

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/17/10

УДК 681.324, 519.17

ИССЛЕДОВАНИЕ ВЛИЯНИЯ СТЕПЕНИ СПЕЦИАЛИЗАЦИИ ШАБЛОНА НА ПРОСТРАНСТВО ПОИСКА ПРИ ЭВОЛЮЦИОННОМ СИНТЕЗЕ МОДЕЛЕЙ¹

О. Г. Монахов

*Институт вычислительной математики и математической геофизики СО РАН,
г. Новосибирск, Россия*

E-mail: monakhov@rav.sscs.ru

Описан алгоритм эволюционного синтеза, объединяющий преимущества генетических алгоритмов и генетического программирования, основанный на эволюционных вычислениях, шаблонах (темплейтах, скелетонах) алгоритмов и заданном множестве пар входных — выходных данных. Исследовано влияние степени специализации шаблона на пространство поиска при эволюционном синтезе, получены оценки величины сокращения пространства поиска при введении в шаблон дополнительной информации в виде формул, уточняющих модель и имеющих представление в виде бинарного дерева. Теоретически и экспериментально показано для данного случая экспоненциальное и сверхэкспоненциальное сокращение времени работы алгоритма синтеза при увеличении степени специализации шаблона.

Ключевые слова: *генетический алгоритм, генетическое программирование, эволюционный синтез, шаблон, темплейт, скелетон.*

Введение

В работе рассматривается проблема синтеза алгоритма (модели) A как задача поиска параметров и функций данного шаблона T (темплейта) алгоритма с целью оптимизации заданной функции F , характеризующей качество алгоритма A . Шаблон (template, skeleton [1–3], design pattern [4]) представляет собой управляющую параметризованную структуру алгоритма, которая описывает порядок сканирования структур данных алгоритма и определяет динамику вычислительного процесса в пространственно-временных координатах.

Пусть шаблон T алгоритма A содержит параметры $P = \{p_k : k \geq 0\}$, описывающие значения целочисленных и действительных коэффициентов и переменных, значения индексов, параметры структур данных, константы и некоторые примитивные операции алгоритма (величины инкрементов и декрементов, знаки переменных, логические операции и отношения, типы округления переменных). Пусть шаблон T содержит также множество функций (формул) $FM = \{f_n : n \geq 0\}$ алгоритма A . При задании значений параметров P и определении функций FM в данном шаблоне T получаем (синтезируем) алгоритм $A(T, P, FM)$.

¹Работа поддержана грантом РФФИ № 08-01-00857.

Если заданы наборы значений входных параметров алгоритма X_i , $1 \leq i \leq N$, и выходных параметров Y_i , которые необходимо получить в результате исполнения алгоритма, то качество работы алгоритма характеризует целевая функция F , оценивающая величину рассогласования между требуемыми Y_i и полученными выходными значениями алгоритма при данных входных значениях $Y_i' = A(T, P, FM, X_i)$. Кроме того, функция F может оценивать не только точность, но и сложность получаемого алгоритма (например, время выполнения, число итераций, сложность (длину) формул [5]).

Таким образом, проблема синтеза алгоритма состоит в следующем: для данного шаблона T и заданных наборов значений входных — выходных параметров $\{X_i, Y_i : 1 \leq i \leq N\}$ необходимо найти такие значения параметров P^* и функций FM^* шаблона T , определяющие алгоритм $A^*(T, P^*, FM^*)$, что

$$F(A^*(T, P^*, FM^*, X_i)) \leq F(A(T, P, FM, X_i))$$

для всех $1 \leq i \leq N$ при любых других значениях параметров $P \in \text{Dom}(P)$ и функций $FM \in \text{Dom}(FM)$.

Для решения проблемы автоматизированного синтеза (открытия) алгоритмов используется эволюционный алгоритм, основанный на шаблонах [5–8], оптимизирующий заданный критерий качества F . Этот гибридный алгоритм синтеза использует структуры данных и операторы генетических алгоритмов (ГА) [9] и генетического программирования (ГП) [10] и позволяет получить новые свойства (меньшее время исполнения и синтез алгоритмов с большей циклической сложностью и наличием рекурсивных функций, по сравнению с ГП, и синтез новых функций и предикатов, что достаточно трудно при традиционном ГА).

Эти свойства подхода с использованием шаблонов основаны на предварительном знании прикладной области и выборе обобщённой вычислительной схемы алгоритма, задаваемой в виде шаблона. Традиционное генетическое программирование страдает от слабо ограниченного слепого поиска в огромных пространствах параметров для реальных прикладных проблем и имеет большие временные затраты. В данной работе исследуется использование шаблонов алгоритмов для ограничения и сокращения пространства поиска к допустимым моделям и структурам решений. Этот подход представляет гибкий и прямой путь включения знаний эксперта относительно параметров, функций и структуры решаемой проблемы в разрабатываемую вычислительную модель алгоритма. В предельных случаях, с одной стороны, если мы не знаем шаблон искомого алгоритма, данный подход даёт традиционное генетическое программирование. С другой стороны, если мы знаем полную структуру алгоритма и не знаем только некоторые его параметры, данный подход даёт традиционный генетический алгоритм для поиска оптимальных параметров. Примером такой параметрической оптимизации на основе заданного шаблона с помощью генетического алгоритма служит алгоритм, предназначенный для открытия аналитических описаний новых плотных семейств оптимальных регулярных сетей [6, 7].

В п. 1 представлено краткое описание алгоритма эволюционного синтеза, в п. 2 получены оценки влияния степени специализации шаблона на пространство поиска при эволюционном синтезе.

1. Алгоритм эволюционного синтеза

Алгоритм эволюционного синтеза основан на эволюционных вычислениях и моделировании процесса естественного отбора популяции особей, каждая из которых

представлена точкой в пространстве решений задачи оптимизации. Особи представлены структурами данных Gen — хромосомами, включающими недоопределённые параметры p_k и функции (формулы) f_n шаблона T : $Gen = \{P, FM\} = \{p_1, p_2, \dots, p_k; f_1, f_2, \dots, f_n\}$, $k, n \geq 0$. Эти параметры и функции определяют необходимый алгоритм $A(T, Gen)$ на основе заданного шаблона T и хромосомы Gen . Каждая популяция является множеством структур данных Gen и определяет множество алгоритмов $A(T, Gen)$, образующихся из данного шаблона T .

Для представления и реализации хромосомы предложен подход, основанный на комбинации линейной структуры хромосомы для параметров p_k (как в ГА [9]) и древовидной структуры хромосомы для каждой из функций (формул) f_n (как в ГП [10]). Линейная структура хромосомы Gen используется для представления таких параметров p_k шаблона T , как значения целочисленных и действительных коэффициентов и переменных, значения индексов, величины инкрементов и декрементов, а также знаков переменных, логических операций и отношений, типов округления переменных.

Древовидные структуры в хромосоме Gen используются для представления каждой из функций (формул) f_n . Переменные и константы формулы f_n образуют листья (терминальные вершины) дерева — TS , а операции и примитивные функции, используемые в формуле f_n , соответствуют остальным (нетерминальным) вершинам дерева — NS . Каждая операция (примитивная функция) и её операнды (аргументы примитивной функции) представлены соответственно вершиной и её сыновьями в дереве.

Пусть дана формула $f_n = (5x + a)(x - 2)^2$, тогда имеем следующие множества терминальных $TS = \{x, a, 2, 5\}$ и нетерминальных $NS = \{+, -, \cdot, ^2\}$ вершин. Пример соответствующего дерева представлен на рис. 1.

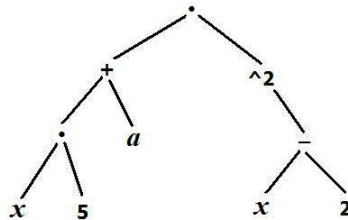


Рис. 1. Представление формулы в виде дерева

Для заданного шаблона T при создании хромосом Gen порождаются аналитические выражения для функций f_n и задаются значения параметров p_k , по которым можно производить оценивание и модификации алгоритма $A(T, Gen)$.

Примем, что целевая функция (fitness function, функция качества, функция пригодности) F вычисляет сумму квадратов отклонений выходных данных алгоритма $Y'_i = A(T, Gen, X_i)$ от заданных эталонных значений Y_i для определённых наборов множества входных данных X_i , $1 \leq i \leq N$:

$$F = \sum_{i=1}^N (A(T, Gen, X_i) - Y_i)^2.$$

Целью алгоритма является поиск минимума F . На практике если получается несколько решений с одинаковым значением целевой функции, то выбирается решение с меньшей оценкой структурной сложности, т. е. с меньшей общей длиной формул решения.

Таким образом, для фиксированных значений функций f_n и параметров p_k можно вычислять выходные значения Y'_i алгоритмов $A(T, Gen, X_i)$, порождённых на основе

заданных шаблонов T и полученных в ходе эволюции хромосом Gen , для требуемых наборов множества входных данных X_i , $1 \leq i \leq N$. После выполнения алгоритмов A получаем значения целевой функции F и выбираем лучшие алгоритмы в популяции.

Основная идея алгоритма синтеза состоит в эволюционном преобразовании множества хромосом (параметров и формул шаблона) с помощью генетических операций селекции (отбора), мутации, кроссовера и добавления новых элементов в процессе естественного отбора с целью выживания «сильнейшего», т. е. имеющего наименьшее значение целевой функции, алгоритма.

Оператор *мутации* применяется к особям, случайно выбранным из текущей популяции с вероятностью $p_m \in [0, 1]$. Мутация линейной части хромосомы Gen состоит в изменении значения случайно выбранного параметра p_k на другую случайно выбранную величину из множества допустимых значений. Мутация древовидной части хромосомы Gen состоит в изменении значения случайно выбранной вершины в представлении функции f_n на другую случайно выбранную величину из множества допустимых значений.

Оператор *кроссовера* (скрещивания) применяется к двум особям (родителям), случайно выбранным из текущей популяции с вероятностью $p_c \in [0, 1]$. Кроссовер состоит в порождении двух новых особей путём обмена частями хромосом родителей (подчастями линейных структур и поддеревьями для древовидных структур хромосомы).

Оператор создания *нового элемента* (особи) состоит в генерации случайных значений параметров p_k и функций f_n , описывающих алгоритмы. Это позволяет увеличить степень разнообразия особей при создании популяции.

Оператор *селекции* (отбора) реализует принцип выживания наиболее приспособленных особей. Он выбирает наилучших особей с минимальными значениями целевой функции.

Для поиска оптимума заданной целевой функции F итерационный процесс вычислений в алгоритме эволюционного синтеза организован следующим образом.

П е р в а я и т е р а ц и я (инициализация): порождение начальной популяции. Все особи популяции создаются с помощью оператора *новый элемент* с проверкой и отсеиванием всех непригодных особей. После заполнения массива популяции лучшие особи отбираются и запоминаются в массиве *best*.

П р о м е ж у т о ч н а я и т е р а ц и я (генерация новой популяции): шаг от текущей к следующей популяции. Основной шаг алгоритма состоит в создании нового поколения особей на основе массива *best* и текущей популяции с использованием операций селекции, мутации, кроссовера и добавления новых элементов.

После оценки целевой функции для каждой особи в поколении проводится сравнение значений этих функций со значениями целевых функций тех особей, которые сохранены в массиве *best*. В том случае, если элемент из нового поколения лучше, чем элемент $best[i]$, для некоторого i , помещаем новый элемент на место i в массив *best* и сдвигаем в нём все остальные элементы на единицу вниз. Таким образом, лучшие элементы локализуются в массиве *best* на местах с меньшими значениями индекса.

П о с л е д н я я и т е р а ц и я (критерий останова): итерации заканчиваются либо после исполнения заданного числа шагов, либо после нахождения оптимального алгоритма $A(T, Gen)$ (с заданным значением целевой функции F). После выполнения данного количества шагов алгоритма синтеза получаем множество (популяцию) алгоритмов $A(T, Gen)$, содержащее в элементе $best[0]$ алгоритм $A^*(T, Gen)$, имеющий минимальное значение целевой функции F .

2. Оценки влияния степени специализации шаблона на пространство поиска

Алгоритм эволюционного синтеза на основе шаблонов был успешно применён для автоматизированного получения следующих алгоритмов: вычисления степени и факториала натурального числа, определения минимального (максимального) элемента массива, определения суммы квадратов элементов массива, вычисления скалярного произведения векторов, определения формул последовательностей Фибоначчи и Трибоначчи, вычисления суммы двух матриц, сортировки методами пузырька, слияния и Шелла, вычисления корней уравнений, определения минимального покрывающего дерева и кратчайших путей в графе, параллельных алгоритмов балансировки нагрузки в многопроцессорных системах. Данный подход был также применён для открытия аналитических описаний новых плотных семейств оптимальных регулярных сетей [6, 7, 11] и синтеза нового алгоритма определения функции расстояния для циркулянтных графов степени 4.

Эволюционный синтез алгоритмов на основе шаблонов реализован в системе TES (Template-based Evolutionary Synthesis) [8] на языке программирования C, шаблоны также задаются на этом языке.

При реализации алгоритма эволюционного синтеза на основе шаблонов основную трудность представляет поиск неопределённых функций с помощью генетического программирования. Пространство поиска в этом случае представлено множеством деревьев, в котором переменные и константы образуют листья дерева, а операции и примитивные функции — внутренние (нетерминальные) вершины дерева. Обозначим через S число терминальных символов, M — число примитивных функций.

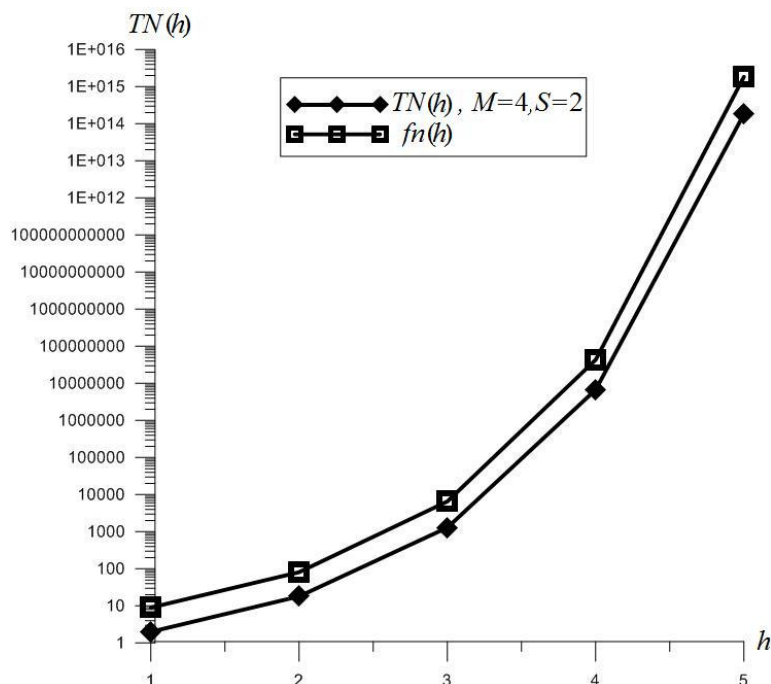
Для оценки влияния степени специализации шаблона на сокращение пространства поиска при эволюционном синтезе с введением в шаблон дополнительной информации в виде формул, уточняющих модель, рассмотрим два простых, но достаточно общих случая: 1) когда в шаблоне известно полное поддерево высотой h и 2) когда в шаблоне известно поддерево с t вершинами. Примем, что множество примитивных функций составляют функции с двумя переменными (операндами), в этом случае достаточно рассматривать бинарные деревья. При поиске полного бинарного дерева высотой h в эволюционном алгоритме перебираются деревья высотой не более h . Обозначим $TN(h)$ число различных бинарных деревьев высотой не более h . Деревья, состоящие из одной терминальной вершины (корня), имеют высоту 0, число таких деревьев $TN(0) = S$. Число различных деревьев высотой не более $h > 0$ получается из комбинации M корневых вершин (по числу примитивных функций), двух различных поддеревьев высотой не более $h - 1$ и дополнительно S деревьев высотой 0. Отсюда число деревьев высотой не более $h > 0$ равно $TN(h) = M \cdot TN^2(h - 1) + S$. При $M = 1$ и $S = 1$ получаем известную формулу для числа различных бинарных деревьев высотой не более h : $TN(h) = TN^2(h - 1) + 1$ (см. [12, 13], последовательность A003095 (M1544)).

Пример числа деревьев высотой не более h для двух различных наборов значений M и S приведён в табл. 1. Как видно из таблицы, с ростом высоты деревьев h пространство поиска увеличивается значительно (сверхэкспоненциально, с порядком роста $O(C^{2^h})$, где C — константа, зависящая от M и S , см. рис. 2), поэтому уменьшение путём использования более специализированного шаблона даже на 1 высоты деревьев, представляющих неизвестные функции, приводит к сверхэкспоненциальному сокращению пространства поиска при эволюционном синтезе от нескольких раз до нескольких порядков.

Т а б л и ц а 1

 $TN(h)$ — число деревьев высотой не более h

h	0	1	2	3	4	5
$M = 1, S = 1$	1	2	5	26	677	458330
$M = 4, S = 2$	2	18	1298	$6,74 \cdot 10^6$	$1,82 \cdot 10^{14}$	$1,32 \cdot 10^{29}$

Рис. 2. Графики функций $TN(h)$ (при $M = 4, S = 2$) и $fn(h) = 3^{2h}$

Для примера рассмотрим влияние степени специализации шаблона с различной высотой поддеревьев при поиске полиномиальной функции $F(x) = x^8 + x^4 + x^2 - (x + 2)$, представленной на рис. 3.

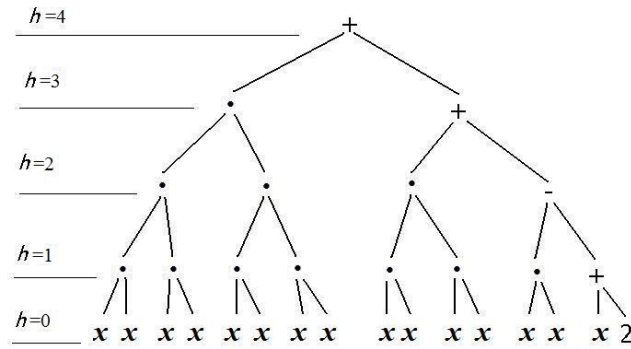
Число итераций и размер популяции выбраны экспериментально на основе параметров из [9, 10]. Значения основных параметров в экспериментах приведены в табл. 2.

Т а б л и ц а 2

Значения параметров алгоритма

Параметр	Значение
Размер популяции, особей	100
Число итераций, шт.	30000
Вероятность кроссовера, %	70
Вероятность мутации, %	15

Пусть заданы 8 пар вход—выход для $x = 0, 1, \dots, 7$ и шаблон $T(x) = T(t_i, x^4 + x^2 - (x + 2))$, где известный терм $x^4 + x^2 - (x + 2)$ и известное поддерево шаблона $t_i \in \{t_1 = x^2, t_2 = x^4, t_3 = x^8\}$ будут последовательно снижать высоту $h = 3, 2, 1$ неизвестного поддерева функции $F(x)$. Множество терминальных символов в этом случае $TS = \{x, Cr\}$, $S = 2$, где Cr —элемент множества случайных натуральных чисел. Набор операций, используемый для синтеза формул, в шаблонах следующий:

Рис. 3. Представление функции $F(x) = x^8 + x^4 + x^2 - (x + 2)$ в виде дерева

$NS = \{+, -, \cdot, /\}$, $M = 4$. В табл. 3 приведены результаты сравнения характеристик алгоритма эволюционного синтеза — времени поиска решения и числа поколений G — до получения решения (функции $F(x)$) при различных шаблонах (для каждого шаблона приведены средние значения, полученные по результатам 100 экспериментов).

Т а б л и ц а 3
Характеристики алгоритма

Шаблон t_i	h	Время, с	G
$t_1 = x^2$	3	43,03	19010,39
$t_2 = x^4$	2	0,076	38,53
$t_3 = x^8$	1	0,0036	1,5

Из табл. 3 видно, что с уменьшением высоты h неизвестных поддеревьев шаблона пространство поиска значительно сокращается и характеристики алгоритма эволюционного синтеза уменьшаются от нескольких раз до нескольких порядков (сверхэкспоненциально, с оценкой порядка $O(C^{2^h})$; так, $G(h) \approx 3,42^{2^h}$ при $M = 4$, $S = 2$).

Рассмотрим влияние степени специализации шаблона на сокращение пространства поиска при эволюционном синтезе во втором случае — когда в шаблоне известно поддерево с t внутренними вершинами. Обозначим $TN(t)$ число различных бинарных деревьев с t внутренними вершинами. Число таких деревьев, состоящих из одной терминальной вершины (корня), $TN(0) = S$. Число различных деревьев с $t > 0$ внутренними вершинами получается из комбинации M корневых вершин (по числу примитивных функций) и всех комбинаций двух различных поддеревьев, имеющих в сумме $t - 1$ внутренних вершин. Отсюда число различных бинарных деревьев с t внутренними вершинами $TN(t) = \sum_{j=0}^{t-1} M \cdot TN(t-1-j)TN(j)$. При $M = 1$ и $S = 1$ получаем извест-

ную формулу для чисел Каталана: $C(t) = \sum_{j=0}^{t-1} C(t-1-j)C(j)$, $C(0) = 1$, $t > 0$. Числа Каталана имеют множество комбинаторных интерпретаций (см. [12, 13], последовательность A000108 (M1459)), одна из которых — число различных бинарных деревьев с t внутренними вершинами [14].

Пример числа деревьев с t внутренними вершинами для двух различных наборов значений M и S приведён в табл. 4. Как видно из таблицы, с ростом числа внутренних

вершин деревьев пространство поиска увеличивается экспоненциально (с порядком роста $O(C^t)$, где C — константа, зависящая от M и S , см. рис. 4 для функций $TN(t)$ при $M = 4, S = 2, f_1(t) = 0,7e^{1,7t}$ и $f_2(t) = 6^t$), поэтому уменьшение путём использования более специализированного шаблона числа вершин деревьев, представляющих неизвестные функции, приводит к экспоненциальному сокращению пространства поиска при эволюционном синтезе.

Т а б л и ц а 4

$TN(t)$ — число деревьев с t внутренними вершинами

t	0	1	2	3	4	5	6	7	8	9	10
$M = 1, S = 1$	1	1	2	5	14	42	132	429	1430	4862	16796
$M = 4, S = 2$	2	4	16	80	448	2688	16896	109824	732160	4978688	34398208

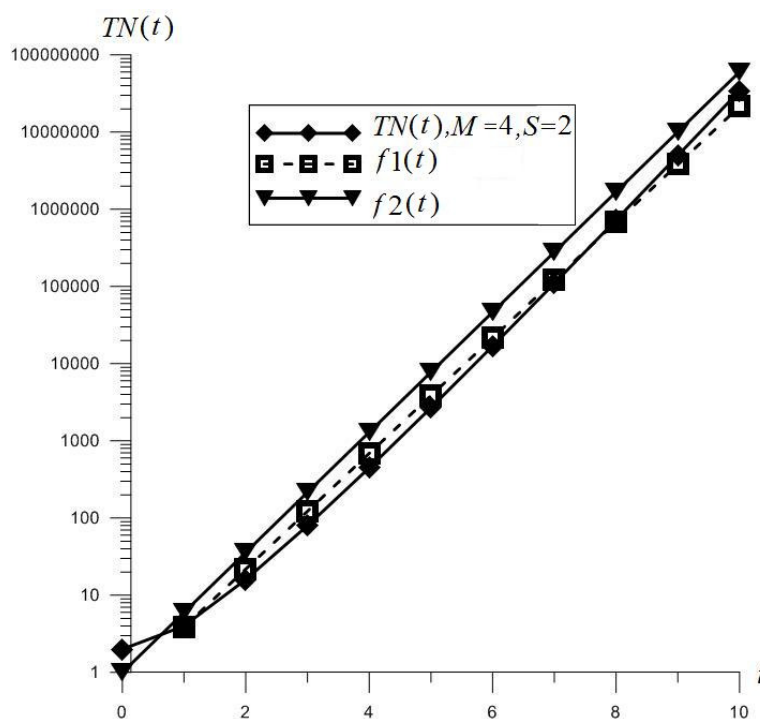


Рис. 4. Графики функций $TN(t)$ при $M = 4, S = 2, f_1(t) = 0,7e^{1,7t}$ и $f_2(t) = 6^t$

Для примера рассмотрим влияние степени специализации шаблона с различным числом вершин поддеревьев при поиске функции $F(x) = 2(x + 2(x + 2(x + 2(x + 2(x + 2))))))$, $t = 10$, представленной на рис. 5.

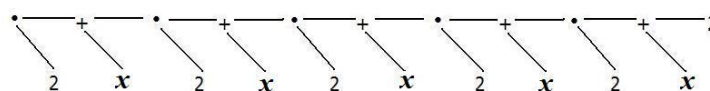


Рис. 5. Представление функции $F(x) = 2(x + 2(x + 2(x + 2(x + 2(x + 2))))))$ в виде дерева

Пусть заданы 8 пар вход — выход для $x = 0, 1, \dots, 7$ и шаблон $T(x) = T(T_i, F_i)$, $i = 1, \dots, 5$, где F_i — известная часть функции, T_i — поддерево, которое представляет

неизвестную часть функции и последовательно имеет $t = 1, 2, 3, 4, 5$ внутренних вершин (см. табл. 5). Множество терминальных символов в этом случае $TS = \{x, Cr\}$, $S = 2$, где Cr — элемент множества случайных натуральных чисел. Набор операций, используемый для синтеза формул, следующий: $NS = \{+, -, \cdot, /\}$, $M = 4$. В табл. 5 приведены результаты сравнения характеристик алгоритма эволюционного синтеза — времени поиска решения и числа поколений G до получения решения — при различных шаблонах (для каждого шаблона приведены средние значения, полученные по результатам 100 экспериментов).

Таблица 5

Сравнение характеристик алгоритма эволюционного синтеза

i	T_i, F_i	t	Время, с	G
1	$T_1 = 2F_1, F_1 = (x + 2(x + 2(x + 2(x + 2))))$	1	0,0045	1,41
2	$T_2 = 2(x + F_2), F_2 = 2(x + 2(x + 2(x + 2(x + 2))))$	2	0,059	25,9
3	$T_3 = 2(x + 2F_3), F_3 = (x + 2(x + 2(x + 2(x + 2))))$	3	0,45	231,15
4	$T_4 = 2(x + 2(x + F_4)), F_4 = 2(x + 2(x + 2(x + 2(x + 2))))$	4	18,5	9321,4
5	$T_5 = 2(x + 2(x + 2F_5)), F_5 = (x + 2(x + 2(x + 2(x + 2))))$	5	48,4	24532,2

Из табл. 5 видно, что с уменьшением числа внутренних вершин t неизвестных поддеревьев шаблона пространство поиска сокращается экспоненциально и характеристики алгоритма эволюционного синтеза уменьшаются также экспоненциально (с оценкой порядка $O(C^t)$; так, $G(t) \approx 0,01e^{2,48t}$ при $M = 4, S = 2$).

Рассмотрим влияние степени специализации шаблона с различным числом вершин поддеревьев при поиске функции $F(x) = e^x x^3 \sin(x) \cos(x)$, $t = 8$, включающей функции от одной переменной и имеющей представление в виде дерева с внутренними вершинами со степенями 2 и 3.

Пусть заданы 15 пар вход — выход для $x = 0, 1, \dots, 14$ и шаблон $T(x) = f(T_i)$, $i = 1, \dots, 5$, где T_i — поддерево, которое представляет неизвестную часть функции, а известная часть последовательно имеет $t = 1, 2, 3, 4, 5$ внутренних вершин (см. табл. 6). Множество терминальных символов в этом случае $TS = \{x, Cr\}$, $S = 2$, где Cr — элемент множества случайных натуральных чисел. Набор операций, используемый для синтеза формул, следующий: $NS = \{+, -, \cdot, /, \exp, \sin, \cos\}$, $M = 7$. В табл. 6 приведены результаты сравнения характеристик алгоритма эволюционного синтеза — времени поиска решения и числа поколений G до получения решения — при различных шаблонах (для каждого шаблона приведены средние значения, полученные по результатам 100 экспериментов).

Таблица 6

Сравнение характеристик алгоритма

i	T, T_i	t	Время, с	G
1	$T = T_1' \cdot T_1'', T_1' = e^x \cdot \sin(x), T_1'' = x^3 \cdot \cos(x)$	1	1977,8	641,7
2	$T = T_2 \cdot \cos(x), T_2 = e^x \cdot x^3 \cdot \sin(x)$	2	696,6	296,7
3	$T = T_3 \cdot \cos(x) \cdot x, T_3 = e^x \cdot x^2 \cdot \sin(x)$	3	150,5	31,9
4	$T = T_4 \cdot \cos(x) \cdot x \cdot x, T_4 = e^x \cdot x \cdot \sin(x)$	4	2,55	11,7
5	$T = T_5 \cdot \cos(x) \cdot x \cdot \sin(x), F_5 = e^x \cdot x^2$	5	0,6	2,5

Из табл. 6 видно, что с увеличением числа внутренних вершин t известных поддеревьев шаблона пространство поиска и характеристики алгоритма значительно умень-

шаются (с оценкой порядка $O(C^{-t})$; так, $G(t) \approx 0,0044e^{-1,75t}$ при $M = 7$, $S = 2$), хотя величина показателя меньше, чем в случае бинарного дерева.

Рассмотрим влияние шаблона при поиске описания последовательности Маккея — Гласса (Mackey — Glass), имеющей в дискретной форме следующий вид: $x(n) = 0,9x(n-1) + 0,2x(n-17)/(1+x(n-17)^{10})$. Пусть заданы 100 пар вход — выход для $n = 0, 1, \dots, 99$, множество терминальных символов $TS = \{x, Cr\}$, $S = 2$, множество примитивных функций $NS = \{+, -, \cdot, /, \text{row}(5, x)\}$, $M = 5$. При поиске описания в форме $x(n) = f(x(n-c), x(n-d))$, где параметры $c \in [0, 21]$, $d \in [0, 21]$ — целые числа, при числе поколений $G = 5000$ и времени поиска решения 4961 с получаем приближённое решение с величиной ошибки $\varepsilon = 0,0058$. При использовании шаблона, выражающего зависимость от предыдущего и любого другого члена последовательности в виде $x(n) = a \cdot x(n-1) + b \cdot f(x(n-d))$, где $a \in [0, 2]$, $b \in [0, 2]$ — действительные, $d \in [0, 21]$ — целые числа, получаем точную формулу последовательности при числе поколений $G = 801$ и времени поиска решения 905 с.

Заключение

Дано описание алгоритма эволюционного синтеза, объединяющего преимущества генетических алгоритмов и генетического программирования, основанного на эволюционных вычислениях, шаблонах (темплейтах, скелетонах) алгоритмов и заданном множестве пар входных — выходных данных. Проведено исследование влияния степени специализации шаблона на пространство поиска при эволюционном синтезе, получены оценки величины сокращения пространства поиска при введении в шаблон дополнительной информации в виде формул, уточняющих модель и имеющих представление в виде бинарного дерева. Для данного случая теоретически и экспериментально показано экспоненциальное и сверхэкспоненциальное сокращение времени работы алгоритма синтеза при увеличении степени специализации шаблона. В планах на будущее стоит исследование влияния на результаты синтеза использования некорректного шаблона и способов его автоматической коррекции алгоритмом эволюционного синтеза.

ЛИТЕРАТУРА

1. Cole M. Algorithmic Skeletons: Structured Management of Parallel Computation. Cambridge: The MIT Press, 1989. 131 p.
2. Mirenkov N. and Mirenkova T. Multimedia Skeletons and “Filmification” of Methods // Proc. First Intern. Conf. Visual Information Systems. Melbourne, Australia: Victoria University, 1996. P. 58–67.
3. Watanobe Y., Mirenkov N., Yoshioka R., and Monakhov O. Filmification of methods: A visual language for graph algorithms // J. Visual Languages and Computing. 2008. No. 1. P. 123–150.
4. Gamma E., Helm R., Johnson R., and Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA.: Addison-Wesley, 1994. 416 p.
5. Монахов О. Г. Эволюционный синтез алгоритмов на основе шаблонов // Автометрия. 2006. № 1. С. 116–126.
6. Монахов О. Г., Монахова Э. А. Синтез новых семейств оптимальных регулярных сетей на основе эволюционных вычислений и шаблонов функций // Автометрия. 2004. № 4. С. 106–116.
7. Monakhov O. and Monakhova E. An Algorithm for Discovery of New Families of Optimal Regular Networks // Proc. 6th Inter. Conf. on Discovery Science (DS 2003), Oct. 17–20, 2003, Sapporo, Japan. Lecture Notes in Artificial Intelligence. 2003. V. 2843. P. 244–254.

8. *Монахов О. Г.* Программа эволюционного поиска математических описаний недоопределенных функций на основе шаблонов // Свид. об официальной регистрации программы на ЭВМ № 2011611546. М.: Федеральная служба по интеллектуальной собственности, патентам и товарным знакам, 2011.
9. *Goldberg D. E.* Genetic Algorithms, in Search, Optimization and Machine Learning. Reading, MA.: Addison-Wesley, 1989. 432 p.
10. *Koza J.* Genetic Programming. Cambridge: The MIT Press, 1992. 814 p.
11. *Monakhova E., Monakhov O., and Mukhoed E.* Genetic Construction of Optimal Circulant Network Designs // LNCS. 1999. V. 1596. P. 215–223.
12. *Sloane N. J. A. and Plouffe S.* The Encyclopedia of Integer Sequences. London: Academic Press, 1995. 587 p.
13. <http://oeis.org> — On-Line Encyclopedia of Integer Sequences.
14. *Koshy T.* Catalan numbers with applications. Oxford: Oxford University Press, 2009. 422 p.