ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Improved Cat Swarm Optimization Approach Applied to Reliability-Redundancy Problem

Carlos Eduardo Klein[1]; Leandro dos S. Coelho[1,2,*];
Ângelo M. O. Sant'Anna[1], Roberto Z. Freire[1], Viviana C. Mariani[2]

1-Pontifical Catholic University of Parana (PUCPR) - Polytechnic School
Industrial and Systems Engineering Graduate Program (PPGEPS)
Rua Imaculada Conceição, 1555. Postal Code: 80215-901, Brazil.
2 – Federal University of Parana (UFPR) - Electrical Engineering Department

Abstract. System reliability-redundancy optimization plays a vital role in real-world applications. Recently, a new meta-heuristic based on swarm intelligence called cat swarm optimization (CSO) algorithm has emerged. CSO is a stochastic optimization paradigm inspired from the natural behavior of cats. To enhance the performance of the CSO algorithm, an improved adaptive CSO (ICSO) algorithm is presented. Both CSO and ICSO approaches were applied to an overspeed protection system for a gas turbine, a benchmark in the reliability-redundancy mixed-integer optimization field. Better results obtained by the ICSO show that the algorithm can be an efficient alternative for solving reliability problems.

1 Introduction

Reliability is one of relevant design measures in industry. A design engineer often tries to improve system reliability with a basic design, to the largest extent possible subject to several constraints such as cost, weight, and volume. In general terms, a reliability-redundancy optimization problem can be formulated to use components, and levels-of-redundancy to maximize some objective function, given system-level constraints on reliability, cost, and/or weight. During the past decades, numerous reliability design approaches based on optimization techniques [1-4] have been proposed.

Recently, many research activities have been devoted to the design of new metaheuristics [5,6]. The Cat Swarm Optimization (CSO) algorithm is a new metaheuristic approach based on swarm intelligence, introduced by Chu and Tsai in 2006 [7,8]. This optimization algorithm was inspired from inspecting the behavior of cats. The strong curiosity about moving objects and the outstanding hunting skill of the cat were modeled for CSO, they are called seeking mode and tracing mode.

This paper proposes an improved CSO (ICSO) based on adaptive tuning of the seeking-tracing rate (*TSRate*) value. This modification was conducted as an effort to produce a better performance with better accuracy level in optimization problems. In this study, the validity and efficiency of the proposed ICSO approach are illustrated with a reliability-redundancy optimization benchmark, an overspeed protection system for a gas turbine [9-11]. Finally, the results of the proposed ICSO approach

[*] National Council of Scientific and Technologic Development of Brazil - CNPq under Grants No. 307150/2012-7 (PQ) and 476235/2011-1 (Universal).

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

are compared with other optimization techniques presented in literature. Comparison results show that ICSO obtains a promising performance in the tested benchmark.

2 Background Information Optimization on Reliability-Redundancy

The reliability-redundancy allocation problem of maximizing the system reliability subject to constraints can be formulated as [12]

$$\text{maximize} \quad R_s = f(\mathbf{r}, \mathbf{n}), \tag{1}$$

subject to

$$\begin{gathered} g(\mathbf{r}, \mathbf{n}) \le l \\ 0 \le r_i \le 1, \quad r_i \in \Re, \ n_i \in Z^+, \ 1 \le i \le m, \end{gathered} \tag{2}$$

where R_s is the reliability of system, g, the set of constraint functions usually associated with system weight, volume and cost, $\mathbf{r} = (r_1, r_2, r_3, \ldots, r_m)$, the vector of the component reliabilities for the system, $\mathbf{n} = (n_1, n_2, n_3, \ldots, n_m)$, the vector of the redundancy allocation for the system (positive integer values), r_i and n_i the reliability and the number of components in the i-th subsystem, respectively, $f(\cdot)$, the objective function for the overall system reliability, l, the resource limitation, and m the number of subsystems. Our goal is to determine the number of components, and the components' reliability in each system, to maximize the overall system reliability. The problem belongs to the category of constrained nonlinear mixed-integer optimization problems.

2.1 Overspeed protection system for a gas turbine

The benchmark considered is an overspeed protection system for a gas turbine [9-11] illustrated in Fig. 1. Overspeed detection is continuously provided by the electrical and mechanical systems. When overspeed occurs, it is necessary to cut off the fuel supply using control valves [9]. For this purpose, 4 control valves (V_1–V_4) must close. The control system is modeled as a 4-stage series system.

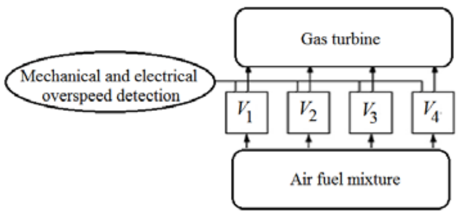

Fig. 1: Representation for the overspeed protection system of a gas turbine.

This problem is formulated as the following mixed-integer nonlinear programming problem [12]:

$$\text{maximize} \quad f(\mathbf{r}, \mathbf{n}) = \prod_{i=1}^{m} \left[1 - (1 - r_i)^{n_i}\right], \tag{3}$$

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

subject to

$$g_1(\boldsymbol{r},\boldsymbol{n}) = \sum_{i=1}^{m} v_i \cdot n_i^2 \le V \tag{4}$$

$$g_2(\boldsymbol{r},\boldsymbol{n}) = \sum_{i=1}^{m} C(r_i) \cdot \left[n_i + e^{0.25 \cdot n_i}\right] \le C \tag{5}$$

$$g_3(\boldsymbol{r},\boldsymbol{n}) = \sum_{i=1}^{m} w_i \cdot n_i \cdot e^{0.25 \cdot n_i} \le W \tag{6}$$

where $1 \le n_i \le 10$, $n_i \in Z^+$, where Z^+ is the space discrete of positive integers, $0.5 \le r_i \le 1\text{-}10^{-6}$, $r_i \in \Re$, v_i, the volume of each component in subsystem i, V, the upper limit on the sum of the subsystems' products of volume and weight, C, is the upper limit on the system cost, $C(r_i) = a_i \cdot [-T/\ln(r_i)]^{b_i}$ is the cost of each component with reliability r_i at subsystem I, T, the operating time during which the component must not fail, and W, the upper limit on the weight of the system. The input parameters of the overspeed protection system for a gas turbine are shown in Tab. 1.

Stage	$10^5 \cdot a_i$	b_i	v_i	w_i	V	C	W	T
1	1.0	1.5	1	6	250	400	500	1000 h
2	2.3	1.5	2	6				
3	0.3	1.5	3	8				
4	2.3	1.5	2	7				

Table 1: Data of overspeed protection system.

3 Optimization Algorithms

This section describes the proposed ICSO. First, the fundamentals of the CSO are briefly introduced, and finally the mechanisms of the proposed ICSO are provided.

3.1 Cat Swarm Optimization (CSO) algorithm

CSO is generated by observing the behavior of cats, and is composed of two sub-models by simulating the behavior of cats termed "seeking mode" and "tracing mode".

The seeking model is used to model the cat during a period of resting but being alert-looking around its environment for its next move. Seeking mode has four essential factors: seeking memory pool (*SMP*), seeking range of the selected dimension (*SRD*), counts of dimension to change (*CDC*), and the self position consideration (SPC). The tracing mode is the sub-model for modeling the case of the cat when tracing targets. Once a cat goes into tracing mode, it moves according to its' own velocities for every dimension. Every cat has its own position composed of D dimensions, velocities for each dimension, a fitness value representing the accommodation of the cat to the benchmark function, and a flag to identify whether the cat is in seeking mode or tracing mode. These two modes are dictated to join with each other by a mixture ratio MR. The final solution would be the best position of one of the cats. Both seeking and tracing steps can be found in [7,8,13].

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

3.2 Improved Cat Swarm Optimization (ICSO) algorithm

Classical CSO as many evolutionary and swarm intelligence approaches can suffer from the problems of premature convergence and stagnation [14-16]. The behavior of CSO is influenced by both seeking and tracing modes and by the control parameters values, where different parameter settings will lead to different performances.

According to the CSO algorithm, if *TSRate* = 0 seeking mode is configured, when *TSRate* = 1 all cats are on tracing mode. The proposed ICSO employs the tuning procedure of *TSRate* (maximization problem) mentioned in the sequence:

If $mean(f_{current}) > mean(f_{old})$ **then**
  $TSRate_{minimum} = 0.1$;
  $TSRate_{maximum} = 0.4$;
Else
  $TSRate_{minimum} = 0.5$;
  $TSRate_{maximum} = 0.9$;
End
$TSRate = TSRate_{minimum} + \mathbf{r}*(TSRate_{maximum} - TSRate_{minimum})$

In the procedure presented above, *r* is a random number generated with uniform distribution in range [0,1], $mean(f_{current})$ is the mean value of objective function cats' population in the current iteration (*it*), and $mean(f_{old})$ is the mean value of objective function of cats' population in the iteration (*it*-1). $TSRate_{minimum}$ and $TSRate_{maximum}$ are the minimum and maximum values of the *TSRate*, respectively. In this paper, the values of $TSRate_{minimum}$ and $TSRate_{maximum}$ were tuned by trial-and-error procedure.

4 Simulation results and analysis

Many reliability-redundancy optimization problems involve discrete variables, which are denoted by n_i that represents the number of components in subsystem *i*. Any n_i adjusted is a real number, and the most direct processing method is adopted here by transforming it into the nearest integer. In this work 30 independent runs and 30 different initial trial solutions were selected to each method. The parameters setting for both CSO and ICSO are the population size, 90, maximum iteration number (it_{max}), 150, *SMP*, was set to *N*/2, *SRD*, 0.5, CDC, 4, and *SPC*, 1. The same number of function evaluations (13,500) was adopted as stopping criterium. Several variants in terms of the *TSRate* values were investigated (Tab. 2).

CSO	*TSRate*	CSO	*TSRate*
(1)	0.1	(6)	0.6
(2)	0.2	(7)	0.7
(3)	0.3	(8)	0.8
(4)	0.4	(9)	0.9
(5)	0.5	(10)	[0.1, 0.9]

Table 2: CSO approach using different setups. CSO(10) represents that *TSRate* was randomly distributed on the whole interval [0.1, 0.9].

Constraints (4)-(6) in the overspeed protection system for a gas turbine are handled using a penalty strategy. In this work, the penalty-based method proposed in

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

[9] was used for both CSO and ICSO approaches for infeasible solutions (constraint violation). A penalty value was defined to consider the constrained violation based on the procedure illustrated in [9], where the terms l are subtracted (maximization problem) from objective function $f(\boldsymbol{r}, \boldsymbol{n})$ if $g(\boldsymbol{r}, \boldsymbol{n}) > 1$. In terms of best result $f(\boldsymbol{r}, \boldsymbol{n})$, the solutions of ICSO are just slightly better than the solution found by CSO(1)-CSO(9) for the overspeed protection system (Tab. 3). For the overspeed protection system (Tab. 4), ICSO has advantages in terms of solution quality (maximum value of $f(\boldsymbol{r}, \boldsymbol{n})$) when compared to the literature results [9-11].

5 Conclusion

In this paper, it was proposed an ICSO approach and a benchmark case has been carried out to show the feasibility of the proposed algorithm. From Tab. 3 and 4, it can be seen that the proposed ICSO is a promising technique for solving many reliability-redundancy optimization problems. Future work will consider extending the proposed ICSO to account for availability and multi-state systems in redundancy allocation problems. The proposed approach aims at identifying the whole Pareto-optimal solution set for multi-objective reliability-redundancy optimization cases.

References

[1] W. Kuo and V. R. Prasad, An annotated overview of system-reliability optimization, *IEEE Transactions on Reliability*, IEEE, 49(2):176-187, 2000.

[2] W. Kuo, V. R. Prasad, F. Tillman and C. L. Hwang, Optimization reliability design: fundamentals and applications. Cambridge: Cambridge University Press, 2001.

[3] W. Kuo and R. Wan, Recent advances in optimal reliability allocation, *IEEE Transactions on Systems, Man, and Cybernetics* - Part A: Systems and Humans, IEEE, 37(2):143-156, 2007.

[4] M. Gen and Y. S. Yun, Soft computing approach for reliability optimization: state-of-the-art survey, *Reliability Engineering and System Safety*, 91(9):1008-1026, 2006.

[5] M. Lozano and C. García-Martínez, Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report, *Computers & Operations Research*, 37(3):481-497, 2010.

[6] E. G. Talbi, Metaheuristics: From Design to Implementation, Wiley, 2009.

[7] S. C. Chu, P. W. Tsai and J. S. Pan, Cat swarm optimization, p*roceedings of the 9th Pacific Rim International Conference on Artificial Intelligence*, pages 854-858, Guilin, China, 2006.

[8] S. C. Chu and P.W. Tsai, Computational intelligence based on the behavior of cats, *International Journal of Innovative Computing, Information and Control*, 3(1):163-173, 2007.

[9] T. C. Chen, IAs based approach for reliability redundancy allocation problems, *Applied Mathematics and Computation*, 182(2):1556-1567, 2006.

[10] A. K. Dhingra, Optimal apportionment of reliability & redundancy in series systems under multiple objectives, *IEEE Transactions on Reliability*, IEEE, 41(4):576-582, 1992.

[11] T. Yokota, M. Gen and H. H. Li, Genetic algorithm for nonlinear mixed-integer programming problems and it's application, *Computers and Industrial Engineering*, 30(4):905-917, 1996.

[12] L. S. Coelho, D. L. A. Bernert and V. C. Mariani, A chaotic firefly algorithm applied to reliability-redundancy optimization, *proceedings of the IEEE Congress on Evolutionary Computation*, pages 517-521, New Orleans, USA, 2011.

[13] P. -W. Tsai, J. -S. Pan, S. -M. Chen and B. -Y. Liao, Enhanced parallel cat swarm optimization based on the Taguchi method, Expert Systems with Applications, 39(7):6309-6319, 2012.

[14] S. Das and P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, *IEEE Transactions on Evolutionary Computation*, IEEE, 15:4-31, 2011.

[15] R. Thangaraj, M. Pant, A. Abraham and P. Bouvry, Particle swarm optimization: Hybridization perspectives and experimental illustrations, *Applied Mathematics and Computation*, 217(12):5208-5226, 2011.

[16] T. T. Nguyen, S. Yang and J. Branke, Evolutionary dynamic optimization: a survey of the state of the art, *Swarm and Evolutionary Computation*, 6:1-24, 2012.

TSRate	Optimization method	$f(\mathbf{r}, \mathbf{n})$			
		Minimum (Worst)	Mean	Maximum (Best)	Standard Deviation
0.1	CSO(1)	0.969784	0.990187	0.999327	7.49e-03
0.2	CSO(2)	0.980470	0.992840	0.999851	5.60e-03
0.3	CSO(3)	0.970672	0.993755	0.999658	6.49e-03
0.4	CSO(4)	0.981049	0.994355	0.999846	5.05e-03
0.5	CSO(5)	0.978006	0.994733	0.999737	5.11e-03
0.6	CSO(6)	0.978905	0.994234	0.999532	5.02e-03
0.7	CSO(7)	0.980207	0.993846	0.999870	6.23e-03
0.8	CSO(8)	0.987490	0.995421	0.999775	3.95e-03
0.9	CSO(9)	0.985095	0.995760	0.999495	3.83e-03
[0.1, 0.9]	CSO(10)	0.986826	0.995228	0.999716	3.56e-03
Algorithm Section 3.2	ICSO	0.997602	0.998991	0.999954	8.05e-04

Table 3: Convergence results of f(r, n) (30 runs) for the overspeed protection system using both CSO and ICSO approaches.

Parameter	Dhingra [10]	Yokota et al. [11]	Chen [9]	This work (ICSO)
$f(\mathbf{r}, \mathbf{n})$	0.99961	0.999468	0.999942	0.999954
n_1	6	3	5	5
n_2	6	6	5	5
n_3	3	3	5	4
n_5	5	5	5	6
r_1	0.81604	0.965593	0.903800	0.901654
r_2	0.80309	0.760592	0.874992	0.888218
r_3	0.98364	0.972646	0.919898	0.948074
r_4	0.80373	0.804660	0.890609	0.849962
MPI (%)	88.6333%	91.6673%	23.5689%	-
Slack (g_1)	65	92	50	55
Slack (g_2)	0.064	-70.733576	0.002152	0.009347
Slack (g_3)	4.348	127.583189	28.803701	15.363463

Note: Slack is the unused resources. $MPI(\%) = [R_S(\text{ICSO}) - R_S(\text{other})]/[1 - R_S(\text{other})]$

Table 4: Comparison of result for the overspeed protection system using ICSO considering the results available in the literature.