

Рассмотрены некоторые особенности параллельной реализации многопопуляционного генетического алгоритма, а также некоторые подходы по повышению его эффективности. Проведены расчеты по генерации эффективной начальной популяции, экспериментальная оценка способов сохранения популяции, а также рассмотрены некоторые модификации генетического алгоритма для ускорения его сходимости. В результате достигнуто уменьшение количества рассмотренных альтернатив на 10 %.

© И.О. Лукьянов, Ф.А. Литвиненко,
Е.А. Криковлюк, 2019

УДК 519.711: 519.711.3: 519.81

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

О ПОВЫШЕНИИ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНОЙ ВЕРСИИ МНОГОПОПУЛЯЦИОННОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Введение. В мировой практике компьютерного моделирования сложных стохастических систем все большее распространение приобретает методология Data Farming [1]. Для некоторых задач оптимизации, решаемых на основе данной методологии, используется генетический алгоритм. Учитывая данное, исследование особенностей его применения, как для классической модели, так и различных ее модификаций (в частности, многопопуляционной версии) остается актуальным [2 – 5].

Экспериментальное исследование особенностей и эффективности многопопуляционного генетического алгоритма при планировании и реализации оптимизационно-имитационных экспериментов показывают, что вырождение генетического материала является весомым фактором, который негативно влияет на скорость сходимости к оптимальному решению [6].

Вполне естественно предположить, что избежать вырождения генетического материала можно с помощью правильно подобранных параметров и стратегий алгоритма. В этой работе представлены некоторые подходы по выбору размера начальной популяции, а также рассмотрены стратегии по эффективному использованию сгенерированной начальной популяции и сохранению ее разнообразности.

Цель данной работы – исследование некоторых стратегий сохранения полезного генетического материала начальной популяции для уменьшения количества рассмот-

ренных альтернатив решений и повышения скорости сходимости алгоритма.

В настоящей работе приняты следующие допущения:

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

– при исследовании на данном этапе считаем, что значения, возвращаемые фитнес-функцией, являются детерминированной величиной (а не случайной величиной, 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 L , состоящую из символов конечного алфавита размерности a . Фитнес-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 $F = 0$.

Используемый алгоритм, его параметры, а также их значения более подробно описаны в [6].

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 являются управляемыми (worker process – WP). На WP реализованы разные параллельно развивающиеся популяции, которые выступают в качестве «островов» в многопопуляционной модели генетического алгоритма.

Также приняты некоторые термины:

- «правильный» ген – ген, значение которого совпадает со значением гена в эталоне на соответствующей позиции;
- начальная популяция – множество всех ненулевых хромосом-решений для всех процессов;
- полным генетическим материалом назовем набор генов, входивших в начальную популяцию, значение которых совпадает со значениями генов эталона на соответствующих позициях.

Генерация начальной популяции. Для генерации по возможности более полного генетического материала необходимо выбрать соответствующее значение размера начальной популяции. Значение этого параметра, для данной задачи, можно обосновать математически.

Для соответствующих расчетов будем использовать следующие значения: длина искомой строки $L = 100$, размер начальной популяции на каждом процессоре $l = 10$, число популяций (процессоров) $k = 15$, число символов конечного алфавита $a = 33$. Соответственно вероятность случайного выбора правильного символа для конкретной позиции искомой строки при равномерном распреде-

лени будет равна $p = 1/a = 1/33$, а число независимых испытаний по каждой позиции искомой строки при случайной генерации начальных популяций (общее количество сгенерированных строк): $kl = 150$.

Тогда вероятность P_a выбора правильного символа в заданной позиции искомой строки хотя бы один раз при генерации всех начальных популяций равна

$$P_a = 1 - (1 - p)^{kl} = 1 - (1 - p)^{150} = 0,990.$$

Под оптимальным будем понимать такой размер начальной популяции, при котором после генерации начальных популяций, вероятность $P(x)$ совпадения символа, хотя бы для одной особи в произвольных x (близких к значению L) позициях искомой строки, будет с заданной точностью равна единице. Эта вероятность в нашем случае вычисляется следующим образом:

$$P(x) = C_L^x P_a^x (1 - P_a)^{L-x} = \frac{L!}{x! (L-x)!} P_a^x (1 - P_a)^{L-x}.$$

Определим соотношение между $P(x)$ и $P(x-1)$:

$$P(x-1) = C_L^{x-1} P_a^{x-1} (1 - P_a)^{L-x+1} = \frac{L!}{(x-1)! (L-x+1)!} P_a^{x-1} (1 - P_a)^{L-x+1},$$

$$\frac{P(x-1)}{P(x)} = \frac{x! (L-x)!}{(x-1)! (L-x+1)!} \times \frac{1 - P_a}{P_a} = \frac{x}{L-x+1} \times \frac{1 - P_a}{P_a},$$

$$P(x-1) = \frac{x}{L-x+1} \times \frac{1 - P_a}{P_a} \times P(x).$$

При $x = L$: $P(L) = P_a^{100} = 0,366$.

Посчитав вероятность совпадения символов в интервале от 97 до 100 позиций искомой строки после генерации всех начальных популяций, можно вычислить их сумму:

$$P(100) + P(99) + P(98) + P(97) = 0,366 + 0,370 + 0,185 + 0,061 = 0,982.$$

Столь высокая вероятность обеспечивает генерацию достаточно полного генетического материала практически во всех экспериментах.

Стратегии сохранения генетического материала на начальном этапе.

В работе [6] приведены результаты, показывающие, что до некоторой точки работы алгоритма операции скрещивания обеспечивают достаточно высокую скорость схождения к оптимуму. В то же время на заключительном этапе алгоритма важное значение приобретают операции мутации.

Эксперименты с использованием данного алгоритма показывают, что чем ближе к оптимуму найденное решение, тем медленнее оно к нему приближается. На рис. 1 показано, что на последние 10 % оптимума приходится около 40 %

всех вычислений. Такое увеличение количества итераций связано с тем, что последние гены для нахождения оптимального решения получены исключительно с помощью операции мутации, вероятность успеха которой крайне мала, если сравнивать ее с вероятностью неудачи данной операции.

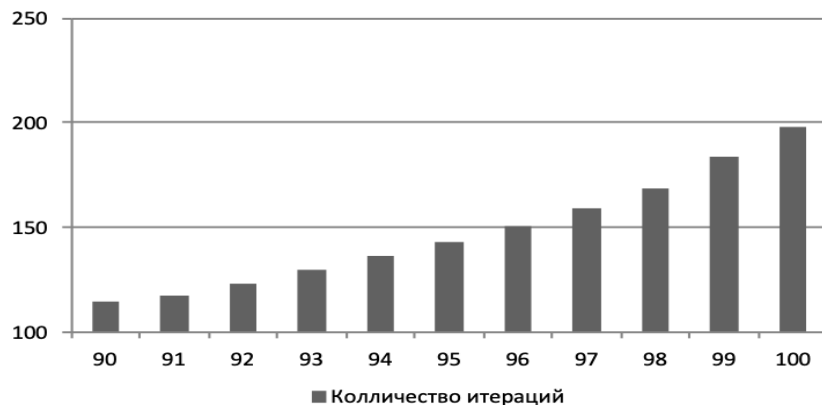


РИС. 1

Экспериментальным путем было определено, что при использовании исключительно операций скрещивания, данный алгоритм в среднем достигает 84 % от оптимума. Если сгенерированная начальная популяция содержит около 100 % генов оптимального решения, логично предположить, что путем направленного скрещивания и селекции можно улучшить процент оптимума, достигнутого исключительно операциями скрещивания, что в свою очередь должно уменьшить количество итераций в целом.

Попробуем модифицировать стратегии алгоритма с целью более эффективного использования генетического материала начальной популяции и сохранения его разнообразности.

Можно выделить следующие причины потерь полезного генетического материала:

- потери при селекции: уникальный правильный ген может попасть в хромосому с малым значением фитнес-функции, которая может быть удалена из популяции в результате селекции;

- потери при скрещивании: при скрещивании мы последовательно случайным образом определяем предка, ген которого будет стоять на текущем месте у потомка. Таким образом, если уникальный правильный ген не будет выбран, он не будет распространен по популяции и может быть потерян в результате селекции.

Для уменьшения потерь при селекции на ранних этапах работы алгоритма модифицируем его следующим образом.

Одним из способов решения проблемы потерь при скрещивании может быть порождение двух «противоположных» или «зеркальных» потомков, гены которых полностью не совпадают между собой. Таким образом, каждый правильный

ген каждого предка будет передан в сгенерированную хромосому-решение (рис. 2). При этом одна из результирующих хромосом-решений гарантированно даст такое же или лучшее значение фитнес-функции, по сравнению с ее родителями, и с большей вероятностью останется в популяции. В результате данного подхода количество итераций алгоритма (поколений) уменьшилось со **195** до **180**.

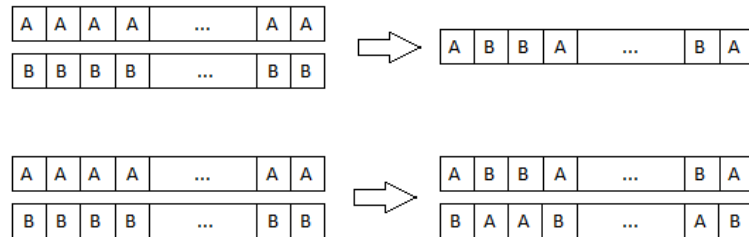


РИС. 2

Так как для нахождения последних генов очень важны операции мутации, логично предположить, что они дадут лучший результат, если проводить их с лучшей хромосомой-решением в популяции, по сравнению с проведением мутации с любой другой из элитной части. Использование этой стратегии на заключительном этапе алгоритма, в комбинации с применением «противоположного» скрещивания, уменьшило количество итераций до **170**.

Добавим, что если при скрещивании перезаписывать всю популяцию, т. е. заменять все имеющиеся хромосомы-решения на полученные сгенерированные хромосомы-решения, то можно практически избежать потери «правильных» генов во время селекции. Исключением будут являться случаи, когда в популяцию попадает хромосома-решение из другой популяции. При использовании данной стратегии важно, чтобы хромосомы-решения использовались во время скрещивания только один раз. Для этого можно использовать различные стратегии их выбора. Рассмотрим несколько вариантов:

– поочередное скрещивание – первая хромосома скрещивается со второй, третья с четвертой и т. д.;

– сбалансированное скрещивание – первая хромосома скрещивается с последней, вторая с предпоследней и т. д.

Для наглядности эксперимента вероятность мутации на начальном этапе алгоритма была равна 0, параметр точки перехода [6] установлен на уровне достижения 90 % оптимума, вероятность мутации на заключительном этапе устанавливалась равной 0.75.

На рис. 3 показано сравнение данных стратегий скрещивания с произвольной стратегией «противоположного» скрещивания. Здесь D – общее значение

«правильных» генов присутствующих во всей популяции, F – значение фитнес-функции для лучшей хромосомы-решения во всей популяции.

Поочередное скрещивание демонстрирует хорошие результаты по сохранению генетического материала, показывая значение D в окрестности 100 %. Однако, так как все предыдущие хромосомы-решения заменяются на сгенерированные, в селекции не участвуют хромосомы-решения из предыдущих итераций алгоритма, что значительно понижает скорость схождения к оптимуму.

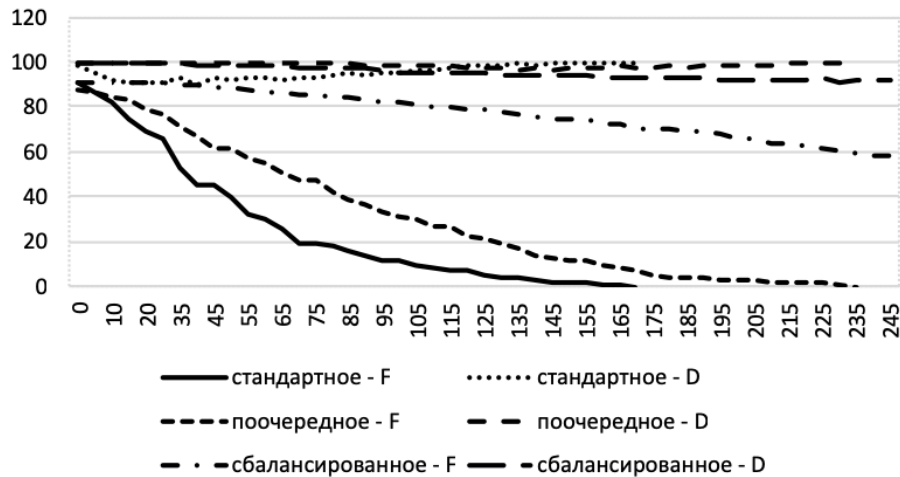


РИС. 3

Сбалансированное скрещивание, в свою очередь, так же показывает хорошие результаты по сохранению популяции, но так как для скрещивания используются «лучшая» и «худшая» хромосомы-решения, в среднем скрещивание не улучшает результата, который уже имелся в популяции. Это приводит к медленному схождению к оптимуму.

Выводы. В данной работе рассмотрены некоторые особенности параллельной реализации многопопуляционного генетического алгоритма, а также некоторые подходы к повышению его эффективности. Проведены расчеты вероятности генерации эффективной начальной популяции, экспериментальная оценка способов сохранения популяции, а также рассмотрены некоторые модификации генетического алгоритма для ускорения его сходимости. В результате достигнуто уменьшение количества рассмотренных вариантов решения (альтернатив) на 10 %. В дальнейших исследованиях в качестве аналога поверхности реакции имитационной модели планируется использование более сложных функций, которые будут включать непрерывные и малозначимые факторы, а также наличие случайного влияния.

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

**ПРО ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПАРАЛЕЛЬНОЇ ВЕРСІЇ
БАГАТОПОПУЛЯЦІЙНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ**

Розглянуті деякі особливості паралельної реалізації багатопопуляційного генетичного алгоритму, а також деякі підходи щодо підвищення його ефективності. Проведено розрахунки по генерації ефективної початкової популяції, експериментальна оцінка способів збереження популяції, а також розглянуті деякі модифікації генетичного алгоритму для прискорення його збіжності. В результаті досягнуто зменшення кількості розглянутих альтернатив на 10 %.

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

**ABOUT INCREASING THE EFFICIENCY OF THE PARALLEL VERSION
OF A MULTIPOPULATION GENETIC ALGORITHM**

In this paper, we consider some features of parallel implementation of a multi-population genetic algorithm, as well as some approaches to improve its efficiency. Calculations were carried out on the generation of an effective initial population, an experimental assessment of the methods of preserving the population, and also some modifications of the genetic algorithm to accelerate its convergence were considered. As a result, a decrease in the number of alternatives considered by 10 % was 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. № 4. P. 65 – 85.
3. Whitley D. An overview of evolutionary algorithms. *J. of Inform. And Software Techn.* 2001. N 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. *Теория оптимальных решений.* 2006. № 5. С. 16 – 22.
6. Литвиненко Ф.А., Лукьянов И.О., Криковлюк Е.А. Особенности реализации параллельной версии многопопуляционного генетического алгоритма. *Компьютерная математика.* 2019. № 2. С. 21 – 29.

Получено 25.03.2019