

УДК 004.8 : 004.056.5

**МОДИФИКАЦИИ АЛГОРИТМА ПРЫГАЮЩИХ ЛЯГУШЕК ДЛЯ ОТБОРА ПРИЗНАКОВ В НЕЧЕТКОМ КЛАССИФИКАТОРЕ ПРИ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ ПО РУКОПИСНОЙ ПОДПИСИ**

**Ходашинский Илья Александрович**

Д.т.н., профессор,

e-mail: [hodashn@rambler.ru](mailto:hodashn@rambler.ru),

**Бардамова Марина Борисовна**

Аспирантка,

e-mail: [722bmb@gmail.com](mailto:722bmb@gmail.com),

Томский государственный университет систем управления и радиоэлектроники (ТУСУР), 634050 г. Томск, пр. Ленина 40.

**Аннотация.** В работе проведено сравнение эффективности модификаций алгоритма прыгающих лягушек, позволяющих метаэвристике функционировать в бинарном пространстве поиска. Для задачи отбора признаков в нечетком классификаторе опробованы методы, основанные на модифицированных алгебраических операциях, функциях трансформации и операции слияния, а также их комбинации. В эксперименте использован набор данных SVC2004, содержащий большое количество признаков для аутентификации пользователя на основе динамических признаков рукописной подписи.

**Ключевые слова:** нечеткий классификатор, алгоритм прыгающих лягушек, метаэвристики, отбор признаков, бинаризация

**Цитирование:** Ходашинский И.А., Бардамова М. Б. Модификации алгоритма прыгающих лягушек для отбора признаков в нечетком классификаторе при аутентификации пользователя по рукописной подписи // Информационные и математические технологии в науке и управлении. 2020. № 4 (20). С.75-83. DOI: 10.38028/ESI.2020.20.4.007

**Введение.** Метаэвристические алгоритмы широко применяются для оптимизации в различных задачах, в том числе с целью отбора информативных подмножеств признаков при построении модели машинного обучения. Удаление лишних признаков помогает избежать переобучение модели и уменьшить её сложность. При использовании метаэвристик в качестве инструмента отбора признаков необходимым условием является способность алгоритма осуществлять поиск в бинарном пространстве. Однако в то время, как некоторые метаэвристики, например, генетический алгоритм, изначально были созданы для работы с бинарными входными векторами, другие были разработаны для работы в непрерывной области поиска. Примерами последних могут служить как алгоритм роящихся частиц, так и множество других метаэвристик. Очевидно, что один только генетический алгоритм не способен быть универсальным инструментом для любых данных, поскольку это противоречит теореме о бесплатных завтраках [13]. Поэтому важно иметь аналогичные альтернативы и разрабатывать модификации, способные трансформировать изначально непрерывный алгоритм в бинарный. Самыми распространенными методами бинаризации являются использование пороговых или трансформационных функций [2, 10, 12].

Алгоритм прыгающих лягушек (АПЛ) успешно применяется для оптимизации параметров классификатора [1, 3]. Существуют и примеры использования этой метаэвристики в бинарном пространстве [6, 12, 15]. В [12] АПЛ совмещается с алгоритмом роящихся частиц, откуда заимствует объемную процедуру вычисления скорости элементов входных векторов, и использует процедуры мутации и кроссовера из генетического алгоритма.

Авторы предлагают вычислительно простые способы бинаризации АПЛ, позволяющие обеспечить высокую скорость работы алгоритма. Все описанные методы будут применены к этой метаэвристике впервые. Проверка эффективности бинарного алгоритма при решении задачи отбора признаков в нечетком классификаторе будет проведена на наборе данных SVC2004. Этот набор содержит сигналы, описывающие динамические характеристики рукописной подписи с целью определения легитимности пользователя при прохождении процедуры аутентификации.

**1. Непрерывный алгоритм прыгающих лягушек.** Идея метаэвристики, разработанной в 2003 году, заключается в воспроизведении поведения группы лягушек в процессе поиска пищи [5]. В данной работе в качестве основы взята версия, описанная в [4]. На вход подаются вектора вида  $\mathbf{Z} = (z_1, z_2, \dots, z_{|Z|})$ ,  $z \in \mathbb{R}$ . В алгоритме комбинируются итерационно повторяющиеся внешний глобальный и внутренний локальный поиск. Глобальный поиск заключается в сортировке входных векторов по значению фитнес-функции и разбиением популяции на подгруппы. Внутри каждой подгруппы независимо осуществляется локальный поиск, в котором происходит обновление худших векторов.

Обновление векторов происходит согласно следующему принципу. Выбираются  $\mathbf{Z}_b$  и  $\mathbf{Z}_w$  – вектора с лучшим и худшим значением фитнес-функции в группе. Далее вычисляется промежуточный вектор  $\mathbf{Z}^*$ :

$$\mathbf{Z}^* = r \times c \times (\mathbf{Z}_b - \mathbf{Z}_w) + \mathbf{Z}_w, \quad (1)$$

где  $r$  – равномерно распределенное случайное число из промежутка от 0 до 1,  $c$  – коэффициент обновления вектора. Если значение фитнес-функции вектора  $\mathbf{Z}^*$  превосходит фитнес-функцию вектора  $\mathbf{Z}_w$ ,  $\mathbf{Z}^*$  замещает собой  $\mathbf{Z}_w$ . В противном случае вектор  $\mathbf{Z}^*$  вычисляется заново, но вместо  $\mathbf{Z}_b$  используется глобально лучший вектор  $\mathbf{Z}_0$ . Если и в данном случае не удастся улучшить вектор  $\mathbf{Z}_w$ , то он перезаписывается со случайно сгенерированным отклонением.

Алгоритм принимает на вход следующие параметры: число подгрупп  $G$ , число векторов в данной группе  $F$ , коэффициент для обновления входных векторов  $c$ , количество итераций для глобального и локального поиска  $T_{gl}$  и  $T_{lc}$  соответственно.

**2. Бинарный алгоритм прыгающих лягушек.** На вход бинарной версии метаэвристики подается популяция, состоящая из векторов вида  $\mathbf{S} = (s_1, s_2, \dots, s_n)$ . Каждый элемент вектора соответствует признаку из таблицы наблюдения,  $n$  – количество признаков. Значение элемента вектора  $s_i = 0$  показывает, что  $i$ -ый признак не участвует в классификации,  $s_i = 1$  означает, что  $i$ -ый признак используется классификатором. Алгоритм должен найти такой вектор  $\mathbf{S}$ , который позволил бы максимально улучшить фитнес-функцию при уменьшении количества признаков.

В фитнес-функции важно учитывать как качество построения модели классификации, так и долю отобранных признаков с целью уменьшения их количества:

$$Fit(\mathbf{S}) = \alpha \times Error(\mathbf{S}) + (1 - \alpha) \times \frac{n^*}{n}, \quad (2)$$

где  $Error(\mathbf{S})$  – ошибка классификации на векторе  $\mathbf{S}$ ,  $n^*$  – количество элементов в векторе  $\mathbf{S}$ , равных единице,  $\alpha$  – коэффициент приоритета одной части функции над другой.

**2.1. Метод модифицированных алгебраических операций** позволяет алгоритму оперировать бинарными входными векторами, так как все арифметические операторы заменяются на логические [8, 9]. В соответствии с этой идеей операции умножения, сложения и вычитания в (1) заменяются на конъюнкцию, дизъюнкцию и строую дизъюнкцию соответственно:

$$\mathbf{S}^* = r \wedge (\mathbf{S}_b \oplus \mathbf{S}_w) \vee \mathbf{S}_w, \quad (3)$$

где  $\mathbf{r}$  – случайный бинарный вектор.

**2.2. Метод слияния**, предполагающий использование оригинальной функции слияния для осуществления бинарного поиска, впервые был предложен в [7]. Функция поэлементно сравнивает два бинарных вектора; если значение элемента на одной и той же позиции совпадает, то в результирующем векторе на эту позицию запишется данное значение. В ином случае осуществляется генерация случайного числа из интервала от 0 до 1. Если оно меньше или равно 0,5, то в соответствующую позицию нового вектора записывается элемент из худшего вектора. В противном случае на этом месте будет выставлен элемент из лучшего вектора.

Таким образом, функция слияния может быть задана следующим образом:

$$\text{merge}(\mathbf{S}_w, \mathbf{S}_b) = \begin{cases} s_i^* = s_{wi} = s_{bi}, & \text{если } s_{wi} = s_{bi} \\ s_i^* = s_{wi}, & \text{если } s_{wi} \neq s_{bi} \text{ и } rand \leq 0,5, \\ s_i^* = s_{bi}, & \text{если } s_{wi} \neq s_{bi} \text{ и } rand > 0,5 \end{cases} \quad (4)$$

где  $rand$  – случайное равномерно распределенное число,  $rand \in [0;1]$ .

В алгоритме прыгающих лягушек описанную функцию удобно применять в локальном поиске вместо оператора (1). При первой генерации вектора  $\mathbf{S}^*$  в функцию передаются худший и локально лучший вектор, если  $\mathbf{S}^*$  не улучшает  $\mathbf{S}_w$ , производится повторная генерация  $\mathbf{S}^*$ . На повторной генерации вместо локального лидера используется глобальный.

### 2.3. Комбинация методов модифицированных арифметических операций и слияния

Данная модификация представляет собой совмещение оператора, использующего логические операции, и функции слияния двух векторов. Создание нового вектора на первой стадии локального поиска выглядит следующим образом:

$$\mathbf{S}^* = \text{merge}(r \wedge (\mathbf{S}_b \oplus \mathbf{S}_w), \mathbf{S}_b). \quad (5)$$

Как и во всех предыдущих случаях, на второй стадии локального поиска вместо  $\mathbf{S}_b$  применяется глобально лучший вектор  $\mathbf{S}_0$ .

**2.4. Функции трансформации.** Для использования функций трансформации необходимо наличие некоторого непрерывного вектора, каким-либо образом характеризующего признаки [11]. Чаще всего таким вектором служит вектор скорости. В оригинальном АПЛ такого вектора нет, но авторы предлагают рассчитывать в локальном поиске скорость худшего вектора  $\mathbf{S}_w$  следующим образом:

$$\mathbf{V} = (\mathbf{S}_0 - \mathbf{S}_w) \times r_1 + (\mathbf{S}_b - \mathbf{S}_w) \times r_2, \quad (6)$$

где  $r_1$  и  $r_2$  – вектора, заполненные случайными вещественными значениями в диапазоне от 0 до 1. Далее полученному значению необходимо поставить в соответствие бинарный эквивалент. Отображение происходит в два шага. На первом шаге функция трансформации, принимая на вход значение скорости элемента вектора  $\mathbf{S}_w$ , вычисляет некоторое число,

принадлежащее диапазону  $[0;1]$ . Второй шаг заключается в непосредственном обновлении элементов в соответствии с трансформационным правилом.

В этом случае локальный поиск сокращается до двух этапов. На первом происходит создание промежуточного вектора  $\mathbf{S}^*$  на базе  $\mathbf{S}_w$  с помощью функции трансформации и обновление  $\mathbf{S}_w$  в том случае, если  $\mathbf{S}^*$  обладает лучшей фитнес-функцией. В противном случае осуществляется второй этап, на котором на место  $\mathbf{S}_w$  записывается случайный вектор.

Далее описаны исследуемые авторами вариации функций и правил.

Выделяют несколько семейств функций трансформации. Два наиболее часто используемых семейства имеют S- и V-образные графики [10]. В качестве S-образной функции трансформации была использована базовая версия «сигмоиды»:

$$F_1(v_i) = 1/(1 + e^{-v_i}), \quad (7)$$

где  $v_i$  – значение скорости  $i$ -ого элемента.

V-образные функции обладают большим разнообразием вариаций, поэтому для исследования были выбраны две функции. Первая вычисляется с помощью гиперболического тангенса:

$$F_2(v_i) = |\tanh(v_i)|, \quad (8)$$

вторая задается следующим выражением:

$$F_3(v_i) = \left| v_i / \sqrt{1 + v_i^2} \right|. \quad (9)$$

Трансформационные правила отличаются принципом обновления элементов. В первом правиле  $R_1$  элементу строго присваивается бинарное значение:

$$\text{Если } rand < F(v_i), \text{ то } s_i = 1, \text{ иначе } s_i = 0, \quad (10)$$

где  $F(v_j)$  – одна из трех функций трансформации,  $rand$  – равномерно распределенное случайное число,  $rand \in [0;1]$ . Второе правило  $R_2$  либо заменяет элемент на противоположный, либо не меняет его:

$$\text{Если } rand < F(v_i), \text{ то } s_i = s_i \oplus 1. \quad (11)$$

**2.5. Комбинация функций трансформации и операции слияния.** В этом случае первый этап локального поиска заключается в применении функции трансформации на векторе  $\mathbf{S}_w$  и попытке замены худшего вектора на промежуточный  $\mathbf{S}^*$ . Если замена не происходит, то на втором этапе используется операция merge, осуществляющая слияние  $\mathbf{S}^*$  и вектора  $\mathbf{S}_0$ .

**3. Описание эксперимента.** Набор данных SVC2004 (Task 1) посвящен задаче аутентификации пользователя по динамическим характеристикам его рукописной подписи, таким, как скорость изменения координат, ускорение изменения координат, давление на поверхность планшета, длительность подписи и так далее [14]. Всего в SVC2004 содержатся 40 наборов различных подписей, соответствующих разным пользователям. Каждый набор включает в себя 20 вариаций подлинной подписи от её автора и 20 высококачественных подделок, как минимум от четырех злоумышленников. Каждая подпись – это набор сигналов, снятых с графического планшета. После проведения процедур предобработки (устранение разрывов, исправление различий в масштабе, позиции и наклоне подписи) и выделения признаков из сигналов, итоговая таблица наблюдений для каждой подписи состоит из 100

признаков и бинарной метки класса, показывающей принадлежность подписи легитимному пользователю или злоумышленнику.

Поскольку каждая подпись содержит всего 40 экземпляров данных, в эксперименте использована двукратная кросс-валидация. На каждой выборке алгоритм запускался 30 раз. Полученные показатели качества классификации были усреднены по количеству запусков и двум выборкам каждой подписи. Снимались следующие показатели: общая точность, ошибка I рода (False Rejection Rate, FRR), ошибка II рода (False Acceptance Rate, FAR) и количество признаков, оставшихся в наборе.

Структура классификатора генерировалась алгоритмом на основе экстремальных значений классов [7]. В качестве термов были выбраны треугольные функции принадлежности. Параметры алгоритма прыгающих лягушек следующие: количество групп равняется 4, в каждой из них по 10 векторов; локальных итераций 5, глобальных итераций 50. Коэффициент  $\alpha$  в фитнес-функции равнялся 0,5. В таблице 1 представлен перечень использованных в АПЛ способов бинаризации и их обозначения.

**Таблица 1.** Нумерация модификаций алгоритма прыгающих лягушек

Метод		Метод	
MAO	Модифицированные алгебраические операции (3)	F3R2	Функция трансформации $F_3$ (9) + правило $R_2$ (11)
merge	Операция слияния (4)	F1R1m	Функция трансформации $F_1$ (7) + правило $R_1$ (10) + операция слияния
MAOm	Модифицированные алгебраические операции + операция слияния (5)	F2R1m	Функция трансформации $F_2$ (8) + правило $R_1$ (10) + операция слияния
F1R1	Функция трансформации $F_1$ (7) + правило $R_1$ (10)	F3R1m	Функция трансформации $F_3$ (9) + правило $R_1$ (10) + операция слияния
F2R1	Функция трансформации $F_2$ (8) + правило $R_1$ (10)	F1R2m	Функция трансформации $F_1$ (7) + правило $R_2$ (11) + операция слияния
F3R1	Функция трансформации $F_3$ (9) + правило $R_1$ (10)	F2R2m	Функция трансформации $F_2$ (8) + правило $R_2$ (11) + операция слияния
F1R2	Функция трансформации $F_1$ (7) + правило $R_2$ (11)	F3R2m	Функция трансформации $F_3$ (9) + правило $R_2$ (11) + операция слияния
F2R2	Функция трансформации $F_2$ (8) + правило $R_2$ (11)		

**4. Результаты эксперимента.** В таблице 2 содержатся полученные показатели классификации на тестовых выборках, усредненные по всем 40 пользователям. Жирным шрифтом отмечены лучшие результаты по показателю.

**Таблица 2.** Результаты построения нечеткого классификатора на подмножествах признаков, отобранных бинарным алгоритмом прыгающих лягушек

Метод	Точность	FRR	FAR	Признаки
MAO	83,9 ± 7,5	25,1 ± 14,0	7,1 ± 8,1	43,9 ± 4,3
merge	86,7 ± 6,9	20,2 ± 13,0	6,3 ± 7,6	34,2 ± 3,6
MAOm	85,5 ± 7,7	21,1 ± 14,3	7,9 ± 9,2	20,5 ± 4,1
F1R1	85,2 ± 7,4	23,0 ± 13,3	6,5 ± 8,0	40,4 ± 4,1
F2R1	85,6 ± 7,3	21,9 ± 13,5	7,0 ± 8,2	28,0 ± 4,2
F3R1	86,1 ± 7,2	20,7 ± 13,7	7,0 ± 8,3	24,6 ± 4,3
F1R2	83,8 ± 7,6	25,4 ± 14,2	7,1 ± 8,2	43,5 ± 4,1
F2R2	87,0 ± 7,2	19,3 ± 13,3	6,8 ± 8,1	30,0 ± 4,0

F3R2	<b>87,1</b> ± 7,0	19,4 ± 13,1	6,4 ± 7,7	31,1 ± 4,0
F1R1m	85,3 ± 7,3	23,2 ± 13,5	<b>6,2</b> ± 7,8	39,1 ± 3,9
F2R1m	85,9 ± 7,5	20,1 ± 13,9	8,0 ± 9,3	19,7 ± 5,1
F3R1m	85,6 ± 7,7	20,1 ± 14,1	8,6 ± 9,6	<b>17,6</b> ± 4,6
F1R2m	84,2 ± 7,4	24,2 ± 13,6	7,5 ± 8,6	44,8 ± 4,1
F2R2m	<b>87,1</b> ± 7,0	<b>19,0</b> ± 12,9	6,9 ± 8,1	29,4 ± 4,0
F3R2m	86,7 ± 6,9	20,0 ± 13,0	6,7 ± 7,9	30,1 ± 3,9

Средние ранги, полученные при сравнении результатов по всем пользователям критерием Фридмана, представлены в таблице 3.

**Таблица 3.** Средние ранги по критерию Фридмана

Метод	Точность	FRR	FAR	Признаки
MAO	4,8	11,3	7,8	13,9
merge	10,9	6,2	6,1	10,0
MAOm	5,9	8,9	10,5	2,8
F1R1	8,1	9,2	6,4	11,9
F2R1	7,1	9,0	8,4	5,3
F3R1	8,0	8,0	8,1	4,0
F1R2	4,7	12,1	7,5	13,5
F2R2	11,0	<b>4,6</b>	7,3	7,4
F3R2	<b>11,6</b>	5,1	5,7	8,9
F1R1m	8,4	9,7	<b>5,3</b>	11,1
F2R1m	6,9	7,4	11,0	2,2
F3R1m	6,1	7,6	12,0	<b>1,0</b>
F1R2m	5,5	10,2	8,9	14,6
F2R2m	11,1	<b>4,6</b>	8,0	6,0
F3R2m	10,1	6,2	7,1	7,5

Лучшие результаты по общей точности продемонстрировали V-образные функции трансформации с правилом R<sub>2</sub> (11). При этом функция F<sub>2</sub> (8) показала лучший результат в комбинации с операцией слияния, а функция F<sub>3</sub> (9) – индивидуально. Ошибка первого рода оказалась наименьшей при использовании F<sub>2</sub> с правилом R<sub>2</sub> как при отсутствии, так и при наличии операции слияния. Наименьшая ошибка второго рода была получена с помощью F<sub>1</sub> (7) и правила R<sub>1</sub> (10) в комбинации с операцией слияния. Наилучшую способность к сокращению признаков продемонстрировала функция F<sub>3</sub> с правилом R<sub>1</sub> и операцией слияния.

**Заключение.** В работе исследована эффективность пятнадцати модификаций алгоритма прыгающих лягушек для осуществления поиска в бинарном пространстве на примере задачи отбора признаков. Во время эксперимента на наборе данных SVC2004 худшие результаты по совокупности рангов были получены комбинацией «сигмоидальной» фитнес-функции, первого трансформационного правила и операции слияния; этой же функцией, но со вторым правилом, а также методом модифицированных алгебраических операций. Лучшие совокупные результаты продемонстрировал алгоритм прыгающих лягушек при применении V-образных функций и второго трансформационного правила.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-37-90064.

#### СПИСОК ЛИТЕРАТУРЫ

1. Бардамова М.Б., Анфилофьев А.Е., Ковалев В.С., Филимоненко И.В. Сравнительный анализ эффективности метаэвристических алгоритмов при построении нечетких классификаторов // Международная школа-семинар «Интеллектуальные системы и технологии: современное состояние и перспективы»: труды. СПб.: Политехника-сервис. 2017. С. 22-31.
2. Al-Tashi Q., Abdulkadir S.J., Rais H.M., Mirjalili S., Alhussian H., Ragab M.G., Alqushaibi A. Binary Multi-Objective Grey Wolf Optimizer for Feature Selection in Classification // IEEE Access. 2020. vol. 8. Pp. 106247-106263. DOI: 10.1109/ACCESS.2020.3000040
3. Bardamova M. B. Construction of fuzzy classifiers with shuffle frog leaping algorithm for handwritten signature authentication // Сборник избранных статей научной сессии ТУСУР. Т. 1. № 1-2. Томск: В-Спектр. 2019. С. 229-232.
4. Elbeltagi E., Hegazy T., Grierson D. A modified shuffled frog-leaping optimization algorithm: applications to project management // Structure and Infrastructure Engineering. 2007. vol. 3. № 1. Pp. 53–60. DOI: 10.1080/15732470500254535
5. Eusuff M. M., Lansey K.E. Optimizing of water distribution network design using the shuffled frog leaping algorithm // Journal of Water Resources Planning and Management. 2003. vol. 129. № 3. Pp. 210-225.
6. Farsangi M., Barati M. Solving unit commitment problem by a binary shuffled frog leaping algorithm // IET Generation Transmission & Distribution. 2014. vol. 8. № 6. Pp. 1050-1060. DOI: 10.1049/iet-gtd.2013.0436
7. Hodashinsky I., Sarin K., Shelupanov A., Slezkin A. Feature selection based on swallow swarm optimization for fuzzy classification // Symmetry. 2019. vol. 11. № 11. Pp. 1423. DOI: 10.3390/sym11111423
8. Jia D., Duan X., Khan M. K. Binary artificial bee colony optimization using bitwise operation // Computers and Industrial Engineering. 2014. vol. 76. Pp. 360–365. DOI: 10.1016/j.cie.2014.08.016
9. Kiran M. S. Gunduz M. XOR-Based Artificial Bee Colony Algorithm for Binary Optimization // Turkish Journal of Electrical Engineering and Computer Sciences. 2013. vol. 21. 2307–2328. DOI: 10.3906/elk-1203-104
10. Mirjalili S., Lewis A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization // Swarm and Evolutionary Computation. 2013. vol. 9. Pp. 1–14. DOI: 10.1016/j.swevo.2012.09.002
11. Mirjalili S., Mohd Hashim S.Z. BMOA: binary magnetic optimization algorithm // International Journal of Machine Learning and Computing. 2012. vol. 2. № 3. Pp. 204–208. DOI: 10.7763/IJMLC.2012.V2.114
12. Vakil Baghmisheh M.T., Madani K., Navarbaф A. A discrete shuffled frog optimization algorithm // Artificial Intelligence Review. 2011. vol. 36. Pp. 267–284. DOI: 10.1007/s10462-011-9213-2
13. Wolpert D.H., Macready W.G. No Free Lunch Theorems for Optimization // IEEE Transactions on Evolutionary Computation. 1997. Vol.1. Pp. 67-82.

14. Yeung D.-Y., Chang H., Xiong Y., George S., Kashi R., Matsumoto T., Rigoll G.. SVC2004: First International Signature Verification Competition // Conference: Biometric Authentication, First International Conference. Proceedings. Lecture Notes in Computer Science. 2004. Pp. 16-22. DOI: 10.1007/978-3-540-25948-0\_3
  15. Zhang H., Liu Y.-A. Binary Shuffled Frog Leaping Algorithm in Population Diversity for Opportunistic Spectrum Assignment // Journal of Beijing University of Posts Ans Telecom. 2013. vol. 36. № 3. Pp. 92-96. DOI: 10.13190/jbupt.201303.96.zhanghg
- 

**UDK 004.8: 004.056.5**

## **FEATURE SELECTION FOR FUZZY CLASSIFIERS USING THE RANKING AND CROSS-VALIDATION**

**Ilya A. Hodashinsky**

Dr., Professor, e-mail: [hodashn@rambler.ru](mailto:hodashn@rambler.ru),

**Marina B. Bardamova**

Postgraduate student, e-mail: [722bmb@gmail.com](mailto:722bmb@gmail.com),

Tomsk State University of Control Systems and Radioelectronics,  
634050 Russia, Tomsk, Prospect Lenina 40

**Annotation.** The paper compares the effectiveness of modifications of the shuffled frog leaping algorithm that allow metaheuristics to function in the binary search space. Methods based on modified algebraic operations, transformation functions and fusion operations, as well as their combinations, have been tested for the task of selecting features in the fuzzy classifier. The SVC2004 data set containing a large number of features for user authentication based on dynamic handwritten signature characteristics was used in the experiment.

**Keywords:** fuzzy classifier, shuffled frog leaping algorithm, metaheuristics, feature selection, binarization

### **Reference**

1. Bardamova M. B., Anfilofiev A.E., Kovalev V.S., Filimonenko I.V. Sravnitelnyj analiz effektivnosti metaevristicheskikh algoritmov pri postroenii nechetkih klassifikatorov [Comparative analysis of the efficiency of metaheuristic algorithms for constructing fuzzy classifiers] // Mezhdunarodnaya shkola-seminar «Intellektualnye sistemy i tekhnologii: sovremennoe sostoyanie i perspektivy»: trudy = International school-seminar "Intelligent systems and technologies: current state and prospects". 2017. P. 22-31. (in Russian).
2. Al-Tashi Q., Abdulkadir S.J., Rais H.M., Mirjalili S., Alhussian H., Ragab M.G., Alqushaibi A. Binary Multi-Objective Grey Wolf Optimizer for Feature Selection in Classification // IEEE Access. 2020. vol. 8. Pp. 106247-106263. DOI: 10.1109/ACCESS.2020.3000040
3. Bardamova M. B. Construction of fuzzy classifiers with shuffle frog leaping algorithm for handwritten signature authentication // TUSUR Scientific Session: Collection of selected papers. 2019. vol. 1. № 1-2. C. 229-232.



4. Elbeltagi E., Hegazy T., Grierson D. A modified shuffled frog-leaping optimization algorithm: applications to project management // *Structure and Infrastructure Engineering*. 2007. vol. 3. № 1. Pp. 53–60. DOI: 10.1080/15732470500254535
5. Eusuff M. M., Lansey K.E. Optimizing of water distribution network design using the shuffled frog leaping algorithm // *Journal of Water Resources Planning and Management*. 2003. vol. 129. № 3. Pp. 210-225.
6. Farsangi M., Barati M. Solving unit commitment problem by a binary shuffled frog leaping algorithm // *IET Generation Transmission & Distribution*. 2014. vol. 8. № 6. Pp. 1050-1060. DOI: 10.1049/iet-gtd.2013.0436
7. Hodashinsky I., Sarin K., Shelupanov A., Slezkin A. Feature selection based on swallow swarm optimization for fuzzy classification // *Symmetry*. 2019. vol. 11. № 11. Pp. 1423. DOI: 10.3390/sym11111423
8. Jia D., Duan X., Khan M. K. Binary artificial bee colony optimization using bitwise operation // *Computers and Industrial Engineering*. 2014. vol. 76. Pp. 360–365. DOI:10.1016/j.cie.2014.08.016
9. Kiran M. S. Gunduz M. XOR-Based Artificial Bee Colony Algorithm for Binary Optimization // *Turkish Journal of Electrical Engineering and Computer Sciences*. 2013. vol. 21. 2307–2328. DOI: 10.3906/elk-1203-104
10. Mirjalili S., Lewis A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization // *Swarm and Evolutionary Computation*. 2013. vol. 9. Pp. 1–14. DOI: 10.1016/j.swevo.2012.09.002
11. Mirjalili S., Mohd Hashim S.Z. BMOA: binary magnetic optimization algorithm // *International Journal of Machine Learning and Computing*. 2012. vol. 2. № 3. Pp. 204–208. DOI: 10.7763/IJMLC.2012.V2.114
12. Vakil Baghmisheh M.T., Madani K., Navarbaф A. A discrete shuffled frog optimization algorithm // *Artificial Intelligence Review*. 2011. vol. 36. Pp. 267–284. DOI: 10.1007/s10462-011-9213-2
13. Wolpert D.H., Macready W.G. No Free Lunch Theorems for Optimization // *IEEE Transactions on Evolutionary Computation*. 1997. Vol.1. Pp. 67-82.
14. Yeung D.-Y., Chang H., Xiong Y., George S., Kashi R., Matsumoto T., Rigoll G.. SVC2004: First International Signature Verification Competition // *Conference: Biometric Authentication, First International Conference. Proceedings. Lecture Notes in Computer Science*. 2004. Pp. 16-22. DOI: 10.1007/978-3-540-25948-0\_3
15. Zhang H., Liu Y.-A. Binary Shuffled Frog Leaping Algorithm in Population Diversity for Opportunistic Spectrum Assignment // *Journal of Beijing University of Posts Ans Telecom*. 2013. vol. 36. № 3. Pp. 92-96. DOI: 10.13190/jbupt.201303.96.zhangh