ВЕСТНИК ВГУ, Серия физика, математика, 2003, № 1

УДК 519.8

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ КВАДРАТИЧНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ СПЕЦИАЛЬНОГО ВИДА

© 2003 И. Л. Каширина

Воронежский государственный университет

В данной статье предлагается генетический алгоритм решения квадратичной задачи о назначениях с запретами и целевой функцией специального вида. Такой класс задач возник при рассмотрении объектного подхода к моделированию процесса составления учебного расписания. Статья содержит подробное описание всех основных этапов генетического алгоритма применительно к данной постановке задачи. В заключение приводятся результаты, полученные в ходе вычислительного эксперимента.

Введение

В данной статье предлагается генетический алгоритм решения квадратичной задачи о назначениях с запретами и целевой функцией специального вида. Такой класс задач возник при рассмотрении объектного подхода к моделированию процесса составления учебного расписания [1].

Процесс составления учебного расписания, как известно, традиционно вызывает немало сложностей, связанных с большим объемом данных и наличием существенного числа требований, накладываемых на готовое расписание. В различных литературных источниках задача составления расписания формализуется в виде общей задачи целочисленного программирования либо имеет графовую постановку [2, 3]. Большие размеры таких задач (как правило, более 100 000 неизвестных) делают невозможным получение их точного решения, а общий вид затрудняет разработку эффективных приближенных алгоритмов.

Используемая в данной статье постановка задачи составления учебного расписания позволила разработать генетический алгоритм ее решения, показавший высокие результаты в ходе проведенного вычислительного эксперимента.

Постановка задачи

Формально задачу составления учебного расписания можно поставить следующим образом.

Заданы множество преподавателей P, множество учебных классов (групп учеников) K, множество учебных дисциплин (предметов) U, учебный план, фиксирующий количество часов каждого предмета в каждом классе, множество аудиторий A. Требуется найти расписание, обслуживающее все дисциплины, удовлетворяющее требованиям преподавателей к проведению занятий, и не имеющее противоречий и накладок относительно аудиторий.

Рассмотрим объектный подход к моделированию данной задачи. В качестве основного элемента расписания выбирается класс “Предметы”, объекты которого обладают следующими свойствами: название (идентификатор) предмета, включающее класс, в котором предмет преподается (например, “Математика 7А”, “История 8Б”); учитель, который преподает данный предмет в данном классе, номер аудитории (кабинета).

Расписание представляется в виде таблицы. Все участвующие в составлении расписания ячейки пронумерованы от 1 до m (m — общее количество таких ячеек). Все объекты — предметы, подлежащие вставке в расписание, также пронумерованы от 1 до n (n — общее количество объектов), причем количество экземпляров каждого объекта равно количеству часов в неделю данного предмета (согласно учебному плану).

Если структура расписания полностью определена, то количество пронумерованных ячеек таблицы совпадает с количеством предметов для расстановки (т.е. $m = n = N$). Заранее зафиксированная структура расписания позволяет гарантировать отсутствие “окон” в расписании (т.е пустых промежутков между уроками в один учебный день), а также позволяет в какой-то степени учитывать санитарно-гигиенические нормы (согласно которым, например, пик учебной нагрузки должен приходиться на среду).

Введем в рассмотрение матрицу совместимости предметов S.

$$S_{ik} = \begin{cases} 1, \text{ если предмет } i \text{ совместим с предметом } k; \\ 0, \text{иначе.} \end{cases}$$

Несовместимость предметов выражается в том, что они используют общий ресурс (учитель или кабинет) и поэтому не могут находиться в расписании на одной строке (то есть преподаваться одновременно). Положим $S_{ii} = 0,\ i = \overline{1,n}$ (так как любой предмет не может появиться на одной строке дважды).

Кроме того, введем матрицу сравнения ячеек R.

$$R_{jl} = \begin{cases} 1, \text{ если ячейка } j \text{ подлежит} \\ \quad \text{сравнению с ячейкой } l; \\ 0, \text{ иначе; } \ j = \overline{1,N},\ \ l = \overline{1,N}. \end{cases}$$

Две ячейки считаются подлежащими сравнению, если они расположены в расписании на одной строке.

Переменные определим следующим образом.

$$x_{ij} = \begin{cases} 1, \text{ если в ячейку } i \text{ назначается} \\ \quad \text{предмет } j; \\ 0, \text{ иначе } i = \overline{1,N},\ \ j = \overline{1,N}. \end{cases}$$

Целевая функция будет минимизировать количество несовместимых предметов в составленном расписании.

$$\sum_{i=1}^{N}\sum_{k=i+1}^{N}\sum_{J=1}^{N}\sum_{l=j+1}^{N} S_{ik}R_{jl}x_{ij}x_{kl} \to \min$$

При формировании целевой функции учтены свойства симметричности матриц S и R. Заметим, что если для выбранной структуры расписания существует непротиворечивый вариант его составления, то оптимальное значение целевой функции равно 0.

Введем в рассмотрение множества $I(j)$ — номера тех ячеек, в которых может находиться предмет j (в соответствии с учебным планом или пожеланиями конкретного преподавателя). Тогда ограничения задачи имеют следующий вид.

$$\sum_{i \in I(j)} x_{ij} = 1,\ \ j = \overline{1,N}.$$

(Каждый предмет назначается ровно в одну ячейку).

$$\sum_{j: i \in I(j)} x_{ij} = 1,\ \ i = \overline{1,N}.$$

(В каждую ячейку назначается ровно один предмет).

Получили модель задачи о назначениях с “запретами” и квадратичной целевой функцией.

$$\sum_{i=1}^{N}\sum_{k=i+1}^{N}\sum_{J=1}^{N}\sum_{l=j+1}^{N} S_{ik}R_{jl}x_{ij}x_{kl} \to \min \tag{1}$$

$$\sum_{i \in I(j)} x_{ij} = 1,\ \ j = \overline{1,N} \tag{2}$$

$$\sum_{j:\ i \in I(j)} x_{ij} = 1,\ \ i = \overline{1,N} \tag{3}$$

$$x_{ij} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \ i = \overline{1,N},\ \ j = \overline{1,N}. \tag{4}$$

Если предусмотреть возможность не фиксировать четко границы расписания (например, в четверг и пятницу есть как минимум 4 урока, а пятый урок только в какой-то один из этих дней), то ограничения (3) разобьются на две группы.

$$\sum_{J=1}^{N} x_{ij} = 1,\ \ i \in I_1 \tag{3'}$$

(I_1 — номера тех ячеек, которые обязательно должны быть заполнены).

$$\sum_{J=1}^{N} x_{ij} \le 1,\ \ i \in I_2 \tag{3''}$$

(I_2 — номера тех ячеек, в которых предметы могут отсутствовать).

Задача (1,2,3’,3”,4) относится к классу открытых задач о назначении и сводится к обычной задаче путем введения дополнительных переменных.

Генетический алгоритм

Для решения квадратичной задачи о назначениях разработано несколько точных алгоритмов [4]. Однако, как известно, эта задача относится к классу *NP*-полных, поэтому при данной размерности отыскание точного решения представляется затруднительным. В связи с этим для решения полученной задачи предлагается использовать генетический алгоритм. Опыт показывает, что генетические алгоритмы особенно эффективны при поиске глобального оптимума, поскольку они осуществляют поиск в широком пространстве решений. Кроме того, выбор генетического алгоритма обусловлен тем, что он в результате предоставляет не одно, а целое множество приближенных решений. По окончании работы генетического алгоритма на этом множестве можно организовать поиск по дополнительным критериям, не учтенным в исходной постановке (например, минимизировать количество "окон" для преподавателей).

Чтобы смоделировать эволюционный процесс, вначале генерируется случайная популяция — несколько индивидуумов со случайным набором хромосом (числовых векторов). Каждая компонента хромосомы называется *геном*. Генетический алгоритм имитирует эволюцию этой популяции как циклический процесс скрещивания индивидуумов и смены поколений. Жизненный цикл популяции — это несколько случайных скрещиваний (посредством специального оператора кроссовера) и мутаций, в результате которых к популяции добавляется какое-то количество новых индивидуумов. Отбор в генетическом алгоритме — это процесс формирования новой популяции из старой, после чего старая популяция погибает. Популяция следующего поколения формируется в соответствии с целевой функцией, называемой функцией приспособленности. Чем приспособленнее индивидуум, тем больше вероятность его участия в кроссовере, т.е. размножении. Существует так называемая "теорема шим" (The schema theorem), которая гарантирует, что от популяции к популяции среднее значение функции приспособленности будет улучшаться [5].

Представление данных

Для построения генетического алгоритма решения квадратичной задачи о назначениях необходимо выбрать способ представления данных в виде хромосом. В данном случае хромосома будет строкой *g* длины *N*, что $g[i]=k$, если $x_{ik}=1$. Эта форма представления называется *перестановочным кодированием* и лучше всего подходит для комбинаторных оптимизационных задач, таких как задача о назначениях. Таким образом, допустимая точка $X=\begin{pmatrix}1 & - & 0 & -\\ 0 & 0 & 1 & 0\\ 0 & 1 & - & -\\ - & 0 & 0 & 1\end{pmatrix}$ (прочерками отмечены запрещенные места) будет кодироваться строкой $g=1324$.

Формирование начальной популяции

Так как рассматриваемая задача о назначениях имеет запреты, то особое внимание нужно уделить формированию начальной допустимой популяции.

В ходе вычислительного эксперимента себя хорошо зарекомендовал следующий алгоритм.

Шаг 1. Для всех $j=\overline{1,n}$ определить множества $I(j)$.

Шаг 2. Выбрать случайный номер $k\in\{1..N\}$.

Шаг 3. Выбрать случайный номер $i1\in I(k)$.

Шаг 4. Положить $g[k]=i1$, $I(j)=I(j)\setminus\{i1\}$, убрать из рассмотрения множество $I(k)$.

Шаг 5. Среди оставшихся множеств $I(j)$ определить новое множество $I(k)$ с минимальным числом элементов.

Шаг 6. Если хромосома *g* еще не сформирована, вернуться на шаг 3, иначе включить особь *g* в начальную популяцию.

Данная процедура повторяется до тех пор, пока начальная популяция не будет сформирована. Количество особей в популяции зависит от размерности исходной задачи (например, для $N=900$ оптимальный размер популяции ≈ 150 особей). Для повышения средней приспособленности начальной популяции рекомендуется включать в нее несколько локальных решений, получаемых с помощью простейших приближенных алгоритмов. Для данной задачи можно предложить следующий приближенный алгоритм.

Шаг 0. Полагается $\Omega=\{(i,j),\, j=\overline{1,n}, i\in I(j)\}$ множество индексов всех неизвестных задачи.

Шаг 1. Для каждой незафиксированной переменной $x_{ij}, (i,j)\in\Omega$ вычисляется величина k_{ij} — оставшееся количество слагаемых целевой функции, в которых присутствует величина x_{ij}.

Шаг 2. Отыскивается переменная x_{rs}, такая что $(r,s)\in\Omega$ и $k_{rs}=\min\limits_{i\in I(j)} k_{ij}$. Полагается $x_{rs}=1$, $x_{rj}=0$ для всех $j\neq s$, $x_{is}=0$ для $i\neq r$.

Шаг 3. Из множества Ω удаляются индексы переменных, для которых зафиксировано нулевое значение: $\Omega=\Omega\setminus(\{(r,j),\ \forall j\neq s\}\cup\cup\{i,s),\ \forall i\neq r\})$.

Шаг 4. Если остались незафиксированные переменные, производится возврат на шаг 1.

Шаг 5. По точке X строится хромосома g.

Скрещивание

Необходимо выбрать такой оператор кроссовера, чтобы он гарантировал допустимость получаемых после скрещивания хромосом (то есть не допускал появления единиц на запрещенных местах). Этому требованию удовлетворяет циклический кроссовер. В циклическом кроссовере каждый ген хромосомы берется от одного из родителей. Первый ген потомка берется от первого родителя. Второй ген потомка берется у второго родителя из последней позиции . Если невозможно взять его от второго родителя, так как этот ген уже находится в хромосоме, то его берут от первого родителя и так до тех пор, пока новая хромосома не будет создана. Например, пусть $g^1=13542$, $g^2=23451$. Тогда их потомками будут $g^3=13452$ и $g^4=23541$. Если в процессе скрещивания возникает ситуация, что оба гена предков уже участвуют в хромосоме, то на очередное j-тое место выбирается любой номер $i\in I(j)$. Если на каком-то шаге таких номеров не окажется, то хромосомы предков просто копируются в хромосомы потомков. Например, пусть $g^1=13425$, $g^2=52314$. Тогда первый из их потомков будет формироваться в следующей последовательности: $1{*}{*}{*}{*}\Rightarrow 1{*}{*}{*}4\Rightarrow 13{*}{*}4\Rightarrow 13{*}24\Rightarrow 13524$ (если значение $x_{35}=1$ допустимо).

Мутация

Для мутации выбирается любая пара генов таких, что их обмен не нарушит допустимости хромосомы. Затем эти гены меняются местами. Пример: хромосома 12435 после мутации принимает вид 52431.

Оценка приспособленности

Поскольку целевая функция в задаче минимизируется, а пропорциональный отбор особей для скрещивания отдает предпочтение хромосомам с наибольшей приспособленностью, то в качестве функции приспособленности целесообразно использовать функцию

$$f(x)=1\Bigg/\left(\sum_{i=1}^{N}\sum_{k=i+1}^{N}\sum_{J=1}^{N}\sum_{l=j+1}^{N}S_{ik}R_{jl}x_{ij}x_{kl}+1\right),$$ где x — допустимая точка, соответствующая хромосоме g.

Вычислительный эксперимент

Описанный в статье генетический алгоритм был реализован в среде программирования Delphi 7.0. Проведенный вычислительный эксперимент показал его высокую эффективность. Для всех 100 случайным образом сгенерированных тестовых задач (с размерностью 900—1000 переменных) было найдено оптимальное решение с нулевым значением целевой функции. Этот результат открывает новые широкие возможности для разработки автоматизированных систем построения учебных расписаний и повышения их качества.

ЛИТЕРАТУРА

1. *Каширина И. Л. Минаков С.В.* Система автоматического построения расписания учебных занятий.// Труды Российской ассоциации “Женщины-математики”. Т. 10, вып. 1./ Под ред. Б. И. Голубева, И. С. Гудович, И. Я. Новикова. Воронеж, 2002. — 119 с.

2. *Коффман Э.Г.* Теория расписаний и вычислительные машины. — М.: Наука, 1984. — 367 с.

3. Система моделей и методов планирования и организации учебного процесса в ВУЗе./ Под ред. В. В. Гусева, Н. Я. Краснера. Воронеж: Изд-во ВГУ, 1984. — 290 с.

4. *Сергеев С.И.* Квадратичная задача назначения.// Авиационная и ракетная техника. — 1999. — № 8. — С. 127—147.

5. *Goldberg D.E., Sastry K.* A Practical Schema Theorem for Genetic Algorithm Design and Tuning, 2001.