

# Determining Minimum Energy Configurations for Constrained Membrane Problems using Two-Dimensional Message-Passing Algorithms

Avinash Vem, Krishna R. Narayanan and Arun R. Srinivasa  
Texas A&M University  
College Station, TX 77843

## Abstract

In this work, we present a technique to find approximate minimum energy configurations of a 2-dimensional membrane using a local message-passing algorithm on the graphical structure equivalent of the membrane. This approach neither requires the energy function to be convex nor the computation of gradients. However, the disadvantage with this approach is it does not always guarantee convergence to a fixed point or a local minima solution. We are currently investigating the dependence of the convergence properties of this algorithm on the type of energy function employed.

## I. INTRODUCTION

The objective of this work is to find minimum energy configurations of certain types of elastic membranes. The main interest is in the case when the energy function is non-convex and when the derivative of the function with respect to the location of the particles is difficult to compute. We propose the use of locally-effective message-passing algorithms to find minimum energy configurations. Message-passing algorithms have been shown to be very effective for solving problems in a variety of areas including coding and signal processing [1] and artificial intelligence [2]. Doraiswamy, Narayanan and Srinivasa used an instance of a message-passing algorithm, known as the Viterbi algorithm to find minimum energy configurations for 1-dimensional (1-D) cantilever beams [3].

In this paper, we extend this work and consider the use of message passing algorithms for finding minimum energy configurations for the 2-dimensional elastic membranes. Unlike in its 1-D counterpart, there are no optimal minimization algorithms with a complexity that scales linearly (or, even polynomially) with the number of variables in the problem. This makes the extension from the 1-D case to the 2-D case to be non-trivial.

In this work, the membrane is modeled as a set of nodes which can take one of finitely many locations (more abstractly, can be thought of as values). The energy function to be minimized is a function of the node (particle) locations. Our approach is to first consider a graphical model for the membrane where node locations can be treated as values taking by variable nodes and the energy arising due to the interactions between nodes are modeled by placing edges between the nodes or, equivalently through a function node. Such a graphical representation enables us to use message passing algorithm and find the configurations that minimize the energy.

It is well known that for graphs of finite sizes, when the underlying graph does not have cycles, locally-optimal iterative algorithms such as message passing algorithms are very effective and are guaranteed to converge. Pearl [2] indeed showed that the messages at convergence are guaranteed to give optimal assignment values. For the case of graphs with cycles, for general energy functions, it cannot be guaranteed that the algorithm converges to a fixed point. Nevertheless, several empirical studies have demonstrated that excellent experimental results are obtained by running the message-algorithm on graphs with cycles and some convergence results on graphs with cycles are also available [4]. Particularly in coding and signal processing, such message-passing algorithms have been very successful in 1-D problems [1]. Even though 2-D message passing algorithms have not been studied in as much detail as their 1-D counterparts, a few works in [5], [6] have shown that message passing algorithms can work well for two-dimensional signal processing problems also.

Motivated by the success of the message-passing algorithm for these problems, we consider the application of these class of iterative algorithms on the approximated graphical model and show that it reaches local minima for a certain energy function.

The rest of the report is organized as follows. In section II, we define a generic minimization problem on any undirected graph and give the message propagation rules for a generic cost function. In section III, we give the graphical equivalent of the energy minimization problem of a constrained membrane and present the message passing rules. Finally, in section IV, we present some simulation results.

## II. MESSAGE-PASSING FRAMEWORK

The general graphical framework considered for the minimization problem can be described as follows. Let the set of all vertices in a graph be defined as  $\mathcal{V}$ , the total energy function on the graph that is to be minimized as  $E$  and the discrete set of values that are valid for any node as  $\mathcal{S}$ .  $E$  is usually a function of the relative positions (or some other property of each

node) of all the pairs of connected nodes in the graph. We denote the mapping that assigns to each node a value from  $\mathcal{S}$  as  $f : \mathcal{V} \rightarrow \mathcal{S}$ . For ease of notation we use the notations  $f(n)$  and  $f_n$  interchangeably. Let  $\mathcal{N}$  denote the set of edges  $\{(p, q)\}$ , where  $p, q \in \mathcal{V}$  are connected. For a given mapping, the total energy or also referred to as the objective function is given by:

$$E(f) = \sum_{(p,q) \in \mathcal{N}} C(f_p, f_q) \quad (1)$$

where  $C(f_p, f_q)$  is the cost associated with the edge  $(p, q) \in \mathcal{N}$  for the given mapping  $f$ . The function  $C(f_p, f_q)$  is also referred to as the cost function. Our aim is to find a mapping that minimizes the objective function  $E$ .

#### A. Message-Passing Rules

The message passing algorithm considered here works by passing messages between nodes in the graph in an iterative fashion. Each message is a vector of dimension  $|\mathcal{S}|$ . Let  $m_{pq}^t$  be the message that node  $p$  sends to a neighboring node  $q$  at time (or, iteration)  $t$ . All messages at time 0,  $m_{pq}^0$  are initialized to zero vectors and new messages at each iteration are computed as follows:

$$m_{pq}^t(f_q) = \min_{f_p} \left( C(f_p, f_q) + \sum_{s \in \mathcal{N}(p) \setminus q} m_{sp}^{t-1}(f_p) \right) \quad (2)$$

where neighborhood of  $p$  is denoted by  $\mathcal{N}(p)$  and is given by  $\{s : (p, s) \in \mathcal{N}\}$  and  $\mathcal{N}(p) \setminus q$  denotes the neighborhood of  $p$  except  $q$ . The messages passed can be roughly thought of as follows: Consider the partial graph with just the nodes  $\{q, p\}$  and also the nodes that are atmost  $t - 1$  levels away from the node  $p$ , then assuming the graph is cycle free,  $m_{pq}^t(f_q)$  is the optimal energy over the partial graph when minimized over all the possible values of  $p$  given the position of the node  $q$  is fixed as  $f_q$ . Messages passed between the nodes according to (2) are demonstrated in Fig.1. After  $M$  iterations, the energy vector is computed for each node according to,

$$e_q(f_q) = \sum_{p \in \mathcal{N}(q)} m_{pq}^M(f_q). \quad (3)$$

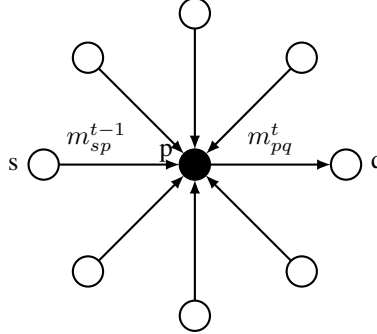


Fig. 1: Message from node  $p$  to node  $q$ ,  $m_{pq}^t$ , is a function of messages  $m_{sp}^{t-1}$  from all other neighbors to  $p$

After a sufficient number of iterations, the mapping  $f^*$  that minimizes  $e_q(f_q)$  individually at each node  $q$  is considered the output mapping i.e  $f^*(q) = \arg \min_{f_q} e_q(f_q)$ . There are several heuristics for determining when the iterations have converged. In this report, we simply iterate for a fixed number of iterations.

### III. APPLICATION OF MESSAGE-PASSING RULES TO 2D MEMBRANE

To use the above described message passing framework on a 2-dimensional membrane for minimizing certain energy function, we need an equivalent discrete representation of the membrane. We consider a simple configuration of a square membrane in shear. The membrane is of side  $L$  units and is attached to rigid blocks along the top and bottom edges and is free along the sides. A relative displacement of  $\delta$  units is imposed between the two blocks without changing their distance thus imposing a geometric shear strain of  $\frac{\delta}{L}\%$ .

Let us consider a grid equivalent of the membrane where the membrane is represented by a grid of  $N^2$  points. These  $N^2$  nodes on the grid form the set  $\mathcal{V}$ . In the limit of large  $N$ , grid behavior approaches the behavior of the membrane.

Any point on the grid can now be represented as  $p_{i,j}$  where  $0 \leq i, j < N$ . A particle on the membrane usually interacts with only its neighboring points and any energy function is a function of all such interactions between the particles. Therefore the energy function considered here depends only on the relative location of a node with respect to its four adjacent neighbors. Hence, the neighborhood of any point is the set of four adjacent nodes given by

$$\mathcal{N}(p_{i,j}) = \{p_{i,j-1}, p_{i,j+1}, p_{i-1,j}, p_{i+1,j}\} \quad (4)$$

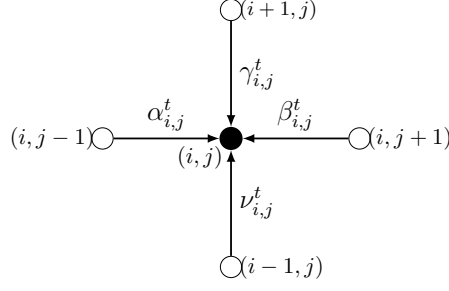


Fig. 2: Four types of incoming messages of iteration  $t$ .

#### A. Energy Function

Before looking at message passing rules, we still need to define  $\mathcal{S}$ . As we have already discussed, any energy in the membrane stems from the interactions between neighboring particles. For the ease of computations we restrict valid locations for any node  $p_{i,j}$  on the grid to a square centered around the node's base position  $(x_{ij}, y_{ij})$ .  $\{(x_{ij}, y_{ij}), 0 \leq i, j < N\}$  forms the input to the algorithm and can be thought of as an initial guess for the positions of the particles. The accuracy of the solution depends on the size and resolution of  $\mathcal{S}$ . Then, the range of any mapping  $f$  is

$$\mathcal{S} = \{-M\Delta, (-M+1)\Delta, \dots, 0, \dots, M\Delta\}^2. \quad (5)$$

To elaborate, given a mapping  $f$ , coordinates of any point  $p_{i,j}$  are given by

$$\tilde{f}(p_{ij}) = (x_{ij}, y_{ij}) + f(p_{ij}).$$

We consider the energy function,

$$\begin{aligned} E(f) = & \sum_i \sum_j C(\tilde{f}(p_{i,j}), \tilde{f}(p_{i,j-1})) + C(\tilde{f}(p_{i,j}), \tilde{f}(p_{i,j+1})) \\ & + C(\tilde{f}(p_{i,j}), \tilde{f}(p_{i-1,j})) + C(f(p_{i,j}), f(p_{i+1,j})) \end{aligned} \quad (6)$$

where

$$\begin{aligned} C(\mathbf{p}_1, \mathbf{p}_2) = & \max\{0, \|\mathbf{p}_1 - \mathbf{p}_2\|^2 - 1\} \quad \text{or} \\ & = [\|\mathbf{p}_1 - \mathbf{p}_2\|^2 - 1]^+. \end{aligned} \quad (7)$$

#### B. Boundary Constraints

The two boundaries of the grid are fixed and their co-ordinates are given by

$$\begin{aligned} (x_{0,j}, y_{0,j}) &= (j, 0) \\ (x_{N-1,j}, y_{N-1,j}) &= (j + \delta, N - 1) \quad 0 \leq j < N. \end{aligned}$$

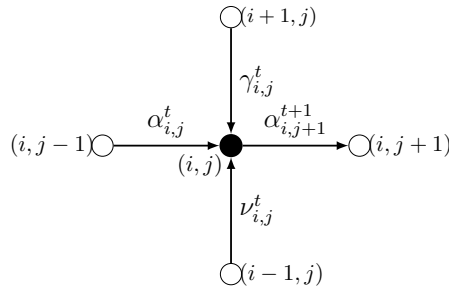


Fig. 3: Outgoing message  $\alpha_{i,j+1}^{t+1}$  as a function of the incoming messages of iteration  $t$  at node  $p_{i,j}$ .

The initial configuration for the nodes other than the constrained boundaries is given by

$$\begin{aligned} x_{i,j}^0 &= \frac{\delta}{N}i + j + \eta_{i,j}^x \\ y_{i,j}^0 &= j + \eta_{i,j}^y; i = 1, 2, \dots, N-2, \forall j. \end{aligned} \quad (8)$$

where  $\eta_{i,j}^x, \eta_{i,j}^y \sim \mathcal{N}(0, \sigma^2)$  are a pair of independent Gaussian random variables with zero mean and variance  $\sigma^2$ . These noise samples help us in introducing the randomness in the initial guesses for the coordinates of the nodes. Further we model these noise samples as independent samples for any two nodes.

### C. Message Passing Rules

For any point  $p_{i,j}$ , neighborhood defined in (4) consists of four points. Hence we can classify the messages in this algorithm into four types and for convenience, we use the following notation

$$\begin{aligned} \alpha_{i,j} &:= \text{Message from } p_{i,j-1} \Rightarrow p_{i,j} \quad (\text{West to East,}) \\ \beta_{i,j} &:= \text{Message from } p_{i,j+1} \Rightarrow p_{i,j} \quad (\text{East to West,}) \\ \gamma_{i,j} &:= \text{Message from } p_{i+1,j} \Rightarrow p_{i,j} \quad (\text{North to South,}) \\ \nu_{i,j} &:= \text{Message from } p_{i-1,j} \Rightarrow p_{i,j} \quad (\text{South to North,}) \end{aligned} \quad (9)$$

Messages defined in (9) are depicted in Fig. 2.

With the above definitions, message passing rules follow directly from (12). We give the update rules as follows:

$$\begin{aligned} \alpha_{i,j+1}^{t+1}(f(p_{i,j+1})) &= \min_{f(p_{i,j})} C(f(p_{i,j}), f(p_{i,j+1})) + \alpha_{i,j}^t(f(p_{i,j})) + \gamma_{i,j}^t(f(p_{i,j})) + \nu_{i,j}^t(f(p_{i,j})), \\ \beta_{i,j-1}^{t+1}(f(p_{i,j-1})) &= \min_{f(p_{i,j})} C(f(p_{i,j-1}), f(p_{i,j-1})) + \beta_{i,j}^t(f(p_{i,j})) + \gamma_{i,j}^t(f(p_{i,j})) + \nu_{i,j}^t(f(p_{i,j})), \\ \nu_{i+1,j}^{t+1}(f(p_{i+1,j})) &= \min_{f(p_{i,j})} C(f(p_{i,j}), f(p_{i+1,j})) + \nu_{i,j}^t(f(p_{i,j})) + \alpha_{i,j}^t(f(p_{i,j})) + \beta_{i,j}^t(f(p_{i,j})) \\ \gamma_{i-1,j}^{t+1}(f(p_{i-1,j})) &= \min_{f(p_{i,j})} C(f(p_{i,j}), f(p_{i-1,j})) + \gamma_{i,j}^t(f(p_{i,j})) + \alpha_{i,j}^t(f(p_{i,j})) + \beta_{i,j}^t(f(p_{i,j})). \end{aligned} \quad (10)$$

Refer to Fig. 3 for an illustration of the message sent by node  $p_{i,j}$  as a function of the messages received in previous iteration.

## IV. SIMULATION RESULTS

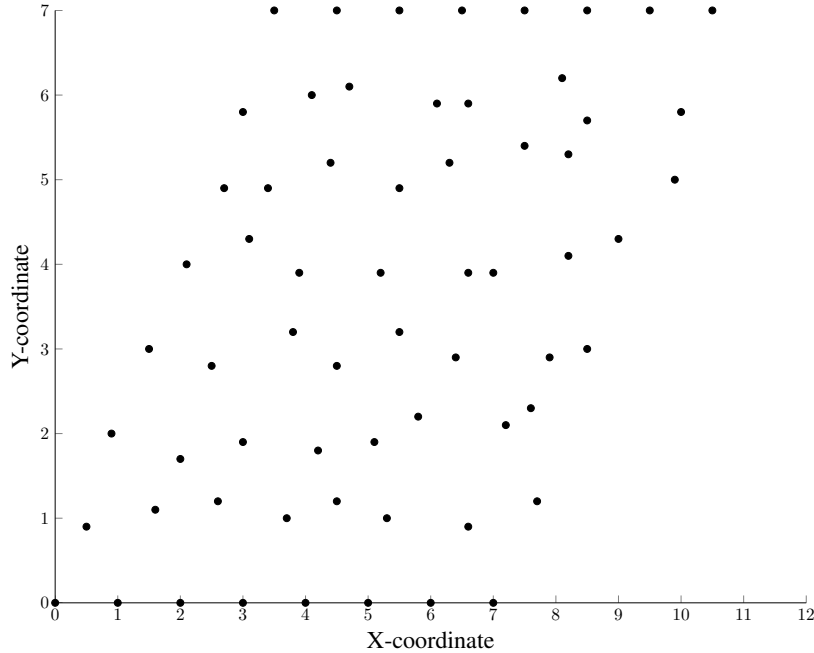


Fig. 4: Initial Configuration of the membrane for  $N=8$ ,  $\delta=3.5$

A random initial configuration for the grid is generated using (8), and is shown in Fig. 4. Applying message passing rules as described in (12) and with 20 iterations of the algorithm, resulted in the final configuration, shown in Fig. 5. Evolution of the value of the objective function with each iteration of message passing algorithm is shown in Fig. 6.

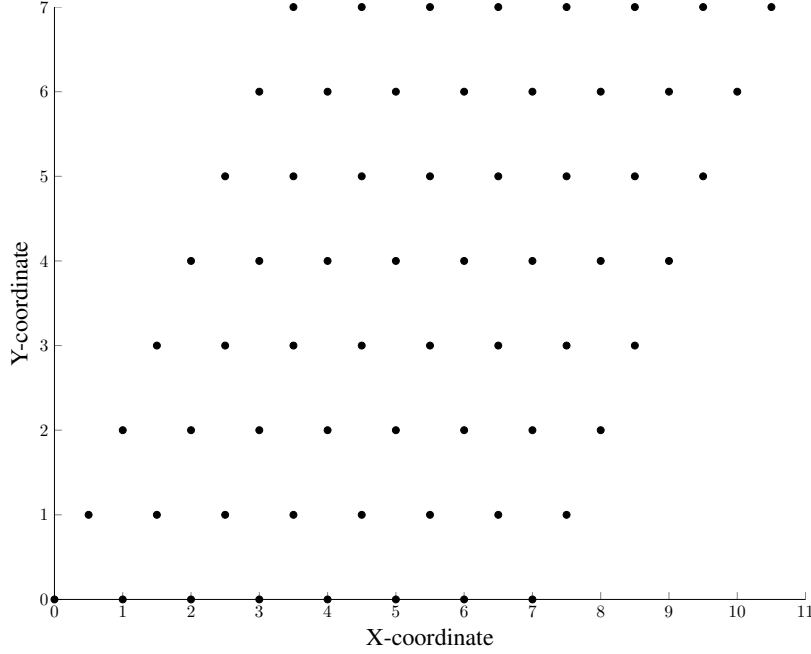


Fig. 5: Final Configuration after 20 iterations of message passing

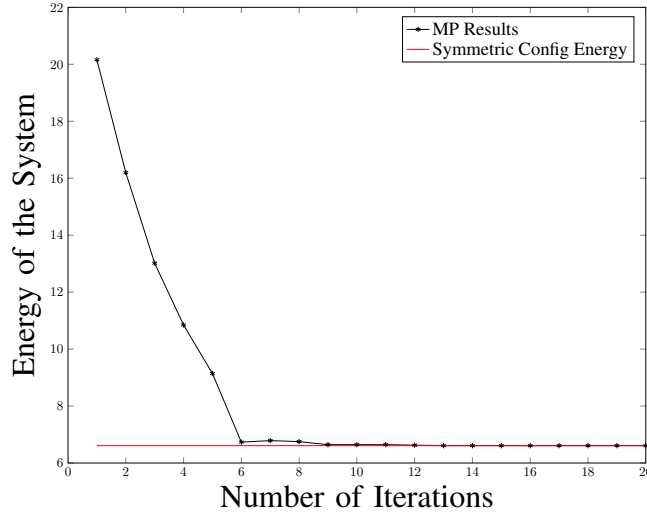


Fig. 6: Energy versus Iterations using message passing algorithm for the 2D-Problem

## V. AN EVEN CONVEX ENERGY FUNCTION

In this section we look at the energy function  $E$  as defined previously in (1) although with a different cost function  $C$  defined as

$$C(\mathbf{p}, \mathbf{q}) = (|p_z - q_z|^2) (|p_z - q_z|^2 - 1) \quad (11)$$

where  $p_z$  and  $q_z$  are the  $z$ -coordinates of the respective vectors. The cost function simplifies the problem by considering just the  $z$ -coordinates in the cost function thus enabling us to study the shape of the membrane when a force is applied. The cost function is plotted in Fig. 7. Observe that the above cost function has two global minima at  $|p_z - q_z| = \pm \frac{1}{\sqrt{2}}$ .

It's important to note here that sum-product message passing is an iterative algorithm that solves the problem of minimizing a function exactly on a factor graph which is equivalent to solving the optimization of a function on cycle free graphs. But the

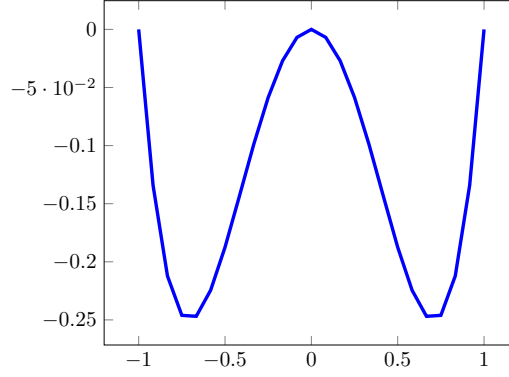


Fig. 7: Cost function vs  $|p_z - q_z|$

graphs we consider in this work i.e., 2D grids, are highly cyclic and hence is not guaranteed to work. For the cost function described in (11), simulations showed that the message passing algorithm described above doesn't work, see Fig. 9.

For this purpose we employ a message passing algorithm that uses a slightly different scheduling. First, we will describe the modified message passing algorithm used, followed by the simulation results and then the intuition behind the scheduling employed in the algorithm.

#### A. Optimal BCJR algorithm

Under the current setting, this problem is equivalent to finding a ML solution on a trellis code. Here's how:

The positions of all nodes in Rows 1 and N have been fixed. With that in mind, we will define each row as one Trellis Node making it a total of N trellis nodes and we will refer to our original nodes as normal nodes. Assuming that each normal node can have a range of values  $\mathcal{S}$  as defined previously, each normal node can take any of the  $|\mathcal{S}|$  values and hence each trellis node can take any of the  $|\mathcal{S}|^N$  values from the set  $\mathcal{S}^N = \mathcal{S} \times \mathcal{S} \cdots \times \mathcal{S}$ . Any mapping  $f$  assigns to each trellis node a valid state from the set  $\mathcal{S}^N$ . So hypothetically one can run a Viterbi algorithm keeping in mind the states of the initial and final trellis which are known (boundaries fixed) to us, which ideally should output the optimal solution. Note that here every set of states from the state space  $\mathcal{S}^{N \times N}$  is an admissible solution and hence the bit-wise optimal (Trellis node-wise optimal in this case) BCJR algorithm or the Forward-Backward algorithm is equivalent to the optimal Viterbi algorithm. We will introduce some notation that makes it convenient for us to describe the equivalent BCJR algorithm. The trellis-node is denoted as the vector  $\mathbf{p}_i := \{p_{n,j}, 1 \leq j \leq N\}$ . Along similar lines the cost associated with a trellis-node  $\mathbf{p}_i$  is denoted as

$$C(f(\mathbf{p}_i)) := \sum_{j < N} C(f(p_{i,j}, f(p_{i,j+1})))$$

and the cost between two trellis-nodes  $\mathbf{p}_i, \mathbf{p}_{i+1}$  as

$$C(f(\mathbf{p}_i \leftrightarrow \mathbf{p}_{i+1})) := \sum_j C(f(p_{i,j}, p_{i+1,j})).$$

With the above definitions, if one were to implement the BCJR algorithm on the equivalent trellis graph described above, the forward and backward messages can be given as follows:

$$\begin{aligned} E_f(f(\mathbf{p}_{n-1})) &= \min_{f(\mathbf{p}_{n-2})} [E_f(f(\mathbf{p}_{n-2})) + C(f(\mathbf{p}_{n-2} \leftrightarrow \mathbf{p}_{n-1})) + C(f(\mathbf{p}_{n-1}))] \\ E_b(f(\mathbf{p}_{n+1})) &= \min_{f(\mathbf{p}_{n+2})} [E_b(f(\mathbf{p}_{n+2})) + C(f(\mathbf{p}_{n+2} \leftrightarrow \mathbf{p}_{n+1})) + C(f(\mathbf{p}_{n+1}))] \end{aligned}$$

where the subscripts 'f' and 'b' in  $E_f$  and  $E_b$  denote the forward and backward respectively. These two quantities  $E_f$  and  $E_b$  are the equivalents of  $\alpha$  and  $\beta$  messages that we deal with in BCJR or a Forward-Backward algorithm respectively. Once we have all the forward and backward messages the optimal assignment for the  $n^{\text{th}}$  trellis-node can be computed as follows:

$$f^*(\mathbf{p}_n) = \arg \min_{f(\mathbf{p}_n) \in \mathcal{S}^N} [E_f(f(\mathbf{p}_{n-1})) + C(f(\mathbf{p}_{n-1} \leftrightarrow \mathbf{p}_n)) + C(f(\mathbf{p}_n)) + C(f(\mathbf{p}_n \leftrightarrow \mathbf{p}_{n+1})) + E_b(f(\mathbf{p}_{n+1}))].$$

This is how a trellis-node wise optimal algorithm for this problem would look like, but the problem here again is having to compute and store  $E_f$  and  $E_b$  which are vectors of size  $|\mathcal{S}|^N$ .

We employ message-passing algorithm with scheduling which closely imitates BCJR algorithm but works with vectors of size  $|\mathcal{S}|$  rather than  $|\mathcal{S}|^N$ . We first describe the rules then we will explain how it is BCJR-like algorithm.

### B. BCJR-like Message Passing with Scheduling

The update rules for the messages in the vertical direction i.e the  $\nu, \gamma$  messages are the same as in the normal message passing algorithm (12) whereas the messages in the horizontal direction i.e., the  $\alpha, \beta$  messages follow a slightly different schedule:

$$\begin{aligned}\alpha_{i,j+1}^{t+1}(f(p_{i,j+1})) &= \min_{f(p_{i,j})} [C(f(p_{i,j}), f(p_{i,j+1})) + \alpha_{i,j}^{t+1}(f(p_{i,j})) + \gamma_{i,j}^t(f(p_{i,j})) + \nu_{i,j}^t(f(p_{i,j}))] \text{ for } 1 \leq j < N, \\ \beta_{i,j-1}^{t+1}(f(p_{i,j-1})) &= \min_{f(p_{i,j})} [C(f(p_{i,j-1}), f(p_{i,j-1})) + \beta_{i,j}^{t+1}(f(p_{i,j})) + \gamma_{i,j}^t(f(p_{i,j})) + \nu_{i,j}^t(f(p_{i,j}))] \text{ for } 1 < j \leq N. \\ \nu_{i+1,j}^{t+1}(f(p_{i+1,j})) &= \min_{f(p_{i,j})} [C(f(p_{i,j}), f(p_{i+1,j})) + \nu_{i,j}^t(f(p_{i,j})) + \alpha_{i,j}^t(f(p_{i,j})) + \beta_{i,j}^t(f(p_{i,j}))] \\ \gamma_{i-1,j}^{t+1}(f(p_{i-1,j})) &= \min_{f(p_{i,j})} [C(f(p_{i,j}), f(p_{i-1,j})) + \gamma_{i,j}^t(f(p_{i,j})) + \alpha_{i,j}^t(f(p_{i,j})) + \beta_{i,j}^t(f(p_{i,j}))]\end{aligned}\quad (12)$$

To understand the above scheduling rules let us consider a simple grid of size  $3 \times 4$  and analyze the messages received in iteration 2 at  $p_{2,3}$ . Recall that the initialization messages i.e., the messages sent in iteration 0 are all zero vectors. If we analyze  $\gamma_{2,3}^2$ , it can be shown that:

$$\begin{aligned}\gamma_{2,3}^2(f(p_{2,3})) &= \min_{f(p_{3,3})} [\alpha_{3,3}^1(f(p_{3,3})) + C(f(p_{2,3}), f(p_{3,3})) + \beta_{3,3}^1(f(p_{3,3}))] \\ &= \min_{f(p_{3,j}), j \neq 1} \left[ \alpha_{3,2}^1(f(p_{3,2})) + \sum_{f(p_{3,j}), j > 1} C(f(p_{3,j}), f(p_{3,j+1})) \right] \\ &= \min_{f(\mathbf{p}_3)} [C(f(\mathbf{p}_3)) + C(f(p_{2,3}), f(p_{3,3}))],\end{aligned}\quad (13)$$

$\gamma_{2,3}^2(f(p_{2,3}))$  is the value of the energy in the trellis-node  $\mathbf{p}_3$  and the edge  $p_{2,3} \leftrightarrow p_{3,3}$  (all the edges shown with solid lines in top half of Fig. 8) for a given  $f(p_{2,3})$  and when minimized over all possible set of states for the trellis-node  $\mathbf{p}_3$ . Similarly it can be shown that  $\nu_{2,3}^2(f(p_{2,3}))$  is the value of the energy in the trellis-node  $\mathbf{p}_1$  and the edge  $p_{2,3} \leftrightarrow p_{1,3}$  (all the edges shown with solid lines in bottom half of Fig. 8) for a given  $f(p_{2,3})$  and when minimized over all possible set of states for the trellis-node  $\mathbf{p}_1$ .

$$\nu_{2,3}^2(f(p_{2,3})) = \min_{f(\mathbf{p}_1)} [C(f(\mathbf{p}_1)) + C(f(p_{2,3}), f(p_{1,3}))]. \quad (14)$$

If we consider the messages  $\alpha_{2,3}^2$  and  $\beta_{2,3}^2$ , similar to (13) and (14), we can deduce that  $\alpha_{2,3}^2(f(p_{2,3})) + \beta_{2,3}^2(f(p_{2,3}))$  is the energy contained in the edges shown as the dashed lines in Fig. 8 given  $f(p_{2,3})$ . We can view  $\gamma_{2,j}$  roughly as equivalent

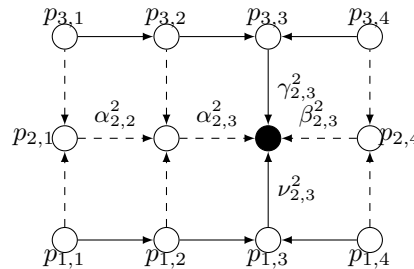


Fig. 8: Messages received by  $p_{2,3}$  in Iteration 2:  $\gamma_{2,3}^2$  (or  $\nu_{2,3}^2$ ) is the minimum energy contained in the edges shown in solid lines in top(bottom) half when optimized over all nodes in Row 3(Row 1). See Eqns. (13) and (14).

to the backward message  $E_b(\mathbf{p}_3)$ ,  $\nu_{2,j}$  roughly as equivalent to the forward message  $E_f(\mathbf{p}_1)$  and  $\alpha_{2,j} + \beta_{2,j}$  as the message vectors that optimize over the nodes in Row 2 and the transitions  $\mathbf{p}_2 \leftrightarrow \mathbf{p}_1$  and  $\mathbf{p}_2 \leftrightarrow \mathbf{p}_3$  (except the edges  $(p_{2,3}, p_{1,3})$  and  $(p_{2,3}, p_{3,3})$ ). Although the optimization done by the messages  $\gamma_{2,j}^2$  and  $\alpha_{2,j}^2 + \beta_{2,j}^2$  are independent they optimize over the same set of nodes  $p_{3,k}, k \neq j$  independently therefore leading to cycles. This leads to our belief(conjecture) that this scheduling leads to lesser number of short cycles thus improving the performance of the message passing algorithm.

### C. Correction Factor

Let's assume the same grid of size  $3 \times 4$  as in the example above. Consider  $\nu_{3,3}^3$ :

$$\begin{aligned} \nu_{3,3}^4(f(p_{3,3})) &= \min_{f(p_{2,3})} [\nu_{2,3}^3(f(p_{2,3})) + \alpha_{2,3}^3(f(p_{2,3})) + \beta_{2,3}^3(f(p_{2,3})) + C(f(p_{2,3}), f(p_{3,3}))] \\ &= \min_{f(\mathbf{p})} \left[ \nu_{2,3}^2(f(p_{2,3})) + \sum_{j \neq 3, 1 \leq j \leq 4} [\nu_{2,j}^2(f(p_{2,j})) + \gamma_{2,j}^2(f(p_{2,j})) + C(f(p_{2,j}, p_{2,j+1}))] + C(f(p_{2,3}), f(p_{3,3})) \right] \end{aligned}$$

Notice that in the term  $\sum_{j \neq 3, 1 \leq j \leq M} [\nu_{2,j}^2(f(p_{2,j})) + \gamma_{2,j}^2(f(p_{2,j}))]$ , the terms of the form  $\nu_{2,j}^2$  appear for  $1 \leq j \leq 4$  although each term computes approximately the same quantity  $\min_{f(\mathbf{p}_1)} [C(f(\mathbf{p}_1)) + C(f(p_{2,j}, f(p_{1,j})))]$  which leads to the addition of the term  $C(f(\mathbf{p}_1))$  4 times. Thus we introduce the correction factor  $\frac{1}{M}$  where the grid is of size  $N \times M$  in the computation of the messages  $\gamma$  and  $\nu$ :

$$\begin{aligned} \nu_{i+1,j}^{t+1}(f(p_{i+1,j})) &= \min_{f(p_{i,j})} \left[ C(f(p_{i,j}), f(p_{i+1,j})) + \frac{1}{M} (\nu_{i,j}^t(f(p_{i,j})) + \alpha_{i,j}^t(f(p_{i,j})) + \beta_{i,j}^t(f(p_{i,j}))) \right] \\ \gamma_{i-1,j}^{t+1}(f(p_{i-1,j})) &= \min_{f(p_{i,j})} \left[ C(f(p_{i,j}), f(p_{i-1,j})) + \frac{1}{M} (\gamma_{i,j}^t(f(p_{i,j})) + \alpha_{i,j}^t(f(p_{i,j})) + \beta_{i,j}^t(f(p_{i,j}))) \right]. \end{aligned} \quad (15)$$

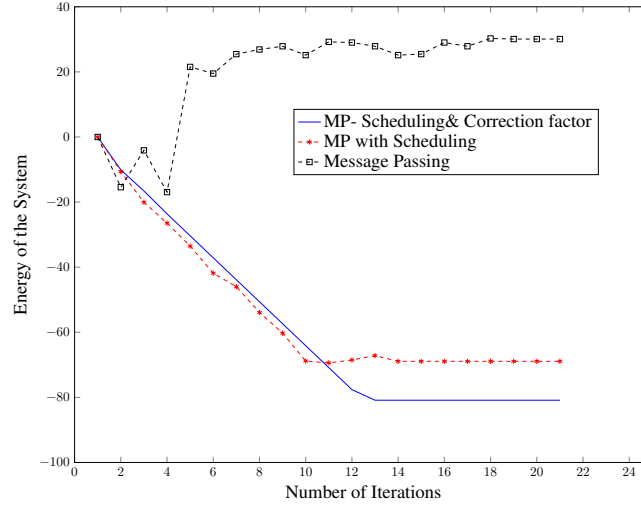


Fig. 9: Comparison of the performance of message passing versus the effects of scheduling and the correction factor included.

### D. Simulation Results

For a gridsize of  $14 \times 14$ , a  $z$ -shear of magnitude 0.3 is induced along one boundary and the other boundary is at a  $z$ -coordinate of 0. The 3D plot of the initial configuration of the membrane is shown in Fig. 10. We will keep both the boundaries fixed throughout the message passing algorithm. The results for total energy output versus the number of iterations for the message passing schemes described are presented in Fig. 9. Observe that the message passing algorithm without the scheduling as described in Sec V-B has a very bad performance and that the scheme with the scheduling and the correction factor included as described in (15) improves the performance significantly.

### REFERENCES

- [1] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.
- [2] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- [3] S. Doraiswamy, K. R. Narayanan, and A. R. Srinivasa, "Finding minimum energy configurations for constrained beam buckling problems using the viterbi algorithm," *International Journal of Solids and Structures*, vol. 49, no. 2, pp. 289–297, 2012.
- [4] 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*, vol. 47, no. 2, pp. 736–744, 2001.
- [5] M. Marrow and J. K. Wolf, "Iterative detection of 2-dimensional isi channels," in *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*. IEEE, 2003, pp. 131–134.
- [6] N. Singla, J. A. O'Sullivan, R. S. Indeck, and Y. Wu, "Iterative decoding and equalization for 2-d recording channels," *Magnetics, IEEE Transactions on*, vol. 38, no. 5, pp. 2328–2330, 2002.



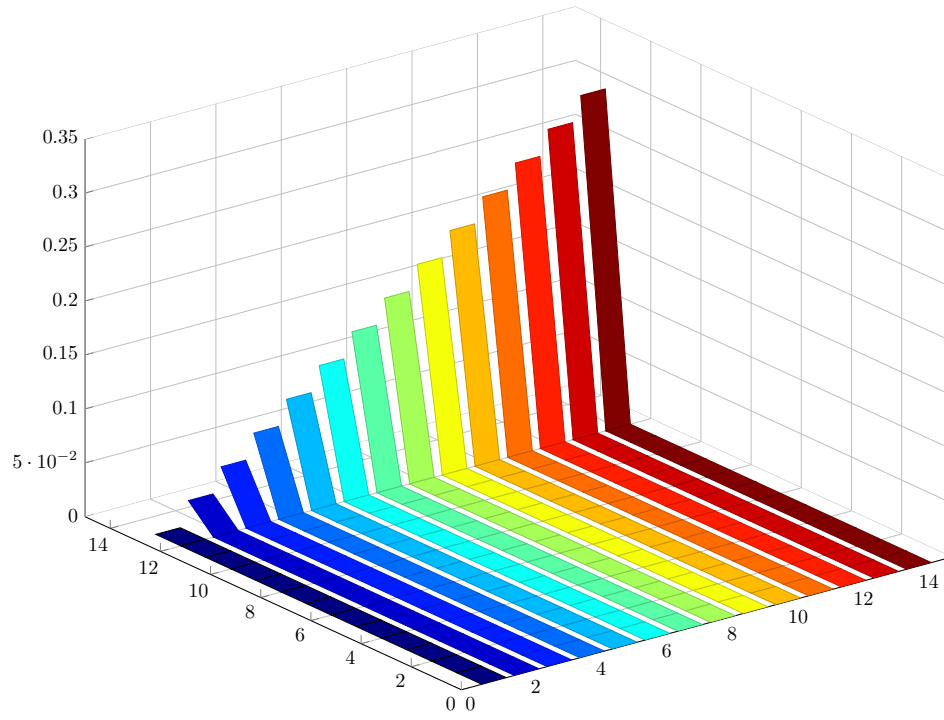


Fig. 10: Initial Config of the membrane for a  $z\text{-shear}=0.3$ .

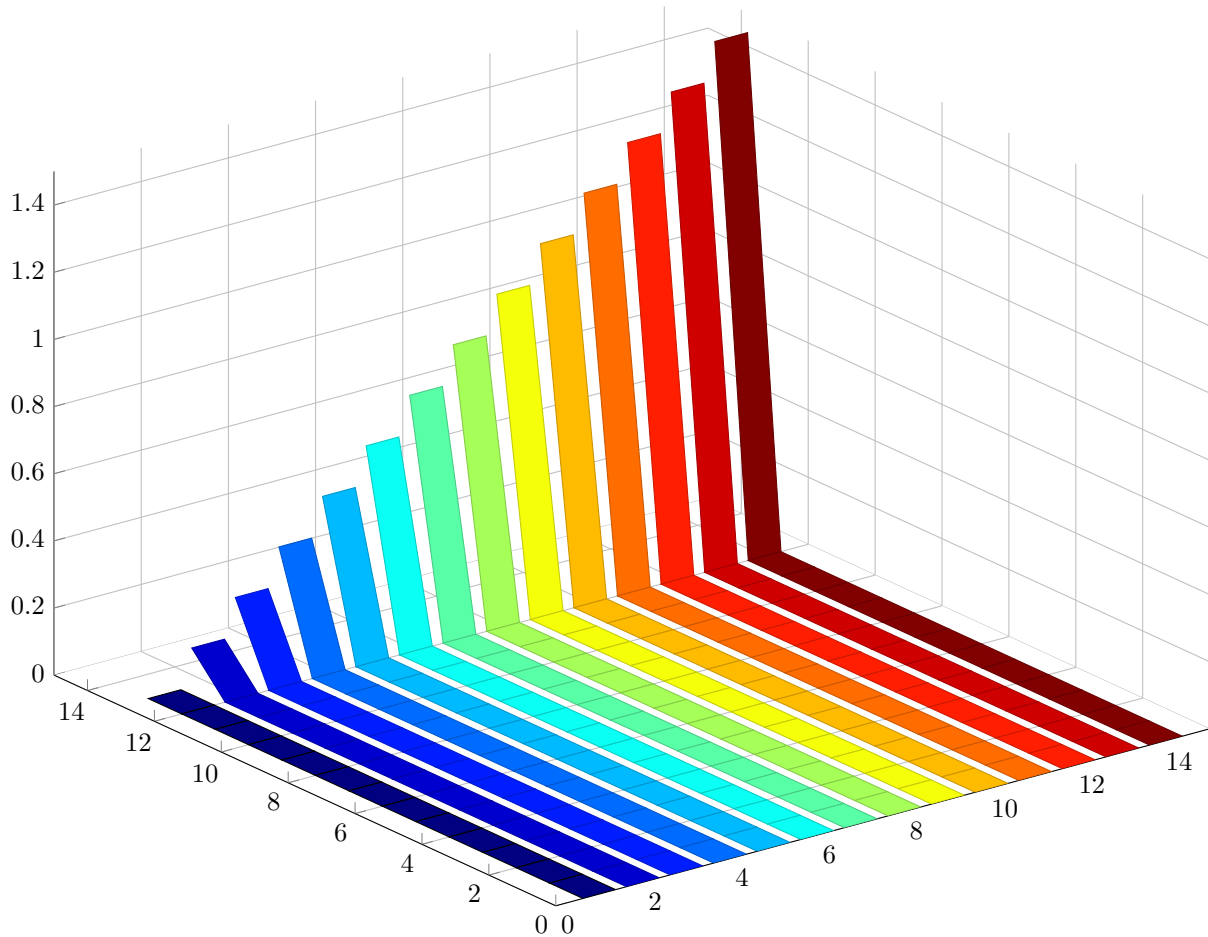


Fig. 11: Final Configuration after 20 iterations of message passing with scheduling and correction factor of  $1/N$ ,  $z\text{-shear}=1.5$ .

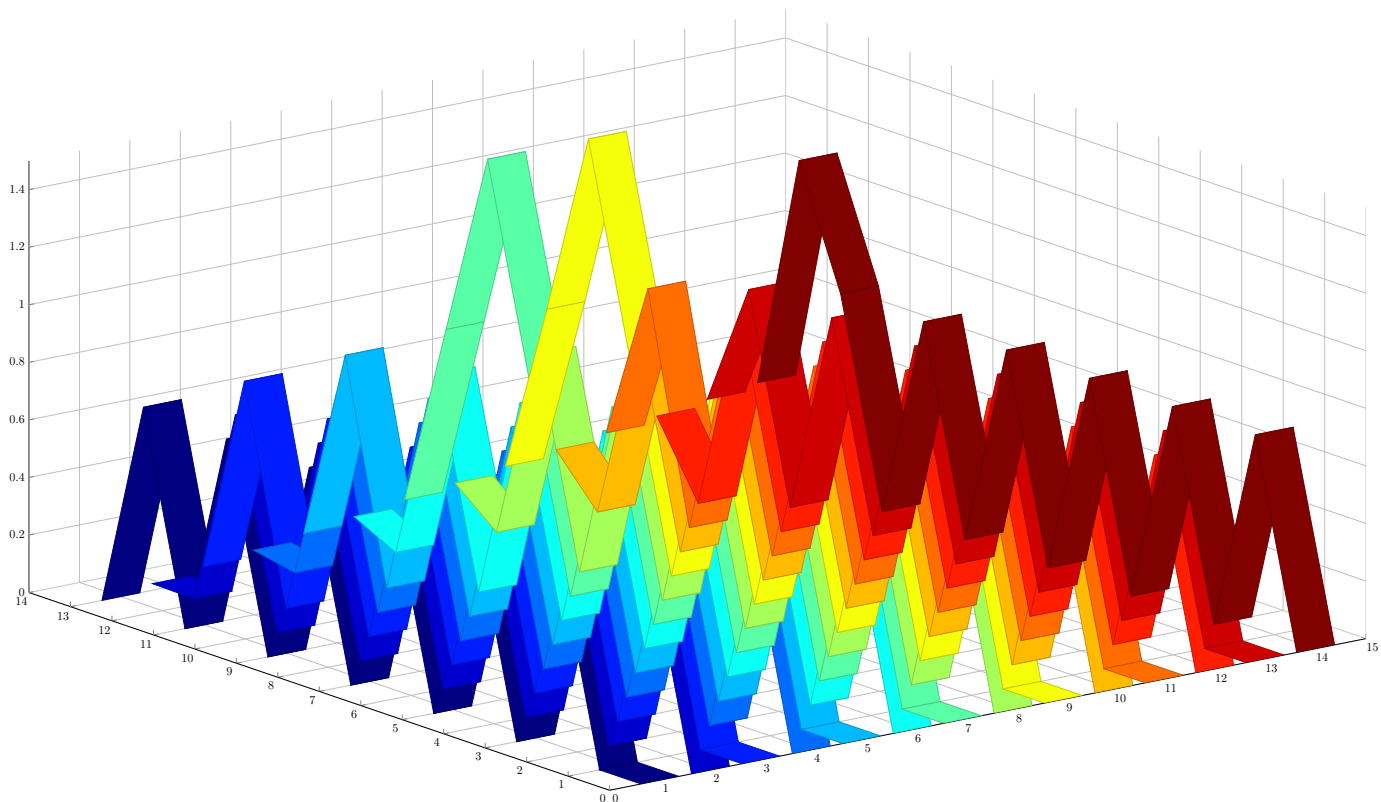


Fig. 12: Final Configuration after 20 iterations of message passing with scheduling and correction factor of  $1/N$ ,  $z\text{-sheer}=0.3$ .

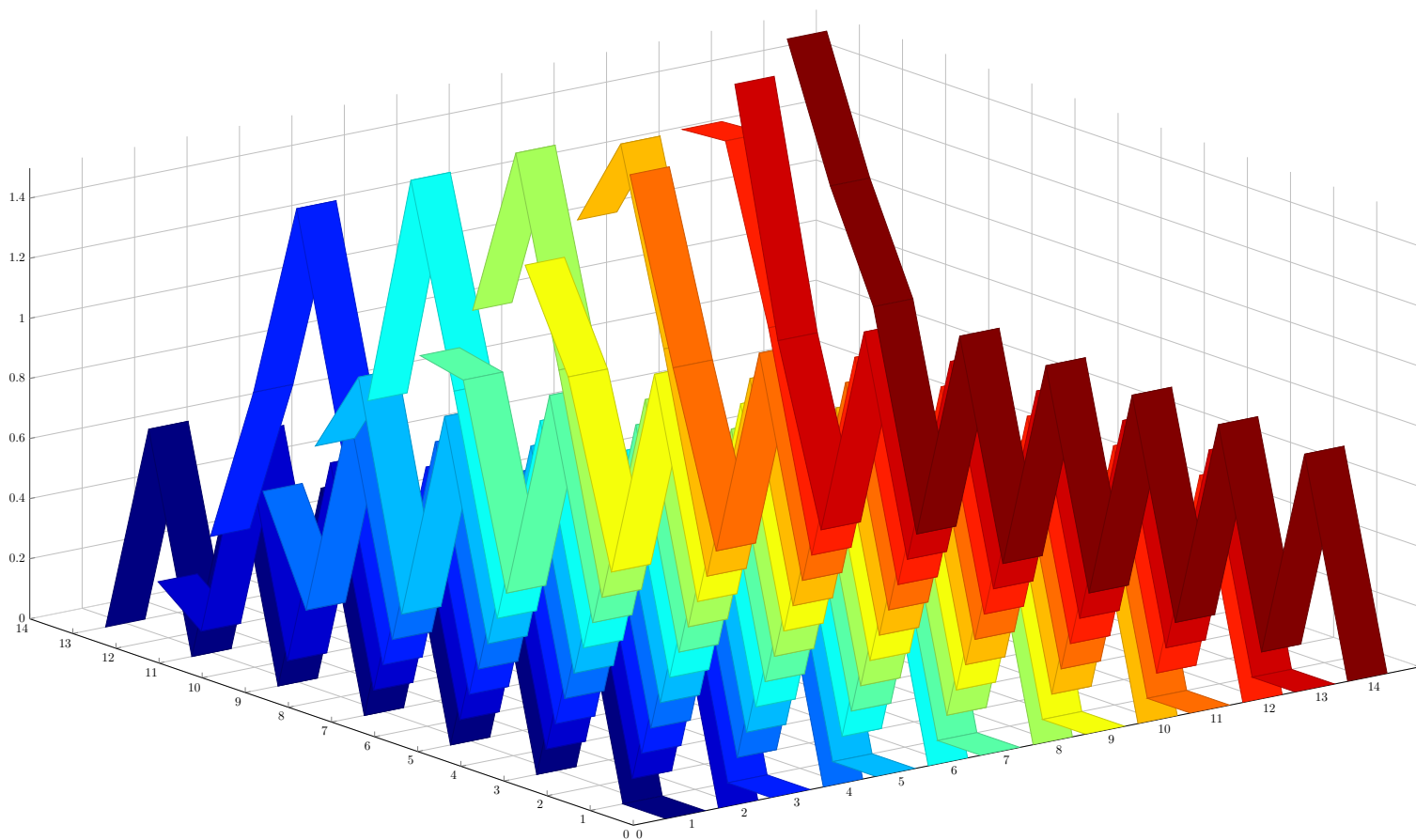


Fig. 13: Final Configuration after 20 iterations of message passing with scheduling and correction factor of  $1/N$ ,  $z\text{-sheer}=1.5$ .

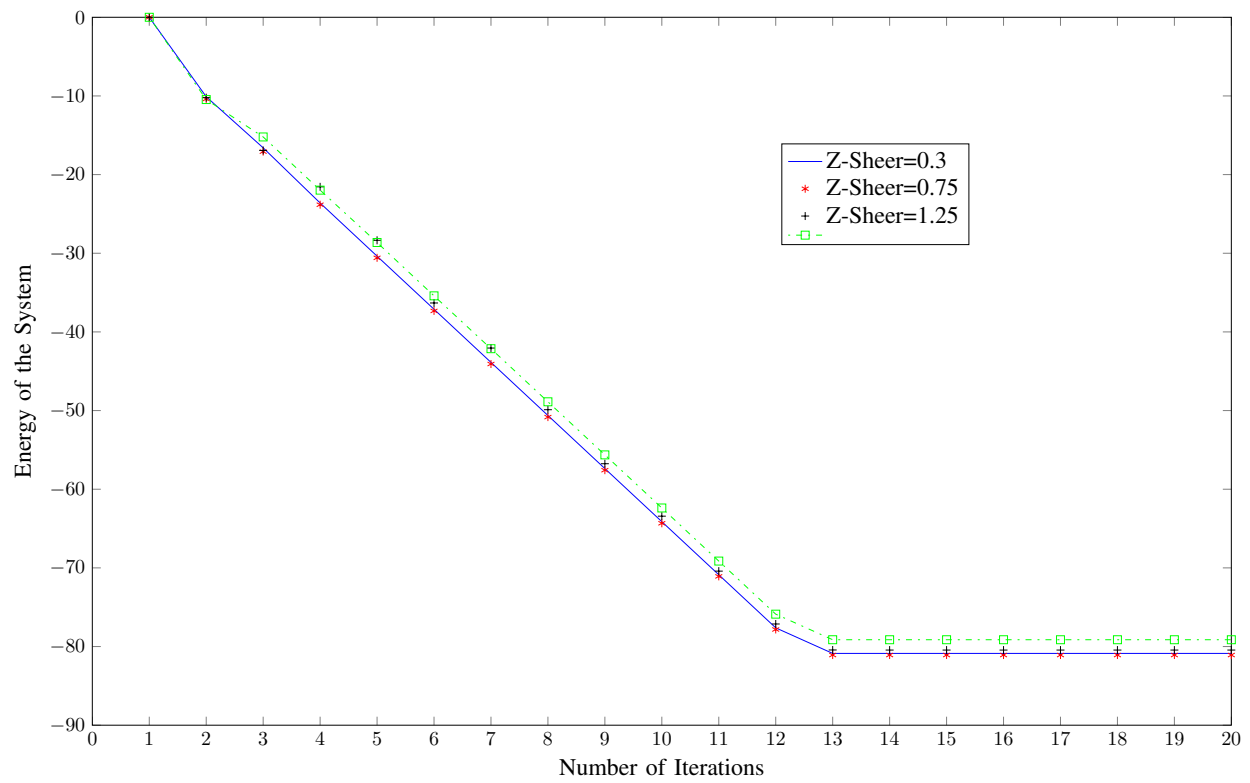


Fig. 14: Total energy output of the algorithm versus the sheer in Z-direction.