

УДК 519.711

В. И. КАУК, В. М. ЛЕВЫКИН, С. Ф. ЧАЛЫЙ,

ПРИМЕНЕНИЕ ПАРАДИГМЫ ЭВОЛЮЦИОННОЙ КЛАССИФИКАЦИИ ПРИ СОЗДАНИИ ФУНКЦИОНАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Проектирование сложных информационных систем и их элементов является длительным процессом, связанным с большими материальными и трудовыми затратами, причем сам процесс проектирования требует четкой взаимосвязи и согласования всех разрабатываемых элементов. Задающим элементом всего процесса проектирования является функциональная структура системы, определяющая все множество функций, реализуемых системой. Для ее реализации создается соответствующее обеспечение, в частности функциональное программное обеспечение (ФПО).

ФПО информационной системы в процессе ее создания и эксплуатации претерпевает революционные и эволюционные изменения. Революционные изменения радикально преобразуют ФПО, однако, при правильном проектировании системы, потребность в них возникает достаточно редко. Основной объем затрат приходится, как правило, на эволюционные изменения. Очевидно, что эволюционные изменения ФПО можно естественным образом описать с помощью методов эволюционных вычислений.

1. Эволюционные вычисления

Эволюционные вычисления возникли на основе исследований в области искусственного интеллекта. Эволюционные вычисления развиваются по следующим направлениям:

- генетические алгоритмы;
- эволюционные стратегии;
- системы классификации;
- генетическое программирование.

В соответствии с задачей создания и эволюции ФПО информационной системы среди этих направлений целесообразно выделить системы классификации и генетическое программирование.

Системы классификации основаны на наборе правил для машинного обучения и классификации. Каждое правило описывает некоторую ситуацию и связанное с ней действие. В процессе эволюции формируется такое множество правил, которое позволяет выполнять наилучшие действия для некоторого множества ситуаций во внешней среде.

Метод генетического программирования (ГП) разработан Дж. Коза (J. R. Koza) в 1992 году и основан на использовании эволюционных алгоритмов в области создания прикладных программ. Эволюционный алгоритм представляет собой оптимизационный метод, основанный на эволюции популяции индивидуумов. Каждый индивидуум содержит набор генов и характеризуется приспособленностью – многомерной функцией ее генов. Набор генов обычно описывается в виде символьных строк фиксированной длины. В процессе моделирования эволюции выполняется максимизация функции приспособленности с помощью многократного выполнения операторов отбора, рекомбинации («кроссинговера») и мутации геномов особей. На каждом шаге эволюции происходит отбор большинства (но не всех) особей с высоким значением функции приспособленности и, таким образом, в целом качество популяции особей на каждом шаге возрастает [3].

В области ГП объекты, из которых состоит популяция, представляют собой не строки фиксированной длины, кодирующие возможные решения проблемы, а программы, которые после своего выполнения являются кандидатами на решение проблемы. При этом для ГП программа представляется не просто как набор строк кода, а в виде дерева синтаксического разбора и соответствующей структуры данных. Фактически, программа в ГП уже имеет двухмерный характер.

Такая программа состоит из элементов функционального и терминального наборов. Функциональный набор представляет собой набор операторов, используемых в программе, и отображает внутренние вершины дерева синтаксического разбора. Терминальный набор соответствует терминаль-

ным вершинам дерева синтаксического разбора, например, переменным, константам или функциям без аргументов.

2. Парадигма эволюционной классификации

Системы классификации на основе эволюционных методов впервые были представлены в работе [1] в качестве примера обучающихся систем, которые не зависят от предметной области. Каждый классификатор в такой системе представляет собой правило вида «условие-действие» с некоторым набором связанных параметров. Условие классификатора является троичной строкой, элементы которой принадлежат множеству $\{0, 1, \#\}$. Элементы действия принадлежат множеству $\{0, 1\}$. Несколько строк условий в классификаторе могут объединяться логическим «И». Возможно также использование префикса «НЕТ» для указания того факта, что классификатор сработает только в том случае, когда ни одно из его условий не выполнено. Каждый классификатор имеет параметр силы, который является системной оценкой полезности классификатора.

Система классификации реализует три основных функции [2]:

- генерация классификаторов;
- оценка классификаторов;
- использование классификаторов при принятии решений при организации взаимодействия с внешней средой.

Соответственно, система классификации включает в себя три основных подсистемы:

- исполняющую подсистему;
- подсистему обучения;
- подсистему создания новых правил.

Исполнительная подсистема состоит из четырех частей:

- входного интерфейса, который генерирует сообщения в форме двоичных строк, характеризующих обнаруженные свойства внешней среды;
- списка сообщений, который сохраняет сообщения, генерируемые входным интерфейсом и классификаторами;
- списка классификаторов, который обрабатывает список сообщений и который может добавлять новые сообщения к этому списку;
- выходной интерфейс, который преобразует выходные сообщения из списка в некоторые действия во внешней среде;

Обобщенный алгоритм работы классификационной системы включает в себя следующие шаги:

1. Дополнение списка сообщениями, генерируемыми входной подсистемой.
2. Сопоставление каждого сообщения в списке с частью «условие» каждого классификатора, причем элемент # может соответствовать как 0, так и 1 во входном сообщении.
3. В том случае, если входное сообщение совпадет с частью «условие» классификатора, выполняется генерация выходного сообщения с помощью части «действие» классификатора. Классификаторы конкурируют друг с другом за право внести выходное сообщение в список сообщений, и только лишь часть сработавших классификаторов занесет в список свое выходное сообщение.
4. Замена содержимого списка сообщений в соответствии с п.3.
5. Использование выходного интерфейса для преобразования выходных сообщений в действия системы.
6. Назначение вознаграждения классификаторам в зависимости от успешности действий системы во внешней среде.
7. Переход к шагу 1.

Следует отметить, что система классификации обладает вычислительной полнотой. Она может выполнить любые вычислимые функции.

Подсистема обучения обеспечивает адаптивные свойства системы классификации. Адаптация выполняется путем корректировки параметра силы классификатора на основе вознаграждения, которое система получает в результате ее действий. В данном случае сила классификатора определяет собой награду, которую получит система в том случае, если она выберет этот классификатор из подмножества классификаторов с выполнившимися входными условиями, и позволит этому классификатору поместить в список выходное сообщение.

Выбор классификаторов, для которых выполнены входные сообщения, выполняется по соревновательному принципу. Каждый классификатор, для которого совпало условие, делает заявку на выполнение. Только классификаторы с наивысшим значением заявки могут выполняться. Значение заявки зависит от двух факторов: силы классификатора и его специфики. Специфика определяется числом нулей и единиц в условии классификатора. Специфику можно рассматривать как меру пригодности классификатора для определенного набора сообщений. Соответственно, величина заявки Z определяется как:

$$Z = k * S * P,$$

где k – универсальная константа для всех классификаторов, k лежит в интервале $(0,1)$; S – сила классификатора; P – специфика классификатора.

Следует отметить, что сила классификатора уменьшается на величину заявки в том случае, если классификатор получает право поместить выходное сообщение в список сообщений.

Подсистема создания новых правил позволяет генерировать новые правила на основе генетического алгоритма. В данной подсистеме сила классификатора используется в качестве значения приспособленности классификатора. В процессе создания новых правил выполняются следующие основные шаги:

- отбор классификаторов для создания новой популяции с вероятностью, пропорциональной силе каждого классификатора;
- применение генетических операторов – воспроизводства, кроссинговера и мутации к классификаторам из отобранного множества;
- отбор лучших из получившихся классификаторов в новую популяцию, обычно с вероятностью пропорциональной их силе.

3. Применение парадигмы эволюционной классификации при создании ФПО

Функциональное ПО информационных систем обладает рядом особенностей, обеспечивающих использование систем классификации при его создании и развитии. В общем случае данное программное обеспечение включает в себя элементы интерфейса с пользователем, элементы интерфейса с данными и модули, реализующие логику работы прикладных программ. Соответственно прикладные программы реализуют функции:

- взаимодействия с хранилищами (базами) данных;
- обработки информации в соответствии с логикой работы программ;
- предоставления информации пользователю;
- обработки реакции пользователя в соответствии с логикой работы прикладных программ.

В процессе взаимодействия с пользователем в прикладной программе осуществляется распознавание ситуации и формирование наиболее рациональных действий. Аналогичным образом в общем случае осуществляется и доступ к данным. Эволюция программного обеспечения постепенно приводит к более точному распознаванию ситуации и к более адекватным действиям в соответствующей ситуации. При этом оценка функционирования программы на каждом шаге развития ПО осуществляется пользователем, т.е. из внешней, по отношению к прикладной программе среды. В соответствии с результатами этой оценки выполняется соответствующая корректировка программного обеспечения.

Таким образом, парадигма систем классификации позволяет описать как процесс функционирования прикладных программ, так и процесс их развития.

Применение данной парадигмы при разработке и эволюции ПО связано с трудностями восприятия и модификации человеком алгоритмов, которые представлены в виде набора правил. Следовательно, данный подход предполагает автоматизацию разработки функционального ПО на основе эволюционных методов и алгоритмов.

Автоматизированное создание программ на основе эволюционных алгоритмов является темой исследования в области генетического программирования.

Традиционно исследования в области генетического программирования проводятся с использованием функциональных языков программирования (Lisp, Schema). В то же время при создании при-

кладного ПО информационных систем, которое работает в интерактивном режиме, взаимодействуя с пользователем и с внешней средой, используются алгоритмические языки программирования. Автоматизированное построение прикладных программ на основе таких языков легко реализуется в соответствии с подходом, принятым в системах классификации:

- программа представляется в виде набора правил (классификаторов) «условие-действие»;
- с каждым из правил- классификаторов связывается переменная – значение силы правила (данное значение вычисляется отдельной процедурой);
- условие каждого классификатора формируется в виде последовательности конструкций «если» соответствующего языка программирования;
- действие каждого классификатора формируется в виде последовательности процедур, написанных на соответствующем языке программирования;
- в процессе «выведения» рабочего варианта программы, а также в процессе ее эволюции правила конкурируют за возможность исполнения; арбитраж выполняется отдельной процедурой;
- величина силы правила после его выполнения снижается на величину заявки, последняя вычисляется в соответствии с подходом, принятым в системах эволюционной классификации.

Таким образом, применение парадигмы систем классификации при разработке ФПО позволяет, с одной стороны, создавать программное обеспечение на основе эволюционных методов, а с другой – выполнять обучение прикладных программ в процессе их эксплуатации.

Список литературы: 1. *Holland, J.* Adaption in Natural and Artificial Systems. University of Michigan Press, 1975. 2. Kovacs, Tim. Evolving Optimal Populations with XCS Classifier Systems. MSc Advanced Computer Science, September 1996, University of Birmingham, Birmingham U.K. 3. В.М.Левыкин Концепция создания распределенных информационных управляющих систем//АСУ и приборы автоматизации. 1998. Вып. 108. С. 2-41.

Поступила в редколлегию 28.08.2000