Theoretical Computer Science 738 (2018) 53–64

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Belief propagation for the maximum-weight independent set and minimum spanning tree problems ☆

Kamiel Cornelissen, Bodo Manthey

University of Twente, Department of Applied Mathematics, The Netherlands

A R T I C L E   I N F O

Article history:
Received 11 April 2017
Received in revised form 19 April 2018
Accepted 24 April 2018
Available online 2 May 2018
Communicated by P. Krysta

Keywords:
Belief propagation
Independent sets
Minimum spanning trees

A B S T R A C T

The belief propagation (BP) algorithm is a message-passing algorithm that is used for solving probabilistic inference problems. In practice, the BP algorithm performs well as a heuristic in many application fields. However, the theoretical understanding of BP is limited. To improve the theoretical understanding of BP, the BP algorithm has been applied to many well-understood combinatorial optimization problems. In this paper, we consider BP applied to the maximum-weight independent set (MWIS) and minimum spanning tree (MST) problems.

Sanghavi et al. (2009) [12] applied the BP algorithm to the MWIS problem. We denote their algorithm by BP-MWIS. They showed that if the LP relaxation of the MWIS problem has a unique integral optimal solution and BP-MWIS converges, then BP-MWIS finds the optimal solution. Also, they showed that if the LP relaxation has a non-integral optimal solution, then BP-MWIS does not converge. In this paper, we precisely characterize the graphs for which BP-MWIS is guaranteed to find the optimal solution, regardless of the node weights.

Bayati et al. (2008) [2] applied the BP algorithm to the MST problem. We denote their algorithm by BP-MST. They showed that if BP-MST converges, then it finds the optimal solution. In this paper, however, we provide an instance for which BP-MST does not converge. Also, since this instance is small and simple, we believe that BP-MST does not converge for most instances encountered in practice.

1. Introduction

The belief propagation (BP) algorithm is a message-passing algorithm that is used for solving probabilistic inference problems on graphical models. It was proposed by Pearl in 1988 [8]. Typical graphical models to which BP is applied are Bayesian networks, Markov random fields, and factor graphs. In this paper, we consider the max-product variant of BP (or the functionally equivalent min-sum variant), which is used to compute maximum *a posteriori* probability (MAP) estimates.

Recently, BP has experienced great popularity. It has been applied in many fields, such as machine learning, image processing, computer vision, and statistics. For an introduction to BP and several applications, we refer to Yedidia et al. [17]. There are two main reasons for the popularity of BP. First, it is widely applicable and easy to implement because of its simple and iterative message-passing nature. Second, it performs well in practice in numerous applications [14,16].

☆ This research was supported by NWO grant 613.001.023.
E-mail addresses: kamielcornelissen@gmail.com (K. Cornelissen), b.manthey@utwente.nl (B. Manthey).

https://doi.org/10.1016/j.tcs.2018.04.046
0304-3975/

If the graphical model is tree-structured, BP computes exact MAP estimates. However, if the graphical model contains cycles, the convergence and correctness of BP have been shown only for specific classes of graphical models. To improve the general understanding of BP and to gain new insights about the algorithm, it has recently been tried to rigorously analyze the performance of BP as either a heuristic or an exact algorithm for several combinatorial optimization problems. Amongst others, it has been applied to the maximum-weight matching (MWM) problem [1,3,4,9,10], the minimum spanning tree (MST) problem [2], the minimum-cost flow (MCF) problem [4,7], the maximum-weight independent set (MWIS) problem [11,12], and the 3-coloring problem [5]. BP has even been used to analyze the satisfiability threshold [6]. The reason to consider BP applied to these combinatorial optimization problems is that these optimization problems are well understood. This facilitates a rigorous analysis of BP, which is often difficult for other applications.

In this paper, we consider BP applied to the MWIS and the MST problem. Sanghavi et al. [12] introduced a variant of BP for the MWIS problem, which we denote by BP-MWIS. They showed that BP-MWIS does not converge if the LP relaxation of the problem has a non-integral optimal solution. Also, they showed that even if the LP relaxation of the problem has a unique integral optimal solution, BP-MWIS is not guaranteed to converge. In this paper we characterize precisely the graph structures for which BP-MWIS is guaranteed to work well. This means that we characterize the graph structures for which BP-MWIS is guaranteed to converge to the correct solution irrespective of the node weights (as long as the MWIS is unique). We show (Section 3) that the graphs for which BP-MWIS converges to the correct solution for all possible node weights are exactly those graphs that contain at most one even cycle and no odd cycles.

Bayati et al. [2] introduced a variant of BP for the MST problem, which we denote by BP-MST. The MST problem is easily solvable using a variety of algorithms. Still, it is interesting to analyze the performance of BP applied to the MST problem since the MST problem has a global connectivity constraint. This is in contrast to, for example, the MWM, MCF, and MWIS problems, which only have local constraints. Bayati et al. showed the following positive result for BP-MST: if BP-MST converges, then it converges to the correct solution. In this paper, we show a negative result for BP-MST. In Section 4, we show a small instance for which BP-MST does not converge. In addition, the property of this instance that ensures that BP-MST does not converge is quite general and carries over to many other instances. Therefore, we believe that BP-MST does not converge for most instances in practice.

The rest of this paper is organized as follows. First we introduce the MWIS (Section 1.1) and MST (Section 1.2) problems. In Section 2, we introduce the BP algorithm and the variants for the MWIS problem by Sanghavi et al. [12] and the MST problem by Bayati et al. [2]. In Section 3 we state our results for BP-MWIS. Finally, in Section 4 we state our results for BP-MST.

To conclude this section, we introduce some notation and assumptions. We denote the weight of a node v by $w(v)$. Also, we denote the weight of a set of nodes V by $w(V)$. That is,

$$w(V) = \sum_{v \in V} w(v).$$

For a graph $G = (V, E)$ we define the *neighborhood* $N(v)$ of a node v as

$$N(v) = \{u \mid (u, v) \in E\}.$$

In this paper, we assume that all graphs are connected. For the MST problem we do this since no spanning tree exists for a disconnected graph. For the MWIS problem we do this since maximum-weight independent sets on disconnected graphs can be computed by separately computing maximum-weight independent sets on the individual components and then taking the union of those sets. Finally, as is commonly done, we assume that the optimal solutions for the MST and MWIS problems are unique, since it is well-known that BP does not converge for instances that have multiple optimal solutions for these problems [1–3,10].

1.1. Maximum-weight independent set problem

Let $G = (V, E)$ be an undirected weighted graph. An independent set S is a subset $S \subset V$ of nodes such that for every edge $(u, v) \in E$ at most one of u and v is in S. The MWIS problem consists of finding an independent set of maximum weight. A subset of nodes $S^* \subset V$ is an *MWIS* of G if and only if

$$S^* \in \operatorname{argmax}\{w(S) \mid S \text{ is an independent set of } G\}.$$

It is straightforward to formulate the MWIS problem as an integer program by identifying with each node $u \in V$ a binary variable $x_u \in \{0, 1\}$. Here $x_u = 0$ can be interpreted as x not being part of the independent set S, while $x_u = 1$ can be interpreted as x being part of S. The integer program contains constraints that prevent two neighboring nodes from both being included in S. The integer program (IP-MWIS) is as follows