

Курейчик Виктор Михайлович

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: kur@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634393260.

Kureychik Viktor Mihailovich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: kur@tsure.ru.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +78634393260.

УДК 519.7:004.8 + 004.8.023; 004.81.85

В.В. Курейчик, С.И. Родзин

О ПРАВИЛАХ ПРЕДСТАВЛЕНИЯ РЕШЕНИЙ В ЭВОЛЮЦИОННЫХ АЛГОРИТМАХ*

Рассматривается на основе общей теории эволюционных вычислений конфигурация основных элементов эволюционных алгоритмов. Анализируются известные шаблоны для проектирования генотипа и способы кодирования решений. Предлагаются дополнительные правила проектирования генотипа и обобщаются свойства кодирования. Формулируются основные проблемы представления решений в эволюционных алгоритмах.

Генотип; фенотип; эволюционный алгоритм; теория эволюционных вычислений.

V.V. Kureichik, S.I. Rodzin

ON THE RULES FOR THE SUBMISSION DECISIONS IN EVOLUTIONARY ALGORITHM

On the basis of the general theory of evolutionary computation of the configuration key elements of evolutionary algorithms. Analyzed the known patterns for the design of the genotype and ways of coding decisions. We offer additional design rules genotype and summarizes the properties of coding. Formulated the basic problem of representation of solutions in evolutionary algorithms.

Genotype, phenotype, evolutionary algorithm, evolutionary computation theory.

Введение. Процесс эволюции биологических организмов шел миллиарды лет. Если учесть скорость, с которой размножаются простейшие организмы, и подсчитать количество комбинаций, опробованных природой в процессе эволюции, то становится ясно, что современные вычислительные мощности просто несопоставимы с требуемыми. Поэтому многие исследователи эволюционных алгоритмов заняты сейчас проблемами оптимизации процесса эволюции. Это и создание представлений решений, которые могли бы сохранить и повторно использовать возникающие в процессе моделирования эволюции «строительные блоки», и совершенствование механизмов селекции и скрещивания, порождающих относительно большее количество хороших, не вырожденных решений. Создание таких механизмов позволит моделировать эффективную эволюцию с приемлемой скоростью.

* Работа выполнена при поддержке: РФФИ (гранты № 09-01-00492, № 10-01-00115), г/б № 2.1.2.1652.

Согласно общей теории эволюционных вычислений [1] множество эволюционных алгоритмов (ЭА) включает генетические алгоритмы, генетическое программирование, эволюционные стратегии, эволюционное программирование, обучающие классификаторы. Эволюционные алгоритмы представляют собой процесс математических преобразований, позволяющих трансформировать входной поток информации в выходной по правилам, основанным на имитации механизмов эволюционного поиска, на статистическом подходе к исследованию ситуаций и итерационном приближении к искомому решению. Поиск – это движение в структурированном пространстве от одних узлов этого пространства к другим. Движение по структуре поискового пространства определяется стратегией поиска. ЭА решают задачи при помощи поиска, т.е. они генерируют потенциальные решения и модифицируют их до тех пор, пока решения не будут удовлетворять заданным условиям поиска.

С точки зрения эффективности (вычислительной мощности) поиска, согласно NFL-теореме [2], все ЭА «в среднем» (по всем возможным целевым функциям) являются идентичными. Это означает, что удачное применение ЭА для решения той или иной задачи, во многом определяется их удачной настройкой. Строгих правил настройки параметров ЭА и кодирования решений универсальных для всех задач, согласно NFL-теореме, нет и быть не может, однако в данной работе мы сформулируем некоторые рекомендации по правилам представления решений.

Конфигурация элементов эволюционных алгоритмов. Важными элементами эволюционных алгоритмов являются представление (кодирование) решений, а также правила, по которым эволюционирует реализуемая ЭА модель.

Вначале фиксируется популяция P_0 решений (подмножество объектов X), обладающих некоторыми фенотипическими свойствами и связанных друг с другом посредством определенной структуры. На основе свойств объектов формируется критерий F (не обязательно аналитическое выражение), согласно которому среди объектов выбираются наилучшие:

$$X_{opt} = \operatorname{argmax} F [\varphi^{-1}(s)], s \in S,$$

где в качестве множества S строится, так называемое, генотипическое кодовое пространство – множество представлений объектов x в виде кодов (хромосом) s . Эволюция множества X задается эволюцией представления S . Таким образом, представление генотипа S и отображение генотип – фенотип $\varphi^{-1}: S \rightarrow X$ имеет важное значение для организации работы ЭА. Поясним это положение теории эволюционных вычислений.

Предполагается, что целью задачи поиска является нахождение объекта с некоторыми свойствами. Как правило, поиск производится среди конечного (иногда бесконечного) множества объектов (потенциальных решений). Первый шаг при решении задачи поиска состоит в том, чтобы определиться относительно пространства исследуемых объектов множества X . В качестве этого пространства могут фигурировать пространство n -мерных векторов вещественных чисел, множество позиций в игре или множество вариантов размещения и трассировки микросхемы.

Второй шаг, который должен предшествовать процедуре поиска состоит в выборе некоторого представления объектов из пространства X . Представление определяется множеством S – пространством представлений. Множество S выбирается с таким расчетом, что алгоритму поиска будет легче манипулировать с элементами множества S , нежели X .

В отличие от пространства объектов X , пространство представлений S обязательно конечно. Таким образом, отображение $\varphi^{-1}: X \rightarrow S$ есть функция кодирования, которая описывает связь между исследуемыми объектами, выступающими в качес-

ве потенциальных решений задачи поиска, и объектами, управлением и манипулированием которых занимается ЭА. Обратное отношение (функция декодирования) $\varphi^{-1}: S \rightarrow X$ используется тогда, когда по новому представлению, полученному в результате поиска, требуется восстановить соответствующее ему решение из X .

Использование представлений решения позволяет осуществлять поиск практически при минимуме информации о характере и свойствах пространства объектов. Достаточно только той информации, которая позволяет описать ландшафт критериальной функции в пространстве представлений.

Существуют разнообразные типы задач поиска: задача о назначении, задача распознавания, задача верификации, игровые задачи и др. Одним из самых распространённых типов являются задачи поисковой оптимизации. Их особенность состоит в том, что число шагов и число вариантов одного шага при поиске решения больше единицы, целевые состояния неизвестны, но требуется найти наилучшее решение при существующих ограничениях. При этом на множестве объектов X должен быть определен критерий качества (оптимальности) F , позволяющий сравнивать решения. Об оптимальности того или иного решения можно говорить лишь тогда, когда исследовано все пространство представлений S .

Для реализации алгоритма поиска в пространстве представлений вводится целевая функция оценки представлений F^{-1} , аналогичная функции F , определенная на элементах из множества S . Иногда, в зависимости от конкретных операторов алгоритма поиска, бывает необходимо, чтобы целевая функция принимала положительные значения.

Таким образом, прежде, чем построить поисковый алгоритм, включающий композицию генетических операторов $A(P_i)$, необходимо определиться с символической моделью задачи. Она включает в себя пространство потенциальных решений X , пространство представлений S , функции кодирования φ и декодирования φ^{-1} , функцию качества (пригодности, приспособленности) F и целевую функцию оценки представлений F^{-1} . Конфигурация основных элементов, характеризующих ЭА, имеет вид, представленный на рис. 1.

Шаблоны представления решений в алгоритмах эволюционных вычислений. Возникает вопрос: каковы общие правила их проектирования? Гольдберг в [3] определил два общих шаблона для проектирования генотипа ЭА:

- ◆ представление решений в пространстве поиска должна быть как можно более коротким, а различные совместимые по фенотипу представления не должны взаимно влиять друг на друга;
- ◆ алфавит и кодирование длин различных генов должны быть, по возможности, минимальными.

В течение последних десяти лет были разработаны различные способы кодирования для обеспечения эффективного выполнения генетических алгоритмов (ГА) [4]. Эти способы можно разделить на следующие классы:

- ◆ бинарное кодирование;
- ◆ кодирование действительными числами;
- ◆ целочисленное кодирование;
- ◆ кодирование общей структуры данных.

Пусть, например, генетический алгоритм работает с множеством (популяцией) искусственных хромосом (индивиды или строки). Каждая хромосома \bar{a} состоит из L бит (генов): $\bar{a} = (a_1, a_2, \dots, a_L)$, где $a_i \in \{0, 1\}$ – аллель. Хромосомы могут состоять из n отдельных сегментов ($n \leq L$), каждому из которых может соответствовать некоторая переменная в рассматриваемой задаче. Сегмент j ($j = 1, 2, \dots, n$) содержит при двоичном способе кодирования значение переменной, причем длина сегментов может различаться.

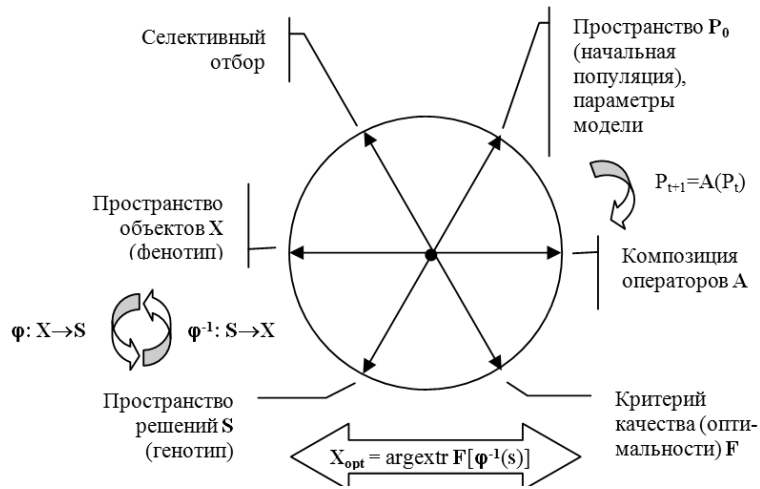


Рис. 1. Конфигурация основных элементов генетического алгоритма

Предположим, что в задаче необходимо максимизировать целевую функцию $F(\bar{U})$, где $\bar{U} = (u_1, u_2, \dots, u_n)$ – переменные, от которых зависит функция. При двоичном кодировании строинги имеют конечную длину, так как каждая переменная u_j ($j = 1, 2, \dots, n$) принимает значения в интервале $[u_{j \min}, u_{j \max}]$, что делает пространство поиска решения ограниченным, т.е.

$$F : \prod_{j=1}^n [u_{j \min}, u_{j \max}] \rightarrow R^+,$$

где R^+ – множество положительных вещественных чисел. Тогда при заданной функции декодирования вида

$$\Gamma : \{0,1\}^L \rightarrow \prod_{j=1}^n [u_{j \min}, u_{j \max}]$$

можно получить декодируемое значение переменной $\bar{u} = \Gamma(\bar{a})$ из двоичного строинга. Действительно, рассмотрим j -й сегмент. Его длина равна L_j бит, он кодирует переменную u_j . Обозначим через a_{jz} бит с номером z ($z = 1, 2, \dots, L_j$), принадлежащий j -ому сегменту строинга. Тогда декодирование происходит следующим образом:

$$\Gamma_j(a_{j1}, \dots, a_{jL_j}) = u_{j \min} + \frac{u_{j \max} - u_{j \min}}{2^{L_j} - 1} \cdot \left(\sum_{z=1}^{L_j} a_{j(L_j-z+1)} \cdot 2^{z-1} \right) = u_j,$$

при этом вещественные числа в двоичном коде можно представить с ограниченной точностью. Например, для двоичного представления вещественных переменных в интервале $-1 \leq u \leq 2$ с шагом 0,1 требуется не менее $L = 5$ бит. Числу $u = -1$ будет соответствовать строинг 00000, числу $u = 2$ – строинг 11111, а строингу $a_1 a_2 a_3 a_4 a_5 = 11001$ будет соответствовать декодируемое значение:

$$u = -1 + \frac{2 - (-1)}{2^5 - 1} \cdot \left(\sum_{z=1}^5 a_{5-z+1} \cdot 2^{z-1} \right) = -1 + \frac{3}{31} \cdot 25 \approx 1,4.$$

В генетических алгоритмах это служит одной из причин того, что при поиске оптимального значения функций от вещественных переменных оптимум достигается лишь приблизительно.

Недостатки бинарного кодирования заключаются в существовании хеммингова сдвига для пары закодированных значений, имеющих большое хеммингово расстояние, в то время, как эти величины принадлежат к точкам с минимальным расстоянием в фенотипическом пространстве. Например, пара хромосом 01111111 и 10000000 принадлежит соседним точкам в фенотипическом пространстве (точки с минимальным евклидовым расстоянием), но имеют максимальное хеммингово расстояние в генотипическом пространстве. Для преодоления хеммингова сдвига все биты должны изменяться одновременно, но вероятность такого события очень мала. Кроме того, бинарное кодирование для задач с сотней параметров оказывается слишком громоздким, длинная кодировка повышает вероятность «преждевременной» сходимости. К тому же применение длинных кодировок вовсе не гарантирует, что найденное решение будет обладать требуемой точностью, поскольку эволюционный алгоритм этого, в принципе, не гарантирует. Поэтому для многих задач затруднительно представить их решение с помощью только бинарного кодирования, т.е. бинарное кодирование во многих случаях может оказаться неэффективным.

С этой точки зрения кодирование действительными числами является более эффективным при решении задач оптимизации функций. Причина состоит в том, что топологическая структура пространства генотипов для этого вида кодирования идентична структуре в пространстве фенотипов. Поэтому появляется возможность сформировать эффективные генетические операторы заимствованием полезных приемов у традиционных методов.

Практика применения различных разновидностей эволюционных алгоритмов показывает, что целочисленное кодирование лучше всего подходит для комбинаторных оптимизационных задач [4].

В соответствии с кодированием общей структуры данных различают одномерный и многомерные способы кодирования. В большинстве случаев используется одномерное кодирование, однако многие проблемы требуют решений в многомерном пространстве, с использованием многомерного кодирования [4].

Дополнительные правила проектирования генотипа в эволюционных алгоритмах. Общая теория эволюционных вычислений [1] позволяет дополнить представленные выше два общих шаблона следующими правилами для проектирования генотипа в эволюционных алгоритмах:

- ◆ представление генотипа должно быть сюръективным по отношению ко всем фенотипам, т.е. каждый элемент множества S является образом хотя бы одного элемента множества X :

$$\forall x \in X \Rightarrow \exists s \in S : x = \varphi^{-1}(s);$$

- ◆ для любой пары фенотипов мощность их генотипов должна быть примерно одинакова:

$$\forall x_1, x_2 \in X \Rightarrow |\{s \in S : x_1 = \varphi^{-1}(s)\}| \approx |\{s \in S : x_2 = \varphi^{-1}(s)\}|;$$

- ◆ небольшие изменения в генотипе должны приводить к небольшим изменениям в фенотипе:

$$\forall x_1, x_2 \in X, s \in S : x = \varphi^{-1}(s) \& x_2 = \varphi^{-1}(s) \Rightarrow x_1 \approx x_2.$$

Действительно, ЭА два типа пространств: генотипа и фенотипа. Согласно общей теории эволюционных вычислений, операторы эволюции работают в пространстве генотипов, а оценка и отбор происходят в фенотипическом пространстве, при этом в процессе отбора осуществляется прямая взаимосвязь между хромосомами и поведением декодированных решений.

Таким образом, отображение из генотипического пространства в фенотипическое ($\varphi^{-1}: S \rightarrow X$) оказывает значительное влияние на поведение эволюционных алгоритмов. Одна из главных проблем здесь заключается в том, что некоторые хромосомы могут соответствовать или недопустимым или незаконным решениям задачи. Эта проблема представлена на рис. 2 в виде отображения пространств генотип-фенотип. Уточним эти термины, поскольку в некоторых литературных источниках они используются не совсем корректно.

Недопустимость относится к ситуации, когда решение, декодированное из хромосом, лежит вне области допустимых решений данной задачи. Незаконность относится к ситуации, когда в процессе работы операторов алгоритма ЭВ получаются хромосомы, которые не описывают решение конкретной задачи.

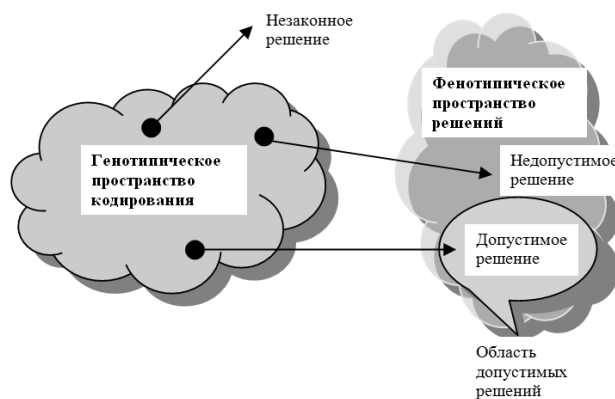


Рис. 2. Отображение из генотипа в фенотип

Таким образом, недопустимость хромосом следует из существования многих оптимизационных задач с ограничениями. Независимо от того, какой метод решения оптимизационной задачи используется – традиционный или эволюционный алгоритм – он должен иметь дело с ограничениями. Для многих оптимизационных задач допустимая область обычно представляется в виде системы уравнений или неравенств, причём оптимальное решение зачастую находится на границе между допустимыми и недопустимыми областями. В таких случаях при работе с недопустимыми хромосомами целесообразно применять процедуры по аналогии с методом штрафных функций, что позволяет ещё более эффективно организовать процесс эволюционного поиска с целью приближения к оптимуму с двух сторон (допустимой и недопустимой).

Что касается незаконности хромосом, то она вытекает из природы их кодирования. В процессе решения многих задач комбинаторной оптимизации методы кодирования обычно дают незаконное потомство посредством простой одноточечной операции кроссинговера. Однако, поскольку незаконная хромосома не может быть декодирована в решение, то штрафные методы в такой ситуации являются неприменимыми.

Как следствие, при использовании кодирования необходимо рассмотреть, можно ли в базовом цикле ЭА (вычисление целевой функции – оценка качества

решений – селекция – репродукция) провести эффективный поиск решений этим способом или нет. Иными словами, необходимо дать предварительную оценку способу кодирования. Одна из таких оценок известна – легальность кодирования. Свойство легальности кодирования означает, что любая перестановка в хромосоме, вследствие применения большинства операторов эволюции, должна соответствовать законному решению. Общая теория эволюционных вычислений позволяет построить обобщение свойств кодирования:

- ◆ недостаточность;
- ◆ полнота;
- ◆ причинность.

Свойство недостаточности вытекает из дополнительных правил, представленных выше для проектирования генотипа в эволюционных алгоритмах, и означает, что отображение между кодированием (генотипом) и решением (фенотипом) должно иметь вид: $1 - k - 1$. В общем случае такое отображение может относиться к одному из следующих типов, представленных на рис. 3:

- ◆ $1 - k - 1$ («один к одному»);
- ◆ $n - k - 1$ («многие к одному»);
- ◆ $1 - k - n$ («один ко многим»).

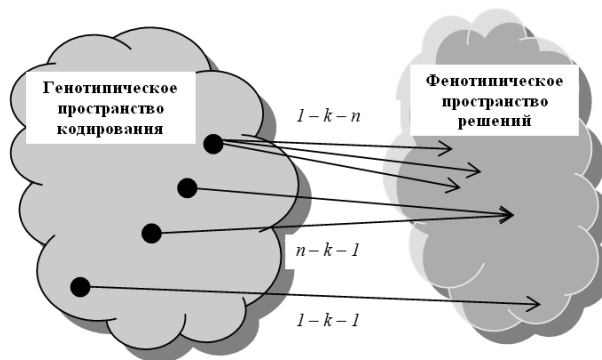


Рис. 3. Типы отображений пространств генотип/фенотип

Самый лучший способ – это отображение «один к одному». Если имеет место отображение «многие к одному», то для ЭА неизбежны дополнительные затраты времени на поиск, так как необходимы дополнительные процедуры на фенотипическом пространстве для определения единственного из n возможных решений.

Свойство полноты означает, что любое решение имеет соответствующее кодирование и любая точка в пространстве решений доступна для эволюционного поиска.

Свойство причинности связывает малые вариации в хромосомах пространства генотипов, которые появляются вследствие мутации, с малыми вариациями в фенотипическом пространстве. Это свойство было предложено, в первую очередь, для эволюционных стратегий [5]. Оно определяет сохранение родственных (соседних) структур. Таким образом, для успешного введения новой информации посредством мутации ее оператор должен сохранять структуру соседства в соответствующем фенотипическом пространстве.

Проблемы представления решений в эволюционных алгоритмах. Развитие теории эволюционных вычислений, инспирированных природными системами, идёт, в основном, по пути имитации механизмов эволюционного поиска за

счёт модификации разнообразных эволюционных операторов. Однако нельзя игнорировать следующие проблемы представления решений в ЭА:

1. *Полнота* (представление полно, если любое принципиально возможное решение может быть сконструировано в ходе работы алгоритма).
2. *Замкнутость* (представление замкнуто, если каждый генотип может быть декодирован в допустимый фенотип). Возможно, не все генотипы будут соответствовать допустимым фенотипам, но система никогда не сгенерирует недопустимый генотип.
3. *Компактность*. Если два генотипа декодируются в один и тот же фенотип, то компактнее тот из них, который занимает меньше места в памяти.
4. *Масштабируемость по времени и по размеру*. Масштабируемость по времени оценивает зависимость времени декодирования генотипа от его размера, масштабируемость по размеру оценивает зависимость размера генотипа от размеров соответствующего ему фенотипа.
5. *Множественность*. Представление обладает генотипической множественностью, если несколько различных геномов могут декодироваться в один фенотип. Фенотипическая множественность возникает тогда, когда несколько фенотипов соответствуют одному генотипу.
6. *Приспособляемость*. Представление обладает этим свойством, когда декодирование фенотипа по генотипу зависит от внешней среды.
7. *Сложность* (структурная сложность генотипа, сложность декодирования).

Эти проблемы имеют существенное значение для понимания взаимосвязей генотипа и фенотипа. Недетерминированность эволюции связана с тем, что требуемая для наилучшей приспособленности системы функция может реализовываться разными эволюционно возникшими структурами. Следовательно, одной из причин непредсказуемости пути и исхода эволюции является неоднозначность отношений «структура-функция», «фенотип-генотип».

Заключение. Генотип в эволюционных алгоритмах представляет собой не механическую сумму автономных, независимо действующих генов, а сложную и целостную систему – генотипическую среду, в которой работа и реализация каждого гена зависят от влияния других генов. Фенотип есть результат взаимодействия генотипа и внешней среды.

Содержание генотипа в большой мере зависит от свойств «читающего» его фенотипа, а не только от свойств генотипа: генотип пассивен, а фенотип является активным агентом в процессе взаимодействия генотипа и фенотипа. Это означает, что устойчивость содержания наследственной информации в большей мере зависит от устойчивости фенотипа, чем от устойчивости элементов генотипа [6]. Генотип характеризует не большая, чем у фенотипа, устойчивость, а неуничтожимость при смене поколений.

Эти общие соображения относительно представления решений в эволюционных алгоритмах вполне коррелируются с происходящей сейчас «революцией» в геномике. На смену менделевской, популяционной и классической молекулярной генетике приходит новая генетическая парадигма – постгеномика. Изучение генов как физических объектов предоставило свидетельства, отрицающие даже самые фундаментальные предположения Менделя.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-27.
2. Wolpert D.H., Macready W.G. No free lunch theorems for search // Operations research: Santa Fe Institute. – 1995.

3. *Goldberg D.E.* Genetic algorithms in search, optimization, and machine learning. – USA: Addison–Wesley publishing company, inc., 1989.
4. *Гладков Л.А., Курейчик В.В., Курейчик В.М., Сорокалетов П.В.* Бионспирированные методы в оптимизации. – М.: Физматлит, 2009. – 384 с.
5. *Родзин С.И.* Эволюционные стратегии: концепция и результаты // Перспективные информационные технологии и интеллектуальные системы. – 2002. – № 2. – С. 4-12. <http://www.pitis.tsure.ru/>.
6. *Курейчик В.М.* Об одной модели эволюции Шмальгаузена // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 7-16.

Курейчик Владимир Викторович

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: vkur@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634383451.

Родзин Сергей Иванович

E-mail: rodzin@mopevm.tsure.ru.

Тел.: 88634371673.

Kureichik Vladimir Viktorovich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University».

E-mail: vkur@tsure.ru.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +78634383451.

Rodzin Sergey Ivanovich

E-mail: rodzin@mopevm.tsure.ru.

Phone: +78634371673.

УДК 004.93

Р.А. Нейдорф, А.А. Деревянкина

**РЕШЕНИЕ ЗАДАЧ РАСПОЗНАВАНИЯ МЕТОДОМ РОЯЩИХСЯ ЧАСТИЦ
С ДЕЛЕНИЕМ РОЯ**

Показывается эффективность применения метода роящихся частиц, модифицированного авторами за счет дополнения его алгоритмом деления роя, к решению задач исследования многоэкстремальных зависимостей в таких постановках, которые требуют нахождения нескольких экстремумов одновременно.

Эвристические методы; многоэкстремальные задачи; распознавание образов, полутоновое изображение.

R.A. Neidorf, A.A. Derevyankina

**THE DECISION OF TASKS OF RECOGNITION BY THE METHOD
OF SWARMING PARTICLES WITH DIVISION OF THE PLENTY**

In the article efficiency of application of the modified method of swarming particles is shown. The method is added with algorithm of division of a plenty for the decision of research problems of multiextreme dependences.

Heuristic methods; multiextremum problem; pattern recognition; grayscale.