

УДК 681.3.001.63

DOI: 10.15827/2311-6749.18.4.7

СИНТЕЗ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ НА ОСНОВЕ БИОИНСПИРИРОВАННЫХ МЕТОДОВ

Б.К. Лебедев, д.т.н., профессор, lebedev.b.k@gmail.com; О.Б. Лебедев, к.т.н., доцент, lebedev.ob@mail.ru; А.А. Нагабедян, студент, andrewnagabedyan@yandex.ru;

А.Э. Фонова, магистрант, fonova.anya@yandex.ru

(Институт компьютерных технологий и информационной безопасности Южного федерального университета, каф. систем автоматизированного проектирования, г. Таганрог, 347928, Россия)

При решении многих задач одной из часто используемых моделей является дерево, в частности бинарное. В работе рассматривается методика представления деревьев с произвольной локальной степенью вершин в виде линейной записи. Предлагаются структура и принципы кодирования гомологичных хромосом, несущих информацию о дереве. Для получения новых решений применяются модифицированные генетические операторы кроссинговера и мутации, близкие к естественным. Временная сложность новых способов кодирования и декодирования хромосом для представления деревьев составляет $O(n^2)$.

Ключевые слова: математические выражения, генетическое программирование, структура, дерево, кодирование решений, хромосома.

В начале 1990-х гг. была разработана область эволюционных вычислений под названием *генетическое программирование* (ГП) [1, 2]. Его основной идеей было использование эволюционных алгоритмов для создания компьютерных программ. Для представления программ использовался язык программирования Лисп, в котором программы могут легко рассматриваться как структуры дерева. Поэтому вместо обычного использования двоичных последовательностей для отображения решения в ГП в качестве хромосом использовались деревья [2–4].

Деревья в ГП составлены из узлов двух типов – узлов функций и узлов терминалов. Узлы функций – внутренние вершины дерева, обычно они соответствуют арифметическим операциям (+, −, *, \, % и т.д.), математическим функциям (синус, косинус, тангенс, логарифм и т.д.), булевым функциям (и, или, не и т.д.), условным операторам (если ... тогда ... иначе), операторам циклов (до тех пор ... пока), любой другой функции из предметной области задачи [2, 5]. Терминалы – листья дерева, соответствующие либо переменной данной области задачи, либо постоянной. Например, выражение $x^2 + y$ может быть представлено деревом (рис. 1).

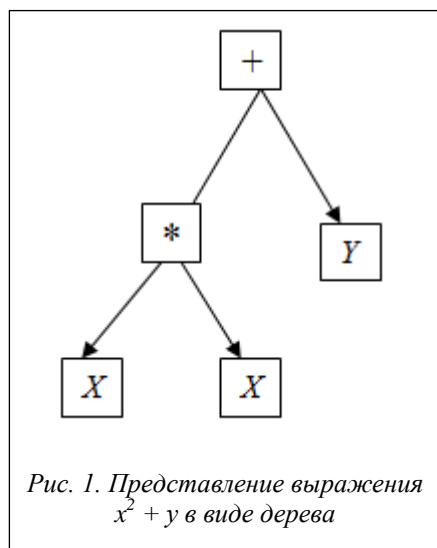


Рис. 1. Представление выражения $x^2 + y$ в виде дерева

Типичной для ГП является задача символического регресса. Символический регресс заключается в построении математического выражения, задаваемого примерами пар $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, где x и y – входные и выходные записи.

Основу генетического алгоритма составляют принципы кодирования и декодирования хромосом, генетические операторы и структура генетического поиска [2, 5–7].

Способы представления исходной формулировки задачи в виде трех компонент в очень большой степени определяют усилия, необходимые для ее решения [1, 2].

Гомологичными называют хромосомы, имеющие общее происхождение и поэтому морфологически и генетически сходные, то есть при применении стандартных генетических операторов не образуются недопустимые хромосомы [8].

В негомологичных хромосомах не может быть двух генов с одинаковым значением. Для негомологичных хромосом применяются различные специальные генетические операторы, которые не создают недопустимых решений. В связи с этим трудоемкость алгоритмов, реализующих генетические операторы для негомологичных хромосом, больше, что увеличивает трудоемкость генетического алгоритма в целом. Это обстоятельство является побудительным мотивом для исследований и разработок гомологичных структур хромосом [9, 10].

Одной из часто используемых моделей при решении многих задач является дерево, в частности бинарное. В работе рассматривается методика представления деревьев с произвольной локальной степенью

вершин в виде линейной записи. Предлагаются структура и принципы кодирования гомологичных хромосом, несущих информацию о дереве [10, 11].

Структуры хромосом для деревьев. Рассмотрим структуру выражения для описания бинарного дерева.

Введем алфавит $A = \{O, \bullet\}$. Структуру бинарного дерева можно задать, используя на базе алфавита A польское выражение, где O соответствует листьям дерева, а \bullet – внутренним вершинам дерева. Каждая внутренняя вершина подвергается бинарному ветвлению. Польские выражения для деревьев, представленных на рисунках 2а и 2б, имеют вид $O O \bullet O O \bullet O \bullet \bullet$ и $O O \bullet O O \bullet \bullet O \bullet$.

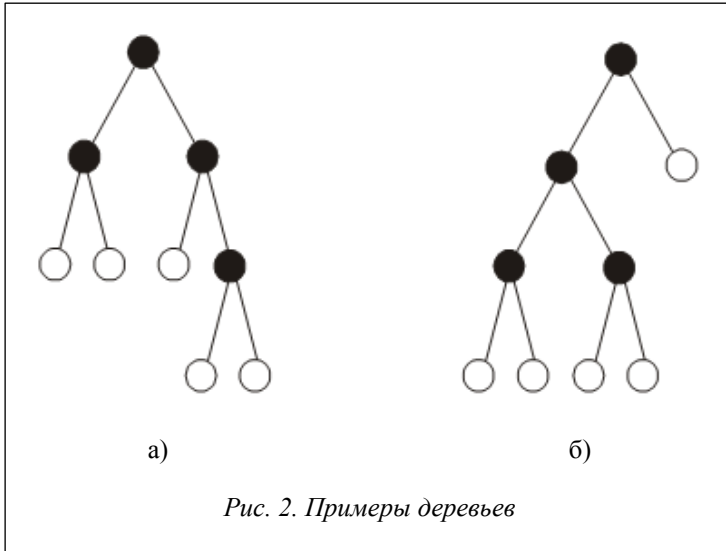


Рис. 2. Примеры деревьев

Процесс восстановления дерева по польскому выражению достаточно прост. Слева направо последовательно просматривается польское выражение и отыскиваются буквы типа \bullet , соответствующие внутренним вершинам. Каждая внутренняя вершина объединяет два ближайших подграфа, сформированных на предыдущих шагах и расположенных в польской записи слева от знака \bullet . С помощью скобок покажем подграфы, образованные при просмотре польского выражения слева направо:

$\{(O O \bullet) [(O O \bullet) O \bullet] \bullet\}$ и $\{[(O O \bullet) (O O \bullet) \bullet] O \bullet\}$.

Отметим основные свойства польского выражения, выполнение которых необходимо, чтобы ему соответствовало бинарное дерево.

Обозначим через n_o число элементов польского выражения типа O , а через n_\bullet – число элементов типа \bullet . Для дерева всегда выполняется равенство $n_o = n_\bullet + 1$.

Если в польском выражении провести справа от знака \bullet сечение, то слева от сечения число знаков O больше числа знаков \bullet , по крайней мере, на единицу: $n_o - n_\bullet \geq 1$. Первый знак \bullet в польском выражении (при просмотре слева направо) может появиться только после двух знаков O .

Пронумеруем позиции между знаками O :

$O O \underline{1} O \underline{2} O \underline{3} O \underline{4} \dots O \underline{n_o - 1}$.

Максимальное число \bullet , которое может появиться в i -й позиции, равно номеру позиции $-i$. Напомним, что общее число $n_\bullet = n_o - 1$.

Если польское выражение обладает вышеперечисленными свойствами, то ему соответствует бинарное дерево.

Рассмотрим структуру и принципы кодирования и декодирования хромосомы для представления польского выражения [3].

Хромосома H имеет вид $H = \{g_i / i = 1, 2, \dots, n_\bullet\}$.

Используем строку с n_o знаками O в качестве опорного множества для построения польского выражения при декодировании хромосомы.

Число генов в хромосоме равно n_\bullet , то есть числу знаков \bullet . Значение $Z(g_i)$ гена g_i колеблется в пределах от i до n_\bullet , то есть $i \leq Z(g_i) \leq n_\bullet$. Значение гена указывает номер позиции между знаками O опорного множества, в которую необходимо поместить знак \bullet . Декодирование хромосом, то есть построение польской записи, осуществляется следующим способом. Формируется базовое множество символов O мощностью $n_o = n_\bullet + 1$ и определяется n_\bullet позиций, расположенных между символами O для помещения в них символов \bullet . Затем последовательно выбираются гены, определяются задаваемые ими номера позиций, в которые и помещаются знаки \bullet .

Например, пусть для $n_\bullet = 4$ имеется хромосома $H = \langle 4, 2, 2, 4 \rangle$. Это значит, что число $n_o = 5$, а число позиций – 4. Два знака \bullet назначаются во вторую позицию, а два знака \bullet – в четвертую. Польское выражение, соответствующее хромосоме, имеет вид

$O O O \bullet \bullet O O \bullet \bullet$.

Дерево, соответствующее данному польскому выражению, представлено на рисунке 3.

Рассмотрим теперь структуру и принципы формирования линейной записи для иерархического дерева без ограничений на локальную степень внутренних вершин. Запись представляет собой набор элементов $A = \{a_i / i = 1, 2, \dots, l\}$, где l – число вершин дерева. Причем n элементов $a_i \in A_o$ соответствуют листь-

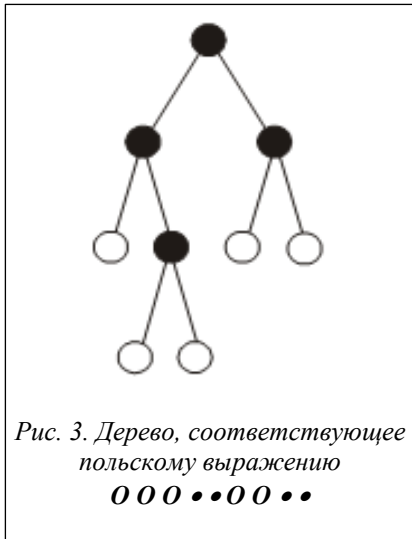


Рис. 3. Дерево, соответствующее польскому выражению

000••00••

ям дерева, а m элементов $a_i \in A_v$ соответствуют внутренним вершинам n -арного дерева разрезом: $n + m = l, A = A_v \cup A_l$.

Формирование дерева в соответствии с древовидной записью осуществляется на основе иерархического подхода при просмотре записи слева направо, начиная с первого элемента.

Вершина x_i в соответствии с записью расположена слева от x_j , если в записи элемент a_i расположен левее a_j . Древовидная запись организована так, что каждая внутренняя вершина x_i , с одной стороны, является корнем некоторого поддеревья для вершин, расположенных слева от x_i , а с другой – может быть дочерней вершиной для некоторой внутренней вершины, расположенной в соответствии с записью справа от x_i . Последнему элементу a_n списка A соответствует вершина x_l , являющаяся корнем всего дерева.

Значением элемента $a_i \in A_v$, соответствующего внутренней вершине x_i , является число поддеревьев, корни которых – дочерние вершины x_i . Описания поддеревьев расположены в линейной записи непосредственно слева от a_i . Если $a_i \in A_o$, то $a_i = 0$, то есть x_i является листом дерева.

Рассмотрим запись **0,0,0,3,0,0,2,0,0,2,0,0,2,3,3**. Выделим с помощью скобок иерархически вложенные друг в друга поддеревья, образованные при просмотре записи слева направо: **((0,0,0,3),(0,0,2),((0,0,2),0,(0,0,2),3),3)**.

Соответствующее дерево представлено на рисунке 4.

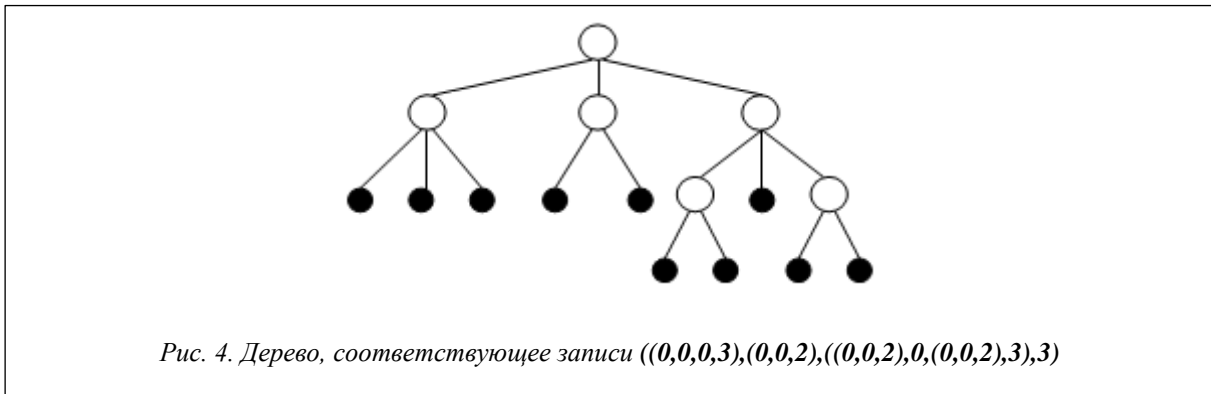


Рис. 4. Дерево, соответствующее записи ((0,0,0,3),(0,0,2),((0,0,2),0,(0,0,2),3),3)

Запись дерева обладает следующими свойствами:

- число элементов a_i записи с нулевым значением равно числу листьев дерева;
- число элементов a_i записи с ненулевым значением колеблется от 1 до $n - 1$;
- значение любого элемента $a_i \in A_v$, соответствующего внутренней вершине x_i , равно или больше 2 ($a_i \geq 2$), так как при разбиении любая внутренняя вершина связана минимум с двумя дочерними вершинами;

- a_1 и a_2 всегда равны нулю;
- между номером позиции k и значениями элементов, расположенных в позициях с 1-й по k -ю, существует зависимость

$$\sum_{i=1}^k a_i < k,$$

то есть сумма значений элементов, расположенных в позициях с 1-й по k -ю, меньше числа позиций k ;

- для записи $A = \{a_i / i = 1, 2, \dots, l\}$ существует зависимость

$$\sum_{i=1}^k a_i = l - 1.$$

Если запись удовлетворяет перечисленным свойствам, то она является древовидной и ей соответствует некоторое дерево.

Структура и принципы кодирования хромосомы для представления древовидной записи разработаны с учетом вышеперечисленных свойств.

Будем использовать множество элементов a_i с нулевым значением в качестве опорного множества для построения древовидной записи при декодировании хромосомы.

Расположим между этими элементами множество позиций:

$$00 \underline{1} 0 \underline{2} 0 \underline{3} \dots \underline{n-2} 0 \underline{n-1}.$$

Если число элементов $a_i \in A_0$ равно n , то число позиций равно $n - 1$. В каждую позицию может быть помещено несколько элементов с ненулевыми значениями, но при этом должны быть соблюдены вышеперечисленные свойства.

Хромосома имеет вид $H = \{g_i / i = 1, 2, \dots, n - 1\}$.

Число генов в хромосоме равно $n - 1$, где n – число листьев дерева. Ген g_i может принимать значение в интервале от i до $n - 1$, кроме того, ген может быть либо помеченным, либо нет. Значение гена указывает номер позиции. Возможны два варианта действий. Если ген не помечен, то в позицию, соответствующую значению гена, к ненулевым элементам последним справа записывается элемент со значением, равным 2. Если же ген помечен, то в соответствующей позиции значение последнего справа ненулевого элемента увеличивается на единицу, а при отсутствии в позиции ненулевых элементов в нее записывается элемент со значением, равным 2. Метки хромосом задаются вектором $M = \{m_i / i = 1, 2, \dots, n - 1\}$. $m_i = 1$, если ген g_i помечен; $m_i = 0$, если ген g_i не помечен. Таким образом, древовидная запись кодируется парой хромосом. Для разметки листьев дерева используется третья хромосома $R = \{r_i / i = 1, 2, \dots, n\}$. Пример: пусть задана пара хромосом

$$H = \{2, 2, 4, 6, 9, 9, 9, 9, 9\};$$

$$M = \{0, 1, 0, 0, 0, 0, 1, 0, 1\}.$$

Поскольку число генов в хромосоме H равно 9, число нулевых элементов в древовидной записи равно 10. После декодирования пары хромосом H и M полученная древовидная запись имеет вид $((0, 0, 0, 3), (0, 0, 2), ((0, 0, 2), 0, (0, 0, 2), 3), 3)$, а соответствующее ей дерево представлено на рисунке 4.

Предложенные структуры хромосом имеют линейную пространственную сложность.

Разметка множества вершин дерева кодируется двумя хромосомами: $H1$ содержит информацию о разметке листьев дерева, $H2$ – о разметке внутренних вершин дерева.

Если допускается повторение меток, то значением гена в $H1$ или $H2$ является метка, которой помечается соответствующая вершина дерева.

Для случая неповторяющихся меток разработаны структура и принципы кодирования хромосом, обладающих свойством гомологичности.

Пусть n – число вершин. Хромосома $H1$ имеет вид $H1 = \langle g_1, g_2, \dots, g_{n-1} \rangle$. В результате декодирования строится вектор R , задающий разметку вершин.

Каждый ген g_i может принимать значение в интервале от 1 до $(n + 1 - i)$. Например, для $n = 8$ $1 \leq g_1 \leq 8$, $1 \leq g_2 \leq 7$, $1 \leq g_3 \leq 6$, ..., $1 \leq g_7 \leq 2$.

Декодирование хромосомы $H1$ производится следующим образом. Пусть для $n = 8$ имеется хромосома $H1 = \langle 3, 5, 3, 4, 4, 2, 2 \rangle$ и пусть имеется опорный вектор $B^1 = \langle a, b, c, d, e, f, g, h \rangle$, число элементов которого равно n . Рассматриваем по порядку гены хромосомы, в соответствии с их значениями выбираем элементы в опорном векторе и записываем их в порядке выборки в вектор R .

Значение $g_1 = 3$. Выбираем в B^1 элемент b^1_j ($j = g_1 = 3$, $b^1_3 = c$) и записываем его на первое место формируемого вектора R , то есть $r_1 = b^1_3 = c$.

Удаляем элемент b^1_3 из B^1 и получаем вектор $B^2 = \langle a, b, d, e, f, g, h \rangle$, содержащий 7 элементов. Следующим выбирается g_2 , $g_2 = 5$. Отыскиваем элемент b^2_5 вектора B^2 : $b^2_5 = f$. Следовательно, $r_2 = f$. Удаляем из B^2 элемент b^2_5 , получаем вектор $B^3 = \langle a, b, d, e, g, h \rangle$. Далее:

$$g_3 = 3, b^3_3 = d, r_3 = d, B^4 = \langle a, b, e, g, h \rangle; \quad g_4 = 4, b^4_4 = g, r_4 = g, B^5 = \langle a, b, e, h \rangle;$$

$$g_5 = 4, b^5_4 = h, r_5 = h, B^6 = \langle a, b, e \rangle; \quad g_6 = 2, b^6_2 = b, r_6 = b, B^7 = \langle a, e \rangle;$$

$$g_7 = 5, b^7_2 = e, r_7 = e, B^8 = \langle a \rangle, r_8 = b^8_1 = a.$$

В итоге получаем вектор $\langle c, f, d, g, h, b, e, a \rangle$, задающий разметку множества вершин.

Генетические операторы. Основными генетическими операторами являются кроссинговер и мутация [1, 2, 8].

У описанной выше структуры хромосом гены, расположенные в одних и тех же локусах, являются гомологичными. Реализация кроссинговера осуществляется следующим образом [4–6].

У предварительно выбранной родительской пары хромосом (на основе использования принципа рулетки) последовательно просматриваются гомологичные пары генов, и с вероятностью P_k осуществляется обмен генами. При таком подходе степень модификации родительских хромосом определяется значением параметра P_k .

Суть оператора мутации в произвольном изменении значений генов. Реализация оператора мутации осуществляется следующим образом. Последовательно просматриваются хромосомы, и с вероятностью P_{M1} они подвергаются мутации. Если хромосома мутирует, то последовательно просматриваются локусы хромосомы и с вероятностью P_{M2} осуществляется мутация гена в рассматриваемом локусе. Мутация заключается в принятии геном случайного значения из заданного диапазона значений для гена в данном локусе [4–6].

Представление решения в виде набора хромосом дает возможность использовать оператор комбинирования набором хромосом в одном решении, что является приближением к естественной эволюции.

С другой стороны, представление решения набором из n хромосом дает возможность организации поиска решений в различных постановках, оставляя отдельные виды хромосом неизменными в процессе генетического поиска [11].

Очевидно, что фиксация отдельных хромосом в некоторой постановке приводит к сужению пространства поиска, и при этом возможна потеря оптимальных решений. В этой связи представляется целесообразным комбинирование отдельными постановками при поиске оптимального решения [10, 11].

В общем случае возможны три подхода к комбинированию постановок: последовательный, параллельный и параллельно-последовательный.

При последовательном подходе на каждом i -м этапе осуществляется генетический поиск путем модификации хромосом, входящих в заданный для этого этапа набор NH_i модифицируемых типов хромосом. Это означает, что в полном объеме используется кроссинговер $K1$, заключающийся в комбинировании наборов хромосом, входящих в решение, а кроссинговер $K2$ и мутация применяются только к тем хромосомам, которые входят в набор типов модифицируемых хромосом.

Приведем комбинацию, при которой в наборы входят по одному типу хромосом: $NH_1 = \{H1\}$, $NH_2 = \{H2\}$, $NH_3 = \{H3\}$. В набор могут входить от одного до четырех типов хромосом. На первом этапе в качестве исходной служит популяция P_u . На втором – популяция P_1 , сформированная после отработки первого этапа, и т.д. Отметим возможность циклического повторения этапов.

При параллельном поиске производится распараллеливание процесса генетического поиска.

Вначале формируется исходная популяция. Для каждой параллельной ветви задается набор NH_i типов хромосом, подвергающихся модификации. Затем на первом шаге на базе этой исходной популяции на каждой параллельной ветви осуществляется генетический поиск путем модификации хромосом, входящих в соответствующий набор типов модификаций хромосом.

После некоторого числа генераций (поколений) осуществляется случайное или направленное перемещение хромосом между любыми популяциями P_1-P_3 , сформированными на данный момент на параллельных ветвях. После этого на втором шаге модифицированные популяции $P_{10}-P_{30}$ вновь подвергаются обработке генетическими процедурами на параллельных ветвях. Число шагов является управляющим параметром [8, 10, 11].

При параллельно-последовательном подходе на каждой параллельной ветви реализуется последовательная комбинация постановок.

Как видно из алгоритмов, реализующих операторы кроссинговера и мутации, оценка их временной сложности на итерации имеет вид $O(n)$.

Заключение. Рассмотренные в работе новые принципы и способы кодирования и декодирования хромосом для представления деревьев исключают некорректные решения, отличаются простотой и линейными оценками временной и пространственной сложности, что упрощает использование генетических операторов и позволяет использовать модификации генетических операторов, близких к естественным.

Многохромосомные представления решений позволили создать иерархические структуры генетических операторов, что дает возможность организовать целенаправленный поиск.

Предложенные процедуры кодирования и декодирования повышают целенаправленность поиска, включают процедуры улучшения решения и обеспечивают более высокую вероятность локализации глобального экстремума задачи. В среднем это позволяет повысить на 2–4 % качество решения.

Работа выполнена при финансовой поддержке гранта РФФИ № 17-07-00997.

Литература

1. Quagliarella D. [et all], eds. Genetic algorithms and evolution strategy in engineering and computer science. John Wiley & Song, 1998.
2. Davis L., Ed. Handbook of Genetic Algorithms. NY, van Nostrand Reinhold, 1991, 385 p.
3. Гладков Л.А., Зинченко Л.А., Курейчик В.В. Методы генетического поиска. Таганрог: Изд-во ТРТУ, 2002. 145 с.
4. Лебедев Б.К. Интеллектуальные процедуры синтеза топологии СБИС. Таганрог: Изд-во ТРТУ, 2003. 108 с.
5. Курейчик В.В. Эволюционные, синергетические и гомеостатические методы принятия решений. Таганрог: Изд-во ТРТУ, 2001. 221 с.
6. Курейчик В.В., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Изв. ТРТУ. Тематич. вып.: Интеллектуальные САПР. 2009. № 4. С. 16–25.
7. Тарасов В.Б., Голубин А.В. Эволюционное проектирование: на границе между проектированием и самоорганизацией // Изв. ТРТУ. Тематич. вып.: Интеллектуальные САПР. 2006. № 8. С. 77–82.

8. Гладков Л.А., Курейчик В.В., Курейчик В.М., Сороколетов П.В. Биoinsпирированные методы в оптимизации. М.: Физматлит, 2009. 384 с.
9. Норенков И.П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Баумана, 2006. 348 с.
10. Лебедев О.Б. Синтез дерева разрезов при планировании на основе метода муравьиной колонии // Интегрированные модели и мягкие вычисления в искусственном интеллекте: матер. VI Междунар. науч.-практич. конф. М.: Физматлит, 2011. № 1. С. 521–529.
11. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. 448 с.