

ОСОБЕННОСТИ ПОСТРОЕНИЯ ОПЕРАТОРОВ МУТАЦИИ И КРОССОВЕРА В ВЕКТОРНОМ ВАРИАНТЕ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

В.А. Егоров, Г.Н. Рогачев

Самарский государственный технический университет
443100, г. Самара, ул. Молодогвардейская, 244
E-mail: innewworld@mail.ru

Работа посвящена вопросам построения и использования операторов мутации и кроссовера особей в векторном варианте генетического программирования. В статье дан анализ различных подходов к решению этой проблемы и рассмотрена эффективность каждого из них при эволюционном синтезе систем управления.

Ключевые слова: генетическое программирование, генетические операторы, автоматическое управление, кроссовер, мутация, гибридный автомат.

Эволюционные алгоритмы являются методами поиска глобального экстремума. К их преимуществам относятся устойчивость от попадания в локальный экстремум, отсутствие особых требований к поверхности целевой функции, возможность использования для решения широкого класса задач [1]. Эволюционные алгоритмы используют в своей работе механизмы, схожие с механизмами естественного отбора в живой природе. Так же, как и в природе, эволюционные алгоритмы оперируют множеством уникальных особей. Одним из вариантов эволюционного поиска является генетическое программирование (ГП). Каждая особь в ГП – это некоторая функция, представленная в виде графа. Алгоритм поиска решения аналогичен алгоритму естественного отбора в природе и включает ряд этапов.

1. Создание начальной популяции – множества особей с уникальным набором признаков (генерирование множества функций).
2. Выбор из общего множества некоторого числа функций, в наибольшей степени соответствующих требованиям задачи.
3. Кроссовер, или скрещивание особей. Комбинирование пар выбранных функций между собой. Результатом каждого такого комбинирования является некоторая новая функция, содержащая признаки обеих родительских функций.
4. Применение оператора мутации. Производится произвольное изменение некоторых выбранных функций.
5. Оценка полученных в результате произведенных операций решений. Если даже наилучшее из решений нас не устраивает, то переходим к шагу 2 (эволюция продолжается).

Пространством поиска генетического программирования является множество всех возможных рекурсивных композиций функций $S=F \cup T$. Функциональное множество $F = \{F_1, F_2, \dots, F_n\}$ – множество возможных внутренних узлов в деревьях, представляющих программы в генетическом программировании. Каждая из этих функций характеризуется таким параметром, как арность – количество ее аргументов. $T = \{T_1, T_2, \dots, T_m\}$ – это множество листовых узлов в деревьях, его называют

терминальным множеством. Элементы из терминального множества также могут быть рассмотрены как элементы функционального множества с нулевой арностью.

Генетическое программирование получило значительное распространение и нашло применение в различных задачах поиска решений в виде той или иной скалярной функции. Однако в литературе практически не встречаются упоминания об использовании генетического программирования в ситуациях, когда решение является векторной функцией. Такая проблема возникает, например, в задачах прямого синтеза регуляторов систем автоматического управления (САУ).

Общеизвестно, что одной из ключевых особенностей современности является повсеместное использование устройств с компьютерным управлением. Это бытовые приборы, средства связи и передвижения и т. д. Эволюция данных устройств характеризуется перманентно возрастающей сложностью задач, решаемых их системами управления. Предъявляются все более жесткие требования к надежности, эффективности, компактности, срокам разработки и модернизации систем управления. Это приводит к необходимости разработки новых подходов к построению САУ.

Одним из недостатков большинства методов синтеза САУ является упрощенный взгляд на сложное взаимодействие непрерывной (объект управления) и дискретной (цифровой регулятор) частей, рассмотрение всей САУ как чисто дискретной или непрерывной системы. При этом неизбежны потери, вызванные упрощением модели объекта или регулятора. Другой существенный недостаток традиционных подходов – многоступенчатость. Так, результатом первого этапа синтеза цифрового регулятора, как правило, является определение непрерывного регулятора, решающего стоящую перед САУ задачу. Далее следует этап «переоборудования», вычисления дискретного аналога найденного ранее непрерывного регулятора. И, наконец, имея описание дискретного регулятора в виде разностных уравнений или дискретной передаточной функции, записывают алгоритм и программу работы цифрового регулятора. Частично устранить указанные недостатки может использование прямых методов синтеза. Одним из них является императивный¹ метод. Охарактеризуем его.

Анализ поведения регулятора, работающего в рамках компьютерной системы управления, позволяет выделить две циклически сменяющиеся фазы его работы. Пока ни в самой системе, ни в ее «окружении» ничего не происходит, реакции системы управления отсутствуют. Это – фаза ожидания. Появление стимула, воздействия внешнего (сгенерированного средой) и/или внутреннего (сгенерированного самой системой – ее управляемой или управляющей частью) вызывает ответную реакцию. Под реакцией следует понимать либо изменение, либо, по крайней мере, попытку изменения каких-либо характеристик управляемого процесса. Это – фаза реагирования. Наиболее существенными элементами, задающими поведение конкретного регулятора, являются правила его перехода от фазы ожидания к фазе реагирования и

¹ В когнитивной науке, исследующей процессы познания, существует деление знаний на декларативные и императивные. Декларативные знания – это правила связи, а императивные – это правила преобразования. Декларативные знания представляют собой утверждения об объектах, их свойствах и отношениях между ними. Это – факты из той или иной предметной области. Однако знание может быть представлено не только в форме статичных структур – декларативного знания, но и в форме операций (императивное, или процедурное, знание). Императивные знания описывают правила преобразования объектов предметной области. Это могут быть рецепты, алгоритмы, методики, инструкции, стратегии принятия решений. Примером императивного знания являются системы продукции. Они представляют собой набор пар типа «условие – действие». Если на вход системы продукций попадает одно из «условий», то оно автоматически приводит к соответствующему «действию».

то, каким образом определяется реакция этого регулятора на тот или иной стимул. Именно эти элементы и положены в основу императивного метода описания. Процедура описания регуляторов вне зависимости от их формы сводится к заданию одного или нескольких наборов вида «условие – действие», т. е. продукций. «Условие» – логическое высказывание. Оно определяет правило перехода регулятора от фазы ожидания к фазе реагирования. «Действие» – правило вычисления реакции системы управления на стимул. Процедура синтеза регулятора может быть описана единообразно. Вне зависимости от конкретной задачи определению подлежит наполнение системы продукции (пар типа «условие – действие»).

Удобным способом представления компьютерной системы управления с заданным императивно регулятором является гибридный автомат [2]. Гибридный автомат полностью отражает специфику компьютерной системы управления как объекта моделирования, позволяя учесть при необходимости:

- гетерогенность, или наличие в компьютерной системе управления дискретных (регулятор) и непрерывных (объект управления) элементов;
- гомеостаз при наличии внешней по отношению к системе управления среды;
- цикличность функционирования регулятора;
- присущую задачам управления событийность;
- необходимость синхронизации процесса функционирования регулятора с физическими процессами в объекте управления;
- параллелизм физических процессов в объекте управления и вычислений в регуляторе.

Универсальная гибридно-автоматная модель САУ представляет собой простейший направленный граф (рис. 1). Вершина графа (состояние гибридного автомата) – это модель объекта управления, непрерывного физического процесса. Переходом моделируется работа контроллера [3]. Согласно логике работы системы управления возможная смена сигнала управления происходит при выполнении некоторого условия. Действие перехода – вычисление значения управляющего сигнала и передача его на объект управления. Такая модель весьма универсальна.



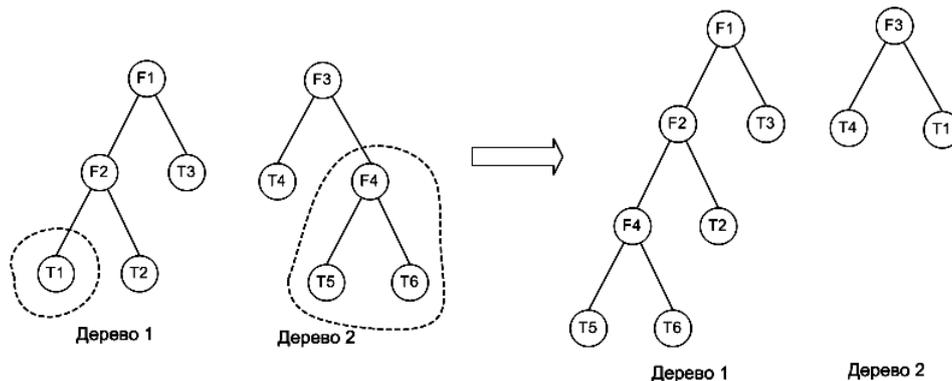
Р и с. 1. Представление в виде гибридного автомата системы автоматического управления с императивной моделью регулятора

Так, условие перехода может включать время (что справедливо для систем с дискретным временем), состояние (в случае систем с дискретными событиями, релейных и проч.) или их комбинацию. Вычисление управляющего воздействия может осуществляться по различным алгоритмам, характерным для того или иного закона управления. Кроме того, передача управляющего воздействия может происходить с временными задержками, потерей части информации и наложением шумовой составляющей, что имеет место в распределенных системах. Такая гибридно-автоматная модель позволяет осуществить прямой синтез САУ.

Задача синтеза может быть решена с использованием эволюционных подходов, в частности генетического программирования в его векторном варианте. Рассмотрим, какие особенности привносит векторный характер функции решения в реализацию специфических для генетического программирования операторов кроссовера

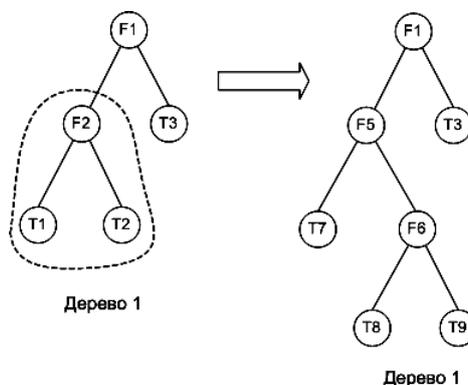
(также называемого оператором скрещивания или рекомбинации) и мутации, являющихся основной движущей силой поиска функционального решения в генетическом программировании.

В классическом (скалярном) виде оператор кроссовера представляет собой обмен частями двух функциональных деревьев относительно случайно выбранных «точек». Различают также одноточечный (рис. 2) и двухточечный кроссовер.



Р и с. 2. Процесс кроссовера в ГП

Оператор мутации представляет собой замену части функционального дерева на некоторую другую, сгенерированную случайным образом. Пример такого оператора приведен на рис. 3.



Р и с. 3. Процесс мутации в ГП

Схема использования данных операторов при поиске одной функции вполне очевидна. Однако в случае, когда особь содержит в себе сразу несколько функциональных решений, могут возникнуть проблемы при их использовании.

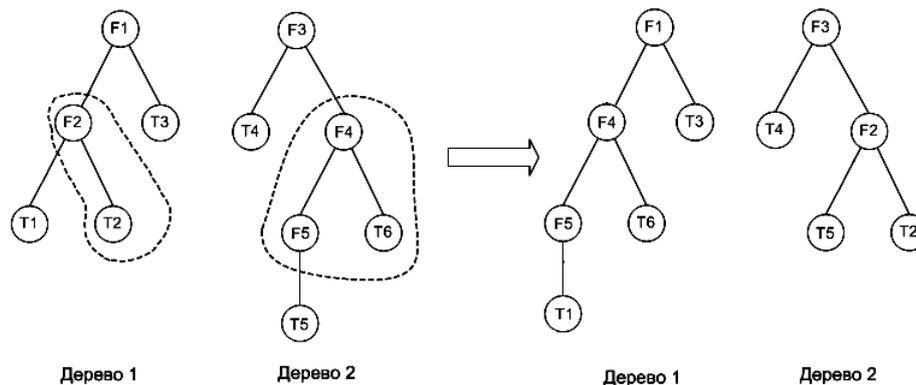
Особое внимание следует уделить оператору кроссовера. Можно выделить четыре особенности осуществления рекомбинации функциональных деревьев двух особей.

Во-первых, для приведенного примера возможна как попарная, так и перекрестная схемы применения оператора кроссовера к функциональным деревьям скрещиваемых особей. Попарная схема скрещивания подразумевает, что для i -того функционального дерева первой особи для осуществления рекомбинации может быть выбрано только i -тое функциональное дерево второй особи. Перекрестная схема крос-

совера подразумевает, что для любого i -того функционального дерева первой особи для осуществления рекомбинации может быть выбрано любое функциональное дерево второй особи.

Во-вторых, возможна полная или частичная схема скрещивания особей. Полная схема скрещивания подразумевает, что оператор кроссовера применяется последовательно к каждому функциональному дереву особи. Частичная схема скрещивания подразумевает, что оператор кроссовера будет применен только к части функциональных деревьев особи.

Третьим важным моментом является количество точек пересечения двух функциональных деревьев. Как показывает практика, в скалярном случае наибольшую эффективность имеет лишь двухточечный кроссовер (рис. 4).



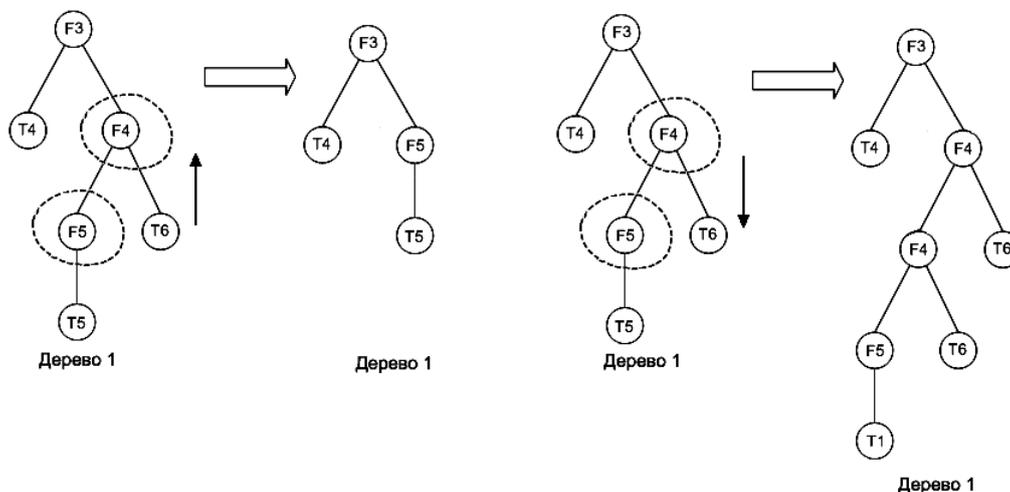
Р и с. 4. Двухточечный кроссовер

Четвертый важный момент заключается в том случае, если для каждого из функциональных деревьев особи задается больше одного функционального множества. В таком случае может возникнуть ситуация, при которой одна из точек пересечения первого дерева будет несовместима с соотносимой ей точкой пересечения во втором функциональном дереве. Необходимость использования дополнительных функциональных множеств возникает, например, когда в основное функциональное множество вводится функция ветвления (условный оператор ЕСЛИ...ТО), одним из параметров которой является результат вычисления некоторого логического выражения, построенного на основании другого функционального множества.

Для оператора мутации, так же как и для оператора рекомбинации, можно отметить два важных момента. Во-первых, подвергать мутации можно как все функциональные деревья особи, так и произвольное их количество. Во-вторых, возможна любая из трех схем мутации. Первая схема – это замена произвольного узла функционального дерева на сгенерированный случайно фрагмент (классическая схема мутации, представлена на рис. 3). Вторая схема – подмена функции (или константы) произвольно взятого узла на функцию равной аргументности. Т. е. если выбранным узлом является функция с двумя параметрами, то вместо нее подставляется выбранная случайным образом функция с двумя параметрами, взятая из того же функционального множества, что и подменяемая. Третья схема – это дублирующая мутация. Данный вариант оператора мутации подразумевает замену одного произвольного узла функционального дерева на другой, также выбранный случайно (рис. 5).

Частоту применения операторов рекомбинации и мутации, применяемых при эволюции популяции, целесообразно определять как непостоянную величину, зависящую от характера динамики изменения пригодности лучшей особи популяции.

При этом чем меньше прирост пригодности лучшей особи от популяции к популяции, тем больший процент особей подвергается мутации.



Р и с. 5. Оператор мутации с дублированием

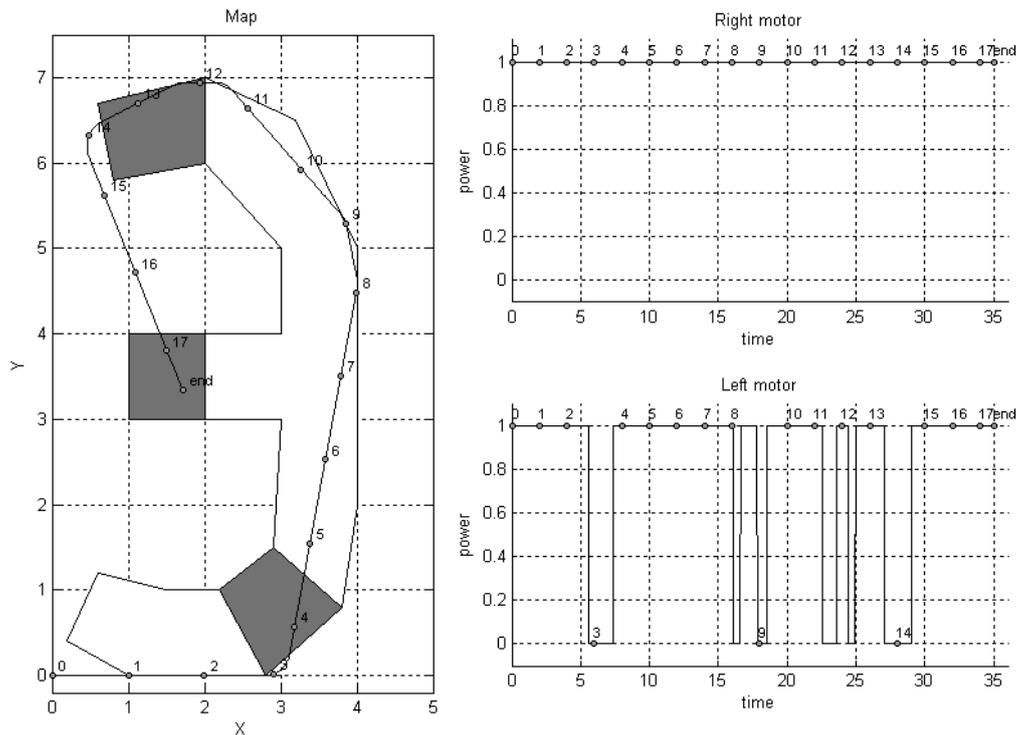
Рассмотренные ранее особенности операторов мутации и кроссовера в векторном варианте генетического программирования были учтены при разработке программного средства с рабочим названием GPD toolbox, реализующего механизмы поиска векторного варианта функции решения методом генетического программирования. В качестве среды разработки был выбран математический пакет MATLAB, позволяющий использовать мощный математический аппарат матричных вычислений для моделирования дискретных и непрерывных систем.

Далее была произведена апробация программного средства GPD toolbox на задачах синтеза САУ различными техническими объектами – как линейными, так и нелинейными. Так, например, решалась задача поиска оптимального по критерию обобщенной работы управления объектом вида «двойной интегратор». Подобный тип поведения объектов имеет довольно широкое распространение: в качестве примера можно привести задачу управления поворотом высотного крана [4] и углом наклона космического аппарата. Задача синтеза регулирующего устройства сводилась к задаче поиска двух функций – условия, определяющего момент срабатывания регулятора, и закона формирования управляющего воздействия. При этом исходные функциональные и терминальные множества для искомым функциональных деревьев различны. В результате произведенных исследований было установлено, что применение комбинации векторного генетического оператора кроссовера с частичной попарной схемой выбора рекомбинируемых функциональных деревьев, а также трех описанных схем операторов мутации позволило повысить эффективность поиска решения более чем на 15% (усредненный показатель, взятый на основании 100 запусков процедуры решения в GPD toolbox). Следует отметить, что для операторов кроссовера и мутации выбиралось именно то функциональное дерево особи, изменение которого данным оператором на предыдущем шаге повысило показатель пригодности наилучшим образом.

Далее была рассмотрена задача поиска оптимального по быстродействию векторного управления моделью транспортного средства Dubins Car:

$$\begin{aligned}\dot{x} &= (u_1 + u_2) \cos(\varphi) / 2; \\ \dot{y} &= (u_1 + u_2) \sin(\varphi) / 2; \\ \dot{\varphi} &= \pi(u_1 - u_2) / 2.\end{aligned}\quad (1)$$

В приведенной формуле φ – направление движения модели относительно оси x ; x , y – координаты положения модели в декартовой системе координат. Управление для такой модели представляет собой вектор из двух функций $[u_1, u_2]$, задающих уровень напряжения на левом и правом двигателях модели соответственно, что создает тяговое усилие для перемещения модели в пространстве.



Р и с. 6. Пример решения задачи оптимального по быстродействию векторного управления мобильным роботом

При этом множество функциональных деревьев, составляющее решение данной задачи, строится на основании единого функционального и терминального базиса. Используемые при поиске операторы мутации и рекомбинации, а также схемы их применения аналогичны приведенным в предыдущем примере. Отличие состоит в использовании дополнительной, перекрестной схемы применения оператора кроссовера. На основании результатов проведенных испытаний было установлено, что использование данной схемы рекомбинации позволяет повысить эффективность поиска решения более чем на 12%. Объясняется это тем, что искомые функциональные деревья содержат в себе фрагменты, описывающие сходное управляющее воздействие при определенном состоянии объекта. Примером может служить задача перемещения мобильного робота по некоторой траектории, на отдельных фрагментах которой оптимальным видом движения является прямолинейное. Обязательное условие такого движения – одинаковая скорость вращения тяговых колес робота. На рис. 6 в качестве примера приведен результат поиска оптимальной траектории, охватываю-

щей три ключевые точки (отмечены серыми четырехугольниками) и пролегающей преимущественно в пределах заданной области в виде буквы «Э». В правой части рисунка приведены графики уровня управляющего сигнала power (напряжения, передаваемого на двигатели) в каждый из моментов времени (time) для левого (left) и правого (right) двигателей. Из рисунка видно, что в траектории мобильного робота присутствуют участки, уровень управляющего сигнала на которых одинаков.

Операторы рекомбинации и мутации являются основной движущей силой поиска функционального решения в генетическом программировании. Естественное возрастание сложности решаемой задачи при векторном характере решения неизбежно приводит к повышению требований к алгоритмам функционирования данных операторов. Кроме того, важным фактором является выбор схем применения операторов кроссовера и мутации в соответствии с характером искомого решения до запуска алгоритма поиска.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Koza J.R.* Genetic Programming II: Automatic Discovery of Reusable Programs (Complex Adaptive Systems). – MA: MIT Press, 1994. – 768 с.
2. *Колесов Ю.Б.* Моделирование систем: Динамические и гибридные системы. – СПб.: БХВ-Петербург, 2006. – 224 с.
3. *Рогачев Г.Н.* Гибридный автомат как модель цифровой системы управления // Вестник СамГТУ. – 2007. – С. 59-63.
4. *Gustafsson T.* On the design and implementation of a rotary crane controller // Eur. J. Contr. – 1996. – Т. 2. – С. 166-175.

Статья поступила в редакцию 24 февраля 2011 г.

UDC 621.8

FEATURES OF CONSTRUCTION OF GENETIC OPERATORS OF THE MUTATION AND CROSSOVER BY SEARCH VECTOR OF FUNCTIONS BY MEANS OF GENETIC PROGRAMMING

V.A. Egorov, G.N. Rogachev

Samara State Technical University
244, Molodogvardeyskaya st., Samara, 443100

The paper refers to the usage and building of individuals crossover and mutation operators in vector genetic programming. In article the analysis of various approaches and their efficiency for evolutionary synthesis of control systems is given.

Keywords: *genetic programming, regulator synthesis, automatic control, genetic operators, hybrid automata.*

*Vladimir A. Egorov – Postgraduate student.
Genadiy N. Rogachev – Candidate of Technical Sciences, Associate professor.*