Alpinist CellularDE: A Cellular based Optimization Algorithm for Dynamic Environments

Vahid Noroozi
Dept. of Computer Eng. and IT
Amirkabir University of Technology
Tehran, Iran
vnoroozi@aut.ac.ir

Ali B. Hashemi
Dept. of Electrical and Computer Eng.
University of Toronto
Toronto, Canada
ali.b.hashemi@utoronto.ca

Mohammad Reza Meybodi
Dept. of Computer Eng. and IT
Amirkabir University of Technology
Tehran, Iran
mmeybodi@aut.ac.ir

ABSTRACT
In this paper, we propose Alpinist CellularDE to address dynamic optimization problems. Alpinist CellularDE tries to detect different regions of the landscape and uses this information to perform more effective search and increase its performance. Moreover, in Alpinist CellularDE a directed local search is proposed to track local optima after detecting a change in the environment. The proposed algorithm is evaluated on various dynamic environments, modeled by Moving Peaks Benchmark. Experiments show superior performance of Alpinist CellularDE in all test cases in comparison with some of the best performing evolutionary algorithms for dynamic optimization problems.

Categories and Subject Descriptors
G.1.6 [**Numerical Analysis**]: Optimization – global optimization; I.2.8 [**Artificial Intelligence**]: Search – *Heuristic methods*

General Terms
Algorithms

Keywords
Dynamic Environments, Evolutionary Algorithms, Cellular Automata, Differential Evolution, Moving Peaks Benchmarks, CellularDE

1. INTRODUCTION
There are many optimization problems that are dynamic in which the fitness function of the problem changes over time [1]. One of the algorithms which are developed for DOPs is CellularDE that has shown a good performance in various tested dynamic environments [2].

In this paper, a modified version of CellularDE, named Alpinist CellularDE, is proposed. In the following, first CellularDE is introduced briefly in section 2. Then, the proposed approach is presented in Section 3. In Section 4, the settings of the experiments with the experimental results and relevant analysis are presented. Finally, section 5 concludes the paper.

2. CellularDE
CellularDE, inspired by Cellular PSO [3, 4], is a cellular based algorithm which utilizes Differential Evolution (DE) as the main search algorithm and uses Cellular Automata (CA) to control the search process. It utilizes CA to partition the search space into sub-spaces, known as cells, and to control the search process in each cell locally. The evolution of individuals in each cell is performed locally by DE. This helps the algorithm to maintain diversity meanwhile it can exploit the local optimums appropriately.

CellularDE has performed well in most of tested dynamic environments. However, the search process in CellularDE is not performed efficiently. For example, peak tracking after detecting a change in the environment is performed with a random local search, which is not efficient. In addition, individuals may try to over exploit peaks which consume the precious fitness evaluation inefficiently, especially if the peak is not the global optimum. The latter is more important in dynamic environments since in dynamic environments finding approximate location of many peaks is more helpful for tracking the peaks than finding the exact location of a few peaks. The proposed Alpinist CellularDE tries to tackle these problems.

3. Proposed Algorithm: Alpinist CellularDE
In Alpinist CellularDE, each cell estimates the presence of a peak within its boundary or its neighbors' and adopts its search strategy based on that. Moreover, to track the peaks more efficiently after detecting a change in the environment, Alpinist CellularDE utilizes a proposed directed local search instead of the random local search performed in CellularDE. The proposed algorithm works as follows.

First, for an environment with D dimensions, cellular automata with N_p^D cells are embedded into the search space. Then, M individuals are randomly initialized in the cells of the cellular automata. At each iteration, if there is at least one individual in cell i, the cell checks to update the fittest location found in its boundary ($cBest_i$). Then, local best memory of cell i ($NBest_i$) is updated to contain the best position found in cell i and its neighbors. Afterwards, cell i determines its type based on which the search strategy of the individuals in cell i is selected and applied. Each cell determines its type as follows.

In Alpinist CellularDE a cell can be either of these types: PEAK, SLOPE, PLATUE, or UNKNOWN. Initially all type of cells are UNKNOWN. A PEAK cell is a cell which probably contains a local optimum. A cell i is a PEAK cell if the fitness of the best position found in cell i ($cBest_i$) improves less than ϕ during the last t_h updates, and at the same time the local best memory of cell i ($cBest_i$) is located at cell i, i.e. $cBest_i = NBest_i$. A SLOPE cell is a cell adjacent to a PEAK cell. Hence, all neighbors of a PEAK cell are considered SLOPE cells. In PLATUE cells, *NBest* is located on a neighboring SLOPE cell.

Since PEAK cells are the most promising cells to contain local optima, in these cells the search process is performed through hill climbing around the *NBest* instead of DE algorithm. This helps faster and more efficient convergence to the local optima.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

Although it is possible that SLOPE and PLATUE cells contain a better local optimum that their neighbor PEAK cell, it is more efficient to perform the search process in these cells with less individuals. Hence, in SLOPE and PLATUE cells, the capacity of cells, i.e. the maximum number of individuals allowed to search in a cell, is decreased to half.

In cell i that is not a PEAK cell, individuals search using DE algorithm. The mutant vector for an individual k with the position vector x_k which resides in cell i is calculated by eq. (1).

$$v_k = NBest_i + \lambda_{rand1}(x_k - x_1) + \lambda_{rand2}(x_{r2} - x_{r3}) \quad (1)$$

where x_{r1} and x_{r2} are the position vectors of three random selected individuals in the cell i. λ_{rand1} and λ_{rand2} are uniform random variables in [0,1]. Moreover, to improve exploration capability of the individuals, after calculating the mutant vector, a normal noise with the probability of 0.25 is added to a randomly selected dimension.

3.1 Dealing with the Change

After a change is detected in the environment, in order to track previously found local optima more effectively, a directed local search, similar to a directed search introduced in [7], is proposed to substitute the simple random local search of CellularDE. In the proposed local search, the best memory of a cell is considered as the base point to start the local search as follows.

For cell i, a binary direction vector $dir^i=(dir_1^i, dir_2^i,\ldots, dir_D^i)$, $dir_j^i \in \{-1,1\}$ is defined which specifies the direction of the search in each dimension. This vector is initialized randomly at the beginning of the local search, i.e. after detecting a change in the environment.

In each iteration, a random point around $cBest_i$, is considered such that it only differs from $cBest_i$ in the j^{th} dimension. The location of the selected random point is calculated using eq. (2).

$$x_{k,j} = cBest_{i,j} + b.(a)^{ndrs_{i,j}}.dir_{i,j} \quad (2)$$

where $b$is the initial step of random search. a is a constant in (0,1) which controls the speed of random search convergence. $ndrs_{i,j}$ is the number of unsuccessful random search performed on dimension j of cell i. If the fitness of the new position is better than the fitness of $cBest_i$, $cBest_i$ is replaced by the position. Otherwise, the search is performed on the opposite direction on dimension j. If the search in the new direction for dimension j is not successful too, $ndrs_{i,j}$ increases by one. The above procedure is performed for all dimensions until $ndrs_{i,j}$ for all dimensions reaches its limit ($ndrs^*$).

4. Experiments

For all the experiments, Moving Peaks Benchmark (MPB) [5] is adopted as the dynamic benchmark function [1]. Experiments are conducted on the well-known scenario II of MPB with the default settings described in [5, 3, 2]. Size of cellular automata, initial step size (b), reduction factor (a), and $ndrs^*$ are empirically set to 10^5, 10, 0.2, and 4, respectively. The size of history of best memory if cell (t_h) and fitness improvement threshold (ϕ) are set to 6 and 0.1, respectively. Parameter Cr of DE is set to 0.7. Other parameters are set to values proposed in [2].

The proposed algorithm is compared with CellularDE [2] and Adaptive mQSO [6]. For all experiments, parameters of CellularDE and Adaptive mQSO are set to the values reported in [2] and [3], respectively. All algorithms are experimented for 500000 fitness evaluations and the average offline errors of the algorithms in 100 runs with 95% confidence interval are reported in Table 1. Experiments are performed on dynamic environments with different frequency of change, f={1000, 2500, 5000}, and different number of peaks, m={10, 50, 100}. As depicted, Alpinist CellularDE outperforms other tested algorithms in all cases. This is due to the efficient use of fitness evaluations by increasing the exploitation power of the algorithm around the local optima by replacing DE with hill climbing in PEAK cells, and utilizing the proposed directed local search which results in faster tracking of the peaks after detecting a change in the environment.

Table 1. Offline error for different environments

m	f	CellularDE	Alpinist CellularDE	Adaptive mQSO
10	1000	4.58±0.05	**3.43±0.04**	4.11±0.08
50		5.70±0.03	**4.62±0.02**	5.12±0.05
100		5.76±0.04	**4.46±0.02**	5.03±0.09
10	2500	3.04±0.05	**2.06±0.05**	2.33±0.11
50		4.14±0.05	**2.44±0.03**	3.24±0.07
100		4.15±0.03	**2.38±0.02**	3.20±0.06
10	5000	2.19±0.05	**1.28±0.04**	1.43±0.04
50		3.34±0.05	**2.04±0.04**	2.28±0.02
100		3.37±0.05	**1.90±0.03**	2.31±0.03

5. Conclusion

In this paper we proposed Alpinist CellularDE to address dynamic optimization problems. In Alpinist CellularDE, a proposed DE scheme which is suitable for dynamic environments is used. Moreover, the search process in each cell is performed differently based on its location and estimated position of local optima. In addition, the random local search in CellularDE is replaced by a proposed directed local search to improve the efficiency of tracking local optima after detecting a change in the environment. Experiments on various dynamic environments modeled by Moving Peaks Benchmark show that Alpinist CellularDE outperforms other tested algorithm in all cases.

6. REFERENCES

[1] Cruz, C., González, J. R., and Pelta, D. A. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, (2011), 1-22.

[2] Noroozi, V., Hashemi, A., and Meybodi M. 2010., CellularDE: a cellular based differential evolution for dynamic optimization problems. In *Proceedings of the Adaptive and Natural Computing Algorithms*. 340-349.

[3] Hashemi, A. and Meybodi, M. 2009. Cellular PSO: A PSO for Dynamic Environments. In *Proceedings of the Advances in Computation and Intelligence*. 422-433.

[4] Hashemi, A. and Meybodi, M. 2009. A multi-role cellular PSO for dynamic environments. In *Proceedings of the 14th International CSI Computer Conference*. 412-417.

[5] Branke, J. 2002. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*,1875-1882.

[6] Blackwell, T., Branke, J., and Li, X. Particle swarms for dynamic optimization problems. *Swarm Intelligence*, (2008), 193-217.

[7] Hooke, R. and Jeeves, T. A. "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the ACM*, 8 (1961), 212-229.