. д. Например, студент подключает конденсатор к источнику постоянного напряжения и наблюдает, как стрелка вольтметра плавно отклоняется вправо. По мере освоения простых действий учащиеся переходят к сложным действиям, заключающимся в одновременном выполнении двух простых и т. д.

В результате одного из таких сложных действий установка переходит из состояния 1 в состояние 4. В состоянии 1 происходит заряд конденсатора до напряжения питания, секундомер выключен, затем испытуемый, одновременно переключая два тумблера, переводит прибор в состояние 4, при котором конденсатор начинает разряжаться, запускается электронный секундомер.

Детальное изучение действий студента при проведении физического эксперимента позволило построить следующую блок-схему его деятельности (рис. 5.5). Первым этапом является накопление исходных знаний и умений, требующихся для выполнения опыта. Этот этап включает в себя приобретение информации о рассматриваемом явлении и методах его экспериментального изучения на уроках физики из дополнительной литературы, в результате предшествующих опытов и наблюдений, а также осмысление проблемы, цели и задач планируемого эксперимента.

Перед проведением опыта испытуемый ставит задачу экспериментально исследовать какое-то явление. Будем считать, что при этом он обладает необходимым оборудованием и примерно представляет результат опыта. Возможны две ситуации: 1) учащийся не способен предложить алгоритм проведения опыта, то есть не знает вообще, что следует делать; 2) учащийся предлагает алгоритм проведения опыта, который на данном этапе ему кажется верным.

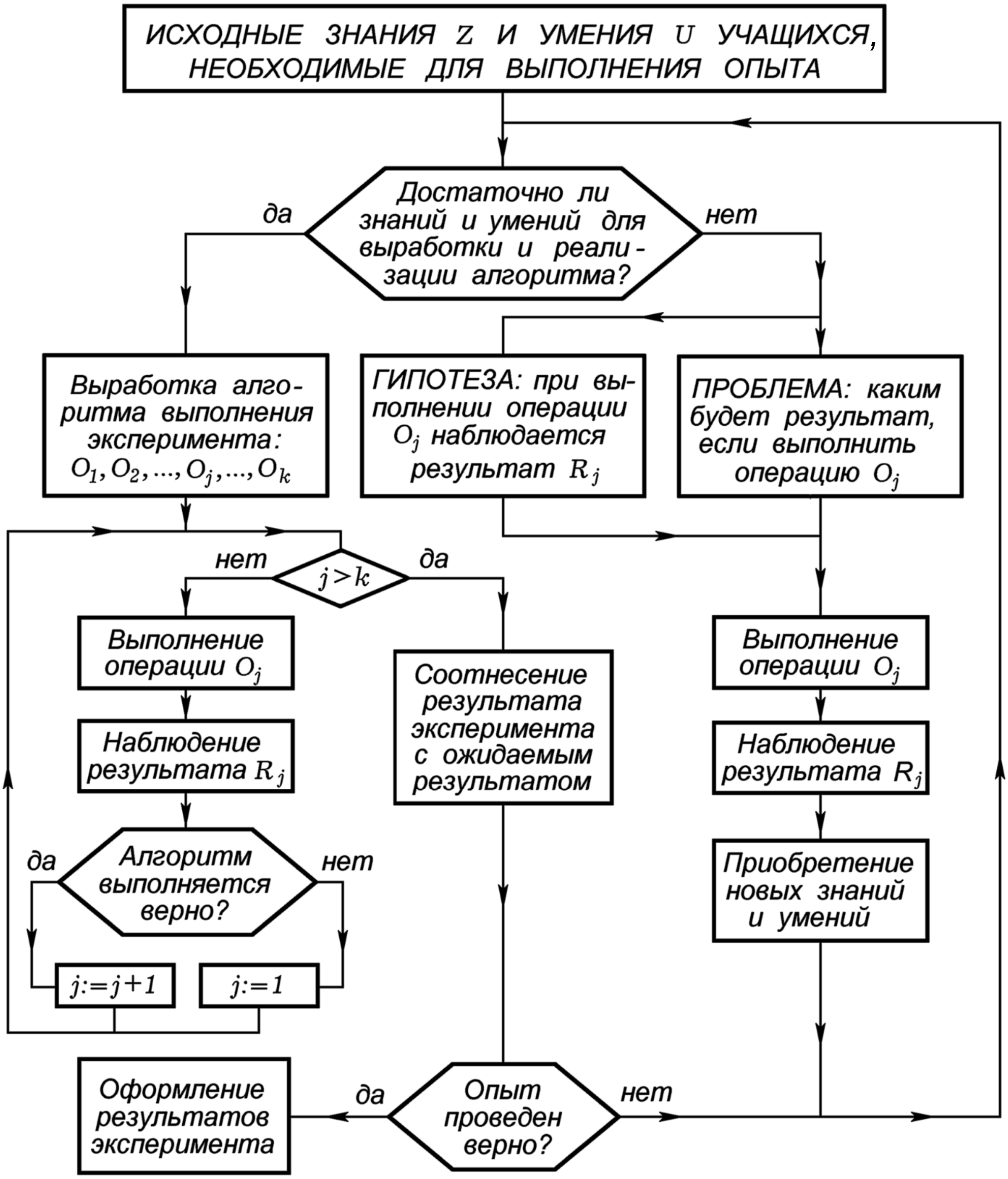


Рис. 5.5. Блок-схема деятельности ученика при проведении опыта

В первом случае школьник начинает выполнять различные операции и, наблюдая получающийся результат, приобретает новые знания и умения работы с установкой. Перед выполнением каждой операции он либо предполагает, что она приведет к какому-то определенному результату, либо не знает, что за этой операцией последует. Если приобретенных знаний и умений по-прежнему недостаточно для выработки и реализации алгоритма, этот цикл повторяется снова. Понятно, что выполнение отдельных операций может и не привести к выработке алгоритма, и учащийся не сможет выполнить опыт. Во втором случае, когда знаний и умений ученика достаточно для выполнения эксперимента, осуществляется выработка алгоритма $O_1 \to O_2 \to \ldots \to O_k$. Затем начинается выполнение опыта, то есть учащийся проводит операцию O_1 и наблюдает результат R_1, сопоставляет его с ожидаемым, контролируя тем самым правильность выполнения эксперимента.

Если операция выполнена верно, то ученик переходит к следующей операции O_2, и все повторяется снова до тех пор, пока номер выполняемой операции j не станет равным k. В случае неправильного выполнения выработанного алгоритма учащийся возвращается к первой операции O_1 и снова пытается выполнить требуемую последовательность действий. По окончании проведения эксперимента, когда выполнены все k операций, производится соотнесение экспериментальных результатов с ожидаемыми. Если цель достигнута, то учащийся оформляет письменный отчет. В случае, когда эксперимент не позволил исследовать явление требуемым образом, испытуемый предпринимает еще одну попытку.

5.4. Решение учебной задачи как поиск выхода из лабиринта

Решение многих математических, физических и других задач сводится к последовательности операций, выполняемых в определенном порядке. Часто заранее неизвестно, какие именно операции требуются, в какой последовательности их следует выполнять. Как правило, испытуемый способен осуществить значительно большее количество операций, чем это необходимо для достижения цели. Решение задачи аналогично поиску выхода из виртуального лабиринта, соединяющего пункт А с пунктом В и представляемого в виде графа. Лабиринт имеет прямой путь, соответствующий правильному решению задачи, а также множество различных ответвлений, заканчивающихся тупиками. Если задача имеет несколько решений, то ей соответствует лабиринт с несколькими различными путями из А в В. Самый короткий путь отвечает оптимальному способу решения задачи.

Всем известны классические опыты с мышью, бегающей по лабиринту и ищущей путь к еде. Во втором и третьем испытании мышь быстрее достигает цели, так как она «запомнила решение задачи». Нечто похожее происходит и с человеком: решая аналогичную задачу второй, третий или четвертый раз, он быстрее «находит выход из виртуального лабиринта».

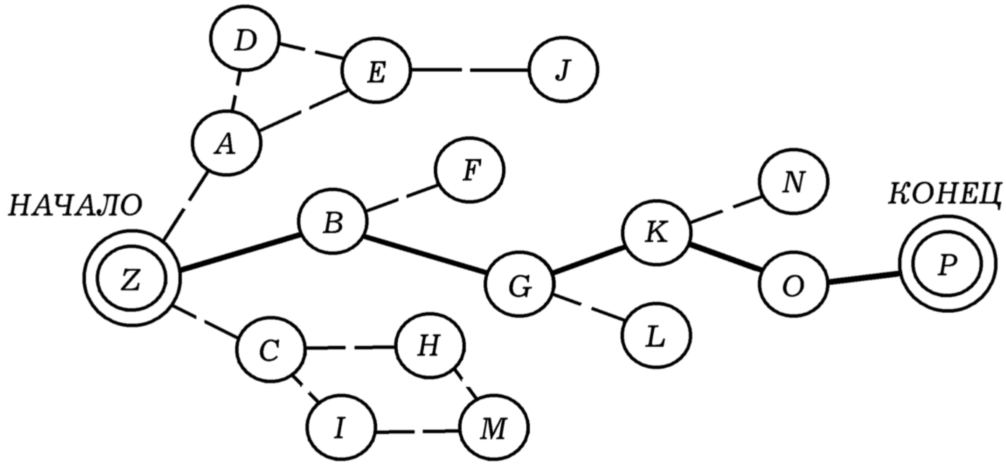


Рис. 5.6. Виртуальный лабиринт, моделируемый компьютерной программой

Автором была создана компьютерная программа, моделирующая лабиринт, изображенный на рис. 5.6. При ее запуске испытуемому сообщается, что он находится в комнате Z (начало), которая имеет три двери с буквами A, B и C. За дверью A находится комната A, за дверью B – комната B, за дверью C – комната C. Испытуемый случайно выбирает одну из дверей и по воображаемому коридору проходит в следующую комнату. Если он открыл дверь C, то он попадает в комнату C, где ему предлагается выбрать двери Z, H или N. После выбора той или иной двери испытуемый попадает в соответствующую комнату, где он делает следующий выбор. Когда человек попадает в комнату P, на экране появляется сообщение о том, что он вышел из лабиринта (дошел до его конца), то есть справился с задачей. Программа ПР–5.1 общается с испытуемым только в текстовом режиме, выбор той или иной двери осуществляется с помощью клавиатуры и регистрируется в текстовом файле data.txt. В этом же файле записывается время, затрачиваемое на каждом шаге (рис. 5.7). Если незначительно изменить программу, то можно сделать так, чтобы испытуемый на любом этапе имел возможность, нажав на соответствующую клавишу, вернуться к началу пути Z и предпринять еще одну попытку выхода из лабиринта.

Эта программа была апробирована с несколькими школьниками. Перед ними была поставлена задача – пройти лабиринт за минимальное время. Результаты расшифровки пути решения этой задачи дали следующее. При первом прохождении лабиринта ученики ведут себя как вероятностные автоматы, случайным образом выбирая ту или иную дверь. Попав в тупик J, они возвращаются обратно в Z, кружат в цикле C–H–M–I, затем случайно находят правильное решение ZBGKOP (рис. 5.6). При повторном прохождении того же самого лабиринта на решение задачи затрачивается существенно меньше времени: испытуемый сразу выбирает комнату B и находит путь к P, ошибаясь гораздо реже. При третьем прохождении лабиринта испытуемый безошибочно выбирает правильный путь ZBGKOP.

```
A  1.2634900136E+01
AD  6.5024999534E+00
ADE  7.4599999292E+00
ADEA  6.9775999414E+00
ADEAZ  6.9919999410E+00
ADEAZC  6.5076999533E+00
ADEAZCI  6.3390999575E+00
ADEAZCIM  6.8378999449E+00
ADEAZCIMH  6.9929999410E+00
ADEAZCIMHC  6.9891999411E+00
ADEAZCIMHCI  6.2592999596E+00
ADEAZCIMHCIM  7.3922999309E+00
ADEAZCIMHCIMH  7.5107999279E+00
ADEAZCIMHCIMHM  7.0250999402E+00
ADEAZCIMHCIMHMC  6.5862999513E+00
ADEAZCIMHCIMHMCI  7.5237999276E+00
ADEAZCIMHCIMHMCIC  6.8854999437E+00
ADEAZCIMHCIMHMCICZ  6.7917999461E+00
ADEAZCIMHCIMHMCICZB  7.0314999400E+00
ADEAZCIMHCIMHMCICZBG  7.0286999401E+00
ADEAZCIMHCIMHMCICZBGK  7.0246999402E+00
ADEAZCIMHCIMHMCICZBGKO  7.0388999399E+00
ADEAZCIMHCIMHMCICZBGKOP  7.0452999397E+00
```

Рис. 5.7. Пример прохождения виртуального лабиринта (файл data.txt)

Затем была поставлена более сложная задача: полностью исследовать лабиринт и на листочке бумаги нарисовать граф, изображающий комнаты, соединенные коридорами (то есть получить рис. 5.6). Как показывает практика, справиться с этим заданием сложнее. Испытуемые делают ошибки, не указывая все соединения (коридоры) между комнатами или рисуя на своем листочке две разные комнаты с одинаковыми буквами.

Возникает вопрос о длине решения задачи и количестве действий, которые следует предпринять, чтобы выйти из лабиринта. Если испытуемому сообщили правильное решение ZBGKOP, то он должен выполнить 5 шагов, то есть $L_{\min} = 5$. Понятно, что длина пути зависит от выбора ученика, который носит случайный характер. Находясь в комнате Z, человек с вероятностью 1/3 выбирает один из трех путей решения задачи (двери A, B или C). Двигаясь по пути A, он с равными вероятностями 1/2 совершает 3 или 4 шага до тупика J, а затем возвращается обратно. Поэтому длина пути от Z до J и обратно $L_1 = 7$. Аналогично, что длина пути от Z до M и обратно $L_3 = 6$. Найдем длину пути от Z до G. От Z до B – 1 шаг, далее человек с вероятностью 1/2 идет по пути B–F–B–G, совершая 3 шага, либо с той же вероятностью сразу по пути B–G, совершая 1 шаг. Получаем, что длина $L_{ZG} = 1 + 3 \cdot 0,5 + 1 \cdot 0,5 = 3$. Рассуждая аналогично, получаем, что $L_{GK} = 2$, $L_{KO} = 2$, $L_{OP} = 1$, поэтому средняя длина пути по ветви B равна: $L_2 = 3 + 2 + 2 + 1 = 8$. В результате для выхода из лабиринта испытуемый, действуя совершенно случайно, должен совершить в среднем $L_1 / 3 + L_2 / 3 + L_3 / 3 = 7/3 + 8/3 + 6/3 = 7$ шагов.

При решении теоретических задач ученик, почувствовав, что движется в неправильном направлении или попал в тупик, в любой момент может вернуться к началу пути и все повторить снова. Обычно школьник понимает, что данная задача требует 2, 3 или 4 действия. Поэтому, если учащийся выполнил несколько достаточно сложных шагов и не пришел к результату, это означает, что он находится на неправильном пути. Школьник снова возвращается к началу и предпринимает повторную попытку решить задачу.

Широкое распространение получили компьютерные игры, в которых пользователь управляет графической моделью человека, ищущего выход из виртуального лабиринта. Существует обучающая программа, при запуске которой на экране появляются изображения комнаты и человека, стоящего перед несколькими дверями с различными буквами. Человек открывает дверь, попадает в другую комнату с дверями и т. д. При этом может оцениваться время прохождения лабиринта. Испытуемый должен не только пройти весь лабиринт, но и нарисовать его схему. Эта программа может быть использована для развития детей младшего и среднего школьного возраста.

5.5. Использование информационных технологий в образовании

Информатизация образования предполагает повышение качества учебного процесса, совершенствование методов обучения, основанных на использовании информационных технологий. Это предусматривает приобщение учащихся к информационной культуре, построение в их сознании информационно-кибернетической картины мира, овладение современными методами обработки информации. Важным аспектом кибернетической педагогики является использование электронных устройств (ПЭВМ) для сообщения новой информации, тестирования учащихся, моделирования изучаемых объектов и ситуаций.

Даже простая компьютерная программа, предлагающая ребенку выполнить несколько заданий, при умелом использовании может привести к положительному результату. Рассмотрим, например, программу, в которой школьнику предлагается решить 10–15 примеров на сложение и вычитание целых чисел, а в конце ставится оценка за правильность выполнения задания. Ее можно использовать один раз в день, причем ребенок может самостоятельно ее запускать и выполнять задания. Функция учителя (родителей) состоит в создании соответствующей мотивации у ребенка к данному виду деятельности, а также к анализу получающихся результатов (оценок). Преимущества применения такого программного средства заключаются в следующем: 1) компьютер в отличие от человека может многократно повторить один и тот же вопрос и спокойно оценить правильность ответа; 2) обучаемый не испытывает отрицательных эмоций в случае ошибки и поэтому не боится ошибиться; 3) ребенок приучается работать с компьютером; 4) выполнение заданий практически не требует никаких временных затрат со стороны родителей или учителя. Вопрос о затратах времени и усилий важен потому, что наибольшее распространение получают методические приемы, требующие мало затрат со стороны учителя и приносящие большой положительный эффект.

В информатике под информационной (компьютерной) технологией понимают технологию переработки информации на ЭВМ, в результате которой получается новый информационный продукт (текстовый, графический, звуковой или видеофайл). Цель использования компьютеров в педагогической деятельности состоит в оказании педагогического воздействия на ученика, связанного с сообщением ему новых знаний, формированием умений, созданием оптимальных условий развития существенных сторон его личности, а также в тестировании, оценке знаний и умений учащихся.

На наш взгляд, **понятие информационной технологии в педагогике** означает технологию обработки числовой, текстовой, графической, аудио- и видеоинформации на электронных устройствах, связанную с сообщением учебного материала, решением учебных задач, созданием компьютерных моделей, выполнением учебных экспериментов, тестированием учащихся, оценкой их знаний и умений. Для информационной поддержки учебного процесса применяются автоматизированные и экспертные обучающие системы, учебные базы знаний, тестирующие программы, электронные книги и энциклопедии, информационно-поисковые системы, мультимедийные системы, создающие эффект виртуальной реальности, образовательные телекоммуникационные сети. Основные направления использования компьютерной техники в образовании представлены в табл. 5.4.

Таблица 5.4

Применения ИТ в образовании

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ
1. Изучение методов обработки информации. 1.1. Создание и обработка текстовых и графических файлов с помощью текстовых и графических редакторов. 1.2. Использование баз данных и динамических таблиц для систематизации информации. 1.3. Обработка видео-, аудио- и фотоматериалов с помощью фото- и видеокамер, видео- и аудиотехники. Создание презентаций, анимации
2. Программирование на компьютере. 2.1. Изучение языков программирования. 2.2. Решение математических, физических, экономических и других задач с помощью математических пакетов. 2.3. Компьютерное моделирование и вычислительный эксперимент
3. Мультимедиатехнологии. 3.1. Получение информации с помощью электронных энциклопедий, словарей, учебников, переводчиков. 3.2. Использование обучающих программ и компьютерных игр для развития учащихся. 3.3. Оценка уровня знаний с помощью тестирующих программ
4. Сетевые технологии. 4.1. Получение информации из энциклопедий и словарей, информационно-поисковых систем Интернет. 4.2. Дистанционное обучение и тестирование в Интранет и Интернет. 4.3. Создание Web-сайта, размещение информации в Интранет и Интернет
5. Эксперименты с компьютером. 5.1. Использование ПЭВМ как измерителя времени, напряжения, частоты сигнала. 5.2. Применение ПЭВМ в качестве источника сигналов заданной формы. 5.3. Использование цифрового осциллографа, спектроанализатора на базе ПЭВМ. Компьютерный измерительный комплекс

Изучение методов обработки информации на ПЭВМ предполагает знакомство учащихся с различными текстовыми и графическими редакторами, с базами данных и динамическими таблицами, а также создание и обработку видео-, аудио- и графических файлов. При изучении информатики учащиеся осваивают методы алгоритмизации и программирования, изучают языки Basic, Pascal, Visual Basic, Delphi и т. д., создают несложные программы.

К **мультимедиа** относят компьютерную технологию одновременной обработки и передачи текстовой, графической, аудио- и видеоинформации, различных анимаций и компьютерных моделей. При этом используются **гипермедиадокументы** – текстовые файлы, содержащие в себе связи с другими текстовыми, графическими, видео- или звуковыми файлами. Внутри гипертекстового документа некоторые фрагменты текста выделены. При их активизации можно перейти на другую часть этого же файла или запустить другой файл на этом или другом ПЭВМ.

В учебном процессе мультимедиатехнологии могут использоваться для обработки графических, видео- и аудиофайлов, для создания различных презентаций, обучающих, развивающих программ, компьютерных энциклопедий и гипермедиа- и телемедиакниг. При этом достигается эффект **виртуальной реальности**: в сознании пользователя создается некоторая модель реального мира, содержащая несуществующие объекты, которыми он оперирует. Преимущество мультимедийных продуктов заключается в одновременном использовании нескольких каналов восприятия, в создании виртуальных моделей реальных ситуаций, явлений и экспериментов, в визуализации абстрактной информации за счет динамического отображения процессов, в установлении ассоциативных связей между различными объектами.

Система виртуальной реальности погружает обучаемого в воображаемую трехмерную модель реального мира. Она обеспечивают «непосредственное» взаимодействие с различными объектами этого мира и манипулирование ими. Это качественно изменяет механизм восприятия и осмысления получаемой информации, способствует формированию чувственно-наглядного образа изучаемого явления. Мультимедийные средства обучения должны соответствовать дидактическим требованиям научности, доступности, проблемности, наглядности, сознательности, систематичности и последовательности обучения.

Современные электронные учебники, учебные энциклопедии и словари представляют собой комплекс программного и педагогического обеспечения, в котором широко используются интерактивный текст, мультимедийные картинки, видеофрагменты, анимации, учебный материал разбит на систему модулей, связанных гиперссылками. Иногда содержатся методические рекомендации и задания для учащихся. Развитие мультимедиатехнологии превратило ПЭВМ в эффективное средство для создания чувственно-наглядных образов изучаемых объектов и явлений, для построения виртуальной модели реального мира.

Развитие компьютерной техники и средств связи обусловило появление и распространение вычислительных сетей. Школы и вузы имеют компьютерные классы и лаборатории, в которых ПЭВМ объединены в локальную сеть, допускающую выход в Интернет. Совокупность ПЭВМ после их объединения в сеть приобретает качественно иные свойства, расширяя возможности пользователя. Использование общих информационных и аппаратных ресурсов позволяет изменить работу учителя и учащихся, применяемую методику.

Всемирная паутина (World Wide Web) позволяет получать доступ к различным каталогам, базам данных, пользоваться электронной доской объявлений, проводить компьютерные конференции, общаться в реальном масштабе времени, то есть читать информацию по мере ее ввода другим пользователем. Доступ обучаемых к информационным ресурсам по Интернет, использование электронной почты для рассылки учебных текстов и контрольных работ, выполнение различных тестовых заданий, размещенных на удаленном сервере, делает возможным **дистанционное образование**.

5.6. Автоматизированные обучающие системы

Обучающая машина (ЭВМ с обучающей программой) необходима не для замены учителя, а для повышения эффективности учебного процесса. Совокупность таких ЭВМ с соответствующим программным обеспечением, объединенных в сеть и подключенных к серверу, образует **систему электронного обучения**. **Автоматизированная обучающая система** (АОС) включает в себя учителя (преподавателя), учеников (студентов), комплекс учебно-методических и дидактических материалов и автоматизированную систему обработки данных. Теперь учитель в большей степени занимается планированием процесса обучения, в то время как ЭВМ может заниматься обучением и тестированием учащихся. Она предъявляет ученику блоки учебного материала, дает контрольные задания, вопросы теста; оценивает правильность ответа, указывает на ошибки, предоставляет возможность их исправить, позволяет реализовать дистанционное обучение и тестирование. Ученик во время работы с машиной вынужден вести себя более активно и самостоятельно, пытаться усвоить требуемый учебный материал, чтобы затем пройти тестирование.

Важным качеством АОС является возможность адаптироваться к способностям конкретного ученика, учитывая его ответы на вопросы теста. **Адаптивная обучающая программа** может изменять способы изложения учебного материала, подстраиваясь под конкретного учащегося, тем самым повысить эффективность обучения. Можно сказать, что такая адаптивная ЭВМ должна уметь самообучаться, то есть изменять свое состояние и поведение в зависимости от предистории взаимодействия с конкретным учащимся. **Использование самообучающейся кибернетической системы, имеющей способность к самоорганизации, позволит приблизиться к решению проблемы оптимизации обучения.**

Функционирование АОС заключается в создании условий усвоения учащимися отдельных блоков (порций) учебного материала [43]. АОС предъявляет порцию теоретической информации, учащийся выполняет упражнения для осмысления и закрепления полученных знаний, АОС при необходимости оказывает помощь при выполнении упражнений, оценивает их правильность [43]. Порядок работы АОС определяется жестким либо адаптивным алгоритмом. Адаптивные АОС, анализируя результаты тестирования, способны гибко приспосабливаться к индивидуальным психофизиологических особенностям учеников и выбирать ту или иную стратегию предъявления учебной информации и учебных заданий.

В общем процесс управления обучением с помощью АОС сводится к следующему [15]: «1) на вход системы обучения подается цель обучения, соответствующая программе обучения; 2) на основании цели обучения устройство управления формирует порцию учебного материала, которая соответствует определенным ЗУН; 3) обучаемый изучает полученную информацию и проходит тестирование знаний в канале обратной связи, на основании которой формируется результат обучения (усвоение входного материала); 4) система сравнивает результат обучения и передает информацию в устройство управления для дальнейшего управления процессом обучения».

Понятно, что возможности АОС не безграничны. В статье [16] анализируются возможности компьютерного управления обучением. Авторы считают, что компьютер без учителя не способен эффективно управлять познавательной деятельностью учащегося, так как для этого необходима оперативная и высококачественная обратная связь. АОС не может оценить правильность слабоформализуемых ответов учащихся, не способна воспитать творческую личность и может быть только помощником учителя. Поэтому необходимо использовать систему обучения, состоящую из компьютера и преподавателя. Дальнейшее развитие АОС требует совершенствования системы контроля знаний учащегося.

5.7. Приложение к главе 5

```
uses dos, crt;                                              {ПР–5.1}
var S,aa: string; t: real; ff: text;                   {Free Pascal}
Label Z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P;
BEGIN clrscr; Assign(FF,'data.txt'); Rewrite(FF);
Z: Writeln('*** NACHALO. PUNKT Z *** ');
Writeln('VIBERI ODNU DVER: A, B, C ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='A' then goto A; If aa='B' then goto B;
If aa='C' then goto C; Writeln('OSHIBKA'); goto Z;
A: Writeln('VIBERI ODNU DVER: D, E, Z ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='Z' then goto Z; If aa='D' then goto D;
If aa='E' then goto E; Writeln('OSHIBKA'); goto A;
B: Writeln('VIBERI ODNU DVER: F, G, Z ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='Z' then goto Z; If aa='F' then goto F;
If aa='G' then goto G; Writeln('NEVERNO'); goto B;
C: Writeln('VIBERI ODNU DVER: H, I, Z ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='Z' then goto Z; If aa='H' then goto H;
If aa='I' then goto I; Writeln('NEVERNO'); goto C;
D: Writeln('VIBERI ODNU DVER: A, E ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='A' then goto A; If aa='E' then goto E;
Writeln('NEVERNO'); goto D;
E: Writeln('VIBERI ODNU DVER: A, D, J');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='A' then goto A; If aa='D' then goto D;
If aa='J' then goto J; Writeln('NEVERNO'); goto E;
F: Writeln('TUPIK. VIBERI DVER: B ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='B' then goto B; Writeln('NEVERNO'); goto F;
G: Writeln('VIBERI ODNU DVER: B, L, K ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='B' then goto B; If aa='L' then goto L;
If aa='K' then goto K; Writeln('NEVERNO'); goto G;
H: Writeln('VIBERI ODNU DVER: C, M ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='C' then goto C; If aa='M' then goto M;
Writeln('NEVERNO'); goto H;
```

```
I: Writeln('VIBERI ODNU DVER: C, M ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='C' then goto C; If aa='M' then goto M;
Writeln('NEVERNO'); goto I;
J: Writeln('TUPIK. DVER E ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='E' then goto E; Writeln('NEVERNO'); goto J;
K: Writeln('VIBERI ODNU DVER: G, N, O ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='G' then goto G; If aa='N' then goto N;
If aa='O' then goto O; Writeln('NEVERNO'); goto K;
L:  Writeln('TUPIK. DVER G ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='G' then goto G; Writeln('NEVERNO'); goto L;
M: Writeln('VIBERI ODNU DVER: H, I ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='H' then goto H; If aa='I' then goto I;
Writeln('NEVERNO'); goto M;
N: Writeln('TUPIK. DVER K ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='K' then goto K; Writeln('NEVERNO'); goto N;
O: Writeln('VIBERI ODNU DVER: K, P ');
t:=0; Repeat t:=t+0.0001; until Keypressed;
aa:=readkey; S:=S+aa; Writeln(FF,S,' ',t); Writeln(aa);
If aa='K' then goto K; If aa='P' then goto P;
Writeln('NEVERNO'); goto O;
P: Writeln('** KONEC PUTI ** ');
Repeat until Keypressed; Close(FF);
END.
```

ЗАКЛЮЧЕНИЕ

Кибернетической педагогикой называется область знаний, находящаяся на стыке педагогики и кибернетики и изучающая функционирование системы обучения и воспитания, закономерности управления учителем учебной деятельностью ученика. К задачам кибернетической педагогики в том числе относятся **информационно-кибернетическое моделирование** различных ситуаций, возникающих в процессе обучения и воспитания, и разработка методов обучения, опирающихся на кибернетические принципы, создание компьютерных программ, имитирующих поведение системы «учитель – ученик».

Целесообразность использования кибернетического подхода при изучении дидактических процессов обусловлена их целенаправленностью и наличием различных цепей управления. В настоящей монографии рассматриваются некоторые аспекты **имитационного моделирования процесса обучения**, предложены компьютерные модели различных дидактических систем, соответствующие реальным ситуациям. При этом решены следующие задачи:

1. Перечислены кибернетические принципы функционирования дидактических систем, предложена кибернетическая модель учебного процесса. Рассмотрены различные подходы к моделированию процесса обучения, в которых поведение ученика имитируется с помощью вероятностного автомата, нейросети, системы, описывающейся дифференциальными уравнениями, и т. д.

2. Проанализирована дискретная модель деятельности ученика, рассмотрена математическая теория обучения вероятностного автомата, осуществлено компьютерное моделирование различных ситуаций, возникающих в процессе обучения. Изучены возможные стратегии взаимодействия учителя и ученика, а также промоделировано решение учеником учебной задачи.

3. Исследованы различные варианты непрерывной модели обучения, проанализированы результаты моделирования различных ситуаций, возникающих в процессе обучения. В частности, рассмотрены: модель самоадаптирующейся системы «учитель – ученик»; модель изучения курса, заканчивающегося экзаменами; модель изучения нескольких тем, имеющих различную сложность; модель работы ученика с учетом изменения его работоспособности в течение дня. Предложена многокомпонентная модель обучения, исходящая из различной скорости забывания «прочных» и «непрочных» знаний, а

также обобщенная модель обучения, учитывающая все перечисленные выше факторы.

4. Изучены возможности использования дискретных и непрерывных имитационных моделей для исследования ситуаций, возникающих в процессе обучения. Рассмотрены различные подходы к проблеме оптимизации процесса обучения, к проблеме изучения вопросов, связанных генетической связью.

5. На примере имитационной модели формирования эмпирических знаний по физике рассмотрен способ согласования параметров модели обучения с результатами тестирования и распределением учебной информации. Проанализирован алгоритм деятельности ученика при выполнении учебного эксперимента и при решении учебной задачи. Рассмотрены возможности использования информационных технологий в учебном процессе.

В монографии рассмотрены дискретные и непрерывные модели, основывающиеся на автоматном подходе и решении дифференциальных уравнений [19–32]. Кроме этого существуют имитационные модели, использующие **мультиагентный подход, сети Петри, генетические алгоритмы, матричное моделирование** [6; 7; 10; 14; 41; 44]. Из анализа этих моделей следует, что информационно-кибернетическое моделирование системы «учитель – ученик» действительно позволяет: 1) исследовать процесс обучения в ситуациях, при которых проводить педагогический эксперимент над реальными людьми невозможно или нецелесообразно; 2) снизить временные и иные затраты на изучение той или иной ситуации; 3) четко и полно контролировать начальное состояние системы, условия протекания дидактических процессов, что невозможно сделать в педагогическом эксперименте; 4) дополнить и обосновать качественные рассуждения, установить закономерности, найти «оптимальный путь» обучения.

Одно из направлений использования имитационных моделей процесса обучения связано с созданием обучающей программы, моделирующей учебный процесс в школе и предназначенной для тренировки студентов педагогических вузов. Она должна допускать изменение параметров учеников, длительности занятий, распределения учебного материала и стратегии поведения учителя. В процессе ее работы пользователь («учитель») изменяет скорость подачи учебной информации, быстро реагирует на вопросы учеников, проводит контрольные работы, ставит оценки, пытаясь добиться наибольшего уровня знаний за заданное время. После окончания обучения на экран выводятся графики, показывающие изменение знаний учеников класса, обучающая программа анализирует работу «учителя» и ставит ему оценку.

СПИСОК ЛИТЕРАТУРЫ

1. *Адамар, Ж.* Исследование психологии процесса изобретения в области математики [Текст] / Ж. Адамар. – М.: Сов. радио, 1970. – 152 с.

2. *Арбиб, М.* Метафорический мозг / М. Арбиб. – М.: Мир, 1976. – 296 с.

3. *Арнольд, В. И.* Теория катастроф / В. И. Арнольд. – М.: Наука; Гл. ред. физ.-мат. лит., 1990. – 128 с.

4. *Аткинсон, Р.* Введение в математическую теорию обучения / Р. Аткинсон, Г. Бауэр, Э. Кротерс. – М.: Мир, 1969. – 486 с.

5. *Вентцель, Е. С.* Исследование операций: задачи, принципы, методология / Е. С. Вентцель. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. – 208 с.

6. *Добрынина, Н. Ф.* Математические модели распространения знаний и управления процессом обучения студентов / Н. Ф. Добрынина // Фундаментальные исследования. – 2009. – № 7.

7. *Доррер, А. Г.* Моделирование интерактивного адаптивного обучающего курса / А. Г. Доррер, Т. Н. Иванилова // Современные проблемы науки и образования. – 2007. – № 5.

8. *Егидес, А. П.* Лабиринты мышления, или Учеными не рождаются [Текст] / А. П. Егидес, Е. М. Егидес. – М.: АСТ-ПРЕСС КНИГА, 2004. – 320 с.

9. *Зыкина, А. В.* Кибернетика: конспект лекций / А. В. Зыкина. – Омск: Изд-во ОмГТУ, 2005. – 28 с.

10. *Ивашкин, Ю. А.* Мультиагентное имитационное моделирование процесса накопления знаний / Ю. А. Ивашкин, Е. А. Назойкин // Программные продукты и системы. – 2011. – № 1. – С. 47–52.

11. *Ительсон, Л. Б.* Математические и кибернетические методы в педагогике / Л. Б. Ительсон. – М., 1964.

12. Кибернетическая педагогика: онтологический инжиниринг в обучении и образовании: монография / К. А. Метешкин, О. И. Морозов, Л. А. Федорченко, Н. Ф. Хайрова. – Харьков: ХНАГХ, 2012. – 207 с.

13. *Колмогоров, А. Н.* Автоматы и жизнь / А. Н. Колмогоров // Кибернетика ожидаемая и кибернетика неожиданная. – М.: Наука, 1968. – С. 12–31.

14. *Кудрявцев, В. Б.* Об автоматном моделировании процесса обучения / В. Б. Кудрявцев, К. Вашик, А. С. Строгалов и др. // Дискретная математика. – 1996. – Вып. 4. – Т. 8. – С. 3–10.

15. *Лаптев, В. В.* Изучение поведения моделей обучения с использованием марковского процесса / В. В. Лаптев, В. И. Сербин // Управление, вычислительная техника и информатика: Вестник АГТУ. – 2010. – № 1. – С 42–45.
16. *Леонтьев, Л. П.* Проблемы управления учебным процессом: математические модели / Л. П. Леонтьев, О. Г. Гохман. – Рига, 1984. – 239 с.
17. *Ломакин, В. В.* Адаптивные системы управления обучением как инструмент повышения эффективности профессионального образования [Электроный ресурс] / В. В. Ломакин, Р. Г. Асадуллаев // http://www.econ.asu.ru/pdf/Programma_ PIEGMU /asadul laev.pdf.
18. *Магазинников, Л. И.* Компьютерное управление обучением: пределы возможности [Электроный ресурс] / Л. И. Магазинников, М. Ю. Шевелев, Ю. П. Шевелев // http:// elibrary.ru/item.asp?id=12831123.
19. *Майер, Р. В.* Автоматы и их обучение / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2011. – № 8. – С. 304–311.
20. *Майер, Р. В.* Задачи, алгоритмы, программы [Электронный ресурс] / Р. В. Майер // http://maier-rv.glazov.net, http://komp-model.narod.ru.
21. *Майер, Р. В.* Имитационное моделирование процесса обучения / Р. В. Майер // Научный электронный архив Академии естествознания // http://econf.rae.ru/ article/6645 (Дата обращения 28.03.2012) (10 с.)
22. *Майер, Р. В.* Имитационная модель процесса обучения / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2012. – № 3. – С. 358–366.
23. *Майер, Р. В.* Имитационное моделирование процесса обучения как метод педагогического исследования / Р. В. Майер // Наука и современность – 2012: сб. материалов XVIII Междунар. науч.-практ. конф. – Новосибирск: Изд-во НГТУ, 2012. – С. 81–85.
24. *Майер, Р. В.* Исследование многокомпонентной модели обучения на ЭВМ / Р. В. Майер // Наука и современность – 2012: сб. материалов XVI Междунар. науч.-практ. конф.: в 2 ч. – Новосибирск: Изд-во НГТУ, 2012. – Ч. 2. – С. 33–38.
25. *Майер, Р. В.* Исследование процесса обучения на ЭВМ: дискретная и непрерывная модели / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2013. – № 3. – С. 254–265.
26. *Майер, Р. В.* Исследование процесса формирования эмпирических знаний по физике / Р. В. Майер. – Глазов: Глазов. гос. пед. ин-т, 1998. – 132 с. // http://rmajer.narod.ru.
27. *Майер, Р. В.* Кибернетическая модель «учитель – ученик» / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2012. – № 10. – С. 333–340.

28. *Майер, Р. В.* Контент-анализ школьного учебника физики / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2013. – № 4. – С. 280–294.
29. *Майер, Р. В.* Многокомпонентная модель обучения / Р. В. Майер // Домашняя лаборатория: интернет-журнал. – 2012. – № 4. – С. 346–351.
30. *Майер, Р. В.* Обобщенная имитационная модель обучения и ее исследование на ПЭВМ / Р. В. Майер // Психология, социология и педагогика. – 2013, апрель [Электронный ресурс] // http://psychology.snauka.ru/2013/04/2054.
31. *Майер, Р. В.* Проблема формирования системы эмпирических знаний по физике: дис. … д-ра пед. наук / Р. В. Майер. – СПб., 1999. – 350 с.
32. *Майер, Р. В.* Психология обучения без огорчения: кн. для начинающего учителя / Р. В. Майер. – Глазов: Глазов. гос. пед. ин-т, 2010. – 116 с.
33. *Мякишев, Г. Я.* Физика: учебник для 10 класса средней школы / Г. Я. Мякишев, Б. Ь. Буховцев. – М.: Просвещение, 1990. – 223 с.
34. *Мякишев, Г. Я.* Физика: учебник для 11 класса средней школы / Г. Я. Мякишев, Б. Б. Буховцев. – М.: Просвещение, 1991. – 254 с.
35. *Новиков, Д. А.* Закономерности итеративного научения / Д. А. Новиков. – М.: Ин-т проблем управления РАН, 1998. – 77 с.
36. *Новиков, Д. А.* Теория управления образовательными системами / Д. А. Новиков. – М.: Народное образование, 2009. – 416 с.
37. *Перышкин, А. В.* Физика: учебник для 7 класса средней школы / А. В. Перышкин. – М.: Просвещение, 1989. – 175 с.
38. *Перышкин, А. В.* Физика: учебник для 8 класса средней школы / А. В. Перышкин, Н. А. Родина. – М.: Просвещение, 1993. – 191 с.
39. *Плотинский, Ю. М.* Модели социальных процессов: учебное пособие для высших учебных заведений / Ю. М. Плотинский. – М.: Логос, 2001. – 296 с.
40. *Поспелов, Д. А.* Вероятностные автоматы / Д. А. Поспелов. – М.: Энергия, 1970. – 88 с.
41. Применение математических методов и ЭВМ в педагогических исследованиях: сб. науч. тр. / под ред. Л. Н. Лексина. – Свердловск: Сверловск. гос. пед. ин-т, 1989. – 181 с.
42. *Разумовский, В. Г.* Физика в школе: научный метод познания и обучения / В. Г. Разумовский, В. В. Майер. – М.: Гуманит. изд. центр ВЛАДОС, 2004. – 463 с.
43. *Растригин, Л. А.* Адаптация сложных систем / Л. А. Растригин. – Рига: Зинатне, 1981. – 375 с.

44. *Редько, В. Г.* Модели адаптивного поведения и проблема происхождения интеллекта / В. Г. Редько // Математическая биология и биоинформатика. – 2007. – Т. 2. – № 1. – С. 160–180.
45. *Робертс, Ф. С.* Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам / Ф. С. Робертс. – М.: Наука, Гл. ред. физ.-мат. лит., 1986. – 496 с.
46. *Розанова, Л. В.* Основы кибернетики: конспект лекций / Л. В. Розанова. – Омск: Изд-во ОмГТУ, 2009. – 60 с.
47. *Советов, Б. Я.* Моделирование систем: учебник для вузов / Б. Я. Советов, С. А. Яковлев. – М.: Высш. шк., 2001. – 343 с.
48. *Соловов, А. В.* Дискретные математические модели в исследования процессов автоматизированного обучения / А. В. Соловов, А. А. Меньшиков. – 2001. – № 4. – С. 205–210 // Educational Technology & Society.
49. *Суходольский, Г. В.* Структурно-алгоритмический анализ и синтез деятельности / Г. В. Суходольский. – Л.: Изд-во Ленинград. ун-та, 1976. – 120 с.
50. *Тарасов, Л. В.* Мир построенный на вероятности: кн. для учащихся / Л. В. Тарасов. – М.: Просвещение, 1984. – 191 с.
51. *Украинцев, Б. С.* Кибернетика и система новых научных принципов / Б. С. Украинцев // Кибернетика и современное научное познание. – М.: Наука, 1976. – С. 7–20.
52. *Фирстов, В. Е.* Математические модели управления дидактическими процессами при обучении математике в средней школе на основе кибернетического подхода: дис. … д-ра пед. наук / В. Е. Фирстов. – СПб., 2011. – 460 с.
53. *Фридман, Л. М.* Психологический справочник учителя / Л. М. Фридман, И. Ю. Кулагина. – М.: Просвещение, 1991. – 288 с.
54. *Харин, Ю. С.* Основы имитационного и статистического моделирования: учебное пособие / Ю. С. Харин, В. И. Малюгин, В. П. Кирлица и др.– Минск: Дизайн ПРО, 1997. – 288 с.
55. *Цетлин, М. Л.* Исследования по теории автоматов и моделированию биологических систем / М. Л. Цетлин. – М.: Наука, 1969. – 316 с.
56. *Шапиро, С. И.* От алгоритмов – к суждениям: эксперименты по обучению элементам математического мышления [Текст] / С. И. Шапиро. – М.: Сов. радио, 1973. – 288 с.
57. *Шеннон, Р.* Имитационное моделирование систем: искусство и наука / Р. Шеннон. – М.: Мир, 1978. – 302 с.

СОДЕРЖАНИЕ

Н А У Ч Н О Е   И З Д А Н И Е

РОБЕРТ ВАЛЕРЬЕВИЧ МАЙЕР

КИБЕРНЕТИЧЕСКАЯ ПЕДАГОГИКА: ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ОБУЧЕНИЯ

Монография

Редактор *Л. В. Ларионова*
Оригинал-макет: *И. С. Леус*
Дизайн титульного экрана: *Р. В. Майер*

Подписано в печать 24.01.2014. Формат 60 x 84 $^{1}/_{8}$. Усл. печ. л. 16,4. Уч.-изд. л. 5,9.
Тираж 11 экз. Заказ № 7381 – 2014.

ФГБОУ ВПО «Глазовский государственный педагогический институт
имени В. Г. Короленко»

427621, Удмуртская Республика, г. Глазов, ул. Первомайская, д. 25
Тел./факс: 8 (34141) 5-60-09, e-mail: izdat@mail.ru