УДК 519.68

ОБ ЭФФЕКТИВНОСТИ КОЭВОЛЮЦИОННОГО ПОДХОДА В ПРАКТИЧЕСКИХ ЗАДАЧАХ ОПТИМИЗАЦИИ

Г.В. Процыков, Е.С. Семенкин, К.А. Токмин*

Предлагаются самонастраивающиеся коэволюционные алгоритмы условной и безусловной оптимизации, исследуется их эффективность на тестовых и практических задачах. Продемонстрирована работоспособность коэволюционного подхода и его преимущество над стандартными генетическими алгоритмами.

Практические задачи оптимизации сложных систем обладают свойствами, существенно затрудняющими их решение: дискретные или смешанные переменные, алгоритмически заданные целевые функции и/или ограничения, отсутствие удобных для оптимизации свойств или, по крайней мере, отсутствие информации о таких свойствах, и т.д. Применение в таких задачах классических методов оптимизации является зачастую невозможным, поэтому обычно применяются алгоритмы прямого поиска, в частности, так называемые эволюционные алгоритмы [1, 2].

Практическое применение эволюционных алгоритмов встречается с определенными трудностями в силу специфики данного подхода [2, 3]. Основной трудностью является выбор оптимальных настроек алгоритмов, требующий многократного решения одной и той же задачи, что на практике неприемлемо.

В данной работе исследуется коэволюционный подход, позволяющий автоматизировать этап настройки эволюционных алгоритмов на задачу в ходе ее однократного решения.

Стандартный генетический алгоритм

Генетические алгоритмы (ГА) относятся к так называемым эволюционным методам поиска, моделирующим процессы естественной эволюции. Генетические и эволюционные алгоритмы являются стохастическими алгоритмами глобального поиска и используются для решения различных задач оптимизации, в том числе и для задач оптимизации сложных систем.

Каждый индивид в ГА представляет собой решение, закодированное определенным образом (обычно в виде бинарной строки, что превращает ГА в алгоритмы псевдобулевой оптимизации) и называемое также хромосомой. Совокупность таких индивидов в фиксированный момент времени составляет популяцию. Этап эволюционного развития индивидов определяется номером поколения, в котором в данный момент времени находятся индивиды. Индивиды последующей популяции (*потомки*) получаются путем наследования признаков своих родителей и случайного изменения их генотипа, известного в природе как мутация. Каждый индивид в ГА характеризуется некоторым числом (*пригодностью*), обозначающим меру его адаптации к окружающей среде.

Процессы эволюции, происходящие в ГА, основаны на следующем принципе: «Каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наилучшим образом приспособиться к окружающей среде».

Основными настройками ГА, определяющими его эффективность на решаемой задаче, являются так называемые "генетические" операторы. Кроме того, очень важен размер выделяемого вычислительного ресурса, который чаще всего определяется как произведение размера популяции на число поколений (т.е. разрешенное количество вычислений функций задачи). При фиксированном размере ресурса эффективность ГА определяется выбором типов операторов. В стандартном генетическом алгоритме (СГА) обычно используются следующие операторы: селекция (пропорциональная, ранговая, элитарная, турнирная), рекомбинация (одноточечная, двухточечная, равномерная) и мутация (низкая, средняя, высокая). Кроме того, могут использоваться различные способы управления популяциями (инбридинг, аутбридинг, панмиксия и др.).

Показателями эффективности ГА служат надежность и скорость. В связи с тем, что ГА является стохастической процедурой, оценка его эффективности осуществляется усреднением по многократным прогонам. Тогда надежность ГА на данной задаче – это отношение количества прогонов, в которых был найден оптимум, к общему количеству тестовых прогонов. Скоростью называется среднее количество вычислений целевой функции до первого обнаружения экстремума (при одинаковых размерах популяции – средний номер поколения, на котором был впервые обнаружен экстремум). У наилучшего ГА должна быть наибольшая надежность, при равных показателях надежности – наибольшая скорость (наименьшее количество вычислений до обнаружения экстремума).

Различные комбинации "генетических" операторов могут дать несколько сот различных СГА, эффективность которых в конкретной задаче может значительно отличаться.

При решении задач условной оптимизации применяются различные методы учета ограничений [4]: статические, динамические и адаптивные штрафные функции, "лечение" недопустимых индивидов, барьерные функции, специализированные генетические операторы и др. Все эти методы обладают собственными параметрами, также требующими настройки. Это значит, что и в этом случае понадобится большое количество экспериментов для выбора эффективного метода учета ограничений в конкретной задаче.

Одним из наиболее перспективных подходов для преодоления трудностей с выбором эффективных настроек СГА, возникающих при решении практических задач, является использование идеи коэволюции [3], состоящей в том, что в ГА используется несколько взаимодействующих популяций. Правильное формирование и организация сотрудничества популяций позволяют получить достаточно эффективные алгоритмы, настройка которых намного облегчается за счет самоадаптации и исключения большого числа настраиваемых параметров.

Коэволюционный подход в задачах безусловной оптимизации

Эффективность работы генетических алгоритмов определяется выбором "генетических" операторов, а также ряда других параметров. Численные эксперименты показали, что даже для простых функций нельзя говорить о преимуществе того или иного вида операторов, поскольку достаточно трудно априорно определить, какой вид оператора более подходит для конкретной задачи [5].

В данной работе рассмотрено два типа самоадаптирующихся генетических алгоритмов.

Первый тип – коэволюционный генетический алгоритм – это несколько параллельно действующих простых генетических алгоритмов с разными параметрами генетических операторов [3, 5]. Дальше простой генетический алгоритм в составе коэволюционного будем называть подпопуляцией.

Эти подпопуляции могут сотрудничать или конкурировать. Например, одна подпопуляция выбирает высокую степень мутации и двухточечное скрещивание, а другая – низкую мутацию и одноточечное скрещивание. Возможно также варьирование типа селекции.

Для подпопуляций необходимо ввести свою функцию пригодности. С помощью этой функции можно будет определять лучшую подпопуляцию и предоставлять ей больше возможностей для воспроизводства. Пусть T – интервал адаптации, $b_i(k)$ равно единице, если i-я подпопуляция в момент k содержит наилучшего индивида, k = 0 означает текущую ситуацию, k = 1 – предыдущую, и т.д. Тогда качество подпопуляции можно вычислить следующим образом:

$$q_i = \sum_{k=0}^{\mathrm{T}-1} \frac{\mathrm{T}-k}{k} \cdot b_i(t)\,.$$

Изменение размеров группы можно выполнять, например, сокращением каждой проигравшей подпопуляции (не достигшей минимального гарантированного размера) на 10-12 % и увеличением победившей подпопуляции на число, равное сумме потерь проигравших. Таким образом, общее число индивидов остается неизменным. Необходимо также предусмотреть миграцию лучшего индивида в другие подпопуляции через определенное число поколений.

В работе реализовано два коэволюционных генетических алгоритма. Первый основан на различных параметрах генетических операторов (селекция – пропорциональная, ранговая, турнирная, элитарная; мутация – высокая, средняя, низкая; рекомбинация (скрещивание) – одноточечная, двухточечная, равномерная;). Второй – на различных способах формирования нового поколения (элитизм, отбор с вытеснением, только потомки, случайным образом) и выбора родительской пары (турнирный, «панмиксия», инбридинг, аутбридинг) [5].

Описанный подход позволяет автоматически выбирать лучшую подпопуляцию, т.е. лучший набор генетических операторов из имеющихся, и включать ее в необходимый момент.

Второй тип – генетический алгоритм с измененными хромосомами.

Основная идея – осуществлять адаптацию не на уровне подпопуляций, а на уровне отдельных индивидов. Для этого предлагается в хромосому добавить два параметра, которые отвечают за скрещивание и селекцию данного индивида. В этом случае скрещивание происходит следующим образом: выбираем двух родителей, вид скрещивания и дополнительные параметры хромосомы, которые передадутся по наследству, определяем по родителю с наибольшей пригодностью. Индивиды с определённым видом селекции образуют группу, из которой выбираются индивиды в следующее поколение, с соответствующим видом селекции. Первоначально дополнительные параметры в хромосоме кодируются случайно. На первом этапе работы алгоритма в поколении представлены все виды операторов. В данном алгоритме в рамках одной популяции коэволюционируют различные типы индивидов.

Кодирование мутации представляется нецелесообразным, так как со временем индивиды с низкой мутацией захватывают всю популяцию, что приводит к снижению глобальных свойств алгоритма. Поэтому все индивиды имеют среднюю мутацию, в среднем 1 ген на хромосому.

После прохождения нескольких поколений соотношение индивидов по видам скрещивания и селекции изменяется, следовательно, происходит адаптация к определённой задаче.

Для генетического алгоритма с измененными хромосомами также созданы две реализации, одна – с кодированием типа селекции и скрещивания, вторая – с кодированием способа формирования нового поколения и способа выбора родительской пары.

Тестовые задачи безусловной оптимизации

Для оценки эффективности предложенных подходов использовались стандартные задачи, обычно применяемые для тестирования эволюционных алгоритмов. Эти задачи отражают набор свойств задач оптимизации, встречающихся на практике, в том числе обладают "неудобными", с точки зрения алгоритмов оптимизации, свойствами (многоэкстремальность, узкие зоны притяжения локальных экстремумов, незначительное различие значений функции в точках локальных экстремумов и т.п.).

А. Обычные тестовые задачи

Задача № 1

Квадратичная функция:

$$f(x,y) = (x-c_1)^2 + (y-c_2)^2 \to \min; x,y \in [-10,10].$$

Задача № 2

Функция Растригина:

$$f(x,y) = 20 + x^2 + y^2 - 10\cdot\cos(2\cdot\pi\cdot x) - 10\cdot\sin(2\cdot\pi\cdot y) \to \min\,,\ x,y \in [-5,5].$$

Задача №3

Функция «Волна-1»:

$$f(x) = e^{-x^2} + 0.01\cdot\cos(100\cdot x) \to \max\,,\ \ x,y \in [-2,2].$$

Задача № 4

Функция "Волна-2" (производная от функции 3):

$$f(x) = -2e^{-x^2} - \sin(100 \cdot x) \to \max\,,\ \ x, y \in [-2{,}2].$$

Задача № 5

Функция Де Йонга:

$$\frac{100}{100 \cdot (x-y) + (1-x)^2} \to \max,\ x{,}y \in [1{,}28,\ 1{,}28].$$

Задача № 6

Функция Шекеля:

$$\frac{1}{f(\mathbf{X})} = \frac{1}{K} + \sum_{j=1}^{25} \frac{1}{c_j + \sum_{i=1}^{2} (x_i - a_{ij})^6}$$

$$\left\| a_{ij} \right\| = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32 \end{pmatrix},$$

K=500, $f(a_{1j}, a_{2j}) \approx c_j = j$, $-65.536 < x_i < 65.536$, $i = 1{,}2$, $\mathbf{X}^* = (-32,-32)$, $f(\mathbf{X}^*) \approx 1$.

Б. Практические задачи

№ 7. Задача выбора эффективного варианта технологического контура системы управления космическим аппаратом.

№ 8. Задача выбора эффективного варианта целевого контура системы управления космическим аппаратом.

В этих задачах моделирование функционирования системы выполняется с помощью марковских процессов, а вычисление показателей эффективности (коэффициенты готовности, среднее время реакции системы) осуществляется по финальным вероятностям, определяемым из системы уравнений Колмогорова-Чэпмена [3]. Эффективность вариантов определяется составом аппаратно-программного комплекса, реализующего функции контуров, что приводит к необходимости решать задачи безусловной оптимизации с алгоритмически заданными функциями 18 и 24 булевых переменных.

Исследование эффективности коэволюционных алгоритмов безусловной оптимизации

Приведенные задачи были использованы для исследования эффективности коэволюционных алгоритмов безусловной оптимизации. Надежность каждого алгоритма оценивалась по 50 прогонам. Количество используемых ресурсов (число вычислений целевых функций) было одинаковым для всех алгоритмов.

Сначала была определена надежность 27 стандартных генетических алгоритмов (СГА), отличающихся настройками генетических операторов, и 48 СГА, отличающихся стратегиями формирования нового поколения и родительской пары. Надежность алгоритмов значительно различалась от задачи к задаче. Для каждой задачи был определен наилучший СГА. Кроме того, надежность СГА усреднялась – была определена средняя надежность всех СГА на каждой задаче и средняя надежность лучшего СГА на каждой задаче. Основная идея этих усреднений состоит в следующем: при решении реальных задач пользователь не может многократно решать свою задачу, поэтому он выберет один из СГА, в худшем случае – наугад. Поэтому самонастраивающийся ГА должен, как минимум, превзойти по эффективности этот виртуальный "средний" алгоритм, выбираемый наугад. В дальнейшем была определена надежность самонастраивающихся ГА - коэволюционного на основе операторов, коэволюционного на основе стратегий и коэволюционного с измененными хромосомами на каждой задаче и в среднем на всех задачах. Полученные результаты сведены в табл. 1.

Таблица 1

№	Виды алгоритмов	Тестовые функции								
		1	2	3	4	5	6	7	8	ср.
1	Лучший СГА на основе операторов	94	94	96	90	83	100	83	79	82
2	Среднее по СГА на основе операторов	80	54	84	74	56	76	38	48	73
3	Лучший СГА на основе стратегий	100	98	96	94	64	90	84	100	91
4	Среднее по СГА на основе стратегий	82	71	84	65	37	76	82	99	75
5	Коэволюционный ГА на основе операторов	90	66	98	78	74	94	94	90	86
6	ГА с измененными хромосомами	100	86	80	90	70	84	100	98	89
7	Коэволюционный ГА на основе стратегий	98	92	98	98	60	98	96	100	93

Анализ результатов показывает, что средние показатели эффективности всех коэволюционных алгоритмов значительно превосходят средние показатели всех СГА и лучшего СГА на основе операторов. Коэволюционный ГА на основе стратегий в среднем превосходит все другие алгоритмы, в том числе и лучшие СГА, отобранные после многократного решения задач.

Таким образом, работоспособность коэволюционного метода получила подтверждение: средняя эффективность коэволюционных алгоритмов, не требующих настройки, превосходит среднюю эффективность стандартных генетических алгоритмов даже после значительных затрат на их настройку.

Коэволюционный алгоритм условной оптимизации

Пусть решается задача условной оптимизации в следующем виде:

$$\min f(X),\ X \in B^n,\ g_i(X) > 0,\ i = 1,2,...,m.$$

Тогда функция Лагранжа имеет вид

$$L(X,Y) = f(X) + \sum_{i=1}^{m} g_i(X) \cdot y_i\ ,$$

где y_i - множители Лагранжа.

При этом в точке, являющейся решением, достигается минимум функции Лагранжа по переменным задачи и максимум по коэффициентам Лагранжа. В теории оптимизации известен метод обобщенного Лагранжиана, когда условный оптимум ищется как седловая точка функции Лагранжа. Решение задачи условной оптимизации сводится при этом к поочередному решению задач безусловной минимизации функции Лагранжа по объектным переменным (при фиксированных коэффициентах Лагранжа) и максимизации функции Лагранжа по коэффициентам Лагранжа (при фиксированных объектных переменных).

Этот подход может быть практически напрямую применён в генетическом алгоритме, если хромосома будет представлять собой объект, составленный из объектных переменных и коэффициентов Лагранжа. Однако возможен и другой подход, состоящий в применении коэволюционного алгоритма, где две популяции (объектных переменных и коэффициентов Лагранжа) эволюционируют независимо друг от друга, используя каждая свою собственную процедуру генетического алгоритма [2]. Размеры и представление популяций, а также параметры генетического алгоритма устанавливаются независимо, а согласование процессов эволюции выполняется через оценивание функции пригодности. Основой для вычисления функции пригодности служит функция L(X,Y), где X берётся из популяции A объектных переменных, а Y - из популяции B коэффициентов Лагранжа. Поэтому функция пригодности индивида из популяции A зависит от популяции B, и наоборот.

Общая структура коэволюционного ГА:

начало
инициализировать популяцию A
инициализировать популяцию B
for k=1, 2,..., c_{max} do
for i=1, 2,..., a_{max} do
сгенерировать новую популяцию A
оценить A
for j= 1, 2,..., b_{max}
сгенерировать новую популяцию B
оценить B

Выполнять, пока не выполнится условие останова.

Процесс решения задачи коэволюционным алгоритмом состоит в поочередной работе генетического алгоритма с обеими популяциями. Сначала генетический алгоритм работает с популяцией A, давая ей эволюционировать определенное пользователем количество поколений a_{max}. При этом функция Лагранжа минимизируется, а индивиды популяции B – фиксированы. Затем генетический алгоритм работает с популяцией B в течение b_{max} поколений. При этом функция Лагранжа максимизируется, а индивиды популяции A – фиксированы. Этот цикл повторяется c_{max} раз. Таким образом a_{max}, b_{max} и c_{max} являются параметрами коэволюционного алгоритма, от которых зависит его эффективность, и которые должны быть настроены в процессе работы.

Исследование эффективности коэволюционного алгоритма условной оптимизации

Работоспособность коэволюционного алгоритма была проверена на множестве тестовых задач линейного и нелинейного программирования с вещественными и целочисленными переменными, параметры которых были сгенерированы случайным образом. Алгоритму разрешалось сделать ограниченное количество вычислений функций в 0.125% точках пространства оптимизации. Решение задач условной оптимизации с целыми и вещественными переменными с использованием данного подхода не представляет больших сложностей, так как сначала выполняется бинаризация переменных (процедура стандартная для генетического алгоритма), а затем решается задача условной псевдобулевой оптимизации, которая с помощью метода обобщённого Лагранжиана сводится к последовательности задач безусловной оптимизации.

Средняя надёжность коэволюционного генетического алгоритма с настроенными параметрами на тестовых задачах приведена в табл. 2.

Таблица 2

Тип переменных	Тип задачи	Средняя надежность коэволюционного алгоритма
Вещественные	Линейного программирования	0,92
	Нелинейного программирования	0,88
Целочисленные	Линейного программирования	0,86
	Нелинейного программирования	0,80

После тестирования работоспособность алгоритма была проверена на практической задаче псевдобулевой оптимизации – нахождения оптимального набора кредитных заявок банка при согласованном виде структуры активов–пассивов [6].

Для формализованной записи критерия получения максимальной доходности от проводимых банком в i-м временном интервале кредитных операций при соблюдении ограничений на общую сумму и риск невозврата вводятся следующие обозначения:

F_i - сумма свободных пассивов, которыми располагает банк в i-м временном интервале;

k_{ij} - сумма кредита, запрашиваемая j-м заемщиком с погашением долга в i-м временном интервале, $i = 1, \ldots, I$, $j = 1, \ldots, J$;

t_{ij} - период размещения средств в k_{ij}-й кредит;

x_{ij} - булева переменная, принимающая значения: 1, если кредит k_{ij} выдается, и 0, если заявка на получение кредита отклоняется банком;

d_{ij} - проценты за пользование k_{ij}-м кредитом (предполагается, что d_{ij} выплачиваются единовременно с возвратом самого кредита);

P_{ij} - вероятность невыполнения заемщиком обязательств по возврату кредита и процентов по нему ($k_{ij}+k_{ij}d_{ij}$).

В постановке задачи предполагается два варианта обслуживания долга заемщиком: 100 % возврат суммы кредита и процентов по нему в установленный срок, либо полное отсутствие платежей в погашение кредита и процентов по нему.

Ожидаемые проценты от комбинации кредитных заявок будут определяться по формуле:

$$E(x_{ij}) = \sum_{j \in J} (k_{ij} + k_{ij} d_{ij} t_{ij}) \cdot x_{ij} \ .$$

Таким образом, целевая функция задачи максимизации дохода может быть представлена в следующем виде:

$$E(x_{ij}) \to \max_{x_{ij}},$$

при ограничениях:

- на объем выдаваемых кредитов в i-м временном интервале:

$$\sum_{j \in J} k_{ij} \cdot x_{ij} \le F_i;$$

- на рискованность рассматриваемой кредитной заявки будет иметь вид:

$$R(x_{ij}) = \sum_{j \in J} P_{ij} \cdot x_{ij} \le R_i.$$

Данная задача в математической формулировке представляет собой задачу условной псевдобулевой оптимизации, а точнее – задачу о ранце.

Исследование эффективности коэволюционного ГА проводилось сначала для случая, когда количество кредитных заявок было равно 25. В этом случае истинный условный оптимум может быть определен полным перебором и использован для контроля работы ГА. Количество разрешенных вычислений целевой функции и ограничений было выбрано в размере 0,3 % мощности допустимого множества (2^{25} точек). Усреднение проводилось по 50 прогонам алгоритма при каждом выборе параметров коэволюционного алгоритма.

В результате экспериментов установлено, что надежность коэволюционного ГА составила 100 % при оптимальном выборе установок – турнирная селекция, слабая мутация, равномерная рекомбинация, a_{max}=10,

b_{max}=40, c_{max}=100. При этом оптимальное решение всегда определялось после просмотра 0,08 % поискового пространства.

С помощью коэволюционного алгоритма была решена также исходная задача формирования кредитного портфеля банка, когда количество кредитных заявок составляло 52 [6] (мощность допустимого множества 2^{52} точек). Количество разрешенных вычислений целевой функции было выбрано в размере $8 \cdot 10^{-9}$ % мощности допустимого множества. Истинный оптимум в данном случае неизвестен и не может быть установлен полным перебором, поэтому оценка эффективности проводилась по наилучшему найденному портфелю и сравнивалась с результатами, полученными другими авторами [6, 7]. При этом было установлено, что решение, полученное с помощью коэволюционного ГА, во многих случаях было лучше и никогда не было хуже решений, полученных другими методами. Наилучшее известное решение (f = 199634435,34 руб.) было получено в 30 % прогонов. Приведенные другими авторами [б, 7] решения были хуже, чем найденное коэволюционным ГА.

Таким образом, работоспособность коэволюционного алгоритма условной оптимизации также получает подтверждение.

Очевидным направлением развития рассмотренных в данной статье подходов является комбинирование подходов в единый комплексный коэволюционный алгоритм условной оптимизации. Предполагаемый результат - самонастраивающийся алгоритм условной оптимизации с алгоритмически заданными функциями смешанных переменных, не требующий от применяющего его пользователя глубоких знаний в теории оптимизации вообще и эволюционных алгоритмов – в частности.

СПИСОК ЛИТЕРАТУРЫ

1. Goldberg D.E. Genetic algorithms in search, optimization and machine learning. – Reading, MA: Addison-Wesley, 1989.
2. Goodman E. et al (Eds). Evolutionary computation and its applications // Proc. of the Int. Conference. - Moscow: IHPCS of RAS, 1996. –350 pp.
3. Семенкин Е.С. Метод обобщенного адаптивного поиска для оптимизации сложных систем / Е.С. Семенкин, В.А. Лебедев // Конверсия в машиностроении – Conversion in machine building of Russia. - 2002. - № 2.
4. Michalewicz Z. Genetic algorithms, numerical optimization and constraints // Proc. of the Sixth Int. Conf. on Genetic Algorithms and their Applications, Pittsburgh, PA, 1995.
5. Жукова М.Н. Коэволюционный алгоритм решения сложных задач оптимизации: Дисс. … канд. техн. наук / М.Н. Жукова. – Красноярск: СибГАУ, 2004. – 124 с.
6. Пуртиков В.А. Оптимизация управления формированием кредитного портфеля банка: Дисс. … канд. техн. наук / В.А. Пуртиков. - Красноярск: САА, 2001. - 148 с.
7. Хоролич Г.Б. Формализация и решение генетическими алгоритмами задачи формирования кредитного портфеля банка / Г.Б. Хоролич // Интеллектуальные технологии и адаптация. – Красноярск: НИИ СУВПТ, 1999. – С. 42 – 45.

ON EFFECTIVENESS OF COEVOLUTIONARY APPROACH IN REAL WORLD OPTIMIZATION PROBLEMS

G.V. Protsykov, E.S. Semenkin, K.A. Tokmin

Self-adapted coevolutionary algorithms for constrained and non-constrained optimization are suggested, their effectiveness on test functions and real world problems is investigated. The workability of coevolutionary approach and its advantage upon conventional genetic algorithms has been demonstrated.