The cavity approach for Steiner trees packing problems

Alfredo Braunstein\textsuperscript{1,2,3,4} and Anna Paola Muntoni\textsuperscript{5,1,6}

\textsuperscript{1} DISAT, Politecnico di Torino, Corso Duca Degli Abruzzi 24, Torino, Italy
\textsuperscript{2} Italian Institute for Genetic Medicine (form. HuGeF), Via Nizza 52, Torino, Italy
\textsuperscript{3} Collegio Carlo Alberto, Via Real Collegio 1, Moncalieri, Italy
\textsuperscript{4} INFN Sezione di Torino, Via P. Giuria 1, I-10125 Torino, Italy
\textsuperscript{5} Laboratoire de physique théorique, Département de physique de l’ENS, École normale supérieure, PSL University, Sorbonne Université, CNRS, 75005 Paris, France

E-mail: alfredo.braunstein@polito.it and anna.muntoni@polito.it

Received 16 January 2018
Accepted for publication 16 October 2018
Published 7 December 2018

Online at stacks.iop.org/JSTAT/2018/123401
https://doi.org/10.1088/1742-5468/aaeb3f

Abstract. The belief propagation approximation, or cavity method, has been recently applied to several combinatorial optimization problems in its zero-temperature implementation, the max-sum algorithm. In particular, recent developments to solve the edge-disjoint paths problem and the prize-collecting Steiner tree problem on graphs have shown remarkable results for several classes of graphs and for benchmark instances. Here we propose a generalization of these techniques for two variants of the Steiner trees packing problem where multiple ‘interacting’ trees have to be sought within a given graph. Depending on the interaction among trees we distinguish the vertex-disjoint Steiner trees problem, where trees cannot share nodes, from the edge-disjoint Steiner trees problem, where edges cannot be shared by trees but nodes can be members of multiple trees. Several practical problems of huge interest in network design can be mapped into these two variants, for instance, the physical design of very large scale integration (VLSI) chips.

The formalism described here relies on two components edge-variables that allows us to formulate a massage-passing algorithm for the V-DStP and two algorithms for the E-DStP differing in the scaling of the computational time with respect to some relevant parameters. We will show that through one of the
two formalisms used for the edge-disjoint variant it is possible to map the max-
sum update equations into a weighted maximum matching problem over proper
bipartite graphs. We developed a heuristic procedure based on the max-sum
equations that shows excellent performance in synthetic networks (in particular
outperforming standard multi-step greedy procedures by large margins) and on
large benchmark instances of VLSI for which the optimal solution is known, on
which the algorithm found the optimum in two cases and the gap to optimality
was never larger than 4%.

**Keywords:** message-passing algorithms, optimization over networks

---

**Contents**

1. Introduction 3
2. Two Steiner packing problems 5
3. An arborescent representation 5
   3.1. Branching model ................................................................. 6
   3.2. The flat formalism ................................................................. 7
       3.2.1. Constraints for the vertex-disjoint Steiner trees problem .... 7
       3.2.2. Constraints for the edge-disjoint Steiner trees problem ......... 8
4. Boltzmann distribution and marginals 9
5. The cavity equations 11
   5.1. Vertex-disjoint Steiner trees problem ........................................ 11
   5.2. Edge-disjoint Steiner trees problem .......................................... 13
       5.2.1. Neighbors occupation formalism ........................................ 13
       5.2.2. Mapping into a matching problem ..................................... 15
   5.3. The parameter $D$ .............................................................. 17
6. Max-sum for loopy graphs 17
   6.1. Reinforcement ................................................................. 18
   6.2. Max-sum based heuristics .................................................... 18
       6.2.1. Shortest path tree .......................................................... 18
       6.2.2. Minimum spanning tree ................................................. 19
7. Numerical results 19
   7.1. Greedy algorithm ............................................................ 20
   7.2. Fully connected graphs ...................................................... 20
       7.2.1. Uncorrelated edge weights ............................................ 21
       7.2.2. Correlated edge weights .......................................... 21
       7.2.3. Fixed fraction of terminals ........................................ 21
   7.3. Regular graphs .............................................................. 21
       7.3.1. Energy as a function of the degree .................................. 22
       7.3.2. Energy as a function of the number of communications .... 23

https://doi.org/10.1088/1742-5468/aaeb3f 2
1. Introduction

The minimum Steiner tree problem (MStP) is an important combinatorial problem that consists in finding a connected sub-graph within a given weighted graph, able to span a subset of vertices (called terminals) with minimum cost. It is easy to see that if weights are strictly positive the sub-graph satisfying all these constraints must be a tree.

The decisional problem of determining whether a solution within a given cost bound exists is NP-complete (it was one of Karp’s original 21 problems [13]). The large difficulty of the MStP can be seen to arise from the large space of subsets of non-terminal vertices (Steiner nodes). There exist several variants and generalizations of the MStP; one of the most studied is the prize-collecting Steiner problem (PCStP) that have many applications in network technologies, such as optimal heating and optical fibers distribution [14], in biology, e.g. in finding signal pathways in a cell [4]. In the prize-collecting variant the notion of terminals is relaxed so that every vertex has an associated prize (or reward). The prize of included nodes is counted negatively in the solution cost (so that profitable vertices with positive reward lower the total cost). In this variant the cost of the optimal tree will be the best trade-off between prizes of included nodes and the cost of their connections given by edge-weights.

In this work we will address the Packing of Steiner trees problem where we aim at finding, within the same graph, multiple Steiner trees which span disjoints sets of terminals in its original and prize-collecting versions. We consider two different variants regarding the interaction among trees. In the vertex-disjoint Steiner trees problem (V-DStP), different trees cannot share vertices (and consequently they cannot share edges either); in the edge-disjoint Steiner tree problem (E-DStP) only edge sets are pairwise disjoint but nodes can be shared by different trees. Being generalizations of PCStP, both problems are NP-hard; from a practical point of view the packing problems are more difficult than their single-tree counterpart as it can be seen from the fact that even finding feasible solutions, i.e. trees satisfying the interaction constraints (regardless the cost), is NP-hard [12]. In addition to its mathematical interest, a lot of attention is devoted to the practical solution of packing of Steiner trees problems since...
several layout design issues arising from very large scale integration (VLSI) circuits [9, 10, 15] can be mapped into these variants of the MStP [11, 12]. Integrated systems are composed by a huge number of logical units, called cells, typically arranged in 2D or 3D grids. Some specific cells, forming the so-called net, must be connected to one another in order to satisfy some working conditions. The physical design phase of these circuits addresses the problem of connecting each element of a net minimizing some objective function, namely the power consumption induced by the wires of the connection. It can be easily seen that connecting the cells of a net at minimum power consumption is equivalent to solve a MStP on a 2D or 3D grid graph. Thus, the problem of concurrently connecting multiple and disjoint sets of nets can be easily mapped into a V-DStP or an E-DStP. The most common approaches to these combinatorial optimization problems rely on linear programming formulations, for instance, the multi-commodity flow model [12].

In this work we devise three different models to represent these two problems: one for V-DStP and two for E-DStP, the first one more suitable for graphs where the density of terminals is low and the second for instances with low graph connectivity. We attempt their solution through the cavity method of statistical physics and its algorithmic counterpart, the belief propagation (BP) iterative algorithm (or rather, its zero-temperature limit, the max-sum (MS) algorithm [16, 17]). This technique is an approximation scheme first developed to study disordered systems and nowadays applied to a wide range of optimization problems. Once a proper set of variables is defined, the optimization problem, i.e. the constrained minimization of a cost function, can be mapped into the problem of finding the ground-state(s) of a generalized system with local interactions. Ground-states can be investigated through observables related to the Boltzmann–Gibbs distribution at zero temperature but, in most of the interesting cases, their exact evaluation involves impractical computations. MS consists in iterating closed massage-passing equations on a factor graph, closely related to the original graph, that, at convergence, provide an estimate of the marginal probability distribution of the variables of interest.

The cavity method can be proven to be exact on tree graphs (and also in some models on random networks in the asymptotic limit) but nevertheless in practice reaches notable performances on arbitrary graphs. It should be noted that in a simplified version of the problem (the minimum spanning tree), fixed points of max-sum can be proven to parametrize the optimal solution [6]. As usual, the iterative solution of the max-sum equations involve the solution of a related problem in a local star-shaped sub-graph which for some of these models is not trivial (i.e. its naive solution is exponentially slow in the degree). We devise a mapping of the problem into a minimum matching problem that can be solved in polynomial time in the degree (leading e.g. to linear time per iteration on Erdős–Rényi random graphs).

In combination with these three initial models, a variant called the flat formalism, borrowed from [8], can be independently included, leading to six different model combinations for the two problems. The flat formalism is more suitable for graphs with large diameter and/or few terminals as it allows to reduce considerably the solution space. Interestingly, the resulting flat models can be seen as generalizations of both [8] and [1, 3] as the edge-disjoint path problems in the last two publications can be seen as a packing of Steiner trees problem in which each tree has exactly two terminals.
With these algorithmic tools on hand, we perform numerical simulations of complete, Erdős–Rényi and random regular graphs and on benchmark instances of V-DStP arising from the VLSI design problem.

2. Two Steiner packing problems

Given a graph $G = (V, E)$ whose vertices have non-negative real prizes $\{c^\mu_i : i \in V, \mu = 1, \ldots, M\}$ and whose edges have real positive weights $\{w^\mu_{ij} : (i, j) \in E, \mu = 1, \ldots, M\}$, we consider the problem of finding $M$ connected subgraphs $G^\mu = (V^\mu, E^\mu)$ spanning disjoint sets of terminals $\{T^\mu \subseteq V^\mu, \mu = 1, \ldots, M\}$ that minimize the following cost or energy function

$$H = \sum_\mu \left[ \sum_{i \in V \setminus V^\mu} c^\mu_i + \sum_{(i,j) \in E^\mu} w^\mu_{ij} \right]. \quad (1)$$

This definition of the cost is extremely general: node prizes and edge costs can depend on sub-graph $\mu$. For directed graphs, we can admit $w^\mu_{ij} \neq w^\mu_{ji}$ by considering oriented trees (the trees we will consider will be ultimately rooted and thus oriented). In the following we refer to vertices with strictly positive prizes as (generalized) terminals in analogy with the MStP. This particular case can be integrated in our formalism by imposing $c^\mu_i = +\infty$ if node $i$ is a (true) terminal of tree $\mu$ (it suffices to have a large enough value for $c^\mu_i$ instead of $+\infty$), and $c^\mu_j = 0, \forall \mu$ for any non-terminal node $j \in V$. Since we interpret the solution-trees as networks that allow terminals to ‘communicate’ we will refer to each sub-graph $G^\mu$ as a ‘communication’ $\mu$ flowing within the graph.

Subsets $G^\mu$ must satisfy some interaction constraints depending on the packing variant we are considering. In the vertex-disjoint Steiner trees problem (V-DStP), vertex-sets $V^\mu$ must be pairwise disjoint, i.e. $V^\mu \cap V^\nu = \emptyset$ if $\mu \neq \nu$ and, consequently, also edge sets will be pairwise disjoint. In the edge-disjoint Steiner trees problem (E-DStP), only edge sets must be pairwise disjoint, i.e. $E^\mu \cap E^\nu = \emptyset$ if $\mu \neq \nu$, but vertex sets can overlap.

3. An arborescent representation

To deal with these two combinatorial optimization problems we will define a proper set of interacting variables defined on a factor graph which is closely related to the original graph $G$. The factor graph is the bipartite graph of factors (or compatibility functions) and variables, in which an edge between a factor and a variable exists if the function depends on the variable. More precisely, to each vertex $i \in V$ we associate a factor node $\psi_i$, and to each edge $(i, j) \in E$ we associate a two components variable $(d_{ij}, \mu_{ij}) \in \{-D, \ldots, 0, \ldots, D\} \times \{0, \ldots, M\}$. Our choice of the edge-variables is similar to the one adopted in [8] but here, in addition to a ‘depth’ component, we introduce a ‘communication’ variable $\mu_{ij}$ by which we label edges forming different trees.
Compatibility functions \( \psi_i \) are defined in a way that allowed configurations of variables \((d, \mu) \in \{(d_{ij}, \mu_{ij}) : (i, j) \in E\}\) are in one to one correspondence to feasible solutions of the Vertex-disjoint or Edge-disjoint variant of the Steiner trees problem. In particular, in order to ensure Steiner sub-graphs to be trees, i.e. to be connected and acyclic, we impose local constraints on variables \( d_i = \{d_{ij} : j \in \partial i\} \) and \( \mu_i = \{\mu_{ij} : j \in \partial i\} \) through compatibility functions \( \psi_i (d_i, \mu_i) \) that will be equal to one if the constraints are satisfied or zero otherwise.

Consider a solution to the V-DStP or the E-DStP. Each variable \( \mu_{ij} \) takes value from the set \( \{0, 1, \ldots, M\} \) and denotes to which sub graph, if any, does the edge \((i, j)\) belongs; the state \( \mu_{ij} = 0 \) will conventionally mean that no tree employs the edge \((i, j)\). Components \( d_{ij} \in \{-D, \ldots, 0, \ldots, D\} \) have a meaning of ‘depth’ or ‘distance’ within the sub-graph. Value \( d_{ij} = 0 \) conventionally means that such edge is not employed by any communication and thus it is admitted if and only if the associated \( \mu_{ij} = 0 \).

Being the interactions among nodes different as we deal with the V-DStP or the E-DStP, we will define two different compatibility functions, \( \psi^V_i \) and \( \psi^E_i \), for the two problems. Both functions will be written with the help of a single-tree compatibility function \( \psi^\mu_i \) for two different formulations of the constraints, the branching and the flat model.

### 3.1. Branching model

Let us consider a sub-graph \( G_\mu \) constituting part of the solution for the V-DStP or the E-DStP. For each node \( i \in V_\mu \), the variable \( d_{ij} \) measures the length, in ‘steps’, of the unique path from node \( i \) to root \( r_\mu \) passing through \( j \in \partial i \). Variable \( d_{ij} \) will be strictly positive (negative) if \( j \) is one step closer (farther) than \( i \) to root \( r_\mu \). Thus, every edge will satisfy the anti-symmetric condition \( d_{ij} = -d_{ji} \) and \( \mu_{ij} = \mu_{ji} \). A directed tree structure is guaranteed if, mathematically, the following single-tree compatibility function

\[
\psi^\mu_i (d_i, \mu_i) = \prod_{j \in \partial i} \delta_{\mu_j, 0} \delta_{d_{ji}, 0} + \sum_{d > 0} \sum_{j \in \partial i} \left[ \delta_{\mu_j, \mu_{ij}} \delta_{d_{ji}, -d} \prod_{k \in \partial i \setminus j} (\delta_{\mu_k, \mu_{ki}} \delta_{d_{ki}, d + 1} + \delta_{\mu_k, 0} \delta_{d_{ki}, 0}) \right]
\]  

(2)

equals to 1 for every nodes in the graph. Here \( \delta_{x,y} \) is the discrete Kronecker delta function equal to 1 if \( x = y \) and 0 otherwise. The first part of the equation describes a feasible assignment for a node \( i \) that does not participate to any communication: its local variables satisfy \( d_i = 0 \) and \( \mu_i = 0 \). The second addend considers a second case in which \( i \) is member of tree \( \mu \) at distance \( d \) from the root; in this case there will exist only one neighbor \( j \), one step closer to the root than \( i \), such that \( d_{ij} = d \) (as a consequence \( d_{ji} = -d_{ij} < 0 \)) and \( \mu_{ij} = \mu \). All the remaining neighbors \( k \in \partial i \setminus j \) may not be members of the communication (\( \mu_{ki} = 0 \), \( d_{ki} = 0 \)) or being part of \( G_\mu \) (\( \mu_{ki} = \mu \)) as children of \( i \). In the latter any \( k \) is one step further than \( i \) from the root \( r_\mu \), and thus their depths must be increased of one unit, namely \( d_{ki} = d + 1 \). Notice that for a given feasible assignment of the local variables the summations over the possible positive depths and neighbors in (2) reduce to a single term that corresponds to the unique \( j \in \partial i \) such that \( d_{ji} < 0 \); in this case \( \psi^\mu_i (d_i, \mu_i) = 1 \). Instead, for all unfeasible assignments \( \psi^\mu_i (d_i, \mu_i) = 0 \).

In the following this representation of the tree structure will be referred as the branching formalism and it will be denoted using an apex ‘\( b \)’ in its compatibility function as in (2).
3.2. The flat formalism

The diameter of solutions representable using the branching model strongly depends on the value of the parameter $D$ which is the maximum allowed distance from any leaf and the root of the tree. A small value of the depth parameter can certainly prevent the representation of more elongated and, possibly, more energetically favored solutions but, at the same time, a big value of $D$ will significantly slow down the computation of the compatibility function in (2). The flat model relaxes the depth-increasing constraint in the sense that, under certain conditions, it allows chains of nodes, within the solution, with equal depth. According to a flat representation, the depth variable increases of one unity if a node $i$ is a terminal node or there exist two or more neighbors connected to $i$ within a sub-graph $G_{\mu}$, i.e. the degree of node $i$ within the sub-graph is more than two.

It can be shown [8] that for $D = T$, where $T = |T_{\mu}|$ is the number of terminals per communication, these constraints admit all feasible trees plus some extra structures which contain disconnected cycles with no terminals and thus are energetically disfavored. This additional ‘flat’ assignment applies to only non-terminal and non-branching nodes in the solution and the corresponding compatibility function can be stated as follows:

$$\psi_{i}^{\mu,f}(d_{i},\mu_{i}) = \delta_{\delta_{i},0}^{\mu} + \sum_{d > 0} \delta_{\mu,i}^{\mu} \delta_{d_{i} - d - 1} \delta_{\mu,i}^{\mu} \delta_{\delta_{i},d} \prod_{m \in \partial i \setminus \{k,l\}} \delta_{\mu,m}^{\mu} \delta_{d_{m},0}^{\mu}.$$  

More precisely, if node $i$ is not a terminal of sub-graph $\mu$ but it is a Steiner node at distance $D \leq d < 0$, there exists one of its neighbors $k$ (its parent within the solution) such that $d_{k} = d$. Node $i$ cannot be a leaf and therefore there must be a child of $i$, a node $l \in \partial i \setminus k$, at the same depth $d$. All the remaining neighbors $m \in \partial i \setminus \{k,l\}$ must have $d_{m,i} = 0$ to preserve the chain structure. As for the compatibility function in (2) and (3) returns 1 for an assignment that satisfies the ‘flat’ constraints: among all the addends only the term in which two neighbors have the same non-zero depth (with opposite sign) will survive. For all assignments not satisfying this property we will have $\psi_{i}^{\mu,f}(d_{i},\mu_{i}) = 0$. Finally, the single-tree compatibility function $\psi_{i}^{\mu} = 1 - \left(1 - \psi_{i}^{\mu,b}\right)\left(1 - \psi_{i}^{\mu,f}\right)$ of configuration satisfying exactly one of the two constraints can be written as

$$\psi_{i}^{\mu}(d_{i},\mu_{i}) = \psi_{i}^{\mu,b}(d_{i},\mu_{i}) + \psi_{i}^{\mu,f}(d_{i},\mu_{i})$$

since $\psi_{i}^{\mu,b} \psi_{i}^{\mu,f} \equiv 0$.

3.2.1. Constraints for the vertex-disjoint Steiner trees problem. In the V-DStP a node $i$ can belong to none or at most one sub-graph $G_{\mu}$ and, as a consequence, its neighbor edges can either participate to the same communication or be unused. For this reason we can consider communication wise the topological constraints applied to the neighborhood of each node. The compatibility function $\psi_{i}^{V}$ can be then expressed as the sum over all possible trees of a single-tree compatibility function $\psi_{i}^{\mu}$ in (4).

$$\psi_{i}^{V}(d_{i},\mu_{i}) = \sum_{\mu = 1}^{M} \psi_{i}^{\mu}(d_{i},\mu_{i}).$$

Being a non-terminal node, the path connecting $i$ to the closest terminal does not carry any advantage in terms of connection and only increases the cost of the solution.
3.2.2. Constraints for the edge-disjoint Steiner trees problem. Differently to the V-DStP, in the E-DStP a vertex can belong to an arbitrary number of communications (including zero) with the constraint that the local tree structure must be concurrently satisfied for each communication. If the node does not participate in the solution we must admit configurations in which \( d_i = 0 \) if \( \mu_i = 0 \). For the remaining cases, if some neighbors \( k \in \partial i \) is a members of a Steiner tree \( \mu \), its distances \( d_{ki} \) will be different from zero if \( \mu_{ki} = \mu \), and, additionally, they will satisfy the topological constraints. We can mathematically express such conditions through the compatibility functions

\[
\psi^E_{i,b}(d_i, \mu_i) = \prod_{\mu=1}^{M} \left[ \prod_{k \in \partial i} \delta_{d_{ki},\delta_{\mu_{ki}},0} + \sum_{d>0} \sum_{k \in \partial i} \delta_{d_{ki},\delta_{\mu_{ki}},d} \prod_{l \in \partial i \setminus k} \left( \delta_{d_{li},\delta_{\mu_{li}},d+1} + \delta_{d_{li},\delta_{\mu_{li}},0} \right) \right] 
\]

for the branching model and

\[
\psi^E_{i,f}(d_i, \mu_i) = \prod_{\mu} \left[ \delta_{d_{\mu},0} \sum_{d>0} \sum_{k \in \partial i} \delta_{d_{ki},\delta_{\mu_{ki}},d} \sum_{l \in \partial i \setminus k} \delta_{d_{li},\delta_{\mu_{li}},d} \prod_{m \in \partial i \setminus \{k,l\}} \delta_{d_{mi},\delta_{\mu_{mi}},0} \right] 
\]

for the flat model. Notice that we can express

\[
\psi^E_{i}(d_i, \mu_i) = \psi^E_{i,b}(d_i, \mu_i) + \psi^E_{i,f}(d_i, \mu_i) 
\]

or eventually, if we define \( \tilde{d}_{ki} = d_{ki}\delta_{\mu_{ki},\mu} \), we can rephrase it as a product over single-tree compatibility functions (compare to (5))

\[
\psi^E_{i}(d_i, \mu_i) = \prod_{\mu=1}^{M} \psi^\mu_{i}(d_i, \mu_i). 
\]

Some examples of feasible assignments of variables for both branching and flat models are shown in figures 1 and 2. On the left (figure 1(a)) we see one instance of the V-DStP represented through the branching formalism and containing two sub-graphs, the ‘red’ having root ‘4’ and the ‘blue’ rooted at node ‘3’; on the right (figure 1(b)) two ‘red’ and ‘blue’ edge-disjoint Steiner trees, rooted at ‘10’ and ‘5’ respectively. Roots are represented as square nodes in contrast to circle colored nodes that are terminals. Edges employed in the solutions are figured as arrows whose labels denote the value of the (positive) depth component while the color mirrors the communication component. In agreement with our branching representation we see that depth components increase as we cover the solution from the root to the leaves for both problems. In the figure on the left, nodes of the vertex-disjoint trees are members either of the ‘red’ or the ‘blue’ trees but such constraint is relaxed in the figure 1(b) for E-DStP. In fact, we allow node ‘9’ to be a terminal of communication ‘blue’ and a Steiner node of the ‘red’ tree as two incident edges, (10, 9) and (9, 8), belong to the ‘red’ solution.

To underline the differences between the branching and the flat formalism, we picture in figures 2(a) and (b) the same solution to the V-DStP on a grid graph using both models. According to the branching formalism, we see in figure 2(a) that we need a minimum depth of \( D = 7 \) to allow all terminals of the ‘blue’ communication to reach the root node ‘1’. Notice that since the tree is actually a chain of nodes, the same solution can be represented in the flat formalism using \( D = 3 \) as shown in figure 2(b).
In fact, only each time we reach a terminal node the depth variable increases of one unit. Depth variables must increase in another condition, precisely when we reach a branching point: this is exactly what happens in the neighborhood of node ‘23’ of the ‘red’ solution in figure 2(b).

4. Boltzmann distribution and marginals

The formalism introduced above allows us to map each solution of the packing of Steiner trees to a certain assignment of variables \( d = \{d_{ij} : (i, j) \in E\} \) and \( \mu = \{\mu_{ij} : (i, j) \in E\} \) of the associated factor graph. The cost function in (1) can be then expressed in terms of the new variables as

\[
H(d, \mu) = \sum_{\mu=1}^{M} \left[ \sum_{i \in V} \psi_{i}(d_{i}, \mu_{i}) \prod_{k \in \partial i} (1 - \delta_{\mu_{ki}, \mu}) + \sum_{d_{ij} > 0, \mu_{ij} = \mu} w_{ij} \right]
\]

(10)

where, for sake of simplicity, we consider the ‘homogeneous’ case \( w_{ij} = w_{ij} \), and \( \psi_{i} \) can be either equal to \( \psi_{V}^{i} \) or \( \psi_{E}^{i} \).

The Boltzmann–Gibbs distribution associated with the energy \( H(d, \mu) \) is given by

\[
P_{\beta}(d, \mu) = \frac{\prod_{i} \psi_{i}(d_{i}, \mu_{i}) e^{-\beta H(d, \mu)}}{Z_{\beta}}
\]

(11)

in which \( \beta \) is a positive parameter, called the ‘inverse temperature’ as in the statistical mechanics framework, and the normalization constant

\[
Z_{\beta} = \sum_{d, \mu} \prod_{i} \psi_{i}(d_{i}, \mu_{i}) e^{-\beta H(d, \mu)}
\]

is the partition function. Configurations of the variables that do not satisfy the topological constraints will have zero probability measure, whereas all other configurations will be weighted according to the sum of weights of used edges and of the penalties of non-employed nodes. In the limit \( \beta \to +\infty \) the distribution will be concentrated in the configuration that minimizes \( H(d, \mu) \) (the ground state of the system) that are exactly the solutions of the optimization problems. As a consequence also the marginal probability densities, for \( \beta \to +\infty \), associated with each edge

\[
P_{ij}^{\beta \to +\infty}(d_{ij}, \mu_{ij}) = \sum_{d, \mu} P_{\beta \to +\infty}(d, \mu) \delta_{d_{ij}, d_{ij}} \delta_{\mu_{ij}, \mu_{ij}} \quad \forall (i, j) \in E
\]

(12)

will be concentrated and therefore it suffices to estimate them to gain knowledge about the optimal configuration. Thus our assignment of the variables will be given by

\[
(d_{ij}^{*}, \mu_{ij}^{*}) = \arg \max_{(d_{ij}, \mu_{ij})} P_{ij}^{\beta \to +\infty}(d_{ij}, \mu_{ij}) \quad \forall (i, j) \in E.
\]

(13)

Unfortunately the computation of (12) is impractical as it would require the calculation of a sum of an exponential number of terms. We seek to estimate these marginals

https://doi.org/10.1088/1742-5468/aaeb3f
via the cavity method approach. We report here a standard formulation of the cavity equations and we refer the interested reader to [16] for the detailed derivation. At finite $\beta$ the BP equations on our factor graph are:

$$
\begin{align*}
  m_{ij} (d_{ij}, \mu_{ij}) &= \frac{1}{Z_{ij}} \sum_{\{d_{k_i}, \mu_{k_i}\}} \psi_i (d_i, \mu_i) e^{-\beta \sum_k \sum_{\delta_{ij} = 1} \Pi_{k \in \partial_i} (1 - \delta_{\mu_{k_i}, \mu_{ij}}) \prod_{k \in \partial_i \setminus j} n_{ki} (d_{ki}, \mu_{ki})} \\
  n_{ki} (d_{ki}, \mu_{ki}) &= e^{-\beta w_{ki} \mathbb{1}_{[d_{ki} > 0]}} m_{ki} (d_{ki}, \mu_{ki})
\end{align*}
$$

where

$$
Z_{ij} = \sum_{\{d_{ij}, \mu_{ij}\}} m_{ij} (d_{ij}, \mu_{ij})
$$

is the normalization constant or ‘partial’ partition function. The functions $m_{ij}$ are called cavity marginals or ‘messages’, suggesting that some information is flowing on
edge \((i,j)\) within the factor graph from node \(i\) to node \(j\). In fact, the values of the messages \(m_{ij}\) are in some sense proportional to the probability of a particular assignment \((d_{ij}, \mu_{ij})\) for edge \((i,j)\) if the node \(j\) were temporarily erased from the graph.

The system of equations in (14) can be seen as fixed point equations that can be solved iteratively. Starting from a set of initial cavity marginals at time \(t = 0\), we iterate the right-hand-side of (14) until numerical convergence to a fixed point is reached. At convergence we calculate an approximation to marginals in (12) via the cavity fields defined as

\[
M_{ij}(d_{ij}, \mu_{ij}) \propto n_{ij}(d_{ij}, \mu_{ij}) n_{ji}(-d_{ij}, \mu_{ij})
\] (15)

where the proportional sign denotes that a normalization constant is missing.

Cavity equations for optimization problems can be easily obtained by substituting the \(m_{ij}\) and \(M_{ij}\) with the variables \(h_{ij}(d_{ij}, \mu_{ij}) = \lim_{\beta \to \infty} \frac{1}{\beta} \log n(d_{ij}, \mu_{ij})\) and \(H_{ij}(d_{ij}, \mu_{ij}) = \lim_{\beta \to \infty} \frac{1}{\beta} \log M_{ij}(d_{ij}, \mu_{ij})\) into (14) and (15) that play the role of cavity marginals and fields in the zero-temperature limit; the resulting closed set of equations is known as the max-sum algorithm. At convergence we can extract our optimal assignment of variables by the computation of the decisional variables

\[
(d_{ij}^*, \mu_{ij}^*) = \arg \max_{(d_{ij}, \mu_{ij})} H_{ij}(d_{ij}, \mu_{ij})
\] (16)

\[
H_{ij}(d_{ij}, \mu_{ij}) = h_{ij}(d_{ij}, \mu_{ij}) + h_{ji}(-d_{ij}, \mu_{ij}) - C'
\] (17)

where \(C'\) is an additive constant that guarantees that normalization condition in the zero-temperature limit, i.e. \(\max_{(d_{ij}, \mu_{ij})} H_{ij}(d_{ij}, \mu_{ij}) = 0\), is satisfied. In practice, convergence is reached when the decisional variables computed as in (16) do not change after a predefined number of successive iterations (often 10–30). Notice that taking the \(\beta \to +\infty\) limit of the message-passing equations at finite \(\beta\) is not equivalent to the zero-temperature limit of the Boltzmann distribution in (11).

In the following section we will show how to derive equations for the cavity marginals and cavity fields, for finite \(\beta\) and in the limit \(\beta \to +\infty\), depending on we are dealing with the V-DStP or the E-DStP problem.

5. The cavity equations

5.1. Vertex-disjoint Steiner trees problem

To derive the belief propagation equations for the V-DStP problem suffices to impose \(\psi_i(d_i, \mu_i) = \psi_i^V(d_i, \mu_i)\) in (14). By a change of variables, we will determine a max-sum algorithm for this variant.

Equations for messages can be easily obtained by using the properties of Kronecker delta functions in \(\psi_i^V(d_i, \mu_i)\); the explicit derivation is reported in appendix A. We can differentiate three cases depending on we are updating messages \(n_{ij}\) for positive, negative or null depth \(d_{ij}\):
The cavity approach for Steiner trees packing problems

\[
\begin{align*}
\begin{cases}
m_{ij} (d, \mu) = m^b_{ij} (d, \mu) + m^f (d, \mu) & \forall d > 0, \mu \neq 0 \\
m_{ij} (d, \mu) = m^b_{ij} (d, \mu) + m^f (d, \mu) & \forall d < 0, \mu \neq 0 \\
m_{ij} (0, 0) = e^{-\beta \sum_{\nu \in \partial i} n_{ki} (0, 0)} + m^b_0 + m^f_0
\end{cases}
\end{align*}
\]

where \( m^b_{ij} (d, \mu) \), \( m^b_{ij} (d, \mu) \), \( m^f (d, \mu) \), and \( m^b_0 \), \( m^f_0 \) are defined as

\[
\begin{align*}
m^b_{ij} (d, \mu) &= \prod_{k \in \partial i \setminus j} [n_{ki} (d + 1, \mu) + n_{ki} (0, 0)] \\
m^f (d, \mu) &= \delta_{c_i,0} \sum_{k \in \partial i \setminus j} n_{ki} (d, \mu) \prod_{l \in \partial i \setminus \{j,k\}} n_{li} (0, 0) \\
m^b_{ij} (d, \mu) &= \sum_{k \in \partial i \setminus j} n_{ki} (d + 1, \mu) \prod_{l \in \partial i \setminus \{j,k\}} [n_{li} (d, \mu) + n_{li} (0, 0)] \\
m_0^b &= \sum_{\mu \neq 0} \sum_{d < 0} m^b_{ij} (d, \mu) \\
m_0^f &= \sum_{\mu \neq 0} \sum_{d < 0} \sum_{k \in \partial i \setminus j} n_{ki} (d, \mu) \sum_{l \in \partial i \setminus \{j,k\}} n_{li} (-d, \mu) \prod_{m \in \partial i \setminus \{k, l\}} n_{mi} (0, 0).
\end{align*}
\]

Replacing \( h_{ij} (d_{ij}, \mu_{ij}) = \lim_{\beta \to +\infty} n_{ij} (d_{ij}, \mu_{ij}) \) in (18) we obtain the max-sum equations:

\[
\begin{align*}
\begin{cases}
h_{ij} (d, \mu) = \max \left\{ h^b_{ij} (d, \mu), h^f_{ij} (d, \mu) \right\} & \forall d > 0, \mu \neq 0 \\
h_{ij} (d, \mu) = \max \left\{ h^b_{ij} (d, \mu), h^f_{ij} (d, \mu) \right\} & \forall d < 0, \mu \neq 0 \\
h_{ij} (0, 0) = \max \left\{ -\sum_{\nu \in \partial i} c^\nu_i + \sum_{k \in \partial i \setminus j} h_{ki} (0, 0), h^b_0, h^f_0 \right\}
\end{cases}
\end{align*}
\]

for

\[
\begin{align*}
h^b_{ij} (d, \mu) &= -w_{ij} + \sum_{k \in \partial i \setminus j} \max \{ h_{ki} (d + 1, \mu), h_{ki} (0, 0) \} \\
h^f_{ij} (d, \mu) &= -w_{ij} + \log \delta_{c_i,0} + \max_{k \in \partial i \setminus j} \left\{ h_{ki} (d, \mu) + \sum_{l \in \partial i \setminus j} h_{li} (0, 0) \right\} \\
h^b_{ij} (d, \mu) &= \max_{k \in \partial i \setminus j} \left[ h_{ki} (d + 1, \mu) - w_{ik} + \sum_{l \in \partial i \setminus \{j,k\}} \max \{ h_{li} (d, \mu), h_{li} (0, 0) \} \right] \\
h^f_{ij} (d, \mu) &= \log \delta_{c_i,0} + \max_{k \in \partial i \setminus j} \left\{ h_{ki} (d, \mu) - w_{ik} + \sum_{l \in \partial i \setminus j} h_{li} (0, 0) \right\} \\
h^b_0 &= \max_{\mu \neq 0} \max_{d < 0} h^b_{ij} (d, \mu) \\
h^f_0 &= \max_{\mu \neq 0} \max_{d < 0} \left[ \max_{k \in \partial i \setminus j} \max_{l \in \partial i \setminus \{j,k\}} h_{ki} (d, \mu) - w_{ik} + h_{li} (-d, \mu) + \sum_{m \in \partial i \setminus \{j,k,l\}} h_{mi} (0, 0) \right].
\end{align*}
\]
5.2. Edge-disjoint Steiner trees problem

As for the V-DSstP, the belief propagation equations for the E-DSstP can be computed imposing \( \psi_i(d_i, \mu_i) = \psi_i^E(d_i, \mu_i) \) into (14):

\[
m_{ij}(d_{ij}, \mu_{ij}) \propto \sum_{\{d_{ki}, \mu_{ki}\} : k \in \partial i} \psi_i^E(d_i, \mu_i) e^{-\beta \sum_{u} c_u^E \prod_{k \in \partial_i} (1-\delta_{\mu_{ki} u})} \prod_{k \in \partial_i \setminus j} n_{ki}(d_{ki}, \mu_{ki}).
\]

Instead of considering the cavity messages as in section 5.1, to compute (20) we will first define a partial partition function

\[
Z_i = \sum_{d_i, \mu_i} \psi_i^E(d_i, \mu_i) e^{-\beta \sum_{u} c_u^E \prod_{k \in \partial_i} (1-\delta_{\mu_{ki} u})} \prod_{k \in \partial_i} n_{ki}(d_{ki}, \mu_{ki})
\]

and then calculate the set of messages \( m_{ij}(d, \mu) \) (for all possible values of \( d \) and \( \mu \)) from \( i \) to \( j \) through (21) by temporarily setting the message that \( i \) received from \( j \) as \( n_{ji}(d_{ji}, \mu_{ji}) = \delta_{d_{ji}, \mu_{ji}} \). In fact, here \( m_{ij} \) has a unique non-zero value in the state that satisfies the anti-symmetric property, namely \( d_{ij} = d, \mu_{ij} = \mu; \) under this condition \( Z_i = m_{ij}(d_{ij}, \mu_{ij}) \) up to a normalization constant. Due to the explicit expression of \( \psi_i^E \) message-passing equations become intractable and, therefore, the update step of the algorithm cannot be efficiently implemented. In the following subsections we overcome this issue by proposing two different approaches for the computation of (21) where we make use of two different sets of auxiliary variables. The first formalism relies on ‘binary occupation’ variables that denote, for each node of the factor graph, if edges incident on it are used or not by any communication; as we will see the associated computation scales exponentially in the degree of the nodes. The second one consists in a mapping between the E-DSstP update equation and a weighted matching problem over bipartite graphs, that, in the \( \beta \to +\infty \), becomes a weighted maximum matching problem which can be solved efficiently. This implementation scales exponentially with respect to \( M \) but it may be more efficient for vertices with large degrees with respect to the first algorithm.

5.2.1. Neighbors occupation formalism. Suppose of associating with each vertex \( i \) a vector \( x = \{0, 1\}^{[k]} \) that denotes if edges incident on \( i \) are employed or unemployed within the solution. A feasible assignment of these auxiliary variables is guaranteed if, for every link \((i, k) \in E \) incident on \( i \), we impose \( x_k = 1 \) if the edge belongs to a tree (i.e. \( d_{ki} \neq 0 \) and consequently \( \mu_{ki} \neq 0 \) or \( x_k = 0 \) otherwise (for \( \mu_{ki} = 0, d_{ki} = 0 \). Variables \((d_i, \mu_i)\) must locally satisfy the following identity \( \prod_{k \in \partial i} [x_k = 1 - \delta_{d_{ki}, 0}] = 1 \) for every node \( i \in V \). If we insert this expression in (21) and we sum over all possible assignments of \( x \) variables we obtain

\[
Z_i = \sum_{d_i, \mu_i} \psi_i^E(d_i, \mu_i) e^{-\beta \sum_{u} c_u^E \prod_{k \in \partial_i} (1-\delta_{\mu_{ki} u})} \sum_x \prod_{j \in \partial_i} [x_j = 1 - \delta_{d_{ji}, 0}] \; n_{ji}(d_{ji}, \mu_{ji})
\]

\[
= \sum_x Z_x^M
\]

where \( Z_x^M \) is defined by computing the following expression for \( q = M \)
The cavity approach for Steiner trees packing problems

\[
Z_x^q = \sum_{\substack{d_i, \mu_k \\ \mu_k \in \partial i}} \psi_x^E (d_i, \mu_k) e^{-\beta \sum_{\mu} c_i^\mu \prod_{k \in \partial i} \prod_{k} [x_k = 1 - \delta_{d_k, 0}] n_{ki} (d_{ki}, \mu_k)}.
\]  

(24)

The computation of \(Z_x^q\) is then performed using the following recursion (the equivalence of (25) to (24) is proven in appendix B)

\[
Z_x^q = \sum_{y \in x} (g_y^0 + g_y^b + g_y^f) Z_y^{q-1}
\]

(25)

\[
Z_x^0 = e^{-\beta \sum_{\mu} c_i^\mu} \prod_{j \in \partial i} [n_{ki} (d + 1, q) + n_{ki} (0, 0)]
\]

(26)

where the auxiliary functions \(g_y^0, g_y^b, g_y^f\) are defined as

\[
g_y^0 = e^{-\beta c_i^\mu} \prod_{k \in \partial i} [n_{ki} (0, 0)]
\]

\[
g_y^b = \sum_{d > 0} \sum_{j \in \partial i} n_{ji} (-d, q) \prod_{k \in \partial i \setminus j} [n_{ki} (d + 1, q) + n_{ki} (0, 0)]
\]

\[
g_y^f = \delta_{c_i^\mu, 0} \sum_{d > 0} \sum_{j \in \partial i \setminus \mu_k} n_{ji} (-d, q) \prod_{k \in \partial i \setminus \mu_k} [n_{ki} (d, q) + n_{ki} (0, 0)]
\]

and the trace over \(y \leq x\) denotes all possible vectors \(y = \{0, 1\}^{\partial i}\) satisfying

\[
y_k = \begin{cases} y_k \leq x_k & \text{if } \mu_k \neq q \\ 0 & \text{if } \mu_k = q. \end{cases}
\]

(27)

Within the max-sum formalism, we can equivalently define a partial free entropy \(F_i = \lim_{\beta \to +\infty} \frac{1}{\beta} \log Z_i\) and express it as function of max-sum messages \(h_{ij} (d_{ij}, \mu_{ij}) = \lim_{\beta \to +\infty} \frac{1}{\beta} \log n_{ij} (d_{ij}, \mu_{ij})\) as

\[
F_i = \max_{\substack{d_i, \mu_i, \psi_x^E (d_i, \mu_i) = 1}} \max_x \left[ \sum_{k \in \partial i} \log [x_k = 1 - \delta_{d_k, 0}] + h_{ki} (d_{ki}, \mu_{ki}) - \sum_{\mu} c_i^\mu \prod_{k \in \partial i} (1 - \delta_{\mu_k, 0}) \right]
\]

(28)

where the function \(\sum_{k \in \partial i} \log [x_k = 1 - \delta_{d_k, 0}]\) takes value zero if variables satisfy the constraints or minus infinity otherwise. As in the BP formulation, we rewrite it as

\[
F_i = \max_x F_x^M
\]

(29)

with

\[
F_x^M = \max_{\substack{d_i, \mu_i, \psi_x^E (d_i, \mu_i) = 1}} \sum_{k \in \partial i} \left[ \log [x_k = 1 - \delta_{d_k, 0}] + h_{ki} (d_{ki}, \mu_{ki}) - \sum_{\mu} c_i^\mu \prod_{k \in \partial i} (1 - \delta_{\mu_k, 0}) \right].
\]

The computation can be performed recursively from

\[
https://doi.org/10.1088/1742-5468/aaeb3f
\]

J. Stat. Mech. (2018) 123401

14
The cavity approach for Steiner trees packing problems

\[ F^q_x = \max_{y \leq x} \{ F^y_{x-1} + \max \{ h_0, h_b, h_f \} \} \]  
(30)

\[ F^0_x = - \sum_{\mu} e^\mu_i + \log \| \mathbf{x} = 0 \| + \sum_{k \in \partial i} h_{ki} (0, 0) \]  
(31)

where

\[ h_0 = \sum_{k \in \partial i} h_{ki} (0, 0) - c^q_i \]  
(32)

\[ h_b = \max_{d > 0} \max_{k \in \partial i} \left[ h_{ki} (-d, q) + \sum_{\mu} \max \left\{ h_{li} (d + 1, q), h_{li} (0, 0) \right\} \right] \]  
(33)

\[ h_f = \log \delta_{i,0}^\sigma + \max_{d > 0} \left[ \max_{k \in \partial i} h_{ki} (-d, q) + h_{li} (d, q) + \sum_{m \in \partial i \setminus \{ k \}} h_{mi} (0, 0) \right] \]  
(34)

5.2.2. Mapping into a matching problem. We will develop an alternative method for the computation of the messages of BP and MS update equations, that can lead to an exponential speedup in some cases. Let us introduce an auxiliary vector \( s \in \{0, 1, \ldots, D\}^M \) associated with each vertex of the graph. Components \( s_\mu \) take value in the set of the possible positive depths \( \{1, \ldots, D\} \) if this node is member of communication \( \mu \) or 0 otherwise. For a node \( i \) that is not a root but a member of the communication \( \mu \), there exists exactly one neighbor \( k \) such that \( d_{ik} > 0, d_{ki} = -s_{\mu_k}, \mu_{ki} = \mu \) and for the remaining ones, \( d_{li}\delta_{li,k} = s_{\mu_k} + 1 \) or \( d_{li}\delta_{li,k} = 0, l \in \partial i \setminus k \). The compatibility function for E-DStP can be expressed as a function of the new variables as

\[ \psi_i^E (d_i, \mu_i) = \prod_{\mu=1}^M \left[ \sum_{s_\mu > 0} \sum_{k \in \partial i} \delta_{d_{ki},-s_\mu} \prod_{l \in \partial i \setminus k} \left( \delta_{d_{li},s_\mu+1} + \delta_{d_{li},0} \right) \right] + \prod_{k \in \partial i} \delta_{d_{ki},0} \]  
(35)

\[ = \sum_{s} \left\{ \prod_{\mu=1}^M (1 - \delta_{s_\mu,0}) \sum_{k \in \partial i} \delta_{d_{ki},-s_\mu} \prod_{l \in \partial i \setminus k} \left( \delta_{d_{li},s_\mu+1} + \delta_{d_{li},0} \right) \right\} + \prod_{\mu=1}^M \delta_{s_\mu,0} \prod_{k \in \partial i} \delta_{d_{ki},0} \]  
(36)

As introduced in the neighbors occupation formalism, we will compute the update equations from \( Z_i \), that, within this formalism can be written as

\[ Z_i = \sum_s R_s Z_s \]  
(37)

https://doi.org/10.1088/1742-5468/aaeb3f
The cavity approach for Steiner trees packing problems

where

\[ R_s = \prod_{k \in \partial i} \left[ \sum_{\nu} n_{ki} (s_{\nu} + 1, \nu) + n_{ki} (0, 0) \right] \] (38)

\[ Z_s = \sum_{t} \prod_{\mu} e^{-\beta c_{i} \sum_{s_{\mu} = 0}^{1}} \left[ \sum_{k \in \partial i} t_{k\mu} = 1 - \delta_{s_{\mu}, 0} \right] \prod_{\mu} \left[ \sum_{k \in \partial i} t_{k\mu} \leq 1 \right] \prod_{k \in \partial i} \left[ \sum_{\nu} n_{ki} (s_{\nu} + 1, \nu) + n_{ki} (0, 0) \right] t_{k\mu}. \] (39)

The components \( t_{k\mu} \) of vectors \( t \) take value 1 if \( k \in \partial i \) participate to sub-graph \( \mu \) at distance \( s_{\mu} > 0 \) from the root or 0 otherwise. The derivation of (37) is reported in appendix C. The term \( Z_s \) is the partition function of a matching problem on the complete bipartite graph \( G = (V = A \cup B, E = A \times B) \) with \( A = \partial i \) and \( B = \{ \mu : s_{\mu} > 0 \} \), and the energy of a matching is

\[ \epsilon (t) = \sum_{k \mu} t_{k\mu} \log \frac{n_{ki} (-s_{\mu}, \mu)}{\sum_{\nu} n_{ki} (s_{\nu} + 1, \mu) + n_{ki} (0, 0)} - \beta c_{i} \sum_{s_{\mu} = 0}. \] (40)

Notice that the partition function \( Z_s \) is computationally intractable as it corresponds to the calculation of a matrix permanent. In the \( \beta \to \infty \) limit we can introduce the max-sum messages \( h_{ki} (-s_{\mu}, \mu) = \frac{1}{\beta} \log n_{ki} (-s_{\mu}, \mu) \) and directly compute the free entropy \( F_i = \frac{1}{\beta} \log Z_i \) that fortunately reduces to the evaluation of

\[ F_i = \max_{s} \left[ \frac{1}{\beta} \log R_s + \log Z_s \right] \] (40)

\[ = \max \left\{ \sum_{k \in \partial i} \max_{\mu} \left[ \max_{\nu} h_{ki} (s_{\nu} + 1, \mu), h_{ki} (0, 0) \right] + F_s \right\}. \] (41)

The second term \( F_s = \frac{1}{\beta} \log Z_s \) is the free entropy of the solution of a weighted maximum matching problem on a bipartite graph which can be performed in polynomial time (precisely, in \( O((M + |\partial i|)^2 M |\partial i|) \)). Indeed, for each assignment of the \( s \) we can define the weights \( w_{k\mu} \) associated with each edge \((k, \mu)\) as

\[ w_{k\mu} = \begin{cases} h_{ki} (-s_{\mu}, \mu) - \max_{\nu} \max_{\mu} \left\{ h_{ki} (s_{\nu} + 1, \mu), h_{ki} (0, 0) \right\} & \text{if } s_{\mu} > 0 \\ -c_{i} & \text{if } s_{\mu} = 0 \end{cases} \] (42)

and solve

\[ \begin{align*}
F_s &= \max \sum_{(k, \mu)} w_{k\mu} t_{k\mu} : \\
\sum_{k \in \partial i} t_{k\mu} &\leq 1 & \forall \mu \\
\sum_{\mu} t_{k\mu} &\leq 1 & \forall k \in \partial i.
\end{align*} \] (43)

The system in (43) describes an integer linear problem for the resolution of a bipartite maximum weighted matching problem. Its relaxation to real variables \( t \) can be efficiently solved and moreover the optimal solution is proven to be integer, that is for binary \( t \).

https://doi.org/10.1088/1742-5468/aaeb3f

16
5.3. The parameter $D$

The branching formalism introduced in section 3 relies on a parameter $D$ that denotes the maximum allowed distance between the root and the leaves of any tree. This parameter limits the depth of solution-trees and therefore the goodness of the results: a small value for $D$ may prevent the connection of some terminals but a large value of $D$ will slow down the algorithm affecting the converge. Thus the value of $D$ needs to be carefully designed to ensure good performances. Although there is not a clear technique able to predict the best setting, some heuristics have been proposed in recent works to determine a minimum feasible value of $D$ for the MStP and PCStP [7]. In this work, we adopt methods described in [8] to find a minimum value of $D_\mu$ for each communication $\mu$ and we than set $D = \max_\mu D_\mu$.

It is clear that the computing cost of both V-DStP and E-DStP strongly depends on the value of $D$, more precisely linearly for the V-DStP and the binary occupation formalism and polynomially for the matching problem formulation for the E-DStP, and it could be still prohibit for graph with large diameter. Fortunately, the use of the flat formalism allows us to reduce the parameter $D$ to $D = \max_\mu |T_\mu|$ being $|T_\mu|$ the number of terminals of communication $\mu$. A proof of this property is reported in [8] for the single tree problem.

6. Max-sum for loopy graphs

The goodness of the approximation of the marginals is strictly related to the properties of the factor graph over which we run the Belief Propagation algorithm. BP is exact on tree graphs but nevertheless benefits from nice convergence properties even on general, loopy, graphs that are locally tree-like [18]. In the framework of the PCStP and multiple trees variants, there are several instances of practical interest, such as square or cubic lattices (2D or 3D graphs) modelling VLSI circuits, where many very short loops exist and the assumption of negligible correlation among variables is not satisfied. In many of these cases MS fails to converge in most of the trials or it requires a prohibitive run-time [8].

We employ here a reinforcement scheme [5, 7] that is able to make the algorithm converge on a tunable amount of time with the drawback that the solution may be sub-optimal in terms of cost. From the viewpoint of the factor graph it adds an extra factor to edge-variables that acts as an external field oriented in the direction of the cavity fields of past iterations. It slightly modifies the original problem into an easier one where a feasible assignment of variables is more likely to occur. The strength of this perturbation increases linearly in time in a way that, after few iterations, first inaccurate predictions will be neglected but, after many iterations of MS, it let the algorithm converge to, hopefully, an assignments of variables satisfying all the constraints. We report in section 6.1 how to modify the max-sum equations for the V-DStP and E-DStP for including the reinforcement factor.

The reinforcement or bootstrapping procedure described in the following sub-section is generally sufficient to guarantee convergence on random networks. In practice, however, MS did not converge in some benchmark instances, even adopting the bootstrapping
procedure. In [8] we have shown how to complement the MS equations with heuristics to solve PCStP instances in an efficient and competitive way. At each iteration we perform a re-weight of node prizes and edge weights according to temporarily max-sum predictions and we then apply a heuristics to find a tree connecting all nodes of the modified graph. After a pruning procedure, we obtain a pruned minimum spanning tree which is surely a feasible candidate solution for the PCStP. The motivation is based on the fact that although max-sum often outputs inconsistent configurations while trying to reach the optimal assignment of variables, it still contains some valuable information. Heuristics have the responsibility of adjusting the assignments of the temporarily decisional variables guaranteeing a tree-structured solution for any iteration of the main algorithm. Furthermore heuristics results do not depend on the parameter $D$ of the model and they can provide solution-trees of any diameter. We show in section 6.2 how to generalize the combination of max-sum and heuristics in the case of multiple trees for the V-DStP and the E-DStP.

6.1. Reinforcement

Within the bootstrap procedure and for each iteration $t$ of the algorithm, we compute the messages $h_{ij}^t$ and the cavity fields $H_{ij}^t$ as functions of the original messages $\tilde{h}_{ij}^t$ as

$$ h_{ij}^t (d_{ji}, \mu_{ji}) \leftarrow \tilde{h}_{ji}^t (d_{ij}, \mu_{ji}) + \gamma_t H_{ji}^{t-1} (d_{ji}, \mu_{ji}) $$

(44)

$$ H_{ij}^t (d_{ij}, \mu_{ij}) \propto \tilde{h}_{ij}^t (d_{ij}, \mu_{ij}) + \tilde{h}_{ji}^t (-d_{ij}, \mu_{ij}) + \gamma_t H_{ij}^{t-1} (d_{ij}, \mu_{ij}). $$

(45)

The parameter $\gamma_t = t\gamma_0$ is linearly proportional to $\gamma_0$ which is the reinforcement factor that governs the strength of the bootstrap. It is usually very small, of the order of $10^{-5}$ not to deviate the dynamics towards the minimum of the energy and thus affect the goodness of the solution.

6.2. Max-sum based heuristics

At each iteration of the main algorithm we perform a re-weighting of the graph to favor MS temporarily predictions and we then apply two fast heuristics to find as many spanning trees as the number of communications that we want to pack. These trees will be carefully pruned in order to decrease the cost of the solution. In this work we design two different re-weighting schemes for two different heuristics and we refer the interested reader to the single-tree heuristics explained in [8] for additional details. For each sub-graph $\mu$ we apply one of the two schemes as follows.

6.2.1. Shortest path tree. For any MS iteration $t$ we compute the auxiliary weights of a sub-graph $\mu$ as

$$ w_{ij}^t = \max_{d \neq 0} |H_{ij}^t (d, \mu)|. $$

(46)

Notice that since the field $H_{ij}^t$ is normalized, there exists only one assignment of the variables such that $H_{ij}^t (d^*, \mu) = 0$ in correspondence of the most probable state $(d^*, \mu^*)$; the field computed in all remaining states will be as negative as the probability of the
corresponding edge to be not employed in the (temporary) MS solution. For this reason we allow edges in (46) to have zero weights if they are likely to be exploited within communication \( \mu \) (they would have \( d^* \neq 0 \)); differently, we penalize edges that, according to MS, must not be used (for which \( d^* = 0 \)) imposing strictly positive weights equal to the (minus) MS field computed at the most probable non-zero depth, which corresponds, in this case, to the second maximum of \( \max_d |H_{ij}^\mu (d, \mu)| \).

We then compute the shortest paths tree (SPT) of the modified graph and we prune the solution tree removing a leaf \( i \) if it is not a terminal (for the MStP), and edges \((i, j)\) satisfying \( w_{ij} > c^\mu_i \) (for the PCStP); we repeat this procedure until we do not find such leaves.

6.2.2. Minimum spanning tree. In this scheme we assign auxiliary costs to nodes of the graph according to MS prediction. Let us consider the two auxiliary functions

\[
\begin{align*}
    h_i (d, \mu) &= \max_{k \in \partial i} \left\{ h_{ik} (-d, \mu) + \sum_{l \in \partial i \setminus k} \max \left[ h_{li}^l (d+1, \mu), h_{li}^l (0, \mu) \right] \right\} \quad \text{for } d > 0 \\
    h_i (0, \mu) &= \sum_{k \in \partial i} h_{ki}^l (0, \mu) - c^\mu_i.
\end{align*}
\]

(47)

A node satisfying \( \max_{d>0} h_i (d, \mu) < h_i (0, \mu) \) will be penalized assigning to edges incident on it a large cost \( C \). We then apply the minimum spanning tree (MST) algorithm to the modified graph and we prune the solution as in the case of the SPT.

Heuristics are applied to the graph for all the communications providing, for both E-DStP and V-DStP, a superposition of single-tree solutions. Notice that heuristics are sequentially applied, i.e. we consider one communication at the time, and depending on we are dealing with V-DStP or E-DStP, edges (and Steiner nodes for the V-DStP) selected in the first spanning trees cannot be further used by the successive applications. To overcome this problem, we add an erasing step before the application of each heuristics in which we delete edges (and eventually Steiner nodes) used by other communications. For V-DStP we only need to cut edges incident on terminals of other sub-graphs to satisfy nodes-disjoint constraints. Unfortunately such strong edge cutting procedure may lead to a graph with disconnected components or a graph in which the terminals that we aim at connecting may be isolated. In these scenarios we cannot find further trees able to span the modified graph and thus this heuristic approach fails. One way of preventing this problem is to randomize the order of the trees over which we apply the heuristics.

7. Numerical results

In this section we report the results for several experiments on synthetic networks and on benchmark, real-world, instances for the VLSI. In all the cases we will solve the V-DStP or the E-DStP where terminals have infinite prizes, i.e. the MStP variant, and a predefined root is selected for each sub-graph. The synthetic networks we chose are fully connected, regular or grid graphs, whose properties will allow us to underline the main features of the models and formalisms introduced in this work. In particular, by means of the fully connected graphs we will illustrate the improvements
carried by max-sum against a ‘greedy’ search of the solutions introduced in section 7.1. Furthermore, regular graphs allow us to verify the different scaling of the running time with respect to the degree of the graph of the two algorithms presented for the E-DStP. Motivated by their importance on technological applications, namely in the design of VLSI, we also show some results on grid, both synthetic and real-word, graphs: here we will underline the improvements carried by the flat model. Generally, energies are averaged over several instances, meaning different realizations of the weighting of the edges and assignment of the terminals, of the same graph. To measure the energy gap of the solutions found by the two different procedures, for instance ‘x’ and ‘y’ algorithms, we measure the quantity \( \frac{E_x - E_y}{E_y} \) assuming that \( E_x \) and \( E_y \) are the energies of the solutions found by algorithm ‘x’ and ‘y’ respectively. If the gap is positive (negative) the ‘x’ (‘y’) algorithm outperforms the other one.

We underline that, due to the intrinsic difficulty of the problem, there are very few (exact or approximate) results in literature and few algorithms to use for the comparison. In the case of VLSI circuits, we report the solution costs of a state-of-the art linear programming technique for the V-DStP published in [12]. This algorithm is not publicly available and thus it cannot be used for further comparisons.

### 7.1. Greedy algorithm

The ‘greedy’ procedure consists in solving, for each communication of the graph, the corresponding single-tree MStP by means of the MS algorithm combined with the bootstrap. To ensure that the superposition of these trees is a feasible candidate solution for the packing problem, we performed, as in the case of the heuristics described in section 6.2, a pre-processing of the graph before any application of the single-tree algorithm. In particular, whenever we try to propose a solution for the sub-graph \( \mu \), we cut any terminal node, and all edges incident on it, of the communications that have not yet been considered, together with edges (and Steiner tree nodes for the V-DStP variant) of the communications that we have already connected. The ‘greedy’ energy is given by the sum of energies of single-tree solutions. Notice that this ‘greedy’ procedure is actually as hard as the packing problem, since even the MStP belongs to NP-hard class of problem; nevertheless this procedure will be useful to underline the benefits carried by the parallel (packing) search against the ‘greedy’ and sequential one.

### 7.2. Fully connected graphs

Here we report results for the V-DStP on fully connected graphs where we aim at packing \( M = 3 \) trees. We compare our performances against the ‘greedy’ procedure.

We deal with fully connected graphs because here the existence of a trivial solution of the packing problem, consisting in a chain of terminal nodes, is always guaranteed. We perform two different experiments: we first fix the size of the graphs (500 nodes) and we study how energies and gaps change for an increasing number of terminals nodes. Secondly, we fix the fraction of terminals per communication, more precisely for \( \alpha = \frac{T_N}{N} = 0.08 \), \( \mu \in \{1, 2, 3\} \) and we compare the performances as we increase the size of the graphs (from 100 to 700 nodes). We run both algorithms with fixed parameters \( D = \{3, 5, 10\} \) and fixed reinforcement factor \( \gamma_0 = 10^{-5} \).
7.2.1. Uncorrelated edge weights. These experiments are performed on fully connected graphs where weights associated with edges are independently and uniformly distributed random variables in the interval $(0, 1)$. In this scenario, energies obtained by the greedy procedure are always larger than the ones achieved by the parallel search, for all values of the number of terminals and for any value of the parameter $D$ used, as it is suggested by the plot of the gaps (right plot) in figure 3. Notice that the gaps are slightly greater than zero suggesting that solutions found by the two methods are very similar in terms of energy cost, as reported in the plot in figure 3, left panel.

7.2.2. Correlated edge weights. To underline the benefits carried by the optimized strategy, we run reinforced and greedy reinforced max-sum on complete graphs with correlated edge weights. With each node $i$ we assign a uniformly distributed random variable $x_i$ in the interval $(0, 1)$ and for each edge $(i, j)$ we pick a variable $y_{ij} \in (0, 1)$. Then an edge $(i, j)$ will be characterized by a weight $w_{ij} = x_i x_j y_{ij}$. Here we expect that the cheapest edges will be chosen by the ‘greedy’ algorithm for the solution of the first trees and, as we proceed with the sequential search, the algorithm will become more and more forced to use the remaining expensive edges. In fact, as shown in figure 4, the gaps notably increase of one order of magnitude for most of the number of terminals considered in these experiments.

Notice that energies encountered for $D = \{5, 10\}$ are very close to one another suggesting that a further increasing of the parameter $D$, and thus of the solution space, will not lead to a significant improvement of the solutions.

7.2.3. Fixed fraction of terminals. To study the performances in the asymptotic limit, namely for $N \to +\infty$, $T_\mu \to +\infty$ for each communication $\mu$ and constant $\alpha$, we attempted the solution of V-DStP on complete graphs having a fixed fraction of terminals $\alpha = 0.08$ and for an increasing number of nodes $N$. Although non-rigorous, this procedure can suggest us the behavior of the energies and the energy gaps in the large $N$ limit. As reported in figure 5 panel (a), when the number of nodes reaches $N \in [500, 700]$, the energy of both max-sum and greedy solutions, for all values of $D$, seems to stabilize to a constant value. As a consequence, as plotted in figure 5 panel (b), also energy gaps fluctuates around a fixed value that seems to be different if one considers $D = 3$ or $D = \{5, 10\}$.

7.3. Regular graphs

In section 5.2 we have seen how to deal with the update equations of BP and MS algorithms for the E-DStP with the help of two different auxiliary set of variables. Although the final expressions of the equations are very different, the energies obtained by both algorithms must be identical; the only differences rely on the computational cost that strongly depends on the properties of the graph, precisely on the degree of the nodes of the graph and on the number of communications. To underline these two features of the neighbors occupation formalism and matching problem mapping, that from now will be denoted as NeighOcc and Matching algorithms, we perform two different experiments on regular, fixed degree, graphs for different values of the degree and of the number of sub-graphs. For these simulations we have fixed the values of the parameter $D = 10$ and the reinforcement factor $\gamma_0 = 10^{-4}$.
7.3.1. Energy as a function of the degree. Similarly to the experiments in section 7.3.2, here we consider regular graphs of $N = 50$ nodes containing $M = 3$ sub-graphs for four possible degrees $d \in \{3, 4, 5, 6\}$. The energies provided by NeighOcc and Matching and plotted in figure 6, panel (a), can be statistically considered the same, as for the fixed degree experiment shown before. Here the computational costs (panel (b) and (c) of
7.3.2. Energy as a function of the number of communications. In this experiment we try to solve the E-DStP on two sets of regular graphs of \( N = 50 \) nodes having fixed degree 4, for an increasing number of trees. Each communication has the same number of terminals \( T = 3 \). As shown in figure 6, panel (d), the energy costs of the solutions provided by NeighOcc and Matching algorithms are almost identical as we expected. At the same time, the computing time plotted in figure 6, panels (e) and (f), shows that the Matching procedure needs a time that scales exponentially, i.e. linearly in an log-scale plot, on the number of sub-graphs while it becomes polynomial for the NeighOcc algorithm.

7.4. Grid graphs

This section is devoted to the illustration of results of both V-DStP and E-DStP on 2D and 3D lattices. The first experiments are performed on synthetic 3D lattices of dimension \( 5 \times 5 \times 5 \) containing \( N = 125 \) nodes. Here we fix the number of communications \( M \in \{2, 3, 4\} \) and we study how energies behave when the number of terminals \( T \) per communication changes in the range \([3, 6]\). For the V-DStP and the E-DStP (only in the NeighOcc formalism) we compare the results provided by the branching and flat models. While the parameter \( D \) can be arbitrary large for the branching model, we keep the value \( D = T \) for the flat one since, as discussed in section 3.2, it is sufficient to explore all the solution space. In the second part of this section we comment the performances of the MS algorithm and of the MS-based heuristics presented in section 6.2 applied to several benchmark instances for the design problem of VLSI circuits.
7.4.1. Branching and flat models for the V-DStP e E-DStP (neighbors occupation formalism). As shown in figure 7, left panel, the energies of the solutions found by the flat model for the V-DStP are always smaller than the energies found by the branching one. We underline that, as plotted in the right panel of figure 7, the flat version of MS equations has the advantage of converging in a running time that is always smaller than the one needed by the branching model. This is reasonable as the parameter $D$, which linearly influences the computation time of both algorithms, is often greater (on average $D = 8$) for the branching model than the one fixed for the flat representation.

A different behavior is observed for the resolution of the E-DStP on grids using our two models. As remarked in figure 8, left panel, energies found by the flat and branching representations are comparable; here the depth used by the branching model, on average equal to $D = 8$ and $D = 9$ for $T \in [3, 4]$ and $T \in [5, 6]$ respectively, probably suffices to explore the same solution space considered by the flat formalism for smaller $D$. Still, the flat model is preferable as it requires a computing time that is smaller than the one needed by the branching model for all the cases we have considered.
7.4.2. V-DStP for VLSI circuits. In this section we report several results for standard benchmark instances of circuit layout where we solve the V-DStP. Instances are 3D grid graphs modelling VLSI chips where we pack relatively many trees, usually 19 or 24, each of which typically contains few terminal nodes (3 or 4). Such grid graphs can be seen as multi-layers graphs where we allow two different kinds of connections. In the multi-crossed layers, each node is connected to all its possible neighbors in all directions: the resulting graphs are cubic lattices. The multi-aligned layers are similar to the multi-crossed ones but in each layer we allow only connections in one direction.

Figure 7. Energy (a) and computational time (b) as a function of the number of terminals per communications for 3D grid graphs, V-DStP variant.

Figure 8. Energy (a) and computational time (b) as a function of the number of terminals per communications for 3D grid graphs, E-DStP variant.
Table 1. Results for circuit layout instances.

| Type              | Size   | $M$ | $T_{tot}$ | Heur. ‘J’ | Heur. ‘N’ | Rein. Max Sum | LP (opt) | Gap (MS, LP) (%) | $\langle M \rangle$ Greedy |
|-------------------|--------|-----|-----------|-----------|-----------|----------------|----------|-----------------|------------------------------|
| Augmenteddense-2  | Multi-aligned | 16 x 18 x 2  | 19 | 59 | 504 (b)  | 507 | 508 (f)  | 504 | 0 | 14 (b) 2 (f) |
| Augmenteddense-2  | Multi-crossed | 16 x 18 x 2  | 19 | 59 | 503 | | 504 (b) | 498 | 1.0 | 7 (b) 5 (f) |
| Dense-3           | Multi-crossed | 15 x 17 x 3  | 19 | 59 | 487 | 488 | 485 | 464 | 4.0 | 11 (b) 7 (f) |
| Difficult-2       | Multi-aligned | 23 x 15 x 2  | 24 | 66 | 535 | 538 | 538 | 526 | 1.7 | 15 (b) 6 (f) |
| Difficult-2x      | Multi-aligned | 23 x 15 x 2  | 24 | 66 | 560 | | | Unknown | 14 (b) 4 (f) |
| Difficult-2y      | Multi-aligned | 23 x 15 x 2  | 24 | 66 | 4776 | 4829 | 4816 | Unknown | 14 (b) 8 (f) |
| Difficult-2z      | Multi-aligned | 23 x 15 x 2  | 24 | 66 | 1060 | 1063 | 1061 | Unknown | 16 (b) 7 (f) |
| Modifieddense-3   | Multi-crossed | 16 x 17 x 3  | 19 | 59 | 492   | 496 | 495 | 479 | 2.6 | 10 (b) 3 (f) |
| Moredifficult-2   | Multi-aligned | 22 x 15 x 2  | 24 | 65 | 542 | 542 | 546 | 522 | 3.8 | 13 (b) 5 (f) |
| Pedabox-2         | Multi-aligned | 15 x 16 x 2  | 22 | 56 | 405 | 405 | 405 | 390 | 3.8 | 10 (b) 5 (f) |
| Terminalintensive-2| Multi-aligned | 23 x 16 x 2  | 24 | 77 | 596 (f) | 617 | 620 | 596 | 0 | 13 (b) 5 (f) |
either east-to-west or north-to-south [12]. For sake of simplicity, consider a cubic lattice in a three dimensional Cartesian coordinate system: depending on the value of the \( z \)-coordinate, the allowed connections will be present in directions parallel to the \( x \) or to the \( y \) axes. In Table 1 we first report some information (type of the layers, size, number of sub-graphs and total number of terminals) concerning each instance and our results. We show the energies achieved by reinforced max-sum along with the ones of the two heuristics described in Section 6.2; in analogy with [8], we label as ‘J’ heuristics that performs a modified SPT and as ‘N’ if instead we use the MST. Energies obtained using the \( \text{flat} \) model are labeled as ‘(f)’ while if nothing is specified or ‘(b)’ is used, we make use of the \( \text{branching} \) representation. Results are compared with respect to the ones obtained through state-of-the-art linear programming (LP) techniques [12] which is able, for these particular instances, to find the optimal solutions. The sign ‘-’ denotes that no solution has been found. As shown in Table 1, the gaps between the best energies achieved by MS (in bold letters) and LP are always smaller than 4% and in two cases, for the multi-aligned \( \text{augmenteddense-2} \) and \( \text{terminalintensive-2} \) instances, we reach the same performances of LP, obtaining the optimal solutions. We stress that these graphs are very loopy and far from being locally tree-like but nevertheless

Figure 9. Examples of solutions for the V-DStP on VLSI circuits for multi-aligned, (a) and (b) figures, and multi-crossed, (c) and (d) figures, layouts.

https://doi.org/10.1088/1742-5468/aaeb3f
we achieve good performances thanks to the reinforcement procedure along with the introduction of the modified heuristics. Four examples of VLSI solutions are plotted in figure 9.

It is worth noting that the greedy procedure (repeated for several permutations of the order in which the trees were considered) fails after few sequential searches. The average number of packed trees $\langle M \rangle$ before the stop of the algorithm is reported in the last column of table 1. After these greedy steps there exists one communication for which the connection of all terminals is impossible: either the remaining graph, after the pruning described in section 7.1, is composed of disconnected components (one or more terminals are disconnected from the rest of the network) or a possible connection may violate the hard topological constraints.

8. Summary of results

Using max-sum algorithm, we have explored through simulations some interesting theoretical questions in random graphs which we summarize here. Simulations (up to $N = 700$, or around $2 \times 10^5$ edges) suggest that for the Steiner tree packing problem on complete graphs with uniform independent weights, the energy converges to a constant value if the fraction of terminal vertices is kept constant, in agreement with known results for single Steiner trees [2].

We have observed a non-negligible gap (up to 7% in the solution energy and increasing with tree depth) between a greedy solution (which is commonly used by practitioners and consists in sequentially optimizing each communication and removing its used components from the graph) and the joint optimum computed by MS. Interestingly this gap is greatly expanded (up to 80% in experiments) with weights that are positively correlated. For the edge-disjoint problem, we have compared all model variants on random regular graphs with various parameters (degree, number of terminals, number of trees), confirming the convenience of each of them in a different parameter region. Simulations on regular lattice graphs give qualitatively similar results.

Finally, we have attempted to optimize a set of publicly available benchmark problems (including 3D tree packing problems), some of which have known optimum. Results are encouraging, as the solutions provided by the max-sum algorithm show a gap no larger than 4% in all cases (0% in some cases) when the optimum is known. We expect this gap to be generally independent of the problem size, which suggests that this strategy could be extremely useful for large-scale industrial problems.

Acknowledgments

We acknowledge Fondazione CRT for project SIBYL under the initiative ‘La ricerca dei Talenti’. AB acknowledges funding by INFERNET, European Union’s Horizon 2020
research and innovation program under the Marie Sklodowska-Curie grant agreement No 734439, and PRIN project 2015592CTH_003 from the Italian ministry of university and research. We warmly thank F Altarelli, R Zecchina for interesting discussions and T Koch for providing us with the VLSI instances.

Appendix A. Message-passing equations for V-DStP

Consider the compatibility function for node $i$

$$
\psi^Y_i(d_i, \mu_i) = \prod_{k \in \partial i} \delta_{\mu_k,0} \delta_{d_{ki},0} + \sum_{\mu=1}^{M} \sum_{d>0} \sum_{k \in \partial i} \delta_{\mu_k,\mu} \delta_{d_{ki},-d} \prod_{l \in \partial i \setminus k} \left( \delta_{\mu_l,\mu} \delta_{d_{li},d+1} + \delta_{\mu_l,0} \delta_{d_{li},0} \right)
$$

$$
+ \sum_{\mu} \delta_{\mu,0} \sum_{d>0} \sum_{k \in \partial i} \delta_{\mu,\mu} \delta_{d_{ki},-d} \sum_{l \in \partial i \setminus k} \delta_{\mu_l,\mu} \delta_{d_{li},d} \prod_{m \in \partial i \setminus \{k,l\}} \delta_{\mu_m,0} \delta_{d_{mi},0}.
$$

(A.1)

For sake of simplicity we split $\psi^Y_i(d_i, \mu_i)$ in

$$
\psi^Y_i(d_i, \mu_i) = \psi_i^{(1)}(d_i, \mu_i) + \psi_i^{(2)}(d_i, \mu_i) + \psi_i^{(3)}(d_i, \mu_i)
$$

(A.2)

where

$$
\psi_i^{(1)}(d_i, \mu_i) = \prod_{k \in \partial i} \delta_{\mu_k,0} \delta_{d_{ki},0}
$$

(A.3)

$$
\psi_i^{(2)}(d_i, \mu_i) = \sum_{\mu=1}^{M} \sum_{d>0} \sum_{k \in \partial i} \delta_{\mu_k,\mu} \delta_{d_{ki},-d} \prod_{l \in \partial i \setminus k} \left( \delta_{\mu_l,\mu} \delta_{d_{li},d+1} + \delta_{\mu_l,0} \delta_{d_{li},0} \right)
$$

(A.4)

$$
\psi_i^{(3)}(d_i, \mu_i) = \sum_{\mu} \delta_{\mu,0} \sum_{d>0} \sum_{k \in \partial i} \delta_{\mu,\mu} \delta_{d_{ki},-d} \sum_{l \in \partial i \setminus k} \delta_{\mu_l,\mu} \delta_{d_{li},d} \prod_{m \in \partial i \setminus \{k,l\}} \delta_{\mu_m,0} \delta_{d_{mi},0}.
$$

(A.5)

Using $\psi_i^{(1)}(d_i, \mu_i), \psi_i^{(2)}(d_i, \mu_i)$ and $\psi_i^{(3)}(d_i, \mu_i)$ inside (14) we can compute the message $m_{ij}(d_{ij}, \mu_{ij})$ as a sum of three contributions, namely

$$
m_{ij}(d_{ij}, \mu_{ij}) = m_{ij}^{(1)}(d_{ij}, \mu_{ij}) + m_{ij}^{(2)}(d_{ij}, \mu_{ij}) + m_{ij}^{(3)}(d_{ij}, \mu_{ij})
$$

(A.6)

$$
m_{ij}^{(1)}(d_{ij}, \mu_{ij}) = \sum_{\{k_i, \mu_{ki}\}} e^{-\beta \sum_{\mu} c_{\mu} \Pi_{k_i \in \partial i} (1-\delta_{\mu_k,\mu}) \prod_{l \in \partial i} \delta_{\mu_l,0} \delta_{d_{li},0} \prod_{k \in \partial i \setminus j} n_{ki}(d_{ki}, \mu_{ki})}
$$

$$
= e^{-\beta \sum_{\mu} c_{\mu} \delta_{\mu_{ij},0} \delta_{d_{ij},0} \prod_{k \in \partial i \setminus j} m_{ki}(0,0)}
$$

(A.6)

$$
m_{ij}^{(2)}(d_{ij}, \mu_{ij}) = \sum_{\mu} e^{-\beta \sum_{\mu} c_{\mu} \Pi_{k \in \partial i} (1-\delta_{\mu_k,\mu}) \sum_{d>0} \left\{ \delta_{d_{ji},-d} \delta_{\mu_{ij},\mu} \prod_{k \in \partial i \setminus j} n_{ki}(d+1, \mu) + n_{ki}(0,0) \right\} + \left( \delta_{d_{ji},d+1} \delta_{\mu_{ij},\mu} + \delta_{d_{ji},d} \delta_{\mu_{ij},0} \right) \sum_{k \in \partial i \setminus j} n_{ki}(-d, \mu) \prod_{l \in \partial i \setminus \{j,k\}} n_{li}(d+1, \mu) + n_{li}(0,0) \right\}}
$$

(A.7)
\[ m_{ij}^{(3)}(d_{ij}, \mu_{ij}) = \sum_{\mu} \delta_{\mu,0} \sum_{d > 0} \left[ \delta_{\mu,d_{ij},-d} \sum_{k \in \partial \setminus j} n_{ki}(d, \mu) \prod_{l \in \partial \setminus \{j, k\}} n_{li}(0, 0) \right. \\
+ \delta_{\mu,d_{ij},d} \sum_{k \in \partial \setminus j} n_{ki}(-d, \mu) \prod_{l \in \partial \setminus \{j, k\}} n_{li}(0, 0) \\
\left. + \delta_{\mu,d_{ij},0} \sum_{k \in \partial \setminus j} n_{ki}(d, \mu) \sum_{\mu_{m} \in (\partial \setminus \{k, l, j\})} n_{mi}(0, 0) \right] \]  

(A.8)

If now we use that \( d_{ji} = -d_{ij} \) and \( \mu_{ij} = \mu_{ji} \) we can write the following set of equations:

\[ m_{ij}(d, \mu) = \prod_{k \in \partial \setminus j} [n_{ki}(d + 1, \mu) + n_{ki}(0, 0)] \\
+ \delta_{\mu,0} \sum_{k \in \partial \setminus j} n_{ki}(d, \mu) \prod_{l \in \partial \setminus \{j, k\}} n_{li}(0, 0) \quad \forall d > 0, \mu \neq 0 \\
m_{ij}(d, \mu) = \sum_{k \in \partial \setminus j} n_{ki}(d + 1, \mu) \prod_{l \in \partial \setminus \{j, k\}} [n_{li}(d, \mu) + n_{li}(0, 0)] \\
+ \delta_{\mu,0} \sum_{k \in \partial \setminus j} n_{ki}(d, \mu) \prod_{l \in \partial \setminus \{j, k\}} n_{li}(0, 0) \quad \forall d < 0, \mu \neq 0.
\]

For \( d = \mu = 0 \)

\[ m_{ij}(0, 0) = e^{-\beta \sum_{e_i} e_i} \prod_{k \in \partial \setminus j} n_{ki}(0, 0) + \sum_{\mu \neq 0} \sum_{d < 0} \sum_{k \in \partial \setminus j} n_{ki}(d + 1, \mu) \prod_{l \in \partial \setminus \{j, k\}} [n_{li}(d, \mu) + n_{li}(0, 0)] \\
+ \sum_{\mu \neq 0} \sum_{d < 0} \sum_{k \in \partial \setminus j} n_{ki}(d, \mu) \sum_{\mu_{m} \in (\partial \setminus \{k, l, j\})} n_{mi}(0, 0). \]

**Appendix B. Recursive expression of \( Z^q \) for the E-DStP**

From equation (24)

\[ Z^q = \sum_{d_{ij}, \mu_{ij}, \mu_{ki} \in \partial_{n}} \psi_{i}^{F}(d_i, \mu_{ij}) e^{-\beta \sum_{e_i} e_i} \Pi_{k \in \partial \setminus n}(1-\delta_{\mu_{ki}, \mu}) \prod_{k \in \partial} \| x_k = 1 - \delta_{d_{ki}, 0} \| n_{ki}(d_{ki}, \mu_{ki}) \]  

(B.1)

we underline the possible contribution to a communication \( q \) from at least one on the neighbors of \( i \) as

\[ Z^q_{\bm{x}} = \sum_{d_{ij}, \mu_{ij} \in \Gamma} \psi_{i}^{F}(d_i, \mu_{ij}) e^{-\beta \sum_{e_i} e_i} \Pi_{k \in \partial \setminus n}(1-\delta_{\mu_{ki}, \mu}) \\
\times \prod_{k \in \partial} \| x_k = 1 \| [d_{ki} \neq 0] n_{ki}(d_{ki}, \mu_{ki}) \prod_{k \in \partial, \mu_{ki} \in \partial_{n}} \| x_k = 1 - \delta_{d_{ki}, 0} \| n_{ki}(d_{ki}, \mu_{ki}) \].

Consider a vector \( \bm{x} \) such that there exists at least one component \( x_k = 1 \) for \( d_{ki} \neq 0, \mu_{ki} = q \) and possibly other components different from zero assigned to one of the possible sub-graph \( \mu \leq q - 1 \). This vector can be seen as the superposition of all
vectors \( y \leq x \), that is, all vectors having at most the same number of non-zeros of \( x \) and the component \( y_k = 0 \) each time \( \mu_{ki} = q \); all remaining components must satisfy \( y_{k'} = 1 - \delta_{d_{k'i},0} \) for \( \mu_{ki} \leq q - 1 \). Thus:

\[
Z^q_x = \sum_{d_i, \mu_i} \sum_{y \leq x} \sum_{\mu_k \leq q} e^{-\beta c_i^q} \prod_{k \in \partial i} (1 - \delta_{\mu_{ki},q}) \psi_i^q (d_i, \mu_i) \prod_{k \in \partial i} n_{ki} (d_{ki}, \mu_{ki}) \delta_{\mu_{ki},q} \]

\[
\times \prod_{p < q - 1} \prod_{k \in \partial i} e^{-\beta c_i^p} \prod_{k \in \partial i} (1 - \delta_{\mu_{ki},q}) \psi_i^p (d_i, \mu_i) \prod_{k \in \partial i} n_{ki} (d_{ki}, \mu_{ki}) \]

where we have made use of the expression of \( \psi_i^E \) in (9). If we now collect the sum over \( y \leq x \) and we explicitly use the constraints on depth and communication variables we find

\[
Z^q_x = \sum_{y \leq x} \left\{ \sum_{d_i, \mu_i} \sum_{\mu_k \leq q} e^{-\beta c_i^q} \prod_{k \in \partial i} (1 - \delta_{\mu_{ki},q}) \psi_i^q (d_i, \mu_i) \prod_{k \in \partial i} n_{ki} (d_{ki}, \mu_{ki}) \delta_{\mu_{ki},q} \right\}
\]

\[
\times \sum_{d_i, \mu_i} \prod_{\mu_k \leq q - 1} \left[ \sum_{\mu_i} \prod_{k \in \partial i} (1 - \delta_{\mu_{ki},q}) \psi_i^q (d_i, \mu_i) \prod_{k \in \partial i} n_{ki} (d_{ki}, \mu_{ki}) \right]
\]

\[
= \sum_{y \leq x} (g^0_y + g^b_y + g^f_y) Z^q_y^{-1}
\]

where

\[
g^0_y = e^{-\beta c_i^q} \prod_{k \in \partial i} n_{ki} (0, 0)
\]

\[
g^b_y = \sum_{d > 0} \sum_{j \in \partial i} n_{ji} (-d, q) \prod_{k \in \partial i} \left[ n_{ki} (d + 1, q) + n_{ki} (0, 0) \right]
\]

\[
g^f_y = \delta c_i^0 \sum_{d > 0} \sum_{j \in \partial i} n_{ji} (-d, q) \sum_{\mu_k} n_{ki} (d, q) \prod_{l \in \partial i \setminus \{j, k\}} n_{li} (0, 0)
\]

In the special case in which no communications is flowing within the graph, that is for \( q = 0 \), we must impose the value of \( Z^0_x \) through

\[
Z^0_x = e^{-\beta \sum \epsilon_i^q} \prod \left[ x = 0 \right] n_{ji} (0, 0)
\]

https://doi.org/10.1088/1742-5468/aaeb3f
Appendix C. From E-DStP to a weighted maximum matching problem

Let us explicit the dependency on $\mu_i$ of (36)

$$
\psi_i^E (d_i, \mu_i) = \sum_{s} \left\{ \prod_{\mu: s_\mu > 0} \sum_{k \in \partial i} \delta_{\mu_k, i} \delta_{d_{ki}, -s_\mu} \prod_{l \in \partial i \setminus k} \left[ \delta_{\mu_l, i} \delta_{d_{li}, s_\mu + 1} + (1 - \delta_{\mu_l, i}) \right] \right. 

\left. + \prod_{\mu: s_\mu = 0} \prod_{k \in \partial i} (1 - \delta_{\mu_k, i}) \delta_{d_{ki}, s_{\mu_k}} \right\} 
$$

(C.1)

and let us introduce a partial partition function as in (22) and let us underline the $s$–dependence as

$$
Z_i = \sum_{d_i, \mu_i} \psi_i^E (d_i, \mu_i) e^{-\beta \sum_{\mu} c_{\mu} \prod_{k \in \partial i} (1 - \delta_{\mu_k, i})} \prod_{k \in \partial i} n_{ki} (d_{ki}, \mu_{ki}) 

= \sum_{s} Q_s
$$

(C.3)

where the function $Q_s$ reads (here we collect the topological constraints in $f_{k\mu}$ for a neighbor $k \in \partial i$ participating in sub-graph $\mu$)

$$
Q_s = \sum_{d_i} \sum_{\{\mu_k: s_{\mu_k} > 0, \mu_{\partial i} = 0\}} \prod_{k \in \partial i} n_{ki} (d_{ki}, \mu_{ki}) \left\{ \prod_{\mu: s_\mu > 0} e^{-\beta \sum_{\mu} c_{\mu} \prod_{k \in \partial i} (1 - \delta_{\mu_k, i})} \sum_{k \in \partial i} f_{k\mu} + e^{-\beta \sum_{\mu} c_{\mu} \prod_{k \in \partial i} \delta_{\mu_k, i}} \right\} 

f_{k\mu} = \delta_{\mu_k, i} \delta_{d_{ki}, -s_\mu} \prod_{l \in \partial i \setminus k} \left[ \delta_{\mu_l, i} \delta_{d_{li}, s_\mu + 1} + (1 - \delta_{\mu_l, i}) \right].
$$

(C.5)

(C.6)

Let us concentrate in the computation of $Q_s$ for a fixed $s$. For simplicity of notation, we will assume, unless explicitly noted, that $\mu$ indices run over the set $\{\mu: s_\mu > 0\}$. Now as $f_{k\mu} f_{k\nu} = 0$ if $\mu \neq \nu$ (because $\delta_{\mu_k, i} \delta_{\mu_{\partial i}, \nu} = 0$), we have that

$$
\delta_{\mu_k, i} \delta_{d_{ki}, -s_\mu} [\delta_{\mu_{\partial i}, \nu} \delta_{d_{li}, s_{\mu} + 1} + (1 - \delta_{\mu_{\partial i}, \nu})] = \delta_{\mu_k, i} \delta_{d_{ki}, -s_\mu}
$$

(C.7)

and equivalently

$$
\prod_{\nu} [\delta_{\mu_{\partial i}, \nu} \delta_{d_{ki}, s_{\mu} + 1} + (1 - \delta_{\mu_{\partial i}, \nu})] = \sum_{\nu} \delta_{\mu_{\partial i}, \nu} \delta_{d_{ki}, s_{\mu} + 1} + \delta_{\mu_{\partial i}, 0} \delta_{d_{ki}, 0}.
$$

(C.8)

Thus

$$
\prod_{\mu} \sum_{k \in \partial i} f_{k\mu} = \sum_{\pi} \prod_{\mu} f_{\pi_{\mu} \mu}
$$

where the sum $\sum_{\pi} s$ runs over all the possible coupling between communications and neighbors of node $i$. Mathematically we have defined the one-to-one functions $\pi$

$$
\pi: \{\mu: s_\mu > 0\} \to \partial i
$$
with $\pi : \mu \mapsto \pi_\mu$. In the following, we will switch to an alternative representation of functions $\pi$. If we denote by $t_{k\mu} = \delta_{k, \pi_\mu}$ for a fixed $\pi$ we obtain

$$
\prod_{\mu} f_{\pi_\mu} = \prod_{\mu} \delta_{\mu_\pi, 0} \prod_{\nu} \left[ \delta_{\mu_\pi, 0} \delta_{d_{\nu}, 1} \delta_{s_\mu, 0} + (1 - \delta_{\mu_\pi, 0}) \right]
$$

$$
= \prod_{k \in \partial i} \left( \sum_{\nu} \delta_{\mu_k, 0} \delta_{d_{\nu}, 1} \delta_{s_\mu, 0} + (1 - \delta_{\mu_k, 0}) \delta_{d_{\nu}, 0} \right)^{1-\sum_{\nu} t_{k\nu}} \prod_{\mu} \left( \delta_{\mu_k, 0} \delta_{d_{\nu}, 1} \delta_{s_\mu, 0} \right)^{t_{k\mu}}
$$

with the convention that $0^0 = 1$. Note that the vector $t$ and the function $\pi$ contain the same information: we have that $\sum_{k \in \partial i} t_{k\mu} = 1 - \delta_{s_\mu, 0}$ for each $\mu$ and $\sum_{\mu} t_{k\mu} \leqslant 1$ for each $k \in \partial i$. These two conditions are complete; for a vector $t$ that satisfies these two constraints, the corresponding function $\pi$ can be defined naturally. We will have then

$$
Z_i = \sum_s \sum_t \prod_{\mu} e^{-\beta c_{\mu}} \prod_{k \in \partial i} \left[ \sum_{\nu} (1 - \delta_{s_\mu, 0}) \right] \prod_{k \in \partial i} \left[ \sum_{\mu} t_{k\mu} \right] \leqslant 1
$$

$$
\times \prod_{k \in \partial i} \left[ \sum_{\nu} n_{k\nu} (s_{\nu} + 1, \nu) + n_{k\nu} (0, 0) \right]^{1-\sum_{\nu} t_{k\nu}} \prod_{\mu} n_{ki} (-s_\mu, \mu)^{t_{k\mu}}
$$

$$
= \sum_s R_s Z_s.
$$

References

[1] Altarelli F, Braunstein A, Dall’Asta L, De Bacco C and Franz S 2015 The edge-disjoint path problem on random graphs by message-passing PLOS ONE 10 e0145222
[2] Angel O, Flaxman A D and Wilson D B 2012 A sharp threshold for minimum bounded-depth and bounded-diameter spanning trees and Steiner trees in random networks Combinatorica 32 1–33
[3] De Bacco C, Franz S, Saad D and Yeung C H 2014 Shortest node-disjoint paths on random graphs J. Stat. Mech. P07009
[4] Bailly-Bechet M, Borgs C, Braunstein A, Chayes J, Dagkessamanskaia A, François J-M and Zecchina R 2011 Finding undetected protein associations in cell signaling by belief propagation Proc. Natl Acad. Sci. 108 882–7
[5] Bayati M, Borgs C, Braunstein A, Chayes J, Ramezanpour A and Zecchina R 2008 Statistical mechanics of Steiner trees Phys. Rev. Lett. 101 037208
[6] Bayati M, Braunstein A and Zecchina R 2008 A rigorous analysis of the cavity equations for the minimum spanning tree J. Math. Phys. 49 125206
[7] Biazzo I, Braunstein A and Zecchina R 2012 Performance of a cavity-method-based algorithm for the prize-capturing Steiner tree problem on graphs Phys. Rev. E 86 026706
[8] Braunstein A and Muntoni A 2016 Practical optimization of Steiner trees via the cavity method J. Stat. Mech. 073302
[9] Burstein M and Pelavin R 1983 Hierarchical wire routing IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 2 223–34
[10] Cohoon J P and Heck P L 1988 BEAVER: a computational-geometry-based tool for switchbox routing IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 7 684–97
[11] Grötschel M, Martin A and Weismantel R The Steiner tree packing problem in VLSI design Math. Program. 78 265–81
[12] Hoang N-D and Koch T 2012 Steiner tree packing revisited Math. Methods Oper. Res. 76 95–123
[13] Karp R M 1972 Reducibility among Combinatorial Problems (Berlin: Springer) pp 85–103

https://doi.org/10.1088/1742-5468/aaeb3f
The cavity approach for Steiner trees packing problems

[14] Ljubić I, Weiskircher R, Pferschy U, Klau G W, Mutzel P and Fischetti M 2005 An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem Math. Program. 105 427–49

[15] Luk W K 1985 A greedy switch-box router Integ. VLSI J. 3 129–49

[16] Mézard M and Montanari A 2009 Information, Physics, and Computation (Oxford: Oxford University Press)

[17] Mézard M and Parisi G 2003 The cavity method at zero temperature J. Stat. Phys. 111 1–34

[18] Weiss Y and Freeman W T 2001 On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs IEEE Trans. Inf. Theory 47 736–44

https://doi.org/10.1088/1742-5468/aaeb3f