

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО АТОМНОЙ ЭНЕРГИИ
РОССИЙСКАЯ АКАДЕМИЯ НАУК
РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ
МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)
ИНСТИТУТ ОПТИКО-НЕЙРОННЫХ ТЕХНОЛОГИЙ РАН

НАУЧНАЯ СЕССИЯ МИФИ–2007

НЕЙРОИНФОРМАТИКА–2007

**IX ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

**ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ
Часть 2**

По материалам Школы-семинара
«Современные проблемы нейронинформатики»

Москва 2007

УДК 001(06)+004.032.26 (06) Нейронные сети
ББК 72я5+32.818я5
М82

НАУЧНАЯ СЕССИЯ МИФИ–2007. IX ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2007»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. Часть 2. – М.: МИФИ, 2007. – 148 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 24–26 января 2007 года в МИФИ в рамках IX Всероссийской конференции «Нейроинформатика–2007».

Материалы лекций связаны с рядом проблем, актуальных для современного этапа развития нейроинформатики, включая ее взаимодействие с другими научно-техническими областями.

Ответственный редактор
Ю. В. Тюменцев, кандидат технических наук

ISBN 5–7262–0708–4 © *Московский инженерно-физический институт
(государственный университет), 2007*

Содержание

Ю. Р. Цой. Введение в нейроэволюционный подход: основные концепции и приложения	43
Введение	44
Эволюционные алгоритмы	44
Адаптация в ЭА	49
Проблема формирования обучающего множества данных для ИНС	52
Нейроэволюционный подход	53
Эволюционная настройка весов связей ИНС	53
Эволюционная настройка структуры ИНС	56
Одновременная эволюционная настройка весов связей и структуры ИНС	58
Приложения НЭ-подхода	60
НЭ алгоритмы в мире	65
Заключение	67
Благодарности	68
Литература	68

Ю. Р. ЦОЙ

Томский политехнический университет,
г. Томск

E-mail: qai@mail.ru

**ВВЕДЕНИЕ В НЕЙРОЭВОЛЮЦИОННЫЙ ПОДХОД: ОСНОВНЫЕ
КОНЦЕПЦИИ И ПРИЛОЖЕНИЯ**

Аннотация

В лекции рассматривается нейроэволюционный (НЭ) подход — направление исследований, находящееся на стыке искусственных нейронных сетей (ИНС) и эволюционных вычислений. Выделены преимущества и недостатки НЭ-подхода при решении задач эволюционной настройки весов и структуры ИНС. Рассматривается применение НЭ-алгоритмов к решению задач адаптивного управления, адаптивного поведения, многоагентных систем, эволюционной робототехники, поиска игровых стратегий и компьютерного творчества.

Yu. R. TSOY

Tomsk Polytechnic University,
Tomsk

E-mail: qai@mail.ru

**AN INTRODUCTION TO NEURO-EVOLUTIONARY APPROACH:
MAIN CONCEPTS AND APPLICATIONS**

Abstract

This lecture concerns neuroevolutionary (NE) approach — a research domain inspired by both artificial neural networks (ANNs) and evolutionary computations. Advantages and disadvantages of NE approach for ANN design and training are lined out. Application of the NE algorithms for problems of adaptive control, adaptive behavior, multi-agent systems, evolutionary robotics, game strategies search and computer creativity is described.

Введение

Выбор топологии и настройка весов связей искусственной нейронной сети (ИНС) [7, 11] являются одними из важнейших этапов при использовании нейросетевых технологий для решения практических задач. От этих этапов напрямую зависит качество (адекватность) полученной нейросетевой модели, системы управления и т. д. В данной лекции в общем виде рассматривается нейроэволюционный (НЭ) подход, как один из возможных вариантов решения проблем настройки структуры и обучения ИНС.

Статья организована следующим образом. В первом разделе обзорно описываются эволюционные алгоритмы и методы адаптации эволюционного поиска. Во втором разделе кратко формулируется проблема формирования обучающего множества данных, необходимого для обучения ИНС с использованием градиентных алгоритмов. В третьем разделе в общем виде описывается нейроэволюционный подход и рассматриваются задачи эволюционного поиска структуры и весов связей ИНС. Некоторые приложения НЭ-алгоритмов для решения трудноформализуемых задач описаны в четвертом разделе. Пятый раздел посвящен краткому освещению исследований НЭ-подхода в мире и России.

Эволюционные алгоритмы

Рассмотрим следующую «математическую» игру, в которой есть ведущий и участники.

Ведущий загадывает произвольную функцию F одного переменного, непрерывную на некотором промежутке (например, $[-10; 10]$) и имеющую на нем минимум. Игра состоит из этапов. На каждом этапе все участники по очереди называют по одному значению аргумента, а ведущий вслух объявляет соответствующие значения функции F , после чего участники на основе полученных ответов «измышляют» новые значения переменной. Победителем считается тот участник, который первым сумеет найти значение переменной, при котором F принимает минимальное значение (с заданной точностью).

На самом первом этапе значения переменной будут, скорее всего, названы участниками случайно. Но уже на втором этапе, зная предыдущие ответы ведущего, участники будут стараться подобрать значения аргумента

так, чтобы они, например, были как можно ближе к предыдущему «лучшему» значению. По мере прохождения все большего числа этапов, участники будут накапливать информацию о функции F и по прошествии достаточно большого промежутка времени можно предположить, что значение аргумента, доставляющего минимум функции F , будет отгадано и при этом практически «вслепую», методом проб и ошибок, без какой-либо информации о свойствах функции, кроме ее значений в определенных точках.

Данный «игрушечный» пример необходим для того, чтобы показать саму суть идеи угадывания с использованием множества (*популяции*) решений-кандидатов, которую можно применить и для решения более сложных задач. При этом можно провести аналогии между процессом игры и природной эволюцией, выявив аналогии принципов наследственности (используется информация о значениях функции для рассмотренных на предыдущих этапах значений аргумента), изменчивости (ясно, что на следующем этапе целесообразно использовать не встречавшиеся ранее значения переменной) и естественного отбора (значения переменной, в которых значение F сравнительно велико не стоит, по всей видимости, рассматривать в дальнейшем). И здесь, очевидно, представляет интерес разработка определенной общей стратегии «угадывания», которую можно формализовать и запрограммировать, и которая могла бы использоваться для генерации новых вариантов решений, на основании информации о значениях функции F .

Группу методов и алгоритмов, опирающихся на механизмы, аналогичные описанным выше, называют *эволюционными алгоритмами* (*evolutionary algorithms*) (ЭА), а соответствующую область исследований и разработок — *эволюционными вычислениями* (*evolutionary computation*) [3,4,22,26]. Каждое генерируемое решение-кандидат представляют в виде вектора параметров, называемого *хромосомой* или *особью*. Множество особей, сгенерированных на одной этапе, именуют популяцией. Подмножество особей, используемое для генерации новых особей, называют *родительской подпопуляцией*, а множество сгенерированных новых особей именуют, соответственно, *популяцией потомков*. Операции, используемые для создания популяции потомков, называют *генетическими операторами*. Качество (*приспособленность*) каждого решения-кандидата определяется в результате оценивания этого решения с помощью функции F , которую называют *функцией приспособленности*. Общая схема эволюционного поиска, описывающая большинство существующих моделей ЭА, и соответствующие каждому этапу эволюционные принципы представлены на рис. 1. Также

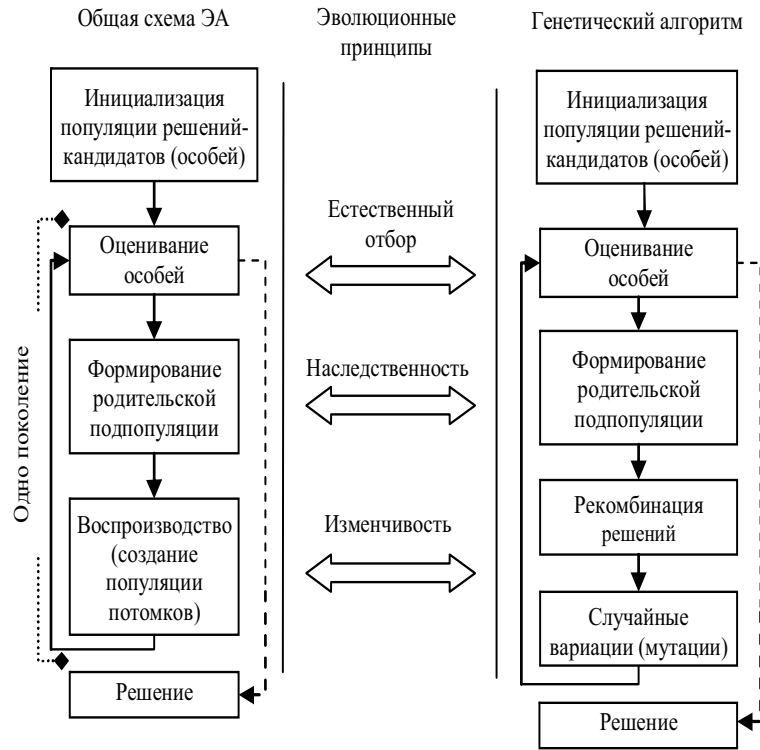


Рис. 1. Общая схема эволюционного поиска, пример схемы генетического алгоритма и соответствующие эволюционные принципы

на рис. 1 для сравнения показана общая схема *генетического алгоритма* (ГА), одного из самых распространенных видов ЭА. В ГА для генерации потомков в качестве основных используются два оператора: *кроссинговер* и *мутация*. Первый оператор необходим для рекомбинации параметров фиксированного числа родительских особей (как правило, двух), выбираемых из родительской подпопуляции, при этом создаются особи-потомки (обычно 1 или 2). Второй оператор используется для внесения случайных изменений (вариаций) в существующие хромосомы для более интенсивного поиска. Таким образом, популяция потомков наследует некоторые свойства родительской подпопуляции, но отличается от них.

Сама идея применения эволюционных принципов для машинного обучения присутствует также и в известном труде А. Тьюринга, посвященном проблемам создания «мыслящих» машин [89]. Использование одновременно множества решений делает возможным параллельный поиск и часто позволяет компенсировать возможную неопределенность при вычислении приспособленностей [20], а также исправить, «починить» (repair) ненужные изменения хромосом [25].

Одним из преимуществ использования ЭА при решении задач оптимизации является их нетребовательность к «знанию» предметной области. Эффективность и качество работы ЭА во многом определяется следующими составляющими и их параметрами:

1. *Генетическое кодирование*. Определяет исследуемое пространство поиска и отношение между генотипом (хромосомами) и фенотипом (соответствующими решениями) («один-к-одному», «один-к-многим», «многие-к-одному»)
2. *Генетические операторы*. Используются для рекомбинации и вариации представленных в популяции хромосом и соответствующих им решений. Сюда же относятся операторы селекции и формирования нового поколения.
3. *Функция приспособленности*. Необходима для определения качества получаемых решений и определяет характеристики ландшафта пространства поиска (количество локальных экстремумов, наличие плато, ограничения).
4. *Общая схема эволюционного поиска*. Определяет последовательность выполнения различных операций эволюционного поиска.

Настройка параметров ЭА существенно влияет на его эффективность и качество получаемых решений. Различные параметры влияют на разные аспекты эволюционного поиска, среди которых выделим два наиболее общих [30]:

1. *Исследование пространства поиска (exploration)*.
2. *Использование найденных «хороших» решений (exploitation)*.

Первый аспект отвечает за способности ЭА к эффективному поиску решения и характеризует способности алгоритма избегать локальных экстремумов. Второй аспект важен для постепенного улучшения имеющихся результатов от поколения к поколению на основе уже найденных «промежуточных» решений.

Пренебрежение исследовательскими способностями приводит к существенному увеличению времени работы ЭА и ухудшению результатов из-за «застревания» алгоритма в локальных экстремумах. В итоге становится возможной *преждевременная сходимост* генетического алгоритма (также говорят о *вырождении популяции*), когда решение еще не найдено, но в популяции практически все особи становятся одинаковыми и долгое время (порядка нескольких десятков и сотен поколений) не наблюдается улучшения приспособленности. Игнорирование найденных решений может привести к тому, что работа ГА будет напоминать случайный поиск, что также отрицательно сказывается на эффективности поиска и качестве получаемых решений.

Основная цель в настройке параметров ГА и, одновременно, необходимое условие для стабильного получения хороших результатов — это достижение баланса между исследованием пространства поиска и использованием найденных решений [3, 30]. Взаимосвязь между параметрами генетического алгоритма, а также их влияние на эволюционный процесс носит сложный характер. На рис. 2 схематично изображено влияние изменения некоторых параметров ГА на характеристики эволюционного поиска.

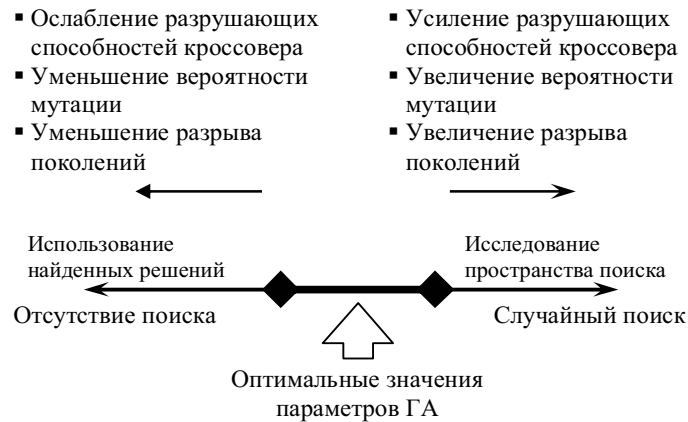


Рис. 2. Влияние некоторых параметров ГА на характеристики эволюционного поиска

Адаптация в ЭА

Одной из отличительных особенностей ЭА являются их адаптивные способности, что дает возможность реализовать подстройку параметров ЭА в процессе его работы для повышения эффективности ЭА и качества результатов. Целесообразно выделить следующие варианты настройки значений параметров ЭА [48]:

1. Детерминированная настройка параметров.
2. Адаптивная настройка параметров.
3. Адаптивная самонастройка параметров.

В случае детерминированной настройки параметров ЭА [12,46,65] задается процедура, управляющая изменением значений параметров. При этом состояние процесса эволюционного поиска не учитывается, а принимается во внимание только количество поколений, прошедших с момента запуска ЭА. Например, в [46] предложена следующая формула для подстройки вероятности битовой мутации P_m в зависимости от номера поколения t :

$$P_m(t) = \sqrt{\frac{\alpha}{\beta} \frac{\exp(-\gamma t/2)}{N\sqrt{L}}},$$

где N — размер популяции, L — длина хромосомы в битах, α , β и γ — константы.

При использовании варианта с адаптивной настройкой параметров ЭА [3, 33, 53, 60, 96] также определяется процедура изменения значений параметров. Но, в отличие от предыдущего варианта, параметры изменяются в зависимости от характеристик эволюционного поиска. В данном варианте становится доступен уровень адаптации, учитывающий приспособленность, генотипические и фенотипические характеристики каждой отдельной особи. Например, уменьшение вероятности мутации для более приспособленных особей, и увеличение — для менее приспособленных. Из ранних работ отметим известное эмпирическое правило «1/5» Рехенберга [74], где предполагается, что доля успешных мутаций составляет 1/5 часть от общего числа мутаций и вероятность мутации адаптивно настраивается, чтобы соответствовать этому правилу. В работе [21] предлагается алгоритм подстройки размера популяции, основанный на идее из [19], где вводится дополнительный параметр, время жизни Λ_i хромосомы, вычисляемый в зависимости от ее приспособленности F_i [21] (для задачи

минимизации целевой функции F):

$$\Lambda_i = \begin{cases} \Lambda_{min} + \mu \frac{F_{worst}(t) - F_i}{F_{worst}(t) - F_{avg}(t)}, & \text{если } F_i \geq F_{avg}(t), \\ \frac{1}{2}(\Lambda_{max} + \Lambda_{min}) + \mu \frac{F_{worst}(t) - F_i}{F_{worst}(t) - F_{best}(t)}, & \text{если } F_i < F_{avg}(t), \end{cases}$$

$$\mu = \frac{1}{2}(\Lambda_{max} - \Lambda_{min}),$$

где Λ_{min} и Λ_{max} — минимальное и максимальное возможное время жизни; $F_{worst}(t)$, $F_{best}(t)$ и $F_{avg}(t)$ — соответственно, худшая, лучшая и средняя приспособленность в поколении t . В каждом поколении время жизни всех особей, кроме лучшей, уменьшается на 1 и при $\Lambda_i = 1$ хромосома удаляется из популяции.

При адаптивной самонастройке [9, 31, 41, 44, 47, 55, 74, 75, 81] значения параметров кодируются в хромосоме особи и, как и оптимизируемые параметры решения, претерпевают эволюционные изменения. Значения закодированных параметров ЭА не влияют на величины приспособленностей особей напрямую, но «хорошие» значения параметров часто способствуют улучшению особей и с течением времени распространяются в популяции. Часто подобным образом кодируют вероятности операторов скрещивания и мутации, а также их параметры. В [78] самоадаптация использовалась для определения позиции и числа точек разрыва кроссинговера. Для этого к L -разрядным хромосомам добавлялись L дополнительных бит, в которых кодировалось наличие («1») или отсутствие («0») точки разрыва в соответствующей позиции хромосомы. При скрещивании точки разрыва определялись по единичным разрядам в дополнительных разрядах хромосом скрещиваемых особей.

Необходимо отметить, что возможно комбинирование различных вариантов настройки параметров ЭА. К примеру, возможна адаптация размера популяции вместе с самоадаптацией оператора мутации [49].

Адаптация в ЭА возможна не только на уровне параметров, но и на уровне кодирования и использования генетических операторов скрещивания (рекомбинации) и мутации. Неоднократно отмечалось, что использование проблемно-зависимых способов кодирования и операторов позволяет повысить эффективность работы ЭА [3, 23, 57]. Причина заключается в том, что по «классической схеме» генетические операторы применяются

на уровне хромосом (на уровне генотипов), но не на уровне декодированных решений (фенотипический уровень), что может привести к непредсказуемым изменениям соответствующих решений, однако позволяет реализовать унифицированную схему поиска. Выходом из данной ситуации может быть использование специализированных генетических операторов, работающих на уровне фенотипов и адаптирующихся к характеристикам декодированных решений, учитывающих особенности поставленной задачи, хотя такая организация эволюционного поиска менее универсальна. Отличия между двумя схемами поиска представлены на рис. 3.

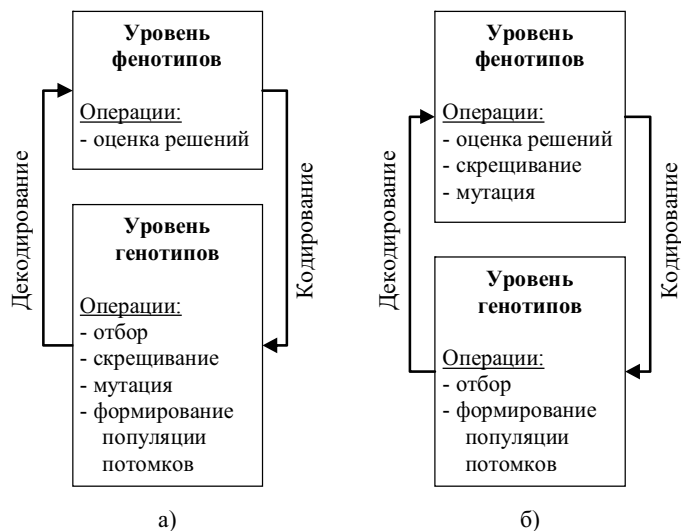


Рис. 3. Различия между схемами эволюционного поиска при реализации генетических операторов, работающих на уровне а) генотипов; б) фенотипов

Использование генетических операторов скрещивания и мутации на уровне фенотипов показало свою эффективность в генетическом программировании [57], в задачах настройки и обучения ИНС [42, 83], при решении многих задач на графах [3], а также в ряде других случаев.

Проблема формирования обучающего множества данных для ИНС

Обучение ИНС, с «классической» точки зрения, подразумевает использование обучающей выборки, включающей наборы входных сигналов $\mathbf{X} = \{\mathbf{X}_i : 1 \leq i \leq N\}$ и соответствующих эталонных значений выходных сигналов $\mathbf{Y} = \{\mathbf{Y}_i : 1 \leq i \leq N\}$. Использование обучающего множества данных дает возможность подстраивать веса связей ИНС с помощью градиентных алгоритмов, на основании информации об отклонении значений действительных выходных сигналов ИНС $\mathbf{F} = \{\mathbf{F}_i : 1 \leq i \leq N\}$ от требуемых, которое часто представляется в виде ошибки ε ИНС, например, для $\mathbf{F}_i = \{f_{ij} : 1 \leq j \leq K\}$, $\mathbf{Y}_i = \{y_{ij} : 1 \leq j \leq K\}$:

$$\varepsilon = \frac{1}{NK} \sum_i \sum_j (f_{ij} - y_{ij})^2.$$

Традиционные подходы к обучению ИНС чаще всего основываются на схеме обратного распространения ошибки [76], обычно использующей алгоритмы градиентного типа, в частности, метод сопряженных градиентов [82], метод Левенберга-Марквардта [63] и ряд других. Однако в ряде случаев формирование обучающей выборки сопряжено со сложностью определения значений компонент вектора \mathbf{Y} из обучающего множества. Данная проблема возникает при решении некорректных задач, а также если необходимо оценить последовательность выходных сигналов ИНС. К таким задачам относятся задачи, связанные с адаптивным поведением, игровыми стратегиями, обработкой изображений и др.

Одним из возможных решений этой проблемы является обучение ИНС с использованием приближенной интегральной оценки, отражающей качественные, внешние особенности функционирования сети, когда оценивается не соответствие выходных сигналов ИНС эталонным, а качество результата работы сети в целом. Примерами таких оценок являются: время поддержания стабильного состояния объекта управления для задачи нейроруправления [15, 39, 51, 67, 83]; процент выигранных игр, либо оценка позиции для задачи поиска игровых стратегий [54, 61, 71, 84]; качество изображения для задачи обработки изображений [17] и т. д. Такие оценки более естественны и «интуитивны», однако, как следствие, они не могут быть использованы градиентными алгоритмами обучения для подстройки весов связей. Поэтому «подходящий» обучающий алгоритм должен из-

менять веса связей ИНС, не имея информации о величине ошибки для каждого выхода этой сети. Одним из подходов, позволяющих использовать подобные «качественные» оценки, является рассмотренный в следующем разделе нейроэволюционный подход.

Нейроэволюционный подход

Комбинация ИНС и ЭА дает возможность совместить гибкость настройки ИНС и адаптивность ЭА, что позволяет реализовать во многом унифицированный подход к решению широкого спектра задач классификации, аппроксимации и моделирования [2, 16, 53, 68, 80, 93]. Предлагались различные варианты названий для данного направления исследований. В настоящей работе будет использоваться термин «нейроэволюция» (НЭ) [67], а соответствующий подход будем называть нейроэволюционным. Отметим, что предлагаемый вариант не является устоявшимся и окончательным.

Первые работы, посвященные применению ЭА для обучения и настройки ИНС, появились около 20 лет назад. Исследования в этой области, связаны, как правило, с решением следующих задач:

- поиск значений весов связей ИНС при фиксированной структуре;
- настройка структуры ИНС без поиска весов связей;
- настройка параметров алгоритма обучения;
- настройка параметров активационных функций нейронов;
- фильтрация обучающих данных;
- различные комбинации вышеперечисленных задач.

Рассмотрим некоторые из перечисленных выше задач и отметим их особенности.

Эволюционная настройка весов связей ИНС

Одной из типичных задач, решаемых в рамках НЭ подхода, является задача поиска весов связей ИНС при ее фиксированной структуре [1, 2, 5, 6, 39, 66, 71, 87, 90, 92]. При этом, как правило, рассматривается задача минимизации ошибки выхода ИНС, а в качестве оптимизируемых параметров используются веса связей, значения которых подбираются с помощью ЭА. Одним из обоснований эволюционного обучения ИНС является «застревание» градиентных алгоритмов в локальных экстремумах в процессе

обучения ИНС. Преимуществами использования эволюционной настройки весов связей ИНС являются:

1. Независимость от структуры ИНС и характеристик функций активации нейронов.
2. Отсутствие необходимости в обучающем множестве данных.

Первое преимущество дает возможность использовать единый подход к обучению ИНС с различной структурой. Второе преимущество позволяет осуществлять обучение ИНС без информации об эталонных значениях выходных сигналов, а на основе оценки функционирования ИНС «в целом». Для градиентных алгоритмов оптимизации обучающее множество необходимо, чтобы подстраивать веса связей ИНС на основе расхождения значений реального и требуемого выходного сигнала ИНС.

При использовании ЭА для настройки весов связей ИНС достаточно использовать оценку, отражающую качество нейросетевого решения в целом. Здесь важна адекватность оценки, т.е. если согласно используемой оценке качество первой ИНС лучше качества второй ИНС, то из этого должно следовать, что первая ИНС лучше второй. Кроме адекватности оценки также важно ее существование для любого решения, которое может быть закодировано в хромосоме. При этом такая оценка может вычисляться независимо от расхождения выходного сигнала ИНС с эталонным значением. Таким образом, отсутствие обучающей выборки не является серьезным препятствием для эволюционного обучения ИНС, если существует альтернативный способ оценки ее функционирования. При этом появляется возможность использовать приблизительные оценки функционирования ИНС и оцениваться может не каждый выходной сигнал сети, а последовательность сигналов. Это позволяет расширить круг практических приложений нейроинформатики, а также упростить процесс решения многих задач, связанных с адаптивным нейроуправлением, адаптивным поведением и моделированием.

К недостаткам использования ЭА для обучения ИНС следует отнести следующие:

1. Трудность «тонкой» настройки весов связей на поздних этапах эволюционного поиска.
2. Большие, по сравнению с градиентными алгоритмами, требования к объему оперативной памяти из-за использования популяции ИНС.
3. Проблема конкурирующих решений.

Первая проблема во многом обусловлена использованием целочисленного кодирования весов связей ИНС, что может привести к значительным «скачкам» в пространстве поиска в результате применения операторов скрещивания и мутации. Данное свойство полезно на начальном этапе работы НЭ-алгоритма, когда необходимо быстро исследовать пространство поиска, но затрудняет работу алгоритма на более поздних этапах. При использовании вещественно кодирования эта проблема не так актуальна, но не исчезает полностью из-за необходимости адаптивной настройки параметров операторов ЭА, т.к. характеристики эволюционного поиска изменяются с течением времени.

Проблема конкурирующих решений (competing conventions problem) [79, 91] известная также как *проблема перестановки (permutation problem)* [72, 77] связана с кодированием весов связей ИНС. Ее суть заключается в том, что генетическое представление допускает существование нескольких вариантов хромосом, кодирующих одну и ту же ИНС (рис. 4).

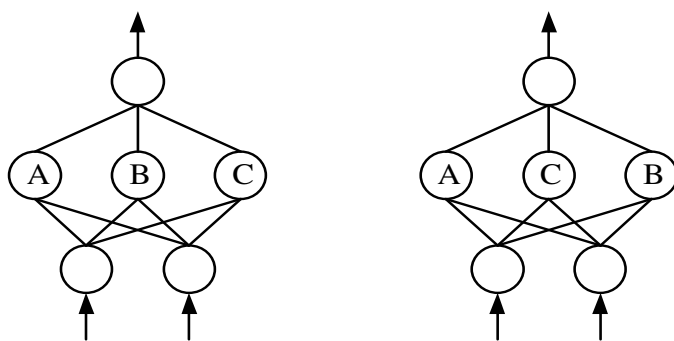


Рис. 4. Проблема перестановки. Две одинаковые ИНС могут быть закодированы различным образом «перестановкой» скрытых нейронов. Считается, что «скрещивание» таких сетей практически бесполезно [83]

Предложено несколько вариантов решения проблемы перестановки, среди которых необходимо отметить следующие:

1. Упорядочение списка весов связей кодируемых ИНС (выравнивание хромосом) по значениям весов [87], либо с использованием специальных меток [83]. Считается, что такие меры позволят уменьшить вероятность появления особей, представляющих одинаковые ИНС.

Также возможно раннее удаление особей-дубликатов на этапе формирования популяции следующего поколения.

2. Разработка адаптивных операторов скрещивания [72, 77], уменьшение размера популяции, а также повышение вероятности мутации.
3. Уменьшение роли оператора кроссинговера, либо полный отказ от его использования [18, 94]. При этом предполагается, что: (а) вероятность возникновения одинаковых ИНС в результате мутаций различных особей незначительна; (б) в случае, если такие особи все-таки появились, мутации приведут к меньшим «разрушениям», чем скрещивание.

Отметим, что вопрос об актуальности проблемы конкурирующих решений остается открытым. Из-за того, что, в общем случае, вероятность появления в одной популяции двух одинаковых ИНС мала, проблема конкурирующих решений может игнорироваться [43, 62, 67].

Эволюционная настройка структуры ИНС

Еще одной типичной задачей НЭ подхода является задача эволюционной настройки структуры ИНС. В хромосоме кодируется топология ИНС, а настройка весов осуществляется, например, с использованием градиентных алгоритмов [13, 36, 56]. Каждая особь, представляющая собой сеть той или иной структуры, оценивается в зависимости от результатов обучения: чем лучше результат, тем более приспособлена особь. Поскольку выбор топологии ИНС является, как правило, сложной задачей, решаемой методом проб и ошибок, то эволюционный поиск нейросетевой структуры способен облегчить и в определенной степени автоматизировать процесс решения задачи настройки и обучения ИНС.

Перечислим преимущества и недостатки эволюционной настройки структуры ИНС.

Преимущества:

1. Возможность автоматического поиска топологии ИНС и получения более точной нейросетевой модели за счет рассмотрения «нестандартных», нерегулярных топологий.
2. Независимость от характеристик функций активации нейронов.

Для упрощения задачи и повышения качества результатов, в процессе поиска топологии ИНС возможно использование дополнительных регулирующих ограничений, помогающих избежать чрезмерного «разрастания»

сети, которое выражается в быстром увеличении количества скрытых нейронов и связей между ними.

Недостатки:

1. Сложность оценки структуры ИНС без информации о значениях весов связей.
2. Сложность организации поиска топологии ИНС.

Первый недостаток представляет основную проблему эволюционной настройки структуры ИНС. Он, в основном, обусловлен чувствительностью результатов обучения к начальным условиям и значениям параметров алгоритма обучения. Таким образом, хромосоме, представляющей ИНС с некоторой структурой, могут соответствовать совершенно разные нейросетевые модели с точки зрения их качества и свойств (в таких случаях говорят, что одному генотипу соответствует множество фенотипов). Для уменьшения влияния случайных факторов на оценку структуры ИНС проводится несколько независимых операций обучения [13, 36], усредненный результат которых и используется в качестве оценки топологии ИНС. Несмотря на существующее мнение, что ИНС с более простой структурой обладают лучшими способностями к обобщению, этот вопрос все еще нельзя считать до конца решенным, т. к. многое зависит от результатов обучения [14].

Сложность объективной оценки топологии ИНС порождает также сложность оценки отдельных структурных частей сети и возможных структурных модификаций. Данная проблема характерна также и для алгоритмов одновременной настройки и обучения ИНС (см. раздел «Одновременная эволюционная настройка весов связей и структуры ИНС»), поэтому многие дальнейшие замечания относятся в равной степени и к эволюционному поиску структуры ИНС, и к одновременной настройке структуры и весов связей ИНС. Для изменения структуры ИНС часто применяются следующие операции:

- добавление/удаление нейронов;
- добавление/удаление связей.

Встречаются также модификации этих операций, включающие, например, «расщепление» существующего нейрона, «перенаправление» связи [67, 68] и др.

Итак, выбор типа структурного изменения ИНС в общем случае неоднозначен из-за того, что оценка необходимости модификации топологии ИНС и варианта этой модификации не всегда возможна. Известны следующие

варианты решения проблемы выбора операции модификации структуры ИНС:

- определение эффективности каждой операции преобразования топологии ИНС на основании оценки качества получаемых НС-решений [52];
- использование дополнительной информации об ИНС для ее модификации (например, вычисление «значимости» связей [94], эвристическая оценка структуры ИНС [15, 18]).

Заметим, что существуют неэволюционные подходы к настройке топологии ИНС либо во время ее обучения (алгоритм каскадной корреляции Фальмана [34] и его модификации [59] и др.), либо по завершении процесса обучения (алгоритмы Optimal Brain Damage (OBD) [58], Optimal Brain Surgeon (OBS) [45] и др.). Несмотря на то, что в [73] показана эффективность использования эволюционного подхода для настройки топологии ИНС по сравнению с алгоритмами OBD и OBS, однако детального исследования на эту тему, насколько известно автору, не проводилось.

Одновременная эволюционная настройка весов связей и структуры ИНС

Использование эволюционного подхода позволяет одновременно настраивать веса связей и структуру ИНС [15, 18, 42, 51–53, 61, 67, 68, 83, 84, 86, 94]. При этом в хромосоме кодируется информация о весах и межнейронных связях ИНС. Возможно использование как бинарного, так и вещественного кодирования для записи весов связей, а структура сети может быть представлена с использованием различных способов кодирования [16, 93].

В силу того, что в случае одновременной настройки весов связей и структуры ИНС фактически комбинируются две различные задачи, пространство поиска многократно увеличивается и часто включает подпространства различной размерности (для ИНС с разным количеством связей). Для упрощения задачи поиска в таком сложном пространстве часто в явном или неявном виде вводятся ограничения на топологию ИНС и/или ее изменения:

- ограничение количества скрытых нейронов [18, 52, 53, 67, 68];
- ограничение количества связей [35, 67, 68];
- ограничение изменения топологии ИНС (рассматриваются только растущие ИНС [83, 86], либо только многослойные ИНС [8]).

Следствием первых двух ограничений часто является фиксация длины хромосомы, что существенно уменьшает пространство возможных решений и не способствует эффективному поиску структуры ИНС, но существенно упрощает задачу и позволяет использовать «стандартные» операторы скрещивания и мутации. Для повышения гибкости поиска НЭ-алгоритма используют генетическое кодирование с переменной длиной хромосомы, которое позволяет добавлять и (в некоторых реализациях) удалять гены в процессе работы алгоритма [15, 52]. Это ведет к изменению общего порядка следования генов и требует реализации специализированных генетических операторов, учитывающих особенности используемого кодирования.

Одновременное решение двух отдельных задач — настройки весов связей и структуры ИНС, позволяет в некоторой степени скомпенсировать недостатки, присущие каждой из них в отдельности и объединить их преимущества. С другой стороны, «платой» за это является огромное пространство поиска, а также объединение ряда недостатков, вызванных использованием эволюционного подхода. Суммируя, перечислим преимущества и недостатки.

Преимущества:

1. Независимость от структуры ИНС и характеристик функций активации нейронов.
2. Отсутствие необходимости в обучающей выборке.
3. Возможность автоматического поиска топологии ИНС и получения более точной нейросетевой модели.

Одним из основных преимуществ одновременной эволюционной настройки весов связей и структуры ИНС является возможность автоматизированного поиска ИНС, располагая только критерием оценки ИНС для осуществления эволюционного поиска. При этом, как и для эволюционного обучения ИНС (см. раздел «Эволюционная настройка весов связей ИНС»), наличие обучающей выборки не является обязательным, а НЭ-алгоритм может применяться для поиска ИНС с любыми активационными функциями нейронов.

По сравнению с раздельным эволюционным поиском структуры ИНС и настройкой весов связей, одновременное решение этих задач позволяет избежать некоторых недостатков. Так появление в популяции особей, которым соответствуют ИНС с различными топологиями, уменьшает значимость проблемы конкурирующих решений (см. раздел «Эволюционная

настройка весов связей ИНС»), а наличие информации о весах связей позволяет обойти проблему субъективной оценки структуры ИНС, в силу того, что оценивается не структура нейросети, а вся ИНС «целиком».

Недостатки:

1. Сложность «тонкой» настройки весов связей на поздних этапах эволюционного поиска.
2. Большие, по сравнению с градиентными алгоритмами, требования к объему оперативной памяти из-за использования популяции ИНС.
3. Сложность организации поиска топологии ИНС.

Приложения НЭ-подхода

Помимо «стандартных» нейросетевых задач классификации и аппроксимации с использованием обучающего множества данных НЭ-алгоритмы представляют большой интерес для решения трудноформализуемых задач, где присутствует сложность формирования обучающего множества и/или затруднена оценка качества работы ИНС. Рассмотрим коротко примеры таких задач и некоторые особенности использования НЭ-подхода.

Адаптивное управление [2, 15, 40, 53, 67, 83]. При использовании НЭ-алгоритмов для решения задач адаптивного управления можно перейти к безмодельному обучению, когда не нужно знать прямую либо обратную математическую модель объекта управления¹ (ОУ), а ИНС оценивается в зависимости от используемого ОУ.

Например, в [2] для обучения ИНС управлению инерционным колебательным звеном второго порядка с передаточной функцией вида

$$W = \frac{k}{T^2 s^2 + 2T\zeta s + 1}$$

использовалась следующая оценка:

$$F = \frac{1}{3} \sum_{i=1}^3 \left(\frac{1}{t_{max}} \int_0^{t_{max}} (x(t) - u_r(t))^2 dt \right),$$

где $x(t)$ и $u_r(t)$ — соответственно выход ОУ и сигнал задания в момент времени t ; t_{max} — время интегрирования переходного сигнала в системе.

¹Поиск такой модели сам по себе часто является нетривиальной задачей.

В [15, 67, 83] при решении задачи балансирования на тележке одного либо двух шестов разной длины использовалась следующая оценка ИНС:

$$F = t,$$

где t — количество итераций, в течение которых удавалось удержать шесты от падения. В случае использования двух шестов без информации о скорости их движения, когда появляется дополнительное требование подавления колебательных движений тележки, в [40, 53, 83] оценка ИНС определяется уже следующим образом (обозначения из [83]):

$$F = 0.1f_1 + 0.9f_2,$$

$$f_1 = t/1000,$$

$$f_2 = \begin{cases} 0, & t < 100, \\ \frac{0,75}{\sum_{i=1}^{100} (|x^i| + |\dot{x}^i| + |\theta_1^i| + |\dot{\theta}_1^i|)}, & t \geq 100, \end{cases}$$

где x^i и \dot{x}^i — положение и скорость тележки, θ_1^i и $\dot{\theta}_1^i$ — положение и угловая скорость первого шеста.

Адаптивное поведение и многоагентные системы [1, 6, 37, 85]. Здесь большой интерес представляет возможность реализации интеллектуальных агентов [10] с использованием ИНС, при этом возможно наличие коллективного поведения [85, 95]. В качестве примера можно привести проект NERO (Neuro-Evolution of Robotic Operatives²) [85]. Ключевой технологией проекта является модифицированный алгоритм NEAT для одновременной настройки структуры и весов связей ИНС, разработанный Кеннетом Стэнли (Kenneth Stanley). Пользователь сначала тренирует на специальном полигоне команду НС-агентов, а затем его команда «соревнуется» с агентами компьютера или другого пользователя на своеобразном «поле боя». Поведение агентов определяется соответствующими им ИНС, которые настраиваются в процессе обучения на полигоне. Действия агентов зависят от создающейся на тренировочном полигоне ситуации и определяются приоритетами поведения (расстояние до противника, подвижность агентов,

²URL: http://dmc.ic2.org/nero_public

активность стрельбы и др.), заданными пользователем. Более приспособленные с точки зрения выбранных приоритетов агенты используются для генерации агентов-потомков, которые замещают худших из существующих агентов.

Для выбора действия (направление движения и стрельба) агентам доступна следующая информация:

- расположение агентов-противников, находящихся в поле зрения агента;
- находится ли агент-противник на линии огня;
- расположение препятствий (стены, различные объекты);
- направление стрельбы ближайшего агента-противника.

Использование для реализации проекта NERO НЭ-алгоритма позволяет в реальном времени осуществлять поиск разнообразных стратегий поведения агентов. В процессе обучения агенты учатся решать такие задачи, как поиск маршрута движения в присутствии препятствий, преследование агентов-противников, использование стен для прикрытия от огня и др. Отметим, что во время обучения агентов задачи ставятся в общем виде (дойти до определенной точки, атаковать цель и т. д.) и возникновение сложных стратегий поведения (например, укрытие за стенами) зависит от ситуации, которую создается на тренировочном полигоне.

Эволюционная робототехника (Evolutionary robotics) [28, 70]. Представляет собой сравнительно молодое направление исследований, направленное на создание и исследование автономных роботов, которые рассматриваются как искусственные организмы, поведение которых развивается без человеческого вмешательства в результате взаимодействия с окружающей средой. Эволюционная робототехника во многом опирается на науки о природе, биологию и этологию, и использует такие концепции и подходы, как нейронные сети, генетические алгоритмы, динамические системы и бионика. Основное отличие исследований в области эволюционной робототехники от исследований адаптивного поведения интеллектуальных агентов заключается в использовании реальных роботов, а не их программных моделей. Это ограничивает условия проведения экспериментов (нельзя выйти за рамки действующих физических законов и технических и материальных ограничений), но добавляет значительно больше реализма за счет необходимости учета инерции, помех, характеристик датчиков, микроконтроллеров и материалов и т. д., что представляет интерес с практической

точки зрения. Для управления роботами используются, как правило, разнообразнейшие модели ИНС и нейронов. Например, в [37] рассматривается обучение микроробота Alice [27] навигации с избеганием препятствий с использованием эволюционной настройки структуры ИНС с упрощенной моделью спайкового нейрона (используются обозначения из [37]):

$$o_i^t = \begin{cases} 1 & \text{and } v_i^t = v_i^{min}, \quad v_i^t > v_i^{max} + r^t, \\ 0, & v_i^t \leq v_i^{max} + r^t, \end{cases}$$

где o_i^t — выходной сигнал; v_i^t , v_i^{min} и v_i^{max} — соответственно текущий, минимальный и максимальный мембранный потенциал нейрона; r^t — случайное целое, необходимое для предотвращения осцилляций и «блокировок» в ИНС с обратными связями. Значение мембранного потенциала накапливается с течением времени с учетом входных сигналов нейрона и постоянной утечки. Оценка ИНС определяется в зависимости от скорости движения робота и расстояния до препятствий, которые необходимо максимизировать:

$$F = \sum_t V^t (1 - \Delta V^t) (1 - i),$$

где V^t — суммарная скорость вращения колес робота; ΔV^t — модуль разности скорости вращения левого и правого колеса; i — максимальная текущая активность одного из инфракрасных датчиков препятствий.

Поиск игровых стратегий [54, 61]. Данное направление исследований рассматривает использование ИНС для принятия решений в настольных играх. Часто рассматриваются крестики-нолики [54] и го [61, 71, 84]. Здесь оценка работы ИНС затрудняется необходимостью учета большого количества различных вариантов развития событий и связанной с этим неопределенностью, что делает саму оценку неточной. В [84] показано, что при использовании НЭ-подхода возможно успешное обучение ИНС игре в го против программы с детерминированным алгоритмом [38] на поле 5×5 на основе оценки

$$F = 100 - \left(\frac{2 \sum_{i=1}^n e_i}{n} + e_f \right),$$

где e_i — счет в партии после i -го хода, n — число ходов в партии, e_f — финальный счет партии. Также отметим вариант с «автообучением», когда

настройка ИНС осуществляется путем соревнования одной эволюционирующей популяции ИНС с другой [61] (так называемая конкурентная коэволюция (competitive coevolution)). При этом оценка ИНС формируется в результате серии матчей с ИНС из «противоборствующей» популяции:

$$F = \sum_{i \in O} \frac{1}{L_i},$$

где O — множество оппонентов, побежденных данной ИНС, а L_i — число проигранных i -й побежденной ИНС. НЭ-обучение направлено на минимизацию оценки F .

Компьютерное творчество [24, 64]. Применение ЭА-алгоритмов для компьютерного синтеза изображений [86, 88], звуков и мелодий [29, 32] и геометрических форм [69, 86] возможно благодаря использованию концепции *эстетической селекции* (*aesthetic selection*, называемой также *aesthetic evolution*, *interactive evolution*) [64]. Особенностью этой концепции является интерактивное взаимодействие ЭА с пользователем, который «вручную» определяет, какие решения (изображения, звуки и т. д.) в популяции являются более приемлемыми и, следовательно, какие особи будут допущены к скрещиванию. Это существенно увеличивает время эволюционного обучения и накладывает значительные ограничения на параметры ЭА и получаемых решений (нельзя использовать популяцию большого размера; решения должны быть такими, чтобы их можно было оценить за сравнительно небольшой промежуток времени и др. [64]), однако позволяет решать поставленную задачу, избежав очевидных и неразрешенных в настоящее время трудностей с формализацией критериев оценки решений.

Например, в [86] описан способ генерации разнообразных геометрических образов и узоров, где ИНС реализует следующее преобразование: $L = g(x_1, x_2, d)$, здесь L — нормированная яркость пикселя с прямоугольными координатами (x_1, x_2) , d — расстояние до центра изображения. Таким образом, ИНС используется для «рисования», при этом яркость пикселей зависит от их координат. Отбор ИНС по всей популяции осуществляется пользователем путем выбора понравившихся ему соответствующих образов. Пример эволюции образа, названного «космический корабль» («spaceship»), сгенерированного ИНС в различных поколениях эволюционного поиска, показан на рис. 5.

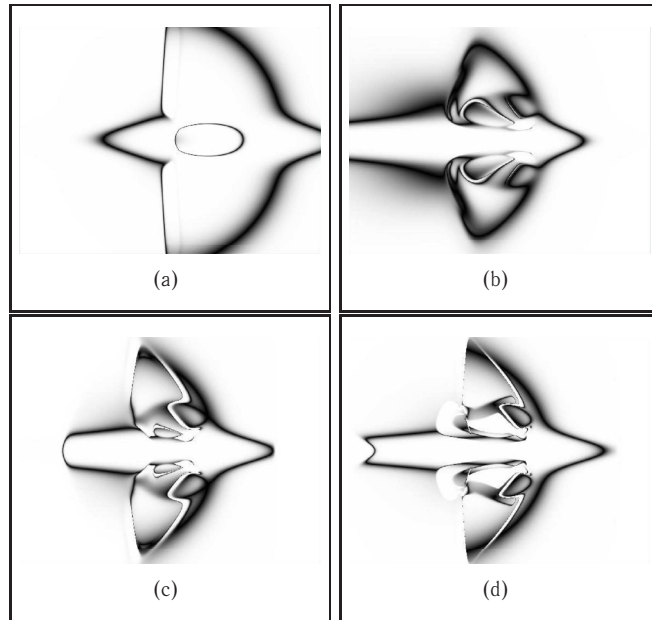


Рис. 5. Пример эволюции образа «космический корабль», сгенерированного ИНС в различных поколениях: а) начальный образ; б) появление узоров на «крыльях»; с) изменение формы «хвоста»; д) появление «выхлопов» за «крыльями».

НЭ-алгоритмы в мире

Отдельными вопросами разработки НЭ-алгоритмов активно начали заниматься в конце 80-х – начале 90-х годов, несмотря на то, что сама идея применения эволюционного подхода к настройке ИНС появилась значительно раньше (например, в [50]). В настоящее время в мире существуют лаборатории и исследовательские группы, среди которых отметим следующие.

Группа нейросетевых исследований (Neural Networks Research Group) в Университете Остина, штат Техас, под руководством *Ристо Мииккулайнена* (Risto Miikkulainen). Тематика исследований этой группы необычайно широка и включает в себя:

- обучение навигации Индивидуального Спутника-Ассистента (Personal Satellite Assistant) космонавта;
- использование НЭ в компьютерных играх реального времени;
- разработка сложных систем с использованием эволюционирующих генераторов образов (evolving patterns generators);
- разработка автомобильной системы предупреждений об опасности;
- кооперативная коэволюция (coevolution) в многоагентных системах.

Приведенный список не является полным. Также отметим, что термин «нейроэволюция», использованный применительно к моделям эволюционирующих нейронных сетей, появился именно в статьях этой группы.

Лаборатория интеллектуальных систем (The Laboratory of Intelligent Systems) в Цюрихе, под руководством *Дарио Флореано* (Dario Floreano). Исследования лаборатории направлены на развитие робототехнических систем и методов искусственного интеллекта, основанных на биологических принципах самоорганизации. Основные темы исследований:

- бионические летающие микророботы;
- эволюционное аппаратное и программное обеспечение, включая также эволюционную робототехнику;
- исследование коллективных и ройных (swarm) систем.

Лаборатория автономных роботов и искусственной жизни (Laboratory of Autonomous Robotics and Artificial Life), Рим, под руководством *Стефано Нолфи* (Stefano Nolfi). Основное направление исследований — изучение адаптивных искусственных организмов, взаимодействующих со внешней средой и управляемых ИНС, направленное на понимание механизмов, лежащих в основе индивидуального и коллективного поведения.

Центр исследований вычислительного интеллекта и его приложений (The Centre of Excellence for Research in Computational Intelligence and Applications — CERCIА) в Университете Бирмингема под руководством *Зина Яо* (Xin Yao). Среди многочисленных направлений исследований центра есть немало проектов по разработке и применению НЭ-алгоритмов с использованием эволюционного программирования.

Группа исследований оптимизации адаптивных систем (Research Group Optimization of Adaptive Systems), Институт нейроинформатики в Университете Бохума, Германия, под руководством *Кристиана Игеля*

(Christian Igel). Исследования направлены на изучение взаимодействия самоорганизации, обучения и эволюции при создании адаптивных систем и анализ роли структуры в нейросетевой обработке информации.

В **России** исследования НЭ-алгоритмов ведутся в следующих вузах и научных институтах (в скобках указаны основные направления исследований):

- Ульяновский государственный технический университет (интеллектуальные системы, принятие решений);
- Калужское отделение МГТУ им. Н. Э. Баумана (интеллектуальные системы, принятие решений);
- Институт оптико-нейронных технологий совместно с Институтом прикладной математики им. М. В. Келдыша (адаптивное поведение);
- Красноярский государственный технический университет (классификация и медицинская диагностика);
- Томский политехнический университет (обработка изображений).

Заключение

Актуальность НЭ-подхода объясняется следующими его возможностями:

- облегчение решения ряда существующих трудноформализуемых задач в силу отсутствия необходимости в этапе формирования обучающей выборки, который сам по себе может быть достаточно ресурсоемким и сложным;
- расширение прикладных областей нейроинформатики и эволюционных вычислений (например, улучшение качества цифровых изображений);
- развитие перспективных направлений, связанных с созданием интеллектуальных агентов, робототехникой, компьютерным творчеством и др.

Исследования в области ИНС, НЭ и ЭА также представляют интерес ввиду активного развития аппаратных и программных средств параллельных и распределенных вычислений. Так, например, представляет интерес реализация НЭ-алгоритмов с использованием геометрических сопроцессоров современных видеокарт.

В заключение отметим, что часто для успешного решения задачи с использованием НЭ-алгоритма достаточно адекватного понимания самой

проблемы и НЭ-подхода, свидетельством чего является большое число интересных и успешных работ в данном направлении.

Благодарности

Автор выражает благодарность *В. Г. Редько* и *Ю. В. Тюменцеву* за помощь в подготовке материала и *Ю. В. Тюменцеву* за консультации по использованию пакета \LaTeX .

Литература

1. *Бурцев М. С.* Эволюция кооперации в многоагентной системе // VII Всероссийская научно-техническая конференция «Нейроинформатика-2005»: Сб. науч. трудов. В 2-х частях. Ч. 1. – М.: МИФИ, 2005. – с. 217–224.
URL: <http://mbur.narod.ru/>
2. *Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А.* Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. – Харьков: Основа, 1997.
URL: <http://neuroschool.narod.ru/books/gannvirt.html>
3. *Емельянов В. В., Курейчик В. М., Курейчик В. В.* Теория и практика эволюционного моделирования. – М.: Физматлит, 2003.
4. *Гладков Л. А., Курейчик В. М., Курейчик В. В.* Генетические алгоритмы. – М.: Физматлит, 2006.
5. *Комарцова Л. Г.* Исследование алгоритмов обучения многослойного персептрона // *Нейрокомпьютеры: Разработка и применение.* – 2002, № 12.
6. *Мосалов О. П., Редько В. Г.* Модель эволюционной ассимиляции приобретенных навыков в нейросетевых системах управления адаптивных агентов // VII Всероссийская научно-техническая конференция «Нейроинформатика-2005»: Сб. науч. трудов. В 2-х частях. Ч. 1. – М.: МИФИ, 2005. – с. 210–217.
URL: <http://www.iont.ru/projects/rfbr/90197/>
7. *Осовский С.* Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002.
8. *Паклин Н.* Обучаем нейронную сеть генетическим алгоритмом. 2003.
URL: <http://paklin.newmail.ru>
9. *Редько В. Г.* К теории эволюции. Модель возникновения «программ жизнедеятельности» // *Журнал общей биологии.* – 1991, № 3. – с. 334–342.

10. *Тарасов В. Б.* От многоагентных систем к интеллектуальным организациям: Философия, психология, информатика. – М.: Эдиториал УРСС, 2002.
11. *Хайкин С.* Нейронные сети: Полный курс. Пер. с англ. 2-е издание. – М.: Издательский дом «Вильямс», 2006.
12. *Хомич А. В., Жуков Л. А.* Анализ и оптимизация операций мутации и кроссовера в генетических алгоритмах // Доклады V Всероссийской конференции «Новые информационные технологии в исследовании сложных структур». – Томск, Вестник ТГУ, 2004. – с. 111–114.
13. *Хомич А. В., Жуков Л. А.* Эволюционный метод оптимизации структуры нейронной сети с учителем // VII Всероссийская научно-техническая конференция «Нейроинформатика–2005»: Сб. науч. трудов. В 2-х частях. Ч. 1. – М.: МИФИ, 2005. – с. 11–18.
14. *Царегородцев В. Г.* Редукция размеров нейросети не приводит к повышению обобщающих способностей // Материалы XII Всероссийского семинара «Нейроинформатика и ее приложения». – Красноярск, 2004. – с. 163–165.
URL: <http://www.neuropro.ru/>
15. *Цой Ю. Р., Спицын В. Г.* Применение генетического алгоритма для решения задачи адаптивного нейроуправления. // VII Всероссийская научно-техническая конференция «Нейроинформатика–2005»: Сб. науч. трудов. В 2-х частях. Ч. 1. – М.: МИФИ, 2005. – с. 35–43.
URL: <http://qai.narod.ru/Publications/>
16. *Цой Ю. Р., Спицын В. Г.* Эволюционный подход к настройке и обучению искусственных нейронных сетей // Электронный рецензируемый журнал «Нейроинформатика». – 2006, том 1, № 1. – с. 34–61.
URL: <http://ni.iont.ru/Journal/N1/>
17. *Цой Ю. Р., Спицын В. Г., Чернявский А. В.* Способ улучшения качества монохромных и цветных изображений, основанный на применении эволюционирующей нейронной сети // Информационные технологии. – 2006, №. 7. – с. 27–33.
18. *Angeline P.J., Saunders G.M., Pollack J.B.* An evolutionary algorithm that constructs recurrent neural networks // *IEEE Transactions on Neural Networks*. – January 1994.
URL: <http://citeseer.ist.psu.edu/>
19. *Arabas J., Michalewicz Z., Mulawka J.* GAVAPS – a genetic algorithm with varying population size // Proc. of the *First IEEE International Conference on Evolutionary Computation*. – New York: IEEE Press, 1994. – pp. 73–78.
20. *Arnold D. V., Beyer H.-G.* Performance analysis of evolution strategies with multi-recombination in high-dimensional R^N -search spaces disturbed by noise. – Reihe CI 94/00, SFB 531, University of Dortmund, Germany, 2000.
URL: <http://sfbc.cs.uni-dortmund.de>

21. Baeck T., Eiben A.E., van der Vaart N.A.L. An empirical study on GAs “without parameters” // *Proceedings of the 6th Conf. on Parallel Problem Solving from Nature*. – LNCS No. 1917. – Berlin: Springer, 2000. – pp. 315–324.
URL: <http://www.wi.leidenuniv.nl/~gusz/>
22. Baeck T., Fogel D., Michalewicz Z. *Evolutionary computation*. – Berlin: Springer-Verlag, 2000.
23. Barnett L. *Evolutionary search on fitness landscapes with neutral networks*. – PhD Thesis. The University of Sussex, England, 2003.
URL: <http://www.cogs.susx.ac.uk/users/lionelb/>
24. Banzhaf W. *Interactive evolution* / In Baeck et al. (Eds.): *Evolutionary Computation I: Basic Algorithms and Operators*. – Bristol and Philadelphia: IOP Publishing, 2000. – pp. 228–234.
25. Beyer H.-G. An alternative explanation for the manner in which genetic algorithms operate // *BioSystems*. – 1997, No. 41, pp. 1–15.
URL: <http://sfbc.cs.uni-dortmund.de>
26. Beyer H.-G., Schwefel H.-P., Wegener I. *How to analyse evolutionary algorithms*. – Technical Report No. CI-139/02. University of Dortmund, Germany, 2002.
URL: <http://sfbc.cs.uni-dortmund.de>
27. Caprari G., Estier T., Siegart R. Fascination of downscaling – Alice the sugarcube robot // *Journal of Micro-Mechatronics*. – 2002, vol. 3, No. 1. – pp. 177–189.
28. Cliff D., Harvey I., Husbands P. Explorations in evolutionary robotics // *Adaptive Behavior*. – 1993, No. 2. – pp. 73–110.
29. Chen C.-C.J., Miikkulainen R. Creating melodies with evolving recurrent neural networks // In: *Proc. of the 2001 International Joint Conference on Neural Networks (IJCNN-01)*. – IEEE, 2001.
URL: <http://nn.cs.utexas.edu/>
30. De Jong K.A., Sarma J. Generation gaps revisited // *Foundations of Genetic Algorithms 2*. – 1993. – pp. 19–28.
URL: <http://www.cs.gmu.edu/~eclab>
31. Deb K., Beyer H.-G. Self-adaptive genetic algorithms with simulated binary crossover // *Evolutionary Computation*. – 2000, vol. 9, No. 2. – pp. 197–221.
32. “Eden” Project.
URL: www.csse.monash.edu.au/~jonmc/projects/eden/
33. Eiben A.E., Marchiori E., Valko V.A. Evolutionary algorithms with on-the-fly population size adjustment / In: X.Yao et al. (Eds.): *Parallel Problem Solving from Nature, PPSN VIII, LNCS Vol.3242*. – Berlin: Springer-Verlag, 2004. – pp. 41–50.
URL: <http://www.wi.leidenuniv.nl/~gusz/>

34. *Fahlman S.E., Lebiere C.* The cascade-correlation learning architecture / In: *Touretzky D.S.* (Ed.) *Advances in Neural Information Processing Systems 2.* – San Mateo, CA: Morgan Kaufman Publishers, 1990. – pp. 524–532.
35. *Ferreira C.* Gene expression programming: Mathematical modeling by an artificial intelligence. – Angra do Heroismo, Portugal, 2002.
URL: <http://gene-expression-programming.com/>
36. *Fiszelew A., Britos P., Perichisky G., Garcia-Martinez R.* Automatic generation of neural networks based on genetic algorithms // *Revista Eletronica de Sistemas de Informacao.* – 2003, vol. 2, No. 1.
URL: <http://citeseer.ist.psu.edu/>
37. *Floreano D., Epars Y., Zufferey J. C., Mattiussi C.* Evolution of spiking neural circuits in autonomous mobile robots // *International Journal of Intelligent Systems.* – 2006, vol. 21, No. 9. – pp. 1005–1024.
URL: <http://asl.epfl.ch/>
38. URL: <http://www.gnu.org/software/gnugo/gnugo.html>
39. *Gomez F., Miikkulainen R.* Incremental evolution of complex general behavior // *Adaptive Behavior.* – 1997, No. 5. – pp. 317–342.
URL: <http://www.nn.cs.utexas.edu/>
40. *Gomez F., Miikkulainen R.* Solving non-markovian control tasks with neuroevolution // *Proc. of the International Joint Conference on Artificial Intelligence.* – San Francisco, CA, 1999. – pp. 1356–1361.
URL: <http://www.nn.cs.utexas.edu/>
41. *Gomez J.* Self adaptation of operator rates in evolutionary algorithms // *Proc. of Genetic and Evolutionary Computation Conference 2004 (GECCO 2004).* – LNCS No. 3102, 2004. – pp. 1162–1173.
URL: <http://citeseer.ist.psu.edu/>
42. *Gruau F.* Neural network synthesis using cellular encoding and the genetic algorithm. – PhD Thesis. – l’Universite Claude Bernard, Lyon, 1994.
URL: <http://citeseer.ist.psu.edu/>
43. *Hancock P.* Recombination operators for the design of the neural nets by genetic algorithm // *Parallel Problem Solving from Nature (PPSN-II).* – North-Holland, 1992. – pp. 441–450.
44. *Hansen N., Ostermeier A.* Completely derandomized self-adaptation in evolution strategies // *Evolutionary Computation.* – 2000. – vol. 9, No. 2. – pp. 159–195.
45. *Hassibi B., Stork D.G.* Second order derivatives for network pruning: Optimal brain surgeon / In: *Hanson S.J., Cowan J.D., Giles C.L.* (Eds.) *Advances in Neural Information Processing Systems 5.* – San Mateo, CA: Morgan Kaufman Publishers, 1993. – pp. 164–171.

46. *Hesser J., Manner R.* Towards an optimal mutation probability for genetic algorithms / In: *H.-P. Schwefel, R. Manner* (Eds.): Proc. of the 1st Conference on Parallel Problem Solving from Nature. – LNCS No. 496. – Berlin: Springer-Verlag, 1990. – pp. 23–32.
47. *Hinterding R.* Gaussian mutation and self-adaptation in numeric genetic algorithms // *IEEE International Conference on Evolutionary Computation*. – IEEE Press, 1995. – pp. 384–389.
48. *Hinterding R., Michalewicz Z., Eiben A.E.* Adaptation in evolutionary computation: A survey // Proc. of the 4th IEEE International Conference on Evolutionary Computation. – Indianapolis, 1997. – pp. 65–69.
URL: <http://www.wi.leidenuniv.nl/~gusz/>
49. *Hinterding R., Michalewicz Z., Peachey T.* Self-adaptive genetic algorithm for numeric functions / In: *H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel* (Eds.): Parallel Problem Solving from Nature – PPSN IV, LNCS Vol. 1141. – Berlin: Springer-Verlag, 1996. – pp. 420–429.
50. *Holland J.H.* Adaptation in natural and artificial systems. – The University of Michigan Press, 1975.
51. *Igel C.* Neuroevolution for reinforcement learning using evolution strategies // Proc. of *Congress on Evolutionary Computation 2003* (CEC 2003), Vol. 4. – IEEE Press, 2003. – pp. 2588–2595.
URL: <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/igel/top.html>
52. *Igel C., Kreutz M.* Operator adaptation in evolutionary computation and its application to structure optimization of neural networks // *Neurocomputing*. – 2003, vol. 55, No. 1–2. – pp. 347–361.
URL: <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/igel/top.html>
53. *Igel C., Wiegand S., Friedrichs F.* Evolutionary optimization of neural systems: The use of strategy adaptation / In: *M.G. de Bruin, D.H. Mache, J. Szabados* (Eds.): Trends and Applications in Constructive Approximation. International Series of Numerical Mathematics. – Birkhauser Verlag, 2005.
URL: <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/igel/top.html>
54. *James D., Tucker P.* A comparative analysis of simplification and complexification in the evolution of neural network topologies // Proc. of *Genetic and Evolutionary Computation Conference* (GECCO-2004). – New York, NY: Springer-Verlag, 2004.
URL: <http://citeseer.ist.psu.edu/>
55. *Kita H.* A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms // *Evolutionary Computation*. – 2000, vol. 9, No. 2. – pp. 223–241.
56. *Kitano H.* Designing neural network using genetic algorithm with graph generation system // *Complex Systems*. – 1990, No. 4. – pp. 461–476.

57. Koza J. Genetic programming: a paradigm for genetically breeding computer population of computer programs to solve problems. – MIT Press, Cambridge, MA, 1992.
58. LeCun Y., Denker J. S., Solla S. A. Optimal brain damage / In: Touretzky D. S. (Ed.): Advances in Neural Information Processing Systems 2. – San Mateo, CA: Morgan Kaufman Publishers, 1990. – pp. 598–605.
URL: <http://yann.lecun.com/>
59. Littmann E., Ritter H. Cascade network architectures // Proc. of *International Joint Conference on Neural Networks*, 1992, vol. 2. – pp. 398–404.
60. Lobo F. The parameter-less genetic algorithm: Rational and automated parameter selection for simplified genetic algorithm operation. – PhD Thesis. – The University of Lisbon, Portugal, 2000.
61. Lubberts A., Miikkulainen R. Co-evolving a go-playing neural network // Proc. of *Coevolution: Turning Adaptive Algorithms upon Themselves*, Birds-of-a-Feather Workshop, Genetic and Evolutionary Computation Conference (GECCO-2001). – San Francisco, CA: Kaufmann, 2001. – pp. 14–19.
URL: <http://www.nn.cs.utexas.edu/>
62. Maniezzo V. Genetic evolution of the topology and weight distribution of neural networks // *IEEE Transactions of on Neural Networks*. – 1994, vol. 5, No. 1.
63. Marquardt D. An algorithm for least squares estimation of nonlinear parameters // *SIAM*, 1963. – pp. 431–442.
64. McCormack J. New challenges for evolutionary music and art // *SIGEVolution newsletter*. – 2006, vol. 1, No. 1. – pp. 5–11.
65. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. – Berlin: Springer-Verlag, 1992.
66. Montana D., Davis L. Training feedforward neural networks using genetic algorithms // Proc. of the *11th International Joint Conference on Artificial Intelligence*. – San Mateo, CA: Morgan Kaufmann, 1989. – pp. 762–767.
67. Moriarty D., Miikkulainen R. Efficient reinforcement learning through symbiotic evolution // *Machine Learning*, 1996, No. 22. – pp. 11–33.
URL: <http://www.nn.cs.utexas.edu/>
68. Moriarty D. E., Miikkulainen R. Forming neural networks through efficient and adaptive coevolution // *Evolutionary Computation*. – 1998, vol. 5, No. 4.
URL: <http://www.nn.cs.utexas.edu/>
69. Nguyen T. C., Huang T. S. Evolvable 3D modeling for model-based object recognition systems / In: Kinnear K. (Ed.): *Advances in Genetic Programming*. – Cambridge, MA: MIT Press, 1994. – pp. 459–475.

70. *Nolfi S., Floreano D.* Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. – Cambridge, MA: MIT Press/Bradford Books, 2000.
71. *Perez-Bergquist A.S.* Applying ESP and region specialists to neuro-evolution for go. – Technical Report CSTR01-24. The University of Texas at Austin, 2001.
URL: <http://www.nn.cs.utexas.edu/>
72. *Radcliffe N.* Genetic set recombination and its application to neural network topology optimization // *Neural Computing and Applications*. – 1993, vol. 1. – pp. 67–90.
73. *Ragg T., Braun H., Landsberg H.* A comparative study of neural network optimization techniques // In Proc. of the *ICANN'97*. – Springer-Verlag, 1997.
74. *Rechenberg I.* Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution // *Werkstatt Bionik und Evolutionstechnik*. – Stuttgart: Frommann-Holzboog, 1973.
75. *Rudolph G.* Self-adaptive mutations may lead to premature convergence // *IEEE Transactions on Evolutionary Computation*. – 2001, vol. 5, No. 4. – pp. 410–414.
76. *Rumelhart D.E., Hinton G.E., Williams R.J.* Learning representations of back-propagation errors // *Nature* (London). – 1986, vol. 323. – pp. 533–536.
77. *Saravanan N., Fogel D.B.* Evolving neural control systems // *IEEE Expert*. – 1995, pp. 23–27.
78. *Schaffer J.D., Morishima A.* An adaptive crossover distribution mechanism for genetic algorithms / In: *J.J. Grefenstette* (Ed.): Proc. of the 2nd International Conference on Genetic Algorithms and Their Applications. – Lawrence Erlbaum Associates, 1987. – pp. 36–40.
79. *Shaffer J., Whitley D., Eshelmann L.* Combination of genetic algorithms and neural networks: A survey of the state of the art // In Proc. of the *International Workshop on Combination of Genetic Algorithms and Neural Networks* (COGANN-92). – Los Alamos, CA: IEEE Computer Society Press, 1992. – pp. 1–37.
URL: <http://www.cs.colostate.edu/~genitor/>
80. *Schmidhuber J., Wierstra D., Gomez F.J.* Evolino: Hybrid neuroevolution / Optimal linear search for sequence learning // Proc. of the *19th International Joint Conference on Artificial Intelligence* (IJCAI), Edinburgh, 2005. – pp. 853–858.
URL: <http://www.idsia.ch/~juergen>
81. *Schwefel H.-P.* Numerische Optimierung von computer-modellen mittels der evolutionsstrategie // *Interdisciplinary Systems Research*, 1977, vol. 26.
82. *Shewchuk J.R.* An introduction to the conjugated gradient without the agonizing pain. – Carnegie-Melon University, Pittsburg, PA, 1994.

83. Stanley K. O., Miikkulainen R. Evolving neural networks through augmenting topologies // *Evolutionary Computation*. – 2002, vol. 10, No. 2. – pp. 99–127.
URL: <http://www.nn.cs.utexas.edu/>
84. Stanley K. O., Miikkulainen R. Evolving a roving eye for go // In Proc. of *Genetic and Evolutionary Computation Conference (GECCO-2004)*. – New York, NY: Springer-Verlag, 2004.
URL: <http://www.nn.cs.utexas.edu/>
85. Stanley K. O., Bryant B. D., Miikkulainen R. Real-time neuroevolution in the NERO videogame // *IEEE Transactions on Evolutionary Computation: Special Issue on Evolutionary and Games*. – 2005, vol. 9, No. 6. – pp. 653–668.
URL: <http://www.nn.cs.utexas.edu/>
86. Stanley K. Patterns without development. – Technical Report CS-TR-06-01. – University of Central Florida, 2006.
87. Thierens D. Non-redundant genetic coding for neural networks. – Technical Report No.UU-CS-1998-46. – Utrecht University, Netherlands, 1998.
URL: <http://www.cs.uu.nl/research/techreps/aut/dirk.html>
88. Todd S., Latham W. Evolutionary art and computers. – London: Academic, 1992.
89. Turing A. M. Computing machinery and intelligence // *Mind*. – 1950, vol. 236, No. 59.
90. Valsalam V., Bednar J., Miikkulainen R. Constructing good learners using evolved pattern generators // In Proc. of the *Genetic and Evolutionary Computation Conference (GECCO)*. – Washington, DC, 2005.
URL: <http://www.nn.cs.utexas.edu/>
91. Whitley D., Starkweather T., Bogart C. Genetic algorithms and neural networks: Optimizing connections and connectivity // *Parallel Computing*. – 1990, vol. 14. – pp. 341–361.
URL: <http://www.cs.colostate.edu/~genitor/>
92. Yamamichi T., Saito T., Taguchi K., Torikai H. Synthesis of binary cellular automata based on binary neural networks // Proc. of *International Joint Conference on Neural Networks*. – Montreal, Canada, 2005. – pp. 1361–1365.
93. Yao X. Evolving artificial neural networks // *Proc. of the IEEE*. – 1999, vol. 87, No. 9. – pp. 1423–1447.
URL: <http://www.cs.bham.ac.uk/~xin>
94. Yao X., Liu Y. Making use of population information in evolutionary artificial neural networks // *IEEE Transactions on Systems, Man, and Cybernetics*. Part B: Cybernetics, 1998. – vol. 28, No. 3. – pp. 417–425.
URL: <http://www.cs.bham.ac.uk/~xin>

95. *Yong C.H., Miikkulainen R.* Cooperative coevolution of multi-agent systems. – Technical Report AI01-287. – The University of Texas at Austin, 2000.
URL: <http://www.nn.cs.utexas.edu/>
96. *Yu T.-L., Sastry K., Goldberg D.E.* Online population size adjusting using noise and substructural measurements. – IlliGAL Report No. 2005017. – The University of Illinois, 2005.
URL: <http://www-illigal.ge.uiuc.edu:8080/>

Юрий Робертович ЦОЙ, ассистент кафедры вычислительной техники Томского политехнического университета (г. Томск). Область научных интересов — эволюционные вычисления, нейроэволюционные алгоритмы, самоорганизация. Автор более 20 научных публикаций.