НАУЧНОЕ ИЗДАНИЕ МГТУ ИМ. Н. Э. БАУМАНА
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС77 - 48211. Государственная регистрация №0421200025. ISSN 1994-0408
электронный научно-технический журнал

# Метод многовариантного эволюционного синтеза моделей на основе темплейтов

**# 03, март 2013**
**DOI: 10.7463/0313.0550970**
**Монахов О. Г.**
УДК 621.3 + 681.3

Россия, Новосибирск, Институт вычислительной математики и математической геофизики СО РАН.
monakhov@rav.sscc.ru

**Введение и постановка задачи.**

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

Пусть темплейт $T$ алгоритма $A$ содержит (неопределенные) параметры $P = \{ p_k \}, k \geq 0,$ описывающие значения целочисленных и действительных коэффициентов и переменных, значения индексов, величины инкрементов и декрементов, а также представляющие знаки переменных, логические операции и отношения, типы округления переменных. Пусть темплейт $T$ содержит множество (неизвестных) предикатов и функций (формул) алгоритма $FM = \{ f_n \}, n \geq 0$. При задании значений параметров $P$ и определении предикатов и функций (формул) $FM$ в данном темплейте $T$ мы получаем (синтезируем) алгоритм $A(T,P,FM)$. Если задан набор значений входных параметров алгоритма $A$ - $\{X_i\}$, $1 \leq i \leq N$, и задан набор значений выходных параметров $\{Y_i\}$, $1 \leq i \leq N$, которые необходимо получить в результате исполнения алгоритма, то качество работы алгоритма будет характеризовать функция $F$, оценивающая величину рассогласования между требуемыми $Y$ и полученными выходными значениями

алгоритма $Y' = A(T, P, FM, X)$ при данных входных значениях. Кроме того, функция $F$ может оценивать не только точность, но и сложность получаемого алгоритма (например, время выполнения, число итераций, сложность (длину) формул).

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

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

для всех $1 \le i \le N$, при любых других значениях параметров $P$ и функций $FM$.

Для решения данной проблемы в работе рассматривается основанный на темплейтах многовариантный эволюционный алгоритм (Template-based Multi-variant Evolutionary Synthesis - TMES) для автоматизированного синтеза (открытия) алгоритмов, оптимизирующих заданный критерий качества $F$. Этот подход основан на интеграции генетических алгоритмов (ГА) [4] и модификации метода генетического программирования (ГП) [5] с использоварием представления особей множеством выражений (MultiExpression Programming - MEP)[6] и позволяет получить новые свойства, не имеющиеся у исходных компонентов (меньшее время исполнения и синтез алгоритмов с большей циклической сложностью и наличием рекурсивных функций, по сравнению с ГР, и синтез новых функций и предикатов, что достаточно трудно при ГА).

Эти свойства данного подхода основаны на предварительном знании прикладной области и выборе обобщенной вычислительной схемы алгоритма, задаваемой в виде темплейта. 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 темплейтов алгоритмов для ограничения 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 темплейт искомого алгоритма, данный подход дает нам традиционное генетическое программирование. С другой стороны, если мы знаем полную структуру алгоритма и не знаем только некоторые его параметры, данный подход дает нам традиционный генетический алгоритм для поиска оптимальных параметров. Примером такой параметрической

оптимизации на основе заданного темплейта с помощью генетического алгоритма служит алгоритм предназначенный для открытия аналитических описаний новых плотных семейств оптимальных регулярных сетей, представленный в [3]. Отметим, что используемый многовариантный алгоритм эволюционного синтеза, в отличии от стандартного алгоритма генетического программирования, (1) является гибридным (ГП+ГА), (2) имеет два вида хромосом – линейные (для ГА) и последовательность инструкции (для ГП), и, соответственно, обобщенный (для двух видов) набор генетических операторов, (3) основан на использовании темплейтов с различной степенью специализации (в отличии от стандартного ГП, где поиск функции начинается «с нуля»), (4) позволяет осуществлять поиск заданного множества неизвестных функций и параметров темплейта, (5) использует многовариантное представление особей множеством выражений для каждой функции. Для выбора темплейта есть несколько способов: (1) знания и интуиция эксперта-исследователя, имеющего некоторую модель процесса или объекта, включающую неизвестные функции и параметры, (2) некоторые библиотеки темплейтов (шаблонов, скелетонов), например, из [1-3], (3) некоторые системы индуктивного вывода, позволяющие получить предварительную (недоопределенную) схему алгоритма, используемую как темплейт.

В первой части работы будет представлено краткое описание алгоритма многовариантного эволюционного синтеза, во второй части - получены оценки трудоемкости данного алгоритма, показано влияние степени специализации шаблона на пространство поиска, проведено сравнение с ранее предложенным стандартным (простым) методом эволюционного синтеза [7] и показано значительное (в несколько раз) сокращение времени поиска при использовании многовариантного эволюционного алгоритма синтеза.

## 1. Алгоритм многовариантного эволюционного синтеза.

Алгоритм многовариантного 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 недоопределенные параметры $p_k$ и функции (формулы) $f_n$ темплейтов, необходимые для построения заданного алгоритма $A(T,Gen)$ на основе выбранного темплейта $T$ : $Gen=\{P,FM\}=\{p_1,p_2,...,p_k;f_1,f_2,...,f_n\}, \quad k,n\geq 0$. Каждая популяция является

множеством структур данных $Gen$ и определяет множество алгоритмов $A(T,Gen)$, образующихся из данного темплейта $T$.

Основная идея алгоритма синтеза состоит в эволюционном преобразовании множества хромосом (параметров и формул темплейта) в процессе естественного отбора с целью выживания "сильнейшего". В нашем случае этими особями являются алгоритмы, имеющие наименьшее значение целевой функции. Алгоритм начинается с генерации начальной популяции. Все особи в этой популяции создаются случайно, затем отбираются наилучшие особи и запоминаются. Для создания популяции следующего поколения (следующей итерации), новые особи формируются с помощью генетических операций селекции (отбора), мутации, кроссовера и добавления новых элементов (для сохранения разнообразия популяции).

Примем, что целевая функция (fitness function, функция качества, функция пригодности) $F$ вычисляет сумму квадратов отклонений выходных данных алгоритма $A(T,Gen)$ от заданных эталонных значений $Y_i$ для определенных наборов множества входных данных $X_i, 1 \le i \le N$, : $F = \sum_{i=1}^{N} [A(T,Gen,X_i) - Y_i]^2$. Целью алгоритма является поиск минимума $F$. На практике, если получается несколько решений с одинаковым значением целевой функции, то выбиралось решение с меньшой оценкой структурной сложности, т.е. с меньшой общей длиной (суммой числа элементов) формул решения.

Для представления и реализации основной структуры данных - хромосомы $Gen$ был предложен подход, основанный на комбинации линейной структуры хромосомы для параметров $p_k$ (как в генетических алгоритмах [4]) и последовательной операторной структуры хромосомы для каждой из функций (формул) $f_n$ (как в генетическом программировании с использованием множества выражений [5]). Линейная структура хромосомы $Gen$ используется для представления параметров $p_k$ темплейта $T$ : значения целочисленных и действительных коэффициентов и переменных, значения индексов, величины инкрементов и декрементов, а также знаков переменных, логических операций и отношений, типов округления переменных, например: $\{[\,5\,][1.2][-1][+][\&][\ge][\lfloor x \rfloor]\}$.

Операторные структуры в хромосоме $Gen$ используются для представления каждой из функций (формул) $f_n$. Суть заключается в представлении функции интерпретируемым кодом, каждая операция которого

будет считаться отдельной функцией. Первой операцией всегда будет инструкция вызова переменной данной функции. Выполнение происходит сверху вниз и аргументами инструкций могут выступать лишь записи, расположенные ранее, поэтому исполнение получается линейным. Генетическое решение, в этом случае, представляет сразу множество функций-выражений (последовательностями от первого до каждого текущего оператора), что позволяет, в отличии от стандартного ГП, оценить одновременно множество выражений, представленных операторной последовательностью и сократить время поиска оптимального решения. При этом, в качестве оценки данной хромосомы, из множества полученных вариантов выбирается оценка выражения с минимальным (максимальным) значением целевой функции. Переменные и константы формулы $f_n$ образуют множество терминальные символов - $TS$, а операции, используемые в формуле $f_n$, образуют множество нетерминальных символов - $NT$. Пусть $f_n = (x+2)/\sqrt{ax-5}$, тогда $TS = \{x, a, 2, 5\}$, $NT = \{+, -, /, *, \sqrt{\ }\}$ и пример представления данной функции в виде последовательной операторной структуры с интерпретируемым кодом показан в таблице 1, где $N$ – номер операции, $K$ - сама операция (инструкция), $E$ - получаемое выражение (формула) для данной инструкции, $F$ - значение целевой функции для каждой инструкции.

Таблица 1. Пример представления функции в виде последовательной операторной структуры.

| $N$ | $K$ | $E$ | $F$ |
|---|---|---|---|
| 1 | $x$ | $E_1 = x$ | $F_1 = 5$ |
| 2 | 2 | $E_2 = 2$ | $F_2 = 2$ |
| 3 | + 1, 2 | $E_3 = x + 2$ | $F_3 = 7$ |
| 4 | $a$ | $E_4 = a$ | $F_4 = 6$ |
| 5 | * 4, 1 | $E_5 = ax$ | $F_5 = 30$ |
| 6 | 5 | $E_6 = 5$ | $F_6 = 5$ |
| 7 | - 5, 6 | $E_7 = ax - 5$ | $F_7 = 25$ |
| 8 | $\sqrt{\ }$ 7 | $E_8 = \sqrt{ax-5}$ | $F_8 = 5$ |
| 9 | / 3, 8 | $E_9 = (x+2)/\sqrt{ax-5}$ | $F_9 = 1,4$ |

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

Таким образом, для фиксированных значений функций $f_n$ и параметров $p_k$ мы можем вычислять алгоритмы $A(T,Gen,X_i)$, порожденные на основе заданных темплейтов $T$ и полученных в ходе эволюции хромосом $Gen$, для требуемых наборов множества входных данных $X_i, 1 \le i \le N$.

Оператор *мутации* применяется к особям, случайно выбранным из текущей популяции с вероятностью $p_{mut} \in [0,1]$. Мутация линейной части хромосомы $Gen$ состоит в изменении значения случайно выбранного параметра $p_k$ на другую, случайно выбранную величину из множества допустимых значений. Мутация операторной части хромосомы $Gen$ состоит в изменении значения оператора или операндов случайно выбранной инструкции 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 $p_{cros} \in [0,1]$. Кроссовер состоит в порождении двух новых особей путем обмена частями хромосом родителей (обмен подчастями линейных структур и обмен подпоследовательностями для операторных структур хромосомы $Gen$). 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 $F$. Отметим, что при оценке приспособленности особи для каждой инструкции решения рассматривается отдельная формула (выражение), и для нее вычисляется значение целевой функции (среднеквадратическое отклонение относительно эталонных данных). Оценка особи происходит по минимальному значению целевой функции среди всех выражений, представляющих ее.

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

*Первая итерация*: 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*. *Промежуточная итерация*: шаг от текущей к следующей популяции. Основной шаг алгоритма состоит в создании нового поколения особей на основе массива *best*, используя операции селекции, мутации, кроссовера и добавления новых элементов. После оценки целевой функции для каждой особи в поколении проводится сравнение величин этих функций с величинами 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 нем все остальные элементы на единицу вниз. Таким образом, лучшие элементы локализуются в верхней части массива *best*. *Последняя итерация (критерий остановки)*: итерации заканчиваются либо после исполнения заданного числа шагов, либо после нахождения оптимального алгоритма $A(T,Gen)$.

После выполнения алгоритма синтеза, мы получаем алгоритм $A(T,Gen)$, имеющий минимальное среднеквадратическое отклонение выходных данных от заданных эталонных значений $Y_i$ для определенных наборов множества входных данных $X_i, 1 \le i \le N$.

**2. Экспериментальные результаты.**

Алгоритм эволюционного синтеза на основе темплейтов был успешно применен для автоматизированного получения (открытия) следующих алгоритмов: алгоритм Евклида, вычисления наибольшего общего делителя двух целых чисел, вычисления степени и факториала натурального числа, определения минимального (максимального) элемента массива, определения суммы квадратов элементов массива, вычисления скалярного произведения векторов, определение рекурсивных формул последовательностей Фибоначчи и Трибоначчи, вычисления суммы двух матриц, рекурсивных алгоритмов сортировки методом пузырька, слияния и Шелла, алгоритмов вычисления корней уравнений, параллельных алгоритмов балансировки нагрузки в многопроцессорных системах, алгоритмов определения минимального покрывающего дерева и кратчайших путей в графе. Данный подход был также

применен для открытия аналитических описаний новых плотных семейств оптимальных регулярных сетей [8] и функции расстояния для циркулянтных графов степени 4, для оптимизации торговых стратегий на финансовых рынках, для синтеза рекурсивных функций для операций работы со списками и для решения математических задач определения членов последовательностей из IQ теста (теста коэффициента интеллектуальности).

Эволюционный многовариантный синтез алгоритмов на основе темплейтов реализован на языке программирования C, темплейты также задаются на этом языке.

Проведем сравнение эффективности поиска для простого [7] и многовариантного эволюционных методов синтеза. При реализации алгоритма эволюционного синтеза основную трудность представляет поиск неопределённых функций с помощью генетического программирования. В этом случае, время вычисления выражения, представленного последовательностью инструкций длинной $L$ операторов равно $T_v = L * T_{op}$, где $T_{op}$ - среднее время выполнения одного оператора. При многовариантном методе мы получаем $L$ оценок целевой функции (для каждого оператора) при одном вычислении выражения, представленного последовательностью инструкций длинной $L$ операторов, а при простом методе только одну оценку для всего выражения. Т.к. основное время при эволюционном синтезе затрачивается на получение оценок целевой функции, то при использовании многовариантного эволюционного метода синтеза можно получить ускорение поиска решения порядка $O(L)$, где $L$ - длинна последовательности операторов для представления выражения (функции). Обычно используется длинна последовательности операторов $L$ порядка $\sim 10 - 10^3$. Для экспериментальной оценки получаемого ускорения при использовании многовариантного эволюционного метода синтеза приведем несколько примеров поиска различных функций.

Рассмотрим поиск полиномиальной функции $T = x^8 + x^4 + x^2 - x - 2$ при разной степени специализации темплейта. Число итераций и размер популяции выбраны экспериментально на основе параметров из [4, 5]. Значения основных параметров в экспериментах следующие: размер популяции – 100, число итераций – 30000, частота мутации – 0.15, частота кроссовера – 0.7, длинна последовательности операторов – 20.

Пусть заданы 8 пар вход - выход для $x = 0,1,...,7$, и темплейт $T(x) = T(T_i, x^4 + x^2 - x - 2), i = 1,2$, где $x^4 + x^2 - x - 2$ - известный терм и $T_i$ - известная часть функции (см. таблицу 2). Множество терминальных символов в этом случае $TS = \{x, Cr\}$, где $Cr$ - элемент множества случайных натуральных чисел. Набор операций, используемый для синтеза формул, следующий: $NT = \{+, -, /, *\}$. В таблице 2 приведены результаты сравнения характеристик алгоритмов на основе темплейтов простого эволюционного и многовариантного эволюционного методов синтеза - времени поиска решения $t_p$, $t_m$ (с), числа поколений $G_p$, $G_m$ до получения решения (соответственно для каждого метода), а также величину ускорения времени поиска решения $S_m = t_p / t_m$ для многовариантного метода - при различных темплейтах. Для каждого темплейта приведены средние значения, полученные по результатам 100 экспериментов.

Таблица 2. Сравнение характеристик алгоритмов эволюционного синтеза.

| $i$ | $T$ | $t_p$ | $G_p$ | $t_m$ | $G_m$ | $S_m$ |
|---|---|---|---|---|---|---|
| 1 | $T_1 = x^2$ | 43,03 | 19010 | 0,023 | 1,68 | 1871 |
| 2 | $T_2 = x^4$ | 0,076 | 38,53 | 0,0078 | 0,001 | 9,75 |

Из таблицы 2 видно, что применение многовариантного эволюционного метода синтеза позволяет получить значительное ускорение времени поиска (9,75 и 1870 раз для различных темплейтов). Отметим также, что увеличение степени специализации темплейта позволяет значительно уменьшить время поиска (на три порядка).

Рассмотрим поиск другой полиномиальной функции вида: $T = 2*(x + 2*(x + 2*(x + 2*(x + 2*(x + 2)))))$ при различной степени специализации темплейта. Пусть заданы 8 пар вход - выход для $x = 0,1,...,7$, и темплейт $T(x) = T(T_i, F_i), i = 1,...,4$, где $F_i$ - известная часть функции, а $T_i$ - неизвестная часть функции (см. таблицу 3). Множество терминальных символов в этом случае $TS = \{x, Cr\}$, где $Cr$ - элемент множества случайных натуральных чисел. Набор операций, используемый для синтеза формул, следующий: $NT = \{+, -, /, *\}$. В таблице 3 приведены результаты сравнения характеристик

алгоритмов на основе темплейтов $T$ простого эволюционного и многовариантного эволюционного методов синтеза - времени поиска решения $t_p, t_m$ (с), числа поколений $G_p, G_m$ до получения решения (соответственно для каждого метода), а также величину ускорения времени поиска решения $S_m = t_p / t_m$ для многовариантного метода - при различных темплейтах. Для каждого темплейта приведены средние значения, полученные по результатам 100 экспериментов.

Таблица 3. Сравнение характеристик алгоритмов эволюционного синтеза.

| $i$ | $T$ | $t_p$ | $G_p$ | $t_m$ | $G_m$ | $S_m$ |
|---|---|---|---|---|---|---|
| 1 | $T_1 = 2*(x + F_1)$, | 0,059 | 25,9 | 0,012 | 0,001 | 4,92 |
| 2 | $T_2 = 2*(x + 2*F_2)$ | 0,45 | 231,15 | 0,034 | 2,27 | 13,2 |
| 3 | $T_3 = 2*(x + 2*(x + F_3))$ | 18,5 | 9321,4 | 1,56 | 211 | 11,9 |
| 4 | $T_4 = 2*(x + 2*(x + 2*F_4))$ | 48,4 | 24532,2 | 1,86 | 270 | 26 |

Из таблицы 3 видно, что применение многовариантного эволюционного метода синтеза позволяет получить значительное ускорение времени поиска (от 4,92 до 26 раз для различных темплейтов). Отметим также, что увеличение степени специализации темплейта позволяет в этом случае значительно уменьшить время поиска (в несколько раз).

Рассмотрим для примера поиск функции с тригонометрическими, показательными и экспоненциальными членами $T = e^x * \sin(x) * \cos(x) * x^3$ при различной степени специализации темплейта.

Пусть заданы 15 пар вход - выход для $x = 0,1,...,14$, и темплейт $T(x) = f(T_i), i = 1,...,4$, где $T_i$ - неизвестная часть функции (см. таблицу 4). Множество терминальных символов в этом случае $TS = \{x, Cr\}$, где $Cr$ - элемент множества случайных натуральных чисел. Набор операций, используемый для синтеза формул, следующий: $NT = \{+, -, /, *\}$. В таблице 4 приведены результаты сравнения характеристик алгоритмов на основе темплейтов простого эволюционного и многовариантного эволюционного методов синтеза - времени поиска решения $t_p$ , $t_m$ (с), числа поколений $G_p$ , $G_m$ до получения решения

(соответственно для каждого метода), а также величину ускорения времени поиска решения $S_m = t_p / t_m$ для многовариантного метода - при различных темплейтах. Для каждого темплейта приведены средние значения, полученные по результатам 100 экспериментов.

Таблица 4. Сравнение характеристик алгоритмов эволюционного синтеза.

| $i$ | $T$ | $t_p$ | $G_p$ | $t_m$ | $G_m$ | $S_m$ |
|---|---|---|---|---|---|---|
| 1 | $T = T_1 * x$ | 995,3 | 5000 | 113,7 | 14,4 | 8,75 |
| 2 | $T = T_2 * \cos(x)$ | 696,6 | 292,6 | 46,6 | 5,5 | 14,95 |
| 3 | $T = T_3 * \cos(x) * x$ | 150,5 | 31,9 | 8,7 | 0,4 | 17,3 |
| 4 | $T = T_4 * \cos(x) * x * x$ | 2,55 | 11,7 | 1 | 0 | 2,55 |

Из таблицы 4 видно, что применение многовариантного эволюционного метода синтеза позволяет получить значительное ускорение времени поиска (от 2,5 до 17 раз для различных темплейтов). Отметим также, что увеличение степени специализации темплейта позволяет в этом случае значительно уменьшить время поиска (в несколько раз).

Рассмотрим влияние метода синтеза при поиске описания последовательности Маккея – Гласса (Mackey - Glass), имеющей в дискретной форме следующий вид: $x(n) = 0.9 * x(n-1) + 0.2 * x(n-17) / (1 + x(n-17)^{10})$. Пусть заданы 100 пар вход - выход для $x = 0,1,...,99$, множество терминальных символов $TS = \{x, Cr\}$, множество примитивных функций $NT = \{+, -, /, *, pow(5, x)\}$. При использовании темплейта, выражающего зависимость от предыдущего и любого другого члена последовательности в виде $x(n) = a * x(n-1) + b * f(x(n-d))$, где $a \in [0,2], b \in [0,2]$ - действительные, $d \in [0,21]$ - целые числа, получаем точную формулу последовательности Маккея – Гласса в случае простого эволюционного метода синтеза при числе поколений $G_p$=801 и времени поиска решения $t_p$ = 905 с., а в случае многовариантного эволюционного метода синтеза при числе поколений $G_m$=194 и времени поиска решения $t_m$ = 65,3 с., т.е. ускорение времени поиска при многовариантном эволюционном методе составляет $S_m$=14,25.

**Заключение.**

Дано описание многовариантного эволюционного метода синтеза, объединяющего преимущества генетических алгоритмов и генетического программирования, основанного на последовательной операторной структуре хромосомы, темплейтах (шаблонах, скелетонах) алгоритмов и заданном множестве пар входных - выходных данных. Проведено исследование влияния множества оценок хромосомы и степени специализации темплейта на характеристики алгоритма поиска при эволюционном синтезе. Получены оценки трудоемкости данного алгоритма, проведено сравнение со стандартным методом эволюционного синтеза и показано значительное (в несколько раз) сокращение времени поиска при использовании многовариантного эволюционного алгоритма синтеза.

## Список литературы


1. Cole M. Algorithmic Skeletons: Structured Management of Parallel Computation. Cambridge: The MIT Press, 1989. 131 p.
2. Watanobe Y., Mirenkov N., Yoshioka R., Monakhov O. Filmification of methods: A visual language for graph algorithms // Journal of Visual Languages and Computing. 2008. Vol. 19, no. 1. P. 123-150. http://dx.doi.org/10.1016/j.jvlc.2006.09.003
3. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1994. 416 p.
4. Goldberg D.E. Genetic Algorithms, in Search, Optimization and MachineLearning. Addison-Wesley, Reading, MA, 1989.
5. Koza J. Genetic Programming. Cambridge: MIT Press, 1992.
6. Oltean M. Multi Expression Programming. Technical Report. Babes-Bolyai Univ., Romania, 2006.
7. Монахов О.Г. Эволюционный синтез алгоритмов на основе шаблонов // Автометрия. 2006. Т. 42, № 1. С. 116-126.
8. Monakhov O.G., Monakhova E.A. An Algorithm for Discovery of New Families of Optimal Regular Networks // Proc. of 6th Inter. Conf. on Discovery Science (DS 2003). Sapporo, Japan, LNAI, N2843, Springer-Verlag, Berlin Heidelberg, 2003. P. 244-254.

SCIENTIFIC PERIODICAL OF THE BAUMAN MSTU
SCIENCE and EDUCATION
EL № FS77 - 48211. №0421200025. ISSN 1994-0408
electronic scientific and technical journal

# Method of multi-variant evolutionary synthesis of models based on templates

**# 03, March 2013**
**DOI: 10.7463/0413.0550970**
**Monahov O.G.**

Russia, Novosibirsk, Institute of Computational Mathematics and Mathematical Geophysics SB RAS
monakhov@rav.sscc.ru

The author describes an algorithm of a multi-variant evolutionary synthesis which combines advantages of genetic algorithms and genetic programming, based on a sequential operator structure of chromosome and templates (patterns, skeletons) of algorithms and a given set of pairs of input - output data. Influence of many variants of estimating chromosomes and the degree of the template's specialization on search algorithm efficiency in the evolutionary synthesis was investigated. The author obtained estimates of the algorithm complexity and compared them with ones for the standard method of the evolutionary synthesis; this comparison showed a significant (by several times) decrease in search time by using the multi-variant evolutionary synthesis algorithm.

**Publications with keywords:** skeleton, design pattern, genetic algorithm, evolutionary synthesis, template
**Publications with words:** skeleton, design pattern, genetic algorithm, evolutionary synthesis, template

References

1. Cole M. *Algorithmic Skeletons: Structured Management of Parallel Computation*. Cambridge, MIT Press, 1989. 131 p.
2. Watanobe Y., Mirenkov N., Yoshioka R., Monakhov O. Filmification of methods: A visual language for graph algorithms. *Journal of Visual Languages and Computing*, 2008, vol. 19, no. 1, pp. 123-150. http://dx.doi.org/10.1016/j.jvlc.2006.09.003
3. Gamma E., Helm R., Johnson R., Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1994. 416 p.
4. Goldberg D.E. *Genetic Algorithms, in Search, Optimization and MachineLearning*. Addison-Wesley, Reading, MA, 1989.
5. Koza J. *Genetic Programming*. Cambridge, MIT Press, 1992.

6. Oltean M. *Multi Expression Programming*. Technical Report. Babes-Bolyai Univ., Romania, 2006.
7. Monakhov O.G. Evoliutsionnyi sintez algoritmov na osnove shablonov [Evolutionary synthesis of algorithms based on templates]. *Avtometriia*, 2006, vol. 42, no. 1, pp. 116-126.
8. Monakhov O.G., Monakhova E.A. An Algorithm for Discovery of New Families of Optimal Regular Networks. *Proc. of 6th Inter. Conf. on Discovery Science (DS 2003)*. Sapporo, Japan, LNAI, N2843, Springer-Verlag, Berlin Heidelberg, 2003, pp. 244-254.