

УДК 004.8

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ СОСТАВЛЕНИЯ РАСПИСАНИЯ

Астахова И.Ф., Асвад Ф. М.

Статья посвящена составлению расписания учебных занятий, что является одной из важнейших задач управления учебным процессом. В связи с этим проблема автоматизации составления расписаний учебных занятий в образовательных системах массового обучения по-прежнему остается одной из актуальных проблем организации учебного процесса.

THE APPLICATION OF GENETIC ALGORITHM FOR TIMETABLE COMPOSITION

Astahova I.F., Aswad F.M.

The article is devoted to timetable composition for lessons. This is the most important task in managing of educational process. In this connection as before the problem in automation of timetable composition remains one of actual problem in organization of educational process in systems of mass teaching.

Обязательные требования к расписанию могут быть описаны следующей математической моделью [1]:

$$\sum_{i=1}^p \sum_{j=1}^q \sum_{g_{j-1} < k \leq g_j} \sum_{i > g_j} S_{ki} z_{ki} z_{li} \rightarrow \min$$

$$\sum_{i \in I(k)} z_{ki} = 1, \quad z_{ki} = 0, i \notin I(k), k = \overline{1, n}$$

$$\sum_{k \in K_j(i)} z_{ki} = 1, i = \overline{1, p}, j = \overline{1, q}$$

$$z_{ki} = \begin{cases} 1, k - \text{й предмет преподается на } i - \text{м уроке} \\ 0, \text{ иначе; } k = \overline{1, n} \quad i = \overline{1, p}. \end{cases}$$

Здесь

Целевая функция минимизирует количество несовместимых предметов преподающихся на каждом уроке. Первая группа ограничений отвечает за то, что каждый предмет k поводится только один раз, причем на допустимом уроке. Вторая группа ограничений следит за тем, чтобы в каждой группе j на каждом уроке i проводится только один предмет. Заметим, что в результате получилась модель, похожая на постановку квадратичной задачи о назначениях, причем оптимальное значение целевой функции известно заранее и равно 0. Все дополнительные требования к расписанию, не учтенные в модели, выполняются алгоритмически.

Для построения генетического алгоритма необходимо выбрать способ представления данных в виде хромосом [2]. В данном случае хромосома будет матрицей. Ее элемент $y_{ik} = k$, если у i -й группы на i -ом уроке проводится предмет k ($y_{ij} = 0$, если у j -й группы нет i -го урока). Связь с переменными z_{ki} в модели: $y_{il} = k$, если $z_{ki} = 1$, $g_{l-1} < k \leq g_j$, $i \in \overline{1, p}$, $j \in \overline{1, p}$, $k \in K_j(i)$. В качестве стратегии отбора в данной задаче используется турнирный отбор (он оказался для нее более эффективным, чем пропорциональный). Поэтому функцию приспособленности можно не

максимизировать, а уменьшать от популяции к популяции. В ее роли выступает целевая функция задачи с добавками в виде штрафов за нарушение дополнительных ограничений. (Например, вводится матрица: $l_{ij} = 1$, если предметы i, j не могут преподаваться в один день; $l_{ij} = 0$, если могут, и рассчитывается количество нежелательных ситуаций в оцениваемом расписании. и т.п.) Полученные значения добавляются в функцию приспособленности (возможно, с коэффициентами, характеризующими их значимость). При скрещивании происходит обмен фрагментами внутри одного выбранного столбца матрицы Y . В качестве оператора скрещивания используется циклический кроссовер. Он гарантирует допустимость получаемых после скрещивания хромосом, т.к. каждый ген потомка будет находиться на том месте, на котором он находился в одном из родителей. Для мутации внутри одного столбца выбирается любая пара генов таких, что их обмен не нарушит допустимости хромосомы, затем эти гены меняются местами. Проведенные эксперименты показывают эффективность генетического алгоритма построения расписания. Опишем подробнее генетический алгоритм [2]:

1. [Старт] – выбирает случайную популяцию n хромосом (подходящее решение проблемы).
2. [Фитнес] – оценивает пригодность $f(x)$ каждой хромосомы в популяции.
3. [Новая популяция] – создает новую популяцию, повторяя последующие шаги, до тех пор, пока не сформируется новая популяция.
 - 3.1. [Селекция] – выбирает две пары хромосом из популяции в соответствии с их пригодностью (чем лучше пригодность, тем больше шансов в выборе).
 - 3.2. [Скрещивание] – скрещивает родителей, чтобы получить нового потомка. Скрещивание происходит с определенной вероятностью. Если скрещивание не выполнилось, то потомок – это точная копия одного из родителей.
 - 3.3. [Мутация] – изменяет несколько отличительных признаков нового потомка в локусе (участок хромосомы) с определенной вероятностью. Если мутации не произошло, потомок является прямым результатом скрещивания, или копией одного из родителей.
 - 3.4. [Принятие решения] – ставит нового потомка в новую популяцию.
4. [Замена популяции для организации цикла] – использует новую сформированную популяцию для дальнейшего хода алгоритма.
5. [Тест] – если конечное условие выполнено, останавливает, и возвращает наилучшее решение в текущую популяцию.
6. [Повтор] – переход к шагу 2.

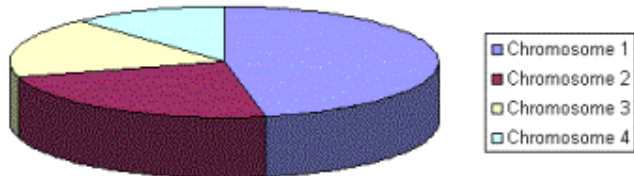
Первым шагом в генетическом алгоритме является «перевод» проблемы в «биологические термины». Формат хромосомы называется кодированием. В общем, используются 4 метода кодирования: двоичное кодирование, перестановочное кодирование, кодирование прямого значения и древовидное кодирование. В данной работе используется двоичное кодирование. Двоичное кодирование одно из самых распространенных и простых. В двоичном кодировании каждая хромосома – это строка из битов, 0 или 1. Например:

Chromosome A:0101101100010011

Chromosome B: 1011010110110101

В порядке воспроизведения потомка, необходимо выбрать родителей. Наиболее распространенными методами являются метод рулетки и метод линейного выбора.

Ключом выбора по правилам рулетки является фитнес. Чем лучше фитнес хромосомы, тем больше шансов их выбрать. Представим рулетку, где все хромосомы размещены так, что каждая имеет свою секцию в соответствии с их функциями фитнеса. У которой самый лучший фитнес занимает самую большую площадь, а у которой наихудший фитнес занимает самую маленькую площадь. Затем бросается шарик и выбирается хромосома.

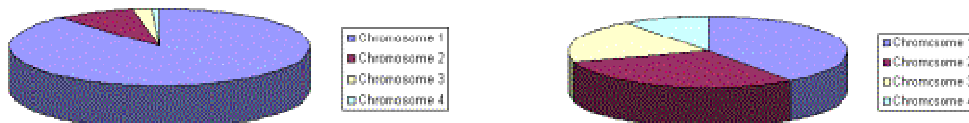


Фигура 1 Выбор по правилам рулетки.

Следующий символический код – это один из способов имитации выбора по правилам рулетки:

1. [Сумма] – считает сумму всех фитнесов хромосом в популяции – сумма S .
2. [Выбор] – выбирает случайные цифры из интервала $(0, S)$ и присваивает в это значение r .
3. [Круг] – проходит через популяцию и суммирует фитнес от 0 – сумма S . Когда сумма S больше чем r , останавливается и возвращает хромосому.

В данной работе используется метод рулетки. Выбор по правилам рулетки может применяться в проблемах, когда фитнес между хромосомами очень сильно отличается. Например, если самая подходящая хромосома занимает 90% всей рулетки, у других хромосом будет очень мало шансов быть выбранными. Ключом рангового выбора является ряд. Перед вращением рулетки, сначала нужно поставить хромосомы в порядке, основанном на их фитнесе. Самые худшие получают фитнес 1, вторые – 2 и т.д., а самые сильные получают наивысший



До ранжирования
(графы фитнеса)

после ранжирования (графы
порядкового номера)

Фигура 2. сравнение выбора по правилам рулетки и ранжирования.

Движущей силой эволюции является воспроизведение. В зависимости от природных проблем и кодирования, выполнение скрещивания и мутации может резко отличаться. Есть несколько основных идей скрещивания и мутации для двоичного и древовидного кодирований.

В данной работе используется одноточечное скрещивание: выбирается одна точка для скрещивания. Двоичная строка от начала родителя 1 до точки скрещивания копируется в нового потомка на такую же позицию. Остальные (от этой же точки скрещивания родителя 2 – в конец) копируются в нового потомка на такую же позицию. Смотри пример ниже.

$$11001|011+1101|111 = 11001111$$

Идея элитарности создана для сохранения самых подходящих хромосом через поколения. Элитарность может очень быстро увеличить действие ГА. Вкратце, генетический алгоритм – это случайный процесс, который выражает различные

действия. Чтобы усилить действие, анализы основываются на разных комбинациях различных параметров, таких как размер популяции, уровень/тип скрещивания, уровень/тип мутации и необходимые методы селекции.

Программный продукт написан на языке C# с применением СУБД SQL Server . Интерфейс разработан на языке C#, БД создана на SQL Server. Интерфейс имеет вид:

The screenshot shows the 'GA For Time Table' application window with the 'Lecturers' tab selected. The table displays the following data:

ID	First_Name	Last_Name	Department
1	Ali	Ahmed	Computer
2	Firas	Aswad	Computer
3	Salam	Abd	Computer
4	Adil	Abd Ali	Computer
5	Mohammed	Ahmed mander	Computer
6	Samera	Ali Mansyr	Computer
7	khaild	Khalf	Computer
8	Sara	Nehad	Computer
9	Hassan	Mohammed	Computer
10	Gaswan	Fahem	Computer
11	Najla Ali	Ahmed	Mathematics
12	Naghah Mohammed	Salah	Mathematics
13	Shama abas	Basam	Mathematics
14	Raed Abas	Ali	Mathematics
15	Muna Al dylami	kalaf	Mathematics
16	Edan Mohammed	Ali	Mathematics
17	Farth Basam	Khaild	Mathematics
18	Saed Abod	Karem	Mathematics
19	Ban Dayad	Ali	Mathematics
20	Haider Mohammed	Abod	Mathematics
21	Dr.Ahmed	Hussia	Mathematics
22	Dr.Hussan	Al Ybady	Mathematics
23	Samera Mansyr	Altae	Mathematics

The screenshot shows the 'GA For Time Table' application window with the 'Subjects' tab selected. The table displays the following data:

ID	Name_Subject	Hours	Department
1	Programming	2	Computer
2	Intermittent Structures	2	Computer
3	Techniques	2	Computer
4	Mathematics	2	Computer
5	Design	2	Computer
6	English	2	Computer
7	Psychology	2	Computer
8	Foundations of Education	2	Computer
9	Arabic	2	Computer
10	The Human Rights	2	Computer
11	Programming (Practical)	2	Computer
12	Techniques (Practical)	2	Computer
13	Design (Practical)	2	Computer
14	Management	2	Computer
15	Architectural	2	Computer
16	Architectural (Practical)	2	Computer
17	Data Structures	2	Computer
18	OOP	2	Computer
19	Computational	2	Computer
20	Systems analysis	2	Computer
23	Systems analysis (Practical)	2	Computer
24	Psychology growth	2	Computer
25	Numerical Analysis	2	Computer

Рис. 1 Интерфейс программного продукта.

Программный продукт позволяет составить расписание вуза Ирака. Результат имеет вид.

Id_Block	Id_Time	Id_Classes
1	12	10
2	12	2
3	3	11
4	5	5
5	18	5
6	13	9
7	2	2
8	9	8
9	14	6
10	9	12
11	7	11
12	10	5
13	16	4
14	1	6
15	17	4
16	19	7
17	12	1
18	16	4
19	6	6
20	18	10
21	14	2
22	3	8

Рис.2.Результаты вычислительного эксперимента

Выводы: в результате вычислительного эксперимента показана эффективность использования генетических алгоритмов для составления расписания.

Литература:

1. Каширина И.Л. Введение в эволюционное моделирование – Воронеж: ВГУ, 2007. – 40 с.
2. Гладков Л.А. Генетические алгоритмы – М.: Физмат, 2006. – 402 с.

Асвад Ф.М., 1979 г., ВГУ в 2010 г., аспирант ВГУ, интеллектуальные технологии в информационно-вычислительных системах, эволюционные вычисления, генетические алгоритмы., 2013, altae13@mail.ru



Астахова И.Ф., окончила Воронежский госуниверситет 1972 г., проф. каф. МО ЭВМ Воронежского госуниверситета, д.т.н. с 1997 г., научные интересы: информационные технологии, математическое моделирование, искусственный интеллект , astachova@list.ru

