Research Article

Smallest-Small-World Cellular Harmony Search for Optimization of Unconstrained Benchmark Problems

Sung Soo Im,[1] Do Guen Yoo,[2] and Joong Hoon Kim[3]

[1] *Daelim Industrial Co. Ltd., Seoul 110-140, Republic of Korea*
[2] *Research Institute of Engineering and Technology, Korea University, Seoul 136-713, Republic of Korea*
[3] *School of Civil, Environmental and Architectural Engineering, Korea University, Seoul 136-713, Republic of Korea*

Correspondence should be addressed to Joong Hoon Kim; jaykim@korea.ac.kr

We presented a new hybrid method that combines cellular harmony search algorithms with the Smallest-Small-World theory. A harmony search (HS) algorithm is based on musical performance processes that occur when a musician searches for a better state of harmony. Harmony search has successfully been applied to a wide variety of practical optimization problems. Most of the previous researches have sought to improve the performance of the HS algorithm by changing the pitch adjusting rate and harmony memory considering rate. However, there has been a lack of studies to improve the performance of the algorithm by the formation of population structures. Therefore, we proposed an improved HS algorithm that uses the cellular automata formation and the topological structure of Smallest-Small-World network. The improved HS algorithm has a high clustering coefficient and a short characteristic path length, having good exploration and exploitation efficiencies. Nine benchmark functions were applied to evaluate the performance of the proposed algorithm. Unlike the existing improved HS algorithm, the proposed algorithm is expected to have improved algorithmic efficiency from the formation of the population structure.

1. Introduction

Studies on network maps provide in-depth understanding of the basic features and requirements of various systems. Many network connection topologies, assumed to be either completely regular or completely random, have been studied [1, 2]. Networks, which can be formally described by the tools of graph theory, are critical for describing many scientific, social, and technological phenomena (Newman [1]). Typical examples include the Internet, World Wide Web, social acquaintances, electric power networks, and neural networks. In recent years, new theoretical and applied results have motivated substantial researches in network science. The pioneering studies of Watts and Strogatz [2] have been performed, and they have been followed by many others in the subsequent years. They constructed a simple computer model of a regular network or lattice, in which each node of the network was connected by a line or edge to each of its four nearest neighbors. This network structure or topology is highly clustered or cliquish by design; however,

movement from one node to another node on the opposite side of the lattice requires traversing a large number of short-range connections. In other words, although the path length (or number of mediating edges) between neighboring nodes is short, the path length between distant nodes is long. Therefore, the minimum path length averaged over all possible pairs of nodes in the network which is quite long. Watts and Strogatz [2] investigated the change in network topology (measured in terms of local clustering and minimum path length) that resulted from randomly rewiring some of the lattice edges to create long-range connections between distant nodes. If many lattice edges were randomly rewired, the network would naturally acquire the topological characteristics of a random graph (short path length and low clustering). It is worth mentioning that they found that even a few long-range connections greatly reduced the minimum path length of the network without affecting its local clustering. Thus, they defined algorithmically for the first time a class of networks having topological properties similar to social networks and demonstrated both the high
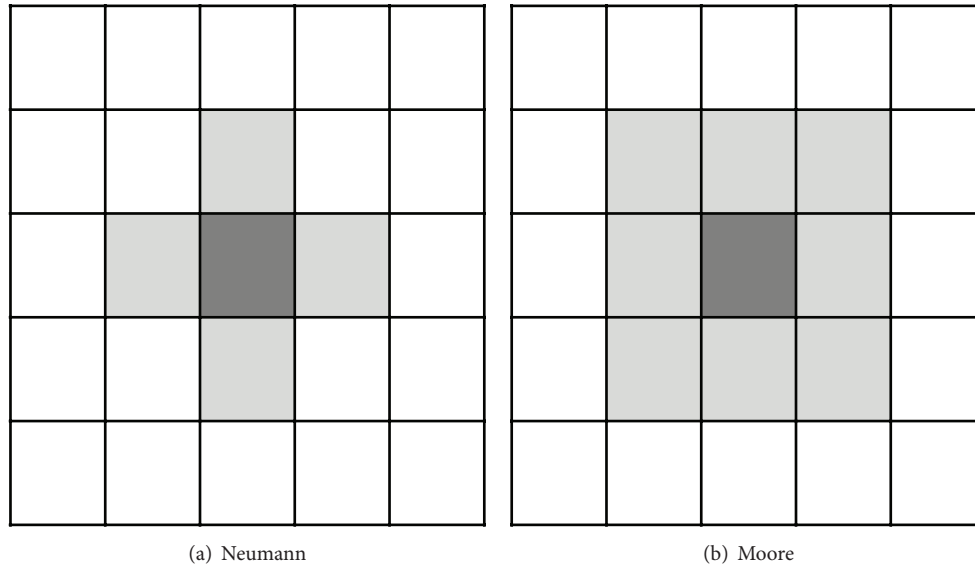
(a) Neumann

(b) Moore

FIGURE 1: Neighboring cells.

clustering of a lattice and the short path length of a random graph. These networks were called Small-World networks. On the other hand, Nishikawa et al. [3] suggested a Smallest-Small-World network (SSWN) and confirmed their theory with examples. They showed that the average path length of a Small-World network with fixed shortcuts became the minimum when there was a "center" node, from which all shortcuts are connected to uniformly distributed nodes in the network [4]. Research on efficient network structures can be applied to fields of optimization, Genetic Algorithms (GAs), Artificial Neural Networks (ANNs), and Particle Swarm Optimization (PSO). In GAs, cellular Genetic Algorithms (cGAs) have been widely studied using cellular automata (CA). Kang [5] particularly proposed Smallest-Small-World cellular Genetic Algorithms (SSWcGAs) to apply the SSWN to the cGAs and evaluated the performance of the algorithm.

The harmony search (HS) algorithm [6, 7] is based on musical performance processes that occur when a musician searches for a better state of harmony. The HS has successfully been applied to many mathematical functions and a wide variety of practical optimization problems like pipe-network design, structural optimization, vehicle routing, combined heat and power economic dispatching, multiple-dam-system scheduling, and so forth. In addition, hybrid HS algorithms that combine with other optimization techniques such as the PSO and ANNs have been proposed.

So far, researches have been carried out to improve the performance of the HS algorithm by changing the parameters such as pitch adjusting rate (PAR) and harmony memory considering rate (HMCR). However, there has been a lack of studies on the performance improvement of the HS through the formation of population structures to transform it into a cGA. In this study, therefore, we proposed the improved HS algorithm, which is used the CA and has the topological structure of the SSWN. The improved HS algorithm has a high clustering coefficient and a short characteristic path length possessing good exploration and exploitation efficiencies.

2. Related Theory

2.1. Cellular Automata. A cellular automaton is a discrete model studied in computation theory, mathematics, physics, complexity science, theoretical biology, and microstructure modeling [8]. Von Neumann [9] has used the concept of cellular automaton space regularly arranged in a grid cell. Individual cells updated simultaneously in a discrete time step. Each cell is a finite state machine. Each cell entered the state of its own and neighboring and, then, printout the state in the next time step. A collection of these cellular automatons is called CA space. Other terms for the CA are "cellular space," "tessellation automata," "homogeneous structures," "cellular structures," "tessellation structures," "iterative arrays," and so on [8].

The cell's dimension in the CA can be defined as one-, two-, and three-dimensional cells. Typically, a 2D cell is the most widely used because it can deal with spatial phenomena. Talking about the shape of the cell, a square cell is usually used because of its highest computer-processing efficiency, compared to the other types of cells. Neighbors in the CA can be defined depending on the form of the adjacent surrounding cells. In the form of a 2D square cell CA, it can be defined as a Von Neumann neighborhood with neighbors in four directions or as a Moore neighborhood having neighbors in eight directions as shown in Figure 1.

The concept of the CA, as described in this subsection, has been applied to a variety of optimization techniques such as GAs [10, 11], ANNs [12], and PSO [13, 14]. For instance, cGA is a subclass of GA in which each individual is placed in a given topological distribution. It is a lattice graph, as one individual can only interact with its nearest
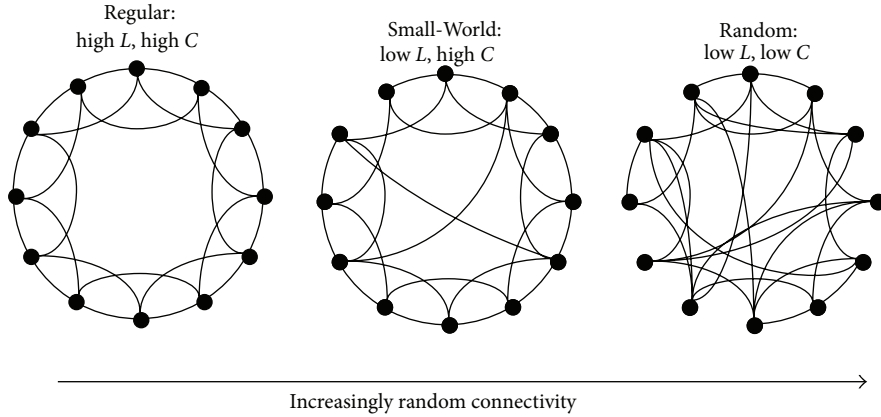
FIGURE 2: Classification of networks according to connectivity.

TABLE 1: Efficiency of average path length (L) and clustering coefficient (C).

Property	Description
Average path length (L)	Exploitation (local search) Low $L \Rightarrow$ good exploitation efficiency
Clustering coefficient (C)	Exploration (Global Search) High $C \Rightarrow$ good exploration efficiency

neighbors. Therefore, these relations can be applied on a set of each individual and the surrounding neighbors, promoting neighborhood exploitation and exploration of the search space.

2.2. Smallest-Small-World Theory. The Small-World network models have received much attention since their introduction by Watts and Strogatz [2] as models of real networks having characteristics which lie between random and regular as shown in Figure 2. They are characterized by two factors: the average path length (L) which measures efficiency of communication or passage time between nodes and the clustering coefficient (C) which represents the degree of local order. The average path length (L) is defined as the average number of links in the shortest path between a pair of nodes in the network. Accordingly, clustering coefficient (C) is defined as the probability that two nodes connected to a common node are also connected to each other. In general, cGA's population space is a regular network that has relatively high L and C from the point of view of the network theory. High L makes the interactions of remote nodes difficult. Otherwise, a Small-World network represents low L between any two arbitrary nodes and contains high C. It can be characterized by lots of local connectivity between nodes as well as the occasional longer links, defined as a shortcut. Only a few of these longer links are required in order to obtain a Small-World network state.

Nishikawa et al. [3] suggested a SSWN and verified their theory with examples. They showed that the L of a Small-World network with fixed shortcuts is a minimum when there
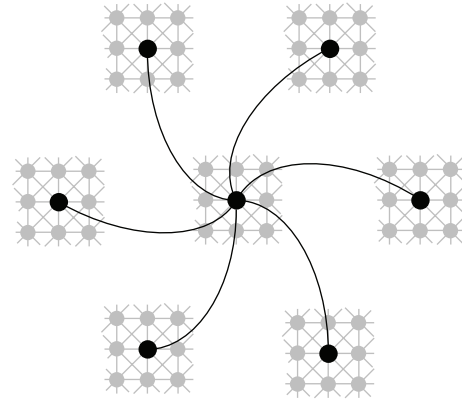

FIGURE 3: Examples of shortcut configuration with a center node.

is a "center" node, from which all shortcuts are connected to uniformly distributed nodes in the network [4]. An example of such a configuration is illustrated in Figure 3.

They defined the SSWN theory as follows.

(i) A SSWN is composed of two parts: the underlying network (e.g., a regular lattice) and the subnetwork of shortcuts containing only the shortcuts and their nodes.

(ii) The nodes in the subnetwork of shortcuts must be uniformly distributed over the network.

(iii) Finally, among all possible configurations of connected subnetworks of shortcuts with uniformly distributed nodes, the ones with a single center involve the largest number of nodes.

These arguments indicate that, given a fixed number of shortcuts, the networks connected with a sub-network of shortcuts having uniformly distributed nodes have smaller L than a typical random configuration, and, among these, the ones with a single center minimize L. In other words, the Smallest-Small-World networks are characterized by these structures.
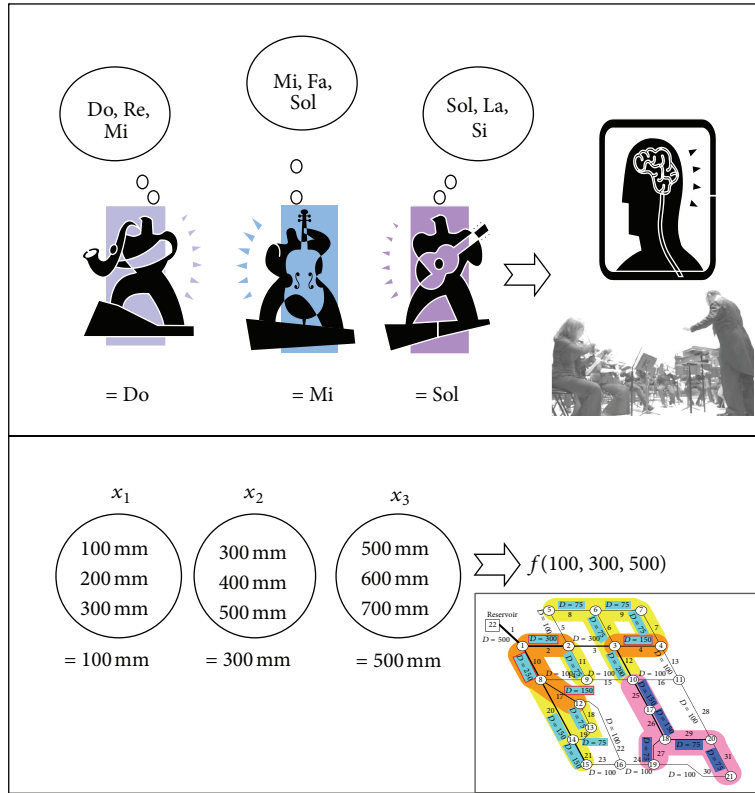
Figure 4: Relation between music improvisation and engineering optimization [15, 16].

Table 2: Definitions and specifications of two design variables problems.

Functions	Definition
Rosenbrock's valley	Minimize $f(x_1, x_2) = 100(x_1^2 - x_2^2)^2 + (1 - x_1)^2$ $-2.048 \leq x_1 \leq 2.048, -2.048 \leq x_2 \leq 2.048$ $x_1 = 1, x_2 = 1, f(1,1) = 0$
Branin's function	Minimize $f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + 10$ $a = 1, b = 5.1/4\pi^2, c = 5/\pi, d = 6, e = 10, f = 1/8\pi$ $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$ $(x_1, x_2) = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475)$ $f(x_1, x_2) = 0.397887$
Easom's function	Minimize $f(x_1, x_2) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1 - \pi)^2 - (x_2 - \pi)^2\right)$ $-100 \leq x_1 \leq 100, -100 \leq x_2 \leq 100$ $(x_1, x_2) = (\pi, \pi), f(x_1, x_2) = -1$
Goldstein price's function	Minimize $f(x_1, x_2) = \left[1 + \left(x_1 + x_2 + 1\right)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2\right)\right]$ $\times \left[30 + \left(2x_1 - 3x_2\right)^2 \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2\right)\right]$ $-2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2$ $(x_1, x_2) = (0, -1), f(x_1, x_2) = 3$
Six-hump camel back function	Minimize $f(x_1, x_2) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$ $-3 \leq x_1 \leq 3, -2 \leq x_2 \leq 2$ $(x_1, x_2) = (-0.0898, 0.7126), (0.0898, -0.7126)\ f(x_1, x_2) = -1.0316$

2.3. *Harmony Search Algorithm.* The HS algorithm is based on musical performance processes that occur when a musician searches for a better state of harmony such as during jazz improvisation. Jazz improvisation seeks musically pleasing harmony (a perfect state) as determined by an aesthetic standard, just as an optimization process seeks a global solution (a perfect state) as determined by an objective function. The pitch of each musical instrument determines the aesthetic quality, just as a set of values assigned to each decision determines the value of the objective function. Figure 4 shows the details of the analogy between music improvisation and engineering optimization [15, 16].

In musical improvisation, each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good harmony, that experience is stored in each player's memory, and the possibility of making a good harmony is increased next time. Similarly, in engineering optimization, each decision variable initially chooses any value within the possible range, together making one solution vector. If all the values of the decision variables result in a good solution, that experience is stored in each variable's memory, and the possibility of making a good solution is also increased at the next iteration. In brief, the steps of HS algorithm are given as follows [16, 17].

Step 1 (initialize the problem and algorithm parameters). The optimization problem is defined as minimize $f(x)$ subject to $x_{iL} \leq x_i \leq x_{iU}$ ($i = 1, 2, \ldots, N$) and other existing constraints. x_{iL} and x_{iU} are the lower and upper bounds of decision variables, respectively. The HS algorithm parameters are also specified in this step. They are the harmony memory size (HMS), or the number of solution vectors in the harmony memory, HMCR, bandwidth (b_w), PAR, and the number of improvisations (K), or stopping criterion.

Step 2 (initialize the harmony memory). The initial harmony memory (HM) is generated from a uniform distribution in the ranges $[x_{iL}, x_{iU}]$ ($i = 1, 2, \ldots, N$), as given in (1):

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^1 & \cdots & x_N^2 \\ \vdots & \vdots & & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} \end{bmatrix}. \quad (1)$$

Step 3 (improvise a new harmony). Generating a new harmony is called improvisation. The new harmony vector $x' = (x_1', x_2', \ldots, x_N')$ is determined by three rules: memory consideration, pitch adjustment, and random selection. The pseudocode for generating a new harmony search is given as shown in Algorithm 1 [17].

x_i' ($i = 1, 2, \ldots, N$) is the ith component of x', and x_i^j ($j = 1, 2, \ldots, HMS$) is the ith component of the jth candidate solution vector in the HM. *rand* is a uniformly generated random number in the region of [0, 1], and b_w is an arbitrary distance bandwidth.

Step 4 (update harmony memory). If the fitness of the improvised harmony vector $x' = (x_1', x_2', \ldots, x_N')$ is better than that of the worst harmony in the HM with x'.

Step 5 (check the stopping criterion). If the stopping criterion (maximum number of iterations, K) is satisfied, computation

is terminated. Otherwise, go to Step 3. The most important step of the HS algorithm is Step 3, which includes memory consideration, pitch adjustment, and random selection. The PAR and b_w have a profound effect on the performance of the HS.

TABLE 3: Obtained statistical results of 2D benchmark problem.

	SSWCHS	CHS	SHS
1. Rosenbrock			
Mean	$1.74E-06$	$3.99E-08$	$2.37E-04$
Best	$3.60E-11$	$1.60E-11$	$4.07E-09$
Worst	$2.30E-05$	$4.29E-07$	$1.63E-02$
SD	$3.77E-06$	$7.16E-08$	$1.65E-03$
Feasible solution	100	100	89
Mean iteration	698.68	1,102.25	20,142.62
Iteration SD	574.89	1,030.11	1,091.92
2. Branin's function			
Mean	$3.98E-01$	$3.98E-01$	$3.98E-01$
Best	$3.58E-07$	$3.58E-07$	$3.58E-07$
Worst	$3.58E-07$	$3.58E-07$	$4.33E-07$
SD	$2.04E-11$	$1.29E-12$	$8.20E-09$
Feasible solution	100	100	89
Mean iteration	211.94	444.13	2,593.45
Iteration SD	124.18	225.88	225.88
3. Easom's function			
Mean	$-6.70E-01$	$-9.41E-01$	$-4.90E-01$
Best	$3.60E-13$	$3.60E-13$	$8.21E-13$
Worst	$1.00E+00$	$7.44E-01$	$1.00E+00$
SD	$4.70E-01$	$1.45E-01$	$5.00E-01$
Feasible solution	67	67	49
Mean iteration	1,501.27	2,461.70	10,817.10
Iteration SD	564.37	969.64	1,133.84
4. Goldstein price's function			
Mean	$3.00E+00$	$3.00E+00$	$1.92E+01$
Best	$0.00E+00$	$0.00E+00$	$0.00E+00$
Worst	$1.21E-05$	$1.12E-05$	$2.70E+01$
SD	$1.62E-06$	$1.93E-06$	$1.32E+01$
Feasible solution	100	100	40
Mean iteration	257.89	423.43	1,009.08
Iteration SD	150.64	177.71	280.99
5. Six-hump camel back function			
Mean	$-1.03E+00$	$-1.03E+00$	$-1.03E+00$
Best	$2.85E-05$	$2.85E-05$	$2.84E-05$
Worst	$2.85E-05$	$2.85E-05$	$2.85E-05$
SD	$2.51E-11$	$2.79E-10$	$1.36E-09$
Feasible solution	100	100	100
Mean iteration	87.34	152.80	852.06
Iteration SD	33.13	67.30	67.30

"SD" stands for standard deviation.

```
for each i ∈ [1, N] do
  if rand() ≤ HMCR then
    x'_i = x^j_i (j = 1, 2, ..., HMS) % memory consideration
    if rand() ≤ PAR then
      x'_i = x'_i ± rand() × b_w % pitch adjustment
      if x'_i > x_{iU}
        x'_i = x_{iU}
      elseif x'_i < x_{iL}
        x'_i = x_{iL}
      end
    end
  else
    x'_i = x_{iL} + rand() × (x_{iU} − x_{iL}) % random selection
  end
end
```
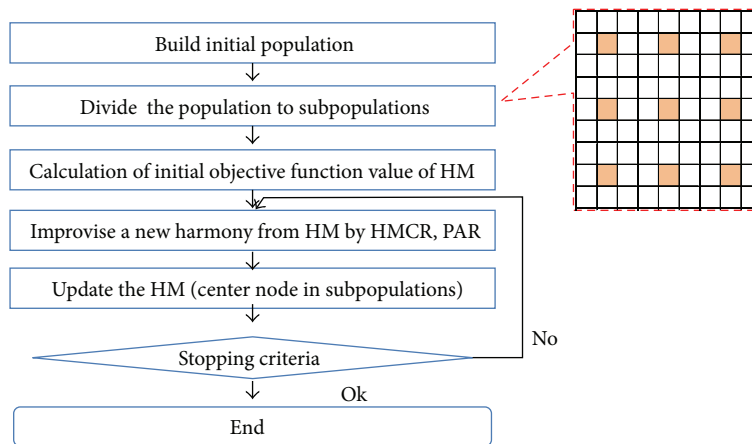
ALGORITHM 1

FIGURE 5: Flowchart of the CHS.

3. Smallest-Small-World Cellular Harmony Search

In the present paper, we proposed the improved HS algorithm that uses the CA concept and has the topological structure of the SSWN. A population is just a group of certain number of arbitrary objects in the same generation as for the generation in the original HS. Meanwhile, these objects are not related to each other. In the configuration of these populations as a kind of random network, the L and C are also very low. In contrast, if the population only consists of a grid of cellular networks, the C is relatively high; however, the L is also high. Because high L makes the interactions of remote nodes difficult, we need to reduce the L. Table 1 shows the efficiency and comparison of clustering coefficient and path length. The Small-World network models have the advantages of both random and regular network. They have low L for fast interaction between nodes, and they have high C ensuring sufficient redundancy for high fault tolerance [3]. In this study, therefore, the population of the original HS consists of a form of cellular networks. And then we can reduce

characteristic path length (L) and increase the clustering coefficient (C) using the shortcuts concepts of Small-World network.

3.1. Cellular Harmony Search (CHS). The operation process of the cellular harmony search (CHS) is shown in Figure 5. The CHS's operation process is the same as the existing original HS. However, the CHS's initial population has a cellular form. In this process, new generations in the sub-network from the PAR and HMCR processes are compared with the existing population, and then replace the lowest ranked object. After each small grid within the objects of the highest priority, a node is located in the center of the sub-network. Finally, the centers of the sub-network nodes are compared, and the object of the entire population of the highest priority and the central node are replaced.

3.2. Smallest Small World Cellular Harmony Search (SSWCHS). Smallest Small World cellular harmony search (SWCHS) performs its operation between the center nodes

TABLE 4: Definitions and specification of 30D benchmark problems.

Functions	Definition
Step function	Minimize $f(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2 - 30 \leq x_i \leq 30$ $\min(f) = f(0,\dots,0) = 0$
The Rastrigin function	Minimize $f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)] - 5.12 \leq x_i \leq 5.12 \min(f) =$ $f(0,\dots,0) = 0$
The Griewank function	Minimize $f(x) = (1/4000)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(x_i/\sqrt{i}\right) + 1 - 600 \leq x_i \leq 600$ $\min(f) = f(0,\dots,0) = 0$
The Ackley function	Minimize $f(x) = -a\exp\left(-b\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}\right) - \exp\left((1/n)\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1)$ $a = 20, b = 0.2, c = 2\pi - 32.768 \leq x_i \leq 32.768$ $\min(f) = f(0,\dots,0) = 0$

TABLE 5: Results of 30ND Problems.

	SSWCHS	CHS	SHS
1. Step function			
Mean	$0.00E+00$	$1.09E+00$	$5.60E+00$
Best	$0.00E+00$	$0.00E+00$	$0.00E+00$
Worst	$0.00E+00$	$3.00E+00$	$1.20E+01$
SD	$0.00E+00$	$9.81E-01$	$2.53E+00$
Feasible solution	100	36	1
Mean iteration	30,673.72	148,717.39	91,353.00
Iteration SD	41,902.01	49,240.37	—
2. Rastrigin function			
Mean	$4.69E-02$	$2.99E+00$	$6.09E+00$
Best	$2.78E-03$	$1.12E-02$	$1.01E+00$
Worst	$1.00E+00$	$5.01E+00$	$8.98E+00$
SD	$1.95E-01$	$1.31E+00$	$1.99E+00$
Feasible solution	92	0	0
Mean iteration	113,790.16	—	—
Iteration SD	47,257.60	—	—
3. Griewank function			
Mean	$1.03E-01$	$1.43E-01$	$6.94E-01$
Best	$1.46E-06$	$2.71E-02$	$5.47E-06$
Worst	$4.86E-01$	$4.85E-01$	$1.25E+00$
SD	$9.80E-02$	$8.06E-02$	$4.47E-01$
Feasible solution	10	0	3
Mean iteration	73,642.00	—	208,726.00
Iteration SD	43,912.52	—	26,918.84
4. Ackley function			
Mean	$4.62E-02$	$1.31E+00$	$1.58E+00$
Best	$3.51E-03$	$6.10E-03$	$5.97E-03$
Worst	$1.50E+00$	$1.65E+00$	$3.46E+00$
SD	$2.11E-01$	$2.78E-01$	$7.17E-01$
Feasible solution	96	3	8
Mean iteration	79,262.84	141,376.33	160,133.00
Iteration SD	64,770.38	35,365.89	58,642.35

in Figure 6. In this process, the SSWCHS is performed to obtain the final optimal solution through operations between the small grids of optimum object.

4. Applications and Results

In this study, the proposed SSWCHS was applied for solving unconstrained benchmark functions widely examined in the literature. The SSWCHS, CHS, and original simple harmony search (SHS) problems were performed in this study to have comparisons among optimizers. The optimization task was carried out using 100 independent runs based on the results depending on the type of problem. Statistical values, including best, worst, and mean values, and mean iteration number were obtained to evaluate the performance of the reported algorithms. Benchmark functions were utilized to evaluate the performance of considered optimization techniques. Among benchmark functions, five benchmark functions have 2 design variables and the rest have 30 design variables [18]. The benchmark functions have difficulty in terms of number of local optimum points and also the search space of these functions is almost wide and can challenge the efficiency of methods. The SSWCHS, CHS, and SHS algorithms were applied and run separately. The SSWCHS and CHS that occur at random initial solution were applied equally, and the initial solution in the SHS was applied differently because of the size of the HM. The HMS in the SSWCHS and CHS was composed by 225 cellular structures, and the SHS was composed by 5 HMS. Therefore, the number of shortcuts can be predetermined as shown in Figure 7. Two and 30 design variable problems were initialized using 0.3 PAR and 0.95 HMCR, respectively. The 2 and 30 design variable functions were performed with 50,000 iterations and 250,000 iterations, respectively. Feasible solution has a margin of error of 10^{-4} and 10^{-2} in the 2 and 30 design variable functions. Mean value and standard deviation of iteration can indicate the degree of convergence. From the number of feasible solution, the accuracy percentage of each algorithm can be compared.

by adding shortcuts in the CHS. The CHS operation between the center nodes is added in the calculation process as shown

4.1. Problems Having Two Design Variables. In this study, the following problems were applied as shown in Table 2.
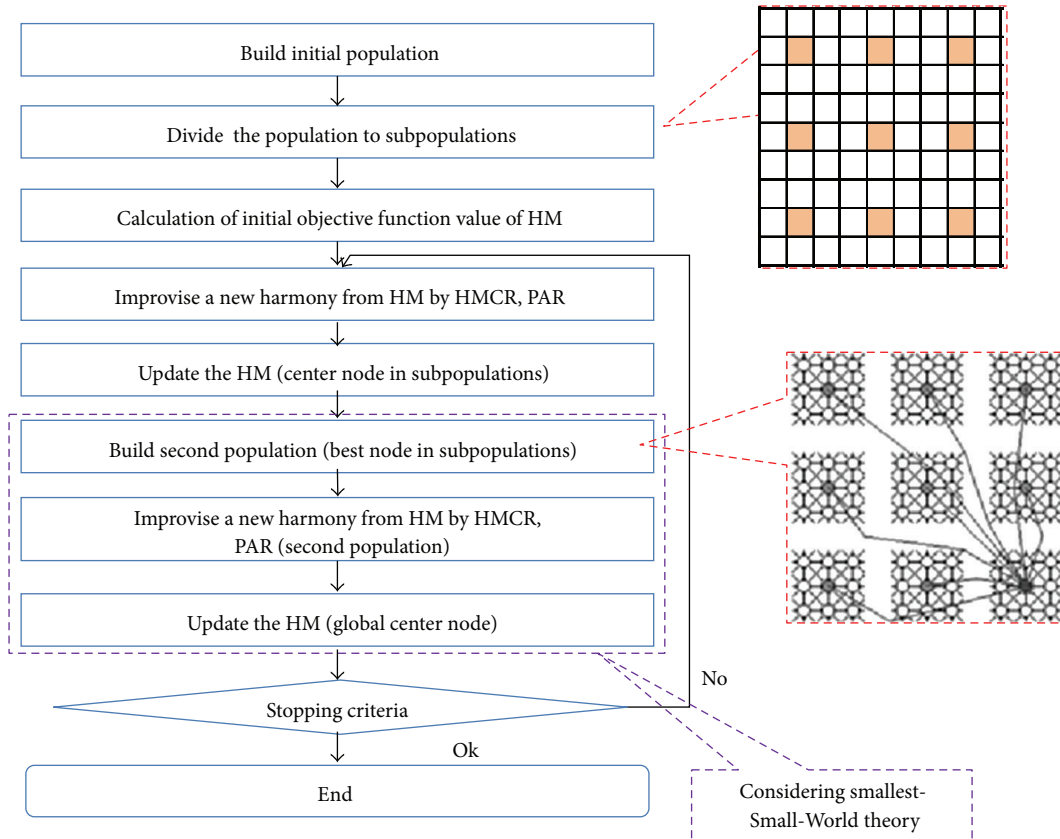
Figure 6: Flowchart of the SSWCHS.

Table 3 shows the obtained results of the 2D benchmark functions. The SSWCHS and CHS relatively showed good results, compared to the SHS. The result of each benchmark function indicated that Easom's function could not find perfect optimal solution by the SSWCHS and CHS. The parameters of Easom's function have a wide search range. For this reason, Easom's function showed less search capability for the optimal value than the other functions. In such case, increasing the number of iterations is considered to reach the optimal solution so that the ratio can be improved. One can see that the value of the entire functions which is mean iteration and iteration standard deviation value of the SSWCHS are smaller than those of the CHS. These results can conclude that the SSWCHS converged faster than the CHS, and it can estimate a reliable optimal solution.

4.2. Problems Having 30 Design Variables. In this study, the following benchmark problems were applied as shown in Table 4.

Table 5 shows the attained results of 30D benchmark functions. The SSWCHS relatively showed good results compared with the CHS and SHS. For all benchmark problems given in Table 4, except Griewank's function, the SSWCHS showed its superiority over other reported methods. In case of Griewangk's function, we can see that with the ratio of 10% the optimal solution is reached. Because it had wide search ranges of parameters from −600 to 600. By contrast to the 2D

problems, 30D functions need more time to reach the optimal solution. However, the SSWCHS showed much higher performance than the CHS and SHS. In particular, the arithmetic mean and the standard deviation in the SSWCHS are lower than those in other algorithms, indicating stability and faster convergence. For the Ackley function and the Griewangk function, however, the CHS and SHS offered lower standard deviation of iteration than the SWSCHS. However, in this case, the number of feasible solutions was very small and mean iteration was larger than that obtained by the SSWCHS. In case of the percentage of optimal solutions of the SHS, the values of the Griewangk function and Ackley function were larger than those of the CHS.

5. Conclusions

In this study, an improved HS algorithm which combines the CA and the topological structure of Smallest-Small-World network is proposed. Most of previous studies, there have been a lack of studies on the performance improvement of the harmony search algorithm by use of population or memory structures. A new hybrid harmony search algorithm having high clustering coefficient and short characteristic path length was required. The hybrid HS algorithm developed in this paper has good exploration and exploitation efficiencies. Nine benchmark functions were applied to assess the performance of the proposed algorithm. The applied benchmark

☐ Center node (global)

↗ Shortcut

The number of shortcuts: 24

Figure 7: Smallest Small World harmony search network.

functions consist of five 2D functions and four 30D functions. The evaluation indexes of the SSWCHS were better than those of CHS and SHS in terms of solution quality. The SSWCHS algorithm showed generally faster convergence and more stability than the CHS or SHS. It shows very competitive solutions with less number of iterations than other considered algorithms. It is recommended that the optimization techniques, as new algorithms became available, be used in a wide range of engineering optimization problems. However, the SSWCHS has so many of the HS structures that it can affect computation time. Therefore, it remains a complementary part. As a further research, parameter variations are expected to develop using the proposed SSWCHS.

Acknowledgment

References

[1] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' network," *Nature*, vol. 398, pp. 440–442, 1998.

[3] T. Nishikawa, A. E. Motter, Y. C. Lai, and F. C. Hoppensteadt, "Smallest small-world network," *Physical Review E - Statistical,*

Nonlinear, and Soft Matter Physics, vol. 66, no. 4, Article ID 046139, 2002.

[4] V. Latora and M. Marchiori, "Notions of local and global efficiency of a network," *Physical Review Letters*, vol. 87, Article ID 198701, 2001.

[5] T. W. Kang, "Smallest-small-world cellular genetic algorithm," *Journal of Computing Science and Engineering*, vol. 34, no. 11, pp. 971–983, 2007.

[6] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[7] J. H. Kim, Z. W. Geem, and E. S. Kim, "Parameter estimation of the nonlinear muskingum model using harmony search," *Journal of the American Water Resources Association*, vol. 37, no. 5, pp. 1131–1138, 2001.

[8] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, no. 3, pp. 601–644, 1983.

[9] J. Von Neumann, *Theory of Self-Reproducing Automata*, University of Illinois Press, Urbana, Ill, USA, 1966.

[10] B. Dorronsoro and E. Alba, "A simple cellular genetic algorithm for continuous optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 2838–2844, Vancouver, Canada, July 2006.

[11] G. Folino, C. Pizzuti, and G. Spezzano, "Combining cellular genetic algorithms and local search for solving satisfiability problems," in *Proceedings of the IEEE 10th International Conference on Tools with Artificial Intelligence*, pp. 192–198, November 1998.

[12] G. Timar and D. Balya, "Regular small-world cellular neural networks: key properties and experiments," in *Proceedings of the IEEE International Symposium on Cirquits and System*, pp. 69–72, British Columbia, Canada, May 2004.

[13] A. B. Hashemi and M. R. Meybodi, "A multi-role cellular PSO for dynamic environments," in *Proceedings of the 14th International CSI Computer Conference (CSICC '09)*, pp. 412–417, October 2009.

[14] V. Noroozi, A. B. Hashemi, and M. R. Meybodi, "CellularDE: a cellular based differential evolution for dynamic optimization problems," in *Proceedings of the ICANNGA*, vol. 6593 of *Lecture Notes in Computer Science*, pp. 340–349, 2011.

[15] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[16] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[17] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.

[18] M. Molga and C. Smutnicki, "Test functions for optimization needs," 2005, http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.