

ОЦЕНКА ВЕСА ЗАДАНИЙ ДЛЯ ФОРМИРОВАНИЯ ИЕРАРХИЧЕСКОЙ СИСТЕМЫ ТЕСТИРОВАНИЯ ПО ИНФОРМАТИКЕ

С.В. Криволапов

Кафедра программного обеспечения
и администрирование информационных систем
Курский государственный университет
ул. Радищева, 33, Курск, Россия, 305000

В статье описана концепция поиска наиболее оптимального решения иерархической системы тестовых заданий при помощи классических задач поиска кратчайшего пути графа при обучении информатики.

Ключевые слова: обучение, проверка знаний, информатика, система заданий, тестирование.

В современном мире возрастает конкуренция на рынке труда. Соответственно, мы можем говорить о возрастании требований к будущему сотруднику, а следовательно, и к современному студенту. С каждым годом объем знаний, которые студенту необходимо усвоить, чтобы стать квалифицированным специалистом, возрастает. Классическая система обучения уже не удовлетворяет возросшим требованиям работодателя.

Представим систему обучения в виде графа (рис. 1).

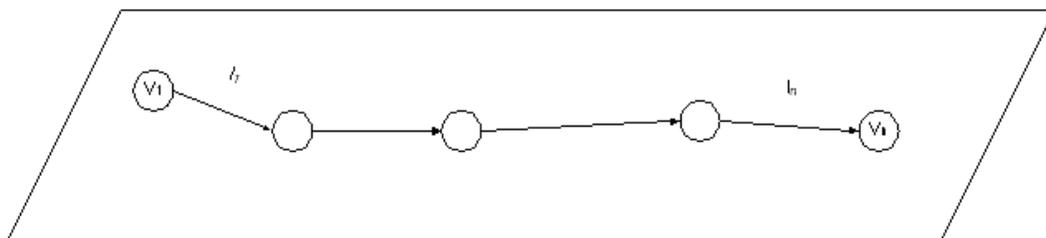


Рис. 1. Классическая система обучения

Процесс обучения осуществляется по следующей схеме: выделяются части (блоки) изучаемого материала; каждая часть должна быть изучена именно тогда, когда она запланирована.

Этот принцип обучения называется «переход от менее сложного материала к более сложному». Если обучаемый не усвоил один из блоков материала, то в дальнейшем он столкнется с проблемами в изучении следующего, более сложного блока. Для того чтобы проверить, насколько успешно обучаемый усвоил материал, после изучения очередного блока ему предоставляется возможность пройти проверку знаний, умений, навыков путем выполнения различных типов заданий. Эта проверка позволяет учащемуся закрепить материал, а преподавателю — узнать, где у обучаемого есть пробелы в знаниях.

Классическая система проверки знаний также может быть представлена в виде графа. Она представляет собой систему заданий, при решении которых обучающийся получает результирующие баллы, а после их подсчета — оценку.

Классическая система проверки знаний, на которой базируется большинство современных электронных тестовых систем, не дает целостной и объективной информации об объеме и качестве усвоения обучаемым знаний, умений и навыков, поэтому возникла необходимость разработки новой тестовой системы. При этом необходимо, с одной стороны, облегчить рутинный труд преподавателя по проверке знаний обучаемого, с другой — уменьшить время обработки результатов и обеспечить получение целостной и четкой информации об уровне знаний учащегося.

Рассмотрим схему, представленную на рис. 2 более подробно с математической точки зрения.

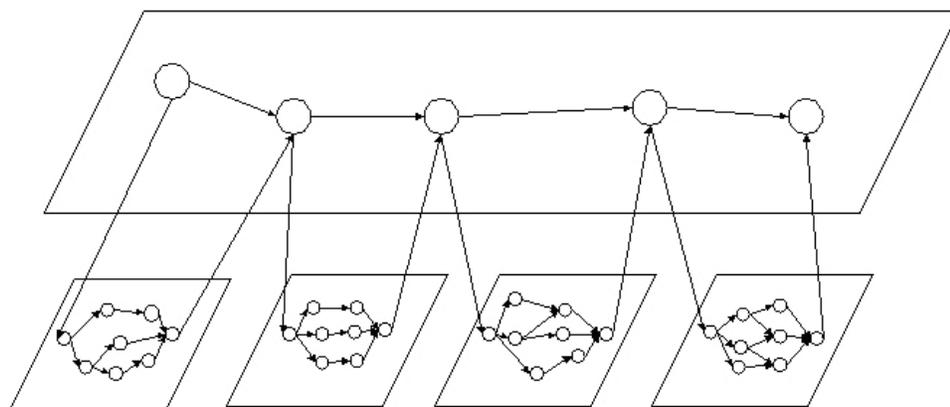


Рис. 2. Двухуровневая система проверки знаний

На верхнем уровне имеется множество тестовых заданий, которые охватывают конкретную предметную область. Для наглядности они представлены в виде графа. Вершины представляют собой множество тестовых заданий, а ребра — связь между заданиями, если таковая имеется.

Рассмотрим модульный процесс построения системы. Данная система представляет собой двухуровневую структуру, верхний уровень которой — множество тестов из данной предметной области, объединенных в граф. Особенность данной модели заключается в том, что при неправильном ответе ученику не ставится сразу 0 баллов, а дается уточняющий набор заданий, который представлен вторым уровнем. Таким образом, при неправильном ответе происходит замена одной дуги графа на подграф, что представляет собой сумму графов, и мы получаем граф с более длинным путем прохождения (см. рис. 2).

Граф разбивается на уровни и подуровни, которые на рис. 2 изображены плоскостями. Подуровни представляют собой подмодели, которые, в свою очередь, могут иметь подграфы. Таким образом, модель может быть представлена в виде двухуровневого графа, причем нижний уровень — рекурсивно определяемая подмодель. Полученный граф с точки зрения теории графов является деревом, а про-

цесс проверки знаний представляет собой путь от первой вершины (корня дерева) до последней — верхнего модуля.

В зависимости от результатов этот путь может быть представлен в виде цепи (в случае всех правильных ответов) или иметь ветвление из вершин графа.

Каждое ветвление представляет собой переход на более низкий уровень, определяемый подмодулем. На этом уровне программа должна выявить причины неправильного ответа, откорректировать знания и перейти на следующую вершину более высокого уровня для дальнейшего прохождения мониторинга.

Система позволяет проверить знания и уточнить, на каком уровне обучаемый допускает ошибку.

Так, на нижнем уровне должны быть унаследованы все свойства верхнего уровня и, кроме того, нижний модуль должен обладать свойством всесторонней проверки, т.е. нижний уровень должен предусмотреть все возможные варианты ошибок. Модель нижнего уровня рекурсивно определяется на верхнем уровне.

При работе с иерархическими системами тестовых заданий одной из основных проблем является поиск оптимального решения. В основах теории графов существует ряд задач, при частном решении которых возможно применение полученных результатов в рамках поиска оптимального решения системы тестовых заданий. Рассмотрим применение одной из классических задач в области проверки результативности обучения.

Задача о кратчайшем решении. Пусть задана адаптивная система тестовых заданий из $n + 1$ вопросов, в которой выделены два вопроса — вход (нулевая вопрос) и выход (вопрос с номером n). Для каждого ответа заданы числа, называемые весом ответа.

Весом решения системы тестовых заданий называется сумма весов входящих в него ответов (если веса ответов не заданы, то вес решений определяется числом входящих в него ответов). Задача заключается в поиске кратчайшего решения (решения минимального веса) от нулевого вопроса до вопроса с номером n .

Известно, что для существования решения с минимальным весом необходимо и достаточно отсутствия в решении ответов отрицательного веса.

Предположим, что в решении нет повторов. Тогда всегда можно пронумеровать вопросы таким образом, что для любого ответа i, j имеет место $j > i$. Такая нумерация называется правильной. Легко показать, что в системе заданий без повторов всегда существует правильная нумерация. Обозначим L_{ij} — вес ответа i, j . Кратчайшее решение в системе заданий, имеющей правильную нумерацию, определяется следующим алгоритмом (рис. 3).

Алгоритм 1:

Шаг 0: помечаем нулевое задание индексом $\lambda_0 = 0$;

Шаг k : помечаем задание k индексом $\lambda_k = \min_{i < k} (\lambda_k - l_{ik})$.

Индекс ответа λ_n будет равен длине кратчайшего решения. На рисунке 3 приведен пример применения алгоритма 1 для определения кратчайшего решения (числа у дуг равны весам ответов, индексы вопросов помечены в квадратные скобки, кратчайшее решение выделено двойными линиями).

Когда индексы (называемые в некоторых задачах потенциалами задания) установятся, кратчайшее решение определяется методом обратного хода от выхода к входу, т.е. кратчайшим является решение $\mu = (0; i_1; i_2; \dots; i_{n-1}; n)$, такое, что $\lambda_{i_{n-1}n} = \lambda_n - \lambda_{n-1}$ и т.д.

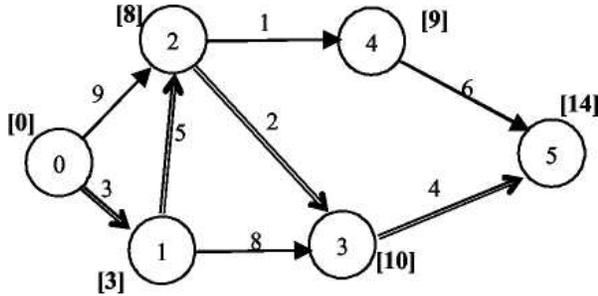


Рис. 3. Поиск кратчайшего пути решения

Следующий алгоритм дает возможность определять кратчайшее решение в общем случае (т.е. при произвольной расположении заданий).

Алгоритм 2 (алгоритм Форда)

Шаг 0: помечаем нулевое задание индексом $\lambda_0 = 0$, все остальные задания индексами $\lambda_i^0 = +\infty$, $i = \overline{1, n}$.

Шаг k : рассматриваем все ответы. Если для ответа $(i; j)$, $\lambda_j^{k-1} - \lambda_i^{k-1} = l_{ij}$, то вычисляем новое значение $\lambda_j^k = \lambda_i^{k-1} + l_{ij}$.

Индексы устанавливаются за конечное число шагов. Обозначим: $\{\lambda_i^*\}$ — установившиеся значения индексов, которые обладают следующим свойством: величина λ_i^* равна весу кратчайшего решения из нулевого задания в задание i . Кратчайшее решение из задания 0 к заданию i определяется методом обратного хода.

Если вес всех ответов неотрицательный, то для поиска кратчайшего решения применим следующий алгоритм.

Алгоритм 3

Шаг 0: помечаем нулевое задание индексом $\lambda_0 = 0$.

Шаг k : пусть уже помечено некоторое множество заданий. Обозначим: Q — множество непомеченных заданий, смежных с помеченными. Для каждого задания $k \in Q$ вычисляем величину $\xi_k = \min(\lambda_i - \lambda_{ik})$, где минимум берется по всем помеченным заданиям i , смежным с заданием k . Помечаем задание k , для которого величина ξ_k минимальна, индексом $\lambda_k = \xi_k$.

Подобную процедуру повторяем до тех пор, пока не будет помечено задание n . Длина кратчайшего решения равна λ_n , а само кратчайшее решение определяется так, как это было описано выше.

Запишем задачу о кратчайшем решении как задачу линейного программирования (ЛП). Пусть $x_{ij} = 1$, если ответ (i, j) входит в решение μ , $x_{ij} = 0$, если ответ (i, j) не входит в решение μ , $i, j = \overline{0, n}$.

Задачу о минимальном решении можно записать в виде

$$L(x) = \sum_{i,j=0}^n l_{ij} x_{ij} \rightarrow \min_x, \quad (1)$$

$$\sum_j x_{0j} = 1, \sum_j x_{jn} = 1, \quad (2)$$

$$\sum_i x_{ki} = \sum_j x_{jk}, k = \overline{1, n-1}. \quad (3)$$

Любое решение системы неравенств (2)—(3) определяет решение в системе заданий без повторов (но не в системе заданий с повторами).

Пусть все повторы имеют строго положительный вес, т.е. нет повторов отрицательного и нулевого веса. Тогда решение задачи (1)—(3) определяет путь кратчайшего решения.

Сформулируем задачу ЛП, двойственную задаче (1)—(3), поставив в соответствие ограничениям (2) двойственные переменные λ_0 и λ_n , а ограничениям (3) — двойственные переменные $\{\lambda_i\}$, $i = \overline{1, n-1}$:

$$\lambda_n - \lambda_0 \rightarrow \max, \quad (4)$$

$$\lambda_j - \lambda_i \leq l_{ij}, i, j = \overline{0, n}. \quad (5)$$

По теореме двойственности линейного программирования, для оптимальных решений задач (1)—(3) и (4)—(5) значения целевых функций совпадают.

Задача (4)—(5) называется задачей о потенциалах заданий тестовой системы. Общая ее формулировка такова: найти потенциалы задания $\{\lambda_i\}$, удовлетворяющие системе неравенств (5) и максимизирующие некоторую функцию $\Phi(\lambda)$, где $\lambda = (\lambda_0, \lambda_1, \dots, \lambda_n)$. Примером является задача о ближайших потенциалах, в которой $\Phi(\lambda) = \sum_j |\lambda_j - \lambda_j^0|$, где $\{\lambda_j^0\}$ могут интерпретироваться как желательные потенциалы.

Аналогично задаче о кратчайшем решении формулируется и решается задача о максимальном (длиннейшем) решении системы заданий: достаточно изменить знаки ответов на противоположные и решить задачу о кратчайшем решении. Для существования решения задачи о максимальном решении необходимо и достаточно отсутствия повторов положительной длины.

В задаче поиска решения максимальной надежности веса ответов интерпретируются, например, как вероятности того, что существует связь между соответствующими двумя заданиями. Заменяя веса ответов их логарифмами, взятыми с обратными знаками, получаем, что решение максимальной надежности в исходной системе заданий будет соответствовать кратчайшему пути в новом графе.

Гораздо более сложными (NP -полными) являются задачи поиска элементарных решений минимального (максимального) веса в случае, когда в системе имеются повторы отрицательного (положительного) веса. Эффективных (не сводящихся к полному перебору) точных алгоритмов для них не существует.

К таким же сложным задачам относятся и задачи поиска кратчайших или длиннейших решений или повторов, проходящих через все задания системы тестирования; элементарный вес, проходящий через все задания системы тестирования, называется *гамильтоновым решением*. Классическим примером задачи поиска гамильтонова решения является *задача коммивояжера*, заключающаяся в следующем. Коммивояжер (бродячий торговец) должен посетить n городов, побывав в каждом ровно один раз, и вернуться в исходный пункт своего путешествия. Заданы неотрицательные длины дуг, интерпретируемые как расстояние между городами или стоимость проезда. Требуется найти гамильтонов контур минимальной длины (в графе из n вершин существует $n!$ гамильтоновых контуров).

Алгоритмы решения задачи о кратчайшем решении позволяют решать широкий класс задач дискретной оптимизации. В качестве примера приведем задачу целочисленного линейного программирования — *задачу о ранце (рюкзаке)*, к которой сводятся многие практически важные задачи определения оптимальной комбинации факторов при ограничениях на общий вес, площадь, объем, финансирование и т.д.

Рассмотренные выше частные решения типовых задач теории графов позволяют усовершенствовать систему проверку уровня усвояемости знаний с использованием систем тестового контроля. Одной из наиболее сложных проблем при работе с тестовыми системами является проверка результата решения обучаемым системы тестовых заданий. Использование экспертной проверки при обработке результатов решения систем тестовых заданий очень сильно увеличивает время обработки результатов. Применение типовых задач теории графов позволит уменьшить время обработки результатов, уменьшит долю использования работы экспертов при проверке результатов, что увеличит объективность оценки результата обучаемого.

ЛИТЕРАТУРА

- [1] *Айсмонтас Б.Б.* Теория обучения: схемы и тесты. — М.: Владос-Пресс, 2002.
- [2] *Дубров С.И., Нечаев Ю.И., Резников Ю.Е.* Рейтинговая система оценки знаний как способ стимулирования работы студентов. URL: <http://www.ict.edu.ru/vconf/index.php>
- [3] *Образцов П.И.* Обеспечение учебного процесса в условиях информатизации высшей школы // Педагогика. — 2003. — № 5. — С. 27—32.
- [4] *Кузовлева. К.Т.* Конструирование педагогических тестов на основе современных математических моделей. URL: <http://www.informika.ru>
- [5] *Криволапов С.В.* Средства информатизации проверки результативности обучения, основанные на иерархических системах заданий // Вестник РУДН. Серия «Информатизация образования». — 2010. — № 3. — С. 5—12.
- [6] *Попкова В.А., Коржуев А.В.* Дидактика высшей школы: Учеб. пособие. — М.: Академия, 2008.

TASK'S WEIGHT ESTIMATION FOR HIERARCHICAL SYSTEM OF TESTS IN INFORMATICS

S.V. Krivolapov

Software and information system administration department

Kursk state university

Radisheva str., 33, Kursk, Russia, 305000

The concept of hierarchical system of tasks most optimal solution search using classical problems of graph's shortest path search in teaching informatics is described in this article.

Key words: teaching, examination, informatics, tasks system, testing.