KU Battle of Metaheuristic Optimization Algorithms 1: Development of Six New/Improved Algorithms

Joong Hoon Kim, Young Hwan Choi, Thi Thuy Ngo, Jiho Choi, Ho Min Lee, Yeon Moon Choo, Eui Hoon Lee, Do Guen Yoo, Ali Sadollah and Donghwi Jung

Abstract Each of six members of hydrosystem laboratory in Korea University (KU) invented either a new metaheuristic optimization algorithm or an improved version of some optimization methods as a class project for the fall semester 2014. The objective of the project was to help students understand the characteristics of metaheuristic optimization algorithms and invent an algorithm themselves focusing those regarding convergence, diversification, and intensification. Six newly developed/improved metaheuristic algorithms are Cancer Treatment Algorithm (CTA), Extraordinary Particle Swarm Optimization (EPSO), Improved Cluster HS (ICHS), Multi-Layered HS (MLHS), Sheep Shepherding Algorithm (SSA), and Vision Correction Algorithm (VCA). This paper describes the details of the six developed/improved algorithms. In a follow-up companion paper, the six algorithms are demonstrated and compared through well-known benchmark functions and a real-life engineering problem.

Keywords Cancer treatment algorithm · Extraordinary particle swarm optimization · Improved cluster HS · Multi-layered HS · Sheep shepherding algorithm · Vision correction algorithm

J.H. Kim(✉) · Y.H. Choi · T.T. Ngo · J. Choi · H.M. Lee · Y.M. Choo · E.H. Lee
School of Civil, Environmental and Architectural Engineering, Korea University, Seoul 136-713, South Korea
e-mail: {jaykim,younghwan87,y999k,dlgh86}@korea.ac.kr, tide4586@yahoo.com, {chooyean,hydrohydro}@naver.com

D.G. Yoo · A. Sadollah · D. Jung
Research Center for Disaster Prevention Science and Technology, Korea University, Seoul 136-713, South Korea
e-mail: godqhr425@naver.com, sadollah@korea.ac.kr, donghwiku@gmail.com

J.H. Kim and Z.W. Geem (eds.), *Harmony Search Algorithm*,
Advances in Intelligent Systems and Computing 382,
DOI: 10.1007/978-3-662-47926-1_19

1 Introduction

Each of six members of hydrosystem laboratory in Korea University (KU) invented either a new metaheuristic optimization algorithm or an improved version of Harmony Search (HS) [1,2] or ParticleSwarm Optimization (PSO) [3] as a class project for the fall semester 2014. The objective of the project was to help students understand the characteristics of metaheuristic optimization algorithms and invent an algorithm by themselves considering convergence, diversification, and intensification. Two new algorithms are Cancer Treatment Algorithm (CTA) and Vision Correction Algorithm and four algorithms are an improved version of existing algorithms: Extraordinary PSO (EPSO) and Sheep Shepherding Algorithm (SSA) are the variants of PSO, while Improved Cluster HS (ICHS) and Multi-Layered HS (MLHS) are the variants of HS. Through performance tests using well-know benchmark functions, the algorithms have been refined to enhance their global and local search ability and to minimize the number of the required parameters. This paper describes the details of the six optimization algorithms.

2 Six Metaheuristic Algorithms

Six new/improved metaheuristic optimization algorithms are CTA, EPSO, ICHS, MLHS, SSA, and VCA. ICHS and MLHS are the variants of HS while EPSO and SSA are those of PSO. CTA and VCA are newly developed in this study. The following subsections describe the details of the new/improved metaheuristic algorithms.

2.1 Cancer Treatment Algorithm (CTA)

CTA mimics the medical procedure of cancer treatment. Cancer is the result of cells that uncontrollably grow and do not die. If a person is diagnosed cancer from medical examinations such as magnetic resonance imaging and computed tomography, a type of treatment/surgery is then decided based on the tumor size and potential of transition. Generally, if the size is less than 2 cm, heat treatment is adopted. On the other hand, if the tumor is large and expected to have high potential of transition to another organ, it is removed by surgery (cancer removal surgery). This generic medical examination, treatment/surgery, and relevant decision processes help cure the disease (the best state), which is similar to the optimization process seeking the global optimum.

CTA, a population-based algorithm, generates new solutions for next generation based on three operators. By operation 1, k new solutions (out of total n new population) are generated considering the high fitness solutions stored in the memory called K (the memory size is K). While a single new solution is generated by operation 3, the remaining solutions (n - (k + 1)) of the next population are generated based on operation 2. Then, the combined $2n$ population (parent and children) is sorted according to fitness. The top n solutions become the parent population for the next generation and the top k solutions are stored in the memory K. This procedure is continued until stopping criteria are met. The following equations describe how new solutions are generated by three operations.

Operation 1:

$$x_{t+1}^{i,j} = x_t^{i,j} + rnd[-1,1] \times I^{-2\alpha} \tag{1}$$

where $x_{t+1}^{i,j}$ is the ith component of the jth new solution ($j \in$ the memory K) in the (t+1)th generation, I is laser beam intensity and $I = \frac{\sum_{j=1}^{k} f_j}{k}$ where f_i is fitness value of the solution j in the memory K, and α is reduction factor and $\alpha = exp\,(\frac{f_j^{n-1}}{f_j^{n-2}})$.

Operation 2:

$$x_{t+1}^{i} = x_t^{i} \quad if\ i \notin n_{remove} \tag{2}$$

$$x_{t+1}^{i} = x_t^{i} \times \gamma \quad if\ i \in n_{remove} \tag{3}$$

where n_{remove} is the set of solutions that will be generated by operation 2, and γ is revival rate and $1 - \text{abs}(\frac{f_{min}}{f_i^{t-1}})$ where f_{min} is the smallest fitness in the population and f_i^{t-1} is the ith solution's fitness in the (t-1)th generation.

Operation 3:

$$x_{t+1}^{i} = x_t^{i} \quad if\ rnd > R\ or\ t = 1 \tag{4}$$

$$x_{t+1}^{i} = x_t^{i} + rnd[-1,1] \times (x_t^{i} - x_{t-1}^{i}) \quad if\ rnd \leq R \tag{5}$$

where R is research rate.

2.2 Extraordinary Particle Swarm Optimization (EPSO)

PSO simulates the movement behaviors of animal swarm. Particles in a swarm have a tendency to share information on the current nearest location to food, which makes all particles move toward the current best particle. In addition, particle's past best location is also considered to determine the next location. However, the aforementioned strategies can result in being entrapped in local optimum because the search diversity can be limited.

Therefore, EPSO employed a different strategy. While high fitness particles have more probability to be selected as a target particle, each particle chooses a target particle which can be any particle in a swarm.

In EPSO, initial N_{pop} particles are randomly generated as original PSO. Each particle's location at the iteration t+1, $X_i(t+1)$, is determined either by adding particle velocity to the current location or by randomly as

$$X_i(t+1) = \begin{cases} X_i(t) + V_i(t+1) & if\ T \in (0, T_{up}) \\ LB_i + rand \times (UB_i - LB_i) & otherwise \end{cases} \tag{6}$$

where $V(t+1)$ is particle velocity for the component i and $C \times (X_{Ti}(t) - X_i(t))$ where C is movement parameter and $X_{Ti}(t)$ is the selected target particle's component i at the iteration t, T is a random integer generated in [0, N_{pop}], T_{up} is selection parameter and round($\alpha \times N_{pop}$) where α is target range parameter, and LB_i and UB_i are the lower and upper bound of the decision variable i, respectively.

2.3 Improved Cluster Harmony Search (ICHS)

HS was inspired by the musical ensemble [1,2]. HS implements the harmony enhancement process and the sets of "good harmony" are saved to a solution space termed harmony memory (HM), which is a unique feature of HS compared to other evolutionary optimization algorithms.

HS generates new solutions either by random generation or by considering the good solutions in HM. For each component x_i of a new solution vector $\underline{x}$, one of the stored values of the component in HM is selected with harmony memory considering rate (HMCR) or a random value within the allowed range is chosen with the probability of 1-HMCR. Each stored value in HM is equally probable to be selected. After HM consideration, HS scans each decision variable in the new solution and changes it to the neighborhood values with pitch adjusting rate (PAR). If the new solution is better than the worst solution in HM, the latter is replaced with the former. The above process continues until stopping criteria are satisfied.

ICHS is a variant of HS. ICHS differs from other HS variants in the HM consideration. That is, ICHS assigns high probability to be selected for the component of the high fitness solutions. Note that in standard HS (SHS) the stored component values of the solutions in HM have the same probability to be selected regardless of the solutions' fitness. Other mechanisms such as pitch adjusting are same as SHS.

First, the solutions in HM are sorted according to fitness. Then, they are divided into subgroups by k-Means clustering. k-Means clustering partitions the solutions into k distinct clusters based on a Euclidean distance to the centroid of a cluster. A probability value of the solution j (P_{ri}) is calculated as

$$P_{rj} = e^{-\frac{\alpha f_j}{f_{max}}} \tag{7}$$

where α is selection pressure, f_j is fitness value of the solution j, and f_{max} is the maximum fitness value in HM.

Then, P_{rj} is normalized to have the cumulative probability of 1 as

$$P_{rni} = \frac{P_{rj}}{\sum_{j=1}^{nm} P_{rj}} \tag{8}$$

where P_{rnj} is the normalized probability of the solution j, and nm is the HM size.

Therefore, the kth cluster is selected with the probability $\sum_{j \in Cluster_k} P_{rnj}$ where $Cluster_k$ is the set of solutions classified as the kth cluster. In ICHS, any stored values of the component in the selected cluster can be selected with equal probability.

2.4 Multi-Layered Harmony Search (MLHS)

MLHS is another variant of HS. The special feature of MLHS is that HMCR and PAR operations are conducted based on multi-layered HMs. A layer is in the form of grid in which each cell has a sub-memory that stores m solutions (see Fig. 1.). There exists a hierarchical tournament between the two neighboring layers in which the best solutions in the lower layer are elevated to fill the sub-memories in the upper layer.

MLHS begins by initializing sub-memories in the first (bottom) layer (the largest grid in Fig. 1). Initial solutions are randomly generated same as SHS. For example, total l^2m solution are generated for the first layer with l by l grid and the sub-memory size of m (8^2 x 4 = 256 where l = 8 and m = 4 as shown in Fig. 1). In the considered example in Fig. 1, the solutions (3, 4), (2,6), (6,2), and (9,6) are stored at the lower left corner cell of the grid in the first layer. Then, a new solution is generated for each sub-memory either by random generation or harmony memory consideration and pitch adjusting, as for SHS. If the new solution is better than the worst solution in sub-memory, the latter is replaced with the former.

The sub-memories of the second layer are filled with the best solutions from the sub-memories of the neighborhood cells in the first (lower) layer. For example, the best solutions from the four cells at the lower left corner of the first layer (the cells in the green box in Fig. 1), (2,6), (5,6), (4,4), and (5,4), are provided to the sub-memory of the lower left corner of the second layer. Similarly, a new solution is generated based on the SHS rules. This tournament between the two neighboring layers continues until the top layer where the grid dimension is 1 by 1.

In addition to the unique hierarchical structure, MLHS adopted dynamic parameters where HMCR and band width are changed over iterations.

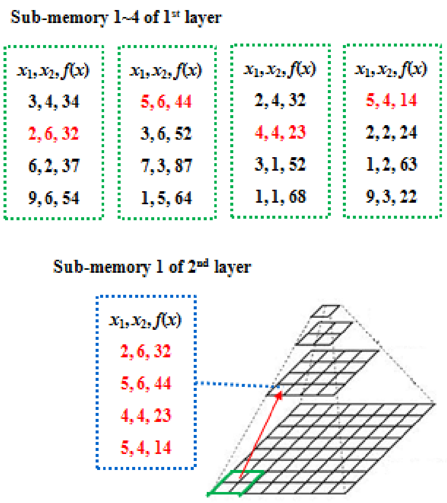

Fig. 1 Hierarchical tournament in MLHS

2.5 *Sheep Shepherding Algorithm (SSA)*

SSA is inspired by sheep shepherding which consists of two main operators: sheep gathering and shepherd herding (or guarding). SSA is similar with PSO in considering animal's flocking nature in the optimization. However, SSA adopts the shepherd operation which is intended to perturb worst solutions to enhance the probability of escaping from local optimum and finding global optimum. This unique feature of SSA and mimics the behavior of a shepherd which encourages the sheep which falls behind.

In SSA, a sheep represents a solution and initial sheep are randomly generated. In each iteration, a sheep can change its position or stay with the probability d and 1-d, respectively. If a sheep is determined to change its location, the component i of the jth sheep at the tth iteration $VecH_t(i,j)$ is determined as follows:

$$VecH_t(i,j) = H \times VecH_{t-1}(i,j) + C \times VecC_t(i,j) + P_s \times VecR_{s,t}(i,j) \tag{9}$$

where $VecC_t(i,j)$ is gathering vector and $CM_t(i) - sheep_t(i,j)$ where $CM_t(i)$ is the ith component of the center of sheep flock at tth iteration and *sheep* represents the location of the sheep, $VecR_{s,t}(i,j)$ is herding vector and $sheep_t(i,j) - VecS_t(i)$ where $VecS_t(i)$ is the ith component of shepherd, and H, C, and P_s are user defined parameters.

If a sheep is determined to change its location in the next iteration, the new location is calculated as

$$sheep_t(i,j) = sheep_{t-1}(i,j) + VecH_t(i,j) \tag{10}$$

If a sheep stays at its previous location, minor movement is added to the previous position as

$$sheep_t(i,j) = (1 \pm m \times rnd) \times sheep_{t-1}(i,j) \tag{11}$$

where m is movement parameter.

Finally, the component i of shepherd $VecS(i)$ is updated to keep perturbing the worst perform sheep.

$$VecS_t(i) = \begin{cases} VecS_{t-1}(i) + sheep_t(i,N) & if\ sheep(i,N) > H \\ VecS_{t-1}(i) + L & otherwise \end{cases} \tag{12}$$

where $sheep_t(i,N)$ is the component i of the worst sheep (N^{th} sheep after sorting sheep), H is sheep criterion parameter, and L is a random term. The above updates of sheep and shepherd locations continues until stopping criteria are met.

2.6 Vision Correction Algorithm (VCA)

VCA simulates vision correction process. The repetitive processes for making a good lens (global optimum) are similar to optimization process. In VCA, a lens is a solution candidate. First, initial lens are generated within the allowed ranges and their fitness are calculated. For each iteration, a real random number *r* is generated and compared with the division rate $dr1$. If r is less than $dr1$, a new solution is generated randomly. Otherwise, roulette wheel selection is performed to select a solution from population. After the selection, only the last three operations (described later) are performed without processing myopia or hyperopia operations. A randomly generated solution is refined either by myopia or hyperopia operations. Myopia correction is carried out to treat near-sightedness (positive direction search) with the probability *dr*2. Hyperopia correction is performed to fix far-sightedness (negative direction search) with the probability 1-*dr*2. Note that the division rates *dr*1 and *dr*2 are dynamic variables which change over iterations.

Myopia correction increases the focal length of lens and performs search in positive direction as given follows:

$$x(t,i) = x(t-1,i) + rnd \times (UB_i - x(t-1,i)) \tag{13}$$

On the other hand, hyperopia correction decreases the focal length of lens and performs negative direction search given in the following equation:

$$x(t,i) = LB_i + rnd \times (x(t-1,i) - LB_i) \tag{14}$$

Regardless of whether the above two operators are processed, the last three operators are conducted: brightness increase, compression, and astigmatism correction. Note that the frequency of processing each of the last three operators is controlled by a predefined probability. The brightness of lens is measured by modulation transfer function (MTF).

$$MTF = \sqrt{\sum_{i=1}^{n} ang_i} \tag{15}$$

where $ang_i = \sqrt{\frac{dx_i}{\sqrt{\sum_{i=1}^{n} dx_i^2}}}$ where dx_i is Euclidean distance in the i^{th} axis (dimension) from the current best solution.

Lens is compressed to decrease its thickness. Compression factor (*CF*) is considered to control convergence speed. The bright increase and compression operation are embedded in the following equation:

$$x(t,i) = x(t,i) \times \left\{1 + MTF \times rnd[-1,1] \times (1 - \frac{t}{T})^{CF}\right\} \tag{16}$$

where *T* is maximum number of iteration allowed.

Finally, astigmatic correction is performed.

$$x(t,i) = x(t,i) \times \{1 + rnd[-1,1] \times (\sin\theta)^2\} \tag{17}$$

where θ is axial parameter.. The whole operations continue until stopping criteria are met.

3 Summary

This paper introduced six new/improved metaheuristic optimization algorithms. CTA mimics generic cancer treatment process which consists of three differentoperators. All three operators are conducted to produce children population. On the other hand, VCA,inspired by vision correction procedures ,consists of myopia and hyperopia correction, brightness increase, compression,etc. The frequency of each operation is controlled by division rates. ICHS and MLHS are the variants of SHS. ICHS increases selection pressure for the solutions classified as high fitness clusters, while MLHS adopts hierarchical tournament for filling the HMs of the cells in the upper layers. EPSO and SSA are considered as the variants of PSO.

Table 1 indicates the number of parameters used in the six algorithms. The number of parameters is smallest in EPSO, while that is largest in MLHS. In a follow-up companion paper, the six algorithms are demonstrated and compared through well-known benchmark functions and a real-world engineering problem.

Table 1 Number of parameters (population size is included as a user parameter, while number of maximum iteration is excluded)

CTA	EPSO	ICHS	MLHS	SSA	VCA
5	3	6	7	6	7

Acknowledgement This work is supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (NRF- 2013R1A2A1A01013886).

References

1. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**(2), 60–68 (2001)
2. Kim, J.H., Geem, Z.W., Kim, E.S.: Parameter Estimation Of The Nonlinear Muskingum Model Using Harmony Search. JAWRA **37**(5), 1131–1138 (2001)
3. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948. IEEE Service Center, Piscataway (1995)