A Coevolution Archive based on Problem Dimension

Liping Yang[1,2]
1. School of Computer and Information Technology
Beijing Jiaotong University
Beijing 100044, China
ylp_117@sina.com

Houkuan Huang
School of Computer and Information Technology
Beijing Jiaotong University
Beijing 100044, China
hkhuang@center.njtu.edu.cn

Yabin Liu
2. School of Compure and Information Engineering
Shandong University of Finance
Jinan 250014, China
abeliu69@yahoo.com.cn

ABSTRACT

Recent work has shown the existence of an implicit dimension structure within coevolution problems, which can uniquely determine the overall performance of a individual. In this paper, we present a reliable dimension identifying method. Based on this , we put forward a suitable archive scheme, which maintains only the most representative individuals in terms of problem dimensional information identified during coevolution, and achieves minimum size while guaranteeing monotonic progress. The experimental results on COMPARE-ON-ONE and COMPARE-ON-ALL demonstrate the viability of the algorithm .

Categories and Subject Descriptors

F.0 [**General**]

General Terms

Algorithm, Performance, Reliability, Experimentation

Keywords

Coevolution, archive, reliability, dimension, structure-archive

1. INTRODUCTION

Due to the unstability of dynamic evaluation, coevolution frequently displays complex dynamics. The archive methods[2] theoretically provide a guarantee of progress. However, the archive size of the existing methods can be prohibitive of practical applications.

Recent research[1] has shown that a large class of coevolution problems can be embedded into an ordered, n-dimensional Euclidean space. Coevolution can be viewed as a Evolutionary Multi-Objective Optimization. Each dimension corresponds to an underlying objective. Therefore, algorithms can guarantee monotonic progress by preventing evolution from regressing on each dimension.

2. PRELIMINARIES

This paper focuses on coevolutionary problems which are expressed with a function $p : S \times T \to R$, where S is the set of all candidate solutions, T is the set of all tests, R is an ordered set of scalar values which represent the outcomes of candidate-test interaction. For simplicity, we assume R to be binary here, i.e. $\{0, 1\}$. A candidate is said to pass a test if it has a positive outcome (1) on the test, and to fail it otherwise.

Theorem 1 Let $p : S \times T \to \{0,1\}$ be a coevolution problem. If there exist n candidates $(s_1, s_2 \ldots s_n)$, n tests $(t_1, t_2 \ldots t_n)$, and their outcome matrix of candidate-test interaction is as follows:

$$Mp= \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & \ldots t_n \\ \hline s_1 & 0 & 1 & 1 & \ldots & 1 \\ s_2 & 1 & 0 & 1 & \ldots & 1 \\ s_3 & 1 & 1 & 0 & \ldots & 1 \\ . & & & & \\ s_n & 1 & 1 & 1 & \ldots & 0 \end{array} \qquad \begin{array}{l} mp_{ii}=0, \\ \\ mp_{ij}=1 \ (i \neq j) \end{array}$$

then, the dimensionality of the problem is definitely not smaller than n, and each of the n tests corresponds to a different dimension respectively, representing the current highest coordinate on the dimension.

3. THE ALGORITHM

3.1 The Archive Design

Two archives are maintained during coevolution:

One structure-archive maintains the current highest test of each identified dimension, and is composed of the pairs (candidate, test), in which interaction outcomes of all pairs satisfy the premise of Theorem 1. Each test corresponds to a different dimension separately, representing the current highest test on the dimension. Candidates serve to support current dimension structure and succedent dimension identification. The archive is used to provide the evaluation support for the candidate archive.

One candidate-archive maintains high performance candidates, namely the pareto non-dominated front over all tests in the structure-archive. The archive is intended to guarantee monotonic progress.

Two archives are periodically updated during execution:

The structure-archive is updated in two kinds of situations. **(1).Found a new dimension.** If there exist a new candidate s and a test t, s can pass all tests in the structure archive, t can be passed by all candidates in the structure, while s cannot pass t; interaction outcomes in the union of (s, t) and all pairs in the structure archive satisfy the premise of Theorem 1. Thus, t represents a new dimension. Finally, the pair (s, t) is appended to the structure-archive. **(2).Found a higher test of a certain dimension.** There exist a new test nt and k new candidates $ns_1, ns_2, \ldots ns_k$, which correspond to the test t_i and k candidates $s_{i1}, s_{i2}, \ldots s_{i\,k}$ in the structure-archive respectively, notice that there exists a particular case when t_i and a certain candidate s_{ir} $(r \in [1,k])$ is just a pair in the structure-archive; except ns_r in particular case, each $ns_j (j=1 \ldots k)$ and corresponding s_{ij} have the same outcomes on all tests in the structure-archive, ns_j can pass nt but s_{ij} cannot; in the

particular case, ns_r and s_{ir} have the same outcomes on all tests except t_i in the structure-archive, both ns_r and s_{ir} cannot pass nt, while ns_r can pass t_i but s_{ir} cannot; nt and t_j have the same outcomes on all candidates except s_{i1}, s_{i2},…$s_{i\,k}$ in the structure-archive. Then, nt is considered to be higher than t_i on the ith dimension. Let nt, ns_1, ns_2, …ns_k replace t_j, s_{i1}, s_{i2}, …$s_{i\,k}$ in the structure-archive , respectively. The outcome relations in new structure-archive still satisfy the premise of Theorem 1.

The candidate-archive is updated while finding a new non-dominated candidate, which maintains current non-dominated candidates only.

3.2 The Coevolutionary Algorithm based on the Archive

The algorithm outline is the following:

PD-CA（）

  Cpop:=random population()

  Tpop:= random pouplation()

  Init_archive（Cpop，Tpop，&stru_arc，& cand_arc）

  While ~end_criterion

   Evolve(&Cpop，&Tpop，stru_arc，cand_arc)

   Update_archive(Cpop，Tpop，&stru_arc，& cand_arc)

  End

End

The main loop of the algorithm contains two parts. One part is the coevolution, carrying out the effective search. Other part is the archivism, maintaining excellent candidates so as to prevent from regressing. Both parts mutually cooperate, impeling coevolution to progress steadily.

4. EXPERIMENT

The test problems are COMPARE-ON-ONE and COMPARE-ON-AL, see [2,3] for details. The experiments employ the five-dimensional versions in which the maximum value of each dimension is 10. The candidate and test populations are both of size 6. Evolutionary operation setups are the same as the setups used in [2,3] except for parent selection. Here candidate parents which generate offspring are selected separately from the candidate population, the candidate-archive and the structure-archives with probability 0.7, 0.2 and 0.1. Test parents are selected separately from the test population, the structure-archive and the individuals generated randomly with probability 0.7, 0.2 and 0.1. This assign proportion not only considers all of sides so as to make evolution progress evenly, but also provides more chance for continual search for new dimensions.

Figure 1 and 2 show the experimental results of our algorithm and existing archive algorithms[2,3] on the two problems. We can see that the performance and efficiency of the PD-CA are clearly improved compared to other existing archive algorithms. AS the archives maintain only the most informative individuals and these individuals directly participate in generating new individuals, new individuals are more targetable. Furthermore, we select randomly a run from the experiments of the PD-CA on each problem respectively. Table 1 shows the tests maintained by

their structure archives in the end of execution. It indicates that the dimension identification made by the PD-CA algorithm for the two problems is accurate and reliable.
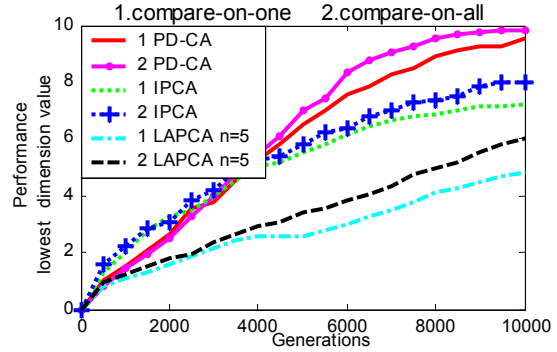

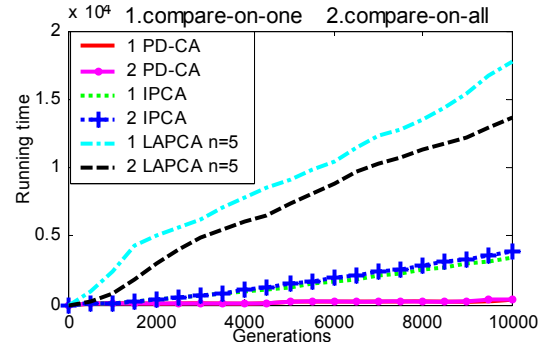**Figure 1. Performance of PD-CA、IPCA[2] and LAPCA[3] on the two problems .**

Figure 2. Running time of PD-CA、 IPCA and LAPCA on the two problems over generations

Table 1 Tests in the structure archives

No	compare-on-one	compare-on-all
1	(0.00 , 0.84 , 0.01 , 0.37 , **10.0**)	(**10.0 ,** 0.50 , 0.66 , 0.00 , 0.00)
2	(0.62 , **10.0** , 0.00 , 0.02 , 0.00)	(0.03 , 1.81 , 0.00 , 0.19 , **10.0**)
3	(0.45 , 0.31 , 1.10 , **10.0** , 1.67)	(0.00 , 0.00 , 0.00 , **10.0** , 1.37)
4	(**10.0** , 0.81 , 0.09 , 0.15 , 0.36)	(0.12 , 0.33 , **10.0** , 1.09 , 0.10)
5	(0.07 , 0.00 , **10.0** , 0.06 , 0.29)	(0.44 , **10.0 ,** 0.08 , 0.24 , 0.02)

5. CONCLUSIONS

The paper proposes a novel coevolution archive, which achieves minimal size while guaranteeing reliable progress and thus greatly enhances the performance and efficiency of the algorithm. Experimental comparison results demonstrate its feasibility and validity.

The algorithm provides a new idea for coevolutionary algorithm design and automated extraction of problem dimension.

REFERENCES

[1] Bucci, A. and Pollack, J. B. Focusing versus intransitivity: Geometrical aspects of coevolution. *GECCO -03.* 250–261

[2] De Jong, E.D. The Incremental Pareto-Coevolution Archive. *GECCO-04.*,.525-536.

[3] De Jong, E.D. Towards a bounded Pareto-Coevolution archive. *CEC-04.* 2341-2348.