The Fast Modification of Evolutionary Bioinspired Cat Swarm Optimization Method

A.Yu. Shafronenko , Ye. V. Bodyanskiy, I.P. Pliss
Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine;
e-mail: alina.shafronenko@nure.ua, yevgeniy.bodyanskiy@nure.ua, iryna.pliss@nure.ua

***Abstract*—This paper discusses the optimization problem based on cat swarm optimization by introducing elements of a random search in a stochastic modification of the basic procedure into the seeking and tracing modes, that improves the speed and accuracy of determining the direction of movement in the seeking mode and improves the global properties of the procedure in the tracing mode. The proposed optimization method, being a representative of evolutionary algorithms, is intended for using hybrid systems of computational intelligence and, above all, in learning tasks of artificial neural networks, neuro-fuzzy systems, as well as in clustering and classification problems.**

***Keywords*— Optimization; Cat swarm algorithm; Seeking mode; Tracing mode**

I. INTRODUCTION

Computational intelligence methods are widely used to solve many complex problems, both purely scientific, and in the field of engineering, business, finance, medical and technical diagnostics, and other areas related to information processing, including, of course, traditional intellectual analysis: Data Mining and such new directions as Dynamic Data Mining, Data Stream Mining, Big Data Mining, Web Mining, Text Mining, etc. [1-6].

One of the main areas of computational intelligence is evolutionary algorithms, which are essentially certain mathematical models of reproduction or development of biological organisms, inspired by nature and intended, in the most general case, to find the global optimum of multi-extremal functions under uncertainty. Historically, the first evolutionary algorithms were the so-called genetic algorithms, which are based on the selection and genetics mechanisms that implement the survival of the strongest individuals in the process of evolution.

The most popular evolutionary bioinspired algorithms for today are the so-called "swarm" procedures (Particle Swarm Optimization - PSO) [7], among which, as for the performance and simplicity of implementation is Cat Swarm Optimization (CSO) [8,9]. These algorithms have confirmed their effectiveness in solving a number of rather complex tasks and have already "managed" to undergo a number of modifications, among which are procedures based on harmonic search, fractional derivatives, adaptation of search parameters and, finally, "crazy cats" [10-16]. At the same time, these procedures are not deprived of some shortcomings that degrade the properties of the global extremum search process.

The aim of the paper is to develop a fast-acting and numerically simple method of evolutionary optimization under conditions of multi-extremal modified functions.

II. BASIC ALGORITHM OF OPTIMIZATION BASED ON CATS SWARM

To search for a global extremum of a scalar function $f(\mathrm{x})$ of vector argument $x=(\mathrm{x}_1,\mathrm{x}_2,...,\mathrm{x}_n)^T \in R^n$ authors [8,9] proposed to use a cat swarm-CS model of behavior, and it is assumed that each cat $\tilde{n}at_p$ swarm consisting of Q individuals $(p=1,2,...,Q)$, can be in one of two states: Seeking Mode (SM) and Tracing Mode (TM). In this case, the seeking mode is associated with slow movements with a slight amplitude near the initial position (space scanning in the vicinity of the current position), and the tracing mode that is determined by fast jumps with a large amplitude and allows the cat cat_p go put from local extremum, if she got there. The combination of local scanning and abrupt changes in the current state makes it more likely to find a global extremum compared to traditional multi-extrema optimization methods.

The process of finding an extremum using a cat swarm can be implemented as the following sequence of steps:

Step CS 1: Create a swarm of Q cats as a set of n-dimensional vectors $x_p^{(0)}$, randomly distributed over the sets of valid argument values R_x^n, $x_p^{(0)} \in R_x^n \subset R^n$; estimate the value of the function being optimized (fitness function) $f(x_p(0))$ in all Q points, it is assumed that the optimization goal is to find a global minimum $f(x)$.

Step CS 2: introduce the self position consideration (SPC) state parameter, that takes two values of 1 or 0; randomly divide the swarm into two groups: cats in seeking (SPC = 1) and cats in tracing (SPC = 0) modes.

Step CS 3: if SPC = 1, start the corresponding group of cats in the seeking mode, the remaining cats with SPC = 0 start in the tracing mode.

Step CS 4: Estimate the value of the fitness function and save new states $x(1)$, corresponding to the smallest values of $f(x_p(1))$.

Step CS 5: Return to Step CS 1 with the updated swarm $x_p(1), p=1,2,...,Q.$

Seeking and tracing modes can be implemented in parallel and also consist of a sequence of steps. In this case, the seeking mode cat swarms optimization corresponds to the local search process in the optimization problem. The seeking mode is determined by three main factors: the seeking memory pool (SMP), which determines the number of copies created for each cat cat_p, seeking range of the

978-1-7281-1814-7/19/31.00

selected dimension R_x^n (SRD), and counts of dimension to change (CDC). The seeking mode can be implemented as the following sequence of steps:

Step SM 1: if SPC = 1, create C (C = SMP) copies cat_p .

Step SM 2: change state cat_p according to the accepted CDC.

SM 3 step: estimate the values of the optimized fitness function for each change state cat_p .

SM 4 step: introduce the probabilities of selecting each changing state

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

and the cat with the maximal value P_p exclude from further consideration. Cat with $P_p = 0$ is the "best" copy cat_p, since it corresponds to the smallest value of the optimized function $f_{\min}(x_p(\tau))$.

The tracing mode corresponds to the global seekig process, which allows us to "skip" the local extrema of the function being optimized, and can also be implemented as a sequence of steps:

Step TM 1: if SPC = 0, for a group of cats in tracing, calculate for each cat_p the speed of movement for each coordinate using a recurrent expression

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

where $\upsilon_{pi}(\tau)$ - movement speed of p- th cat about i -th coordinate on τ -ith iteration trace, $0 < r(\tau) < 1$ - random tracing parameter, η_{TM} - constant tracing step, $x_{best,i}(\tau)$ - the best solution of the optimization problem obtained at the τ - ith iteration.

Step TM 2: introduce minimal and maximal possible speed values $\upsilon_{\min}$ and $\upsilon_{\max}$, for each cat check condition

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

and if it is broken, put $\upsilon_{pi}(\tau+1)$ equal to the corresponding value $\upsilon_{\min}$ or $\upsilon_{\max}$.

Step TM 3: change the position of each cat in tracing mode according to the relation

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

Step TM 4: check whether $x_p(\tau+1)$ to R_{xi}^n .

In the general case, the basic algorithm of optimization based on the cat swarm can be represented in the form shown in Fig.1.

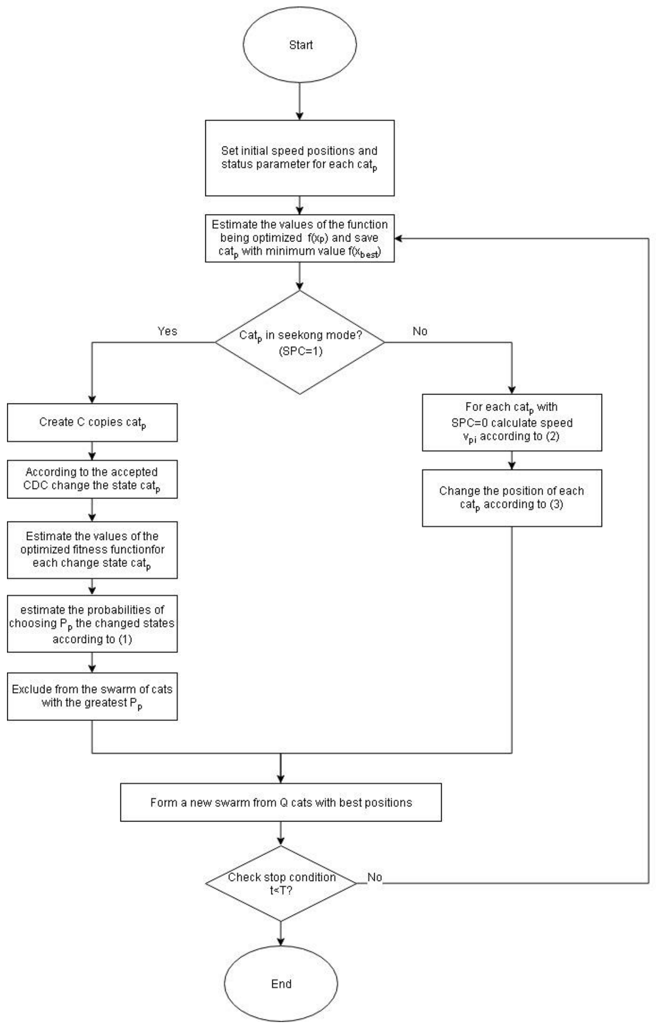

Fig.1. Basic Cat Swarm Optimization Algorithm

It can be noted that the considered search algorithm implements essentially coordinate wise descent (Gauss – Seidel method), that requires multiple estimations of the values of the function being optimized and is characterized by a low convergence rate. In the tracing mode, the gradient search is implemented with a large step, that generally does not guarantee finding a global extremum. In this regard, it seems advisable to modificate the optimization procedure based on cat swarm by randomizing it on the basis of a random search, that has a number of advantages over deterministic extremum search procedures.

III. RANDOMIZED CAT SWARM OPTIMIZATION ALGORITHM

Since the seeking mode (SM) is essentially a process of local minimization, the movement of each cat cat_p with SPC=1 it is advisable to organize in the antigradient direction according to the standard recurrent gradient procedure

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

where $\hat{\nabla} f(x_p(\tau))$ - estimates of the gradient of the function being optimized at the point $x_p(\tau)$, η_{SM} - seeking step in the space R_x^n .

Gradient components $\nabla f(x_p(\tau))$, that are partial derivatives $\frac{\partial f(x_p(\tau))}{\partial x_p}$, can be estimated by measuring the function to be optimized in the test conditions in the neighborhood of $x_p(\tau)$. From the computational point of view, the most simple is the search with a central point; in this case, the function being optimized is evaluated in $(n+1)$ - th point $(CDC=n): x_p(\tau)$, $x_p(\tau)+\eta_{SRD}e_1$, $x_p(\tau)+\eta_{SRD}e_2,...,x_p(\tau)+\eta_{SRD}e_n$, where e_i - coordinate orts, η_{SRD} - value of a test step, determined by received value SRD.

Defining $n+1$ function value $f(x_p(\tau))$, $f(x_p(\tau)+\eta_{SRD}e_2),...,f(x_p(\tau)+\eta_{SRD}e_n)$, instead of gradient

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

we can introduce its estimate $\hat{\nabla} f(x_p(\tau))$ with components

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

Implementing further step in space R_x^n in accordance with (4), we get to a new state cat_p in seeking mode with coordinates

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

Notice that in the case of $f(x_p(\tau+1))<f(x_p(\tau))$, cat_p approaches to the local minimum, i.e. improves its state and can continue to remain in seeking mode. If $f(x_p(\tau+1))\geq f(x_p(\tau))$, cat_p located in a neighborhood of a local minimum, from which it can be derived by transferring to tracing mode.

As a disadvantage of this optimization procedure, a fixed value CDC=n can be noted, which requires alternating changes in all coordinates cat_p in space R_x^n. Enhance the ability of the search process can be turned to a randomized procedure, the simplest of which is purely random rating descent direction, the meaning of which co-worth that of condition $x_p(\tau)$ it is a random sample $x_p(\tau)+\eta_{SRD}\Xi$, where $\Xi=(\xi_1,\xi_2,...,\xi_n)^T$ - single random vector, uniformly distributed in space R_x^n. In case if $x_p(\tau)+\eta_{SRD}\Xi<f(x_p(\tau))$, a search step is realized

$$x(\tau+1)=x(\tau)-\eta\,\Xi \quad (6)$$

(this can take $\eta_{SRD}=\eta_{SM}$), otherwise, the sample is considered unsuccessful and an attempt is made with a new vector Ξ.

The generalization of this procedure is to estimate the direction of the search for the best of several random samples. In this case, from the initial state $x_p(\tau)$ several random samples of the function being optimized $x_p(\tau)+\eta_{SRD}\Xi_l$ in random directions $\Xi_l(l=1,2,...,n,...,L)$, CDC factor may exceed the value of n. During the descent direction is the direction chosen Ξ^*, that provided the smallest value of the function $f(x_p)$, cat_p translated into the new state according to the expression

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

Note also that with $L=1$, the procedures (6) and (7) coincide.

Combining the search procedures (4), (5), (7), we can introduce into consideration a search based on a stochastic gradient. In this case, the weighted average of L random directions is taken as the gradient estimate, each of that is taken with a weight corresponding to the variation $f(x_p)$ along this direction:

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

Substituting further (8) into (7), we obtain the procedure of the gradient descent in the direction of the minimum of the optimized function. Thus, all cats with SPC = 1 are shifted in the direction of local minima of the optimized function.

The tracing mode (TM), in contrast to the local seeking mode (SM), provides for the general optimization procedure based on CS global properties, allowing it not to get stuck in local extremes. It is clear that in addition to the procedure (2), (3) there are other algorithms that have the required properties.

One of the most effective numerically simple algorithms is the “heavy ball” method [18], that is based on the analogy of the movement of a heavy body along a curved surface, taking into account the forces of gravity and friction. At the same time, due to inertia, the ball-cat “slips” local extremes, and due to friction, the movement must stop at a global extremum.

This algorithm for cats in tracing mode (SPC = 0) can be written as

$$x_p(\tau+1)=x_p(\tau)-\alpha(x_p(\tau)-x_p(\tau-1))-\eta_{\grave{O}\grave{I}}\,\hat{\nabla} f(x_p(\tau)) \quad (9)$$

where α - the parameter that determines the inertial properties of the tracing process. In $\alpha=0$ (9) completely coincides with (4), differing only in step η_{SM}. In $\alpha=1$ the tracing process becomes continuous, so this parameter is selected in the interval $0<\alpha<1$, in this case, the closer to unity, the more pronounced are the inertial properties, however, the process weakly decays in the vicinity of the extremum. In this regard, it is advisable for each cat with SPC = 0 to assign different values of the parameter α.

Note also that in procedure (9) a random component can be introduced, introducing an additional "yaw" into the chase process, which improves the global properties of the algorithm. In this case, (9) is modified to the form

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

i.e. cat_p at the same time is in the tracing mode and in the search-scanning space mode R_x^n.

Thus, the procedure (10) combines the seeking and tracing modes depending on the accepted values of the parameters α, η_{TM}, η_{SRD} can have the properties of gradient search, the method of "heavy ball" and global random search, that allows solving a wide class of multi-extremal optimization problems.

IV. Experimental Reseach

To compare the efficiency of the modified cat swarm optimization algorithm (CSO), the simulation is carried out on two different databases. Source data optimized using (PSO), Cat Swarm Optimization (CSO) and the new modified Randomized Cat Swarm Optimization (RCSO). The simulation parameters, presented in Table 1.

TABLE I. Parameters of randomized cat swarm optimization algorithm (RCSO)

Parameters	Value
SRD	Random [0,1]
Seeking memory Pool (SMP)	5
Population size	Number of clusters
r_1	Random in [0,1]
c_1	Const
SPC	Random in [0,1]
Number of iteration	Manually

TABLE II. Comparative results of perfomance indicators of algorithms such as PSO, CSO and RCSO for 1st database

MSE	PSO	CSO	RCSO
Best	4×10^{-7}	9×10^{-10}	8.4×10^{-11}
Median	7×10^{-6}	7.3×10^{-9}	5.2×10^{-10}
Worst	1.2×10^{-5}	9.3×10^{-9}	9.6×10^{-10}

TABLE III. Comparative results of perfomance indicators of algorithms such as PSO, CSO and RCSO for 2nd database

MSE	PSO	CSO	RCSO
Best	1.3×10^{-7}	8×10^{-10}	8.5×10^{-11}
Median	7×10^{-6}	4.4×10^{-9}	7.6×10^{-11}
Worst	2.02×10^{-5}	6.8×10^{-9}	7.8×10^{-10}

TABLE IV. Comparative results of time processing of algorithms

Data Set	PSO	CSO	RCSO
1st Database	6.9	24.3	23.4
2nd Database	14.5	43.5	42.9

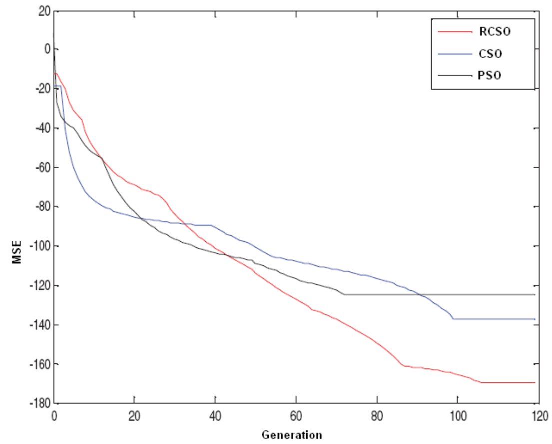

Fig. 2 – Results of performance indicators of RCSO, CSO and PSO for 1st database

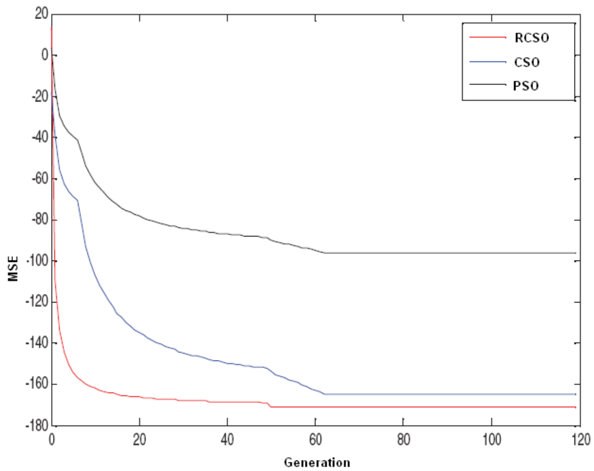

Fig. 3 – Result of performance indicators of RCSO, CSO and PSO for 2nd database

The table 5 shows comparative results of time processing of algorithms such as PSO, CSO and RCSO in seconds. This table demonstrate best result of the randomized cat swarm optimization algorithm same as CSO, but worse than the time PSO.

Conclusion

An optimization problem based on cat swarms is considered, and a randomized modification of the basic procedure is proposed by introducing random search elements into the seeking and tracing processes. The introduced modification allows to increase the accuracy of determining the direction of movement in the seeking mode and improve the global properties of the procedure in the tracing mode. This modification is simple in numerical implementation, does not require the formation of a large swarm and does not use in the search process, the so-called copies of each cat. The proposed optimization method is a representative of evolutionary algorithms intended for use in hybrid systems of computational intelligence, and above all in the tasks of artificial neural networks learning, neuro-fuzzy systems, as well as in problems of clustering and classification.

REFERENCES

[1] Rutkowski, L. (2008) "Computational Intelligence Methods and Techniques", Springer-Verlag, Berlin Heidelberg.

[2] Mumford, C., Jain, L. (2009) "Computational Intelligence. Collaboration, Fusion and Emergence", Springer-Verlag, Berlin Heidelberg.

[3] Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., 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.

[5] Kacprzyk, J., Pedrycz, W. (2015) "Springer Handbook of Computational Intelligence", Springer-Verlag, Berlin Heidelberg.

[6] "InfromationGranualarity, Big Data, and Computational Intelligence / Eds. W. Pedrycz W., Shyi-Ming Chen – Cham: Springer, 2015. – 444p.

[7] Kennedy, J., Eberhart, R. (1995) "Particle swarm optimization", Proc. IEEE Int. Conf. on Neural Networks, Vol. 4, P. 1942 – 1948.

[8] Chu, S.-C., Tsai, P.-W., Pan, J.S. (2006) "Cat swarm optimization", Lecture Notes in Artificial Intellgence,4099, Berlin Heidelberg, Springer-Verlag, P. 854-858.

[9] Chu, S.-C., Tsai, P.-W. (2007) "Computational Intelligence based on the behavior of cats", Int. J. of Innovative Computing, Information, and Control, 3, №1, pp. 163 – 173.

[10] Panda, G., Pranhan, P.M., Majhi, B. (2011) "Direct and inverse modeling of plants using cat swarm optimization", Eds. B.K. Panigrahi, Y. Shi, M-H. Zim "Handbook of Swarm Intelligence", ALO 8, Springer-Verlag, Berlin Heidelberg, pp. 469-485.

[11] Orouskhani, M., Orouskhani, Y., 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., Teshnehlab, M. (2013) "A novel cat swarm optimization algorithm for unconstrained optimization problems", Int. J. "Imformation Technology and Computer Science", 11, pp. 32 - 41.

[13] Lin, K. – C., Zhang, K.-Y., 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., 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., 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.