Swarm and Evolutionary Computation 23 (2015) 1–10

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Regular Paper

Improved sampling using loopy belief propagation for probabilistic model building genetic programming

Hiroyuki Sato [a,*], Yoshihiko Hasegawa [b], Danushka Bollegala [c], Hitoshi Iba [b]

[a] The Graduate School of Engineering, The University of Tokyo, Japan
[b] The Graduate School of Information Science and Technology, The University of Tokyo, Japan
[c] Department of Computer Science, The University of Liverpool, United Kingdom

ARTICLE INFO

Article history:
Received 26 February 2013
Received in revised form
2 January 2015
Accepted 20 February 2015
Available online 6 March 2015

Keywords:
Genetic programming
Estimation of distribution algorithms
Loopy belief propagation
Probabilistic model building GP

ABSTRACT

In recent years, probabilistic model building genetic programming (PMBGP) for program optimization has attracted considerable interest. PMBGPs generally use probabilistic logic sampling (PLS) to generate new individuals. However, the generation of the most probable solutions (MPSs), i.e., solutions with the highest probability, is not guaranteed. In the present paper, we introduce loopy belief propagation (LBP) for PMBGPs to generate MPSs during the sampling process. We selected program optimization with linkage estimation (POLE) as the foundation of our approach and we refer to our proposed method as POLE-BP. We apply POLE-BP and existing methods to three benchmark problems to investigate the effectiveness of LBP in the context of PMBGPs, and we describe detailed examinations of the behaviors of LBP. We find that POLE-BP shows better search performance with some problems because LBP boosts the generation of building blocks.

1. Introduction

In the present paper, we introduce loopy belief propagation (LBP) in probabilistic model building GPs (PMBGPs) in order to generate the most probable solutions in sampling process. We call our novel method as POLE-BP.

Estimation of distribution algorithms (EDAs) are promising evolutionary algorithms and attract much attention from a lot of practical fields. EDAs optimize solution candidates represented by one dimensional arrays as well as Genetic Algorithms (GAs). Although EDA and GA employ the same chromosome representation, EDAs are different from GAs in the sense that EDAs generate new individuals by estimation of probabilistic models and sampling, whereas GAs generate them using genetic operators. EDAs can solve deceptive problems more efficiently than GAs by estimating dependencies between loci [27], which is one of the notable features of EDAs. Because of their effectiveness, many EDAs have been devised by incorporating many distinct statistical and machine learning approaches. Recently, EDAs using loopy belief propagation (LBP) as sampling were proposed in order to improve the sampling process [24,21]. LBP approximately infers marginal and the highest joint probabilities with configurations, and has been applied to a wide range of real world problems [7,6]. In EDAs, the individual with the highest joint probability in learned probabilistic models describes the models most and is often called as most probable solution (MPS). MPS is the individual which most reflects the learned models, and generation of it is important to take advantage of the models efficiently. However, traditional sampling methods used in EDAs, e.g. probabilistic logic sampling (PLS) [15] and Gibbs sampling, do not always generate MPS, and EDAs using only those samplings cannot make the best use of the models. In order to solve this problem, [24,21] generate MPS by LBP in addition to traditional sampling and showed better search performance than existing methods using only traditional samplings (PLS or Gibbs sampling) in benchmark problems.

The estimation of distribution concept employed in EDAs has been applied to the optimization of tree structures, which is traditionally addressed using GP. GP optimizes tree structures using operators, such as crossover and mutation, as well as GA. Numerous improved genetic operators have been proposed because it is difficult to deal with tree structures using only these simple operators. EDAs for tree structures are often called as Genetic Programming-EDAs (GP-EDAs) [11] or Probabilistic Model Building GPs (PMBGPs) [32], and the present paper adopts the latter abbreviation throughout the paper. PMBGPs are broadly classified into two types. One type uses probabilistic context free grammar (PCFG) to represent distributions of promising solutions and learns production rule probabilities. The other type is a prototype tree based method, which converts trees to one dimensional arrays and applies EDAs to them. From the viewpoint of probabilistic models, the prototype tree-based method is essentially equivalent to EDAs and hence it can easily incorporate techniques devised in the field of EDA.

We propose POLE-BP [34], the novel prototype tree-based PMBGP with LBP. POLE-BP generates MPS at every generation in addition to normal samplings (i.e. PLS) and makes the optimal use of the learned

* Corresponding author. Tel.: +81 3 5841 6751.
E-mail addresses: sato@iba.t.u-tokyo.ac.jp, umkn.smag@gmail.com (H. Sato).

http://dx.doi.org/10.1016/j.swevo.2015.02.002
2210-6502/

probabilistic model. We compare our proposed method against existing methods on three benchmark problems: the problem with no dependencies between nodes (MAX problem), the deceptive problem (Deceptive MAX problem) and the problem with dependencies between nodes (Royal Tree Problem). From results of the experiments, we show that the proposed method competes with the existing method in the deceptive problem and beats the existing method in the problems with no deceptiveness from the point of the number of fitness evaluations to get an optimum solution. Moreover, we investigate behaviors of LBP in the context of PMBGP by observing fitness values and structures generated by LBP, and show reasons why the proposed method does not exhibit search performance improvement in deceptive problems whereas it does in other benchmark problems.

The present paper extends our prior work [34] by studying the effectiveness of LBP in detail. The remainder of the paper is organized as follows. Section 2 introduces related work. Section 3 explains details of the proposed method. Section 4 presents the experimental condition and results, which is followed by the discussion in Section 5. Finally Section 6 concludes the paper.

2. Related work

We introduce existing PMBGPs and methods using loopy belief propagation as sampling in this section.

2.1. PMBGP: Probabilistic Model Building GP

PMBGPs are extensions of EDAs for tree structures that generate the next population by estimating probabilistic distributions from better individuals and sampling individuals from them. For recent surveys on EDAs, interested reader is directed to [18,14]. PMBGPs are superior to GP in the sense that PMBGPs can search for solutions with a smaller number of fitness evaluations and they can solve problems that conventional GP cannot [10]. Two types of methods are known in the field of PMBGPs.

- Prototype tree-based method.
- PCFG-based method.

The prototype tree-based method translates trees into one dimensional arrays and applies conventional EDAs to them. By contrast, the PCFG-based method expresses individuals with derivation trees and learns their production rules as well as their parameters. Because derivation trees naturally derive functions and programs, PCFG-based methods can estimate position-independent substructures, and so many approaches have been proposed based on these approaches [2,29,30,35,36,13]. However, the prototype tree-based methods have advantages over PCFG-based methods because they can readily utilize existing EDAs. Furthermore, the prototype tree-based methods are computationally reasonable even when we consider the dependencies between nodes, whereas PCFG-based methods that consider them are very computationally intensive. In addition, the probabilistic distribution concept is also applied to genetic network programming (GNP) [16,23], which expresses programs using directed graphs as chromosomes [19,20].

2.1.1. Prototype tree-based method

Prototype tree-based methods regard all individuals as α-ary perfect trees, where α is the maximum number of arguments among function nodes, and translate them to one dimensional arrays, and then tree structures are optimized by applying EDAs to the translated arrays. Prototype tree-based methods attract considerable attention because they can easily exploit existing EDAs.

The first prototype tree-based method is probabilistic incremental program evolution (PIPE) [31], which is an extension of population-based incremental learning (PBIL) [1] for tree structures. PIPE is weak against problems with dependencies between nodes because PIPE assumes that each node is independent of the others. Estimation of distribution programming (EDP) [38] estimates dependencies between nodes, using Bayesian networks. However, EDP is weaker than other methods with structural learning because EDP estimates only fixed parent–child relationships in tree structures. Extended compact GP (ECGP) [33] bases on extended compact GA (ECGA) [9]. ECGP estimates the multivariate dependencies among nodes using the minimum description length (MDL) principle but ECGP cannot estimate building blocks with practical size because of the large number of symbols in GP. Bayesian optimization algorithm (BOA) programming (BOAP) [22] is an application of BOA to tree structures and it uses a zigzag tree as chromosome. Program optimization with linkage estimation (POLE) [12] also estimates the multivariate dependencies among nodes using Bayesian networks.

The conventional prototype tree-based methods listed above employ PLS as sampling. Therefore, those methods waste a part of learning because PLS does not guarantee that MPSs, which best reflect the learned probabilistic models, will be generated at each generation. In order to overcome this deficiency, we propose an efficient sampling method to generate MPSs at each generation.

2.2. EDAs and PMBGPs with loopy belief propagation

In a field of EDAs, several algorithms using LBP have been hitherto proposed. Those methods focus on sampling the solution with the highest joint probability. It is difficult to calculate joint probabilities directly for graphical models with complex graph structure, which appear frequently in EDA and PMBGP. However, we can calculate approximate joint probability from approximate *local* joint probability easily for those complex graphs. The approximate *local* joint probability is often called as message. The key idea of LBP is message passing, iteration of message updating for getting more accurate approximation of joint probabilities.

Ref. [24] uses EBNA [5] as a foundation and simply applies LBP to generate one individual (MPS) whereas the rest of individuals are generated by PLS. Ref. [24] applies normal EBNA and EBNA-LBP to the Ising problem and shows that LBP boosts the best fitness value in the latter part of the search. Ref. [21] proposes Loopy Substructural Local Search (Loopy SLS), which employs local fitness as the values of factors and all possible individuals are carried over to the next generation if message passing does not converge. Ref. [21] uses BOA [27] as a base method and applies normal BOA, BOA with standard LBP (the same as in [24]) and BOA with Loopy SLS to the trap function. Ref. [21] concludes that Loopy SLS is better than standard LBP when population size is large.

We have already proposed the application of LBP in the contexts of PMBGPs [34], however, the prior work has only shown the effectiveness of LBP in view of the number of fitness evaluations and has not studied how LBP works. In order to discuss roles of LBP in the search process of PMBGP, the present paper applies POLE-BP to three benchmark tests and analyzes not only the number of fitness evaluations but also fitness and tree structures generated by LBP. One of the main contributions in the present paper is the detailed analysis of fitness and tree structures that has not been examined in our prior work [34].

3. The proposed method: POLE-BP

We briefly describe POLE-BP [34] in this section. POLE-BP is the first approach combining PMBGP and LBP. POLE-BP introduces LBP to the sampling process of Program Optimization with Linkage