

10. Гладков Л.А., Курейчик В.В., Курейчик В.М., Родзин С.И. Основы теории эволюционных вычислений. – Ростов-на-Дону: Изд-во ЮФУ, 2010
11. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. – М.: Физматлит, 2006.
12. Курейчик В.М., Кажаров А.А. Использование роевого интеллекта в решении NP-трудных задач // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 30-36.
13. Курейчик В.В., Запорожец Д.Ю. Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 28-32.
14. Лебедев О.Б. Решения комбинаторных задач на графах на основе метода муравьиной колонии // Труды конгресса по интеллектуальным системам и информационным технологиям «AIS- IT'09». Научное издание в 4-х томах. Т.1. – М.: Физматлит, 2009. – С. 51-58.
15. Курейчик В.М., Кажаров А.А. Применение пчелиного алгоритма для раскраски графов // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 30-36.
16. Лебедев Б.К., Лебедев В.Б. Размещение на основе метода пчелиной колонии // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 12-19.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.О. Чернышев.

Лебедев Владимир Борисович – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: lbk@tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44, ГСП 17А; тел.: 88634371743; кафедра системного анализа и телекоммуникаций; доцент.

Lebedev Vladimir Borisovich – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: lbk@tsure.ru; GSP 17A, 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +78634371743; the department of system analysis and telecommunications; associate professor.

УДК 004.8.023 + 004.81.85

С.И. Родзин

ОБОБЩЕННАЯ МОДЕЛЬ БАЗОВОГО ЦИКЛА ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ*

Вычисления, инспирированные природными системами и процессами, являются междисциплинарным направлением и включают три взаимосвязанных области: биоинформатика, вычисления на основе молекулярных, ДНК и квантовых компьютеров, а также эволюционные вычисления. В статье рассматривается базовый цикл эволюционных вычислений. Это математические преобразования, позволяющие трансформировать входной поток информации в выходной по правилам, имитирующим механизм эволюции. Приводится алгоритм организации эволюционных вычислений в базовом цикле и его модификация, а также реализация базового цикла эволюционных вычислений в алгоритме табуцированного поиска.

Эволюционные вычисления; базовый цикл; алгоритм табуцированного поиска.

S.I. Rodzin

GENERALIZED MODEL OF BASIC CYCLE EVOLUTIONARY COMPUTATION

The calculations are inspired by natural systems and processes are interdisciplinary focus and includes three interrelated areas: bioinformatics, molecular computing-based, DNA and quantum computers, as well as evolutionary computation. This article discusses the basic cycle of

* Работа выполнена при поддержке гранта РФФИ 11-07-00094-а.

evolutionary algorithms. This is a mathematical transformation, allowing the flow of information to transform the input into output by the rules, simulating the mechanism of evolution. An algorithm for the organization of evolutionary computation in the basic cycle and its modification, as well as the implementation of the basic cycle of evolutionary computation in the algorithm tabu search.

Evolutionary computation; basic cycle; tabu search algorithm.

Введение. Настоящее время вычисления, инспирированные природными системами и процессами, являются междисциплинарным направлением и включают три взаимосвязанных области: биоинформатика [1], вычисления на основе молекулярных [2], ДНК [3] и квантовых компьютеров [4], а также эволюционные вычисления [5].

Биоинформатика (или вычислительная биология) включает в себя биологические исследования с использованием компьютерных экспериментов. Здесь целью исследований является использование информационных технологий для моделирования биологических систем, чтобы повторить и лучше понять их поведение. Работы в биоинформатике ориентированы на моделировании природных явлений и включают исследования искусственной жизни, фрактальной геометрии, а также клеточных автоматов.

Второе направление, вычислений, инспирированных природными системами, связано с организацией вычислений с использованием компьютеров, построенных не на кремниевой основе (молекулярные, ДНК или квантовые компьютеры).

Эволюционные вычисления подразумевают использование в инженерных расчетах биологических метафор, известных из биомимикрии и биомеметики, с целью разработки математических и технических средства для решения вычислительных задач, изучения и проектирования инженерных систем. Особенностью эволюционных вычислений является применение биологических методов, «подсмотренных» в природе для поиска оптимальных технических решений [6].

Эволюционные вычисления изначально являются недетерминированными, что подчёркивает их принадлежность к разработкам в области вычислительного интеллекта. Они вполне могут составить основу для построения интеллектуальных информационных систем, позволяющих учитывать когнитивные аспекты и оперировать с большими объемами неточной или неполной информации [7].

В [5] представлена теория эволюционных вычислений. Следствием теории эволюционных вычислений является наличие базового цикла при организации вычислений в базовом цикле.

Алгоритмы организации вычислений в базовом цикле. Эволюционные вычисления являются математическими преобразованиями, позволяющими трансформировать входной поток информации в выходной по правилам, основанным на имитации механизмов эволюционного синтеза, статистическом подходе к исследованию ситуаций и итерационном приближении к искомому решению. Базовый цикл эволюционных вычислений включает следующую последовательность шагов: вычисление целевой функции, оценку качества решений, селективный отбор решений для репродукции и репродукцию, т.е. создание новых решений.

Алгоритм на псевдокоде организации вычислений в базовом цикле имеет следующий вид:

```

Input: Функция для оценки качества решений  $cmp_F$ 
Input:  $ps$  – размер популяции решений
Data:  $t$  – текущий номер поколения
Data:  $P(t=0)$  – исходная популяция решений
Data: параметры алгоритма, включая целевую функцию
Output:  $X^*$  – найденное оптимизированное решение
begin
     $t := 0$ 
     $Pop := init\ Pop(ps)$  /*функция  $init$  выполняет первоначальную случайную инициализацию популяции*/
    while (критерий останова) do
         $v := F(Pop, cmp_F)$ 
         $P(t) := selection(Pop, v)$ 
         $t := t + 1$ 
         $Pop := P(t)$  /*репродукция потомков из отобранных родительских решений с использованием композиции операторов эволюции  $A: P(t+1) = A(P(t))$ */
    return /*восстановление фенотипа  $Pop$ */
end.

```

Базовый цикл эволюционных вычислений не исключает существование различных стратегий организации популяции в эволюционном алгоритме. Например, стратегия вымирания предполагает, что популяция на следующем шаге эволюции состоит только из потомков предыдущей популяции (в области генетических алгоритмов стратегия вымирания соответствует, так называемым, поколенческим алгоритмам, в которых очередная популяция не содержит родительских хромосом, а включает лишь хромосомы сгенерированных потомков).

Другой стратегией является способ, когда популяция на следующем шаге эволюции $P(t+1)$ формируется как комбинация текущей популяции $P(t)$ и образованных на ее основе потомков. В этом случае продолжительность жизни отдельных особей в популяции может превышать одно поколение, следовательно, родители и их потомки конкурируют друг с другом за выживание. Например, в такой разновидности эволюционных алгоритмов как эволюционные стратегии [8, 9] для описания перехода от одного поколения к другому используются следующие обозначения: λ – число потомков, μ – число родителей. Тогда запись вида (μ, λ) при $\lambda \geq \mu$ в поколенческом алгоритме означает, что из созданных λ потомков от μ родителей $(\lambda - \mu)$ худших потомков будут исключены из следующего поколения. Стратегия $(\lambda + \mu)$ при $\lambda > \mu$ будет обозначать, что в следующее поколение будут отобраны по определенной стратегии μ особей из родителей и их потомков.

Другой стратегией организации популяции в эволюционном алгоритме является элитизм: в следующем поколении обеспечивается сохранение, по крайней мере, одного лучшего решения из текущего поколения. Следствием этой стратегии является то, что если найден глобальный оптимум, то эволюционный алгоритм гарантирует сходимость процесса эволюционных вычислений, хотя возрастает риск попадания в локальный оптимум. Алгоритм организации элитизма в процессе эволюционных вычислений, представленный на псевдокоде, имеет вид:

```

Input:  $ps$  – размер популяции решений
Input:  $as$  – размер архива элитных решений
Data:  $t$  – текущий номер поколения
Data:  $P(t=0)$  – исходная популяция решений
Data:  $Arc$  – архив элитных решений
Data: параметры алгоритма, включая целевую функцию
Output:  $X^*$  – найденное оптимизированное решение
begin
     $t := 0$ 
     $Arc := \emptyset$ 
     $Pop := \text{init } Pop(ps)$ 
    while (критерий останова) do
         $Arc := \text{updateOptimalSetN}(Arc, Pop)$  /*обновленное оптимальное множество*/
         $Arc := \text{pruneOptimalSet}(Arc, as)$  (сокращение оптимального множества решений до размера  $ps$ )
         $v := \text{функция качества}(Pop, Arc, \text{cmp}_F)$ 
         $P(t) := \text{selection}(Pop, Arc, v, ps)$ 
         $t := t + 1$ 
         $Pop := \text{репродукция } P(t)$ 
    return /*восстановление фенотипа оптимального множества решений  $(Pop \cup Arc)$ */
end.

```

Уточним различия в представленных выше алгоритмах, реализующих базовый цикл эволюционных вычислений и стратегию элитизма.

Во-первых, создается архив Arc элитных решений, который первоначально является пустым множеством, а затем обновляется функцией «*updateOptimalSetN*», которая сохраняет и обновляет полученные элитные решения. Во-вторых, если множество элитных решений становится слишком большим, то функция «*pruneOptimalSet*» сокращает его до величины ps . В принципе, в алгоритмах, построенных по принципам, отличным от элитизма, такой архив можно сделать пустым множеством ($Arc := \emptyset$).

Реализация базового цикла эволюционных вычислений в алгоритме табуированного поиска. Идеи, связанные с табуированным поиском (*tabu search*), появились в середине 80-х годов. Алгоритмы табуированного поиска также являются разновидностями эволюционных алгоритмов. Они имеют определённое сходство с известным методом градиентного спуска с памятью [10].

В процессе поиска ведётся список табуированных (запрещённых для уничтожения) решений из числа уже оценённых. Критические параметры алгоритмов табуированного поиска – глубина списка запретов, определение текущего состояния, размер области вокруг текущего состояния. В ходе поиска осуществляются операции аспирации (включение в запрещённый список решений в окрестности табуированного) и диверсификации, которая добавляет фактор случайности в процесс поиска [5].

Ниже, исходя из базового цикла ЭВ, на примере задачи минимизации функции приводится описание на псевдокоде предлагаемого алгоритма табуированного поиска $x^* \leftarrow \text{tabuSearch}(f, n)$.

```

Input:  $f$  – целевая минимизируемая функция
Input:  $n$  – максимальная длина списка запретов ( $n > 0$ )
Data:  $p_{new}$  – новое решение
Data:  $p^*$  – лучшее (текущее) решение
Data:  $tabu$  – список запретов
Output:  $X^*$  – лучшее из найденных решений
begin
 $p^* := \text{init Pop}(n)$ 
 $tabu := \text{createList}(1, p^*)$ 
while (критерий останова) do
     $p_{new} := \text{мутация}(p^*)$ 
    if поиск( $p_{new}, tabu$ ) < 0 then
        if  $f(p_{new}) < f(p^*)$  then  $p^* := p_{new}$ 
        if длина( $tabu$ )  $\geq n$  then  $tabu := \text{deleteListItem}(tabu, 0)$ 
         $tabu := \text{addListItem}(tabu, p_{new})$ 
return  $p^*$ 
end.

```

Представленный выше алгоритм табуированного поиска применим, в основном, для решения однокритериальных оптимизационных задач. Алгоритм обобщается на случай решения задачи многокритериальной оптимизации

$x^* \leftarrow \text{tabuSearchMO}(cmp_F, n, a):$

```

Input:  $cmp_F$  – функция для оценки качества решений, представляющая собой компаратор, сравнивающий качество решений
Input:  $n$  – максимальная длина списка запретов ( $n > 0$ )
Input:  $as$  – максимальный размер архива решений
Data:  $tabu$  – список запретов
Data:  $p_{new}$  – новое решение
Data:  $Arc$  – архив лучших новых решений
Output:  $X^*$  – лучшее из найденных решений
begin
     $Arc := \emptyset$ 
     $p_{new} := \text{create}()$ 
     $tabu := \emptyset$ 
    while критерий останова() do
        if searchItem( $p_{new}, tabu \chi Arc$ ) < 0 then
             $Arc := \text{updateOptimalSet}(Arc, p_{new})$ 
             $Arc := \text{pruneOptimalSet}(Arc, as)$ 
             $v := \text{функция качества}(Arc, cmp_F)$ 
            if длина( $tabu$ )  $\geq n$  then  $tabu := \text{deleteListItem}(tabu, 0)$ 
             $tabu := \text{addListItem}(tabu, p_{new})$ 
         $p_{new} := \text{select}(Arc, v, 1)$ 
         $p_{new} := \text{mutate}(p_{new})$ 
    return /* восстановление фенотипа множества ( $Arc$ ) */
end.

```

Здесь использованы ранее применявшиеся обозначения функций.

Отметим также, что табуированный поиск вполне применим для решения не только оптимизационных проблем. Эксперименты показывают, что алгоритм демонстрирует целенаправленное, устойчивое к флуктуациям поведение, не являясь, по сути, инструктивным процессом. Он вполне подходит для решения разнообразных задач когнитивной обработки данных, обучения без учителя.

Заключение. Конечно, ход естественной эволюции гораздо сложнее и эффективнее, нежели любой алгоритм эволюционных вычислений. Тем не менее, эволюционные алгоритмы могут находить близкие к оптимальным решения в самых разных областях, когда даже неизвестно как искать оптимальное решение.

Несмотря на кажущуюся хаотичность, процедура эволюционного поиска в базовом цикле эволюционных вычислений является очень устойчивой [11]. Главное, чтобы у нас был хоть какой-то критерий оценки качества получаемых решений, т.е. лучше ли одно решение, чем другое. Этого достаточно, чтобы эволюционный алгоритм заработал. Для реальных практических задач такой критерий обычно имеется.

Оценка решений в базовом цикле эволюционных вычислений является кумулятивной, т.е. накапливается, давая каждому решению некоторую отсрочку, чтобы эволюционировать. Хорошие решения в ходе эволюции обычно с большой вероятностью копируются. Эта особенность любой разновидности эволюционных вычислений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Torshin I.Y.* Bioinformatics in the Post-Genomic Era: The Role of Biophysics. – NY: Nova Biomedical Books, 2006.
2. *Минкин В.И.* Молекулярные компьютеры // Химия и жизнь-XXI век. – 2004. – № 2. – С. 13-17.
3. *Паун Г., Розенберг Г., Саломая А.* ДНК-компьютер. Новая парадигма вычислений. – М.: Мир, 2004.
4. *Курейчик В.М.* Квантовые и генетические алгоритмы - новая технология эволюционного поиска // Вестник Южного научного центра РАН. – 2005. – Т. 1, № 2. – С. 41-50.
5. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Основы теории эволюционных вычислений. – Ростов-на-Дону: Изд-во ЮФУ, 2010. – 224 с.
6. *Курейчик В.М.* Биоинспирированный поиск с использованием сценарного подхода // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 7-13.
7. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Модели параллелизма эволюционных вычислений // Вестник РГУПС. – 2011. – № 3. – С. 93-97.
8. *Rechenberg I.* Evolutionsstrategie – Optimierung technischer systeme nach prinzipien der biologischen evolution. – Stuttgart: Frommann – Holzboog Verlag, 1973. – 195 p.
9. *Rodzin S.I.* Schemes of Evolution Strategies // Proc. of 2002 IEEE Int. Conf. on AI-Systems (ICAIS, sept. 2002) // IEEE Comp. Society: Los Alamos, California. – P. 375-380.
10. *Glover F.* Tabu Search. – Cambridge: MA; MIT Press, 1997.
11. *Родзин С.И., Курейчик В.М., Курейчик В.В.* Эволюционные вычисления. Теория, модели, алгоритмы, параллелизм. – Saarbrücken, Germany: LAP LAMBERT Publishing GmbH, 2011.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

Родзин Сергей Иванович – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: rodzin@mopevm.tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44, ГСП 17А; тел.: 88634371673; кафедра математического обеспечения и применения ЭВМ; профессор.

Rodzin Sergey Ivanovich – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: rodzin@mopevm.tsure.ru; GSP 17A, 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371673; the department of software engineering; professor.

УДК 681.325

В.Б. Лебедев, А.А. Шашелов

**ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ГЛОБАЛЬНОЙ ТРАССИРОВКИ,
ОСНОВАННЫЙ НА КОДИРОВАНИИ АЛЬТЕРНАТИВ ПРОКЛАДКИ
МАРШРУТА В ГРАФЕ***

Предложен новый метод решения задачи глобальной трассировки на основе генетического алгоритма. Основная идея алгоритма заключается в кодировании альтернатив прокладки маршрута в вершинах графа. Отличительной чертой является то, что кодируется не реальное размещение соединения по областям, а процедура его построения на коммутационном поле, а кодом является число в двоичной системе вычислений. Процесс построения маршрута заключается в последовательном перемещении по ребрам графа G от вершины к вершине, в соответствии с заданным кодом. Это значительно упрощает выполнение генетических операторов и сокращает объем необходимой памяти. В работе описываются принципы кодирования и декодирования хромосомы в компактной форме. Трудоемкость кодирования и декодирования хромосомы имеет линейную сложность $O(L)$, где L – длина хромосомы. Приводятся разработанные генетические операторы. Экспериментальные исследования проводились на IBM PC. По сравнению с существующими алгоритмами достигнуто улучшение результатов.

Генетический алгоритм; глобальная трассировка; оптимизация.

V.B. Lebedev, A.A. Shashelov

**GLOBAL ROUTING GENETIC ALGORITHM BASED ON ALTERNATIVES
OF CONSTRUCTING THE ROUTE**

In this paper, we present a new genetic algorithm for solving the global routing problem. The main idea of the algorithm is the coding alternatives of constructing the route in the vertices of the graph. This paper describes the principles for encoding and decoding of the chromosome in a compact form, also describes the genetic operators. Which distinctive line is that real placing of connection on areas is coded not, and procedure of its construction on a switching field, and a code is number in binary system of calculations. Process of construction of a route consists in consecutive moving on edges graph G from vertex to vertex, according to the set code. It considerably simplifies performance of genetic operators and reduces volume of necessary memory. Labour input coding and chromosome decoding has linear complexity $O(L)$, where L - length of a chromosome. Experimental researches were spent on IBM PC. Compared with existing algorithms achieved better results.

Genetic algorithm; global routing; optimization.

Введение. Разработка методов и алгоритмов для решения задачи глобальной трассировки осуществляется на протяжении многих лет и по-прежнему является актуальной. Данная задача является NP-трудной. Для таких задач поиск точного решения за приемлемое время затруднителен. Алгоритмы глобальной трассировки можно разбить на два класса: последовательный и комбинаторный [1].

* Работа выполнена при частичной поддержке РФФИ (проект № 12-01-00100).