ISSN 2279-9362

Collegio Carlo Alberto

The cavity approach for Steiner trees Packing Problems

Alfredo Braunstein
Anna Paola Muntoni

No. 530
December 2017

Carlo Alberto Notebooks

www.carloalberto.org/research/working-papers

The cavity approach for Steiner trees Packing Problems

Alfredo Braunstein*
DISAT, Politecnico di Torino, Corso Duca Degli Abruzzi 24, Torino, Italy
Human Genetics Foundation, Via Nizza 52, Torino, Italy and
Collegio Carlo Alberto, Via Real Collegio 1, Moncalieri, Italy

Anna Paola Muntoni†
DISAT, Politecnico di Torino, Corso Duca Degli Abruzzi 24, Torino, Italy

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 one of the two formalisms used for the edge-disjoint variant allow us 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%.

CONTENTS

* alfredo.braunstein@polito.it
† anna.muntoni@polito.it

I. 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 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 and its prize-collecting version. We consider two different variants regarding the interaction between 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; but 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 six 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 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, its exact computation involves impractical computations. MS consists in iterating closed massage-passing equations on a factor graph that is 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 or 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 subgraph; 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 Erdos-Renyi 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, Erdos-Renyi and random regular graphs and on benchmark instances of V-DStP arising from the VLSI design problem.

II. TWO STEINER PACKING PROBLEMS

Given a graph $G = (V, E)$ whose vertices have non-negative real prizes $\{c_i^\mu : i \in V, \mu = 1, \dots, M\}$ and whose edges have real positive weights $\left\{w_{ij}^\mu : (i,j) \in E, \mu = 1, \dots, M\right\}$, we consider the problem of finding M connected sub-graphs $G_\mu = (V_\mu, E_\mu)$ spanning disjoint sets of terminals $\{T_\mu \subseteq V_\mu, \mu = 1, \dots, M\}$ that minimize the following cost or energy function

$$H = \sum_\mu \left[\sum_{i \in V \setminus V_\mu} c_i^\mu + \sum_{(i,j) \in E_\mu} w_{ij}^\mu \right] \tag{1}$$

This definition of the cost is extremely general: node prizes and edge costs can depend on sub-graph μ. For directed graphs, we can admit $w_{ij}^\mu \neq w_{ji}^\mu$ 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 imposing $c_i^\mu = +\infty$ if node i is a (true) terminal of tree μ (it suffices to have a large enough value for c_i^μ instead of $+\infty$), and $c_j^\mu = 0$, $\forall \mu$ 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_μ as a "communication" μ flowing within the graph.

Subsets G_μ 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_μ 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 disjoints, i.e. $E_\mu \cap E_\nu = \emptyset$ if $\mu \neq \nu$, but vertex sets can overlap.

III. 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 and variables, in which an edge between a factor and a variable exists if the factor depends on the variable. More precisely, to each vertex $i \in V$ we associate a factor node with factor ψ_i and to each edge $(i,j) \in E$ we associate a two components variable $(d_{ij}, \mu_{ij}) \in \{-D, \dots, 0, \dots, D\} \times \{0, \dots, M\}$. 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 μ_{ij} by which we label edges forming different trees.

Compatibility functions ψ_i are defined in a way that allowed configurations of variables $(\boldsymbol{d}, \boldsymbol{\mu}) \doteq \{(d_{ij}, \mu_{ij}) : (i,j) \in E\}$ 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 $\boldsymbol{d}_i = \{d_{ij} : j \in \partial i\}$ and $\boldsymbol{\mu}_i = \{\mu_{ij} : j \in \partial i\}$ through compatibility functions $\psi_i(\boldsymbol{d}_i, \boldsymbol{\mu}_i)$ 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 μ_{ij} takes value from the set $\{0, 1, \dots, M\}$ 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 $d_{ij} \in \{-D, \dots, 0, \dots, D\}$ 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, ψ_i^V and ψ_i^E, for the two problems. Both functions will be written with the help of a single-tree compatibility function ψ_i^μ for two different formulations of the constraints, the *branching* and the *flat* model.

1. Branching model

Let us consider a sub-graph G_μ 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_μ passing through $j \in \partial i$. Variable d_{ij} will be strictly positive (negative) if j is one step closer (farther) then i to root r_μ. Thus, every edge will satisfy $\mu_{ij} = \mu_{ji}$ and the anti-symmetric condition $d_{ij} = -d_{ji}$. A directed tree structure is guaranteed if there exists only one neighbor $j \in \partial i$ such that $\mu_{ij} \neq 0$ and $d_{ij} > 0$ and all remaining neighbors $k \in \partial i \setminus j$ can either not enter in this solution or be member of tree G_μ at the distance $d_{ki} = d_{ij} + 1$ from root r_μ. Mathematically, we can define the following compatibility function

$$\psi_i^{\mu,\,b}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) = \prod_{j\in\partial i}\delta_{\mu_{ji},0}\delta_{d_{ji},0} + \sum_{d>0}\sum_{j\in\partial i}\left[\delta_{\mu,\mu_{ji}}\delta_{d_{ji},-d}\prod_{k\in\partial i\setminus j}\left(\delta_{\mu,\mu_{ki}}\delta_{d_{ki},d+1} + \delta_{\mu_{ki},0}\delta_{d_{ki},0}\right)\right] \tag{2}$$

where $\delta_{x,y}$ is the discrete Kronecker delta function equal to 1 if $x = y$ and 0 otherwise.

2. The flat formalism

The diameter of solutions representable using the *branching model* strongly depends on the value of the parameter D. 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_μ, i.e. the degree of node i within the sub-graph is more then two. It can be shown [8] that for $D = T$, where $T = |T_\mu|$ is the number of terminals per communication, these constraints admits all feasible trees plus some extra structures which contain disconnected cycles with no terminals and thus are energetically disfavored. The indicator function of these constraints is

$$\psi_i^{\mu,\,f}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) = \delta_{c_i^\mu,0}\sum_{d>0}\sum_{k\in\partial i}\delta_{\mu,\mu_{ki}}\delta_{-d,d_{ki}}\sum_{l\in\partial i\setminus k}\delta_{\mu,\mu_{li}}\delta_{d_{li},d}\prod_{m\in\partial i\setminus\{k,l\}}\delta_{\mu_{mi},0}\delta_{d_{mi},0} \tag{3}$$

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

$$\psi_i^\mu\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) = \psi_i^{\mu,\,b}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) + \psi_i^{\mu,\,f}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) \tag{4}$$

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

A. Constraints for the Vertex-Disjoint Steiner trees problem

In the V-DStP a node can belong to none or at most one sub-graph G_μ; thus, if a vertex i is member of a Steiner tree, it neighbor edges can either participate to the same communication or be unused. Being nodes sets of the solution non-overlapping, we can consider separately the ownership of a vertex to a particular tree. The compatibility function ψ_i^V can be then expressed as the sum over all possible trees of a single-tree compatibility function ψ_i^μ in (4).

$$\psi_i^V\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) = \sum_{\mu=1}^{M}\psi_i^\mu\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) \tag{5}$$

B. 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 $\boldsymbol{d}_i = \boldsymbol{0}$ if $\boldsymbol{\mu}_i = \boldsymbol{0}$. For the remaining cases, if some neighbors $k \in \partial i$ is a members of a Steiner tree μ, 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

$$\psi_i^{E,b}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right)=\prod_{\mu=1}^{M}\left[\prod_{k\in\partial i}\delta_{d_{ki}\delta_{\mu_{ki},\mu},0}+\sum_{d>0}\sum_{k\in\partial i}\delta_{d_{ki}\delta_{\mu_{ki},\mu},-d}\prod_{l\in\partial i\setminus k}\left(\delta_{d_{li}\delta_{\mu_{li},\mu},d+1}+\delta_{d_{li}\delta_{\mu_{li},\mu},0}\right)\right] \tag{6}$$

for the branching model and

$$\psi_i^{E,f}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right)=\prod_{\mu}\left[\delta_{c_i^{\mu},0}\sum_{d>0}\sum_{k\in\partial i}\delta_{-d,d_{ki}\delta_{\mu_{ki},\mu}}\sum_{l\in\partial i\setminus k}\delta_{\mu,\mu_{li}}\delta_{d_{li}\delta_{\mu_{li},\mu},d}\prod_{m\in\partial i\setminus\{k,l\}}\delta_{d_{mi}\delta_{\mu_{mi},\mu},0}\right] \tag{7}$$

for the flat model. Notice that we can express

$$\psi_i^{E}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right)=\psi_i^{E,b}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right)+\psi_i^{E,f}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) \tag{8}$$

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 with 5)

$$\psi_i^{E}\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right)=\prod_{\mu=1}^{M}\psi_i^{\mu}\left(\tilde{\boldsymbol{d}}_i,\boldsymbol{\mu}_i\right) \tag{9}$$

Some examples of feasible assignments of variables for both branching and flat models are shown in Figure 1 on page 7. On the top-left (Figure 1 on page 7 (a)) we see one instance of the V-DStP containing two sub-graphs, the "red" having root "4" and the "green" rooted at node "3", and on the right two "red" and "green" edge-disjoint Steiner trees, rooted at "0" and "5" respectively. Roots are represented as square nodes in contrast to circle colored nodes that are terminals. Edges employed in both 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 top-left figure, nodes of the vertex-disjoint trees are members either of the "red" or the "greed" trees but such constraint is relaxed on the top-right of Figure 1 on page 7 (b) for E-DStP. To underline it, we allow node "9" to be a terminal of communication "green" and a Steiner node of the "red" tree as two incident edges, $(0,9)$ and $(9,8)$, belong to the "red" solution.

To show an example of representation through the flat formalism, we picture in Figure 2 on page 8 (a) and (b) two representations, one for each model, of the same solution to the V-DStP on a grid graph. According to the branching formalism, we see in Figure 2 on page 8 (a) that we need a minimum depth of $D=6$ to allow all terminals of the "blue" communication to reach the root node "1" but, since the tree in actually a chain of nodes, we notice that in Figure 2 on page 8 (b) the same solution can be represented in the flat formalism using $D=3$. 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 "21" of the orange solution, Figure 2 on page 8 (b).

IV. 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 $\boldsymbol{d}=\{d_{ij}:(i,j)\in E\}$ and $\boldsymbol{\mu}=\{\mu_{ij}:(i,j)\in E\}$ of the associated factor graph. The cost function in (1) can be then expressed in terms of the new variables as

$$H\left(\boldsymbol{d},\boldsymbol{\mu}\right)=\sum_{\mu=1}^{M}\left[\sum_{i\in V}c_i^{\mu}\mathbb{I}\left[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}\right]+\sum_{\substack{d_{ij}>0:\\ \mu_{ij}=\mu}}w_{ij}\right] \tag{10}$$

where, for sake of simplicity, we consider the "homogeneous" case $w_{ij}^{\mu}=w_{ij}\ \forall\mu$ and ψ_i can be either equal to ψ_i^V or ψ_i^E. The function $\mathbb{I}\left[\cdot\right]$ is the indicator function that is equal to one if its argument is true and zero otherwise. The expression $\boldsymbol{\mu}_i\neq\boldsymbol{\mu}$ means that none of the neighbors $k\in\partial i$ satisfies $\mu_{ki}=\mu$.

The Boltzmann-Gibbs distribution associated with the energy $H\left(\boldsymbol{d},\boldsymbol{\mu}\right)$ is given by:

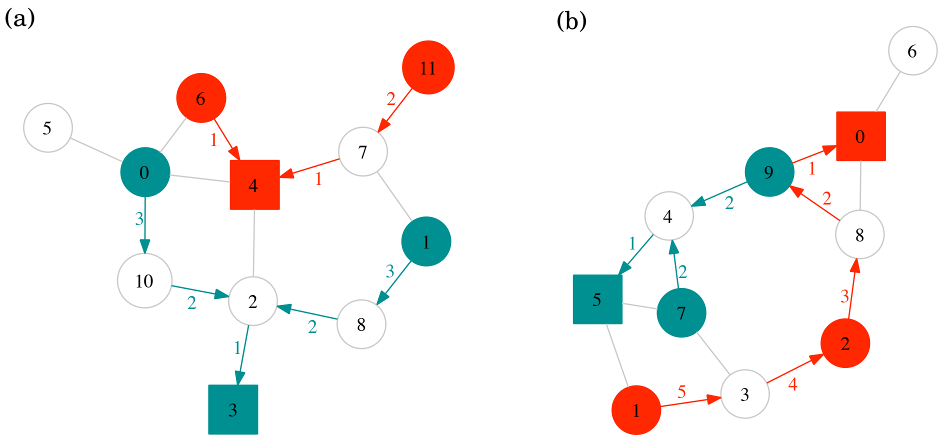


Figure 1. Figures (a) and (b) show a feasible assignment of the variables for the V-DStP (left) and the E-DStP (right) using the branching model.

$$P\left(\boldsymbol{d},\boldsymbol{\mu}\right)=\frac{\prod_{i}\psi_{i}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)e^{-\beta H\left(\boldsymbol{d},\boldsymbol{\mu}\right)}}{Z_{\beta}} \tag{11}$$

in which β is a positive parameter, called the "inverse temperature" as in the statistical mechanics framework, and the normalization constant

$$Z_{\beta}=\sum_{\boldsymbol{d},\boldsymbol{\mu}}\prod_{i}\psi_{i}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)e^{-\beta H\left(\boldsymbol{d},\boldsymbol{\mu}\right)}$$

is the partition function. Configurations of the variables that do not satisfy the topological constraints will have zero probability measure, whereas all other configuration 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(s) that minimizes $H\left(\boldsymbol{d},\boldsymbol{\mu}\right)$ (the *ground state* of the system) that are exactly the solutions of the optimization problems. Thus we are interesting in determining, for each edge $(i,j)\in E$, the assignment of variables that, in the $\beta\to+\infty$, limit maximizes the marginal probability distribution P_{ij} defined as:

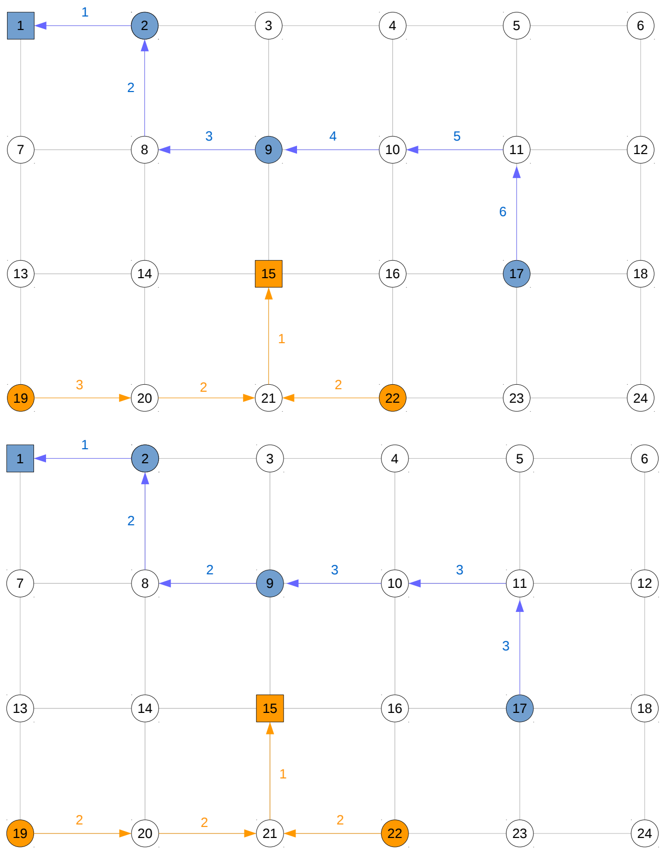
$$P_{ij}\left(\tilde{d}_{ij},\tilde{\mu}_{ij}\right)=\sum_{\boldsymbol{d},\boldsymbol{\mu}}P\left(\boldsymbol{d},\boldsymbol{\mu}\right)\delta_{d_{ij},\tilde{d}_{ij}}\delta_{\mu_{ij},\tilde{\mu}_{ij}} \tag{12}$$

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 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 β the BP equations on our factor graph are:

$$\begin{cases}m_{ij}\left(d_{ij},\mu_{ij}\right) & =\frac{1}{Z_{ij}}\sum_{\substack{\{d_{ki},\mu_{ki}\}:\\ k\in\partial i\setminus j}}\psi_{i}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)e^{-\beta\sum_{\mu}c_{i}^{\mu}\mathbb{I}\left[\boldsymbol{\mu}_{i}\neq\boldsymbol{\mu}\right]}\prod_{k\in\partial i\setminus j}n_{ki}\left(d_{ki},\mu_{ki}\right)\\ \\ n_{ki}\left(d_{ki},\mu_{ki}\right) & =e^{-\beta w_{ki}\mathbb{I}\left[d_{ki}>0\right]}m_{ki}\left(d_{ki},\mu_{ki}\right)\end{cases} \tag{13}$$

where

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



(a)

(b)

Figure 2. Figures (a) and (b) picture the same solution to a V-DStP using the branching formalism (figure (a)) and the flat representation (figure (b))

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}, μ_{ij}) for edge (i, j) if the node j were temporarily erased from the graph.

The system of equations in (13) 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 (13) 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

$$M_{ij}\left(d_{ij}, \mu_{ij}\right) \propto n_{ij}\left(d_{ij}, \mu_{ij}\right) n_{ji}\left(-d_{ij}, \mu_{ij}\right) \tag{14}$$

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 $h_{ij}\left(d_{ij}, \mu_{ij}\right) = \lim_{\beta\to\infty} \frac{1}{\beta} \log n\left(d_{ij}, \mu_{ij}\right)$ and $H_{ij}\left(d_{ij}, \mu_{ij}\right) = \lim_{\beta\to\infty} \frac{1}{\beta} \log M_{ij}\left(d_{ij}, \mu_{ij}\right)$ into (13) and (14) 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

$$\left(d^*_{ij}, \mu^*_{ij}\right) \in \arg\max_{\left(d_{ij}, \mu_{ij}\right)} H_{ij}\left(d_{ij}, \mu_{ij}\right) \tag{15}$$

$$H_{ij}\left(d_{ij}, \mu_{ij}\right) = h_{ij}\left(d_{ij}, \mu_{ij}\right) + h_{ji}\left(-d_{ij}, \mu_{ij}\right) - C' \tag{16}$$

where C' is an additive constant that guarantees that normalization condition in the zero-temperature limit, i.e. $\max_{\left(d_{ij}, \mu_{ij}\right)} H_{ij}\left(d_{ij}, \mu_{ij}\right) = 0$, is satisfied. In practice, converge is reached when the *decisional* variables computed as in (15) do not change after a predefined number of successive iterations (often $10 \div 30$). Notice that taking the $\beta \to +\infty$ limit of the message-passing equations at finite β 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 β and in the limit $\beta \to +\infty$, depending on we are dealing with the V-DStP or the E-DStP problem.

V. THE CAVITY EQUATIONS

A. Vertex-disjoint Steiner trees Problem

To derive the Belief Propagation equations for the V-DStP problem suffices to impose $\psi_i(\boldsymbol{d}_i, \boldsymbol{\mu}_i) = \psi_i^V(\boldsymbol{d}_i, \boldsymbol{\mu}_i)$ in (13). 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 $\psi_i^V(\boldsymbol{d}_i, \boldsymbol{\mu}_i)$; the explicit derivation in 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}:

$$\begin{cases} m_{ij}(d,\mu) = m_+^b(d,\mu) + m^f(d,\mu) & \forall d > 0, \mu \neq 0 \\ m_{ij}(d,\mu) = m_-^b(d,\mu) + m^f(d,\mu) & \forall d < 0, \mu \neq 0 \\ m_{ij}(0,0) = e^{-\beta \sum_\mu c_i^\mu} \prod_{k \in \partial i \setminus j} n_{ki}(0,0) + m_0^b + m_0^f \end{cases} \tag{17}$$

where $m_+^b(d,\mu)$, $m_-^b(d,\mu)$, $m^f(d,\mu)$, and m_0^b, m_0^f are defined as

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

Replacing $h_{ij}(d_{ij},\mu_{ij}) = \lim_{\beta \to +\infty} n_{ij}(d_{ij},\mu_{ij})$ in (17) we obtain the Max-Sum equations:

$$\begin{cases} h_{ij}(d,\mu) = \max\left\{h_+^b(d,\mu), h_+^f(d,\mu)\right\} & \forall d > 0, \mu \neq 0 \\ h_{ij}(d,\mu) = \max\left\{h_-^b(d,\mu), h_-^f(d,\mu)\right\} & \forall d < 0, \mu \neq 0 \\ h_{ij}(0,0) = \max\left\{-\sum_\mu c_i^\mu + \sum_{k \in \partial i \setminus j} h_{ki}(0,0),\ h_0^b,\ h_0^f\right\} \end{cases} \tag{18}$$

for

$$h_+^b(d,\mu) = -w_{ij} + \sum_{k\in\partial i\setminus j} \max\{h_{ki}(d+1,\mu), h_{ki}(0,0)\}$$

$$h_+^f(d,\mu) = -w_{ij} + \log\delta_{c_i^\mu,0} + \max_{k\in\partial i\setminus j}\left\{h_{ki}(d,\mu) + \sum_{l\in\partial i\setminus j} h_{li}(0,0)\right\}$$

$$h_-^b(d,\mu) = \max_{k\in\partial i\setminus j}\left[h_{ki}(d+1,\mu) - w_{ik} + \sum_{l\in\partial i\setminus\{j,k\}} \max\{h_{li}(d,\mu), h_{li}(0,0)\}\right]$$

$$h_-^f(d,\mu) = \log\delta_{c_i^\mu,0} + \max_{k\in\partial i\setminus j}\left\{h_{ki}(d,\mu) - w_{ik} + \sum_{l\in\partial i\setminus j} h_{li}(0,0)\right\}$$

$$h_0^b = \max_{\mu\neq 0}\max_{d<0} h_-^b(d,\mu)$$

$$h_0^f = \max_{\mu\neq 0}\max_{d<0}\left[\max_{\substack{k\in\partial i\setminus j,\\ l\in\partial i\setminus\{j,k\}}} h_{ki}(d,\mu) - w_{ik} + h_{li}(-d,\mu) + \sum_{m\in\partial i\setminus\{j,k,l\}} h_{mi}(0,0)\right]$$

B. Edge-disjoint Steiner trees problem

As for the V-DStP, the Belief Propagation equations for the E-DStP can be computed imposing $\psi_i(\boldsymbol{d}_i,\boldsymbol{\mu}_i) = \psi_i^V(\boldsymbol{d}_i,\boldsymbol{\mu}_i)$ into (13):

$$m_{ij}(d_{ij},\mu_{ij}) = \sum_{\substack{\{d_{ki},\mu_{ki}\}:\\ k\in\partial i\setminus j}} \psi_i^E(\boldsymbol{d}_i,\boldsymbol{\mu}_i)\, e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]} \prod_{k\in\partial i\setminus j} n_{ki}(d_{ki},\mu_{ki}) \tag{19}$$

Instead of considering the cavity messages as in V A, to compute (19) we will first define a partial partition function

$$Z_i = \sum_{\boldsymbol{d}_i,\boldsymbol{\mu}_i} \psi_i^E(\boldsymbol{d}_i,\boldsymbol{\mu}_i)\, e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]} \prod_{k\in\partial i} n_{ki}(d_{ki},\mu_{ki}) \tag{20}$$

and then calculate the set of messages $m_{ij}(d_{ij},\mu_{ij})$ through (20) by temporarily setting $n_{ji}(d_{ji},\mu_{ji}) = \delta_{-d_{ij},d_{ji}}\delta_{\mu_{ij},\mu_{ji}}$. Due to the explicit expression of ψ_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 (20) 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-DStP 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.

1. *Neighbors occupation formalism*

Suppose of associating with each vertex $i\in V$ a vector $\boldsymbol{x} = \{0,1\}^{|\partial i|}$. A feasible assignment of these auxiliary variables is guaranteed if, for every link $(i,k)\in E$ incident on i, we will impose $x_k = 1$ if this 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 $(\boldsymbol{d}_i,\boldsymbol{\mu}_i)$ must locally satisfy the following identity $\prod_{k\in\partial i}\mathbb{I}[x_k = 1-\delta_{d_{ki},0}] = 1$ for every node $i\in V$. If we insert this expression in (20) and we sum over all possible assignments of $\boldsymbol{x}$ variables we obtain

$$Z_i = \sum_{\boldsymbol{d}_i,\boldsymbol{\mu}_i} \psi_i^E\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]} \sum_{\boldsymbol{x}} \prod_{j\in\partial i} \mathbb{I}\left[x_j = 1-\delta_{d_{ji},0}\right] n_{ji}\left(d_{ji},\mu_{ji}\right) \tag{21}$$

$$= \sum_{\boldsymbol{x}} Z_{\mathbf{x}}^M \tag{22}$$

where $Z_{\boldsymbol{x}}^M$ is defined by computing the following expression for $q=M$

$$Z_{\boldsymbol{x}}^q \equiv \sum_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \mu_{ki}\leq q}} \psi_i^E\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]} \prod_{k\in\partial i} \mathbb{I}\left[x_k = 1-\delta_{d_{ki},0}\right] n_{ki}\left(d_{ki},\mu_{ki}\right) \tag{23}$$

The computation of $Z_{\boldsymbol{x}}^q$ is then performed using the following recursion (the equivalence of (24) to (23) is proven in appendix B)

$$Z_{\boldsymbol{x}}^q = \sum_{\boldsymbol{y}\leq\boldsymbol{x}} \left(g_{\boldsymbol{y}}^0 + g_{\boldsymbol{y}}^b + g_{\boldsymbol{y}}^f\right) Z_{\boldsymbol{y}}^{q-1} \tag{24}$$

$$Z_{\boldsymbol{x}}^0 = e^{-\beta\sum_\mu c_i^\mu} \prod_{j\in\partial i} \delta_{x_j,0} n_{ji}\left(0,0\right) \tag{25}$$

where the auxiliary functions $g_{\boldsymbol{y}}^0$, $g_{\boldsymbol{y}}^b$, $g_{\boldsymbol{y}}^f$ are defined as

$$g_{\boldsymbol{y}}^0 = e^{-\beta c_i^q} \prod_{\substack{k\in\partial i\\ y_k=0\\ x_k=1}} n_{ki}\left(0,0\right)$$

$$g_{\boldsymbol{y}}^b = \sum_{d>0} \sum_{\substack{j\in\partial i\\ y_j=0\\ x_j=1}} n_{ji}\left(-d,q\right) \prod_{\substack{k\in\partial i\setminus j\\ y_k=0\\ x_k=1}} \left[n_{ki}\left(d+1,q\right) + n_{ki}\left(0,0\right)\right]$$

$$g_{\boldsymbol{y}}^f = \delta_{c_i^q,0} \sum_{d>0} \sum_{\substack{j\in\partial i\\ y_j=0\\ x_j=1}} n_{ji}\left(-d,q\right) \sum_{\substack{k\in\partial i\setminus j\\ y_k=0\\ x_k=1}} n_{ki}\left(d,q\right) \prod_{\substack{l\in\partial i\setminus\{j,k\}\\ y_l=0\\ x_l=1}} n_{li}\left(0,0\right)$$

Notice that with $\boldsymbol{y}\leq\boldsymbol{x}$ we denote all possible vectors $\boldsymbol{y} = \{0,1\}^{|\partial i|}$ satisfying

$$y_k = \begin{cases} y_k \leq x_k & \text{if } \mu_{ki}\neq q\\ 0 & \text{if } \mu_{ki} = q \end{cases} \tag{26}$$

We can also write the expressions above in the Max-Sum formalism. Define $F_i = \lim_{\beta\to+\infty}\frac{1}{\beta}\log Z_i$ and express it as function of Max-Sum messages $h_{ij}\left(d_{ij},\mu_{ij}\right) = \lim_{\beta\to+\infty}\frac{1}{\beta}\log n_{ij}\left(d_{ij},\mu_{ij}\right)$ as

$$F_i = \max_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \psi_i^E(\boldsymbol{d}_i,\boldsymbol{\mu}_i)=1}} \max_{\boldsymbol{x}} \left[\sum_{k\in\partial i} \log\mathbb{I}\left[x_k = 1-\delta_{d_{ki},0}\right] + h_{ki}\left(d_{ki},\mu_{ki}\right) - \sum_\mu c_i^\mu \mathbb{I}\left[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}\right]\right] \tag{27}$$

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

$$F_i = \max_{\boldsymbol{x}} F_{\boldsymbol{x}}^M \tag{28}$$

with

$$F_{\boldsymbol{x}}^M = \max_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \psi_i^E(\boldsymbol{d}_i,\boldsymbol{\mu}_i)=1}} \sum_{k\in\partial i} \left[\log\mathbb{I}\left[x_k = 1-\delta_{d_{ki},0}\right] + h_{ki}\left(d_{ki},\mu_{ki}\right) - \sum_\mu c_i^\mu \mathbb{I}\left[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}\right]\right]$$

It is computed recursively from

$$F_{\boldsymbol{x}}^{q} = \max_{\boldsymbol{y}\leq\boldsymbol{x}} \left\{ F_{\boldsymbol{y}}^{q-1} + \max\left\{h_0,\, h_b,\, h_f\right\}\right\} \tag{29}$$

$$F_{\mathbf{x}}^{0} = -\sum_{\mu} c_i^{\mu} + \log \mathbb{I}\left[\boldsymbol{x}=\mathbf{0}\right] + \sum_{k\in\partial i} h_{ki}\left(0,0\right) \tag{30}$$

where

$$h_0 = \sum_{\substack{k\in\partial i\\ y_k=0\\ x_k=1}} h_{ki}\left(0,0\right) - c_i^q \tag{31}$$

$$h_b = \max_{d>0} \max_{\substack{k\in\partial i\\ y_k=0\\ x_k=1}} \left[h_{ki}\left(-d,q\right) + \sum_{\substack{l\in\partial i\setminus k\\ y_l=0\\ x_l=1}} \max\left[h_{li}\left(d+1,q\right), h_{li}\left(0,0\right)\right]\right] \tag{32}$$

$$h_f = \log \delta_{c_i^q,0} + \max_{d>0} \left[\max_{\substack{k\in\partial i,\, l\in\partial i,\, k\neq l\\ y_k=0,\, y_l=0\\ x_k=1\, x_l=1}} h_{ki}\left(-d,q\right) + h_{li}\left(d,q\right) + \sum_{\substack{m\in\partial i\setminus\{k,l\}\\ y_m=0\\ x_m=1}} h_{mi}\left(0,0\right)\right] \tag{33}$$

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 $\boldsymbol{s} \in \{0,1,\ldots,D\}^M$ associated with each vertex of the graph. Components s_μ take value in the set of the possible positive depths $\{1,\ldots,D\}$ if this node is member of communication μ or 0 otherwise. For a node i that is not a root but a member of the communication μ, there exists exactly one neighbor k such that $d_{ik} > 0$, $d_{ki} = -s_{\mu_{ki}}\, \mu_{ki} = \mu$ and for the remaining ones, $d_{li}\delta_{\mu_{li},\mu} = s_{\mu_{ki}} + 1$ or $d_{li}\delta_{\mu_{li},\mu} = 0$, $l \in \partial i\setminus k$. The compatibility function for E-DStP can be expressed as a function of the new variables as

$$\psi_i^E\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right) = \prod_{\mu=1}^{M} \left[\sum_{s_\mu>0} \sum_{k\in\partial i} \delta_{\tilde{d}_{ki},-s_\mu} \prod_{l\in\partial i\setminus k} \left(\delta_{\tilde{d}_{li},s_\mu+1} + \delta_{\tilde{d}_{li},0}\right) + \prod_{k\in\partial i} \delta_{\tilde{d_{ki}},0}\right] \tag{34}$$

$$= \sum_{\mathbf{s}} \left\{ \prod_{\mu=1}^{M} \left(1-\delta_{s_\mu,0}\right) \sum_{k\in\partial i} \left[\delta_{\tilde{d}_{ki},-s_\mu} \prod_{l\in\partial i\setminus k} \left(\delta_{\tilde{d}_{li},s_\mu+1} + \delta_{\tilde{d}_{li},0}\right)\right] + \prod_{\mu=1}^{M} \delta_{s_\mu,0} \prod_{k\in\partial i} \delta_{\tilde{d_{ki}},0} \right\} \tag{35}$$

If now make explicit the dependency on $\boldsymbol{\mu}_i$ we obtain

$$\psi_i^E\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right) = \sum_{\mathbf{s}} \left\{ \prod_{\mu:s_\mu>0} \sum_{k\in\partial i} \delta_{\mu_{ki},\mu}\delta_{d_{ki},-s_\mu} \prod_{l\in\partial i\setminus k} \left[\delta_{\mu_{li},\mu}\delta_{d_{li},s_\mu+1} + \left(1-\delta_{\mu_{li},\mu}\right)\right] + \right. \tag{36}$$

$$\left. + \prod_{\mu:s_\mu=0} \prod_{k\in\partial i} \left(1-\delta_{\mu_{ki},\mu}\right) \delta_{d_{ki},s_{\mu_{ki}}} \right\} \tag{37}$$

Now, introduce:

$$Z_i = \sum_{\boldsymbol{d},\boldsymbol{\mu}_i} \psi_i^E\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right) e^{-\beta \sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}]} \prod_{k\in\partial i} n_{ki}(d_{ki}, \mu_{ki}) \tag{38}$$

$$= \sum_{\mathbf{s}} Q_{\mathbf{s}} \tag{39}$$

where

$$Q_{\mathbf{s}} = \sum_{\boldsymbol{d}_i} \sum_{\left\{\mu_{ki}: s_{\mu_{ki}} > 0 \vee \mu_{ki} = 0\right\}} \prod_{k \in \partial i} n_{ki}\left(d_{ki}, \mu_{ki}\right) \left\{ \prod_{\mu: s_\mu > 0} e^{-\beta \sum_{\nu \neq \mu} c_i^\nu \mathbb{I}[\boldsymbol{\mu}_i \neq \boldsymbol{\nu}]} \sum_{k \in \partial i} f_{k\mu} + e^{-\beta \sum_\mu c_i^\mu} \prod_{k \in \partial i} \delta_{\mu_{ki}, 0} \right\} \tag{40}$$

$$f_{k\mu} = \delta_{\mu_{ki}, \mu} \delta_{d_{ki}, -s_\mu} \prod_{l \in \partial i \setminus k} \left[\delta_{\mu_{li}, \mu} \delta_{d_{li}, s_\mu + 1} + \left(1 - \delta_{\mu_{li}, \mu}\right)\right] \tag{41}$$

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

$$\delta_{\mu_{ki}, \mu} \delta_{d_{ki}, -s_\mu} \left[\delta_{\mu_{ki}, \nu} \delta_{d_{ki}, s_\nu + 1} + \left(1 - \delta_{\mu_{ki}, \nu}\right)\right] = \delta_{\mu_{ki}, \mu} \delta_{d_{ki}, -s_\mu} \tag{42}$$

and equivalently

$$\prod_\nu \left[\delta_{\mu_{ki}, \nu} \delta_{d_{ki}, s_\nu + 1} + \left(1 - \delta_{\mu_{ki}, \nu}\right)\right] = \sum_\nu \delta_{\mu_{ki}, \nu} \delta_{d_{ki}, s_\nu + 1} + \delta_{\mu_{ki}, 0} \delta_{d_{ki}, 0} \tag{43}$$

Thus

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

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

$$\pi : \{\mu : s_\mu > 0\} \to \partial i$$

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

$$\begin{aligned}
\prod_\mu f_{\pi_\mu \mu} &= \prod_\mu \delta_{\mu_{\pi_\mu i}, \mu} \delta_{d_{\pi_\mu i}, -s_\mu} \prod_{l \in \partial i \setminus \pi_\mu} \left[\delta_{\mu_{li}, \mu} \delta_{d_{li}, s_\mu + 1} + \left(1 - \delta_{\mu_{li}, \mu}\right)\right] \\
&= \prod_{k \in \partial i} \left(\sum_\nu \delta_{\mu_{ki}, \nu} \delta_{d_{ki}, s_\nu + 1} + \delta_{\mu_{ki}, 0} \delta_{d_{ki}, 0}\right)^{1 - \sum_\nu t_{k\nu}} \prod_\mu \left(\delta_{\mu_{ki}, \mu} \delta_{d_{ki}, -s_\mu}\right)^{t_{k\mu}}
\end{aligned}$$

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

$$\begin{aligned}
Z_i &= \sum_{\boldsymbol{s}} \sum_{\boldsymbol{t}} \prod_\mu e^{-\beta c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}]} \mathbb{I}\left[\sum_{k \in \partial i} t_{k\mu} = 1 - \delta_{s_\mu, 0}\right] \prod_{k \in \partial i} \mathbb{I}\left[\sum_\mu t_{k\mu} \leq 1\right] \times && (44) \\
&\quad \times \prod_{k \in \partial i} \left[\sum_\nu n_{ki}\left(s_\nu + 1, \nu\right) + n_{ki}(0,0)\right]^{1 - \sum_\nu t_{k\nu}} \prod_\mu n_{ki}\left(-s_\mu, \mu\right)^{t_{k\mu}} && (45) \\
&= \sum_{\boldsymbol{s}} R_{\boldsymbol{s}} Z_{\boldsymbol{s}} && (46)
\end{aligned}$$

where

$$R_{\boldsymbol{s}} = \prod_{k \in \partial i} \left[\sum_\nu n_{ki}\left(s_\nu + 1, \nu\right) + n_{ki}(0,0)\right] \tag{47}$$

$$Z_{\boldsymbol{s}} = \sum_{\boldsymbol{t}} \prod_\mu e^{-\beta c_i^\mu \mathbb{I}[s_\mu = 0]} \mathbb{I}\left[\sum_{k \in \partial i} t_{k\mu} = 1 - \delta_{s_\mu, 0}\right] \prod_{k \in \partial i} \mathbb{I}\left[\sum_\mu t_{k\mu} \leq 1\right] \prod_{k \in \partial i} \left[\frac{n_{ki}\left(-s_\mu, \mu\right)}{\sum_\nu n_{ki}\left(s_\nu + 1, \nu\right) + n_{ki}(0,0)}\right]^{t_{k\mu}} \tag{48}$$

The term $Z_{\boldsymbol{s}}$ 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

$$\epsilon(\boldsymbol{t}) = \sum_{k\mu} t_{k\mu} \log \frac{n_{ki}(-s_\mu, \mu)}{\sum_\nu n_{ki}(s_\nu + 1, \mu) + n_{ki}(0,0)} - \beta c_i^\mu \mathbb{I}[s_\mu = 0]$$

In general, the partition function $Z_{\mathbf{s}}$ is hard to compute exactly, because it corresponds to the calculation of a matrix *permanent* which is computationally intractable. Fortunately, the situation is much easier in the $\beta \to \infty$ limit: using $h_{ki}(-s_\mu, \mu) = \frac{1}{\beta} \log n_{ki}(-s_\mu, \mu)$ and taking the limit $\beta \to \infty$, the computation of $F_i = \frac{1}{\beta} \log Z_i$ reduces to the evaluation of

$$F_i = \max_{\boldsymbol{s}} \left[\frac{1}{\beta} (\log R_{\boldsymbol{s}} + \log Z_{\boldsymbol{s}}) \right] \tag{49}$$

$$= \max_{\mathbf{s}} \left\{ \sum_{k \in \partial i} \max \left[\max_\mu h_{ki}(s_\mu + 1, \mu),\ h_{ki}(0,0) \right] + F_{\mathbf{s}} \right\} \tag{50}$$

To evaluate the second term $F_{\boldsymbol{s}} = \frac{1}{\beta} \log Z_{\boldsymbol{s}}$ we need to solve a weighted maximum matching problem on a bipartite graph which can be done in polynomial time (precisely, it can be performed in $O\left((M + |\partial i|)^2 M |\partial i|\right)$). Indeed, for each assignment of the $\boldsymbol{s}$ we can define the weights $w_{k\mu}$ associated with each edge (k, μ) as:

$$w_{k\mu} = \begin{cases} h_{ki}(-s_\mu, \mu) - \max_\nu \max\{h_{ki}(s_\nu + 1, \nu), h_{ki}(0,0)\} & \text{if } s_\mu > 0 \\ -c_i^\mu & \text{if } s_\mu = 0 \end{cases} \tag{51}$$

and solve

$$\begin{cases} F_{\boldsymbol{s}} = \max \sum_{(k,\mu)} w_{k\mu} t_{k\mu} & : \\ \sum_{k \in \partial i} t_{k\mu} \leq 1 & \forall \mu \\ \sum_\mu t_{k\mu} \leq 1 & \forall k \in \partial i \end{cases} \tag{52}$$

That the system in (52) for $t_{k\mu} \in \{0, 1\}$ corresponds to a maximum matching problem on a bipartite graph, and can be solved efficiently through e.g. its linear program relaxation.

C. The parameter D

The *branching* formalism introduced in III 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 this parameter needs to be carefully designed to ensure good performances. Although there is not a technique able to predict the best value of D, 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_μ for each communication μ 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 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 μ. A proof of this property is reported in [8] for the single tree problem.

VI. MAX-SUM FOR LOOPY GRAPHS

The goodness of the approximation of 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 VI A how to modify the Max-Sum equations for the V-DStP and E-DStP for including the reinforcement factor.

The reinforcement scheme described in the following 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 VI B how to generalize the combination of Max-Sum and heuristics in the case of multiple trees for the V-DStP and the E-DStP.

A. Reinforcement

At each iteration t of the algorithm we modify the update equations for cavity marginals incident on node i as

$$\tilde{h}_{ji}^{t}\left(d_{ji},\mu_{ji}\right) \propto h_{ji}^{t}\left(d_{ij},\mu_{ji}\right)+\gamma_t H_{ji}^{t-1}\left(d_{ji},\mu_{ji}\right) \tag{53}$$

and we compute the cavity fields as

$$H_{ij}^{t}\left(d_{ij},\mu_{ij}\right) \propto h_{ij}^{t}\left(d_{ij},\mu_{ij}\right)+h_{ji}^{t}\left(-d_{ij},\mu_{ij}\right)+\gamma_t H_{ij}^{t-1}\left(d_{ij},\mu_{ij}\right) \tag{54}$$

where $\gamma_t = t\gamma_0$ is linearly proportional to γ_0, 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.

B. 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 μ we apply one of the two schemes as follows.

1. Shortest Path Tree

For any MS iteration t we compute the auxiliary weights as

$$w_{ij}^{t} = \max_{d\neq 0}\left|H_{ij}^{t}\left(d,\mu\right)\right| \tag{55}$$

Notice that since marginals H^t_{ij} are normalized, there exists only one assignment of the variables such that $H^t_{ij}(d^*,\mu) = 0$ in correspondence of the most probable state (d^*,μ). According to (55) edges that are likely to be exploited within communication μ, i.e. such that $d^* \neq 0$, will have zero weights; differently, we penalize edges that, according to MS, must not be used (for which $d^* = 0$) imposing strictly positive weights equal to the MS marginals computed at the most probable non-zero depth. 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.

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

$$\begin{cases} h_i(d,\mu) = \max_{k\in\partial i}\left\{h^t_{ik}(-d,\mu) + \sum_{l\in\partial i\setminus k}\max\left[h^t_{li}(d+1,\mu), h^t_{li}(0,\mu)\right]\right\} & \text{for } d>0 \\ h_i(0,\mu) = \sum_{k\in\partial i} h^t_{ki}(0,\mu) - c^\mu_i \end{cases} \tag{56}$$

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, as a feasible solution for E-DStP or 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.

VII. NUMERICAL RESULTS

In this section we report 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 in which we will make use of the single-tree MS algorithm for the MStP. 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 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 available 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.

A. 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 following "greedy" procedure: we apply MS algorithm for the single-tree MStP to each communications, one at the time, and we compute the "greedy" energy as the sum of energies of single-tree solutions. As in the case of the heuristics described in VI B we must perform a pre-processing of the graph before the application of the single-tree algorithm in order not to use edges in more than one sub-graphs. Notice that this "greedy" procedure is actually as hard as the packing problem, since even solving 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.

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 $\alpha = \frac{T_\mu}{N} = 0.08$, $\mu \in \{1, 2, 3\}$ 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 $\gamma_0 = 10^{-5}$.

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 then 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 on page 17. Notice that the gaps are slightly greater then 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 on page 17, left panel.

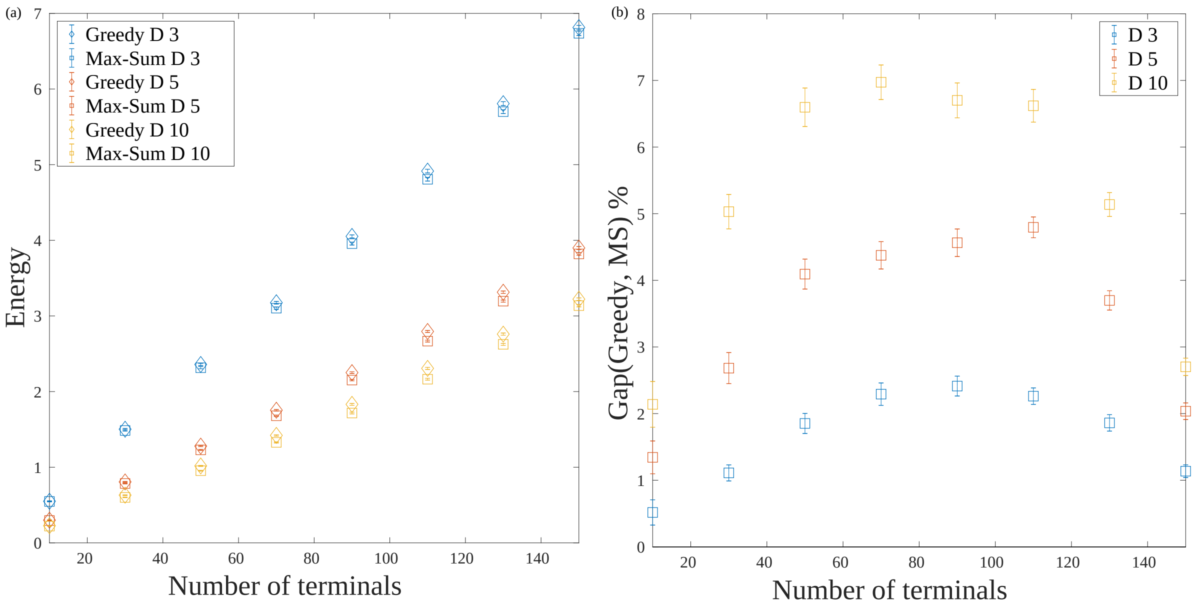


Figure 3. Energy (a) and energy gap (b) of the solutions of Max Sum and Greedy algorithm as functions of the number of terminals. The test instances are fully connected graphs of 500 nodes with uncorrelated edge weights. Gaps reported in panel (b) are always positive suggesting that solutions found by the global search are cheaper in terms of cost then the greedy ones.

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 $w_{ij} = x_i x_j y_{ij}$. In this scenario, 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 the more and more forced to use the remaining expensive edges. In fact, as shown in Figure 4 on page 18, 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.

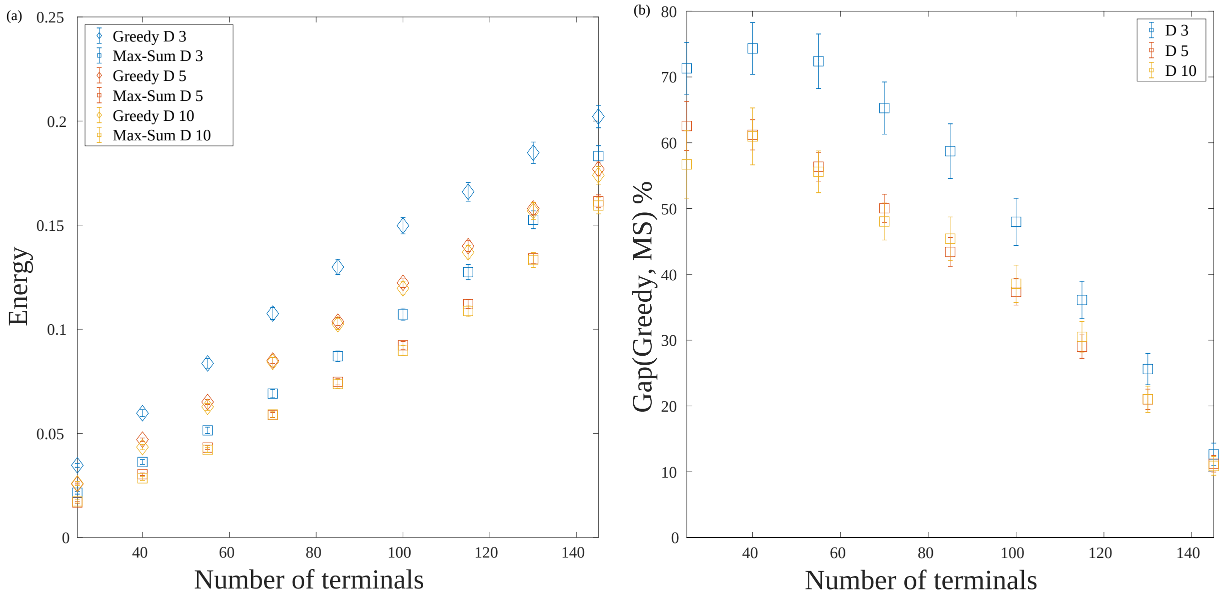


Figure 4. Energy (a) and energy gap (b) for Max Sum results against Greedy results as functions of the number of terminals for correlated edge weighting. The energy gaps of panel (b) are positive and notably large.

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 μ and constant α, 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 on page 19 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 on page 19 panel (b), also energy gaps fluctuates around a fixed value that seems to be different if one considers $D = 3$ or $D = \{5, 10\}$.

B. Regular graphs

In V B 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 $\gamma_0 = 10^{-4}$.

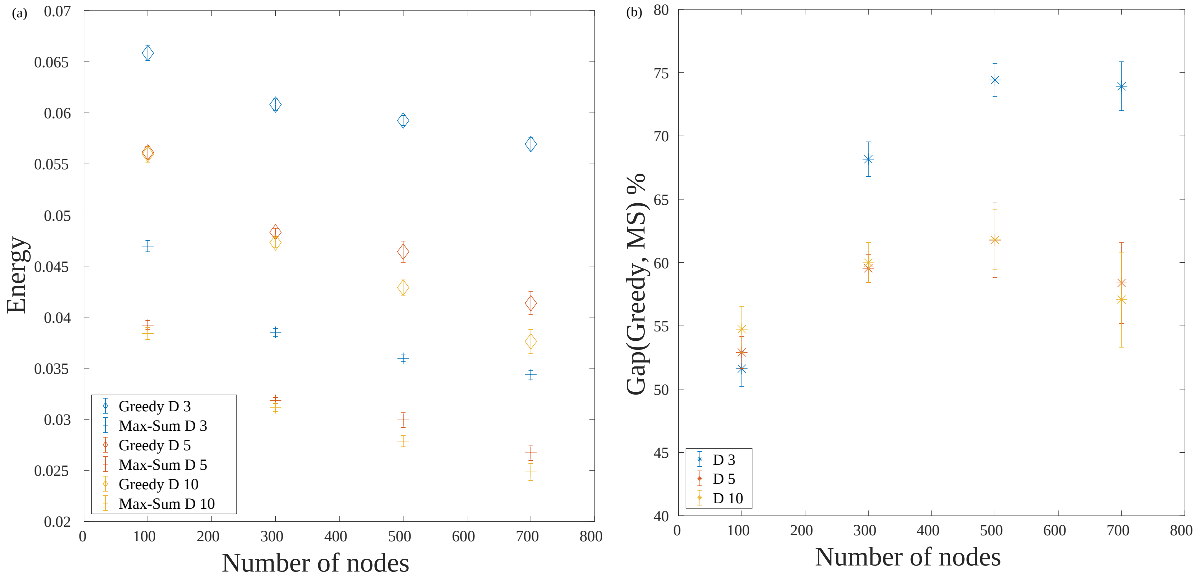


Figure 5. Energy (a) and energy gap (b) for Max Sum against Greedy results as functions of the number of nodes for a fixed fractions of terminals (per communication)$\alpha = 0.08$.

1. *Energy as a function of the degree*

Similarly to the experiments in VII B 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 on page 20, 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 Figure 6 on page 20) scales exponentially only for the *NeighOcc* (as it is remarked by the linear trend in the semi-log plot) while it scales polynomially for the *Matching* formalism as predicted by the analysis on the update equations in V B.

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 on page 20, 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 on page 20, 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.

C. 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 III 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 VI B applied to several benchmark instances for the design problem of VLSI circuits.

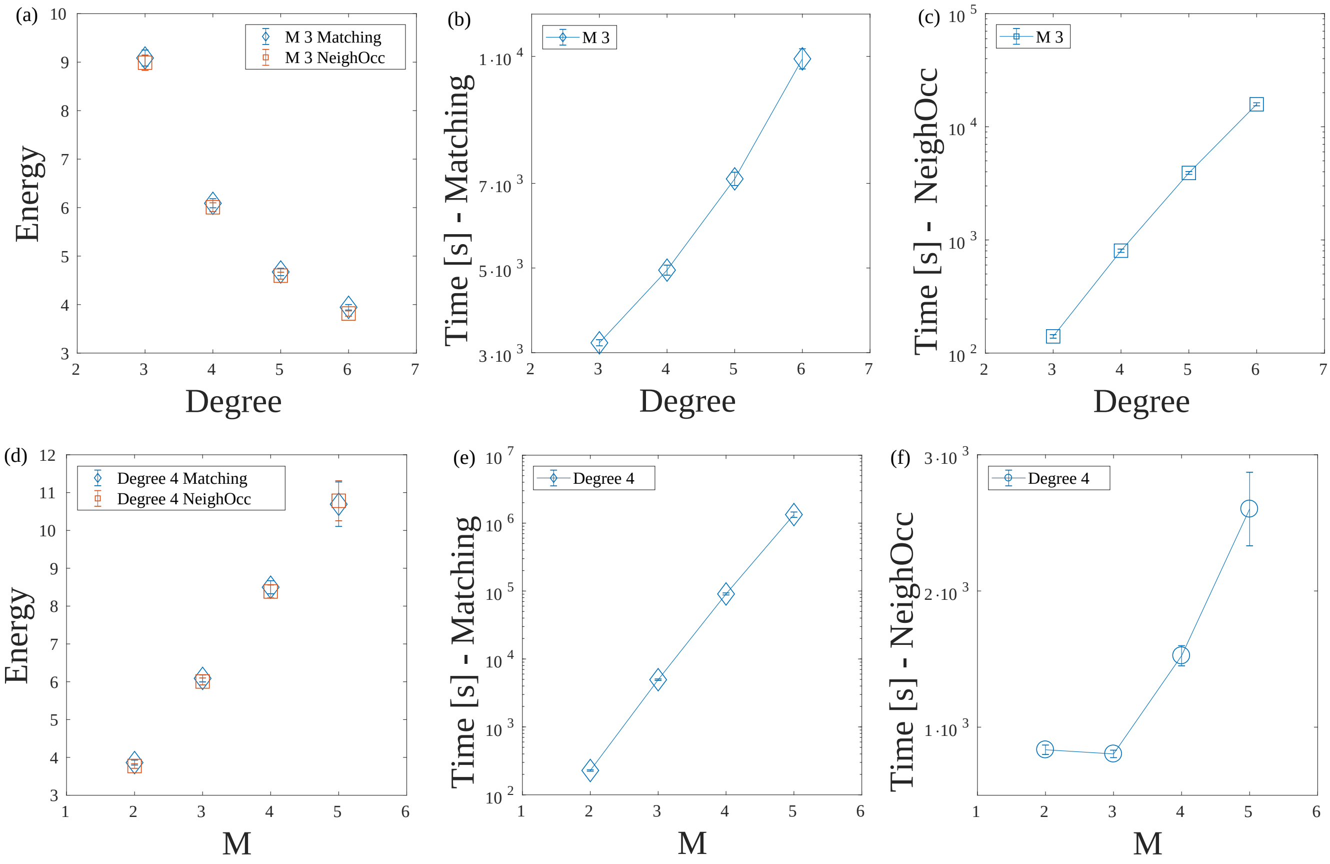


Figure 6. In panel (a), energy of the solutions for the E-DStP on regular graphs of 3 communications as a function of the degree. In (b) and (c) running time of *Matching* and *NeighOcc* algorithms as a functions of degree. In panel (d): energy of the solutions for the E-DStP on regular graphs as a function of the number of packed trees. Panels (e) and (f): running time of the *Matching* and *NeighOcc* algorithms as a functions of the number of communications. In all plots energies of the solutions are almost the same, but the computing time dramatically differ as we are using the *Matching* formalism or the *NeighOcc* algorithm.

1. *Branching and flat models for the V-DStP e E-DStP (neighbors occupation formalism)*

As shown in Figure 7 on page 21, 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 on page 21, the flat version of MS equations has the advantage of converging in a running time that is always smaller then 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 on page 21, 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 then the one needed by the branching model for all the cases we have considered.

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

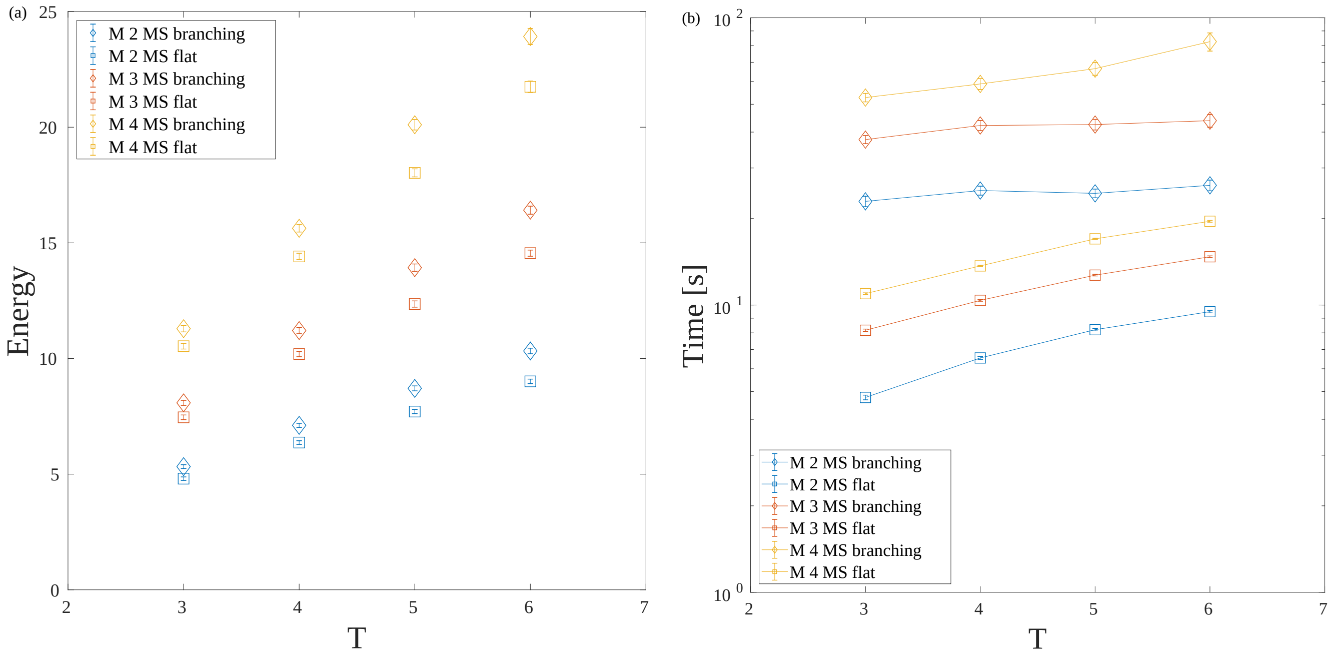


Figure 7. Energy (a) and computational time (b) as a function of the number of terminals per communications for 3D grid graphs, V-DStP variant.

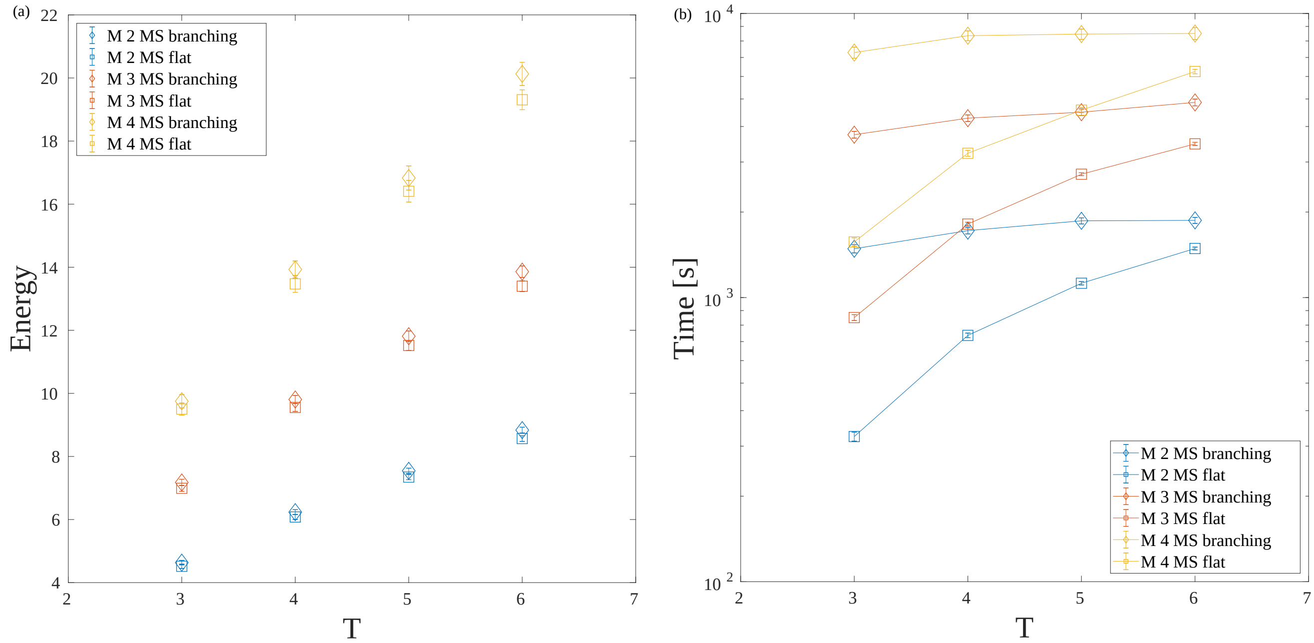


Figure 8. Energy (a) and computational time (b) as a function of the number of terminals per communications for 3D grid graphs, E-DStP variant.

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, 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 I on page 22 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 VI B; 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 *flat* model are labeled as "(f)" while if nothing is specified or "(b)" is used, we made use of the *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 I on page 22, the gaps are always smaller than 4% and in two cases, for the multi-aligned *augmenteddense-2* and *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 we achieve good performances thanks to the reinforcement procedure along with the introduction of the modified heuristics.

	Type	Size	M	T_{tot}	Heur. "J"	Heur. "N"	Rein. Max Sum	LP (opt)	Gap %
augmenteddense-2	Multi-aligned	16x18x2	19	59	504 (b) 506 (f)	507	508 (f) 504 (b)	504	0 %
augmenteddense-2	Multi-crossed	16x18x2	19	59	503	-	-	498	1.0 %
dense-3	Multi-crossed	15x17x3	19	59	487	488	485	464	4.0 %
difficult-2	Multi-aligned	23x15x2	24	66	535	538	538	526	1.7 %
difficult-2x	Multi-aligned	23x15x2	24	66	560	-	-	unknown	
difficult-2y	Multi-aligned	23x15x2	24	66	4776	4829	4816	unknown	
difficult-2z	Multi-aligned	23x15x2	24	66	1060	1063	1061	unknown	
modifieddense-3	Multi-crossed	16x17x3	19	59	492	496	495	479	2.6 %
moredifficult-2	Multi-aligned	22x15x2	24	65	542	542	546	522	3.8 %
pedabox-2	Multi-aligned	15x16x2	22	56	405	405	405	390	3.8 %
terminalintensive-2	Multi-aligned	23x16x2	24	77	596 (f) 599 (b)	617	620	596	0 %

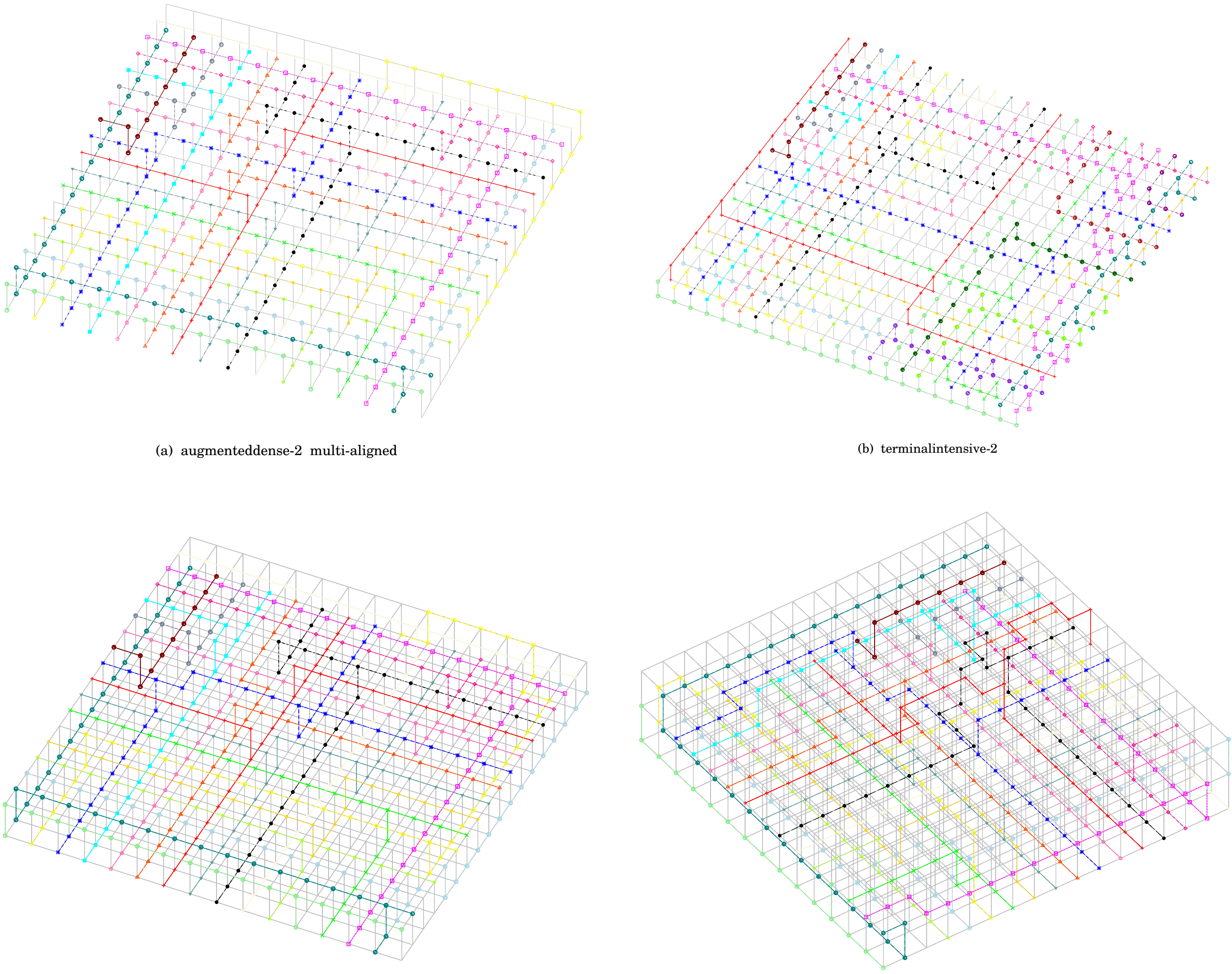
Table I. Results for circuit layout instances

VIII. 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 2×10^5 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 2d and 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.



(a) augmenteddense-2 multi-aligned

(b) terminalintensive-2

(c) augmenteddense-2 multi-crossed

(d) dense-3

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

ACKNOWLEDGMENTS

We acknowledge Fondazione CRT for project SIBYL under the initiative “La ricerca dei Talenti”. We warmly thank F. Altarelli for interesting discussions and R. Zecchina for suggesting this topic.

---

[1] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall’Asta, Caterina De Bacco, and Silvio Franz. The edge-disjoint path problem on random graphs by message-passing. *CoRR*, abs/1503.00540, 2015.

[2] Omer Angel, Abraham D. Flaxman, and David B. Wilson. A sharp threshold for minimum bounded-depth and bounded-diameter spanning trees and Steiner trees in random networks. *Combinatorica*, 32(1):1–33, January 2012.

[3] C. De Bacco, S. Franz, D. Saad, and C. H. Yeung. Shortest node-disjoint paths on random graphs. *J. Stat. Mech.*, 2014(7):P07009, July 2014.

[4] M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J.-M. François, and R. Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *Proceedings of the National Academy of Sciences*, 108(2):882–887, January 2011.

[5] M. Bayati, C. Borgs, A. Braunstein, J. Chayes, A. Ramezanpour, and R. Zecchina. Statistical Mechanics of Steiner Trees.

Physical Review Letters, 101(3):037208, July 2008.

[6] Mohsen Bayati, A. Braunstein, and Riccardo Zecchina. A rigorous analysis of the cavity equations for the minimum spanning tree. *Journal of Mathematical Physics*, 49(12):125206, 2008. Cited by 0012.

[7] Indaco Biazzo, Alfredo Braunstein, and Riccardo Zecchina. Performance of a cavity-method-based algorithm for the prize-collecting Steiner tree problem on graphs. *Phys. Rev. E*, 86:026706, August 2012.

[8] Alfredo Braunstein and Anna Muntoni. Practical optimization of Steiner trees via the cavity method. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(7):073302, 2016.

[9] M. Burstein and R. Pelavin. Hierarchical Wire Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2(4):223–234, October 1983.

[10] J. P. Cohoon and P. L. Heck. BEAVER: a computational-geometry-based tool for switchbox routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(6):684–697, June 1988.

[11] M. Grötschel, A. Martin, and R. Weismantel. The steiner tree packing problem in VLSI design. *Mathematical Programming*, 78(2):265–281.

[12] Nam-Dũng Hoáng and Thorsten Koch. Steiner tree packing revisited. *Mathematical Methods of Operations Research*, 76(1):95–123, 2012.

[13] Richard M Karp. *Reducibility among combinatorial problems.* Springer, 1972.

[14] Ivana Ljubić, René Weiskircher, Ulrich Pferschy, Gunnar W. Klau, Petra Mutzel, and Matteo Fischetti. An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem. *Math. Program.*, 105(2-3):427–449, October 2005.

[15] Wing Kwong Luk. A greedy switch-box router. *INTEGRATION, the VLSI journal*, 3(2):129–149, 1985.

[16] Marc Mézard and Andrea Montanari. *Information, Physics, and Computation.* Oxford University Press, January 2009.

[17] Marc Mézard and Giorgio Parisi. The Cavity Method at Zero Temperature. *Journal of Statistical Physics*, 111(1-2):1–34, April 2003.

[18] Y. Weiss and W.T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *Information Theory, IEEE Transactions on*, 47(2):736–744, February 2001.

Appendix A: Message-Passing equations for V-DStP

Consider the compatibility function for node i

$$\begin{aligned}\psi_i^V(\boldsymbol{d}_i,\boldsymbol{\mu}_i) = \prod_{k\in\partial i}\delta_{\mu_{ki},0}\delta_{d_{ki},0} + \sum_{\mu=1}^{M}\sum_{d>0}\sum_{k\in\partial i}\delta_{\mu_{ki},\mu}\delta_{d_{ki},-d}\prod_{l\in\partial i\setminus k}\left(\delta_{\mu_{li},\mu}\delta_{d_{li},d+1}+\delta_{\mu_{li},0}\delta_{d_{li},0}\right)+ \\ +\sum_{\mu}\delta_{c_i^\mu,0}\sum_{d>0}\sum_{k\in\partial i}\delta_{\mu,\mu_{ki}}\delta_{-d,d_{ki}}\sum_{l\in\partial i\setminus k}\delta_{\mu,\mu_{li}}\delta_{d_{li},d}\prod_{m\in\partial i\setminus\{k,l\}}\delta_{\mu_{mi},0}\delta_{d_{mi},0}\end{aligned} \tag{A1}$$

For sake of simplicity we split $\psi_i^V(\boldsymbol{d},\boldsymbol{\mu})$ in:

$$\psi_i^V(\boldsymbol{d}_i,\boldsymbol{\mu}_i) = \psi_i^{(1)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i)+\psi_i^{(2)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i)+\psi_i^{(3)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i) \tag{A2}$$

where

$$\psi_i^{(1)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i) = \prod_{k\in\partial i}\delta_{\mu_{ki},0}\delta_{d_{ki},0} \tag{A3}$$

$$\psi_i^{(2)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i) = \sum_{\mu=1}^{M}\sum_{d>0}\sum_{k\in\partial i}\left[\delta_{\mu_{ki},\mu}\delta_{d_{ki},-d}\prod_{l\in\partial i\setminus k}\left(\delta_{\mu_{li},\mu}\delta_{d_{li},d+1}+\delta_{\mu_{li},0}\delta_{d_{li},0}\right)\right] \tag{A4}$$

$$\psi_i^{(3)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i) = \sum_{\mu}\delta_{c_i^\mu,0}\sum_{d>0}\sum_{k\in\partial i}\delta_{\mu,\mu_{ki}}\delta_{-d,d_{ki}}\sum_{l\in\partial i\setminus k}\delta_{\mu,\mu_{li}}\delta_{d_{li},d}\prod_{m\in\partial i\setminus\{k,l\}}\delta_{\mu_{mi},0}\delta_{d_{mi},0} \tag{A5}$$

Consider the action of $\psi_i^{(1)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i)$, $\psi_i^{(2)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i)$ and $\psi_i^{(3)}(\boldsymbol{d}_i,\boldsymbol{\mu}_i)$ inside (13) and compute $m_{ij}(d_{ij},\mu_{ij}) = m_{ij}^{(1)}(d_{ij},\mu_{ij})+m_{ij}^{(2)}(d_{ij},\mu_{ij})+\delta_{c_i^\mu,0}m_{ij}^{(3)}(d_{ij},\mu_{ij})$.

$$m_{ij}^{(1)}(d_{ij},\mu_{ij}) = \sum_{\substack{\{d_{ki},\mu_{ki}\}:\\ k\in\partial i\setminus j}} e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]}\prod_{l\in\partial i}\delta_{\mu_{li},0}\delta_{d_{li},0}\prod_{k\in\partial i\setminus j}n_{ki}(d_{ki},\mu_{ki}) \tag{A6}$$

$$= e^{-\beta\sum_\mu c_i^\mu}\delta_{\mu_{ij},0}\delta_{d_{ij},0}\prod_{k\in\partial i\setminus j}m_{ki}(0,0) \tag{A7}$$

$$m_{ij}^{(2)}(d_{ij},\mu_{ij}) = \sum_{\mu}e^{-\beta\sum_\mu c_i^\mu\mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]}\sum_{d>0}\Bigg\{\delta_{d_{ji},-d}\delta_{\mu_{ji},\mu}\prod_{k\in\partial i\setminus j}\left[n_{ki}(d+1,\mu)+n_{ki}(0,0)\right]+ \tag{A8}$$

$$+\left(\delta_{d_{ji},d+1}\delta_{\mu_{ij},\mu}+\delta_{d_{ji},0}\delta_{\mu_{ji},0}\right)\sum_{k\in\partial i\setminus j}n_{ki}(-d,\mu)\prod_{l\in\partial i\setminus\{j,k\}}\left[n_{li}(d+1,\mu)+n_{li}(0,0)\right]\Bigg\} \tag{A9}$$

$$m_{ij}^{(3)}(d_{ij},\mu_{ij}) = \sum_{\mu}\sum_{d>0}\Bigg[\delta_{\mu,\mu_{ji}}\delta_{d_{ji},-d}\sum_{k\in\partial i\setminus j}n_{ki}(d,\mu)\prod_{l\in\partial i\setminus\{j,k\}}n_{li}(0,0)+ \tag{A10}$$

$$+\delta_{\mu,\mu_{ji}}\delta_{d_{ji},d}\sum_{k\in\partial i\setminus j}n_{ki}(-d,\mu)\prod_{l\in\partial i\setminus\{j,k\}}n_{li}(0,0)+ \tag{A11}$$

$$+\delta_{\mu,\mu_{ji}}\delta_{d_{ji},0}\sum_{k\in\partial i\setminus j}n_{ki}(d,\mu)\sum_{l\in\partial i\setminus\{j,k\}}n_{li}(-d,\mu)\prod_{m\in\partial i\setminus\{k,l,j\}}n_{mi}(0,0)\Bigg] \tag{A12}$$

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

$$\begin{cases} m_{ij}(d,\mu) = \prod_{k\in\partial i\setminus j}\left[n_{ki}(d+1,\mu)+n_{ki}(0,0)\right]+\delta_{c_i^\mu,0}\sum_{k\in\partial i\setminus j}n_{ki}(d,\mu)\prod_{l\in\partial i\setminus\{j,k\}}n_{li}(0,0) & \forall d > (\\ m_{ij}(d,\mu) = \sum_{k\in\partial i\setminus j}n_{ki}(d+1,\mu)\prod_{l\in\partial i\setminus\{j,k\}}\left[n_{li}(d,\mu)+n_{li}(0,0)\right]+\delta_{c_i^\mu,0}\sum_{k\in\partial i\setminus j}n_{ki}(d,\mu)\prod_{l\in\partial j\setminus\{j,k\}}n_{li}(0,0) & \forall d < (\\ m_{ij}(0,0) = e^{-\beta\sum_\mu c_i^\mu}\prod_{k\in\partial i\setminus j}n_{ki}(0,0)+\sum_{\mu\neq0}\sum_{d<0}\sum_{k\in\partial i\setminus j}n_{ki}(d+1,\mu)\prod_{l\in\partial i\setminus\{j,k\}}\left[n_{li}(d,\mu)+n_{li}(0,0)\right] & < -AN \end{cases} \tag{A13}$$

For $d=\mu=0$

$$m_{ij}\left(0,0\right)=e^{-\beta\sum_{\mu}c_{i}^{\mu}}\prod_{k\in\partial i\setminus j}n_{ki}\left(0,0\right)+\sum_{\mu\neq0}\sum_{d<0}\sum_{k\in\partial i\setminus j}n_{ki}\left(d+1,\mu\right)\prod_{l\in\partial i\setminus\{j,k\}}\left[n_{li}\left(d,\mu\right)+n_{li}(0,0)\right]+$$
$$+\sum_{\mu\neq0}\sum_{d<0}\sum_{k\in\partial i\setminus j}n_{ki}\left(d,\mu\right)\sum_{l\in\partial i\setminus\{j,k\}}n_{li}\left(-d,\mu\right)\prod_{m\in\partial i\setminus\{k,l,j\}}n_{mi}\left(0,0\right)$$

Appendix B: Recursive expression of Z^q for the E-DStP

From Eq. (23)

$$Z_{\boldsymbol{x}}^{q}=\sum_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \mu_{ki}\leq q,\,\forall k\in\partial i}}\psi_{i}^{E}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)e^{-\beta\sum_{\mu}c_{i}^{\mu}\mathbb{I}[\boldsymbol{\mu}_{i}\neq\boldsymbol{\mu}]}\prod_{k\in\partial i}\mathbb{I}\left[x_{k}=1-\delta_{d_{ki},0}\right]n_{ki}\left(d_{ki},\mu_{ki}\right)\tag{B1}$$

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

$$Z_{\boldsymbol{x}}^{q}=\sum_{\substack{\boldsymbol{d},_i\boldsymbol{\mu}_i\\ \mu_{ki}\leq q}}\psi_{i}^{E}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)e^{-\beta\sum_{\mu}c_{i}^{\mu}\mathbb{I}[\boldsymbol{\mu}_{i}\neq\boldsymbol{\mu}]}\prod_{\substack{k\in\partial i:\\ \mu_{ki}=q}}\mathbb{I}\left[x_{k}=1\right]\mathbb{I}\left[d_{ki}\neq0\right]n_{ki}\left(d_{ki},\mu_{ki}\right)\prod_{\substack{k\in\partial i:\\ \mu_{ki}\leq q-1}}\mathbb{I}\left[x_{k}=1-\delta_{d_{ki},0}\right]n_{ki}\left(d_{ki},\mu_{ki}\right)\tag{B2}$$

Consider a vector $\boldsymbol{x}$ such that there exists at least one component $x_k=1$ for $d_{ki}\neq0$, $\mu_{ki}=q$ and eventually 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 $\boldsymbol{y}\leq\boldsymbol{x}$, that is, all vectors having at most the same number of non-zeros of $\mathbf{x}$ 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 $\mu_{k'i}\leq q-1$. Thus:

$$Z_{\boldsymbol{x}}^{q}=\sum_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \mu_{ki}\leq q}}\sum_{\boldsymbol{y}\leq\boldsymbol{x}}e^{-\beta c_{i}^{q}\mathbb{I}[\boldsymbol{\mu}_{i}\neq\boldsymbol{q}]}\psi_{i}^{q}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)\prod_{\substack{k\in\partial i:\\ y_k=0,\\ x_k=1}}n_{ki}\left(d_{ki},\mu_{ki}\right)\delta_{\mu_{ki},q}\times\tag{B3}$$

$$\times\prod_{p\leq q-1}\prod_{\substack{k\in\partial i:\\ \mu_{ki}\leq q-1}}e^{-\beta c_{i}^{p}\mathbb{I}[\boldsymbol{\mu}_{i}\neq\boldsymbol{p}]}\psi_{i}^{p}\left(\boldsymbol{d}_{i},\boldsymbol{\mu}_{i}\right)\mathbb{I}\left[y_{k}=1-\delta_{d_{ki},0}\right]n_{ki}\left(d_{ki},\mu_{ki}\right)\tag{B4}$$

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

$$Z_{\boldsymbol{x}}^{q}=\sum_{\boldsymbol{y}\leq\boldsymbol{x}}\left\{\sum_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \mu_{ki}\leq q}}e^{-\beta c_{i}^{q}\mathbb{I}[\boldsymbol{\mu}_{i}\neq\boldsymbol{q}]}\psi_{i}^{q}\left(\boldsymbol{d},\boldsymbol{\mu}\right)\prod_{\substack{k\in\partial i\\ y_k=0\\ x_k=1}}n_{ki}\left(d_{ki},\mu_{ki}\right)\delta_{\mu_{ki},q}\times\right.\tag{B5}$$

$$\left.\times\sum_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i\\ \mu_{ki}\leq q-1}}\prod_{p\leq q-1}e^{-\beta c_{i}^{p}\mathbb{I}[\boldsymbol{\mu}_{i}\neq\boldsymbol{p}]}\psi_{i}^{p}\left(\boldsymbol{d},\boldsymbol{\mu}\right)\prod_{k\in\partial i}\mathbb{I}\left[y_{k}=1-\delta_{d_{ki},0}\right]n_{ki}\left(d_{ki},\mu_{ki}\right)\right\}\tag{B6}$$

$$=\sum_{\boldsymbol{y}\leq\boldsymbol{x}}\left(g_{\boldsymbol{y}}^{0}+g_{\boldsymbol{y}}^{b}+g_{\boldsymbol{y}}^{f}\right)Z_{\boldsymbol{y}}^{q-1}\tag{B7}$$

where

$$g_{\boldsymbol{y}}^{0} = e^{-\beta c_i^q} \prod_{\substack{k\in\partial i \\ y_k=0 \\ x_k=1}} n_{ki}\left(0,0\right)$$

$$g_{\boldsymbol{y}}^{b} = \sum_{d>0} \sum_{\substack{j\in\partial i \\ y_j=0 \\ x_j=1}} n_{ji}\left(-d,q\right) \prod_{\substack{k\in\partial i\setminus j \\ y_k=0 \\ x_k=1}} \left[n_{ki}\left(d+1,q\right) + n_{ki}\left(0,0\right)\right]$$

$$g_{\boldsymbol{y}}^{f} = \delta_{c_i^q,0} \sum_{d>0} \sum_{\substack{j\in\partial i \\ y_j=0 \\ x_j=1}} n_{ji}\left(-d,q\right) \sum_{\substack{k\in\partial i\setminus j \\ y_k=0 \\ x_k=1}} n_{ki}\left(d,q\right) \prod_{\substack{l\in\partial i\setminus\{j,k\} \\ y_l=0 \\ x_l=1}} n_{li}\left(0,0\right)$$

In the special case in which no communications is flowing within the graph, that is for $q=0$, we must impose the value of $Z_{\mathbf{x}}^{0}$ through

$$Z_{\mathbf{x}}^{0} = e^{-\beta\sum_{\mu} c_i^{\mu}} \mathbb{I}\left[\mathbf{x}=\mathbf{0}\right] \prod_{j\in\partial i} n_{ji}\left(0,0\right) \tag{B8}$$