Системи штучного інтелекту

УДК 004.8:004.032.26 DOI: 10.30748/soi.2018.152.20

Е.В. Бодянский, А.Ю. Шафроненко

Харьковский национальный университет радиоэлектроники, Харьков

РАНДОМИЗИРОВАННАЯ МОДИФИКАЦИЯ МЕТОДА ОПТИМИЗАЦИИ НА ОСНОВЕ КОШАЧЬИХ СТАЙ

В данной статье рассматривается задача оптимизации на основе кошачьих стай путем введения в процессы поиска и погони элементов случайного поиска в рандомизированной модификации базовой процедуры, которая позволяет повысить точность определения направления движения в режиме поиска и улучшить глобальные свойства процедуры в режиме погони. Предложенный метод оптимизации, являясь представителем эволюционных алгоритмов, предназначен для использования в гибридных системах вычислительного интеллекта и, прежде всего, в задачах обучения искусственных нейронных сетей, нейро-фаззи систем, а так же в задачах кластеризации и классификации.

Ключевые слова: *оптимизация, кошачьи стаи, режим поиска, режим погони.*

Введение

В настоящее время методы вычислительного интеллекта получили широкое распространение для решения множества сложных задач как чисто научных, так и в сфере техники, бизнеса, финансов, медицинской и технической диагностики, других областях, связанных с обработкой информации, включая, естественно, как традиционный интеллектуальный анализ данных (Data Mining), так и такие новые направления, как динамический анализ данных (Dynamic Data Mining), анализ потоков данных (Data Stream Mining), анализ больших массивов информации (Big Data Mining), Web-Mining, Text Mining и т.п. [1–6].

Одним из основных направлений вычислительного интеллекта являются эволюционные алгоритмы, представляющие собой по сути те или иные математические модели размножения или развития биологических организмов, инспирированные природой и предназначенные в самом общем случае для отыскания глобального оптимума многоэкстремальных функций в условиях неопределенности. Исторически первыми эволюционными алгоритмами явились так называемые генетические алгоритмы, в основе которых лежат механизмы селекции и генетики, реализующие выживание сильнейших особей в процессе эволюции.

Наиболее популярными эволюционными биоинспирированными алгоритмами на сегодняшний день являются, так называемые, "роевые" процедуры (Particle Swarm Optimization – PSO) [7], среди которых в качестве весьма перспективных как с позиции быстродействия, так и простоты реализации, являются алгоритмы оптимизации на основе кошачьих стай (Cat Swarm Optimization – CSO) [8; 9]. Эти алгоритмы подтвердили свою эффективность при решении ряда достаточно сложных задач и уже "успели" подвергнуться ряду модификаций, среди которых можно отметить процедуры, основанные на гармоническом поиске, дробных производных, адаптации параметров поиска и, наконец, "crazy cats" [10–16]. Вместе с тем эти процедуры не лишены и некоторых недостатков, которые ухудшают свойства процесса поиска глобального экстремума.

Целью работы является разработка быстродействующего и численно простого метода эволюционной оптимизации в условиях многоэкстремальных модифицированных функций.

Базовый алгоритм оптимизации на основе кошачьих стай

Для поиска глобального экстремума скалярной функции $f(x)$ векторного аргумента $x = (x_1, x_2, ..., x_n)^T \in R^n$ авторами [8–9] было предложено использовать модель поведения стай кошек (cat swarm-CS), при этом предполагается, что каждая кошка cat_p стаи, состоящей из Q особей $(p = 1, 2, ..., Q)$, может находиться в одном из двух состояний: режиме поиска (Seeking Mode – SM) и режиме погони (Tracing Mode – TM). При этом режим поиска связан с медленными движениями с незначительной амплитудой около исходной позиции (сканирование пространства в окрестности текущей позиции), а режим погони определяется быстрыми скачками с большой амплитудой и позволяет вывести кошку cat_p из локального экстремума,

если она туда попала. Сочетание локального сканирования и резких изменений текущего состояния позволяет с большей вероятностью отыскать глобальный экстремум по сравнению с традиционными методами многоэкстремальной оптимизации.

Процесс отыскания экстремума с помощью кошачьей стаи может быть реализован в виде следующей последовательности шагов:

Шаг CS 1: создать стаю из Q кошек в виде набора n-мерных векторов $x_p^{(0)}$, случайным образом распределенных на множестве допустимых значений аргументов R_x^n, т.е. $x_p^{(0)} \in R_x^n \subset R^n$; оценить значение оптимизируемой функции (фитнесс-функции) $f(x_p(0))$ во всех Q точках, при этом предполагается, что целью оптимизации является отыскание глобального минимума $f(x)$.

Шаг CS 2: ввести параметр состояния SPC (self position consideration), принимающий два значения 1 или 0; случайным образом разделить стаю на две группы: кошки в поиске (SPC=1) и кошки в погоне (SPC=0).

Шаг CS 3: если SPC=1, запустить соответствующую группу кошек в поиск, оставшихся кошек с SPC=0 запустить в режим погони.

Шаг CS 4: оценить значение фитнесс-функции и сохранить новые состояния $x_p(1)$, соответствующие наименьшим значениям $f(x_p(1))$.

Шаг CS 5: вернуться к шагу CS 1 с обновленной стаей $x_p(1), p = 1, 2, ..., Q$.

Режимы поиска и погони могут быть реализованы параллельно и также состоять из последовательности шагов. При этом режим поиска кошачьей стаи соответствует процессу локального поиска в задаче оптимизации. Режим поиска определяется тремя основными факторами: объемом памяти поиска (seeking memory pool – SMP), который определяет количество создаваемых копий каждой кошки cat_p, шагом изменения по каждой координате пространства R_x^n (seeking range of the selected dimension – SRD) и изменяемых координат (counts of dimension to change – CDC). Собственно, режим поиска может быть реализован в виде следующей последовательности шагов:

Шаг SM 1: если SPC = 1, создать C (C=SMP) копий cat_p.

Шаг SM 2: в соответствии с принятым CDC изменить состояние cat_p.

Шаг SM 3: оценить значения оптимизируемой фитнесс-функции для каждого измененного состояния cat_p.

Шаг SM 4: ввести вероятности выбора каждого изменяемого состояния

$$P_p = \frac{f(x_p(\tau)) - f_{min}(x_p(\tau))}{f_{max}(x_p(\tau)) - f_{min}(x_p(\tau))}, \tau = 1, 2, ..., T \quad (1)$$

и кошку с максимальным значением P_p исключить из дальнейшего рассмотрения. Кошка с $P_p = 0$ является «наилучшей» копией cat_p, поскольку ей соответствует наименьшее значение оптимизируемой функции $f_{min}(x_p(\tau))$.

Режим погони соответствует процессу глобального поиска, позволяющего «проскакивать» локальные экстремумы оптимизируемой функции, и также может быть реализован в виде последовательности шагов:

Шаг TM 1: если SPC = 0, для группы кошек в погоне рассчитать для каждой cat_p скорости движения по каждой координате с помощью рекуррентного выражения

$$\upsilon_{pi}(\tau+1) = \upsilon_{pi}(\tau) + r(\tau)\eta_{TM}(x_{best,i}(\tau) - x_{pi}(\tau)), \quad (2)$$

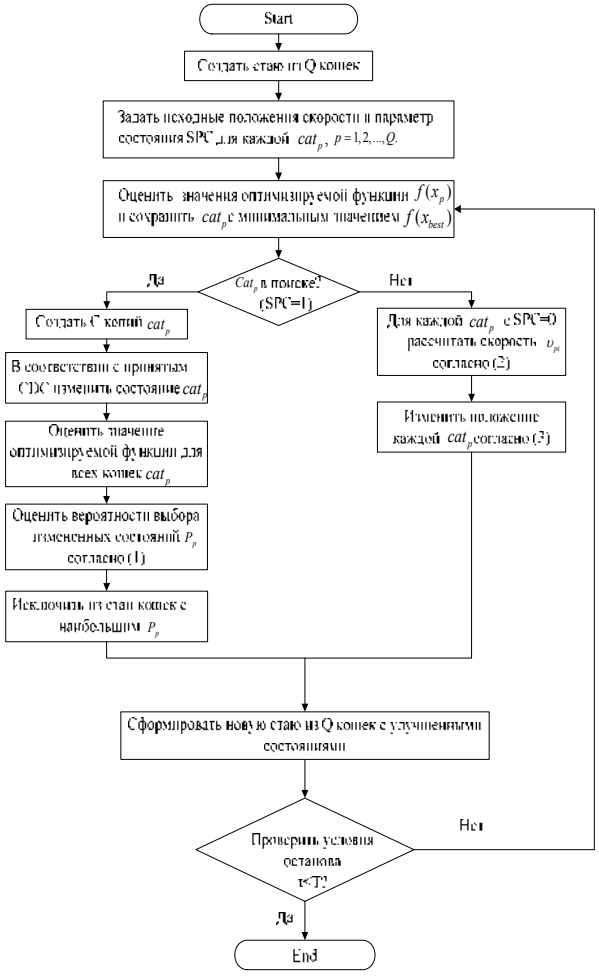

Рис. 1. Базовый алгоритм оптимизации на основе кошачьих стай

где $\upsilon_{pi}(\tau)$ – скорость движения p-й кошки по i-й координате на τ-й итерации погони; $0 < r(\tau) < 1$ – случайный параметр погони; η_{TM} – постоянный шаг погони; $x_{best,i}(\tau)$ – наилучшее решение задачи оптимизации, полученное на τ-й итерации.

Шаг ТМ 2: ввести предельно возможные значения скоростей υ_{min} и υ_{max}, для каждой кошки проверить условие

$$\upsilon_{min} < \upsilon_{pi}(\tau+1) < \upsilon_{max}$$

и если оно нарушается, положить $\upsilon_{pi}(\tau+1)$ равным соответствующему значению υ_{min} или υ_{max}.

Шаг ТМ 3: изменить положение каждой кошки в погоне согласно соотношению

$$x_{pi}(\tau+1) = x_{pi}(\tau) + \upsilon_{pi}(\tau). \qquad (3)$$

Шаг ТМ 4: проверить, принадлежит ли $x_p(\tau+1)$ R_{xi}^n.

В общем случае базовый алгоритм оптимизации на основе кошачьих стай может быть представлен в виде, приведенном на рис. 1.

Можно заметить, что рассмотренный алгоритм поиска реализует, по сути, покоординатный спуск (метод Гаусса - Зейделя), требующий многократного оценивания значений оптимизируемой функции и характеризующийся низкой скоростью сходимости. В режиме погони реализуется градиентный поиск с большим шагом, что в общем случае не гарантирует отыскание глобального экстремума.

В связи с этим представляется целесообразным модернизировать процедуру оптимизации на основе кошачьих стай путем ее рандомизации на основе случайного поиска, обладающего целым рядом преимуществ перед детерминированными процедурами поиска экстремума.

Рандомизированный алгоритм оптимизации на основе кошачьих стай

Поскольку режим поиска SM есть по сути процесс локальной оптимизации, движение каждой из кошек cat_p с SPC=1 целесообразно организовать в антиградиентном направлении согласно стандартной рекуррентной градиентной процедуре

$$x_p(\tau+1) = x_p(\tau) - \eta_{SM}\hat{\nabla}f(x_p(\tau)), \qquad (4)$$

где $\hat{\nabla}f(x_p(\tau))$ – оценки градиента оптимизируемой функции в точке $x_p(\tau)$;

η_{SM} – шаг поиска в пространстве R_x^n.

Составляющие градиента $\nabla f(x_p(\tau))$, являющиеся частными производными $\dfrac{\partial f(x_p(\tau))}{\partial x_p}$, могут быть оценены путем измерения оптимизируемой функции в пробных состояниях в окрестности точки $x_p(\tau)$. Наиболее простым с вычислительной точки зрения является поиск с центральной пробой, при этом производится оценка оптимизируемой функции в $(n+1)$-й точке $(CDC = n): x_p(\tau)$, $x_p(\tau) + \eta_{SRD}e_1$,

$$x_p(\tau) + \eta_{SRD}e_2, ..., x_p(\tau) + \eta_{SRD}e_n,$$

где e_i – координатные орты; η_{SRD} – величина пробного шага, определяемая принятым значением SRD.

Определив $n+1$ значение функции $f(x_p(\tau))$, $f(x_p(\tau) + \eta_{SRD}e_2), ..., f(x_p(\tau) + \eta_{SRD}e_n)$, вместо градиента

$$\nabla f(x_p(\tau)) = \left(\frac{\partial f(x_p(\tau))}{\partial x_{p1}}, \frac{\partial f(x_p(\tau))}{\partial x_{p2}}, ..., \frac{\partial f(x_p(\tau))}{\partial x_{pn}} \right)^T,$$

можно ввести его оценку $\hat{\nabla}f(x_p(\tau))$ с компонентами

$$\frac{\partial \hat{f}(x_p(\tau))}{\partial x_{pi}} = \frac{1}{\eta_{SRD}}\left(f(x_p(\tau)) + \eta_{SRD}e_i \right) - f(x_p(\tau)),$$
$$i = 1, 2, ..., n.$$

Реализовав далее шаг в пространстве R_x^n в соответствии с (4), приходим к новому состоянию cat_p в режиме поиска с координатами

$$\begin{cases} x_{p1}(\tau+1) = \\ = x_{p1}(\tau) - \dfrac{\eta_{SM}}{\eta_{SRD}}(f(x_p(\tau)) + \eta_{SRD}e_1) - f(x_p(\tau)), \\ x_{p2}(\tau+1) = \\ = x_{p2}(\tau) - \dfrac{\eta_{SM}}{\eta_{SRD}}(f(x_p(\tau)) + \eta_{SRD}e_2) - f(x_p(\tau)), \\ x_{pn}(\tau+1) = \\ = x_{pn}(\tau) - \dfrac{\eta_{SM}}{\eta_{SRD}}(f(x_p(\tau)) + \eta_{SRD}e_n) - f(x_p(\tau)). \end{cases} \qquad (5)$$

Можно заметить, что в случае $f(x_p(\tau+1)) < f(x_p(\tau))$, cat_p приближается к локальному минимуму, т.е. улучшает свое состояние и может далее оставаться в режиме поиска. Если же $f(x_p(\tau+1)) \geq f(x_p(\tau))$, cat_p находится в окрестности локального минимума, вывести из которого ее можно, переведя в режим погони.

В качестве недостатка этой процедуры оптимизации можно отметить фиксированное значение $CDC = n$, что требует поочередного изменения всех координат cat_p в пространстве R_x^n.

Расширить возможности процесса поиска можно, обратившись к рандомизированным процедурам, простейшей из которых является чисто случайная оценка направления спуска, смысл которого состоит в том, что из состояния $x_p(\tau)$ делается случайная проба $x_p(\tau)+\eta_{SRD}\Xi$, где $\Xi=(\xi_1,\xi_2,...,\xi_n)^T$ – единичный случайный вектор, равномерно распределенный в пространстве R_x^n.

В случае, если $x_p(\tau)+\eta_{SRD}\Xi < f(x_p(\tau))$, делается рабочий шаг поиска

$$x_p(\tau+1)=x_p(\tau)-\eta_{SM}\Xi \qquad (6)$$

(при этом можно принять $\eta_{SRD}=\eta_{SM}$), в противном случае проба признается неудачной и реализуется попытка с новым вектором Ξ.

Обобщением этой процедуры является оценка направления поиска по наилучшей из нескольких случайных проб. При этом, из исходного состояния $x_p(\tau)$ делается несколько случайных проб оптимизируемой функции $x_p(\tau)+\eta_{SRD}\Xi_l$ в случайных направлениях $\Xi_l(l=1,2,...,n,...,L)$, при том фактор CDC может превышать значение n. За направление спуска выбирается то направление Ξ^*, которое обеспечило наименьшее значение функции $f(x_p)$, т.е. cat_p переводится в новое состояние согласно выражению

$$x_p(\tau+1)=x_p(\tau)+\eta_{SRD}\Xi^*. \qquad (7)$$

Заметим также, что при $L=1$, процедуры (6) и (7) совпадают.

Объединяя процедуры поиска (4–5; 7), можно ввести в рассмотрение поиск на основе статического градиента. В этом случае за оценку градиента принимается средневзвешенное из L случайных направлений, каждое из которых берется с весом, соответствующим вариации $f(x_p)$ вдоль этого направления:

$$\hat{\nabla}f(x_p(\tau))=$$
$$=-\frac{\sum_{l=1}^{L}\Xi_l\left(f\left(x_p(\tau)+\eta_{SRD}\Xi_l\right)-\nabla f(x_p(\tau))\right)}{\left\|\sum_{l=1}^{L}\Xi_l\left(f\left(x_p(\tau)+\eta_{SRD}\Xi_l\right)-\nabla f(x_p(\tau))\right)\right\|}. \qquad (8)$$

Подставляя далее (8) в (7), получаем процедуру градиентного спуска в направлении минимума оптимизируемой функции.

Таким образом, все кошки с SPC=1 смещаются в направлении локальных минимумов оптимизируемой функции.

Режим погони TM в отличие от локального режима поиска SM обеспечивает общей процедуре оптимизации на основе CS глобальные свойства, позволяющие не застревать ей в локальных экстремумах. Понятно, что кроме процедуры (2–3) существуют и другие алгоритмы, обладающие требуемыми свойствами.

Одним из таких наиболее эффективных численно простых алгоритмов является метод тяжелого шарика [18], опирающийся на аналогию движения тяжелого тела по искривленной поверхности с учетом сил тяжести и трения. При этом в силу инерции шарик-кошка «проскакивает» локальные экстремумы, а в силу трения движение должно остановиться в глобальном экстремуме.

Данный алгоритм для кошек в режиме погони (SPC=0) может быть записан в виде

$$x_p(\tau+1)=x_p(\tau)-\alpha(x_p(\tau)-x_p(\tau-1))-$$
$$-\eta_{TM}\hat{\nabla}f(x_p(\tau)), \qquad (9)$$

где α – параметр, определяющий инерционные свойства процесса погони. При $\alpha=0$ (9) полностью совпадает с (4), отличаясь только шагом η_{SM}. При $\alpha=1$ процесс погони становится незатухающим, поэтому этот параметр выбирается в интервале $0<\alpha<1$, при этом, чем ближе α к единице, тем сильнее проявляются инерционные свойства, однако процесс слабо затухает в окрестности экстремума. В связи с этим целесообразно каждой кошке с SPC=0 назначить разные значения параметра α.

Заметим также, что в процедуру (9) может быть введена случайная компонента, вводящая дополнительное «рыскание» в процесс погони, улучшающее глобальные свойства алгоритма. При этом (9) модифицируется к виду

$$x_p(\tau+1)=x_p(\tau)-\alpha(x_p(\tau)-x_p(\tau-1))-$$
$$-\eta_{TM}\hat{\nabla}f(x_p(\tau))+\eta_{SRD}\Xi,$$

т.е. cat_p одновременно находится и в режиме погони и в режиме поиска-сканирования пространства R_x^n.

Выводы

Рассмотрена задача оптимизации на основе кошачьих стай и введена рандомизированная модификация базовой процедуры путем введения в процессы поиска и погони элементов случайного поис-

ка. Введенная модификация позволяет повысить точность определения направления движения в режиме поиска и улучшить глобальные свойства процедуры в режиме погони. Данная модификация проста в численной реализации, не требует формирования большой стаи и не использует в процессе поиска, так называемых, копий каждой кошки. Предложенный метод оптимизации являясь представителем эволюционных алгоритмов предназначен для использования в гибридных системах вычислительного интеллекта, и прежде всего в задачах обучения искусственных нейронных сетей, нейро-фаззи системах, а так же в задачах кластеризации и классификации.

Список литературы

1. Rutkowski, L. (2008), *Computational Intelligence Methods and Techniques*, Springer-Verlag, Berlin Heidelberg, 514 p.
2. Mumford, C. and Jain, L. (2009), *Computational Intelligence. Collaboration, Fusion and Emergence*, Springer-Verlag, Berlin Heidelberg, 729 p.
3. Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M. and Held, P. (2013), *Computational Intelligence. A Methodological Introduction*, Springer, Berlin, 488 p.
4. Kroll, A. (2013), *Computational Intelligence. Eine Einführung in Probleme, Methoden and technishe Anwendungen*, Oldenbourg Verlag, München, 428 p.
5. Kacprzyk, J. and Pedrycz, W. (2015), *Springer Handbook of Computational Intelligence*, Springer-Verlag, Berlin Heidelberg, 1634 p.
6. Pedrycz, W. and Shyi-Ming Chen (2015), *Infromation Granualarity, Big Data, and Computational Intelligence*, Cham, Springer, 444 p.
7. Kennedy, J. and Eberhart, R. (1995), Particle swarm optimization, *Proc. IEEE Int. Conf. on Neural Networks*, Vol. 4, pp. 1942-1948.
8. Chu, S.-C., Tsai, P.-W. and Pan, J.S. (2006), Cat swarm optimization, *Lecture Notes in Artificial Intellgence*,4099, Springer-Verlag, Berlin Heidelberg, pp. 854-858.
9. Chu, S.-C. and Tsai, P.-W. (2007), Computational Intelligence based on the behavior of cats, *Int. J. of Innovative Computing, Information, and Control*, 3, No.1, pp. 163-173.
10. Panda, G., Pranhan, P.M. and Majhi, B. (2011), Direct and inverse modeling of plants using cat swarm optimization, *Handbook of Swarm Intelligence*, ALO 8, Springer-Verlag, Berlin Heidelberg, pp. 469-485.
11. Orouskhani, M., Orouskhani, Y. and Teshnehlab, M. (2011), Average-inertia weighted cat swarm optimization, *Lecture Notes in Computing Science*, Springer-Verlag, Berlin-Heildelberg, pp. 321-328.
12. Orouskhani, M., Orouskhani, Y., Mansouri, M. and Teshnehlab, M. (2013), A novel cat swarm optimization algorithm for unconstrained optimization problems, *Int. J. Imformation Technology and Computer Science*, No. 11, pp. 32-41.
13. Lin, K.-C., Zhang, K.-Y. and Hung, J.C. (2014), Feature selection of support vector machine based on harmonious cat swarm optimization, *Proc. 7th Int. Conf. on "Ubi-Media Computing and Workshops",* pp. 205-208.
14. Zhang, Y. and Tian, Y. (2015), An improved cat swarm optimization algorithm and application research, *Proc. 7th Int. Conf. on "Advanced Computational Intelligence"*, Mount Wuyi, Fujian, China, pp. 207-211.
15. Sarangi, A., Sarangi, S.K., Mukherjee, M. and Panigrahi, S.P. (2015), System identification by crazy-cat swarm optimization, *Proc. Int. Conf. on "Microwave, Optical and Communication Engineering"*, Bhubaneswar, India, pp. 439-442.
16. Zhang, Y. (2016), Fractional – order cat swarm optimization, *Proc. 12th Int. Conf. on "Natural Computation, Fuzzy Systems and Knowledge Dicovery"*, pp. 191-197.

Поступила в редколлегию 6.12.2017
Одобрена к печати 20.02.2018

Відомості про авторів:

Бодянський Євгеній Володимирович
доктор технічних наук професор
науковий керівник Проблемної НДЛ АСУ
Харківського національного університету
радіоелектроніки,
Харків, Україна,
https://orcid.org/0000-0001-5418-2143
e-mail: yevgeniy.bodyanskiy@nure.ua

Information about the authors:

Yevgeniy Bodyanskiy
Doctor of Technical Sciences Professor
Scientific Head of Control Systems Research Laboratory
of Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine
https://orcid.org/0000-0001-5418-2143
e-mail: yevgeniy.bodyanskiy@nure.ua

Шафроненко Аліна Юріївна
кандидат технічних наук
Старший викладач кафедри Харківського національного університету радіоелектроніки,
Харків, Україна,
https://orcid.org/0000-0002-8040-0279
e-mail: alina.shafronenko@nure.ua

Alina Shafronenko
doctor of Philosophy,
Senior Lecturer of Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine;
https://orcid.org/0000-0002-8040-0279
e-mail: alina.shafronenko@nure.ua

РАНДОМІЗОВАНА МОДИФІКАЦІЯ МЕТОДУ ОПТИМІЗАЦІЇ НА ОСНОВІ КОТЯЧИХ ЗГРАЙ

Є.В. Бодянський, А.Ю. Шафроненко

В статті розглядається задача оптимізації на основі котячих зграй шляхом введення в процеси пошуку і погоні елементів випадкового пошуку в рандомізованій модифікації базової процедури, яка дозволяє підвищити точність визначення напрямку руху в режимі пошуку і поліпшити глобальні властивості процедури в режимі погоні. Запропонований метод оптимізації будучи представником еволюційних алгоритмів призначений для використання у гібридних системах обчислювального інтелекту і, перш за все в задачах навчання штучних нейронних мереж, нейро-фаззі систем, а також в задачах кластеризації та класифікації.

***Ключові слова**: оптимізація, котячі зграї, режим пошуку, режим погоні.*

RANDOMIZED MODIFICATION OF THE METHOD OPTIMIZATION BASED ON CAT SWARM

Ye. Bodyanskiy, A. Shafronenko

Novadays, the methods of computational intelligence are widely used to solve a problems related to information processing, including both traditional Data Mining and new directions like Dynamic Data Mining, Data Stream Mining, Big Data Mining, Web-Mining, Text Mining, etc.

One of the main areas of computational intelligence are evolutionary algorithms, which are represented in the form of global models of the reproduction or development of biological organisms, which are intended, in general, in order to find a global optimum of multiextremal functions under conditions of uncertainty.

The most popular evolutionary bioinspiral algorithms are the so-called Particle Swarm Optimization (PSO), among which the most promising both from the point of view of speed and ease of implementation are the optimization algorithms based on the Cat's Swarm (Cat Swarm Optimization - CSO).

In this paper, we consider the optimization problem based on modified Cats Swarm approach by introducing random search elements into the seeking and tracing processes that permits to improve the position of results and provide global properties to tracing mode. The proposed optimization method, being the representative of evolutionary algorithms, is intended for use in hybrid systems of computation intelligence and, above all, in the tasks of learning artificial neural networks, neuro-fuzzy systems, as well as in the tasks of clusterization and classification.

***Keywords**: optimization, cats swarm, tracing mode, seeking mode.*