block ascent to reach their optimum. Using the entropy barrier function enables us to derive a closed-form update rules.

**Theorem 7.** Consider the primal and dual programs in Theorem 6. Then the block dual ascent takes the form

\[
\forall r, \forall i \in r \quad \lambda_i(r) = \epsilon \log \left( \sum_{r_i \neq r} \exp \left( \left( \phi_i(r_i) + \lambda_i(r_i) / \epsilon \right) \right) \right) - \phi_i(r)
\]

additively normalize \( \lambda_i(r) \) such that \( \sum_{i \in r} \lambda_i(r) = 0 \)

Also, block dual ascent is guaranteed to converge to the dual optimum, the probabilities \( q_i(r_i) \) that are generated throughout the algorithm runtime converge to the primal optimal point and whenever the dual sequence is bounded it is guaranteed to converge to an optimal dual solution.

**Proof:** See supplementary material. \( \square \)

For convenience, the algorithm appears in its explicit form in Fig. 2. The above block dual ascent algorithm can be seen as message-passing, where messages are sent between regions and variables, \( \lambda_i(r) \leftrightarrow \lambda_i \rightarrow r \). This demonstrates the differences between our two algorithms, where the first sends two types of real-valued vectors \( \lambda \rightarrow p \rightarrow c(x_c), \mu \leftarrow p \rightarrow c(x_c) \) along the edges of the region graph, and the second sends a real number \( \lambda \rightarrow r \) between variables and regions. The similarity between the two algorithms is a consequence of the block dual steps and the entropy barrier function.

### 5 Empirical Evaluation

In our experiments we first compared our message-passing algorithm for computing the fractional covering upper bound in equation (2) with the current state-of-the-art solver [12]. We compared these algorithms on the grid shape spin glass model. A spin glass model consists of \( n \) spins \( x_1, \ldots, x_n \in \{-1, 1\} \), whose local potentials are \( \phi_i(x_i) = \theta_i x_i \) and its pairwise potentials are \( \phi_{i,j}(x_i, x_j) = \theta_{i,j} x_i x_j \). The field parameters \( \theta_i \) were chosen uniformly at random from \([-0.05, 0.05]\) and the coupling parameters \( \theta_{i,j} \) were chosen uniformly at random from \([-1, 1]\).

We used the same covering numbers for all edges and squares in the grid, setting \( c_{i,j} = c_{i,j,k,l} = 1/9 \) and \( c_i = 1 - \sum_{r \in R \setminus i} c_r \). Therefore we satisfy the covering number constraints, \( c_r \geq 0 \) and \( \sum_{r \in R} c_r = 1 \). Our algorithm described in Fig. 1 used a region graph which connects squares to pairs and pairs to singletons, namely the Hesse diagram. [12] use a bipartite inner-outer region graph, for which all outer regions, i.e., squares, are connected to all inner regions, i.e., pairs and singletons. We used the same Matlab code, on a single core of Intel i5 with 8GB RAM, for both algorithm as our algorithm can be applied to bipartite region graphs as well. We compared both methods on \( 5 \times 5, \ldots, 100 \times 100 \) grids and the stopping criteria was a primal-dual gap of \( 10^{-4} \). Fig. 3 shows that passing messages over the Hesse-diagram is better than working over the bipartite inner-outer region graph, and the gap between these methods is significant for large graphical models. We attribute this behavior to the fact that on the Hesse diagram we can pass many messages between pairs and singletons, while using time consuming square messages only when they are needed. In contrast [12] use square messages in every message-passing iteration.

In our experiments we also compared our dual decomposition algorithm for recovering the direction of descent over the fractional covering numbers in the tightening procedure, as described in Fig. 2. In this experiment we used the singletons, pairs and squares regions that correspond to the grid shape graphs. We used \( \phi_i(r_i) = H(b^*_r(x_r)) / |r| \), where \( b^*_r(x_r) \) were the optimal beliefs in our previous experiments. The results in Fig. 4 show that the state-of-the-art off-the-shelf solver, the CPLEX, is good for small scale problems but significantly worse when applied to large scale graphical models. We note that our implementation is in Matlab, which has a significant overhead when applied to small scale problems.

![Figure 3: Comparing our message-passing algorithm for computing the fractional covering upper bound in Section 3 with [12]. Comparison is for \( 5 \times 5, \ldots, 100 \times 100 \) grid shape spin glass models, with singletons, pairs and squares regions. Our approach can utilize intermediate size message, between pairs and singletons, while [12] use the inner-outer region graph, thus sending square based messages in every iteration.](image)