РАЗВИТИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ НА ПРИМЕРЕ ЗАДАЧ ТЕОРИИ РАСПИСАНИЙ

Аснина А. Я., Родькина М. Б.

Разрабатывается модель генетического алгоритма, основанная на попытке моделирования развития цивилизации. Разработанная модель тестируется на задаче Беллмана – Джонсона теории расписаний. Алгоритм оперирует несколькими популяциями. Имеет место деление особей по принадлежности к полу. Популяции могут вести между собой войны с целью захвата новых особей, присоединяться к другим популяциям или отделяться от них.

Введение. Генетические алгоритмы — это стохастические, эвристические оптимизационные методы. Круг задач, для решения которых они используются, очень широк. В основном генетические алгоритмы применяются для решения задач оптимизации, т. е. задач, в которых есть некоторая функция нескольких переменных $F(x_1, x_2, ..., x_n)$ и необходимо найти либо её максимум, либо её минимум. Функция F называется целевой функцией, а переменные — параметрами функции. Чтобы реализовать генетический алгоритм для задачи, необходимо сначала выбрать подходящую структуру для представления её решений.

Используется следующая структура. Параметры задачи являются генетическим материалом — *генами*. Совокупность генов составляет *хромосому* (*генотип*). Каждая *особь* обладает своей хромосомой, а, следовательно, своим набором параметров и представляет собой *точку в пространстве поиска всех возможных решений*. Подставив параметры в целевую функцию, можно получить какое-то значение. То, насколько это значение удовлетворяет поставленным условиям, определяет характеристику особи, которая называется *приспособленностью*. Чем "лучше" особь, тем выше приспособленность. Генетические алгоритмы работают с *популяциями* — множествами особей. Популяция — это подмножество допустимого множества задачи.

Чаще всего хромосома — это битовая строка. Но некоторые реализации используют целочисленное или вещественное кодирование. В задачах теории расписаний каждое расписание можно задать перестановкой. Поэтому в качестве хромосом для таких задач можно использовать перестановки, соответствующие расписаниям.

Работа генетического алгоритма основана на использовании ряда операторов кроссинговера и мутации, которые, используя допустимые генотипы «родительских» особей, сохраняют допустимость генотипов потомков.

Имитация генетических принципов — вероятностный отбор родителей, среди членов популяции, скрещивание их хромосом, отбор потомков для включения в новые поколения объектов на основе оценки целевой функции — ведёт к эволюционному улучшению значений целевой функции (функции приспособленности) от поколения к поколению (Каширина, 2007).

Общая схема генетического алгоритма:

1. формирование начальной популяции;
2. оценка особей популяции (определение приспособленности);
3. отбор (выбор особей для скрещивания);
4. скрещивание (получение потомков на основе родительских особей);
5. мутация (изменение уже существующих особей);
6. формирование новой популяции (принятие решения о том, какие особи из родителей и потомков войдут в популяцию);
7. если не выполняется критерий останова, то 2, иначе — останов.

Как правило, все модели генетических алгоритмов имитируют развитие биологического вида. В разработанную модель вводится некоторое подобие социального поведения (для имитации развития цивилизации): популяциям дается возможность принимать решения в зависимости от их состояния на текущий момент времени.

Модель тестируется на классической задаче Беллмана – Джонсона о перестановочном расписании, поэтому привязана к определенной структуре. Задача формулируется следующим образом. Рассматривается система обслуживания, состоящая из M приборов и N требований. Последовательность обслуживания каждого требования одинакова и совпадает с нумерацией приборов. Очередность обслуживания требований одинакова для всех приборов и задаётся перестановкой $\sigma = (i_1,.i_2,...i_N)$. Необходимо найти расписание минимальной длины.

Модель задачи имеет вид:

$$\begin{aligned}&\sum_{i=1}^{N} x_{ij} = 1 \quad j = 1..N;\\ &\sum_{j=1}^{N} x_{ij} = 1 \quad i = 1..N;\end{aligned} \tag{1}$$

$$x_{ij} \in \{0;1\};$$

$$\begin{aligned}&-T_1 X_1 + y_{21} - z_{21} = 0;\\ &T_2 X_{j-1} - T_1 X_j + z_{2j-1} + y_{2j} - z_{2j} = 0, \quad j = 2..N;\end{aligned} \tag{2}$$

$$\begin{aligned}&-T_{k-1} X_1 - y_{k-1,1} + y_{k1} - z_{k1} = 0, \quad k = 3..M;\\ &T_k X_{j-1} - T_{k-1} X_j - y_{k-1,j} + z_{k,j-1} + y_{kj} - z_{kj} = 0, \quad j = 2..N;\\ &\min L(X,Y) = \sum_{j=1}^{N} T_M X_j + \sum_{j=1}^{N} y_{Nj}.\end{aligned} \tag{3}$$

T_k — k-я строка матрицы длительностей обслуживания требований, элементы которой — длительности обслуживания требований на приборе с номером k; X_j — столбец матрицы X, определяющий номер требования, стоящего на j-м месте в расписании; y_{kj} — время простоя k-го прибора перед обслуживанием на нем требования, стоящего на j-м месте в расписании; z_{kj} — время простоя j-го требования в расписании перед обслуживанием на приборе с номером k.

Здесь равенства (1) определяют некоторый порядок обслуживания требований, т.е. задают расписание. (2) определяют зависимость простоев приборов и требований от порядка обслуживания этих требований. (3) — целевая функция, вычисляющая длину расписания как чистое время работы последнего прибора, которое зависит только от длительности обслуживания требований на этом приборе, в сумме со всеми его простоями. На простои влияет порядок обслуживания.

Пусть задан некоторый порядок выполнения работ, т.е. задана перестановка $\sigma = (i_1,.i_2,...i_N)$. Она однозначно определяет матрицу X по правилу $X_j = t_{i_j}$, где t_{i_j} — единичный вектор столбец с единицей на месте i_j. (Аснина и Тищенкова, 1982).

Алгоритм. Особь в алгоритме представлена одной хромосомой, которая является перестановкой, соответствующей определённому расписанию.

Количество начальных популяций не фиксировано: одна или несколько в зависимости от размеров задачи. Каждая популяция набирается из перестановок, составленных случайным образом.

В качестве функции приспособленности особей выступает функция вычисления длины расписания. Так как задача состоит в нахождении расписания минимальной длины, то более приспособленной считается особь (расписание), которой соответствует меньшее значение целевой функции.

Отбор особей для составления пар осуществляется следующим образом. Все особи популяции делятся на две группы по принадлежности к определённому полу (мужскому или женскому). Четная перестановка считается женской особью, нечетная — мужской. Родительская пара формируется из двух особей разного пола, которые отбираются из соответствующих групп путем рулеточного отбора.

После выбора родителей происходит скрещивание, в результате которого получаются потомки. Количество потомков — случайная величина, принимающая значения от 1 до 4. Появление одного-двух потомка более вероятно, вероятность получения трех-четырех новых особей мала. Потомки получаются умножением квадрата одной из перестановок на другую (2 потомка), а также перемножением перестановок (2 потомка). После скрещивания и потомки, и родители возвращаются в популяцию, из которой были взяты родители. В случае, когда потомок совпадает с какой-либо уже существующей особью, происходит его мутация путем двойной транспозиции двух произвольных элементов перестановки, соответствующей потомку.

Каждой особи популяции ставится в соответствие число — её возраст. У особей начальных популяций и у получаемых при скрещивании потомков возраст равен 0. Затем при каждом скрещивании возраст всей популяции возрастает на единицу.

Как только возраст особи достигает определенного порога, она заносится в так называемую критическую область. При занесении в эту область особь сохраняет способность к скрещиванию, но подвержена риску гибели при каждом скрещивании особей популяции. Чем старше особь и чем меньше её приспособленность, тем выше её риск погибнуть. На каждом шаге развития популяции возможна гибель до 4-х особей критической области.

В каждой популяции самая приспособленная особь считается лидером. Смена лидера происходит, если лидер погибает или появляется более приспособленная особь.

Лучшее значение функции приспособленности (рекорд) за все время работы алгоритма хранится отдельно даже после смерти своего носителя.

Популяция может находиться в трех состояниях: состоянии стабильности, упадка и процветания. Для определения этих состояний вводятся две величины: порог упадка — $\underline{T}$ и порог процветания — $\overline{T}$. При этом $\overline{T} >> \underline{T}$.

Популяция находится в стабильном состоянии, если количество её особей принадлежит отрезку $[\underline{T}, \overline{T}]$, и количество мужских и женских особей примерно одинаково.

Если численность популяции становиться меньше порога упадка, или число особей одного пола превышает количество особей противоположного пола более чем в 1,5 раза, то популяция переходит в состояние упадка.

Если же численность популяции превышает порог процветания, то популяция переходит в состояние процветания и считается при этом перенаселенной. При этом если все же какой-то пол сильно преобладает, популяция все равно считается процветающей.

По ходу алгоритма каждая популяция стремится сохранить свое состояние стабильности. Поэтому при переходе в состояние процветания или в состояние упадка популяция должна предпринять какие-либо действия по возвращению своей стабильности.

Популяция может быть агрессивной, т.е. склонной к войне, или быть склонной к молитве. Склонность к агрессии, либо к молитве — начальная характеристика популяции (задается пользователем), она не зависит от состава популяции и не меняется на протяжении всего времени существования популяции. Склонность влияет на вероятность принятия того или иного решения.

В состоянии упадка популяция может прибегнуть к трём тактикам.

Популяция может объявить войну другой популяции с надеждой захватить новых особей (такое поведение характерно для агрессивных популяции и популяций, где женщин значительно меньше, чем мужчин). Популяция-противник выбирается случайным образом. Выбранная популяция может сдаться и при этом отдать часть своих особей без военных потерь с обеих сторон или же принять вызов. На исход войны и на военные потери оказывает влияние количество мужчин в популяциях и приспособленность мужчин и лидеров популяций. Победившая популяция может забрать часть особей проигравшей, если пожелает.

Популяция может прибегнуть к молитве (используют не агрессивные популяции и те, у которых женщин значительно больше, чем мужчин). Если молитва будет услышана, то на некоторое время возрастет вероятность получения большего количества потомков при каждом скрещивании или возрастет вероятность получения потомков определенного пола (в зависимости от причины перехода в состояние упадка).

Если численность популяции меньше порога упадка, то она может принять решение о вырождении. При этом она выбирает другую популяцию и предлагает ей принять часть своих особей. Выбранная популяция может отказаться, а может принять часть особей или всю популяцию. Эту процедуру вырождающаяся популяция повторяет пока все её особи не перейдут в другие популяции. После этого популяция считается исчезнувшей.

В случае процветания популяции, чем больше отрыв численности популяции от порога процветания, тем выше вероятность того, что часть популяции (≈1/3) отделится вместе с каким-нибудь молодым лидером и образует новую популяцию.

Останов алгоритма происходит по количеству скрещиваний. Другие остановы не используются.

Результаты. Алгоритм был протестирован на системе обслуживания из 3-х машин и 10-ти работ с матрицей длительностей выполнения работ:

$$T = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ a_i & 3 & 7 & 8 & 9 & 7 & 8 & 5 & 3 & 6 & 1 \\ b_i & 2 & 10 & 9 & 7 & 6 & 9 & 9 & 1 & 1 & 2 \\ c_i & 3 & 4 & 9 & 7 & 8 & 4 & 7 & 1 & 3 & 9 \end{pmatrix}$$

Для заданного примера известна длина оптимального расписания равная 66. Известно также, что метод ветвей и границ получает точное решение только после полного перебора вариантов.

Тестирование показало, что алгоритм довольно быстро получает оптимальное решение за небольшое количество итераций при моделировании развития нескольких популяций (3–4) небольшой численности (≈50 особей).

Заключение. Разработанный генетический алгоритм быстро находит приближенные и часто даже оптимальные решения тестовых задач. Следовательно, его можно использовать для решения задач теории расписаний конвейерного типа и, скорее всего, также для решения других оптимизационных задач.

СПИСОК ЛИТЕРАТУРЫ

Аснина А. Я., Тищенкова Р. А. Об опыте решения задач Джонсона с произвольным числом станков // Системный анализ и моделирование социально-экономических процессов: Тр.II Всесоюз. семинара. — М., 1982. — Ч. 2. — С. 163–164.

Каширина И. Л. Введение в эволюционное моделирование: учеб. пособие. — Воронеж: Воронежский государственный университет, 2007. — С. 5–6.

GENETIC ALGORITHM IN SCHEDULING THEORY

Asnina A. Y., Rodkina M. B.

The genetic algorithm is used to solve Bellman-Johnson problem. The algorithm is based on modeling of development of multiple populations. All individuals are divided into two gender groups: male and female. Populations can fight with each other to capture some individuals, merge with each other or form new populations by fission.