КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 519.711: 519.711.3: 519.81

И.О. ЛУКЬЯНОВ, Ф.А. ЛИТВИНЕНКО, Е.А. КРИКОВЛЮК

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПАРАЛЛЕЛЬНОЙ ВЕРСИИ МНОГОПОПУЛЯЦИОННОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Рассмотрены некоторые особенности параллельной реализации многопопуляционного генетического алгоритма, а также некоторые подходы к его оптимизации. Проведена экспериментальная оценка принципиально разных стратегий миграции хромосом-решений между популяциями, а также выбора вероятностей операций кроссовера и мутации в зависимости от близости к оптимальному решению. В результате достигнуто значительное (до 25 %) уменьшение количества рассмотренных вариантов решения (альтернатив).

© И.О. Лукьянов,
Ф.А. Литвиненко,
Е.А. Криковлюк, 2018

Введение. В мировой практике компьютерного моделирования сложных стохастических систем все большее распространение приобретает методология Data Farming [1]. В ее основе заложена интеграция возможностей средств имитации, эвристических стратегий оптимизации, методов интеллектуального анализа данных и технологий распределенных вычислений на базе вычислительных платформ кластерных или сетевых архитектур.

Учитывая данное часто упоминается эвристический генетический алгоритм и его различные модификации [2 – 5]. Данные алгоритмы предназначены для поиска оптимальных решений на больших и сверхбольших пространствах значений параметров, при этом он используется в некоторых задачах как метод интеллектуального анализа данных [6].

Поскольку генетический алгоритм имеет вероятностный характер, от эксперимента к эксперименту популяция может с разной скоростью сходиться к допустимым решениям [2 – 4]. Для преодоления данного недостатка классического варианта генетического алгоритма, а также учитывая активное развитие технологий распределенных вычислений, актуальным становится исследование так называемой островной (или многопопуляционной) модели. Данная модификация основана на параллельной эволюции нескольких популяций [7, 8], координирующихся между собой различными стратегиями обмена особями (хромосомами-решениями), а также допускает одновременное использование разных значений параметров на всем множестве популяций. Таким образом, достигается контроль над скоростью нахождения допустимого решения и его близостью к оптимуму.

Цель данной работы – исследование особенностей и эффективности многопопуляционного генетического алгоритма при планировании и реализации оптимизационно-имитационных экспериментов.

В настоящей работе приняты такие допущения:

– вместо имитационной модели предполагается использование однокритериальной fitness-функции F, выступающей аналогом поверхности реакции некоторой имитационной модели;

– при исследовании на данном этапе считаем, что значения, возвращаемые fitness-функцией – детерминированная величина (а не случайная величина, как это имеет место при моделировании сложных стохастических систем), и имеют четко выраженный экстремум.

Описание тестовой задачи. Используя генетический алгоритм и распределенные вычисления, найти некоторую заданную строку длины ***N***, состоящую из символов конечного алфавита размерности *K*. Алфавит считается заранее определенным, размером в *K* = 33 символов, среди которых латинские буквы нижнего регистра, знаки препинания необходимые для формирования базовых предложений (,.’-!?) и пробел. Для исследования использовалась строка длиной в *N* = 100 символов. Fitness-функция *F* сообщает о количестве позиций в оцениваемом текущем решении, значения в которых не совпадают со значениями в целевой (эталонной) строке. Таким образом минимальное значение известно и достигается при *F* = 0. Заметим, что при таком выборе *F* ее оптимальное значение не зависит от *N*.

Сложность данной задачи при полном переборе альтернатив оценивается как K^N. Используя допущения, описанные выше, а также знания о самой fitness-функции и природе задачи, можно реализовать алгоритм нахождения входной строки сложностью приблизительно $K \times N$: если для каждой позиции в строке поочередно перебирать все символы алфавита вплоть до уменьшения (на одну единицу) значения fitness-функции. Поскольку – это исследование особенностей универсального варианта многопопуляционного генетического алгоритма, реализация была основана на его классической модели и особенности конкретной задачи не использовались.

Начальные параметры реализуемого алгоритма.

init_size = 10 – размер начальной популяции: определяет количество хромосом-решений, которые будут сгенерированы для каждой популяции в качестве нулевого поколения;

popul_size = 20 – размер популяции: определяет количество хромосом-решений, которые будут содержаться в каждой популяции в процессе селекции;

elite_part = 0.5 – доля элитной части: определяет количество хромосом-решений, оставшихся после селекции в каждом очередном поколении популяции;

cross_prob = 0.5 – вероятность применения операции кроссовера при генерации новой хромосомы-решения;

mute_prob = 0.5 – вероятность применения операции мутации после операции кроссовера;

share_interval = 5 – количество смен поколений, после которого происходит обмен хромосомами-решениями между популяциями.

Эксперименты проводились на отечественном суперкомпьютере с кластерной архитектурой СКИТ-4 Института кибернетики НАН Украины, с использованием средств MPI (Message Passing Interface). Для осуществления стратегий обмена выделен один процессор с индексом 0 в качестве управляющего (master process – MP). Все остальные процессоры – управляемые (worker process – WP). На WP реализованы разные параллельно развивающиеся популяции, выступающие в качестве «островов» в многопопуляционной модели генетического алгоритма.

Описание алгоритма.

MP считывает исходные данные, параметры алгоритма из начального файла и отправляет их на каждый WP.

На каждом WP генерируется своя независимая начальная популяция размером init_size, в которой каждая хромосома-решение состоит из *N* случайных символов заданного конечного алфавита размерности *K*. Соответственно, в каждой хромосоме-решении начальной популяции некоторые символы могут совпадать с символами искомой строки.

Каждое следующее поколение осуществляется посредством мутации и скрещивания решений-хромосом, полученных в предыдущем поколении. Хромосомы-решения из неэлитной части вытесняются хромосомами-решениями, полученными в результате операции отбора в соответствии со значениями *F*.

Формирования каждого последующего поколения.

Шаг 1. Равновероятно выбирается случайное число x из интервала [0; 1]. Если $x < cross\_prob$, выполняется операция скрещивания: из элитной части равновероятно выбираются 2 хромосомы-решения и новая хромосома-решение формируется путем выбора каждого гена (позиции в соответствующей строке) случайным образом у одного из родителей.

Шаг 2. Равновероятно выбирается случайное число $\boldsymbol{y}$ из интервала [0;1]. Если $y < mute\_prob$, в полученной хромосоме выполняется операция мутации: M случайных генов заменяется случайными символами из заданного алфавита. Значение M определяется в зависимости от номера процесса – для первой половины популяций – 1, для второй – 2.

В случае, если скрещивание не проводилось ($x \geq cross\_prob$) – выполняется мутация случайно выбранной хромосомы из элитной части.

Шаг 3. Из полученной популяции удаляются все дубликаты, после чего популяция сортируется по возрастанию значений fitness-функции *F*, формируя элитную и неэлитную части популяции.

Шаг 4. Каждые share_interval поколений происходит обмен лучшими хромосомами-решениями всех популяций. Каждый WP отправляет свой лучший результат на MP, последний из полученных лучших хромосом выбирает единственную, для которой *F* возвращает наилучший результат, после чего эта хромосома-решение отправляется на каждый WP и добавляется на позицию лучшей в популяции. Если данный вариант хромосомы-решения достиг результата приемлемого решения задачи, на каждый WP приходит сигнал завершения работы.

Анализ результатов экспериментов.

В таблице приведены результаты, полученные вследствие выполнения описанного алгоритма. Здесь min, max и average обозначают минимальное, максимальное и среднее количество поколений соответственно, которое понадобилось для нахождения исходной строки хотя бы на одном из WP.

Эксперименты проводились с использованием пятнадцати WP с различными значениями параметров. Причем значения init_size, popul_size и elite_part в ходе экспериментов были фиксированы и равны начальным значениям параметров. Значения share_interval, mute_prob, cross_prob последовательно менялись (изменения выделены жирным шрифтом) и в последующих экспериментах использовались те значения, которым соответствует лучший результат.

Из таблицы видно, что результаты улучшаются при увеличении вероятности кроссовера cross_prob, что позволяет предположить, что на начальных этапах работы алгоритма скрещивание – это более эффективная процедура, чем мутация, поскольку довольно продуктивно происходит объединение совпадающих с эталоном (искомой строкой) генов из различных хромосом-решений элитной части популяции.

При этом при уменьшении вероятности мутации mute_prob результаты ухудшаются. Причина – это высокая однородность популяции на завершающих этапах алгоритма. Так как все большее количество «правильных» генов каждой хромосомы-решения становится одинаковым – скрещивание практически не дает улучшений. Эксперименты показали, что улучшить сходимость алгоритма можно за счет применения разных стратегий на разных этапах работы алгоритма: применять большее количество операций скрещивания на начальных этапах и большее количество операций мутации на завершающих этапах.

ТАБЛИЦА

share_interval	mute_prod	cross_hrod	min	max	average
5	0.5	0.5	265	370	308
10	0.5	0.5	280	380	317
20	0.5	0.5	300	420	348
5	**0.75**	0.5	270	355	315
5	**0.25**	0.5	250	430	335
5	**0.1**	0.5	280	420	354
5	0.5	**0.25**	345	410	377
5	0.5	**0.75**	220	420	378
5	0.5	**1**	210	420	300
5	**0.75**	**0.25**	370	460	419
5	**0.75**	**0.75**	230	415	273
5	**0.75**	**0.9**	210	320	**252**
5	**0.9**	0.9	235	290	255
5	**0.25**	0.9	285	500	394
5	1	1	235	415	277

Для достижения большей эффективности алгоритма экспериментально определена переходная точка смены стратегий, в которой наиболее выгодно ее менять. Значение переходной точки будем определять как отношение количества «правильных» генов в наилучшей хромосоме-решении популяции к их общему количеству *N*. На рис. 1 показаны результаты экспериментов, из которых видно, что в нашем случае значение переходной точки приблизительно равно 0.9 (соответствует $F = 10$). Эксперименты проводились при начальных значениях параметров cross_prob = 0.9, mute_prob = 0,75, init_size = 10, popul_size = 20, elite_part = 0.5 и share_interval = 5. При достижении точки перехода параметры вероятности скрещивания и мутации меняются на новые значения: cross_prob = 0,75 и mute_prob = = 0.9, а количество мутирующих генов при достижении этой точки становится равным одному независимо от номера процесса. На данном рисунке показано как снизилось среднее количество поколений с 252 до 231.

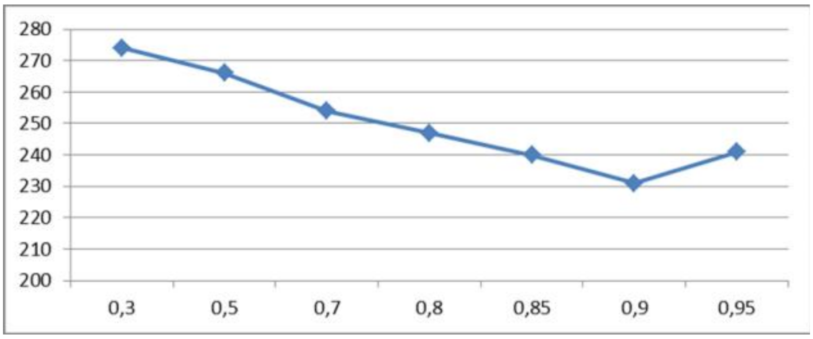

РИС. 1. Зависимость количества поколений от значения точки перехода

Сначала оценим, а затем и улучшим результат, используя дополнительные знания о разнообразности популяций, которые нам доступны при исследовании эффективности алгоритма. Для этого сопоставим каждой хромосоме-решению строку той же длины, состоящую из нулей и единиц: единица ставится в той позиции, где значения символа в хромосоме-решении совпадает с символом исходной строки, и 0 – во всех остальных позициях. Обозначим эту строку s_k^i, где i – номер популяции (WP), а k – номер хромосомы-решения в популяции.

Далее введем следующие обозначения: $\tilde{s}^i$ – строка длины *N*, полученная поэлементной дизъюнкцией всех s_k^i элитной части *i*-ой популяции $i = 1,\ldots,15$, т. е.

$$\tilde{s}^i = \bigvee_{k=1}^{elite\_part} s_k^i .$$

D^i – количество единиц в строке $\tilde{s}^i$. Таким образом, получим общее количество символов, совпадающих с символами исходной строки хотя бы в одной хромосоме-решении элитной части *i*-ой популяции $i = 1,..., 15$.

S – строка длины *N*, полученная поэлементной дизъюнкцией всех $\tilde{s}^i$, i = 1,..., 15, т. е.

$$S = \bigvee_{i=1}^{15} \tilde{s}^i .$$

В итоге *D* – количество единиц в строке *S*. Таким образом, получим общее количество символов, совпадающих с символами исходной строки хотя бы в одной хромосоме-решении элитной части хотя бы в одной популяции.

На рис. 2 показано, что при проведении экспериментов уже в начальных популяциях для строки длинной *N* = 100 и размере элитной части равного 10 значение *D* достигало максимально возможного значения 100. Однако, во время работы алгоритма, значение *D* уменьшалось до 50 (минимальное значение *D* в ходе выполнения алгоритма), что обусловлено жесткой селекцией, а также жадной и однообразной стратегией обмена.

Как уже упоминалось при описании алгоритма, в процессе выполнения шага 3, удаляются дубликаты всех хромосом-решений. Под дубликатом подразумевается хромосома-решение, которая полностью совпадает с другой, уже имеющейся в популяции. Для того, чтобы уменьшить потери уже имеющихся «правильных» генов в хромосомах-решениях, которые потенциально могут быть исключены из популяции, был введен дополнительный критерий сравнения, согласно которому хромосома-решение также исключается из популяции, если все ее «правильные» гены совпадают с другой хромосомой-решением. В среднем минимум *D* увеличился до 65, но количество поколений при этом выросло и в среднем составило 252. Вероятно, причиной является то, что при использовании критерия разнообразности строк для селекции, среднее значение fitness-функции в популяции уменьшается, соответственно результаты скрещивания чаще, по сравнению с селекцией по исходному критерию, попадают в неэлитную часть и исключаются из популяции.

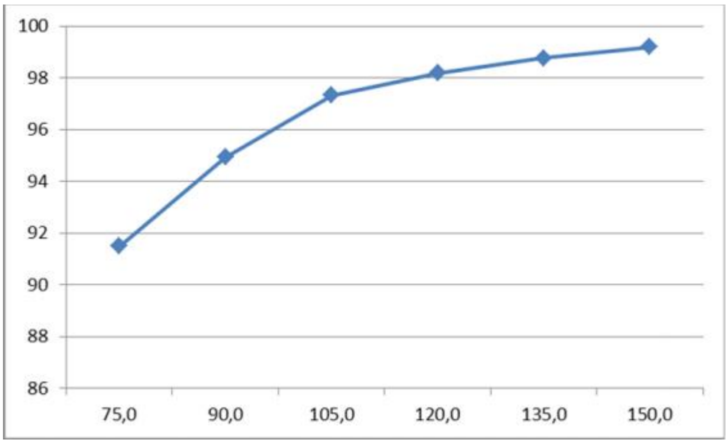

РИС. 2. Зависимость *D* для нулевых поколений от суммарного количества сгенерированных хромосом-решений

Во всех предыдущих экспериментах в качестве стратегии обмена между популяциями использовался так называемый жадный алгоритм: в каждую популяцию MP добавляет хромосому-решение, которая является лучшей для всех популяций. Данная стратегия обмена имеет тенденцию провоцировать вырождение популяции: чем с меньшим значением fitness-функции (с большим количеством «правильных» генов) хромосома-решение появляется в популяции, тем агрессивнее ее гены распространяются в популяции. И если такую хромосому-решение распространить во все популяции, в процессе скрещивания она вытеснит хромосомы-решения с большим значением fitness-функции, несмотря на то, что в них могут содержаться другие уникальные «правильные» гены.

Для избежания этого недостатка, выбрана следующая стратегия обмена: на каждый WP отправляется лучшая хромосома-решение из другой, случайно выбранной, популяции, а не лучшая среди всех. В результате минимум D увеличился до 87, и количество поколений уменьшилось до 205. Заметим, что данная стратегия не требует какой-то дополнительной внутренней (супервизорной) информации о ходе выполнения алгоритма. Конечно, существует много других стратегий обмена между процессами, но наша цель – показать принципиальную разницу между однообразными и случайными видами обмена, а именно: неэффективность распространения одинаковой хромосомы-решения во все популяции.

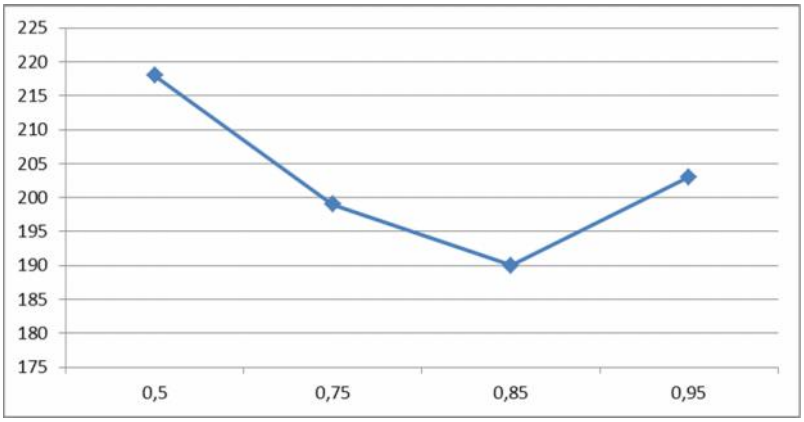

РИС. 3. Зависимость количества поколений от параметра вероятности обмена

В процессе исследования также удалось уменьшить количество поколений необходимых для достижения приемлемого решения, используя знания о количестве «правильных» генов для каждой отдельной популяции D^i. К примеру, если значение D^i больше результата fitness-функции лучшего решения для этой популяции, в нее не добавляются хромосомы-решения из других WP, так как в данной популяции не все «правильные» гены входят в лучшую хромосому-решение. В результате использования данной стратегии обмена минимум D увеличился до 92, а количество поколений уменьшилось до 190. Так как эта информация является супервизорной, введем дополнительный параметр генетического алгоритма, соответствующий этой информации – вероятность обмена.

На рис. 3 показаны результаты экспериментов с разными значениями вероятности обмена, из которых видно, что в нашем случае значение вероятности обмена 0.85 наиболее эффективно. При этом среднее количество поколений снизилось с 205 до 190.

Применяя вышеизложенные стратегии, в результате проведенных экспериментов получены также следующие показатели эффективности параллельной версии многопопуляционного генетического алгоритма в зависимости от количества параллельно развивающихся популяций, представленных на рис. 4 диаграммой.

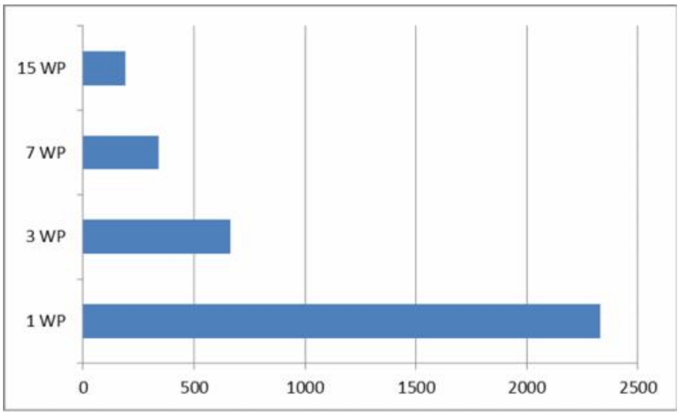

РИС. 4. Среднее количество поколений для разного количества WP

Выводы. В данной работе рассмотрены некоторые особенности параллельной реализации многопопуляционного генетического алгоритма, а также некоторые подходы к его оптимизации. Проведена экспериментальная оценка принципиально разных стратегий миграции хромосом-решений между популяциями, а также выбора вероятностей операций кроссовера и мутации в зависимости от близости к оптимальному решению. В результате достигнуто значительное (до 25 %) уменьшение количества рассмотренных вариантов решения (альтернатив).

В дальнейших исследованиях в качестве аналога поверхности реакции имитационной модели планируется использование более сложных функций, которые будут включать непрерывные и малозначимые факторы с наличием случайного влияния.

І.О. Лук'янов, Ф.А. Литвиненко О.О. Криковлюк

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ПАРАЛЕЛЬНОЇ ВЕРСІЇ БАГАТОПОПУЛЯЦІЙНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ

Розглянуті деякі особливості паралельної реалізації багатопопуляційного генетичного алгоритму, а також деякі підходи до його оптимізації. Проведена експериментальна оцінка принципово різних стратегій міграції хромосом-рішеннь між популяціями, а також вибору ймовірностей операцій кросоверу і мутації, в залежності від наближення до оптимального рішення. В результаті досягнуто значне (до 25 %) зменшення кількості розглянутих варіантів рішення (альтерантив).

I.O. Lukianov, F.A. Lytvynenko, O.O. Krykovliuk

FEATURES OF IMPLEMENTATION OF THE PARALLEL VERSION
OF MULTIPOPULATION GENETIC ALGORITHM

Features of the parallel implementation of a multi-population genetic algorithm and approaches to its optimization are considered. An experimental assessment of fundamentally different strategies of migration of chromosome-solutions between populations and the choice of probabilities of operations of crossover and mutation depending on the proximity to the optimal solution is carried out. As a result, a significant (up to 25%) reduction in the number of considered solution options (alternatives) is achieved.

Список литературы

1. Horne G.E., Meyer T.E. Data Farming: Discovering Surprise. Proc. of the Winter Simulation Conf., 2005. P. 1082 – 1087.
2. Whitley D.A. Genetic Algorithm Tutorial. *Statistics and Computing*. 1994. N 4. P. 65 – 85.
3. Whitley D. An overview of evolutoinary algorithims. *J. of Inform. And Software Techn*. 2001. **43**. P. 192 – 215.
4. Пепеляев В.А. Об эволюционных подходах к оптимизации имитационного моделирования. *Компьютерная математика*. 2005. № 1. С. 48 – 54.
5. Пепеляев В.А. Об оценке эффективности оптимизационных метаэвристических стратегий. *Теорія оптимальних рішень.* 2006. № 5. С. 16 – 22.
6. Дюк В. Data Mining – интеллектуальный анализ данных. 2003. С. 1 – 13 http://www.iteam.ru/publications/it/section_92/article_1448/.
7. Бигдан В.Б., Пепеляев В.А., Сахнюк М.А. Актуальные проблемы и тенденции в области современного имитационного моделирования. *Проблемы программирования*. 2004. № 1-2. С. 544 – 550.
8. Пепеляев В.А., Сахнюк М.А., Черний Ю.М. Параллельная реализация процессов направленного поиска оптимальних решений. *Проблеми програмування*. 2010. № 2-3. С. 572 – 576.

Получено 06.12.2018

Об авторах:

Лукьянов Игорь Олегович,
аспирант Института кибернетики имени В.М. Глушкова НАН Украины,

Литвиненко Федор Антонович,
аспирант Института кибернетики имени В.М. Глушкова НАН Украины,

Криковлюк Елена Александровна,
младший научный сотрудник Института кибернетики имени В.М. Глушкова НАН Украины.