Both Robust Computation and Mutation Operation in Dynamic Evolutionary Algorithm are Based on Orthogonal Design

Sanyou ZENG, Rui WANG, Hui SHI, Guang CHEN
School of Computer Science & Research Center for Space Science and Technology, China University of GeoSciences, 430074 Wuhan, Hubei, P.R.China
Emal:sanyou-zeng@263.net

Hugo de GARIS
Brain Builder Group, Computer Science Dept.,Utah State University, Logan, Utah, 84341, USA

Lishan KANG , Lixin DING
StateKey Laboratory of Software Engineering, Wuhan University,430072Wuhan, Hubei, P.R.China

ABSTRACT
A robust dynamic evolutionary algorithm (labeled RODEA), where both the robust calculation and mutation operator are based on an orthogonal design, is proposed in this paper. Previous techniques calculate the mean effective objective (for robust) by using samples without much evenly distributing over the neighborhood. The samples by using orthogonal array distribute evenly. Therefore the calculation of mean effective objective more robust. The new technique is generalized from the ODEA algorithm [1]. An orthogonal design method is employed on the niches for the mutation operator to find a potentially good solution that may become the representative in the niche. The fitness of the offspring is therefore likely to be higher than that of its parent. We propose a complex benchmark, consisting of moving function peaks, to test our new approach. Numerical experiments show that the moving solutions of the algorithm are a little worse in objective value but robust.

Categories and Subject Descriptors
Cognitive simulation

General Terms: Algorithms

KEYWORDS: Robust Solutions, Genetic algorithms, Dynamics, Optimization methods

1 THE CONCEPT OF DYNAMIC ROBUST OPTIMIZATION

We consider a dynamic optimization problem of the following type:

$$x_{\max}(t) = \arg \underset{x \in S}{Maximize} \quad f(x,t) \qquad (4)$$

where $\boldsymbol{x}=(x_1, x_2, \ldots, x_N)$ and $\boldsymbol{S}$ is the search space. In addition to depending on the variables $\boldsymbol{x}$, the function values of f are time dependable. But the dimension N of $\boldsymbol{x}$ and the search space $\boldsymbol{S}$ are fixed with time. $\boldsymbol{x}_{max}(t)$ is the optimal solution at time t. It is impractical and even impossible to find the precise time-dependable optimal solutions $\boldsymbol{x}_{max}(t)$ for an optimization approach. The practical way is to track the moving optimal. The optimization approaches must spend time on finding the optimal solution at any time point. We must, therefore, suppose that the environment has a sudden change with a comparative long static. That is, the function values of f are only dependent on variables $\boldsymbol{x}$ without time-dependent in the static stage where we hope the algorithm approaches the optimal as much as possible.

When the objective f is highly sensitive to perturbation in variable space, the solutions relatively insensitive to the perturbations are much important. Those solutions are called robust solutions. One of the main approach portrayed in static optimization is to use a mean effective objective function ($f^{\text{eff}}(\boldsymbol{x})$) for robust optimization, instead of the original function $f(\boldsymbol{x})$ itself. Here, we give a definition for a dynamic optimization problem.

Definition 1 Dynamic robust Solution:
A solution $x_{\max}^{robust}(t)$ is called a dynamic robust optimal solution at time t if it is the global feasible solution to the following dynamic maximization problem:

$$x_{\max}^{robust}(t) = \arg \underset{x \in S}{Maximize} \quad f^{eff}(x,t) \qquad (5)$$

where

$$f^{eff}(x,t) = \frac{1}{|y \in B_\delta(x) \cap S|} \int_{y \in B_\delta(x) \cap S} f(y,t)dy \qquad (6)$$

where $B_\delta(x)$ is a super-rectangle with $\boldsymbol{x}$ being its center and side lengthen being the vector $\delta = (\delta_1, \delta_2, \ldots, \delta_N)$. $B_\delta(x)$ is called δ-neighborhood of solution $\boldsymbol{x}$.

2.Computing Robust Solutions Based On Orthogonal Array

As mentioned earlier, definition 1 requires optimizing effective objective function. To compute effective objective function values, H different points in the $\delta(t)$-neighborhood of solution $\boldsymbol{x}$ at time t are computed and their average is taken. It is intuitive that if the more neighbor-points are chosen for computing the mean effective function, the objective value will be closer to the theoretical average value; however, the computation will take more time. We denote the close value by $f^{\text{est}}(\boldsymbol{x}, t)$. Here a comparatively small H is chosen. These H points can be chosen randomly the neighborhood while in the literature

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

[2] they are chosen in a systematic manner. In this paper, The H points are chosen by orthogonal array. To create a orthogonal pattern, the perturbation range (around [-δ$_i$, δ$_i$]) of each variable (factor) is divided into exactly Q-1 equal portions with Q levels. Let Q=5 and J=2, then $H=5^2$=25 (cf. [1]). The 25 points will evenly distribute over the δ-neighborhood

3 Implementing The New Algorithm

According to robust solution definition, the objective function for RODEA algorithm is $f^{est}(\boldsymbol{x}, t)$, while that for ODEA(cf. [1]) is $f(\boldsymbol{x}, t)$. Therefore, the details of the RODEA algorithm are almost the same as ODEA [1]. The details of the RODEA algorithm is skipped.

4 TEST PROBLEM AND THE RESULTS OF NUMERICAL EXPERIMENTS

4.1 The Test Function

Similar to the test problem 3 in literature [2], we propose a benchmark problem. Its robust optimal solution changes with time t as in the following.

$$\begin{aligned} \text{Maximize} \quad & f(x,t) = h(x_1,t)g(x,t) \\ \text{where} \quad & g(x,t) = 1-\sum_{i=2}^{N}50(x_i-t)^2 \\ & h(x_1,t) = t\cdot\exp\left(-\left(\tfrac{x_1-0.35-t}{0.25}\right)^2\right)+\exp\left(-\left(\tfrac{x_1-0.85-t}{0.03}\right)^2\right) \end{aligned}$$

t=0, 0.1, 0.2, …,0.9, 1, 0.1, 0.2, …, 0.9, 1, 0, …… . That is the change period of t is 0, 0.1, 0.2, …,0.9, 1.

The search space S: $0 \le x_1 \le 1, 0 \le x_i \le 2, i = 2,...,N$. Here we suggest N=5. A careful look at the $h(x_1,t)$ function will reveal that two peaks exist in the problem, one is a wider peak at $x_1 \approx 0.35+t, x_i = t, i = 2,...,N$ with a mean squared deviation 0.35. The other is a thinner peak at $x_1 \approx 0.85+t, x_i = t, i = 2,..., N$ with a mean squared deviation 0.03, which is the global optimal. The $g(\boldsymbol{x})$ function makes it is difficult to find the optimal solution.

4.2 The Results Of the Numerical Experiments

Let t=0, 0.1, 0.2, …, 0.9, 1, 0 be the sudden environmental changes. We repeat 100 runs of the RODEA algorithm for each environment change: t=0, 0.1, 0.2, …, 0.9, 1, 0. We have the RODEA algorithm evaluates the f^{est}() 50000 times (about evolves 500 generations) to close to the optimal solution during the first time step (t=0). Afterwards, the RODEA algorithm evaluates the f^{est}() 10000 times (about evolves 100 generations) to close to the optimal solution during each time step. Tables 1, 2 show the (robust) solutions at each time step. Tables 1 shows the RODEA track the global optimal without robust consideration. Tables 2 shows the RODEA can track the robust optimal with neighboring radius δ$_i$=0.02. It also shows the original local optimal solution becomes the robust optimal solution when the time step t increases from 0.8 to 0.9. But, when the time step t changes from 1.0 to 0.0, the original global solution becomes back the robust optimal solution.

TABLE1

100 runs of RODEA yields $x_{1,\max}$, $f_{\max}$ and their occurring times without robust, where the number of evaluation of f^{est}() was 10000 during each time step.

t	$x_{1,\max}$	$f_{\max}$	times
0	0.850	1.0000	75
0.1	0.950	1.0018	100
0.2	1.050	1.0036	100
0.3	1.150	1.0055	100
0.4	1.250	1.0073	100
0.5	1.350	1.0092	100
0.6	1.450	1.0110	100
0.7	1.550	1.0128	100
0.8	1.650	1.0146	100
0.9	1.750	1.0165	100
1	1.850	1.0183	100
0	0.850	1.0000	95

TABLE 2

100 runs of RODEA yields the robust $x_{1,\max}^{robust}$, $f_{\max}^{robust}$ and their occurring times with δ$_1$= δ$_2$=…= δ$_N$=0.02, where the number of evaluation of f^{est}() was 10000 during each time step.

t	$x_{1,\max}^{robust}$	$f_{\max}^{robust}$	times
0	0.850	0.7818	70
0.1	0.950	0.7834	100
0.2	1.050	0.7854	100
0.3	1.150	0.7872	100
0.4	1.250	0.7890	100
0.5	1.350	0.7908	100
0.6	1.450	0.7926	100
0.7	1.550	0.7944	100
0.8	1.650	0.7962	100
0.9	**1.250**	**0.8612**	**100**
1	1.350	0.9569	100
0	0.850	0.7818	83

ACKNOWLEDGMENTS This work was supported by the National Natural Science Foundation of China (Grant Numbers: 60473037, 60483081, 40275034, 60204001, 60133010), and by a project supported by China Postdoctoral Science Foundation (No. 2003034505).

REFERENCES

[1] SanYou Zeng, Hugo de GARIS, Jun He, LiShan Kang. A Novel Evolutionary Algorithm Based on an Orthogonal Design for Dynamic Optimization Problems (ODEA). In Proceedings of the 2005 Congress on Evolutionary Computatoin (CEC05). 2005, UK, Vol 2,pp1188-1195

[2] K. Deb and H. Gupta. Searching for robust Pareto-Optimal Solutions in Multi-objective optimization. *Proceedings of the 2005 Conference on Evolutionary Multi-Criterion Optimization*, LNCS series, Springer-Verlag, pages 150-164, 2005.