

Оптимизация, системный анализ и исследование операций

© 2025 г. С.В. ИВАНОВ, д-р физ.-мат. наук (sergeyivanov89@mail.ru),
А.Е. СТЕПАНОВ (rus.fta@yandex.ru)
(Московский авиационный институт
(национальный исследовательский университет))

ОПТИМИЗАЦИЯ ПО ВЕРОЯТНОСТНОМУ КРИТЕРИЮ В ДИНАМИЧЕСКОЙ МОДЕЛИ ПРОХОЖДЕНИЯ ТЕСТА

Рассматривается динамическая модель прохождения ограниченного по времени теста. Ставятся задачи поиска программной и позиционной стратегий, максимизирующих вероятность прохождения теста. Стратегией является выполнение или невыполнение текущего тестового задания. Позиционная стратегия определяется как функция времени, прошедшего с начала тестирования, и набранных за предыдущие задания баллов. Для синтеза позиционной стратегии используется метод динамического программирования Беллмана. Для поиска оптимальной программной стратегии предлагается алгоритм, основанный на методе ветвей и границ. Приводятся результаты вычислений и производится их сравнение с результатами решения аналогичной задачи, в которой тест не считается пройденным, когда тестируемый не уложился в лимит времени.

Ключевые слова: модель тестирования, вероятностный критерий, синтез управления.

DOI: 10.31857/S0005231025060073, **EDN:** IKHNWB

1. Введение

Современный уровень применения информационных технологий в образовании обосновывает актуальность развития в настоящее время, в частности, теории адаптивного тестирования (САТ-теории), основы которой были заложены в конце прошлого столетия, например, в работах Дж. Раша [1]. Хороший обзор современных результатов в этой области предложен в [2]. Концепция рассматриваемой теории подразумевает использование информации о результатах обучения тестируемого и прохождения им контрольного мероприятия в процессе формирования содержания теста и принятия решения о его завершении. Обработка этой информации и принятие решения часто происходит с применением современных методов машинного обучения и искусственного интеллекта. Эти же технологии используются также при

формировании индивидуальных траекторий обучения в системах дистанционного обучения (СДО), например [3, 4]. К современным работам в этой области можно отнести [5–7]. Подобные технологии используются обычно организаторами тестирования и разработчиками программного обеспечения СДО. Однако на проблему тестирования можно посмотреть и со стороны тестируемого, которому необходимо выработать собственную стратегию прохождения, как правило, ограниченного по времени теста. Время прохождения теста напрямую связано со временем ответа тестируемого на задания теста. Вероятностным моделям времени ответа тестируемого или пользователя СДО на задание посвящено значительное количество работ, например [8–10]. Задача поиска оптимальной стратегии тестируемого по критерию максимизации вероятности набора тестируемым определенного количества баллов, при одновременном выполнении ограничения на время выполнения теста, рассмотрена в [11]. Похожая задача в квантильной постановке, когда максимизируется число набранных за тест баллов при выполнении вероятностного ограничения на время выполнения теста, рассмотрена в [12]. В последних двух работах решаются задачи стохастического программирования, в которых тестируемый априорно, перед прохождением теста, выбирает в качестве собственной стратегии набор заданий, которые он будет решать в первую очередь. Другие задания теста решаются тестируемым в случае, если у него остается время до окончания теста после завершения работы с первой группой заданий. В рассмотренных постановках не принимается во внимание последовательность решения тестируемым заданий теста. Это может быть естественным в случае, если тестируемый не получает в процессе прохождения теста информацию о правильности решения очередного выполненного задания, а следовательно, и о текущем количестве набранных баллов. Однако очень часто, особенно при тестировании в СДО, например в [4], такая информация доступна пользователю и может быть использована им в процессе прохождения теста. Это делает актуальным рассмотрение динамических моделей тестирования с корректировкой стратегии прохождения теста после каждого рассмотренного тестируемым задания.

В данной работе рассматривается динамическая модель прохождения фиксированного (не изменяемого в процессе прохождения) теста по критерию максимизации вероятности набора тестируемым не менее заданного числа баллов в условиях наличия ограничения на время выполнения теста. Решаются задачи поиска программной и позиционной стратегий тестируемого. Предлагаются алгоритмы решения рассмотренных задач, построенные с использованием методов стохастического и динамического программирования. Проводится сравнительный анализ с решениями, полученными в [11, 12], на тех же исходных данных.

2. Описание модели и задачи синтеза управлений

Введем в рассмотрение динамическую систему, описывающую прохождение тестирования студентом. На k -м шаге студенту предлагается выполнить задание, за успешное решение которого начисляется b_k баллов, $k = \overline{1, n}$. Для успешного прохождения теста требуется набрать не менее φ баллов. Время выполнения теста ограничено величиной $\bar{T}$. Время прохождения k -го задания описывается случайной величиной τ_k . Правильность выполнения k -го задания описывается случайной величиной X_k , которая равна 1, если k -е задание выполнено верно, и равна 0 в противном случае. Стратегия определяется набором переменных $u = (u_1, \dots, u_n)$, в котором $u_k = 1$, если студент пытается решать k -е задание, в противном случае $u_k = 0$. Введем переменные состояния динамической системы. Пусть время, затраченное на выполнение первых k заданий, равно T_k , а балл, набранный за это время, равен S_k . Тогда динамика системы описывается уравнениями

$$\begin{aligned} (1) \quad & T_k = T_{k-1} + \tau_k u_k, \\ (2) \quad & S_k = S_{k-1} + b_k X_k I\{T_k \leq \bar{T}\} u_k, \\ (3) \quad & T_0 = 0, S_0 = 0, \quad k = \overline{1, n}, \end{aligned}$$

где $I\{\cdot\}$ – индикатор события, стоящего в скобках, равный 1, если условие выполнено, и равный 0 в противном случае. Таким образом, баллы b_k начисляются, если студент затратил на выполнение первых k заданий не более $\bar{T}$ единиц времени. Будем считать, что все τ_k – дискретные случайные величины с конечным множеством реализаций $\mathcal{T}_k$. В дальнейшем будем использовать обозначение $p_k(t, x) = \mathbf{P}\{\tau_k = t, X_k = x\}$. В настоящей работе предполагается, что для всех $k = \overline{2, n}$ сигма-алгебра, порожденная случайными величинами X_k и τ_k , не зависит от сигма-алгебры, порожденной случайными величинами $X_1, \dots, X_{k-1}, \tau_1, \dots, \tau_{k-1}$. Подобное предположение означает, что ответ на текущие вопросы теста не влияет на дальнейшее прохождение тестирования. Влияние знания о правильности решения предыдущих задач на правильность решения будущих задач требует дополнительного статистического анализа. Пусть $S(u)$ – случайная сумма баллов, равная величине S_n , при выбранном управлении u .

Поставим задачу поиска максимума функции вероятности

$$(4) \quad P_{\varphi, \bar{T}}^1(u) = \mathbf{P}\{S(u) \geq \varphi\} \rightarrow \max_{u \in \{0,1\}^n}.$$

Данная задача направлена на синтез программной стратегии выбора заданий, решение которых позволит студенту максимизировать вероятность прохождения теста.

Теперь будем считать, что управление выбирается в классе позиционных стратегий. Это значит, что на каждом шаге при выборе своей стратегии студент принимает во внимание затраченное к текущему моменту время и общий

балл, набранный за предыдущие задания. Через $\mathbf{u}_k(T_{k-1}, S_{k-1})$ обозначим стратегию студента на k -м шаге как значение некоторой функции $\mathbf{u}_k$ переменных состояния системы, равное 1, если студент пытается решить k -е задание, и равное 0 в противном случае. Пусть $\mathbf{u}$ – вектор, составленный из функций $\mathbf{u}_1, \dots, \mathbf{u}_n$ и описывающий позиционную стратегию. Обозначим через $\mathbf{S}(\mathbf{u})$ случайную сумму баллов S_n , определяемую уравнениями (1) и (2) при подстановке в них $u_k = \mathbf{u}_k(T_k, S_k)$. Здесь используется жирный шрифт, чтобы отличать позиционную стратегию $\mathbf{u}$, являющуюся функцией, от программной стратегии u , описываемой набором переменных, а также чтобы отличить сумму баллов $\mathbf{S}(\mathbf{u})$, получаемую в результате применения позиционной стратегии, от суммы баллов S_n , полученных при программной стратегии.

Поставим задачу поиска максимума функции вероятности при применении позиционной стратегии:

$$(5) \quad P_{\varphi, \bar{T}}^2(\mathbf{u}) = \mathbf{P}\{\mathbf{S}(\mathbf{u}) \geq \varphi\} \rightarrow \max_{\mathbf{u} \in \mathcal{U}},$$

$$(6) \quad \mathbf{u}^* = (\mathbf{u}_1^*, \dots, \mathbf{u}_n^*) \in \text{Arg max}_{\mathbf{u} \in \mathcal{U}} P_{\varphi, \bar{T}}^2(\mathbf{u}),$$

где $\mathcal{U}$ – множество допустимых функций, описывающих управление. Поскольку случайное время на ответ для любой задачи является дискретной случайной величиной с конечным числом значений, а факт правильного ответа на задачу моделируется распределением Бернулли, то количество состояний (T_k, S_k) на k -м шаге является конечным. Однако для удобства оптимизации управление на k -м шаге $\mathbf{u}_k$ будем выбирать как функцию на более широком множестве состояний $\mathbb{T}_k \times \mathbb{S}_k$, где $\mathbb{T}_k$ и $\mathbb{S}_k$ – некоторые конечные множества, содержащие в себе все возможные значения состояний T_k и S_k соответственно. В связи с этим будем считать, что $\mathcal{U}$ является множеством всех функций из $\bigotimes_{k=1}^n \mathbb{T}_k \times \mathbb{S}_k$ во множество $\{0, 1\}^n$.

Замечание 1. В [11] решается задача максимизации вероятности получения требуемого балла и одновременного выполнения ограничения на время тестирования с использованием программных стратегий, которая в обозначениях настоящей статьи записывается в виде

$$(7) \quad \tilde{P}_{\varphi, \bar{T}}^1(u) = \mathbf{P}\{S(u) \geq \varphi, T(u) \leq \bar{T}\} \rightarrow \max_{u \in \{0, 1\}^n},$$

где через $T(u)$ обозначено случайное время прохождения теста при использовании программной стратегии u . Таким образом, постановка (7) предполагает, что тест не считается пройденным, когда тестируемый не уложился в лимит времени, даже если в результате теста набран требуемый балл. В задаче (4) отсутствует требование уложиться в заданный лимит времени, но за решение задач после момента времени $\bar{T}$ баллы не начисляются. Вместе с тем, достоинством постановки (7) является независимость оптимального выбора задач от их порядка в тесте. В задачах (4) и (5) оптимальный набор

решаемых задач зависит от их порядка. Полезно отметить, что оптимальное значение критериальной функции в задаче (7) является нижней оценкой оптимальных значений критериальных функций задач (4) и (5) при любом порядке задач в тесте.

Замечание 2. Постановку (7) можно обобщить и на случай позиционных стратегий. В результате получится задача

$$(8) \quad \tilde{P}_{\varphi, \bar{T}}^2(\mathbf{u}) = \mathbf{P}\{\mathbf{S}(\mathbf{u}) \geq \varphi, \mathbf{T}(\mathbf{u}) \leq \bar{T}\} \rightarrow \max_{\mathbf{u} \in \mathcal{U}},$$

где через $\mathbf{T}(\mathbf{u})$ обозначено случайное время прохождения теста T_n , определяемое уравнениями (1) при подстановке в них $u_k = \mathbf{u}_k(T_k, S_k)$. Ниже будет показано, что постановка (5) оказывается полностью эквивалентной постановке (8) в отличие от пары постановок (4) и (7).

3. Синтез позиционного управления

Для решения задачи синтеза позиционного управления (5) применим метод динамического программирования Р. Беллмана. Определим функцию Беллмана для $k = \overline{1, n}$ по правилу:

$$(9) \quad B_k(T_{k-1}, S_{k-1}) = \max_{\mathbf{u}_k, \dots, \mathbf{u}_n} \mathbf{P}\{\mathbf{S}(\mathbf{u}) \geq \varphi \mid T_{k-1}, S_{k-1}\}.$$

При этом на последнем шаге функцию Беллмана определим равенством

$$(10) \quad B_{n+1}(T_n, S_n) = I\{S_n \geq \varphi\}.$$

Отметим, что зависимости от T_n на последнем шаге нет. Она введена для удобства записи соотношений динамического программирования. Как нетрудно заметить,

$$\max_{\mathbf{u} \in \mathcal{U}} P_{\varphi, \bar{T}}^2(\mathbf{u}) = B_1(T_0, S_0).$$

Получим соотношения динамического программирования, используя формулу полной вероятности:

$$\begin{aligned} (11) \quad B_k(T_{k-1}, S_{k-1}) &= \max_{u_k = \mathbf{u}_k(T_{k-1}, S_{k-1})} \max_{\mathbf{u}_{k+1}, \dots, \mathbf{u}_n} \sum_{t \in \mathcal{T}_k, x \in \{0,1\}} \mathbf{P}\{\mathbf{S}(\mathbf{u}) \geq \\ &\geq \varphi \mid S_k = S_{k-1} + b_k x I\{T_{k-1} + tu_k \leq \bar{T}\} u_k, T_k = T_{k-1} + tu_k\} p_k(t, x) = \\ &= \max_{u_k \in \{0,1\}} \sum_{t \in \mathcal{T}_k, x \in \{0,1\}} B_{k+1}(T_{k-1} + tu_k, S_{k-1} + b_k x I\{T_{k-1} + tu_k \leq \bar{T}\} u_k) p_k(t, x). \end{aligned}$$

Управление $\mathbf{u}_k^*(T_{k-1}, S_{k-1})$ определим из условия максимума полученной функции Беллмана на k -м шаге.

Поскольку распределения случайных величин являются дискретными, все получаемые функции в методе динамического программирования являются

измеримыми, что обеспечивает корректность применения метода динамического программирования.

Таким образом, для решения поставленной задачи можно предложить следующий алгоритм, реализующий метод динамического программирования.

Алгоритм 1 (синтез позиционного управления).

1. Установить $k := n + 1$; вычислить для всех $S_n \in \mathbb{S}_n$ значение $B_{n+1}(T_n, S_n)$.
2. Если $k > 1$, установить $k := k - 1$, иначе перейти к шагу 4.
3. Для всех $T_{k-1} \in \mathbb{T}_{k-1}$, $S_{k-1} \in \mathbb{S}_{k-1}$ вычислить значение $B_k(T_{k-1}, S_{k-1})$ и определить $\mathbf{u}_k^*(T_{k-1}, S_{k-1})$.
4. Вычислить значение $B_1(0, 0)$ и определить управление на первом шаге $\mathbf{u}_1^*(0, 0)$.

Заметим, что если на шаге 4 алгоритма 1 дополнительно вычислить $B_1(t, s)$, то будет найдено решение задачи поиска максимума функционала $P_{\varphi-s, \bar{T}-t}^2(\mathbf{u})$. Данные вычисления можно провести, если множество достижимых состояний динамической системы при начальных условиях $T_0 = t$, $S_0 = s$ на каждом шаге k является подмножеством $\mathbb{T}_k \times \mathbb{S}_k$. Таким образом, можно получать решения поставленной задачи сразу для нескольких значений φ и $\bar{T}$.

Приведенный метод решения задачи позволяет доказать эквивалентность задач (5) и (8).

Утверждение 1. *Оптимальные значения целевых функционалов в задачах (5) и (8) равны. При этом существуют стратегии, являющиеся оптимальными в обеих задачах, а именно стратегии, удовлетворяющие условию $\mathbf{u}_k^*(T_{k-1}, S_{k-1}) = 0$ при $S_{k-1} \geq \varphi$ или $T_{k-1} > \bar{T}$.*

Доказательство. Для решения задачи (8) также применим метод динамического программирования. Для этого нужно на последнем шаге определить функцию Беллмана по правилу $\tilde{B}_{n+1}(T_n, S_n) = I\{T_n \leq \bar{T}, S_n \geq \varphi\}$. Соотношения динамического программирования для решения задачи (8) будут такими же, как и (11), с тем уточнением, что вместо функций B_k записываются функции $\tilde{B}_k$.

Проанализируем получившиеся соотношения динамического программирования. Заметим, что при $T_{k-1} > \bar{T}$ (лимит времени исчерпан) или $S_{k-1} \geq \varphi$ (требуемый балл набран) при вычислении $B_k(T_{k-1}, S_{k-1})$ и $\tilde{B}_k(T_{k-1}, S_{k-1})$ можно положить $\mathbf{u}_k^*(T_{k-1}, S_{k-1}) = 0$ (условие, сформулированное в утверждении 1).

Теперь докажем по индукции следующее утверждение: $B_k(T_{k-1}, S_{k-1}) = \tilde{B}_k(T_{k-1}, S_{k-1})$ при $T_{k-1} \leq \bar{T}$, при этом среди стратегий, на которых достигается максимум, есть совпадающие стратегии, а если $S_{k-1} < \varphi$, то максимум достигается на одних и тех же стратегиях. При $k = n$ утверждение выполнено по определению введенных функций Беллмана. Докажем, что утверждение выполнено для $B_k(T_{k-1}, S_{k-1})$ в предположении, что оно выполнено для

$B_{k+1}(T_k, S_k)$. Если $S_{k-1} \geq \varphi$ и $T_{k-1} \leq \bar{T}$, то $B_k(T_{k-1}, S_{k-1}) = \tilde{B}_k(T_{k-1}, S_{k-1}) = 1$ (как отмечено выше, можно взять $\mathbf{u}_k^*(T_{k-1}, S_{k-1}) = 0$). При вычислении $B_k(T_{k-1}, S_{k-1})$, когда $S_{k-1} < \varphi$ и $T_{k-1} \leq \bar{T}$, может возникнуть два случая: 1) если $T_k > \bar{T}$, тогда выполнено равенство $S_k = S_{k-1}$ и принимается во внимание значение $B_k(T_k, S_{k-1}) = \tilde{B}_k(T_k, S_{k-1}) = 0$; 2) если $T_k \leq \bar{T}$, тогда учитывается значение $B_k(T_k, S_{k-1}) = \tilde{B}_k(T_k, S_{k-1})$ (равенство выполнено по предположению индукции). При этом в обоих случаях $B_k(T_{k-1}, S_{k-1}) = \tilde{B}_k(T_{k-1}, S_{k-1})$ и максимум в определениях функций Беллмана достигается на одних и тех же стратегиях, что и доказывает утверждение 1.

Замечание 3. Среди оптимальных стратегий в задаче (5) могут оказаться стратегии, удовлетворяющие условию $\mathbf{u}_k^*(T_{k-1}, S_{k-1}) = 1$ при $S_{k-1} \geq \varphi$, не являющиеся оптимальными стратегиями в задаче (8). Действительно, решение дополнительных задач, когда требуемый балл уже набран, уменьшает вероятность неперевышения лимита времени, что ведет к уменьшению значения целевого функционала в задаче (8) по сравнению со значением функционала (5).

4. Синтез программного управления

Сначала опишем способ вычисления значения целевого функционала (4) при фиксированном $u \in \{0, 1\}^n$. Определим следующие функции:

$$(12) \quad B_{n+1}^u(T_n, S_n) = I\{S_n \geq \varphi\},$$

$$B_k^u(T_{k-1}, S_{k-1}) =$$

$$= \sum_{t \in \mathcal{T}_k, x \in \{0, 1\}} B_{k+1}^u(T_{k-1} + tu_k, S_{k-1} + b_k x I\{T_{k-1} + tu_k \leq \bar{T}\} u_k) p_k(t, x),$$

где $k = \overline{1, n}$. Получаемое с помощью данных формул значение $B_1^u(0, 0)$ равно значению функционала вероятности $P_{\varphi, \bar{T}}^1(u)$. Это следует из определения функции Беллмана (9), поскольку соотношения (12) аналогичны уравнениям Беллмана, но в них исключен поиск максимума, а стратегия управления на k -м шаге считается равной u_k .

Описанная процедура за счет использования динамических соотношений значительно эффективней непосредственного вычисления вероятности путем перебора всех возможных реализаций случайных величин X_k, τ_k .

Оптимальное программное управление может быть найдено путем перебора всех возможных управлений $u \in \{0, 1\}^n$. При этом можно исключить управления, для которых $\sum_{k=1}^n b_k u_k < \varphi$, поскольку такие управления не обеспечивают прохождение теста при любых реализациях случайных величин с ненулевой вероятностью. Также можно исключить управления u такие, что $u_n = 0$, потому что они не хуже, чем соответствующие управления, в которых $u_n = 1$. Действительно, попытка решить последнее задание не может уменьшить вероятность прохождения теста.

Предложим процедуру для существенного уменьшения перебираемых программных управлений. Будем считать, что известны значения функции Беллмана $B_k(S_{k-1}, T_{k-1})$, которые могут вычислены с помощью алгоритма из предыдущего раздела. Введем обозначения

$$\begin{aligned}
 (13) \quad & C_{k,k}^{u_1, \dots, u_k}(T_{k-1}, S_{k-1}) = \\
 & = \sum_{t \in \mathcal{T}_k, x \in \{0,1\}} B_{k+1}(T_{k-1} + tu_k, S_{k-1} + b_k x I\{T_{k-1} + tu_k \leq \bar{T}\} u_k) p_k(t, x), \\
 & C_{l,k}^{u_1, \dots, u_k}(T_{l-1}, S_{l-1}) = \\
 & = \sum_{t \in \mathcal{T}_l, x \in \{0,1\}} C_{l+1,k}^{u_1, \dots, u_k}(T_{l-1} + tu_l, S_{l-1} + b_l x I\{T_{l-1} + tu_l \leq \bar{T}\} u_l) p_l(t, x), \\
 & l = k-1, k-2, \dots, 1, \quad C_k(u_1, \dots, u_k) = C_{1,k}^{u_1, \dots, u_k}(T_0, S_0).
 \end{aligned}$$

Величина $C_{l,k}^{u_1, \dots, u_k}(T_{l-1}, S_{l-1})$ описывает минимальные потери начиная с шага k , если на первых k шагах применялись управления $u_1, \dots, u_k$. Вычисление этих величин соответствует соотношениям динамического программирования при фиксированных управлениях на первых k шагах. Величины $C_k(u_1, \dots, u_k)$ могут быть использованы в качестве верхних оценок критериальной функции.

Утверждение 2. Для величин, введенных формулой (13), справедливы следующие утверждения:

- 1) $C_k(u_1, \dots, u_k) \geq B_1^u(0, 0) = P_{\varphi, \bar{T}}^1(u) = C_n(u_1, \dots, u_n)$;
- 2) $C_k(u_1, \dots, u_k) \geq C_{k+1}(u_1, \dots, u_{k+1})$ для всех $k = \overline{1, n-1}$.

Доказательство. Уравнения (13) представляют собой динамические соотношения Беллмана, в которых на первых k шагах управления равны $u_1, \dots, u_k$. Поэтому величина $C_k(u_1, \dots, u_k)$ равна значению целевого функционала в задаче с позиционным управлением (5), когда на первых k шагах компоненты вектора управления $\mathbf{u}$ фиксированы и равны $u_1, \dots, u_k$, а на последующих шагах компоненты $\mathbf{u}$ выбираются оптимальным образом относительно достигнутого за k шагов состояния. Значит, для любого программного управления u , в котором фиксированы $u_1, \dots, u_k$, справедливо неравенство $C_k(u_1, \dots, u_k) \geq B_1^u(0, 0)$. Кроме того, если в управлении фиксированы $u_1, \dots, u_{k+1}$, то такое управление обеспечивает меньшее значение функционала (5), чем управление с фиксированными $u_1, \dots, u_k$, поэтому $C_k(u_1, \dots, u_k) \geq C_{k+1}(u_1, \dots, u_{k+1})$ для всех $k = \overline{1, n-1}$. Равенство $B_1^u(0, 0) = P_{\varphi, \bar{T}}^1(u)$ было обосновано выше. Утверждение 2 доказано.

Таким образом, если известна нижняя граница оптимального значения критериальной функции $\psi < \max_{u \in \{0,1\}^n} P_{\varphi, \bar{T}}^1(u)$ и для некоторого k выполнено $C_k(u_1, \dots, u_k) \leq \psi$, то в силу утверждения 2 $P_{\varphi, \bar{T}}^1(u) < \max_{u \in \{0,1\}^n} P_{\varphi, \bar{T}}^1(u)$, а значит, любая стратегия u с соответствующими первыми k компонентами не

является оптимальной. Поэтому для решения задачи поиска программной стратегии можно предложить приведенный ниже алгоритм 2, основанный на методе ветвей и границ. Данный алгоритм использует обход бинарного дерева, корень которого соответствует переменной u_1 , а ребра, выходящие из него, соответствуют значениям 1 и 0 этой переменной, смежные вершины с корнем соответствуют переменной u_2 , выходящие из них ребра – значениям переменной u_2 и т.д. Из вершин, соответствующих переменной u_n , выходят два ребра, соответствующие значениям переменной u_n и соединяющие их с листьями. Для обхода дерева применяется поиск в глубину. При этом поиск в глубину может быть прекращен, если его дальнейшее осуществление не приведет к нахождению оптимальной стратегии. Для описания момента прекращения поиска в глубину вводится функция $L(u_1, \dots, u_k)$, равная 0, если дальнейший поиск в глубину не проводится, и равная 1 в противном случае. Переменные ψ и u^* , модифицируемые при выполнении функции $L(u_1, \dots, u_k)$, являются глобальными.

Алгоритм 2 (синтез программного управления).

0. Определим функцию $L(u_1, \dots, u_k)$ по следующему правилу:

0,а) задать $\tilde{y} = (u_1, \dots, u_k, 1, \dots, 1)$; если $\sum_{k=1}^n b_k \tilde{u}_k \geq \varphi$ или $(k = n \text{ и } u_n = 1)$,

то вычислить $C_k(u_1, \dots, u_k)$, иначе вернуть $L(u_1, \dots, u_k) = 0$;

0,б) если $C_k(u_1, \dots, u_k) \leq \psi$, то вернуть $L(u_1, \dots, u_k) = 0$;

если $k = n$ и $C_n(u_1, \dots, u_n) > \psi$, то присвоить $\psi := C_n(u_1, \dots, u_n)$, $u^* := (u_1, \dots, u_n)$, вернуть $L(u_1, \dots, u_k) = 0$;

если $k = n$ и $C_n(u_1, \dots, u_n) \leq \psi$, то вернуть $L(u_1, \dots, u_k) = 0$;

иначе вернуть $L(u_1, \dots, u_k) = 1$.

1. Установить $k := 1$, $\psi := 0$, $u^* := (0, \dots, 0)$, $u_1 := 1$.

2. Вычислить $L(u_1, \dots, u_k)$.

3. Если $L(u_1, \dots, u_k) = 1$, то установить $k := k + 1$, $u_k := 1$ и перейти к шагу 2;

если $L(u_1, \dots, u_k) = 0$ и $u_k = 1$, то установить $u_k := 0$ и перейти к шагу 2;

если $L(u_1, \dots, u_k) = 0$ и $u_l = 0$ для всех $l = \overline{1, k}$, то прекратить выполнение алгоритма;

иначе присвоить $k := \max\{l \mid u_l = 1, l < k\}$, $u_k := 0$ и перейти к шагу 2.

В результате применения алгоритма 2 будет найдено оптимальное управление u^* и соответствующее оптимальное значение целевой функции $P_{\varphi, \bar{T}}^1(u^*) = \psi$.

В предложенном алгоритме поиск в глубину начинается со значений переменной $u_k = 1$, поскольку в этом случае первой исследуемой стратегией будет стратегия «решать все задачи», которой, как правило, соответствует большое значение вероятности прохождения теста.

Замечание 4. С помощью предложенного алгоритма 2 можно решить и задачу (7). Для этого нужно при определении величин $C_k(u_1, \dots, u_k)$ задать

$B_{n+1}(T_n, S_n) = I\{T_n \leq \bar{T}, S_n \geq \varphi\}$. Кроме того, на шаге 0,а) нужно исключить проверку ($k = n$ и $u_n = 1$), так как в оптимальной стратегии может быть $u_n = 0$.

5. Численные результаты

Две поставленные задачи были успешно решены для данных из статьи [11]. В рассматриваемом тесте имеется 10 заданий, за которые можно получить баллы $b_1 = \dots = b_5 = 1$, $b_6 = b_8 = 2$, $b_7 = b_{10} = 3$, $b_9 = 4$. Тестируемому необходимо набрать $\varphi = 11$ баллов. Известны вероятности правильных ответов на каждое из заданий. Каждой задаче сопоставляются три возможные реализации времени решения в случае правильного решения и три реализации в случае неправильного решения, условные вероятности которых известны и приведены в [11].

В табл. 1 приведены решения задач (4) и (5) в зависимости от параметра $\bar{T}$. Рассматриваются значения $\bar{T}$, равные различным долям от максимально возможного времени решения всех задач $T^m = 3830$ секунд. Указано время счета при использовании алгоритмов 2 и 1 для поиска программной (прогр.) и позиционной (поз.) стратегий соответственно. В последней колонке указано время решения задачи (4) при переборе всех возможных программных стратегий за исключением заведомо неоптимальных с использованием формул (12).

Таблица 1. Зависимость решений задач (4) и (5) от параметра $\bar{T}$ при $\varphi = 11$

$\bar{T}$	Оптимальная программная стратегия u^*	$P_{\varphi, \bar{T}}^1(u^*)$	$\max_{u \in \mathcal{U}} P_{\varphi, \bar{T}}^2(u)$	Время счета прогр./поз. (с)	Время счета (перебор, с)
$0,4T^m$	(0, 0, 0, 0, 1, 1, 1, 1, 0, 1)	0,1447	0,1689	0,83/0,08	1,69
$0,5T^m$	(1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1)	0,3583	0,4220	0,45/0,07	1,78
$0,6T^m$	(1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1)	0,5244	0,6130	0,53/0,07	1,93
$0,7T^m$	(1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1)	0,6906	0,7242	0,28/0,07	2,17
$0,8T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	0,7815	0,7876	0,11/0,08	2,34
$0,9T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	0,8006	0,8007	0,15/0,07	2,64

Таблица 2. Зависимость решений задач (7) и (8) от параметра $\bar{T}$ при $\varphi = 11$

$\bar{T}$	Оптимальная программная стратегия u^*	$\tilde{P}_{\varphi, \bar{T}}^1(u^*)$	$\max_{u \in \mathcal{U}} \tilde{P}_{\varphi, \bar{T}}^2(u)$	Время счета прогр./поз. (с)	Время счета (перебор, с)
$0,4T^m$	(0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1)	0,1447	0,1689	0,99/0,08	3,47
$0,5T^m$	(1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1)	0,2460	0,4220	3,02/0,07	3,45
$0,6T^m$	(1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1)	0,4963	0,6130	1,49/0,07	3,45
$0,7T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1)	0,6059	0,7242	0,64/0,07	3,46
$0,8T^m$	(1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1)	0,7216	0,7876	0,63/0,07	3,41
$0,9T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	0,7965	0,8007	0,14/0,07	3,42

Аналогично в табл. 2 приведены решения задач (7) и (8). Указано время счета при использовании модификаций алгоритмов 2 и 1. Все расчеты проводились на компьютере Acer Aspire A315-54K (Intel Core i5-6300U 2.4 GHz, 8Gb RAM).

Из приведенных таблиц видно преимущество использования позиционных стратегий по сравнению с программными не только с точки зрения оптимального значения целевого функционала, но и с точки зрения скорости вычислений. Если сравнить постановку (4) с постановкой из [11], то можно отметить, что, как правило, решение задачи (4) предполагает большее количество решаемых задач и большее значение целевого функционала. Это связано с тем, что в задаче (4) не требуется уложиться в лимит времени.

Как отмечалось ранее, решение задач (4) и (5) зависит от порядка тестовых заданий. Поэтому были проведены дополнительные вычисления стратегий прохождения того же теста, но с обратным порядком задач, т.е. первой задачей теста считалась десятая задача рассмотренного выше теста, второй задачей являлась девятая задача предыдущего теста и т.д. Баллы, присуждаемые за задания, и вероятности решения заданий остались такими же. Результаты решения этой задачи приведены в табл. 3. Для удобства сравнения таблиц компоненты оптимальных программных стратегий выведены в обратном порядке. Время счета в этой таблице не указано, поскольку оно незначительно отличается от времени счета, указанного в табл. 1. Сравнение полученных решений показывает, что значения критериальных функций незначительно изменились при изменении порядка задач, а программная стратегия отличается для ограничений по времени $0,4T^m$ и $0,6T^m$.

Таблица 3. Зависимость решений задач (4) и (5) от параметра $\bar{T}$ при $\varphi = 11$ при обратном порядке заданий

$\bar{T}$	Оптимальная программная стратегия $(u_{10}^*, \dots, u_1^*)$	$P_{\varphi, \bar{T}}^1(u^*)$	$\max_{\mathbf{u} \in \mathcal{U}} P_{\varphi, \bar{T}}^2(\mathbf{u})$
$0,4T^m$	(1, 0, 1, 0, 1, 1, 1, 1, 0, 1)	0,1639	0,1787
$0,5T^m$	(1, 1, 1, 0, 1, 1, 1, 1, 0, 1)	0,3526	0,3834
$0,6T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 0, 1)	0,5340	0,5718
$0,7T^m$	(1, 1, 1, 0, 1, 1, 1, 1, 1, 1)	0,6781	0,7003
$0,8T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	0,7824	0,7847
$0,9T^m$	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)	0,8007	0,8007

6. Заключение

В работе решены задачи поиска программной и позиционной стратегий прохождения ограниченного по времени теста в условиях доступности тестируемому информации о количестве набранных за тест баллов после решения каждого очередного задания. Продемонстрирована эффективность предложенных алгоритмов решения. Проведен сравнительный анализ полученных результатов с результатами решения ранее рассматриваемой одним из ав-

торов постановки задачи. Анализ показал, что доступность тестируемому в процессе прохождения теста информации о количестве набранных баллов после решения каждого задания позволяет предложить стратегию прохождения теста, существенно повышающую вероятность превышения суммарным количеством набранных за тест баллов требуемого уровня.

В дальнейшем могут быть рассмотрены различные модификации изученной модели. В частности, может быть учтена зависимость результатов решения задач теста от набранного балла, возникающая вследствие психологических особенностей некоторых тестируемых. Требуется изучение возможности модификации уравнений Беллмана (11) в этом случае. Также представляют интерес использование непрерывных случайных величин для описания времени ответов на вопросы теста и исследование точности решения задачи при их дискретизации.

СПИСОК ЛИТЕРАТУРЫ

1. *Rasch G.* Probabilistic models for some intelligence and attainment tests. Chicago: The University of Chicago Press, 1980.
2. *Xiao J., Bulut O.* Item selection with collaborative filtering in On-the-fly multistage adaptive testing // *Appl. Psychol. Meas.* 2022. V. 46. No. 8. P. 690–704.
3. *Наумов А.В., Джумурат А.С., Иноземцев А.О.* Система дистанционного обучения математическим дисциплинам CLASS.NET // *Вест. компьют. и информ. технологий.* 2014. № 10. С. 36–40.
4. СДО МАИ CLASS.NET [Электронный ресурс] // URL: <http://www.distance.kaf804.ru/> (дата обращения: 12.10.2024.)
5. *Kuravsky L.S., Margolis A.A., Marmalyuk P.A., et al.* A Probabilistic Model of Adaptive Training // *Appl. Math. Sci. (Ruse).* 2016. V. 10. No. 48. P. 2369–2380.
6. *Босов А.В., Мартюшова Я.Г., Наумов А.В., Сапунова А.П.* Байесовский подход к построению индивидуальной траектории пользователя в системе дистанционного обучения // *Информатика и ее применения.* 2020. Т. 14. № 3. С. 86–93.
7. *Босов А.В.* Применение самоорганизующихся нейронных сетей к процессу формирования индивидуальной траектории обучения // *Информатика и ее применения.* 2022. Т. 16. № 3. С. 7–15.
8. *Van der Linden W.J., Scrams D.J., Schnipke D.L.* Using Response-Time Constraints to Control for Differential Speededness in Computerized Adaptive Testing // *Appl. Psych. Meast.* 1999. V. 23. No. 3. P. 195–210.
9. *Босов А.В., Мхитарян Г.А., Наумов А.В., Сапунова А.П.* Использование гамма-распределения в задаче формирования ограниченного по времени теста // *Информатика и ее применения.* 2019. Т. 13. № 4. С. 12–18.
10. *Наумов А.В., Мхитарян Г.А., Черыгова Е.Е.* Стохастическая постановка задачи формирования теста заданного уровня сложности с минимизацией квантили времени выполнения // *Вест. компьют. и информ. технологий.* 2019. № 2. С. 37–46.
11. *Наумов А.В., Степанов А.Е., Устинов А.Э.* О задаче максимизации вероятности успешного прохождения ограниченного по времени теста // *АиТ.* 2024. № 1. С. 83–94.

Naumov A.V., Stepanov A.E., Ustinov A.E. On the Problem of Maximizing the Probability of Successful Passing of a Time-Limited Test // Autom. Remote Control. 2024. V. 85. No. 1. P. 64–72.

12. *Мартюшова Я.Г., Наумов А.В., Степанов А.Е.* Оптимизация ограниченного по времени теста по квантильному критерию // Информатика и ее применения. 2024. Т. 18. № 4. С. 44–51.

Статья представлена к публикации членом редколлегии А.И. Кибзуном.

Поступила в редакцию 23.02.2025

После доработки 08.04.2025

Принята к публикации 12.04.2025