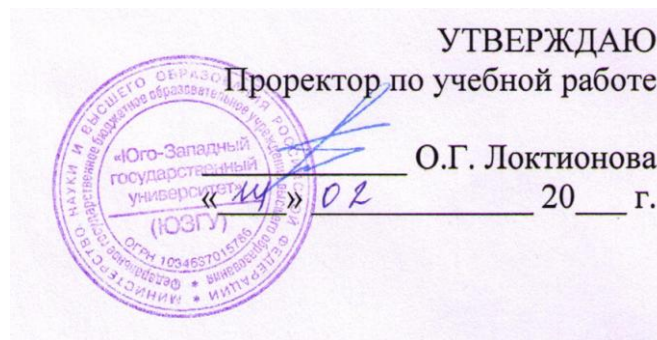


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 16.06.2020
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eab0f75e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии



ПОИСКОВЫЕ АЛГОРИТМЫ В ЭКСПЕРТНЫХ СИСТЕМАХ:

методические указания к лабораторным занятиям для магистров
направления подготовки 02.03.03 Математическое обеспечение и
администрирование информационных систем

Курск 20__

УДК 004

Составитель: Е.А. Титенко

Рецензент

Кандидат технических наук *А. В.С. Киселев*

Поисковые алгоритмы в экспертных системах:
методические указания к лабораторным занятиям / Минобрнауки
России, Юго-Зап. гос. ун-т; сост.: Е.А. Титенко. - Курск, 2022. - 17
с. - Библиогр.: с. 17.

Приводится описание систематических алгоритмов поиска, используемых в экспертных системах. Приведены теоретические положения систематического обхода пространства состояний, примеры и задания.

Методические рекомендации предназначены для студентов, обучающихся по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем очной и заочной форм обучения.

Текст печатается в авторской редакции.

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 0,99 п.л. . Уч.-изд. л. 0,89. Тираж 100 экз. Заказ.
Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Цель лабораторной работы: изучение основных принципов систематических поисково-переборных алгоритмов, назначение информационных структур FIFO и LIFO для организации обхода состояний, определение количественных характеристик графа состояний.

1 Основные понятия

1.1 Основные понятия теории графов для организации поиска

Специфика задач принятия решений, в том числе задач, причисляемым к интеллектуальным, состоит в том, что в них с необходимостью выделяются два этапа:

1. этап генерации полного множества решений;
2. этап выбора подмножества оптимальных решений по заданным критериям.

Этап генерации вариантов решений рассматривается как задача выдвижения гипотезы о целевом состоянии и поиске в пространстве возможных состояний соответствующих путей, соединяющих исходные состояния с заключительными и вычислением в процессе построения графа состояний количественных поисковых оценок. Исходная неопределенность в применении правил преобразования обуславливает, в общем случае, недетерминированность процесса поиска и подтверждения гипотезы. Устранение возникающей неопределенности возможно на основе адекватного математического аппарата – исчислительных систем, описывающих недетерминированные процессы в виде ветвящихся конструктивных процессов. Таким образом, процедура поиска, носящая недетерминированный характер исполнения, сводится к генерации возможных состояний из исходных и их постепенной модификации на основе правил преобразования в заключительные состояния.

Задача поиска формально определяется следующим образом:

$$\{S, F, Z, P, Q\}, \quad (1)$$

где S – начальное состояние, F – множество конечных состояний, Z – множество заключительных состояний ($Z \subseteq F$), P – множество правил преобразования вида (2), носящих недетерминированный

смысл исполнения, Q – множество ограничений на правила преобразования.

$$A \rightarrow B, \quad (2)$$

где A – заменяемое состояние, B – заменяющее состояние.

Теоретической основой описания пространства поиска является теория графов, позволяющая анализировать как структуру и сложность самой решаемой задачи, так и сложность процесса поиска решения. Граф состоит из множества вершин и дуг, соединяющих пары вершин. В модели пространства состояний решаемой задачи вершины графа соответствуют состояниям процесса решения, а дуги описывают переходы между состояниями. Дуги графа могут быть размеченными или ориентированными. Метка дуги используется для задания именованного отношения или веса дуги и определяется исходя из семантики решаемой задачи. Ориентированные дуги задают направление перехода между вершинами.

Путь на графе – это последовательность дуг, соединяющих соседние вершины. Путь представляется последовательностью вершин. Корневой граф – это граф с единственной вершиной, от которой существует путь к любой вершине. Такая вершина называется корнем. Дерево – это корневой граф, на котором для любых двух вершин существует не более одного пути между ними. Поиск на графе характеризует решение задачи как процесс нахождения пути решения, т.е. цепочки состояний, идущих от исходного к целевому.

Отличительная особенность организации поиска в пространстве состояний определяется тем, структура графа зависит от коэффициентов ветвления и сужения. Коэффициент ветвления текущего состояния показывает, сколько состояний-потомков есть у текущего состояния, а значит сколько потенциальных путей проходит через данную вершину на данной стадии завершения. Коэффициент сужения текущего состояния показывает, сколько состояний-родителей есть у текущего состояния. Другими словами, сколько путей сходится в текущей вершине на данной стадии завершения. Наличие ненулевых коэффициентов сужения приводит к тому, что целевые состояния могут быть достигнуты разными путями. Отсюда возникает задача поиска оптимального пути. Как правило, в качестве оптимального рассматривается путь

наименьшей длины, иногда – путь, проходящий через заданные промежуточные состояния. Кроме того, множественные пути к состоянию могут приводить к образованию циклов.

1.2 Количественные характеристики поиска на графах

Решение задачи путем поиска имеет следующие количественные характеристики графа состояний:

1. глубина поиска
 - а) кратчайший путь в графе
 - б) путь в графе максимальной длины
2. ширина поиска
 - а) максимальный коэффициент ветвления
 - б) средний коэффициент ветвления по графу
3. общее количество состояний в графе
4. коэффициент сужения
 - б) средний коэффициент сужения по графу.

В настоящее время для оценки графа состояний используется такая характеристика как пространственная глубина поиска, под которой понимается количество потомков между двумя вершинами. Для количественной оценки алгоритмов поиска целесообразнее использовать такую характеристику как временная глубина поиска. Под ней понимается максимальное количество уровней в графе между начальным и конечным состояниями. В зависимости от способа обхода состояний графа пространственная и временная глубина могут не совпадать. Несовпадение определяется соотношением количества поддерживаемых активных процессов поиска и количеством сгенерированных состояний-потомков у текущей вершины. В случае, если каждый шаг поиска связан с рассмотрением единственной альтернативы и имеется M состояний потомков, то временная глубина будет в M раз больше пространственной, хотя данные состояния-потомки принадлежат одному уровню.

Коэффициент ветвления графа определяется выражением (3)

$$KV = \frac{\sum st' + \sum f}{\sum f} = 1 + \frac{\sum st'}{\sum f}, \quad (3)$$

где st' – вершина, имеющая более одного состояния-потомка, f – конечная вершина графа. Например, пусть в графе у каждой вершины имеется B потомков и граф содержит n уровней. Тогда конечные вершины располагаются на последнем n -ом уровне, а их количество составляет B^n вершин. Общее количество вершин в графе определяется суммой вершин по всем уровням и определяется выражением (4)

$$B + B^2 + \dots + B^{n-2} + B^{n-1} + B^n \quad (4).$$

Коэффициент ветвления для такого графа определяется как (5)

$$KV = \frac{B + B^2 + \dots + B^{n-2} + B^{n-1} + B^n}{B^n} = 1 + \frac{1}{B} + \frac{1}{B^2} + \dots + \frac{1}{B^{n-2}} + \frac{1}{B^{n-1}} \quad (5).$$

Коэффициент сужения графа показывает количество путей, имеющих общими фиксированные целевые вершины. Это означает, что данные пути на разных стадиях завершения поиска характеризуются различными фрагментами, а на последнем участке имеют общую вершину.

Текущий коэффициент сужения определяется как (6)

$$KS_i = \sum sp, \quad (6)$$

где sp – количество состояний-предков у i -ой вершины.

Общий коэффициент сужения – это сумма текущих коэффициентов сужения по всем состояниям графа (7)

$$KS = \sum KS_j, \quad (7)$$

где KS_j – j -ый коэффициент сужения, $j=1-n$ – количество состояний в графе.

Коэффициент сужения определяет «недостроенные» пути и «недополученные» в графе состояния.

1.3 Реализация поиска на графах

Поисковые процедуры подразделяются:

1. по выбору начального и конечного состояния
 - а) поиск от данных (прямой поиск)
 - б) поиск от цели (обратный поиск)
2. по направлению обхода
 - а) в глубину

б) в ширину.

При поиске от данных процесс решения задачи начинается от заданного начального состояния и состоит в применении правил преобразования состояний задачи, которые, в свою очередь, используются для генерации новых состояний. Этот процесс продолжается до тех пор, пока не будет достигнуто целевое состояние. Возможен и другой подход. Целевое состояние рассматривается как потенциально достижимое, но путь к нему неизвестен. Для восстановления пути от целевого состояния к начальному состоянию анализируются допустимые правила преобразования, ведущие к цели. Эти новые состояния становятся новыми подцелями поиска. Поиск ведется в обратном направлении от достигнутых подцелей до достижения начального состояния. Таким образом, определяется путь от цели к исходным данным. Иногда поиск от цели, восстанавливающий путь в обратном порядке называют обратным поиском.

При поиске в глубину в качестве текущей рассматриваемой вершины в графе берется первая вершина-потомок и осуществляется переход на следующий уровень. При поиске в ширину для текущей вершины рассматриваются все его вершины-потомки, поэтому переход на следующий уровень осуществляется после генерирования и просмотра всех потомков текущего уровня.

В общем случае, при решении задач путем поиска требуется найти путь от начального состояния к целевому на графе пространства состояний. Последовательность дуг этого пути соответствует упорядоченной последовательности этапов решения задачи. Если в поисковой процедуре имеется некоторый прогностический механизм предсказания пути (некий оракул), то поисковые действия заменяются на безошибочное движение к цели с запоминанием пути решения. Вместе с тем для большинства задач таких механизмов предсказания не существует. В процедуре приходится рассматривать различные пути до тех пор, пока не будет достигнута цель. Поиск с возвратами (backtracking) – это метод систематической проверки различных путей в пространстве состояний.

2 Примеры реализации систематических алгоритмов

Алгоритм обхода в глубину

Алгоритм поиска в глубину с возвратами запускается из начального состояния и следует по некоторому пути до тех пор, пока не достигнет целевого состояния или тупикового. Если цель достигнута, поиск завершается, и в качестве решения задачи возвращается путь к цели. Если же поиск привел в тупиковую вершину, то алгоритм возвращается в ближайшую из пройденных вершин и исследует ее вершины-потомки, а затем спускается по одной из ветвей в графе, ведущих к новым конечным состояниям.

Обход в глубину реализуется последовательной упорядоченной генерацией текущих потомков. При этом из списка новых состояний будет браться первое сгенерированное состояние, находящееся на следующем уровне.

В данном алгоритме учитывается возможность повторного появления ранее рассмотренных состояний в текущем пути, т.е. коэффициент сужения в графе принимается не равным 0. Для обхода в глубину будут использоваться такие списки как

- 1) SL (State List) - список текущих состояний (текущий путь);
- 2) NSL (New State List) - список открытых, но не просмотренных состояний;
- 3) DL (Dead List) - список тупиковых (просмотренных) состояний.

Также в алгоритме применяются следующие операции:

- 1) FIRST (...) - взятие первого элемента списка;
- 2) DEL (...) – удаление первого элемента списка;
- 3) → - добавление элемента.

Работа алгоритма начинается с текущего состояния – CS.

Блок-схема алгоритма поиска в глубину до первой цели представлена на рис.1.

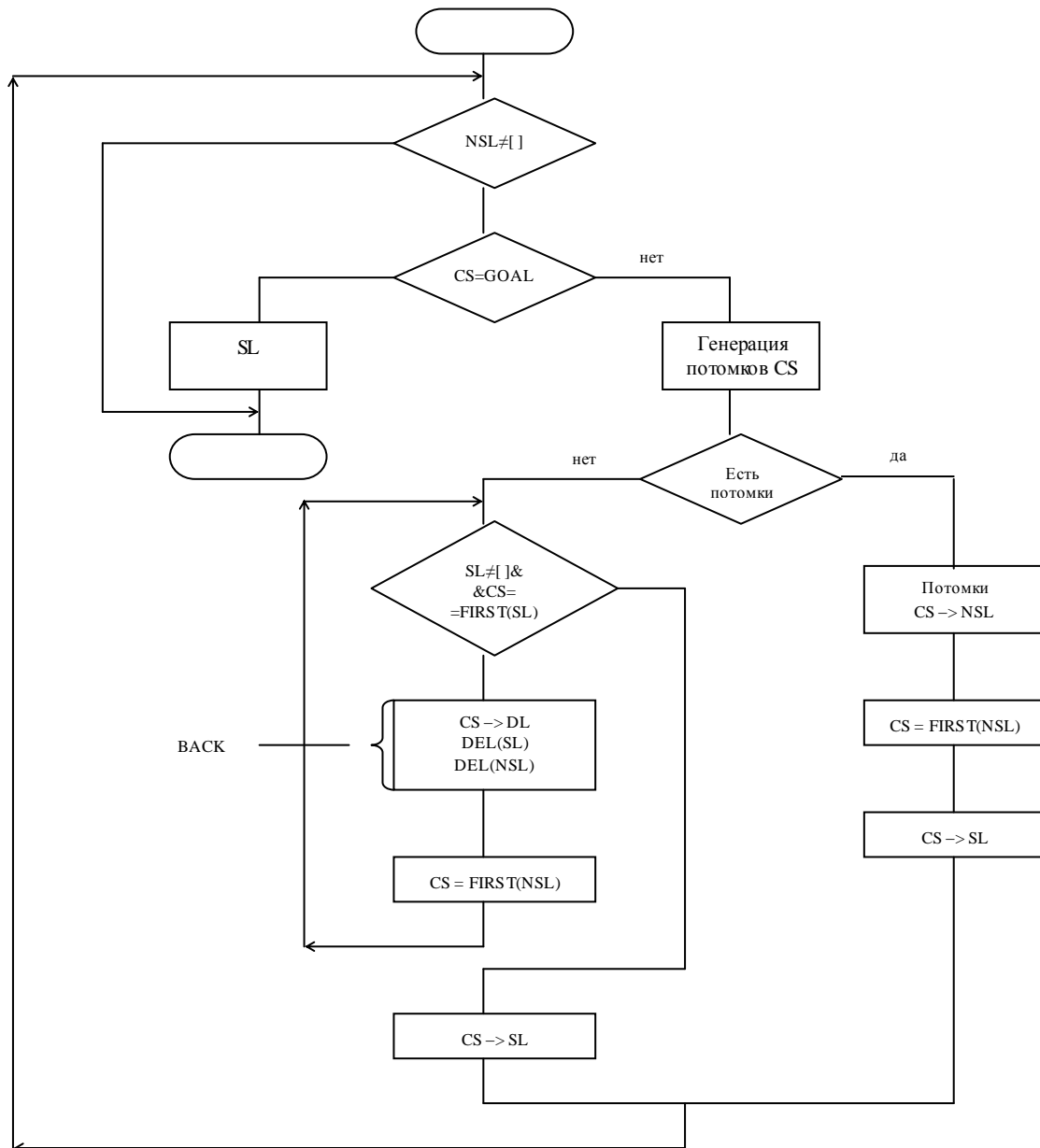


Рис.1 - Блок-схема алгоритма поиска в глубину

В алгоритме выделяются 2 блока операторов, связанных с прямым движением (FORWARD) в глубину пространства, при этом длина списка SL будет увеличиваться, и возвратным движением к корню графа (BACK) с уменьшением SL.

В табл. 1 описана пошаговая работа алгоритма в глубину до состояния-цели N в графе состояний, имеющем вид на рис.2.

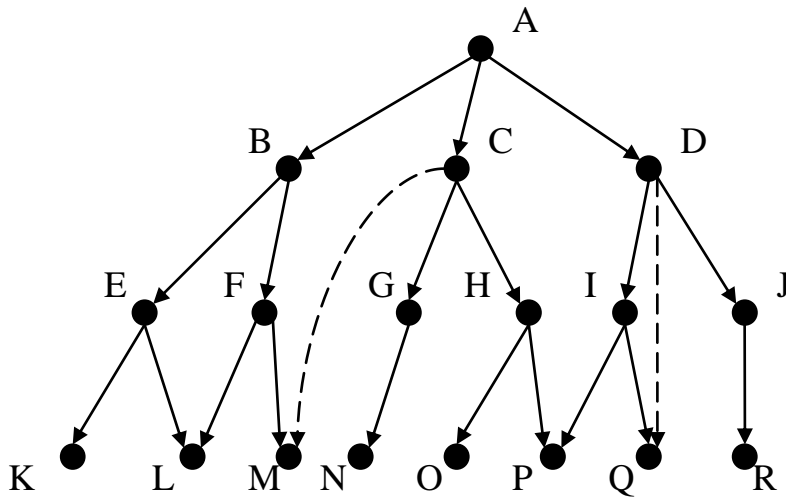


Рис.2 Граф состояний

Таблица 1 – Трассировка состояний в алгоритме поиска в глубину

№	CS	SL	NSL	DL	Направл.
0	A	A	A		Forward
1	B	BA	BCDA		
2	E	EBA	EFBCDA		
3	K	KEBA	KLEFBCDA		Back
4	<u>K</u>	<u>KEBA</u>	<u>KLEFBCDA</u>		
5	L	LEBA	LEFBCDA	K	Forward
6	<u>L</u>	<u>LEBA</u>	<u>LEFBCDA</u>	K	Back
7	<u>E</u>	<u>EBA</u>	<u>EFBCDA</u>	KL	
8	F	FBA	FBCDA	KLE	Forward
9	M	MFBA	MFBCDA	KLE	
10	<u>M</u>	<u>MFBA</u>	<u>MFBCDA</u>	KLE	
11	<u>F</u>	<u>FBA</u>	<u>FBCDA</u>	MKLE	Back
12	<u>B</u>	<u>BA</u>	<u>BCDA</u>	FMKLE	
13	C	CA	CDA	BFMKLE	Forward
14	G	GCA	GHCA	BFMKLE	
15	N	NGCA	NGHCA	BFMKLE	
16	<u>N</u>	<u>NGCA</u>	<u>NGHCA</u>	BFMKLE	Back
17	<u>G</u>	<u>GCA</u>	<u>GHCA</u>	NBFMKLE	
18	H	HCA	HCA	GNBFMKLE	Forward
19	O	OHCA	OPHCA	GNBFMKLE	
20	<u>O</u>	<u>OHCA</u>	<u>OPHCA</u>	GNBFMKLE	
21	P	PHCA	PHCA	GNBFMKLE	Forward
22	<u>P</u>	<u>PHCA</u>	<u>PHCA</u>	GNBFMKLE	Back
23	<u>H</u>	<u>HCA</u>	<u>HCA</u>	PGNBFMKLE	
24	<u>C</u>	<u>CA</u>	<u>CDA</u>	HPGNBFMKLE	
25	D	DA	DA	CHPGNBFMKLE	Forward
26	I	IDA	IJ DA	CHPGNBFMKLE	

Алгоритм обхода в ширину

Поиск в ширину исследует производство состояний последовательно по уровням. Это значит, что в пределах текущего уровня для всех его состояний генерируются всевозможные состояния-потомки. Условием перехода на следующий уровень в пространстве состояний, является просмотр потомков текущего уровня: на втором уровне просматриваются вершины В, С, D (рис.2), а затем осуществляется переход на третий уровень графа.

Для организации поиска в ширину необходимо использовать 2 списка состояний:

1. 1) OPEN – список сгенерированных, но не просмотренных вершин, является эквивалентом списка NSL при поиске в глубину;
2. 2) CLOSED – список просмотренных вершин, является объединением списков SL и DL.

При поиске в ширину для текущей вершины генерируются все дочерние состояния. Эти состояния записываются в список OPEN, который организовывается как очередь (FIFO – FIRST IN FIRST OUT): дочерние состояния добавляются в список справа по указателю ХВОСТ, а удаляются – слева по указателю ГОЛОВА. Таким образом, новые состояния находятся в списке дольше, чем предшествующие, чем обеспечивается перебор вершин в пределах текущего уровня, т.е. поиск в ширину. Поскольку при поиске в ширину состояния графа рассматриваются по уровням, то первыми находятся более короткие пути в графе. В противоположность данному поиску, поиск в глубину эффективен для графов с высокой степенью нерегулярной связности между уровнями, что облегчает проникновение в «глубины» пространства. Например, для целевой вершины М поиск в глубину будет результативным за счет нерегулярной связи С-М и потребует просмотра 3 вершин: А,С,М. Поиск в ширину для данной цели потребует просмотра 13 вершин: А,В,С,Д,Е,Ф,Г,Н,І,Ж,К,Л,М.

Работа алгоритма начинается с текущего состояния – СS.

Блок-схема алгоритма поиска в ширину до первой цели представлена на рис.3.

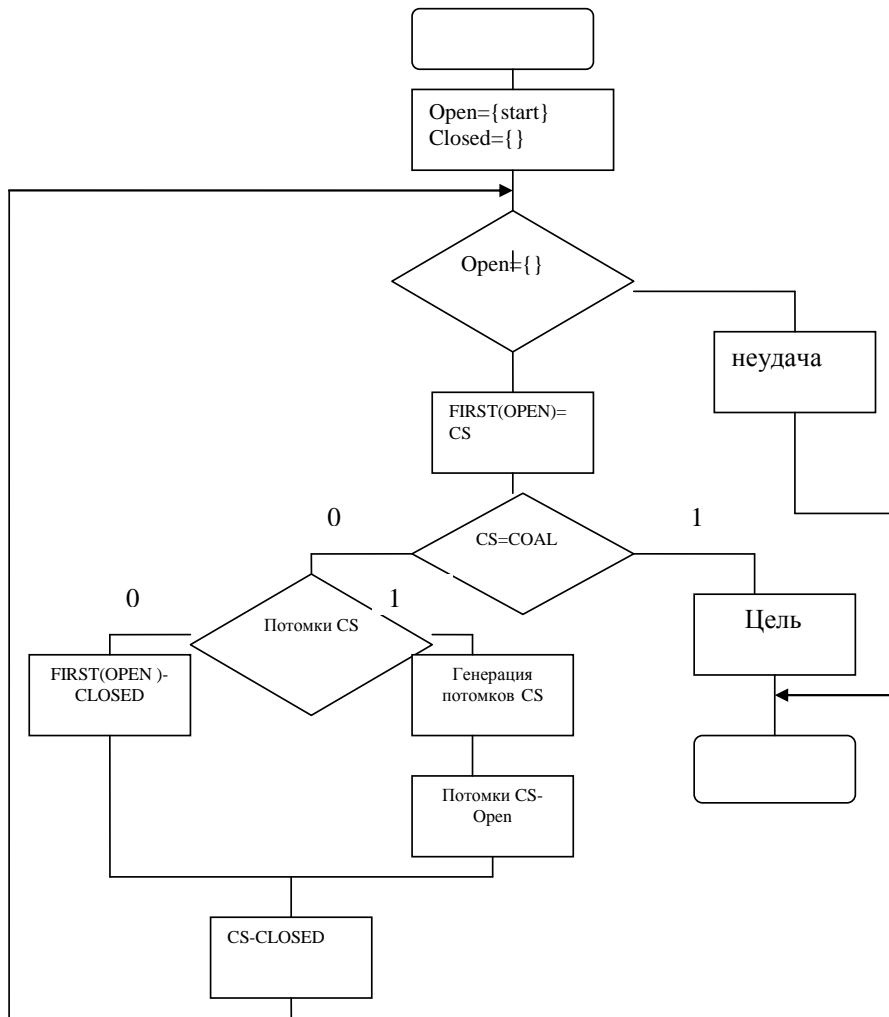


Рис.3 - Алгоритм поиска в ширину.

В табл. 2 описана пошаговая работа алгоритма поиска в ширину до состояния-цели N в графе состояний, имеющем вид на рис.2.

Таблица 2 – Трассировка состояний в алгоритме поиска в ширину

CS	OPEN	CLOSED
A	BCD	A
B	CDEF	BA
C	DEFGH	CBA
D	EFGHIJ	DCBA
E	FGHIJKL	EDCBA
F	GHIJKLM	FEDCBA
G	HIJKLMN	GFEDCBA
H	IJKLMNOP	HGFEDCBA
I	JKLMNOPQ	IHGFEDCBA
J	KLMNOPQR	JHGFEDCBA
K	LMNOPQR	KJIHGFEDCBA
L	MNOPQR	LKJIHGFEDCBA
M	NOPQR	MLKJIHGFEDCBA

При поиске в ширину для формирования пути к найденной цели необходимо, чтобы каждый элемент списка имел составную структуру, которая позволит восстановить путь путем просмотра подчиненных отношений.

Структура элемента списка показана на рис.4.

P	R	U	K	ST
---	---	---	---	----

Рис.4 - Структура элемента при поиске в ширину где P –текущий потомок,R - родитель,U- номер уровня,K- коэффициент ветвления,ST – статус: V – текущий потомок,Z заключительный потомок.

Такая конструкция позволит, просматривая списки OPEN, CLOSED, восстановить путь.

Модификации алгоритмов поиска

В общем случае, систематический поиск в глубину или ширину ограничен по времени необходимостью просмотра всех вершин графа с учетом проверки на возможное зацикливание процессов поиска. Данное обстоятельство не позволяет использовать данные алгоритмы в задачах большой размерности напрямую, так как затраты времени представляют экспоненциальную функцию. Иногда систематические алгоритмы поиска называют «слепыми», поскольку в них в той или иной последовательности просматриваются все вершины-потомки от текущей.

Одним из рациональных подходов к уменьшению времени поиска является систематический просмотр части графа поиска. Это означает, что поиск ведется до некоторого уровня N. По достижении заданного уровня даже при наличии потомков поиск прекращается. Такая модификация применяется, прежде всего, в алгоритме поиска в глубину и носит название ограниченный поиск глубины N. В этом случае при достижении глубины N осуществляется переход к следующей вершине, т.е. горизонтальная развертка графа поиска во времени. Также возможен ограниченный

поиск в ширину, при котором последовательно по уровням просматриваются N уровней графа от корня.

Другим подходом к уменьшению времени поиска является систематический просмотр части графа поиска с переменной величиной заглубления. Это означает, что для первой вершины-потомка величина заглубления равна 1, и от корневой вершины граф просматривается на данное заглубление. Для второй вершины-потомка граф будет просматриваться на заглубление, равное 2. Обобщая систематический неполный поиск, для текущей вершины глубина просмотра определяется увеличивающейся величиной заглубления.

3 Индивидуальное задание студента

Входные данные: определяются вариантом таблицы состояний и представляют собой множество правил преобразования состояний, направлением поиска, видом алгоритма поиска и целевыми состояниями.

Выходные данные: визуальное представление графа поиска, множество путей от начальной к целевой вершинам, количественные характеристики графа поиска: общее количество вершин, средний коэффициент ветвления, максимальный коэффициент ветвления, путь кратчайшей длины, путь максимальной длины.

Система правил					Цел. сост.	Алг. обхода	
N	Кол-во правил	Тек. коэф. ветвления	Кол-во сост. сужения	Тек. коэф. сужения		Направ.	Нач/цел
1	12	3	0	0	1	В глубину	А
2	15	2	1	2	2	В ширину	В
3	15	2-3	3	3	3	В глубину	А
4	12	2	0	0	4	Итерац.	В
5	15	1-3	2	2	4	В ширину	А
6	16	3	3	3	3	В	В

						глубину	
7	15	2	0	0	2	В ширину	А
8	20	1-3	1	2	1	В глубину	В
9	16	2	0	0	2	В глубин	А
10	12	2	2	2	3	В ширину	В
11	15	3	1	3	4	В глубину	А
12	14	3	2	0	3	В ширину	В
13	15	2	0	0	2	Итерац.	А
14	20	2	2	2	1	В глубину	В
15	15	2	1	2	2	В ширину	А
16	20	2	3	0	3	В глубину	В
17	15	3	2	3	4	В глубину	А
18	20	2	3	0	3	В ширину	В
19	15	2	0	0	2	В ширину	А
20	20	2	1	2	12	В глубину	В
21	15	3	3	0	3	В ширину	А
22	20	2	1	2	3	В глубину	В
23	15	3	2	1	2	В глубину	А
24	15	3	0	0	1	В глубину	В
25	20	2	3	1	2	В ширину	А

где А – обход от данных, В – обход от цели.

Во всех вариантах правила задаются со структурой $* \rightarrow *$, где $*$ -состояние, обозначаемое буквой латинского алфавита. Среди

состояний выделяется единственное начальное (буква А) и заданные пользователем целевые состояния.

Пример. Вариант №5. Реализовать алгоритм обхода в ширину в поиском 4-х целевых состояний на основе 15 правил преобразования, задаваемых пользователем в программе с учетом варьируемых коэффициентов ветвления и заданных коэффициентов сужения.

$A \rightarrow B|E$; два правила, *коэф. ветвления* -2
 $B \rightarrow C|D|F$; три правила *коэф. ветвления* -3
 $E \rightarrow F|M|I$; три правила *коэф. ветвления* -3
 $C \rightarrow M|L$; три правила *коэф. ветвления* -2
 $L \rightarrow K$; одно правила *коэф. ветвления* -1
 $F \rightarrow M|J|S$; три правила *коэф. ветвления* -3

Целевые состояния, определяются пользователем в программе, например: D, L, M, I.

4 Контрольные вопросы

1. Виды организаций поиска.
2. Количественные характеристики графа поиска.
3. Назначение списков SL и NSL при поиске в глубину.
4. Назначение списка OPEN при поиске в ширину.
5. Структуры данных FIFO и LIFO при организации поиска.
6. Эвристический и систематические виды поиска.
7. Поиск в глубину с итерационным заглублением.
8. Ограниченный поиск в глубину.
9. Состав элемента списка при поиске в ширину.
10. Алгоритм поиска в ширину с итерационным заглублением.

Библиографический список

1. Автоматизированные информационные системы и интеллектуальные технологии [Текст] : учебное пособие / Е. А. Титенко [и др.] ; Минобрнауки России, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Юго-Западный государственный университет". - Курск : ЮЗГУ, 2013. - 133 с.

2. Потапов А.С. Технологии искусственного интеллекта [Электронный ресурс]: учебное пособие. - СПб: СПбГУ ИТМО, 2010. - 218 с.

3. Серегин, М. Ю. Интеллектуальные информационные системы [Электронный ресурс]: учебное пособие / М. Ю.Серегин , М. А Ивановский, А. В.Яковлев ; Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2012. - 205 с.

4. Сидоркина, И. Г. Системы искусственного интеллекта [Текст] : учебное пособие / И. Г. Сидоркина. - Москва : КНОРУС, 2016. - 246 с.

5. Андрейчиков, А. В. Интеллектуальные информационные системы [Текст] : учебник / А. В. Андрейчиков, О. Н. Андрейчикова. - М. : Финансы и статистика, 2006. - 424 с