УДК 519.712

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ РАБОТ

А.Р. АЙДИНЯН, О.Л. ЦВЕТКОВА
(Донской государственный технический университет)

Исследованы задачи распределения неоднородных работ между неоднородными исполнителями с учетом затрат на выполнение работ и переключение между ними. Предложены генетические алгоритмы распределения работ, которые могут использоваться при решении различных практических задач как в автоматизированных процессах, так и в социальных системах.
Ключевые слова: *генетические алгоритмы, распределение работ, неоднородные работы, неоднородные исполнители.*

Введение. Задачи распределения работ между несколькими исполнителями возникают в различных областях науки, техники и социальных сферах, причем характер работ и правила их распределения могут быть различными в зависимости от рассматриваемой прикладной области.

Как правило, задачи распределения работ относятся к сложным многоэкстремальным задачам, эффективные методы решения которых известны только для ряда частных случаев. Обычно для получения точных решений применяют метод ветвей и границ, метод динамического программирования или эвристические методы [1, 2]. Эффективность метода ветвей и границ в существенной степени зависит от «точности» граничных оценок подмножеств решений. Известные эвристические методы и метод динамического программирования применимы только для определенного класса или даже вида задач и неэффективны или неприменимы при решении наиболее общей и наиболее приближенной к практике задачи – распределения работ в случае, когда затраты зависят от последовательности выполнения работ. Поэтому актуальной является разработка эффективных методов решения задач распределения неоднородных работ между несколькими неоднородными исполнителями.

Постановка задачи. При распределении работ между исполнителями необходимо минимизировать затраты на их выполнение. В понятие затраты включаются потери времени, расход электроэнергии, износ оборудования и т.д. В целях перехода к однокритериальной задаче оптимизации предлагается свести все учитываемые затраты к одному параметру. (Способ перехода от многокритериальной задачи к однокритериальной в данной статье не рассматривается).

Пусть имеется $n\left(i=\overline{1,n}\right)$ неоднородных исполнителей и $m\left(j=\overline{1,m}\right)$ неоднородных работ, представленных в виде множеств: $R=\{r_1,\ldots,r_n\}$ и $W=\{w_1,\ldots,w_m\}$ соответственно.

Особенностью рассматриваемой задачи является то, что процесс переключения исполнителя с работы на работу сопровождается затратами, которые зависят как от исполнителя, так и от последовательности работ.

Необходимо получить последовательность выполнения заданных работ так, чтобы при заданных свойствах исполнителей и работ минимизировать затраты на выполнение работ с учетом затрат на переключение.

Задания поступают на вход системы динамически, т.е. в любой момент времени t на вход системы может поступить задание. Таким образом, необходимо перераспределять работы в реальном масштабе времени, которые на момент времени t не выполнены или не выполняются. То есть работы находящиеся на выполнении не подлежат замещению.

Исполнитель характеризуется затратами на выполнение данного вида работы и затратами на переключение с одного вида работ на другой. Исполнитель может быть не способен выполнить какие-либо работы.

Затраты на выполнение исполнителями определенных работ представляются в виде матрицы Z размерности $n \times m$, в которой элемент z_{ij} показывает затраты на выполнение j-й работы i-м исполнителем. А затраты на переключение исполнителя с работы на работу – в виде матриц $P^{(i)}$ размерности $m \times m$ $\left(i = \overline{1,n}\right)$.

Таким образом, в результате решения задачи необходимо получить такое распределение работ между исполнителями, чтобы обеспечить выполнение следующего критерия:

$$F = \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij} c_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{f=1}^{m} p_{jf}^{(i)} u_{jf}^{(i)} \to \min, \qquad (1)$$

где c_{ij} – элемент матрицы C размерности $n \times m$, принимающий значение 1 или 0 и показывающий, выделена ли исполнителю i для выполнения работа j; $u_{jf}^{(i)}$ – элемент матрицы $U^{(i)}$ размерности $m \times n$, принимающий значение 1 или 0 и показывающий, планируется ли для i-го исполнителя переключение с работы j на работу f.

Первое слагаемое выражения (1) определяет затраты на выполнение всех планируемых работ, а второе – затраты на переключение исполнителей с работы на работу.

Задачу предлагается решать с использованием генетического алгоритма (ГА), который представляет собой простую модель эволюции в природе, реализованную в виде компьютерной программы [3]. В нем используются как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.

Алгоритм решения задачи распределения работ зависит от количества исполнителей (один или несколько) и от того, учитываются ли затраты на переключение с работы на работу. В связи с этим предлагается провести декомпозицию задачи и получить алгоритм формирования последовательности выполнения работ для одного исполнителя и алгоритм распределения работ между исполнителями без учета затрат на переключение, а затем, путем объединения указанных алгоритмов, записать общий алгоритм.

Алгоритм формирования последовательности выполнения работ для одного исполнителя (Алгоритм 1). В случае одного исполнителя задача сводится к минимизации стоимости выполнения всех работ и аналогична задаче коммивояжера без необходимости возвращения в исходный пункт. При этом необходимо получить такую последовательность выполнения работ, чтобы выполнялся критерий:

$$F_1 = \sum_{j=1}^{m}\sum_{f=1}^{m} p_{jf} u_{jf} \to \min.$$

Целевая функция (функция фитнеса) определяется как $\phi = -F_1$ и наилучшей является хромосома, имеющая наибольшее значение функции фитнеса.

Как известно, решение с помощью ГА сводится к последовательному выполнению следующих шагов:

– выбор символьной модели объекта в виде закодированных хромосом;

– генерация начальной популяции;

– модификация хромосом в целях улучшения популяции;

– выбор хромосомы из популяции с максимальным значением функции фитнеса.

Способ кодирования определяется спецификой задачи. При выборе символьной модели необходимо обеспечить возможность кодирования в хромосоме любой точки из пространства поиска. Невыполнение этого условия может привести к невозможности найти решение поставленной задачи.

Хромосома отражает последовательность выполнения работ исполнителем. Как указано выше, исполнитель должен выполнить m работ. Каждая хромосома G представляет собой конкатенацию ряда подкомпонентов, называемых генами $G=(g_1,g_2,\dots g_m)$, где m – длина хромосомы. Число генов в хромосоме равно числу работ, поступивших на выполнение. Ген с номером i определяет номер работы, которую необходимо выполнить i-й в последовательности.

Поскольку одна работа должна быть выполнена только один раз, то каждая работа может входить в хромосому единожды.

Алгоритм получения начальной популяции заключается в следующем. Начальная популяция включает m хромосом. Первым геном i-й хромосомы является i-я работа. Каждым последующим j-м $(j=2,m)$ геном является номер работы с минимальными затратами на переключение между текущей и последующей в соответствии с жадным алгоритмом.

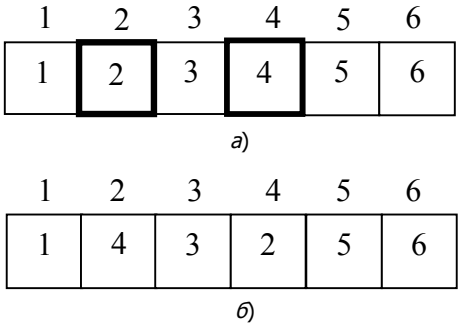
Из полученных хромосом выбирается хромосома с максимальным значением функции фитнеса, т.е. хромосома, которая кодирует последовательность работ с минимальными затратами.

Затем осуществляется модификация выбранной хромосомы с помощью операции мутации.

В данном алгоритме не используется операция кроссовера, а только операция мутации над единственной выбранной хромосомой.

Можно предложить несколько операций мутации.

1. Перестановка двух произвольно выбранных генов. На рисунке приведен пример мутации путем перестановки генов 2 и 4.



Пример операции мутации, заключающейся в перестановке двух произвольных генов:
а – хромосома до мутации; *б* – хромосома после мутации

2. Циклический сдвиг на одну позицию генов, находящихся между двумя случайно выбранными.

3. Перестановка последовательности генов в другое место хромосомы.

4. Замена генов на обратную последовательность.

5. Перестановка последовательности генов в другое место хромосомы в обратной последовательности.

При модификации выбранной хромосомы предлагается повторять все пять мутаций с равной вероятностью до тех пор, пока не будет получено решение с удовлетворительными затратами или в течение некоторого количества мутаций не будет найдено решение с меньшими затратами.

Для увеличения вероятности правильного выбора переставляемых генов и уменьшения времени работы алгоритма введем понятие тесносвязанных работ (ТСР). Это такие работы i и j, для которых затраты на переключение с работы i на работу j меньше или равны затратам переключения с работы i на любую другую, кроме той, что предшествует работе i, или с любой другой на работу j. Соответственно, если эти условия не выполняются, то работы называются нетесносвязанными (НТСР).

Таким образом, разрыв хромосомы предлагается проводить таким образом, чтобы «не разрывались» тесносвязанные работы. Перестановка может осуществляться в любую позицию. После перестановки необходимо заново определить точки разрыва.

Если полученная после модификации последовательность работ, описанная хромосомой, имеет меньшие затраты, то она принимается вместо исходной, если нет – то отбрасывается.

В качестве примера проведем распределение работ, затраты на переключение между которыми представлены матрицей:

$$\begin{pmatrix} 0 & 10 & 20 & 30 & 40 & 50 & 150 & 140 \\ 10 & 0 & 10 & 20 & 30 & 40 & 150 & 50 \\ 20 & 10 & 0 & 10 & 20 & 30 & 90 & 30 \\ 30 & 20 & 10 & 0 & 10 & 20 & 80 & 90 \\ 40 & 30 & 20 & 10 & 0 & 10 & 90 & 120 \\ 50 & 40 & 30 & 20 & 10 & 0 & 100 & 80 \\ 150 & 150 & 60 & 50 & 90 & 100 & 0 & 10 \\ 100 & 90 & 80 & 100 & 110 & 120 & 10 & 0 \end{pmatrix}.$$

Необходимо выполнить каждую работу единожды. Поэтому затраты на выполнение каждой работы неважны.

Воспользовавшись алгоритмом получения начальной популяции, определим следующий набор хромосом:

12345687; 21345687; 32145687; 43215687;
54321687; 65432187; 78321456; 87432156.

В качестве начального приближения выберем первую хромосому 12345687, поскольку она кодирует последовательность работ с минимальными затратами на переключение, равными 140.

Обозначим НТСР знаком «–». В результате получим хромосому 123456-87 с двумя группами ТСР.

В результате выполнения операции мутации получена хромосома 123–87–456 путем применения модификатора 3. Затраты на переключение для этой хромосомы равны 130.

Предложенный метод решения имеет существенное преимущество перед ГА без введения понятия ТСР и полным перебором. Ясно, что количество операций будет зависеть как от количества генов (количества распределяемых работ), так и от количества ТСР.

Сравнительная характеристика полученного алгоритма с классическим ГА приведена в таблице.

Сравнительная характеристика алгоритмов

Количество генов	Количество НТСР	Время работы ГА, мс	Время работы ГА с ТСР, мс	Относительная погрешность полученного решения ГА с ТСР, %
8	3	63	12	0
8	6	63	25	0
24	6	270	31	0
24	12	270	56	0
24	16	270	89	1,1
32	6	634	34	0
32	12	634	64	1,4
32	18	634	93	2,4
100	18	3102	127	2,2
100	42	3102	1136	2,8

Из последнего столбца таблицы видно, что предложенный генетический алгоритм с ТСР может находить решение, отличное от точного с достаточно малой погрешностью.

Алгоритм распределения работ между исполнителями без учета затрат на переключение (Алгоритм 2). Для распределения работ между несколькими исполнителями также можно использовать ГА. В этом случае необходимо получить такое распределение работ, чтобы выполнялся критерий:

$$F_2 = \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}c_{ij} \to \min .$$

Целевая функция (функция фитнеса) определяется как $\phi = -F_2$ и наилучшей является хромосома, имеющая наибольшее значение функции фитнеса.

Предлагается следующий алгоритм генерации начальной популяции при количестве исполнителей равном n. Начальная популяция будет включать хромосомы в количестве, кратном количеству исполнителей. Кодировка хромосомы имеет вид: в i-м гене находится номер исполнителя, который выполняет i-ю работу. Гены j-й хромосомы содержат номер исполнителя $j \bmod m$, где $\bmod$ – операция вычисления остатка от целочисленного деления j на m. В случае, если исполнитель j не может выполнить работу i, то в гене i во всех хромосомах популяции необходимо заменить число j на номер другого исполнителя. Таким образом, можно учесть предпочтение передачи работы какому-то исполнителю.

Поскольку один исполнитель может выполнить несколько работ, то номер одного и того же исполнителя может присутствовать в разных генах.

При этом важным является правильный выбор размера популяций. Размер популяции может повлиять как на скорость сходимости, так и на качество найденного решения. Правильный выбор размера популяций может быть осуществлен только экспериментальным путем. В проведенных экспериментах размер популяции для 8 исполнителей и 32 работ принимался в количестве 80.

Проиллюстрируем алгоритм получения начальной популяции на примере. Пусть имеется 3 исполнителя и 6 работ.

Примем размер начальной популяции равным шести хромосомам. В случае, если любой исполнитель может выполнить любую работу, то начальная популяция имеет вид:

111111; 222222; 333333; 111111; 222222; 333333.

Для случая, если исполнитель 3 не может выполнить работу 2, передадим работу 2 исполнителям 1 и 2 с равной вероятностью. В связи с этим в хромосомах 3 и 6 работа 2 передана исполнителям 1 и 2 соответственно, что обеспечивает равную вероятность выполнения работы 2 этими исполнителями. Таким образом, начальная популяция имеет вид:

111111; 222222; 313333; 111111; 222222; 323333.

Генерация начальной популяции является подготовительным этапом к процессу субоптимизации, который заключается в последовательном применении к полученной популяции генетических операторов согласно ГА.

В данном алгоритме используется модель ГА Уитли (D. Whitley). При скрещивании произвольно выбранных двух хромосом применяется одно- или двухточечный кроссинговер с равной вероятностью. Местоположение точек разрыва выбирается случайным образом. В результате скрещивания получается один потомок, к которому применяется операция мутации.

Мутация заключается в замене в каком-либо гене номера исполнителя на другой, выбранный случайно. Количество мутаций предлагается принимать не более 5% от количества работ. Так, при 32 работах были выполнены мутации в 1-2 генах.

При выполнении L скрещиваний полученная хромосома занимает в популяции во время первых $L/2$ скрещиваний место родительской хромосомы с минимальным значением функции фитнеса, а на последних $L/2$ скрещиваниях – место хромосомы с минимальным значением функции фитнеса во всей популяции.

Алгоритм заканчивает свою работу при достижении удовлетворительного решения или при выполнении указанного количества скрещиваний. При проведении вычислительных экспериментов при 32 работах, 8 исполнителях и размере популяции равном 32 количество скрещиваний принималось равным 750, что позволяло получать решение, близкое к оптимальному.

Алгоритм распределения работ между исполнителями с учетом затрат на переключение (Алгоритм 3). Этот ГА представляет собой комбинацию двух вышеприведенных алгоритмов и позволяет минимизировать критерий (1). Алгоритм имеет вид.

1. Генерация начальной популяции в соответствии с Алгоритмом 2.
2. Определение последовательности работ для каждой хромосомы популяции в соответствии с Алгоритмом 1 и вычисление значения функции фитнеса.
3. Сохранение максимального значения функции фитнеса $\phi_{\max}$.
4. Отбор двух родителей и получение потомка путем скрещивания и мутации в соответствии с Алгоритмом 2.
5. Вычисление значения функции фитнеса ϕ для полученного потомка.
6. Если количество полученных потомков менее $L/2$, то замена потомком родителя с минимальным значением функции фитнеса, иначе – замена потомком наименее приспособленной хромосомы в популяции.
7. Если $\phi > \phi_{\max}$, то перейти на шаг 8, иначе – перейти на шаг 4.
8. Генерация начальной популяции для каждого исполнителя в соответствии с Алгоритмом 1.
9. Получение последовательности работ в соответствии с Алгоритмом 1.
10. Вычисление значения функции фитнеса ψ с учетом последовательности работ.
11. Если $\psi > \phi_{\max}$, то принять $\phi_{\max} = \psi$.
12. Если достигнуто удовлетворительное решение или выполнено указанное количество скрещиваний, то перейти на шаг 13, иначе – перейти на шаг 4.
13. Конец алгоритма.

После получения хромосомы, описывающей распределение работ между исполнителями без учета затрат на переключения, для каждого исполнителя выполняется алгоритм формирования последовательности работ для одного исполнителя с учетом затрат на переключение. Как следует из алгоритма, в целях экономии времени алгоритм формирования последовательности работ для каждого исполнителя (шаги 8-12) выполняется в том случае, если полученная хромосома имеет значение функции фитнеса больше сохраненного максимального значения $\phi_{\max}$. Только в этом случае есть возможность получить распределение работ между исполнителями с затратами меньше ранее найденного решения.

Для уменьшения времени работы алгоритма также используется понятие ТСР. Однако это понятие в данном случае принимает следующий вид. Тесносвязанными являются работы i и j, для которых затраты на переключение с работы i на работу j меньше или равны затратам переключения с работы i на любую другую этого исполнителя, кроме той, что предшествует работе i, или с любой другой работы этого исполнителя на работу j.

Проведенные эксперименты показали, что относительная погрешность найденного решения по сравнению с оптимальным для восьми исполнителей и 32 работ при количестве скрещиваний равном 320 не превышает 7%. При этом время работы алгоритма составило около 9 с на современном ПК.

Заключение. Предложенные алгоритмы позволяют осуществить распределение работ между исполнителями и сформировать последовательность выполнения работ с учетом затрат на переключение. Проведенные вычислительные эксперименты подтверждают возможность использования полученных алгоритмов при решении различных практических задач, связанных с распределением работ, как в автоматизированных процессах, так и в социальных системах. Последующие

доработки связаны с проведением исследований и получением рекомендаций по выбору параметров генетического алгоритма, обеспечивающих уменьшение вычислительных затрат, сходимость алгоритма и нахождение приемлемого субоптимального решения.

Библиографический список

1. Емельянов В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2003. – 432 с.

2. Реклейтис Г. Оптимизация в технике (в 2 кн.) / Г. Реклейтис, А. Рейвиндран, К. Рэксдел. – М.: Мир, 1988. – 348 с.

3. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилинский, Л. Рутковский. – М.: Горячая линия – Телеком, 2006. – 452 с.

Материал поступил в редакцию 03.05.2011.

References

1. Emel`yanov V.V. Teoriya i praktika e`volyucionnogo modelirovaniya / V.V. Emel`yanov, V.V. Kurejchik, V.M. Kurejchik. – M.: Fizmatlit, 2003. – 432 s. – In Russian.

2. Reklejtis G. Optimizaciya v texnike (v 2 kn.) / G. Reklejtis, A. Rejvindran, K. Re`ksdel. – M.: Mir, 1988. – 348 s. – In Russian.

3. Rutkovskaya D. Nejronny`e seti, geneticheskie algoritmy` i nechyotkie sistemy` / D. Rutkovskaya, M. Pilinskij, L. Rutkovskij. – M.: Goryachaya liniya – Telekom, 2006. – 452 s. – In Russian.

GENETIC ALGORITHMS OF WORK DISTRIBUTION

A.R. AIDINYAN, O.L. TSVETKOVA
(Don State Technical University)

The tasks of heterogeneous work distribution among heterogeneous executors taking into account operation and switching process costs are investigated. Genetic algorithms of work distribution that can be used for solving various practical tasks both in the automated processes and in social systems are offered.

Keywords: *genetic algorithms, work distribution, heterogeneous work, heterogeneous executors.*