НАЗАД ОГЛАВЛЕНИЕ ВПЕРЁД

Глава 17
ДИДАКТИЧЕСКАЯ СИСТЕМА “УЧЕНИК-УЧИТЕЛЬ”

Одно из направлений развития теории обучения предполагает построение компьютерной модели дидактического процесса, которая ведет себя как система “учитель-ученик”, и проведение серии вычислительных экспериментов при ее различных параметрах. Изменяя параметры ученика и уровень требований учителя, можно исследовать пути развития системы, определить ее состояние в конце обучения. При этом используются как дискретные, так и непрерывные модели; применяются метод статистических испытаний, мультиагентное моделирование и другие методы [1; 3; 7; 9; 11; 13; 14].

17.1. Компьютерное моделирование обучения с помощью дискретной модели ученика

В главе 5 уже обсуждалась дискретная модель обучения, в которой ученик заменялся вероятностным автоматом (п. 5.6). Проанализируем еще одну ситуацию [7, с. 38-40]. Учащийся решает сложную задачу, и при этом он 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, а последовательность $1 \to 2 \to 3 \to ... \to N$ является правильной. Вероятность выбора операции 2 после 1 обозначим p_1, а вероятность выбора $(i+1)$-й операции после i-й – p_i. Решение задачи аналогично поиску выхода из лабиринта. Все операции ученик совершает безошибочно; если же он и ошибается, то 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 ученик (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, ученик переходит ко второй и т. д. На каждый шаг затрачивается одинаковое время Δt. Допустим, ученик 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.

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

Используется программа ПР-1. 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;}**.

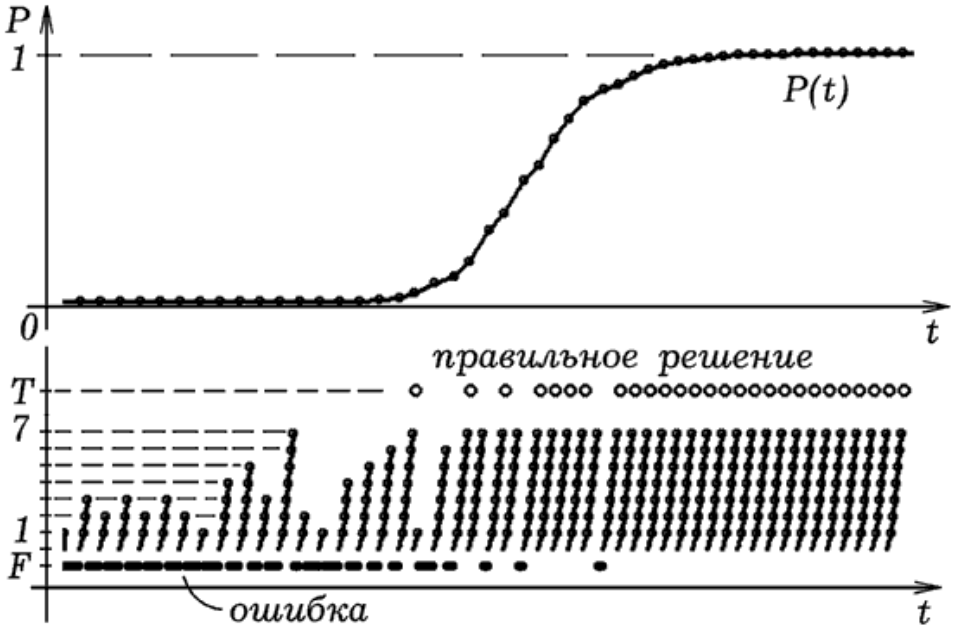

Рис. 17.1. Моделирование решения сложной задачи: стратегия 1

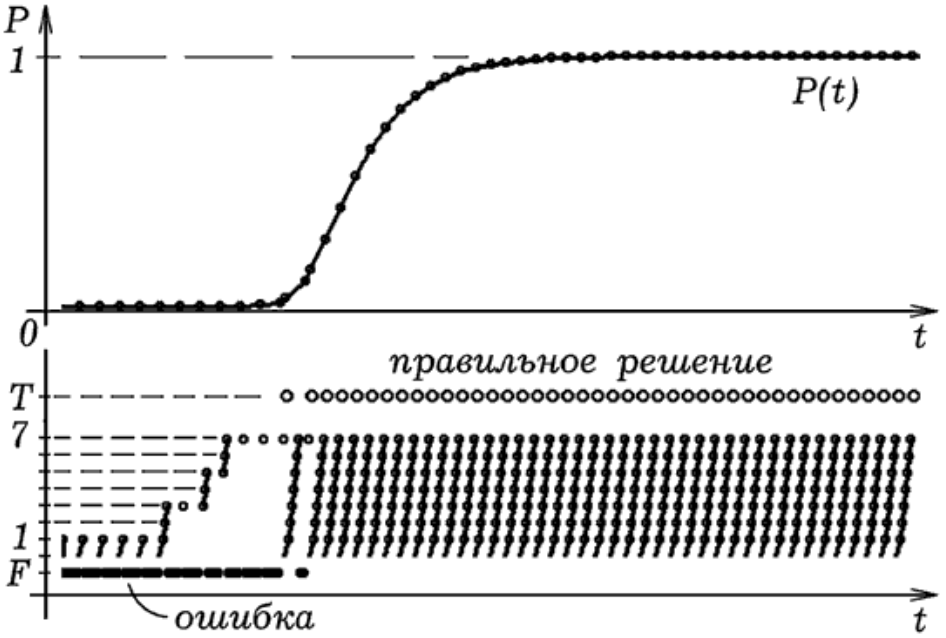

Рис. 17.2. Моделирование решения сложной задачи: стратегия 2

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 17.1); 2) учащийся после совершения ошибки и подсказки учителя продолжает решать данную задачу (рис. 17.2). 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. 17.2) формирование навыка происходит заметно быстрее, чем в первом (рис. 17.1). 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, решая задачу, учится выполнять сразу все операции.

17.2. Построение непрерывной модели процесса обучения

Допустим, имеется ученик, характеризующийся коэффициентами обучения и забывания (α и γ), и учитель, предъявляющий требования $U = U(t)$ [7, с. 52-61]. 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 образует ассоциативные связи и усваивает новые знания. С другой стороны, чем ниже мотивация учащегося, тем меньше усилий F он затрачивает и тем ниже скорость увеличения знаний. Усилия ученика F (мотивация M)

прямо пропорциональны разности между уровнем предъявляемых требований U и уровнем знаний Z: $F = M = k(U - Z)$. Когда $U - Z$ превышает некоторый предел C, ученик перестает прикладывать усилия: $F = 0$. Скорость забывания пропорциональна количеству имеющихся у учащегося знаний: $dZ/dt = -\gamma Z$. В результате получаем, что скорость увеличения знаний выражается уравнением: $dZ/dt = \alpha F Z^b - \gamma Z$, где α и γ – коэффициенты научения и забывания конкретного ученика, а b зависит от степени влияния имеющихся знаний на усвоение новой информации. Скорость увеличения знаний ученика:

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

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, превышающую критическое значение C, то ученик перестает учиться. Для решения диффуравнения $dZ/dt = \alpha Z(U - Z) - \gamma Z$ используется программа ПР-2.

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 $Z_2(t)$.

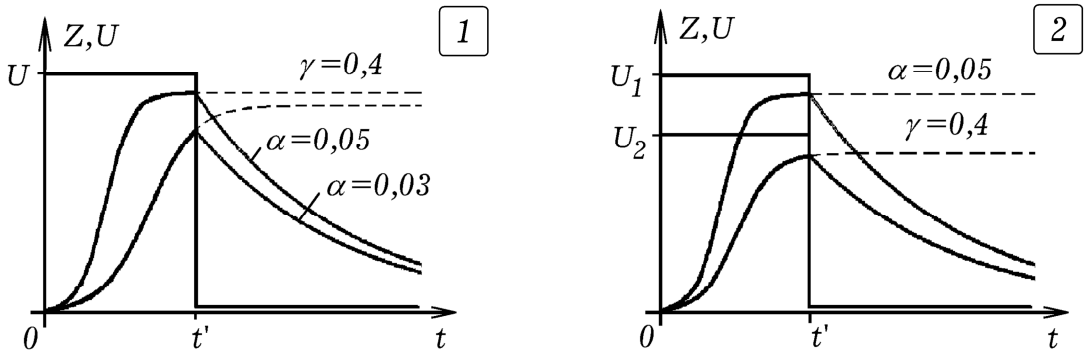

Рис. 17.3. Результаты моделирования обучения

Используется программа ПР-2. 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. 17.3.1. 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.

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 приведены на рис. 17.3.2. 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.

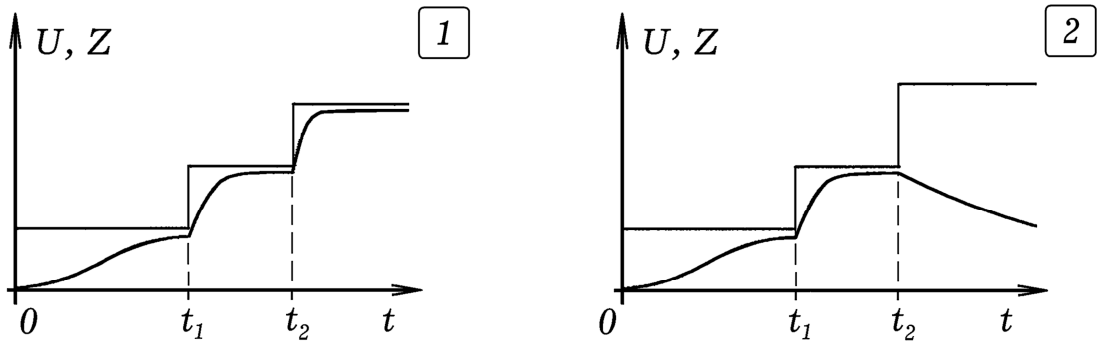

Рис. 17.4. Обучение при скачкообразном повышении U

На рис. 17.4.1 показано, как ведет себя рассмотренная выше модель обучения (программа ПР-3), когда уровень U предъявляемых требований

(количество и сложность заданий) скачкообразно увеличивается. Сначала учащимся предлагают сравнительно простые задания, когда они их освоят, – задания посложнее, затем еще сложнее и т. д. Для того чтобы уровень знаний повышался, необходимо обеспечить не очень большой разрыв между Z и U. Слишком резкое увеличение уровня требований (сложности и объема изучаемого материала) приводит к снижению мотивации и уменьшению уровня знания вследствие забывания (рис. 17.4.2). Если сначала предложить слишком сложные задания (уровень требований U высок), а затем простые, то обучения происходить не будет.

17.3. Учет изменения работоспособности ученика в течение дня

Предположим, что скорость увеличения знаний ученика пропорциональна его коэффициенту научения α, работоспособности r, приложенным усилиям F (мотивации M) и уровню знаний Z в степени b ($b \geq 0$): $dZ/dt = r\alpha FZ^b - \gamma Z$, где r и γ – коэффициент работоспособности, зависящий от степени усталости ученика, и коэффициент забывания. Величина r сначала равна r_0 ($0 < r_0 \leq 1$), а затем, по мере совершения учеником работы P, 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 в два раза от r_0 до $r = r_0/2$. При обучении уровень требований учителя (сообщаемые им знания) больше уровня знаний ученика ($U > Z$), и учебная работа P, совершенная учеником, 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:

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

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 снижение 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 S (трудность понимания учеником) 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 [7, с. 66–68]:

Во время обучения ($U > Z$): $\dfrac{dZ}{dt} = \dfrac{r_0 \alpha (1 - S) F Z^b}{1 + \exp(k_1 (P - P_0))} - \gamma Z\,,$

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

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 – на рис. 17.5 (программа ПР-4). Параметры модели подобраны так, чтобы ее поведе-

ние 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 [7, с. 66–68].

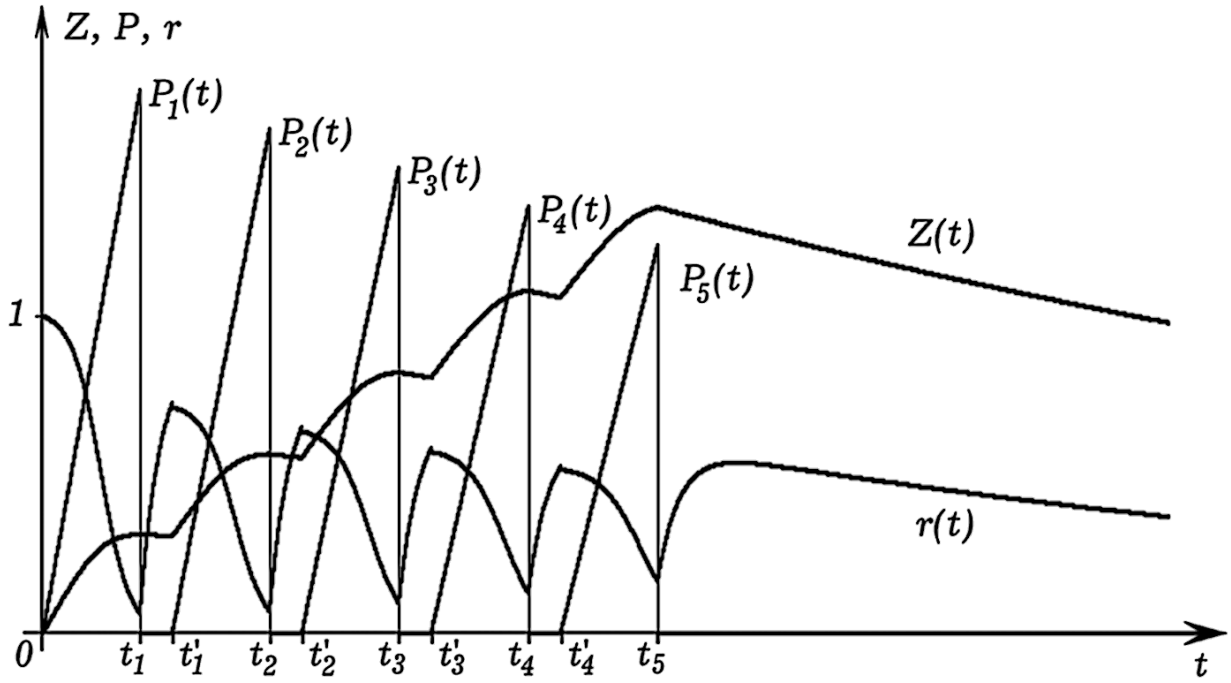

Рис. 17.5. Модель, учитывающая усталость ученика

17.4. Модель обучения с изменяющимся коэффициентом забывания

При моделировании обучения обычно исходят из того, что все элементы учебного материала (ЭУМ) одинаково хорошо усваиваются и одинаково медленно забываются. На самом деле это не так: те ЭУМ, которые включены в деятельность ученика, забываются существенно медленнее, чем ЭУМ, которые были один раз изучены и не используются. Рассмотрим модель, учитывающую уменьшение скорости забывания данного ЭУМ при увеличении числа обращений учащегося [8].

Ситуация 1. Обучение в результате многократного повторения одного ЭУМ на уроке. Рассмотрим ученика, который в процессе обучения вынужден решать последовательность однотипных задач по одной и той же теме. Например, он в течение урока должен в определенные моменты времени складывать числа (или читать отдельные слова, выполнять задания теста). Остальное время на уроке он занимается другой учебной деятельностью, которая нас не интересует. Пусть в момент t_i ученик начинает решать задачу в i-й раз, при этом уровень усвоения соответствующего ЭУМ у него увеличивается до $Z = 1$. Учтем, что время решения задачи τ (или время, затрачиваемое на работу с данным ЭУМ), зависит от того, сколько раз s эта задача решалась ранее. Можно предположить, что с ростом s время τ уменьшается по закону: $\tau = 1 + 1{,}5e^{-s/5}$ условных единиц времени (УЕВ), стремясь к $\tau_\infty = 1$ УЕВ. Выполнив задание и повысив уровень знания соответствующего ЭУМ до 1, ученик переключается на решение другой учебной задачи и начинает забывать усвоенный ЭУМ в соответствии с законом забывания $dZ/dt = -\gamma \cdot Z$. Будем исходить из того, что при увеличении числа s использования данного ЭУМ он запоминается лучше, то есть коэффициент забывания уменьшается, например, по такому закону: $\gamma = 0{,}002e^{-s}$ (УЕВ$^{-1}$). Используется программа ПР-5, моделирующая обучение при обращении к одному и тому же ЭУМ в моменты времени 3, 6, 9, 12, 15, 18 УЕВ.

Результаты моделирования представлены на рис. 17.6. Видно, что после первого и второго обращения к данному ЭУМ приобретенные знания быстро забываются, а после пятого и шестого – забываются очень медленно. То есть в результате многократного использования данного ЭУМ коэффициент забывания γ уменьшается практически до 0, информация прочно запоминается.

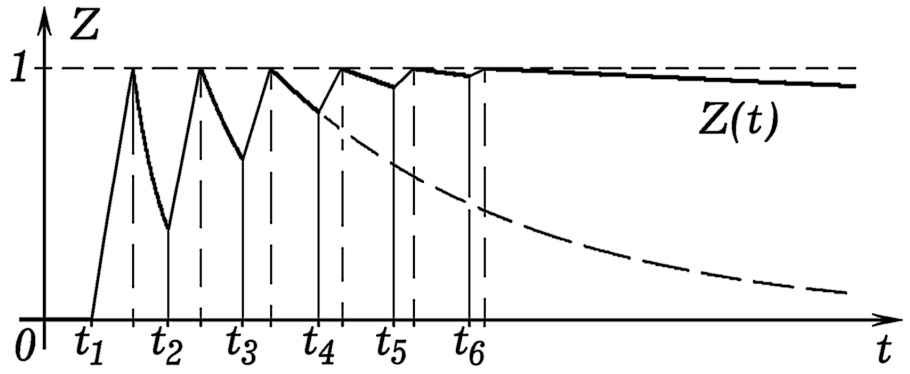

Рис. 17.6. Усвоение одного ЭУМ в результате 6 повторений

Ситуация 2. Обучение в результате многократного повторения множества ЭУМ на одном уроке. Теперь промоделируем изучение N ЭУМ в течение одного урока длительностью T. Например, ученик изучает N новых слов иностранного языка (ЭУМ), которые пронумерованы от 1 до N. Читая текст, он в момент t_1 встречается со словом 2 и в течение времени τ_2 переводит его, в момент t_2 встречается со словом 5 и в течение времени τ_5 переводит его, в момент t_3 – со словом 1 и т. д. Когда ученик переводит i-е слово первый раз ($s_i = 1$), он обращается к словарю и выписывает значение слова, второй раз – смотрит в тетрадь, третий раз – переводит по памяти и т. д., с каждым разом затрачивая меньшее время τ_i. Будем считать, что эти ЭУМ случайным образом встречаются ученику и на работу с i-м ЭУМ он затрачивает время $\tau_i = 1 + 2e^{-s_i/2}$ УЕВ (время решения задачи), где s_i – число обращений. По мере увеличения s_i происходит уменьшение коэффициента забывания i-го ЭУМ по закону $\gamma_i = 0{,}002e^{-s_i/3}$ УЕВ.

Используется программа ПР-6; она строит графики: 1) зависимости суммарных знаний Z от времени; 2) среднего времени решения задачи τ по всем ЭУМ от времени; 3) среднего коэффициента забывания γ по всем ЭУМ от времени. Получающиеся кривые при $N = 10$ и $T = 300$ УЕВ изображены на рис. 17.7. Видно, что во время обучения суммарный уровень знаний в среднем повышается, среднее время решения задачи τ снижается, стремясь к своему пределу τ_∞, средний коэффициент забывания γ уменьшается, стремясь к нулю [8].

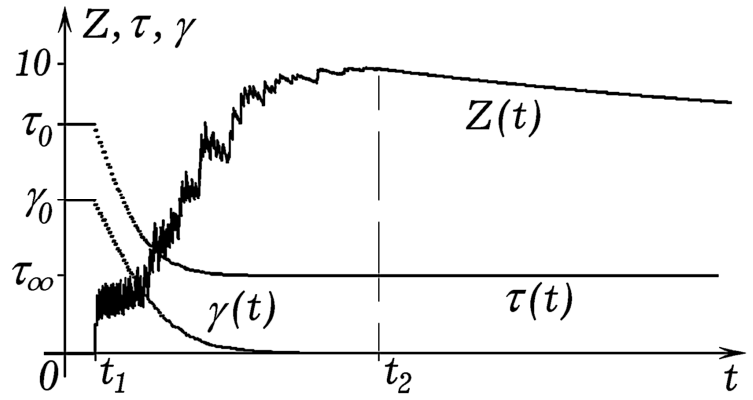

Рис. 17.7. Результаты моделирования при $N = 10$

Результаты моделирования для $N = 6$ и 10 представлены на рис. 17.8. Так как длительность урока одна и та же, то при увеличении количества N изучаемых ЭУМ число обращений к каждому ЭУМ уменьшается, они запоминаются менее прочно, то есть средний коэффициент забывания γ в конце обучения слишком велик. Поэтому при $N = 10$ уровень знаний $Z(t)$ после окончания обучения снижается вследствие забывания, а при $N = 6$ остается практически постоянным.

При еще большем количестве изучаемых ЭУМ кривая $Z(t)$ после окончания обучения быстро убывает. На рис. 17.9 приведен результат моделирования при $N = 20$. Анализируемая модель показывает, что количество N ЭУМ, изучаемых на 1 уроке, не должно быть слишком велико. При больших N знания усваиваются непрочно и потом быстро забываются.

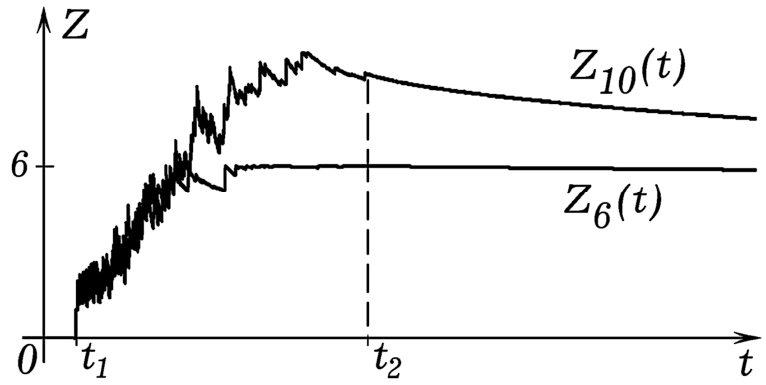

Рис. 17.8. Изменение уровня знаний при $N = 6$ и 10

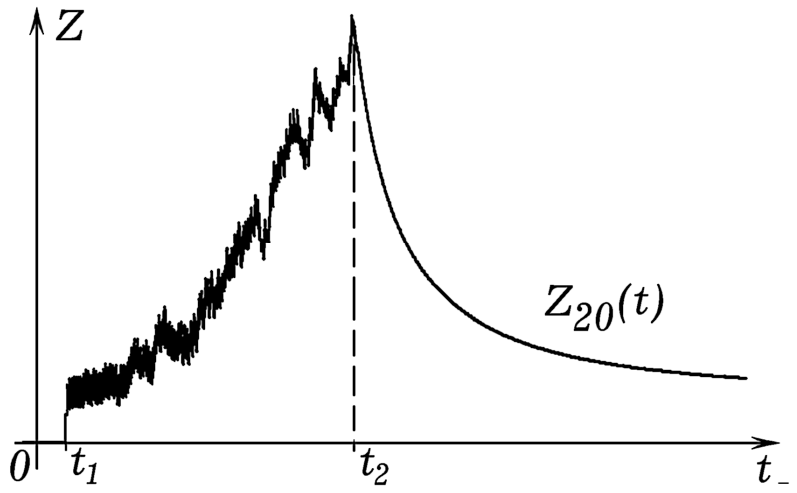

Рис. 17.9. Изменение уровня знаний при $N = 20$

Исследуем зависимость уровня знаний Z ученика через некоторое время t' после окончания урока от числа N изученных ЭУМ (или скорости поступления информации $\upsilon = N / T$). Изменим программу ПР-6 так, чтобы изучаемые ЭУМ следовали по порядку, а не случайным образом, и рассчитаем уровень знаний ученика Z и средний коэффициент забывания γ при различных N. В нашем случае длительность урока составляла $T = 300$ УЕВ, а контрольное время после окончания обучения $t' = 350$ УЕВ. Вычислим показатель эффективности $K = Z / N$, равный отношению количества знаний Z в момент $T + t'$ к общему числу изученных ЭУМ N.

Результаты моделирования приведены на рис. 17.10. С ростом N от 3 до 21 средний коэффициент забывания γ растет от $2 \cdot 10^{-17}$ до $4 \cdot 10^{-5}$, то есть ЭУМ в среднем усваиваются хуже, забываются быстрее. При $N < 12$ количество знаний ученика Z через время t' после окончания обучения с ростом N увеличивается, достигает максимума при $N = 12$, а затем при $N > 12$ уменьшается. Это объясняется влиянием двух факторов: 1) увеличением числа N изученных ЭУМ; 2) уменьшением количества обращений к каждому ЭУМ в течение урока фиксированной длительности T и, как следствие, ухудшением качества усвоения знаний (увеличением коэффициента забывания γ). При небольших N показатель эффективности K равен 1, а с ростом N снижается до нуля. Итак, существует такое N, при

котором уровень знаний Z ученика через время t' после окончания урока длительности T будет максимальным. Чтобы обучение было эффективным, необходимо найти оптимальное значение скорости поступления учебной информации $\upsilon = N / T$.

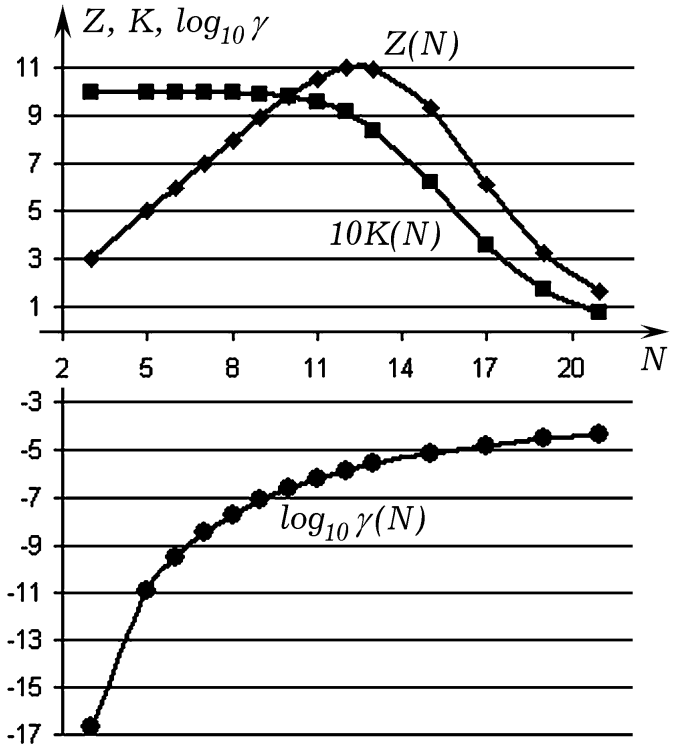

Рис. 17.10. Результаты моделирования обучения на уроке при различных N

Ситуация 3. Обучение в результате многократного повторения множества ЭУМ на нескольких уроках. Теперь промоделируем изучение $N = 30$ ЭУМ в течение трех уроков длительностью $T = 180$ УЕВ, разделенных перерывами продолжительностью $T_n = 220$ УЕВ. Во время урока ученик обращается то к одному, то к другому ЭУМ с равными вероятностями. По мере роста числа s_i обращений к i-му ЭУМ уменьшается затрачиваемое время τ_i и коэффициент забывания γ_i.

Используется программа ПР-7, результаты имитационного моделирования приведены на рис. 17.11. Уроки начинаются в моменты t_1, t_2 и t_3. Видно, что во время обучения номер i рассматриваемого ЭУМ изменяется

случайно от 1 до 30, уровень знаний ученика Z при этом увеличивается. Во время перерывов происходит забывание, Z уменьшается [8].

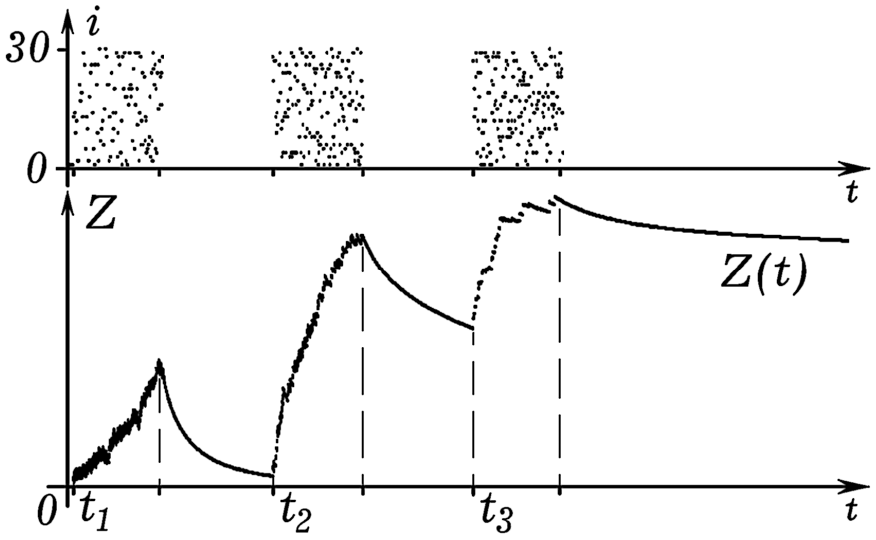

Рис. 17.11. Результаты моделирования изучения 30 ЭУМ на трех уроках

Предложенная компьютерная модель обучения учитывает, что при увеличении числа обращений ученика к данному элементу учебного материала: 1) время его использования уменьшается, стремясь к некоторому пределу; 2) коэффициент забывания уменьшается, стремясь к нулю. При этом показано, что с ростом скорости поступления учебной информации уровень знаний ученика через время t' после окончания обучения сначала увеличивается, достигает максимума, а затем уменьшается.

17.5. Двухкомпонентная модель обучения 1-го типа

С целью повышения точности результатов учтем, что прочность усвоения различных ЭУМ неодинакова, прочные знания забываются существенно медленнее непрочных. Рассмотрим двухкомпонентную модель ученика [7; 9; 10], при этом всю усваиваемую информацию разделим на две категории: 1) знания Зн-1, которые используются ежедневно и поэтому плохо забываются (чтение, письмо, арифметические действия, простые

факты и т. д.); 2) знания Зн-2, которые применяются редко и поэтому быстро забываются (сложные идеи, принципы, факты, теории). Предлагаемая двухкомпонентная модель обучения выражается системой уравнений:

$$dZ_1/dt = k\alpha_1(U_1 - Z_1)Z_1^b - \gamma_1 Z_1,$$

$$dZ_2/dt = k\alpha_2(U_2 - Z_2)Z_2^b - \gamma_2 Z_2,\ Z = Z_1 + Z_2.$$

Здесь U_1 и U_2 – уровни требований учителя, соответствующие Зн-1 и Зн-2, количество которых равно Z_1 и Z_2, а Z – суммарные знания ученика.

Проанализируем следующую ситуацию. Школьник в течение 11 лет учится в школе. Коэффициент усвоения информации по мере обучения увеличивается и задается матрицей $\alpha = (0.01,\ 0.015,\ 0.02,\ 0.025,\ 0.03,\ 0.035,\ 0.04,\ 0.045,\ 0.05,\ 0.055,\ 0.06)$. Уровни требований учителя, соответствующие знаниям Зн-1 и Зн-2, которые нужно усвоить в i-м классе, задаются матрицами: $U_1 = (50,\ 46,\ 42,\ 36,\ 30,\ 25,\ 20,\ 15,\ 10,\ 10,\ 10)$ и $U_2 = (4,\ 8,\ 14,\ 18,\ 24,\ 28,\ 33,\ 38,\ 46,\ 58,\ 62)$. Коэффициенты забывания Зн-1 и Зн-2 $\gamma_1 = 0{,}002$ и $\gamma_2 = 0{,}01$. Необходимо рассчитать суммарный уровень знаний и количество знаний Зн-1 и Зн-2 в различные моменты t.

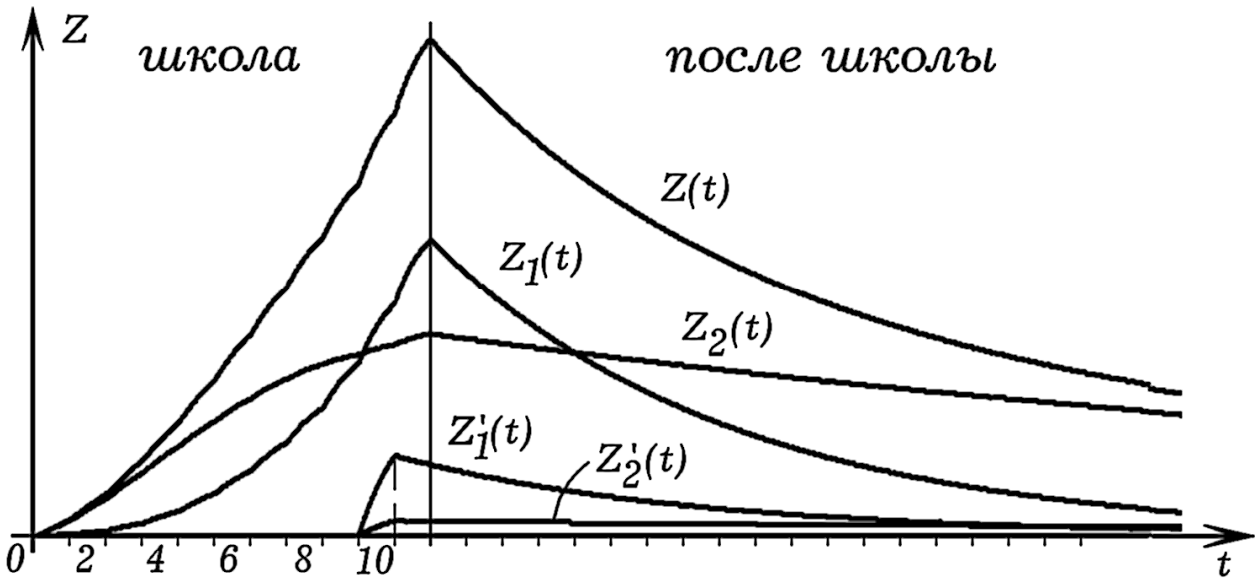

Рис. 17.12. Изменение количества знаний при обучении в школе

Используется программа ПР-8, результаты – на рис. 17.12. На нем представлены: 1) графики $Z_1(t)$ и $Z_2(t)$ зависимостей знаний Зн-1 и Зн-2 от времени; 2) график зависимости общего количества знаний Z от времени; 3) графики $Z_1'(t)$ и $Z_2'(t)$ зависимостей знаний Зн-1 и Зн-2, приобретенных учеником в 10 классе, от времени. Видно, что во время обучения в школе суммарное количество знаний, а также уровни знаний Зн-1 и Зн-2 монотонно возрастают, а после обучения убывают вследствие забывания. Знания Зн-1 забываются существенно быстрее, чем Зн-2. Параметры модели и уровни требований $U_1(t)$ и $U_2(t)$ подобраны так, чтобы модель соответствовала типичной ситуации, встречающейся на практике.

17.6. Многокомпонентная модель обучения 2-го типа

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. Рассмотрим многокомпонентную модель обучения [7, с. 68-70], которая учитывает переход "непрочных" знаний в "прочные":

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

где U – уровень требований, предъявляемый учителем, равный количеству сообщаемых им знаний Z_0, Z – суммарные знания ученика, 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 проис-

ходит 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" усвоения $\mathrm{Pr}=(Z_2/4+$ $+Z_3/2+Z_4)/Z$. 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 двух и четырехкомпонентной модели.

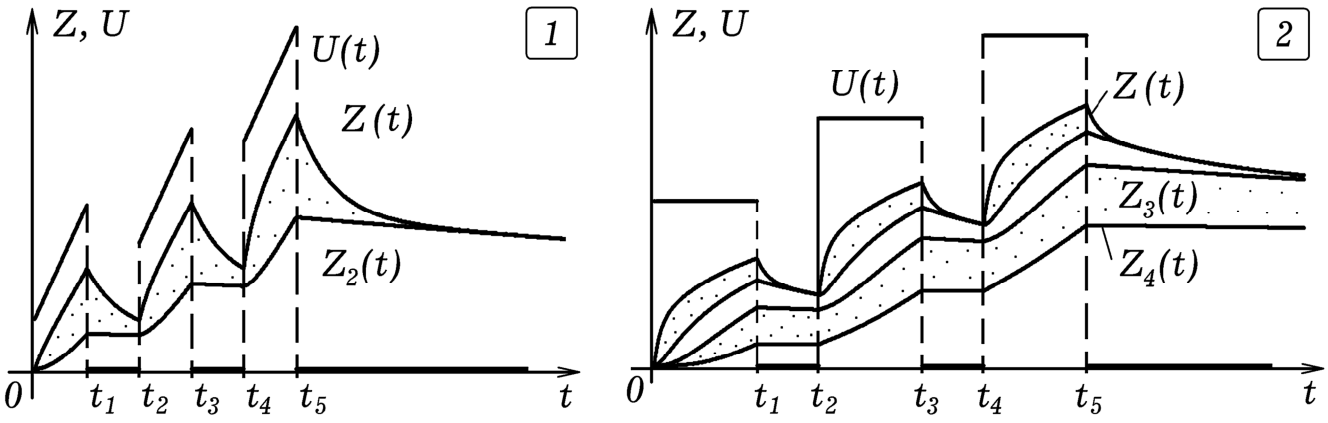

Рис. 17.13. Двух- и четырехкомпонентная модель обучения

Двухкомпонентная модель обучения выражается уравнениями:

$$dZ_1/dt=k\alpha_1(U-Z)-k\alpha_2Z_1-\gamma_1Z_1,$$

$$dZ_2/dt=k\alpha_2Z_1-\gamma_2Z_2,\quad Z=Z_1+Z_2.$$

Используется программа ПР-9; результаты моделирования приведены на рис. 17.13.1. 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 быст-

ро уменьшается, а прочные знания Z_2 забываются существенно медленнее. При использовании четырехкомпонентной модели обучения (программа ПР-10) получаются похожие результаты (рис. 17.13.2). Уровень требований $U(t)$ в течение занятия остается постоянным.

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 его обучает 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 равен θ_i, Z_1 и Z_2 растут. Затем ученик снова пробует решить задачу. Занятия длительностью $T_З >> \Delta t$ чередуются переменами продолжительностью $T_П >> \Delta t$.

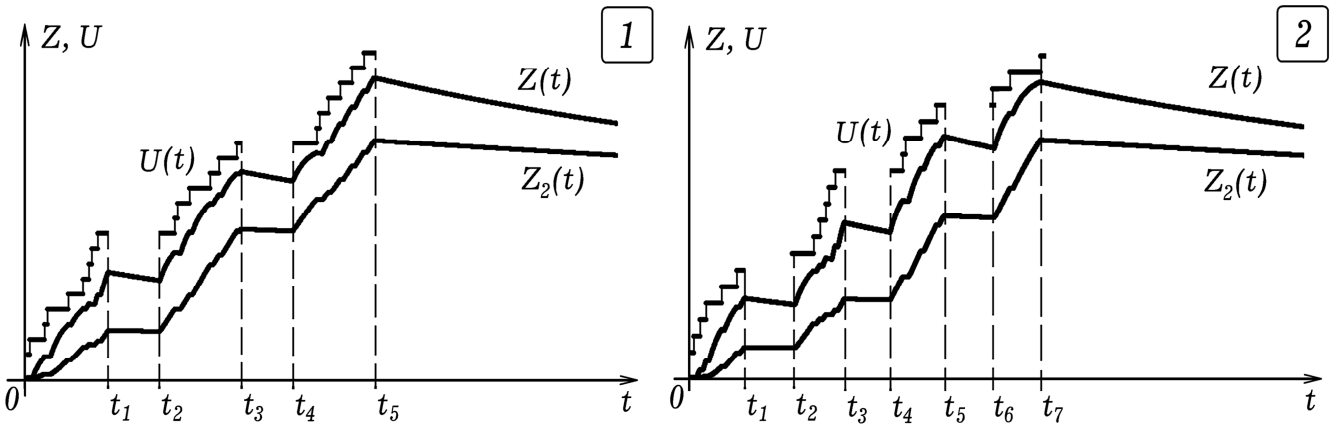

Рис. 17.14. Обучение путем решения задач возрастающей сложности

Для моделирования этой ситуации используется программа ПР-11. 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. 17.14. 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.

17.7. Оптимизация обучения: непрерывная модель

Метод имитационного моделирования позволяет решить **оптимизационную задачу**: 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. 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). Такой режим обучения будем называть **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 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 $P_{\max}$ [7, с. 88–90].

Ситуация 1. В режиме согласованного обучения проводятся три занятия, начинающиеся в фиксированные моменты времени 0, t_2, t_4. Найдем длительность занятий $T_u = t_1 = t_3 - t_2 = t_5 - t_4$, при которой уровень знаний после обучения в момент t' будет равен Z' или выше.

Для решения этой оптимизационной задачи используется программа ПР–12, содержащая цикл, в котором на небольшую случайную величину изменяется длительность урока T_u, затем пересчитывается количество знаний у ученика Z и выясняется, приблизилось Z к требуемому значению Z' или нет. Если да, – изменения T_u принимаются, если нет, – отвергаются, и все повторяется снова. Результаты решения этой задачи – на рис. 17.15.1.

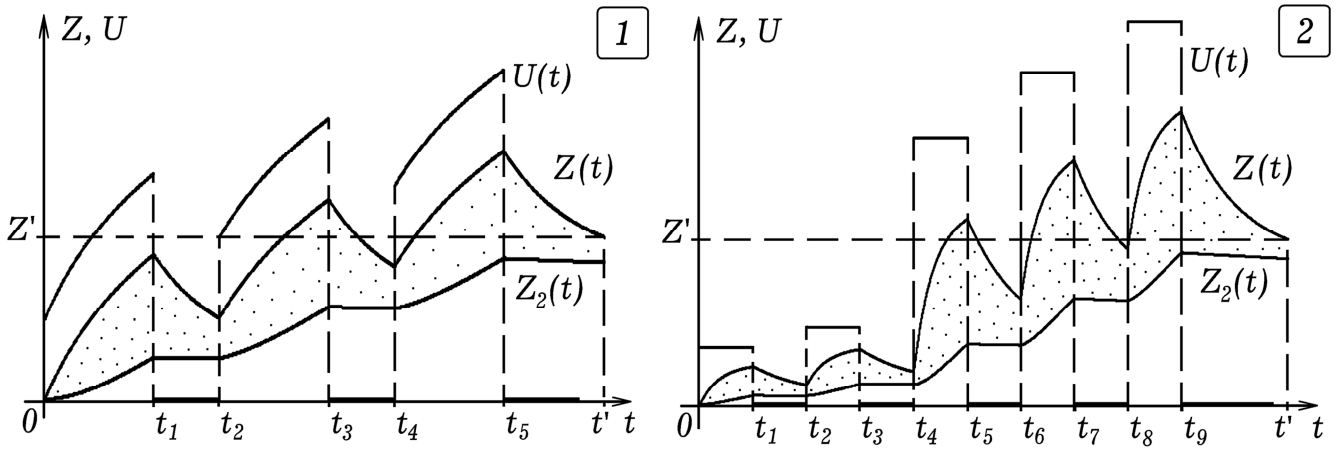

Рис. 17.15. Нахождение оптимальной организации обучения

Ситуация 2. Проводятся пять занятий, начинающиеся в моменты 0, t_2, t_4, t_6, t_8, которые имеют фиксированную длительность $T_u = t_1 = t_3 - t_2 = t_5 - t_4$ и т. д. 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-13. В ней заданы завышенные значения U_i $(i = 1,2,..,5)$ и имеется цикл, с каждым оборотом которого уровни требований U_i изменяются на небольшие случайные величины. Новое значение U_i принимается, если при этом выполняются условия: 1) общая работа P, выполняемая учеником, уменьшается; 2) работа ученика в течение каждого урока не превышает порогового значения $P_{\max}$; 3) его уровень знаний $Z(t')$ в момент t' не опускается ниже Z'. Если эти условия не выполняются, то новое значение U_i отвергается. Результаты моделирования представлены на рис. 17.15.2.

Приложение к главе 17

В приложении представлены тексты программ, которые позволяют промоделировать рассмотренные выше ситуации, возникающие в процессе обучения. Они написаны в средах Borland Pascal 7.0 и Free Pascal 1.0.10.

Программа ПР-1

```
Program Reshenie_zadach_obuchenie;
uses crt, graph;
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.
```

Программа ПР-2

```
Program Odnokomponentn_model;
uses crt, graph;
var DV,MV: integer; U,Z,dt,t,a,g: real;
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.
```

Программа ПР-3

```
Program Skachkoobr_uvelich_U;
uses crt, graph;
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.
```

Программа ПР-4

```
Program Uchet_ustalosti_uchenika;
uses crt, graph;
const  a=0.05; g=0.0002; dt=0.01; Tu=300; Tp=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.
```

Программа ПР-5

```
Program Izuchen_1_EUM;
{$N+}Uses crt, graph;
const N=6; dt=0.001; Mt=20;
t1: array[1..6] of single=(3,6,9,12,15,18);
var t,Z,g: single; i,s,Gd,Gm: integer;
BEGIN
Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi'); t:=-3;
Repeat t:=t+dt;
 For i:=1 to N do If (t>t1[i])and(s=i-1) then
     begin s:=i; Z:=1; t:=t+1.5*exp(-s/5); end;
 g:=0.002*exp(-s/1); Z:=Z-g*Z;
 circle(10+round(Mt*t),400-round(200*Z),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-6

```
Program Izuchen_N_EUM;
{$N+}Uses crt, graph;
const N=13; dt=0.005; Mt=1;
var t,tt,g,t2,SZ,SZ1,ST,Sg: single;
i,j,Gd,Gm: integer; s: array[1..N] of integer;
Z: array[1..N] of single;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
Randomize; t:=-3;
Repeat t:=t+dt; tt:=tt-dt;
  For i:=1 to N do If i<>j then
          Z[i]:=Z[i]-2E-3*exp(-s[i]/3)*Z[i];
  If (tt<=0)and(t>50)and(t<350) then begin
  j:=round(random(N*10)/10)+1;
  If j>N then j:=1; Z[j]:=1; inc(s[j]);
  tt:=1+2*exp(-s[j]/2); t2:=t; end;
  SZ:=0; For i:=1 to N do SZ:=SZ+Z[i];  ST:=0;
```

```
  For i:=1 to N do ST:=ST+(1+2*exp(-s[i]/2))/N; Sg:=0;
  For i:=1 to N do Sg:=Sg+(2E-3*exp(-s[i]/3))/N;
  If t<350 then line(10+round(Mt*t),480-round
     (30* SZ),10+round(Mt*t2),480-round(30*SZ1));
  If t<350 then circle(10+round(Mt*t),485,1);
  circle(10+round(Mt*t),480-round(8E+4*Sg),1);
  circle(10+round(Mt*t),480-round(80*ST),1);
  circle(10+round(Mt*t),480-round(30*SZ),1);
  circle(10+round(Mt*t),480,1); SZ1:=SZ;
  circle(10+round(Mt*t),200-round(100*Z[2]),1);
until (KeyPressed){or(t>700)};
{writeln(t,'  Znaniya ',SZ,' zabivan ',Sg);
readkey;} CloseGraph;
END.
```

Программа ПР-7

```
program Izuch_30_EUM_3_uroka;
{$N+}Uses crt, graph;
const N=30; dt=0.003; Mt=0.5;
var t,g,tt,SZ: single; i,j,Gd,Gm: integer;
s: array[1..N] of integer; Z: array[1..N] of single;
BEGIN Gd:= Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
Randomize; t:=-20; j:=1;
Repeat t:=t+dt; tt:=tt-dt;
  If (tt<=0)and(((t>0)and(t<180))or((t>400)and
     (t<580))or((t>800)and(t<980))) then begin
    j:=round(random*N); If j>N then j:=N;
    If j<1 then j:=1; {inc(j); If j>N then j:=1;}
    Z[j]:=1; inc(s[j]); tt:=1+2*exp(-s[j]/2); end;
  For i:=1 to N do Z[i]:=Z[i]-2E-3*exp(-s[i]/1.5)*Z[i];
  SZ:=0; For i:=1 to N do SZ:=SZ+Z[i];
  circle(20+round(Mt*t),170-round(5*j),1);
  circle(20+round(Mt*t),170,1);
  circle(20+round(Mt*t),500-round(10*SZ),1);
  circle(20+round(Mt*t),500,1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-8

```
Program Obuchen_v_shkole;
uses crt, graph;
const g1=0.002; g2=0.01; dt=0.01; Mt=2; Mz=2;
U1:array[1..11] of inte-
ger=(50,46,42,36,30,25,20,15,10,10,10);
U2:array[1..11] of integer=(4,8,14,18,24,28,33,38,46,58,62);
alfa:array[1..11] of sin-
gle=(1,1.5,2,2.5,3,3.5,4,4.5,5,5.5,6);
var t,U,Z,ZZ1,ZZ2,k: single; DV,MV,i,j: integer;
Z1,Z2: array[1..11] of single;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+dt; U:=0; k:=1;
  If (round(t) mod 12>=9)or(t>12*11-3) then k:=0;
  j:=round(t) div 12 +1;
  ZZ1:=0; For i:=1 to 11 do ZZ1:=ZZ1+Z1[i];
  ZZ2:=0; For i:=1 to 11 do ZZ2:=ZZ2+Z2[i];
  For i:=1 to 11 do begin If j=i then k:=1 else k:=0;
    Z1[i]:=Z1[i]+k*alfa[i]*0.01*(U1[i]-Z1[i])*dt-
    g1*Z1[i]*dt; Z2[i]:=Z2[i]+k*alfa[i]*0.01*(U2[i]-
    Z2[i])*dt-g2*Z2[i]*dt; If Z1[i]<0 then Z1[i]:=0;
    If Z2[i]<0 then Z2[i]:=0; end;
  circle(10+round(Mt*t),450-round(Mz*ZZ1),1);
  circle(10+round(Mt*t),450-round(Mz*ZZ2),1);
  circle(10+round(Mt*t),450-round(Mz*(ZZ2+ZZ1)),2);
  circle(10+round(Mt*t),450-round(Mz*(Z1[10])),1);
  circle(10+round(Mt*t),450-round(Mz*(Z2[10])),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-9

```
Program Dvuhkomponent_model_obucheniya;
uses crt, graph;
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.
```

Программа ПР-10

```
Program Chetirehkomponent_model_obucheniya;
uses crt, graph; 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);}
until KeyPressed; CloseGraph;
END.
```

Программа ПР-11

```
Program Reshenie_zadach_vozrast_slogn;
uses crt,graph; 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.
```

Программа ПР-12

```
Program Optimizaciya1;
uses crt, graph;
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 Dl:=Dl1; 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); end;
  {str(Sum,F); Draw; OutTextXY(20,20,F); delay(5);}
until Keypressed; CloseGraph;
END.
```

Программа ПР-13

```
Program Optimizaciya2;
uses crt, graph;
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.
```

Список литературы

1. Аткинсон Р., Бауэр Г., Кротерс Э. Введение в математическую теорию обучения. М: Мир, 1969. 486 с.
2. Ительсон Л. Б. Математические и кибернетические методы в педагогике. М., 1964. 247 с.
3. Калиткин Н. Н., Карпенко Н. В., Михайлов А. П. и др. Математические модели природы и общества. М.: Физматлит, 2005. 360 с.
4. Кибернетическая педагогика: онтологический инжиниринг в обучении и образовании: монография / К. А. Метешкин, О. И. Морозова, Л. А. Федорченко, Н. Ф. Хайрова. Харьков: ХНАГХ, 2012. 207 с.
5. Леонтьев Л. П., Гохман О. Г. Проблемы управления учебным процессом: Математические модели. Рига, 1984. 239 с.
6. Майер Р. В. Как стать компьютерным гением или книга о информационных системах и технологиях. Глазов: Глазов. гос. пед. ин-т, 2008. 204 с.
7. Майер Р. В. Кибернетическая педагогика: Имитационное моделирование процесса обучения. Глазов: Глазов. гос. пед. ин-т, 2013. 138 с.
8. Майер Р. В. Компьютерная модель обучения с изменяющимся коэффициентом забывания // International Journal of Open Information Technologies. Vol 2. № 1 (2014). P. 12-16. URL: http://injoit.org
9. Майер Р. В. Многокомпонентная модель обучения и ее использование для исследования дидактических систем // Фундаментальные исследования: Педагогические науки. 2013. № 10. С. 2524-2528.

10. Майер Р. В. Основная задача математической теории обучения и ее решение методом имитационного моделирования // Международный журнал прикладных и фундаментальных исследований. 2014. № 2. С. 36-39. URL: www.rae.ru/upfs/?section=content&op=show_article&arti cle_id=5002 (дата обращения: 27.02.2014).
11. Новиков Д. А. Закономерности итеративного научения. М.: Институт проблем управления РАН, 1998. 77 с.
12. Плотинский Ю. М. Модели социальных процессов: учеб. пособие для высш. учеб. заведений. М.: Логос, 2001. 296 с.
13. Робертс Ф. С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам. М.: Наука, Гл. ред. физ.-мат. лит., 1986. 496 с.
14. Фирстов В. Е. Математические модели управления дидактическими процессами при обучении математике в средней школе на основе кибернетического подхода: дис. … д-ра пед. наук. СПб., 2011. 460 с.
15. Шеннон Р. Имитационное моделирование систем - искусство и наука / пер. с англ. М.: Мир, 1978. 302 с.

ВВЕРХ