УДК 519.71

О.Г. Руденко, С.В. Мирошниченко, А.А. Бессонов

Программирование с экспрессией генов: модификации эволюционного процесса

Проанализированы некоторые модификации эволюционного процесса, используемого в программировании с экспрессией генов, направленные на улучшение свойств традиционного алгоритма. Результаты экспериментов свидетельствуют о том, что использование рассмотренных модификаций позволяет существенно повысить качество решений. Дальнейшие исследования целесообразно направить на разработку эффективных методов многопроцессорной реализации алгоритмов программирования.

Проаналізовано деякі модифікації еволюційного процесу, який застосовується у програмуванні з експресією генів і спрямовані на покращення властивостей традиційного алгоритму. Результати експериментів свідчать про те, що використання розглянутих модифікацій дозволяє суттєво підвищити якість розв'язків. Подальші дослідження доцільно спрямувати на розробку ефективних методів багатопроцесорної реалізації алгоритмів програмування.

Введение. На основании анализа и результатов моделирования существующих методов реализации алгоритма программирования с экспрессией генов (ПЭГ) в работах [1, 2] был выявлен ряд ограничений производительности стандартного алгоритма: длительность вычисления фитнеса, отсутствие тонкой подстройки числовых констант, влияние размера хромосомы на скорость сходимости, проблемы с поиском сложных моделей. Данная статья является продолжением указанных работ и посвящена исследованию методов, направленных на преодоление упомянутых ограничений. В ней используется методика оценки эффективности модификаций алгоритма ПЭГ, изложенная в [1] и заключающаяся в оценивании СКО наилучшей модели среди всех запусков (e_b,) и доли успешных запусков (r_f) на основе статистической обработки результатов множества запусков. На тех же моделях: синусоида (*sin*), функция Розенброка (*rosen*) и сумма четырех сигмоид (*sigmas*).

Обеспечение разнообразия начальной популяции

Для наиболее эффективного исследования пространства поиска особи начальной популяции должны быть как можно менее похожи между собой, однако создание начальной популяции случайным образом не предполагает каких-либо процедур по обеспечению генетического разнообразия.

Сравнение особей наиболее удобно проводить по генотипу вследствие его линейности и легкости считывания. При этом целесообразно сравнивать только кодирующие участки, участвующие в построении синтаксического дерева.

Для измерения близости между хромосомами в качестве метрики [3] предлагается придерживаться правила, возвращающего максимальное количество r следующих подряд совпадающих элементов. Если две сравниваемые последовательности равны между собой, r будет равно длине последовательности. Два гена считаются близкими, если значение r превышает определенный порог (авторами использовано значение *семь*). Процедура создания начальной хромосомы с использованием описанной техники выглядит так:

Шаг 1. Создание пустой популяции.

Шаг 2. Создание новой особи случайным образом.

Шаг 3. Сравнение этой особи с каждой особью, добавленной в популяцию.

Шаг 4. Если новая особь близка какой-либо особи в популяции, перейти к шагу 2, иначе добавить особь в популяцию.

Шаг 5. Если популяция полностью заполнена, завершить процедуру, иначе перейти к шагу 2.

Эксперименты по моделированию различных наборов данных показали, что полученные модели обладают большей корреляцией с тестовыми данными, чем результаты работы стандартного алгоритма ПЭГ.

Модификации эволюционного процесса

Возврат в исходное состояние. Когда эволюционный процесс охватывает определенное количество поколений, среднее значение фитнес-функции (ФФ) популяции становится достаточно большим, однако при этом практически устраняется разнообразие, что приводит к

преждевременной сходимости в локальный оптимум и уменьшает шансы успешной глобальной оптимизации. Для решения этой проблемы в [4] предложено использовать явление атавизма – проявление свойств далеких предков.

Современная генетика описывает следующие причины возникновения атавизма: рекомбинация утраченного гена предка в результате скрещивания или мутации и устранение стопового элемента генома, заблокировавшего на определенном этапе экспрессию гена предка. Кратковременно развернуть процесс эволюции в обратном направлении можно путем возврата популяции ПЭГ в исходное состояние (*Back-traced GEP*).

В основе алгоритма возврата лежит структура данных *стек*, хранящая *контрольные точки* – состояния популяции. Если значения ФФ лучшей особи в новой популяции (полученной в результате репликации и применения генетических операторов) выше, чем у лучшей особи в популяции на верхушке стека (в последней контрольной точке), это означает правильное направление эволюции, и новая популяция вталкивается в стек, формируя новую контрольную точку. В противном случае можно сделать вывод о тупиковой ветви эволюции, поэтому последняя контрольная точка выталкивается из верхушки стека.

Применение методики возврата к предыдущему состоянию привело к значительному улучшению качества получаемых решений [4].

Группировка особей и их параллельная эволюция. Начиная с определенного поколения, популяция, находясь в поздней фазе эволюции, приобретает следующие признаки: сложную структуру синтаксических деревьев, замедление эволюции, незначительное разнообразие популяции. Однако замечено [5], что избежать преждевременной сходимости позволяет разбиение популяции на группы и их параллельная обработка.

Происходит разбиение популяции на группы следующим образом. Особи популяции сортируются в порядке возрастания значений ФФ:

$$G = \left\{ I_i \mid f\left(I_{i+1}\right) \geq f\left(I_i\right), 1 \leq i \leq N \right\}.$$

Группой считается последовательность, в которой разница значений ФФ соседних особей не превышает заданную величину d

$$G_i = \left\{ I_j \mid f\left(I_{j+1}\right) - f\left(I_j\right) \leq d, 1 \leq j \leq N, 1 \leq i \leq Q \right\},$$

где Q — количество образовавшихся групп. Под плотностью группы понимается отношение ее размера (количества особей) к размеру популяции $p_i = \dfrac{|G_i|}{N}$.

Очевидно, для групп одной популяции всегда выполняется $\sum_{i=1}^{Q} p_i = 1$.

Энтропия популяции, математическое ожидание и дисперсия ФФ вычисляются соответственно так:

$$E(G) = -\sum_{i=1}^{Q} p_i \times \log p_i;$$

$$M(G) = \sum_{i=1}^{Q} \hat{f}_i \times p_i;$$

$$D(G) = \frac{1}{N} \times \sum_{i=1}^{N} f(I_i) - \hat{f}^2,$$

где

$$\hat{f}_i = \frac{1}{|G_i|} \times \sum_{i=1}^{|G_i|} f(I_i); \quad \hat{f} = \frac{1}{N} \times \sum_{i=1}^{N} f(I_i).$$

Если энтропия и дисперсия популяции меньше определенных заданных пороговых значений, то можно сделать вывод о недостаточной степени генетического разнообразия. В этом случае целесообразно заменить 10 процентов худших особей новыми, созданными случайным образом. Указанные операции необходимо применять к каждому поколению. Итоговый алгоритм работы имеет вид:

До тех пор, пока не достигнуто максимальное поколение, выполнять
- Создание начальной популяции;
- Разбиение популяции на группы;
- ***для каждой*** *группы популяции* ***выполнять***
  - Перенос лучшей особи;
  - Применение генетических операторов;
- ***конец цикла***
- Расчет энтропии и дисперсии популяции;
- Замена худших особей популяции при необходимости;
- Разбиение популяции на группы;
- ***конец цикла***
- Вывести лучшую особь.

Влияние отдельно взятой модификации, удаляющей худшие особи, показано в табл. 1. Из этих результатов следует, что такая модификация, с одной стороны, улучшает точность наилучшей возможной модели при большом количестве запусков, а с другой – снижает вероятность обнаружения приемлемого решения.

Таблица 1. Эффективность алгоритма с заменой худших особей

Эксперимент	*sin*		*Rosen*		*sigmas*	
	e_b	r_f	e_b	r_f	e_b	r_f
Традиционный алгоритм	0,489	0	0,173	0	0,165	0
Замена худших	0,000	100	0,173	0	0,165	0

Популяции неоднородных особей

Одна из проблем алгоритма ПЭГ – определение оптимального размера головы гена (и размера решения). Отсутствие процедуры априорного задания обусловливает необходимость запуска алгоритма многократно с разными параметрами для поиска наиболее подходящих. Для решения этой проблемы в [6] предложено использовать в одной популяции хромосомы различной длины: половина популяции заполняется особями пропорционально с диапазоном размеров, заданным пользователем, а длина генов особей второй половины устанавливается случайным образом. Операторы рекомбинации в таком случае применяются только к особям с хромосомами равной длины.

Подстройка констант осуществляется посредством градиентного алгоритма, неудобного при вычислениях и поэтому применяемого с определенной вероятностью. Константа либо заменяется случайно выбранной, либо изменяется в пределах 10 процентов. Если мутированная особь лучше исходной, то заменяет ее, иначе константа заменяется полусуммой последних двух значений. Процесс повторяется до тех пор, пока мутировавшая особь не станет хуже исходной либо когда будет достигнут предел в 10 итераций.

Аналогичные исследования проведены в работе [7], где предложено использовать хромосомы с переменным числом генов и переменным размером каждого из них. В этой работе введены новые операторы, изменяющие длину генома:

- удаление одного гена из хромосомы;
- создание и добавление одного нового гена в хромосому;
- перенос участка головы одного гена в голову другого, что приводит к укорачиванию первого и удлинению второго;
- рекомбинация разнородных хромосом.

Такой подход привел к двукратному сокращению длины гена и, соответственно, размера синтаксического дерева решения.

Дополнительная популяция

В качестве одной из мер повышения вероятности обнаружения решения в работах [8, 9] было предложено использование дополнительной параллельной независимой популяции. Если при очередной итерации (поколении) работы алгоритма значение ФФ лучшей особи дополнительной популяции превысит значение ФФ лучшей особи основной популяции, данная особь копируется на место худшей особи основной популяции. Как следует из результатов, представленных в табл. 2, использование данного подхода приведет к решению задачи, снижая при этом точность лучшей из получаемых моделей. Это происходит вследствие возрастания времени вычислений в расчете на итерацию алгоритма, т.е. при равном отводимом времени работы модифицированный алгоритм рассчитывает меньшее количество поколений.

Таблица 2. Эффективность алгоритма с дополнительной популяцией

Алгоритм	*sin*		*rosen*		*sigmas*	
	e_b	r_f	e_b	r_f	e_b	r_f
Традиционный	0,350	0	0,289	0	0,165	0
Доп. популяция	0,000	100	0,289	0	0,165	0

Инкрементальная эволюция

При исследовании методов кодирования в [1] получена таблица, в которой очевидно наличие некоторого оптимального размера генома, обеспечивающего достижение максимума производительности алгоритма и требующего определения для каждой задачи и каждого способа кодирования. Выбор этого размера может осуществляться как простым перебором параметров, так и посредством описанных популяций неоднородных особей.

Более эффективным оказался метод инкрементальной эволюции [10, 11], в ходе которого

выполняется ряд последовательных запусков алгоритма с наращиванием длины хромосомы при каждом запуске и копированием лучшей особи предыдущего запуска в начальную популяцию текущего. Для такого подхода требуется меньше вычислительных ресурсов в сравнении с независимыми запусками, так как можно проводить постепенное усложнение дерева. Результаты исследования данной модификации приведены в табл. 3.

Таблица 3. Эффективность алгоритма при традиционной и инкрементальной эволюции

Эволюция	*sin*		*rosen*		*sigmas*	
	e_b	r_f	e_b	r_f	e_b	r_f
Традиционная	0,073	50	0,076	0	0,164	0
Инкрементальная	0,073	50	0,129	0	0,096	0

Комбинирование множества запусков алгоритма

Взвешенная сумма моделей. Полученные в ходе нескольких запусков алгоритма ПЭГ модели можно обобщить в одну [12, 13], представляя итоговую формулу

$$M = aM_1 + bM_2 + cM_3 + \ldots,$$

где a, b, c, ... – весовые коэффициенты; M_1, M_2, M_3, ... – модели, полученные в ходе запусков ПЭГ.

Для подбора весовых коэффициентов с целью минимизации погрешности итоговой модели, повышению ее корреляции с выборками данных в этих работах использован генетический алгоритм *NSGA* II (*Non-Dominated Sorting Genetic Algorithm* II).

Как правило, обобщающая модель обладает лучшими характеристиками, чем ее составляющие в отдельности.

Итеративный разностный подход. Как уже отмечалось, одна из наиболее распространенных задач, для решения которой применяется ПЭГ, – символьная регрессия (поиск математической формулы, описывающей набор численных данных) либо, в более простом варианте, аппроксимация функций. В ряде случаев сложность моделируемого объекта такова, что обеспечить приемлемую точность при описании компактной формулой невозможно, и требуется увеличение размера искомого дерева, что приводит к резкому увеличению пространства поиска, а следовательно, и времени вычислений.

Эффективным средством повышения сложности формулы с минимальным влиянием на производительность алгоритма служит развитие идеи сложных мультигенных хромосом с иерархической структурой, описанные ранее.

Еще более действенным оказался разработанный в [14, 15] разностный подход, основанный на комбинировании синтаксических деревьев, когда математические формулы легко могут быть объединены, например, при помощи функции арифметического сложения, правила классификаторов – булевыми «И» и «ИЛИ» и др. Эта особенность позволяет составить сложную формулу из ряда простых, компактных и быстро вычисляемых в отдельности.

Суть подхода заключается в последовательном применении алгоритма ПЭГ с неизменным набором параметров к поверхностям ошибки, представляющим собой наборы численных данных, образованных в результате вычитания очередной полученной модели из моделируемых данных. Таким образом, разностный подход принципиально отличается от идеи эволюции мультигенных хромосом, где алгоритм пытается найти решение с первой же итерации. При первом запуске на вход алгоритма ПЭГ подается набор данных $T(O_0)$ с ожиданием на выходе модели M, возвращающей набор данных O

$$\{M_1, O_1\} = GEP(T = O_0).$$

На каждом следующем этапе на вход подается разность моделей

$$\{M_{i+1}, O_{i+1}\} = GEP(O_i - O_{i-1}).$$

Итоговая модель после N запусков такова:

$$M = M_1 + M_2 + \ldots + M_N.$$

Результаты моделирования, свидетельствующие о положительном влиянии разностного подхода на показатели алгоритма ПЭГ, приведены в табл. 4. Следует, однако, учитывать, что при этом время, затрачиваемое алгоритмом на поиск каждого слагаемого формулы модели, возрастает пропорционально количеству разностей.

Таблица 4. Эффективность алгоритма с разностным подходом

Эксперимент	*sin*		*rosen*		*sigmas*	
	e_b	r_f	e_b	r_f	e_b	r_f
Традиционный алгоритм	0,095	0	0,081	0	0,134	0
Разность 1	0,097	0	0,074	0	0,122	0
Разность 2	0,063	50	0,079	0	0,082	0
Разность 3	0,104	0	0,070	0	0,118	0

Параллелизация. Реализация традиционного алгоритма требует значительных вычислительных ресурсов. Самым ресурсоемким есть этап расчета приспособленности программы, так как эту операцию следует выполнять для каждой особи популяции по всему обучающему набору входных и выходных данных на каждой итерации алгоритма. Как правило, ФФ в ПЭГ при решении задачи регрессии основывается на среднеквадратичном отклонении.

Для ускорения процесса целесообразно использовать такой ресурс компьютера, как наличие нескольких процессоров. Один из способов автоматизации распараллеливания программ – использование библиотеки, реализующей стандарт *OpenMP*.

При расчете ФФ особи не возникает зависимости по данным от остальных особей популяции. Доступ к входным и выходным значениям восстанавливаемой функции предоставляется только на чтение, что позволяет обезопасить разделяемые данные. Выполнение этих условий необходимо и достаточно для эффективного распараллеливания расчета ФФ популяции – к циклу программы можно добавить соответствующую директиву компилятора. Кроме того, распараллеливанию поддается и эволюционный этап алгоритма.

В таком алгоритме значительная доля общего объема вычислений может быть получена параллельными расчетами, что позволяет добиться ускорения выполнения в симметричных многопроцессорных системах [16].

Заключение. Проанализированные в статье некоторые модификации эволюционного процесса, используемого в ПЭГ, направлены на улучшение свойств традиционного алгоритма. Результаты экспериментов свидетельствуют о том, что использование рассмотренных модификаций позволяет зачастую добиться существенного улучшения качества получаемых решений.

Дальнейшие исследования целесообразно направить на разработку эффективных методов многопроцессорной реализации алгоритмов ПЭГ.

1. *Руденко О.Г.*, *Мирошниченко С.В.*, *Бессонов А.А.* Программирование с экспрессией генов: способы кодирования и создания синтаксических деревьев // УСиМ. – 2015. – № 3. – С. 82–92.
2. *Руденко О.Г.*, *Мирошниченко С.В.*, *Бессонов А.А.* Программирование с экспрессией генов: генетические операторы // УСиМ. – 2015. – № 4. – С. 72–82.
3. *The Strategies* of Initial Diversity and Dynamic Mutation Rate for Gene Expression Programming / L. Duan, C. Tang, J. Zhu et al. // Proc. of the Third Int. Conf. on Natural Computation. – **04**. — ICNC ’07. – Washington, DC, USA: IEEE Comp. Society, 2007. – P. 265–269.
4. *Improve* KDD Efficiency of Gene Expression Programming by Backtracking Strategy / Y. Zhong, C. Tang, Y. Chen et al. // Journal-Sichuan university natural science edition. – 2006. – **43**, № 2. – P. 299–304.
5. *Mining* Projection Transformation Based on Gene Expression Programming of Multi-Variable Niches / Y. Jiang, C. Tang, H. Zheng et al. // Proc. of the 2008 Fourth Int. Conf. on Natural Computation – **06**. – ICNC ’08. – Washington, DC, USA: IEEE Comp. Society, 2008. – P. 288–292.
6. *Lopes H.S.*, *Weinert W.R.* EGIPSYS: an Enhanced Gene Expression Programming Approach for Symbolic Regression Problems // Int. J. of Applied Mathematics and Computer Science. – 2004. – **14**, № 3. – P. 375–384.
7. *Browne N.P.A.*, *dos Santos M.V.* Adaptive Representations for Improving Evolvability, Parameter Control, and Parallelization of Gene Expression Programming // Applied Comp. Int. Soft Comp. – 2010. – **2010**. – 19 p. Action ID 409045
8. *Руденко О.Г.*, *Мирошниченко С.В.* Об одной модификации алгоритма программирования с экспрессией генов в задаче аппроксимации функции // Вестн. Херсонского нац. техн. ун-та. – 2013. – № 1(46). – С. 90–94.
9. *Руденко О.Г.*, *Мирошниченко С.В.* Моделирование нелинейных объектов с помощью алгоритмов программирования с генной экспрессией // Информатика, математическое моделирование, экономика. – 2013. – Т. 2. – С. 142–146.
10. *Руденко О.Г.*, *Мирошниченко С.В.* Об одной модификации алгоритма программирования с экспрессией генов в задаче символьной регрессии // Стратегия качества в промышленности и образовании: Материалы IX междунар. конф. – ГИПОпром, 2013. – С. 511–514.

11. *Руденко О.Г.*, *Мирошниченко С.В.* Построение моделей нелинейных объектов на основе нейроэволюционного подхода // Современные направления развития информационно-коммуникационных технологий и средств управления: Материалы третьей междунар. науч.-техн. конф. – Полтава: ПНТУ; Белгород: БГУ; Харьков: ГП «ХНИИ ТМ» К.: НТУ «КПИ»; Кировоград: КЛА НАУ, 2013. – С. 47–48.
12. *Abraham A.*, *Grosan C.* Decision Support Systems Using Ensemble Genetic Programming // JIKM. – 2006. – **5**, № 4. – P. 303–313.
13. *A Novel* Function Regression Algorithm Based on Gene Expression Programming Ensembles / Z. Guo, Z. Wu, X. Dong et al. // Int. J. of Advancements in Comp. Techn. – 2012. – **4**, № 1. – P. 239–247.
14. *Руденко О.Г.*, *Мирошниченко С.В.* Применение разностного подхода в программировании с экспрессией генов // Проблемы информатизации: Материалы второй междунар. научн.-техн. конф. – К.: ГУТ; Полтава: ПНТУ; Катовице: Катовицкий экон. ун-т; Париж: Ун-т Париж VII Винсент-Сен-Дени; Белгород: БГУ; Черкассы: ЧГТУ; Харьков: ХНГИТМ, 2014. – С. 53–54.
15. *Руденко О.Г.*, *Мирошниченко С.В.* Повышение качества моделей, получаемых с помощью программирования с экспрессией генов: Сб. науч. ст. по итогам 4-й Междунар. науч.-прак. конф. «Информатика, математическое моделирование, экономика», Смоленск, 2014. – Т. 1. – С. 195–199.
16. *Руденко О.Г.*, *Мирошниченко С.В.* Параллелизация алгоритма программирования с генной экспрессией для выполнения в симметричных многопроцессорных системах // Автоматизация: проблемы, идеи, решения: Материалы междунар. научн.-техн.конф. – Севастополь: СевНТУ, 2013. – С. 13–15.

Поступила 13.07.2015
Тел. для справок: +38 057 702-1354 (Харьков)
E-mail: *o.bezsonov@gmail.com*, *S.Miroshnichenko@gmail.com*

UDC 519.71

O.G. Rudenko, S.V. Miroshnichenko, O.O. Bezsonov

Gene Expression Programming: Modifications of the Evolutionary Process

Based on the analysis and modeling results of the existing implementations of gene expression programming algorithm (GEP), a number of performance limitations of the standard algorithms such as duration of the fitness computation, lack of the numerical constants fine-tuning, impact of size on the rate of chromosome convergence and some other problems related to finding complex models have been identified.

In this paper some modifications of the evolutionary process that is used in the GEP to improve the properties of the conventional algorithm were analyzed. For example, it is proposed to use chromosomes with a variable number of genes and with a variable size of each gene. This approach has led to halving of the gene length and, consequently, the size of the solutions' syntactical trees. It is shown that the implementation of the traditional algorithm requires considerable computing resources. To speed up the process it is proposed to use such a computer resource as multiple processors. For automated parallelization of software it is recommended to use the library that implements OpenMR standard.

The experimental results indicate that the use of the considered modifications often helps to achieve significant improvement in the quality of the solutions. It seems appropriate to direct further research to development of the effective methods for implementation of multiprocessor GEP algorithms.

●