ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 004.942

РЕШЕНИЕ ЗАДАЧИ ПЛАНИРОВАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ НА ОСНОВЕ ЖАДНОГО АЛГОРИТМА

С. В. Скворцов, д.т.н., профессор кафедры САПР ВС РГРТУ; профессор Академии ФСИН России, Рязань, Россия;
orcid.org/0000-0001-9495-4953, e-mail: s.v.skvor@gmail.com
А. А. Кузнецов, магистрант РГРТУ, Рязань, Россия;
orcid.org/0000-0003-1421-0941, e-mail: alexander.kuznetsov996@gmail.com
Т. С. Скворцова, к.т.н., старший преподаватель Академии ФСИН России;
orcid.org/0000-0001-7504-5973, e-mail: t.s.skvortsova@yandex.ru

Рассматривается задача планирования проведения занятий на основе модели учебного курса, учитывающей структуру дисциплины и ее семантику, представленную в терминах понятий и логических зависимостей между ними. ***Целью работы*** *является разработка и исследование алгоритма определения порядка изучения тем (понятий) учебной дисциплины на основе жадной стратегии. Предложены модификации алгоритма, позволяющие получать решения по критериям качества на основе функции забывания, позволяющей оценить уровень освоения знаний, и числа соседней связности изучаемых понятий, характеризующего их логическую взаимосвязь в процессе изучения. Представлены результаты экспериментальных исследований разработанного алгоритма с целью анализа производительности и качества получаемых решений.*

Ключевые слова: *планирование учебного процесса, модель учебной дисциплины, жадный алгоритм, функция забывания, число соседней связности.*

DOI: 10.21667/1995-4565-2020-72-71-82

Введение

При планировании учебного процесса часто возникает проблема составления расписания изучения учебного материала, так как во многих случаях имеется возможность выбора той или иной очередности освоения тем дисциплин. Данная проблема часто решается интуитивно, хотя при этом нужно учитывать множество факторов, например, таких, как загрузка преподавателя и аудиторий, удобное время проведения занятий и т.п. Порядок изложения учебного материала дисциплин непосредственно влияет на уровень его освоения студентами, что, в свою очередь, определяет качество подготовки специалистов.

Целью данной **работы** является анализ возможных подходов к планированию занятий по учебной дисциплине с учетом ее семантики [1], а также разработка и исследование алгоритма определения последовательности изучения понятий (тем) учебного курса на основе жадной стратегии поиска решения.

Постановка задачи планирования занятий по учебной дисциплине

Для постановки и решения задачи планирования занятий учебного курса будем использовать его модель, которая учитывает содержание дисциплины и позволяет моделировать процессы ее изучения с учетом забывания и семантики изложенного материала [1]. В рамках этой модели структура и содержание курса описываются с помощью ориентированного дерева $G = (X, U)$, корень которого определяет название дисциплины, узлы $x_i \in X$ соответствуют ее разделам или подразделам, а листья из $X' \subset X$ – конечным понятиям, вводимым в рассмотрение и используемым при проведении учебных занятий. Дуги из множества U в таком дереве направлены от родителей (разделов) к потомкам (подразделам, понятиям) и показывают отношения включения вида «раздел А содержит подраздел В» или «(под)раздел С содержит понятие В».

Кроме отношений включения, модель учебного курса определяет отношения порядка. Данный тип отношений учитывает логические зависимости между понятиями и разделами дисциплины, которые задаются множеством дуг $E = \{e_{ij} = (x_i, x_j)\}$ между связанными понятиями (листьями дерева) или (под)разделами (узлами дерева). Каждая дуга означает, что основополагающее понятие должно быть изучено раньше зависимого понятия.

Полная графовая модель учебного курса описывается как $G^* = (X, U, E)$ и совмещает учет структурных характеристик изучаемого материала и их семантику, выраженную логическими зависимостями. Пример такой модели изображен на рисунке 1, где сплошными линиями показаны дуги, отражающие отношение включения, а штриховыми - дуги из множества E, соответствующие логическим зависимостям. Например, дуга $e_{78} = (x_7, x_8)$ здесь определяет зависимость: «понятие Д основывается на понятии Г». В данном случае предполагается, что основополагающим понятием является Г.

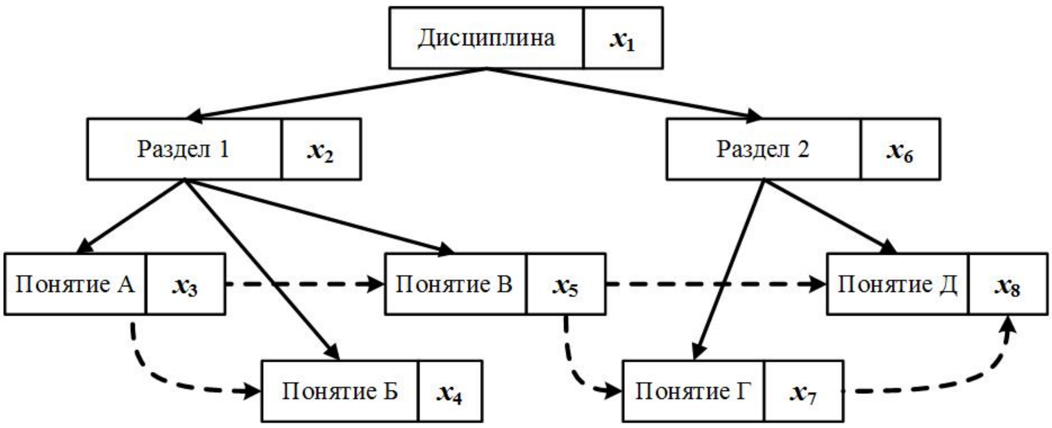

Рисунок 1 – Полная графовая модель дисциплины
Figure 1 - Full graph model of discipline

Временная последовательность изучения тем курса задает порядок (линейный или частичный), в котором будут изучаться соответствующие понятия. Для этого каждому понятию $x_i \in X'$, где X' – множество листьев дерева $G = (X, U)$, ставится в соответствие дата его изучения t_i.

Формируемая модель освоения курса не должна противоречить логическим зависимостям, определяющим отношения предшествования. Если есть логическая зависимость e_{ij} вида «понятие x_j основано на понятии x_i», то очевидно соотношение:

$$t_j \geq t_i, \; e_{ij} \in E, \tag{1}$$

которое означает, что изучение понятия x_j не может осуществляться ранее изучения понятия x_i.

Планирование освоения учебного курса представляет собой задачу упорядочения во времени занятий по изучению соответствующих тем курса. Математически эта задача сводится к нахождению значений переменных t_i, определяющих порядок изучения понятий x_i и обеспечивающих получение экстремума некоторой целевой функции F при соблюдении ограничений, задаваемых деревом $G = (X, U)$ и условием (1), т.е. набором логических зависимостей, представленных дугами множества E.

Для решения сформулированной задачи можно использовать различные варианты целевой функции F, учитывающие определенные особенности учебного процесса или ожидаемых результатов. В частности, в работах [1, 2] предложены критерии оптимальности, основанные на использовании функции забывания [3] и числа соседней связности [4] учебных понятий.

В данной работе будем использовать критерий качества освоения учебного материала, определяемый на основе функции забывания:

$$R(t) = e^{-t/s}, \qquad (2)$$

где t – время, s – стабильность памяти, которая определяет скорость снижения воспроизводимости информации в результате забывания следующим образом [1]. Если граф G содержит дугу $e_{ab} \in E$, которая определяет логическую зависимость (x_a, x_b), то на занятии, запланированном на время t_b, значение функции забывания $R_a(t)$ для понятия x_a возрастает на некоторую величину h, т.е. происходит скачок, имитирующий вспоминание информации. Стабильность памяти s при этом увеличивается на целое число m, равное количеству изученных на этом занятии понятий, основанных на понятии x_a.

Если по формуле (1) получить значения $R_i(t)$ функции забывания для отдельных понятий $x_i \in X'$, то с использованием дерева $G = (X, U)$, которое просматривается снизу вверх от листьев к корню, можно найти обобщенную функцию $R^* = R(t^*)$, где t^* – момент контрольного мероприятия (например, зачет или экзамен). Значение R^* можно использовать в качестве критерия оптимальности $F = R^*$, максимум которого необходимо получить.

Возможные подходы к решению задачи

Наиболее известным подходом к решению задач оптимизации многошаговых процессов принятия решений является применение метода динамического программирования, основанного на принципе оптимальности Беллмана. Для успешного применения этого метода целевая функция задачи должна удовлетворять условию аддитивности, которое означает, что итоговое значение критерия эффективности за n шагов можно получить как сумму значений этого критерия, полученных на отдельных шагах [5, 6]. Однако специфика решаемой задачи не позволяет представить целевую функцию в виде, удовлетворяющем указанному условию, поскольку ее вычисление связано со структурными характеристиками модели учебного курса и логическими зависимостями между его разделами, подразделами и отдельными понятиями, которые описываются полной графовой моделью G^* [1, 2].

Рассмотрим подграф $H = (X', E')$ графа G^*, построенный на множестве X' листьев, связанных дугами $E' \subset E$, где $E' = \{e_{ij} = (x_i, x_j)\text{: } \{x_i, x_j\} \cap X' \neq \varnothing\}$. Очевидно, что ориентированный граф H описывает отношение предшествования, задаваемое логическими зависимостями на множестве конечных понятий учебного курса, и поэтому является бесконтурным. Например, для графа G^*, показанного на рисунке 1, соответствующий подграф H приведен на рисунке 2.

Укладкой (нумерацией) вершин графа $H = (X', E')$ будем называть такую последовательность $L(H) = (x_1, x_2, \ldots, x_n)$ всех его вершин $x_i \in X'$, что для каждой дуги $(x_k, x_p) \in E'$ выполняется условие $k < p$. Решением поставленной задачи будет являться такая укладка $L(H)$, для которой $R^* \to \max$. Величина функции забывания R^* зависит от укладки $L(H)$ и вычисляется следующим образом. Каждому понятию $x_i \in X'$ ставится в соответствие момент его изучения $t_i = k$, где k – позиция вершины x_i в укладке $L(H)$. Затем вычисляются значения $R_i(t)$ функции забывания для отдельных понятий, и с использованием дерева G находится целевая функция $F = R^*$ [1].

Для получения укладки $L(H)$, определяющей решение задачи, можно использовать различные подходы. Самым простым и очевидным представляется полный перебор вариантов. Без учета логических зависимостей, т.е. при $E' = \varnothing$, общее количество вариантов решений (перестановок) равно $n!$, где $n = |X'|$. В примере (рисунок 2) число понятий $n = 5$ и $n! = 120$. Учет логических зависимостей заметно сокращает число допустимых вариантов укладок

$L_1 = (x_3, x_4, x_5, x_7, x_8)$, $L_2 = (x_3, x_5, x_4, x_7, x_8)$, $L_3 = (x_3, x_5, x_7, x_4, x_8)$, $L_4 = (x_3, x_5, x_7, x_8, x_4)$,

формирование которых представлено на рисунке 3. В этом примере общее количество допустимых вариантов укладки равно четырем, но в общем случае объем задачи остается довольно значительным, так как модели реальных учебных курсов содержат существенно большее число изучаемых понятий.

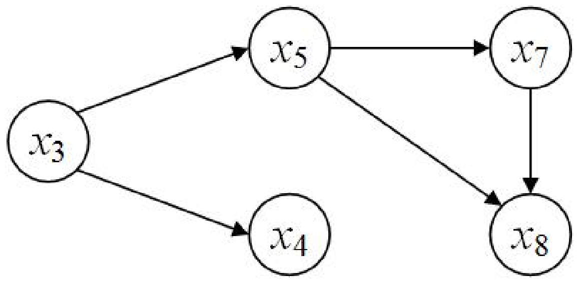

Рисунок 2 – Граф зависимостей понятий учебного курса
Figure 2 – Discipline concepts dependency graph

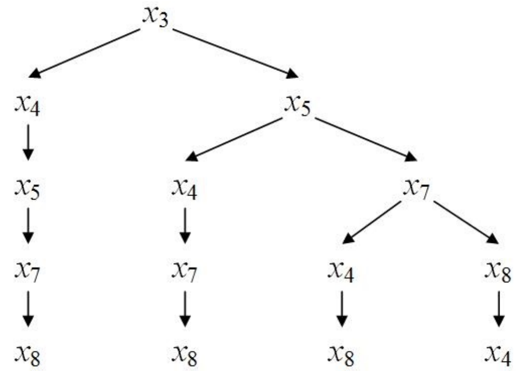

Рисунок 3 – Дерево перебора вариантов укладки графа
Figure 3 – Graph layout enumeration tree

Анализ рисунка 3 позволяет предложить второй способ решения задачи на основе перебора – поиск с возвратами или метод ветвей и границ, поскольку процедура ветвления (разбиения множества допустимых решений на подмножества) легко реализуема. Кроме ветвления, еще одной важной процедурой алгоритма на основе метода ветвей и границ является вычисление оценки целевой функции для множества (подмножества) решений задачи. Однако для этого не удается напрямую использовать функцию забывания $R(t)$, так как ее значения оценивают качество освоения материала в момент времени t и только косвенно связаны с *порядком* изучения понятий, что, в свою очередь, не позволяет определить верхнюю (нижнюю) границу целевой функции.

Еще один возможный подход к решению задачи основан на использовании генетического алгоритма [1]. Его главным достоинством применительно к решаемой задаче является возможность непосредственного использования значений функции забывания $R(t)$ для оценки качества формируемых решений $L(H) = (a_1, a_2, \ldots, a_n)$. Хромосомы представляются в виде массивов из n целых чисел, где в i-й позиции, соответствующей моменту t_j, указывается индекс j изучаемого понятия x_j. Иначе говоря, целое число j представляет собой аллель (значение) i-го гена хромосомы. При этом структура хромосомы должна учитывать ряд ограничений, задаваемых логическими зависимостями и требованием уникальности аллелей в каждой хромосоме [1].

Основная проблема применения генетического алгоритма обусловлена сложностью учета таких ограничений при реализации операторов кроссинговера и мутации. Применение стандартных генетических операторов может привести к формированию хромосом, нарушающих указанные ограничения. Для поддержания корректности хромосом, а значит и допустимости формируемых решений, требуется использовать модифицированные генетические операторы кроссинговера и мутации [1]. Но их реализация сильно усложняется из-за необходимости постоянно проверять факт того, что отношения предшествования, задаваемые ориентированным графом H, не нарушены. В противном случае после выполнения операторов кроссинговера или мутации нужно возвращаться к исходному состоянию хромосом и повторять выполнение генетических операторов до тех пор, пока не будет достигнута корректность формируемых решений.

Таким образом, в рамках рассматриваемой задачи методы перебора решений всегда дают точный и корректный результат, но не применимы к задачам реальной размерности. Кроме того, метод ветвей и границ не позволяет реализовать вычисление оценок критерия оптимальности на основе функции забывания, а использование генетического алгоритма затруднено из-за сложностей применения операторов кроссинговера и мутации.

Решение задачи на основе жадного алгоритма

Рассмотрим ситуацию, в которой преподавателю требуется изложить достаточно много учебного материала, а запланированных аудиторных занятий явно недостаточно. В таких условиях преподаватель вынужден выбирать, какие понятия (темы) учебного курса изучать во время аудиторных занятий, а какие оставить для самостоятельного освоения обучающимися. Для этого можно отсортировать понятия в порядке убывания их значимости и рассмотреть на занятиях только самые важные темы. Но как определить, какие понятия (темы) считать более значимыми? Интуитивно можно предположить, что обязательно нужно объяснять такие понятия, от которых будет зависеть освоение связанных дисциплин, однако возможны и другие подходы к оценке важности учебного материала.

Таким образом, если изучаемые понятия имеют разную значимость, то возможна процедура назначения каждому понятию некоторой величины (веса), количественно характеризующей его важность. В терминах рассмотренной модели будем считать, что каждое понятие $x_i \in X'$ имеет вес v_i, в качестве которого может рассматриваться значение $k_i = |K_i|$, где $K_i \subset X'$ – это подмножество вершин графа H, достижимых из вершины x_i. Значения весов $v_i = k_i$ вершин $x_i \in X'$ легко можно получить за один обход графа H в глубину.

Возвращаясь к решаемой задаче, использование весовых коэффициентов позволяет при составлении плана проведения занятий выбирать изучаемые понятия в порядке убывания их значимости, что предположительно с большой вероятностью позволит получить «хорошую» последовательность изучения тем. Такой подход к планированию занятий учебной дисциплины соответствует стратегии работы «жадного» алгоритма, который не гарантирует достижения оптимального результата относительно целевой функции F, но обеспечивает получение «хорошего» решения за небольшое время.

В большинстве случаев алгоритмы, предназначенные для решения задач оптимизации, реализуют пошаговую стратегию поиска решения, причем на каждом шаге рассматривается некоторое множество вариантов продолжения процесса поиска. В частности, «жадный» алгоритм на каждом шаге делает локально оптимальный выбор, который является лучшим в данный момент в надежде, что он приведет к оптимальному решению задачи в целом [7].

Алгоритм решения задачи планирования занятий, построенный на основе «жадной» стратегии, можно представить в виде следующей последовательности шагов.

1. Получить граф зависимостей понятий $H = (X', E')$ на основе полной графовой модели учебного курса $G^* = (X, U, E)$.

2. Определить множество вершин $M \subset X'$, где $M = \{x_i \in X' \colon \Gamma^{-1}(x_i) = \varnothing\}$, в которые не заходит ни одна дуга (они определяют возможные альтернативы выбора понятий при планировании).

3. Вычислить весовые коэффициенты для вершин $x_i \in M$, которые не имеют оценок v_i.

4. Выбрать из множества M вершину x_i с максимальным весом v_i, если таких вершин несколько, то из них выбрать вершину с минимальным индексом i.

5. Удалить вершину $x_i \in X'$ из графа H вместе с выходящими дугами $(x_i, x_j) \in E'$, где $x_j \in \Gamma(x_i)$, и поместить ее в результирующую укладку L.

6) Повторять шаги 2-5 до тех пор, пока граф H не пустой.

Пример 1. Покажем работу предложенного алгоритма на следующем примере. Пусть полная графовая модель $G^* = (X, U, E)$ учебной дисциплины имеет вид, представленный на рисунке 4, где сплошной линией изображены дуги из множества U, а штриховой – из множества E. Полученный граф зависимостей понятий $H = (X', E')$ формируется на множестве листьев $X' = \{x_4, x_5, x_6, x_8, x_9, x_{10}\}$ исходного графа G^* и содержит дуги $E' = \{(x_4, x_6), (x_4, x_8), (x_5, x_8), (x_6, x_9), (x_6, x_{10}), (x_8, x_9), (x_9, x_{10})\}$, как это показано на рисунке 5, где рядом с вершинами указаны значения весов v_i, оценивающих значимость соответствующих понятий дисциплины.

Итерационный процесс работы алгоритма, формирующего искомую укладку L, приведен в таблице 1. Значение критерия оптимальности, определяемого на основе функции забывания, для полученного решения составляет $F = R^* = 0.334289$. Процедура вычисления такой целевой функции использует полную графовую модель учебного курса и рассмотрена в работах [1, 2].

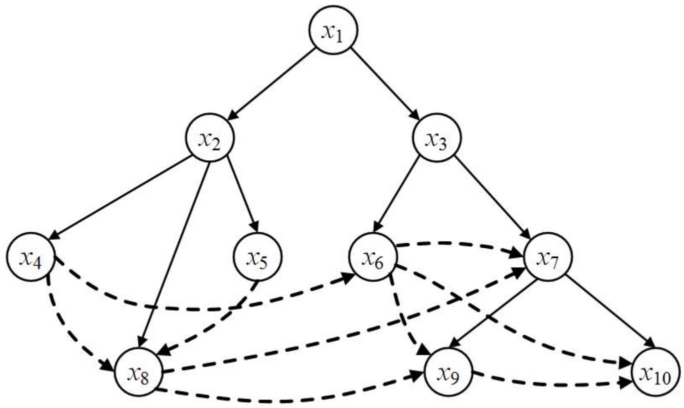

Рисунок 4 – Полная графовая модель G^*
Figure 4 - Full graph model G^*

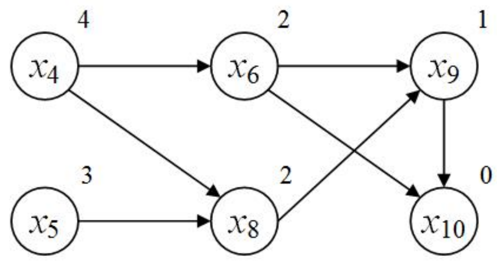

Рисунок 5 – Граф *H* с весами вершин
Figure 5 – Graph *H* with vertex marks

Таблица 1 – Процесс формирования укладки *L*
Table 1 – Layout graph creation *L*

Итерация алгоритма	Формирование укладки *L*	Множество вершин графа *H*	Множество *M* альтернатив	Выбор
1	-	$\{x_4, x_5, x_6, x_8, x_9, x_{10}\}$	$\{x_4, x_5\}$	x_4
2	(x_4)	$\{x_5, x_6, x_8, x_9, x_{10}\}$	$\{x_5, x_6\}$	x_5
3	(x_4, x_5)	$\{x_6, x_8, x_9, x_{10}\}$	$\{x_6, x_8\}$	x_6
4	(x_4, x_5, x_6)	$\{x_8, x_9, x_{10}\}$	$\{x_8, x_9\}$	x_8
5	(x_4, x_5, x_6, x_8)	$\{x_9, x_{10}\}$	$\{x_9\}$	x_9
6	$(x_4, x_5, x_6, x_8, x_9)$	$\{x_{10}\}$	$\{x_{10}\}$	x_{10}
результат	$(x_4, x_5, x_6, x_8, x_9, x_{10})$	$\varnothing$	$\varnothing$	-

Представленный пример является очень простым и позволяет выполнить полный перебор решений с целью оценки качества результата работы предложенного алгоритма. В таблице 2 показаны все возможные варианты корректных укладок L с соответствующими значениями целевой функции $F = R^*$. Нетрудно заметить, что в этом примере жадный алгоритм дает наилучший результат. Самое плохое решение соответствует третьему варианту укладки графа зависимостей понятий учебной дисциплины.

Таблица 2 – Возможные варианты укладок графа *H*
Table 2 – Possible layouts of graph *H*

Вариант решения	Укладка *L*	Критерий оптимальности	
		$F = R^*$	$F = \varkappa(L)$
1	$L_1 = (x_4, x_5, x_6, x_8, x_9, x_{10})$	0.334289	2
2	$L_2 = (x_4, x_5, x_8, x_6, x_9, x_{10})$	0.334289	3
3	$L_3 = (x_4, x_6, x_5, x_8, x_9, x_{10})$	0.304799	4
4	$L_4 = (x_5, x_4, x_6, x_8, x_9, x_{10})$	0.334289	3
5	$L_5 = (x_5, x_4, x_8, x_6, x_9, x_{10})$	0.334289	3

Необходимо отметить, что определение весовых коэффициентов k_i вершин $x_i \in X'$ графа H на основе числа достижимых вершин не является единственно возможны способом. В частности, если требуется, чтобы в расписании учебного курса было бы больше логически связанных фрагментов, т.е. таких подпоследовательностей запланированных занятий, в которых после освоения некоторого понятия сразу же следуют занятия по изучению зависимых от него понятий, то веса вершин можно вычислять как

$$v_i = (k_i + 1) \cdot c(j), \tag{3}$$

где $c(j)$ – коэффициент, зависящий от номера итерации j, который определяется по следующей формуле:

$$c(j) = 1 + d \cdot (j - 1). \tag{4}$$

Величина $c(j)$ изменяется в процессе выполнения итераций алгоритма в большую (при $d > 0$) или меньшую (при $d < 0$) сторону со скоростью, которая задается разностью d арифметической прогрессии.

Если $d > 0$, то приоритет v_i будет выше у тех понятий из множества M, которые будут доступны для выбора позже, т.е. на последующих итерациях алгоритма с большими номерами j. Это позволяет повысить логическую связность курса при планировании занятий, поскольку в формируемую последовательность L в первую очередь будут добавляться понятия, непосредственно связанные с самым последним изученным. При $d < 0$ наоборот приоритет повышается у понятий, доступных для выбора на более ранних итерациях. Это означает, что после изучения некоторого понятия сначала будут рассматриваться смежные, а только затем логически зависимые от него понятия. В случае $d = 0$ оценка важности изучаемых понятий равнозначна использованию весовых коэффициентов $v_i = k_i$.

Заметим, что добавление единицы к величине k_i в формуле (3) позволяет алгоритму выбирать в соответствии с используемой стратегией такие понятия (имеющие оценку $k_i = 0$), от которых не зависят никакие другие.

Пример 2. Покажем использование предложенного алгоритма для случая, когда веса вершин графа H, представленного на рисунке 5, вычисляются по формулам (3) и (4). Процесс выполнения итераций для случая $d = 2$ приведен в таблице 3.

Таблица 3 – Процесс формирования укладки *L*
Table 3 – Creation graph layout *L*

Итерация j и значение $c(j)$	Формирование укладки L	Множество M альтернатив	Вычисление весовых коэффициентов v_i	Выбор
$j = 1;\ c(1) = 1$	–	$\{x_4, x_5\}$	$v_4 = (4+1)\cdot 1 = 5;\ v_5 = (3+1)\cdot 1 = 4$	x_4
$j = 2;\ c(2) = 3$	(x_4)	$\{x_5, x_6\}$	$v_6 = (2+1)\cdot 3 = 9$	x_6
$j = 3;\ c(3) = 5$	(x_4, x_6)	$\{x_5\}$	–	x_5
$j = 4;\ c(4) = 7$	(x_4, x_6, x_5)	$\{x_8\}$	$v_8 = (2+1)\cdot 7 = 21$	x_8
$j = 5;\ c(5) = 9$	(x_4, x_6, x_5, x_8)	$\{x_9\}$	$v_9 = (1+1)\cdot 9 = 18$	x_9
$j = 6;\ c(5) = 11$	$(x_4, x_6, x_5, x_8, x_9)$	$\{x_{10}\}$	$v_{10} = (0+1)\cdot 11 = 11$	x_{10}
результат	$(x_4, x_6, x_5, x_8, x_9, x_{10})$	$\varnothing$	–	-

Нетрудно заметить, что решение, полученное во втором примере, является наихудшим с точки зрения критерия оптимальности $F = R^*$, построенного на основе функции забывания (таблица 2). Однако в действительности способ формирования весовых коэффициентов вершин графа H по формулам (3) и (4) предполагает использование другого критерия оптимальности, называемого числом соседней связности укладки L [4] и обеспечивающего непрерывность изложения логически связанных понятий [2].

В укладке $L(H) = (x_1, x_2, \ldots, x_n)$ вершины $x_i, x_{i+1} \in X'$ графа $H = (X', E')$ находятся в отношении соседней связности, если $(x_i, x_{i+1}) \in E'$. Число соседней связности укладки L определяется по формуле [4]:

$$\varkappa(L) = \sum_{i=1}^{n-1} \omega(x_i, x_{i+1}), \quad \text{где} \quad \omega(x_i, x_{i+1}) = \begin{cases} 1, \text{если } (x_i, x_{i+1}) \in E'; \\ 0, \qquad\qquad \text{иначе.} \end{cases}$$

Оно показывает количество пар соседних в укладке L вершин, связанных дугами графа H, т.е. находящихся в отношении соседней связности. В контексте решаемой задачи число соседней связности укладки L характеризует непрерывность изучения логически связанных понятий учебного курса и определяется как количество пар таких понятий, рассматриваемых последовательно друг за другом.

Если укладка $L = (x_1, x_2, \ldots, x_n)$ включает n вершин, то максимально возможное значение величины $\varkappa(L)$ равно n - 1. В этом случае все изучаемые понятия логически связаны, т.е. $(x_i, x_{i+1}) \in E'$ для всех $i = 1, 2, \ldots, n - 1$. Минимально возможное значение $\varkappa(L) = 0$. Возвращаясь к примеру 2, отметим, что полученное решение является наилучшим по критерию числа соседней связности $F = \varkappa(L)$. Значения этой целевой функции для всех возможных решений задачи также приведены в таблице 2.

Экспериментальные исследования

Для анализа работы описанных алгоритмов желательно использовать модели реальных учебных курсов, которые не всегда доступны. Поэтому в данной работе предлагается выполнить исследование разработанного алгоритма на основе специально сформированных тестовых исходных данных, моделирующих структуру и семантику учебных дисциплин.

Для генерации ориентированного дерева $G = (X, U)$, которое описывает иерархические связи элементов курса (разделов, подразделов и понятий), предлагается процедура последовательного добавления потомков узлам формируемого дерева на основе очереди вершин. Эта процедура строит описание дерева на основе случайного формирования потомков очередного узла, выбираемого из очереди. Входными данными являются две величины: *leaveCount* – общее число листьев дерева; *childCount* – максимальное число потомков одного узла. Результат – структура данных *root*, описывающая сформированное дерево. Процедура генерации включает следующие действия.

1. Создать корень дерева *root*.
2. Поместить *root* в очередь *queue*.
3. Задать текущее число листьев $l = 1$.
4. Выбрать вершину v из очереди *queue*.
5. Определить предельное число новых листьев $diff = leaveCount - l$.
6. Добавить *Rnd* прямых потомков вершине v, где $Rnd \in \{2, 3, \ldots, \min(diff, childCount)\}$ – целое случайное число.
7. Поместить все сформированные потомки в очередь *queue*.
8. Изменить текущее число листьев $l = l + Rnd - 1$.
9. Если $l < leaveCount$, то выполнить переход к п. 4.
10. Конец процедуры.

Заметим, что использование очереди *queue* обеспечивает для формируемого дерева преимущественный «рост в ширину» и делает его структуру более сбалансированной. Если вместо очереди использовать стек, то структура дерева заметно меняется из-за его «роста в глубину».

Используя полученное дерево G как исходные данные, на его листьях строится граф $H = (X', E')$, моделирующий логические зависимости понятий учебного курса. Поскольку этот граф является ориентированным и бесконтурным, его генерацию также можно выполнить на основе достаточно простой вероятностной процедуры. Для этого узлы $x_i \in X'$ просматриваются в произвольном порядке и для каждого из них случайным образом формируются дуги, направленные к еще не просмотренным узлам. Для управления процессом генерации используется величина *prob* - вероятность соединения очередной вершины с каждой последующей.

Просматривая вершины в порядке возрастания индексов, при $prob = 1$ получаем, что текущая вершина x_i $(i = 1, 2, \ldots, n - 1)$ должна быть связана дугой (x_i, x_j) с каждой последующей вершиной x_j $(j = i + 1, i + 2, \ldots, n)$. Тогда в целом граф H с n вершинами может иметь не более m дуг, где $m = (n - 1) + (n - 2) + \ldots + 1 + 0 = (n - 1)\, n / 2$. Если задать $prob = 0$, то множество дуг графа H будет пустым. При условии $0 < prob < 1$ количество дуг формируемого графа будет случайным со средним значением $prob\,(n - 1)\, n / 2$. Величина *prob* удобна тем, что не зависит от числа вершин n.

Процедура генерации дуг графа H включает следующие действия.

1. Присвоить $i = 1$.
2. Определить предел числа дуг, исходящих из вершины x_i:
$$edgeCount = \max(1, leaveCount - i - 1).$$
3. Присвоить $j = i + 1$.
4. Сформировать дугу (x_i, x_j) с вероятностью *prob*.
5. Если $j < n$, то присвоить $j = j + 1$ и выполнить переход к п. 4.
6. Если $i < n - 1$, то присвоить $i = i + 1$ и выполнить переход к п. 3.
7. Конец процедуры.

При проведении эксперимента использованы программные модули, разработанные на языке C++ и реализующие генерацию тестовых исходных данных, поиск точного решения методом полного перебора, получение приближенного решения жадным алгоритмом, а также вычисление функции забывания и числа соседней связности в соответствующих критериях.

Для оценки качества результатов сравнивались значения целевой функции для решений, полученных методом полного перебора и жадным алгоритмом, и вычислялась относительная погрешность приближенных результатов. Учитывая экспоненциальную сложность поиска точного решения, такое сравнение выполнено только для малых размерностей задачи, где модели учебных курсов включают не более $n = 21$ понятий (листьев дерева G). Поскольку разные учебные дисциплины могут содержать различное число логических зависимостей, при генерации тестовых данных варьировалась величина *prob*, которая характеризует их количество.

В целом при проведении эксперимента число понятий n принимало значения 12, 14, 18, 21, а величина *prob* изменялась в пределах от 0,3 до 0,5. При вычислении целевой функции R^* предполагается, что занятия проводятся раз в неделю, а число понятий, изучаемых на одном занятии, колеблется в пределах от 2 до 3. Для каждого сочетания параметров n и *prob* выполнено по 3 эксперимента с разными данными. Всего было проведено 36 опытов с использованием процессора Intel Core i3-2120.

Значения функции забывания и числа соседней связности при $n = 21$ приведены на рисунке 6, из которого видно, что наименьшее отклонение от оптимального значения критерий на основе функции забывания R^* достигает при $d = 0$, а критерий соседней связности $\varkappa(L)$ - при $d = 2$, где d - параметр алгоритма планирования, управляющий вычислением весовых коэффициентов вершин графовой модели.

Значения относительной погрешности в процентах, полученные для функции забывания (при $d = 0$) и для числа соседней связности (при $d = 2$), приведены в таблице 4. Анализ показывает, что с ростом связности понятий курса (что определяется увеличением значения *prob*) качество решений улучшается. Так для *prob* = 0,5 решение задачи по критерию максимума числа соседней связности совпадает с точным во всех случаях, а по критерию на основе функции забывания дает небольшую погрешность, которая заметно более существенная при меньшей связности.

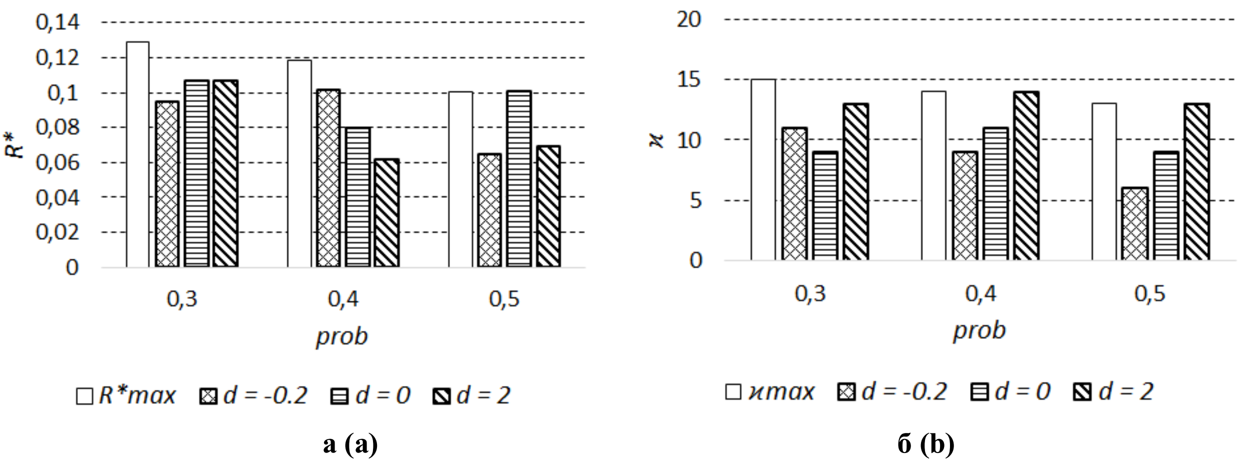

Рисунок 6 – Результаты планирования по критериям:
а – на основе функции забывания; б – числа соседней связности
Figure 6 – Results of planning by criteria:
a – based on forgetting function; b – based on adjacent connection mark

Переходя к оценке скорости работы алгоритма планирования, предварительно отметим, что при полном переборе необходимо вычислять значение целевой функции каждого варианта решения, что существенно влияет на результаты. Так при $n = 21$ и *prob* = 0,5 для нахождения оптимального решения по критерию соседней связности потребовалось в среднем

280 мс, а по критерию на основе функции забывания – 1330 мс. Время работы жадного алгоритма в этих же условиях составляет около 1 мс.

Таблица 4 – Результаты экспериментов

Table 4 – Experiment results

Значение n	Погрешность для критерия $F = R^*$			Погрешность для критерия $F = \varkappa(L)$		
	prob = 0,3	*prob* = 0,4	*prob* = 0,5	*prob* = 0,3	*prob* = 0,4	*prob* = 0,5
12	24,49	28,04	0	0	12,5	0
14	15,45	13,6	2,6	12,5	0	0
18	19,57	0	0	0	8,33	0
21	17,04	32,52	0	13,33	0	0

Для больших значений n = 100, 150 и 200 на рисунке 7 представлен график изменения времени выполнения программы, реализующей жадный алгоритм, при увеличении величины *prob*. Анализ этой зависимости показывает, что с ростом числа логических связей почти линейно растет и время работы жадного алгоритма. Это объясняется тем, что основу алгоритма составляет обход графа в глубину для нахождения оценок v_i, а с ростом числа дуг растет и время обхода вершин графа. Учитывая то, что реальные учебные курсы редко содержат число понятий, превышающее n = 200, время работы программы планирования занятий будет незначительным, что позволит использовать ее многократно для поиска нужного варианта решения.

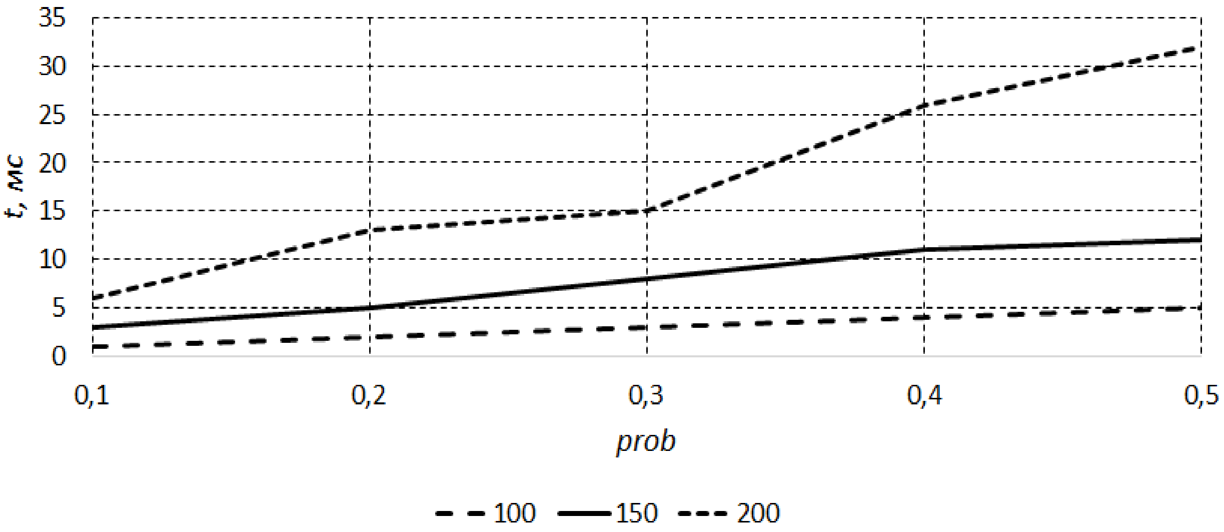

Рисунок 7 – Зависимость среднего времени работы программы планирования занятий от уровня логической связанности понятий учебного курса

Figure 7 - Dependence of average working time of planning program from the level of logical connectedness of training course concepts

Заключение

В статье предложен алгоритм нахождения порядка изучения понятий (тем) учебного курса, основанный на использовании жадной стратегии поиска, и проведены экспериментальные исследования программы, реализующий этот алгоритм. Эксперименты показывают, что алгоритм дает весьма качественные результаты, близкие к оптимальным, если требуется достигнуть максимума логической связности при изложении понятий дисциплины. При использовании критерия на основе функции забывания результаты немного хуже, причем относительное отклонение критерия от оптимального значения убывает с ростом логической связанности понятий курса от нескольких десятков до нескольких единиц процентов. Однако временные затраты при использовании жадного алгоритма не значительны в сравнении с временем поиска оптимального решения методом полного перебора.

В дальнейшем предполагается внедрить разработанный алгоритм в систему моделирования учебных курсов [1], что позволит на основе описания структуры курса и логических зависимостей его понятий (тем), автоматически определить последовательность изучения разделов. Если принять во внимание ряд дополнительных параметров, учитывающих количество и время проведения занятий, аудиторный фонд, преподавательский состав и т.п., то достаточно легко можно реализовать решение задач составления расписания и оценки его эффективности по различным критериям [2].

Библиографический список

1. **Скворцов С. В., Митрошин А. А., Кузнецов А. А., Скворцова Т. С.** Моделирование и оптимизация процессов освоения учебных дисциплин // Вестник Рязанского государственного радиотехнического университета. 2019. № 69. С. 110-122.

2. **Скворцов С. В., Митрошин А. А., Кузнецов А. А.** Критерии качества изложения учебных дисциплин // Интеллектуальные и информационные системы (Интеллект-2019): труды Всероссийской научно-технической конференции. Тула: Изд-во ТулГУ, 2019. С. 88-94.

3. **Wozniak P. A., Gorzelanczyk E. J., Murakowski J. A.** Two components of longterm memory // Acta Neurobiologiae Experimentalis. 1995, vol. 55, no. 4, pp. 301-305.

4. **Корячко В. П.** Конструирование микропроцессорных систем контроля радиоэлектронной аппаратуры. М.: Радио и связь, 1987. 160 с.

5. **Вентцель Е.С.** Исследование операций. Задачи, принципы методология. М.: Дрофа, 2004. 208 с.

6. **Акулич И.Л.** Математическое программирование в примерах и задачах. М.: Высшая школа, 1986. 319 с.

7. **Кормен Т. Х. и др.** Алгоритмы: построение и анализ. М.: ООО «И.Д. Вильямс», 2013. 1328 с.

UDC 004.942

ACADEMIC DISCIPLINE PLANNING TASK SOLUTION BASED ON A GREEDY ALGORITHM

S. V. Skvortsov, Dr. Sc. (Tech.), full professor, RSREU, full professor, Academy of the FPS of Russia, Ryazan, Russia;
orcid.org/0000-0001-9495-4953, e-mail: s.v.skvor@gmail.com
A. A. Kuznetsov, master student, RSREU, Ryazan, Russia;
orcid.org/0000-0003-1421-0941, e-mail: alexander.kuznetsov996@gmail.com
T. S. Skvortsova, Ph.D. (Tech.), lecturer, Academy of the FPS of Russia, Ryazan, Russia;
orcid.org/0000-0001-7504-5973, e-mail: t.s.skvortsova@yandex.ru

*The problem of educational process planning based on the model that takes into account the structure of the academic discipline and its semantics, presented in terms of concepts and logical dependencies between them is considered. **The aim** is to develop and study an algorithm for determining the order of study of topics (concepts) of academic discipline based on greedy strategy. Algorithm modifications to obtain solutions according to quality criteria based on forgetting function, allowing evaluating students' knowledge, and adjacent connectivity of the studied concepts, characterizing their logical relationship are proposed. The experimental studies results of the developed algorithm with the aim of analyzing performance and quality of the resulting solutions are presented.*

Key words: *educational process planning, academic discipline model, greedy algorithm, forgetting function, adjacent connectivity criterion.*

DOI: 10.21667/1995-4565-2020-72-71-82

References

1. **Skvortsov S. V., Mitroshin A. A., Kuznetsov A. A., Skvortsova T. S.** Modelirovanie i optimizacija processov osvoenija uchebnyh discipline. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2019, no. 69, pp. 110-122. (in Russian).

2. **Skvortsov S. V., Mitroshin A. A., Kuznetsov A. A.** Kriterii kachestva izlozhenija uchebnyh disciplin. *Intellektual'nye i informacionnye sistemy (Intellekt-2019): trudy Vserossijskoj nauchno-tehnicheskoj konferencii. Tula: Izd-vo TulGU.* 2019, pp. 88-94. (in Russian).

3. **Wozniak P. A., Gorzelanczyk E. J., Murakowski J. A.** Two components of longterm memory. *Acta Neurobiologiae Experimentalis*. 1995, vol. 55, no. 4, pp. 301-305.

4. **Koryachko V. P.** *Konstruirovanie mikroprocessornyh sistem kontrolja radiojelektronnoj apparatury* (Design of microprocessor control systems for electronic equipment). Moscow, Radio i svjaz'. 1987, 160 p. (in Russian).

5. **Ventcel' E.S.** *Issledovanie operacij. Zadachi, principy metodologija* (Operations research. Tasks, principles, methodology), Moscow, Drofa. 2004, 208 p. (in Russian).

6. **Akulich I.L.** *Matematicheskoe programmirovanie v primerah i zadachah* (Mathematical programming in examples and tasks), Moscow, Vysshaja shkola, 1986. 319 p. (in Russian).

7. **Kormen T. H. i dr.** *Algoritmy: postroenie i analiz* (Algorithms: construction and analysis). Moscow, OOO «I.D. Vil'jams», 2013. 1328 p. (in Russian).