Min-Max Problems on Factor-Graphs

Siamak Ravanbakhsh[*] MRAVANBA@UALBERTA.CA
Christopher Srinivasa[‡] CHRIS@PSI.UTORONTO.CA
Brendan Frey[‡] FREY@PSI.UTORONTO.CA
Russell Greiner[*] RGREINER@UALBERTA.CA
∗ Computing Science Dept., University of Alberta, Edmonton, AB T6G 2E8 Canada
‡ PSI Group, University of Toronto, ON M5S 3G4 Canada

Abstract

We study the min-max problem in factor graphs, which seeks the assignment that minimizes the maximum value over all factors. We reduce this problem to both min-sum and sum-product inference, and focus on the later. In this approach the min-max inference problem is reduced to a sequence of Constraint Satisfaction Problems (CSP), which allows us to solve the problem by sampling from a uniform distribution over the set of solutions. We demonstrate how this scheme provides a message passing solution to several NP-hard combinatorial problems, such as min-max clustering (*a.k.a.* K-clustering), asymmetric K-center clustering problem, K-packing and the bottleneck traveling salesman problem. Furthermore, we theoretically relate the min-max reductions and several NP hard decision problems such as clique cover, set-cover, maximum clique and Hamiltonian cycle, therefore also providing message passing solutions for these problems. Experimental results suggest that message passing often provides near optimal min-max solutions for moderate size instances.

1. Introduction

In recent years, message passing methods have achieved a remarkable success in solving different classes of optimization problems, including maximization (*e.g.*, Frey & Dueck 2007;Bayati et al. 2005), integration (*e.g.*, Huang & Jebara 2009) and constraint satisfaction problems (*e.g.*, Mezard et al. 2002).When formulated as a graphical model, these problems correspond to different modes of inference: (a) solving a CSP corresponds to sampling from a uniform distribution over satisfying assignments, while (b) counting and integration usually correspond to estimation of the partition function and (c) maximization corresponds to Maximum a Posteriori (MAP) inference. Here we introduce and study a new class of inference over graphical models –*i.e.*, (d) the min-max inference problem, where the objective is to find an assignment to minimize the maximum value over a set of functions.

The min-max objective appears in various fields, particularly in building robust models under uncertain and adversarial settings. In the context of probabilistic graphical models, several different min-max objectives have been previously studied (*e.g.*, Kearns et al. 2001;Ibrahimi et al. 2011). In combinatorial optimization, min-max may refer to the relation between maximization and minimization in dual combinatorial objectives and their corresponding linear programs (*e.g.*, Schrijver 1983), or it may refer to min-max settings due to uncertainty in the problem specification (*e.g.*, Averbakh 2001;Aissi et al. 2009).

Our setting is closely related to a third class of min-max combinatorial problems that are known as *bottleneck* problems. Instances of these problems include bottleneck traveling salesman problem (Parker & Rardin 1984), min-max clustering (Gonzalez 1985), k-center problem (Dyer & Frieze 1985;Khuller & Sussmann 2000) and bottleneck assignment problem (Gross 1959).

Edmonds & Fulkerson (1970) introduce a bottleneck framework with a duality theorem that relates the min-max objective in one problem instance to a max-min objective in a dual problem. An intuitive example is the duality between the min-max cut separating nodes a and b – the cut with the minimum of the maximum weight – and min-max path between a and b, which is the path with the minimum of the maximum weight (Fulkerson 1966). Hochbaum & Shmoys (1986) leverage triangle inequality in metric spaces to find constant factor approximations to several NP-hard min-max problems under a unified framework.

Proceedings of the 31[st] International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

The common theme in a majority of heuristics for min-max or bottleneck problems is the relation of the min-max objective with a CSP (*e.g.*, Hochbaum & Shmoys 1986; Panigrahy & Vishwanathan 1998). In this paper, we establish a similar relation within the context of factor-graphs, by reducing the min-max inference problem on the original factor-graph to that of sampling (*i.e.*, solving a CSP) on the reduced factor-graph. We also consider an alternative approach where the factor-graph is transformed such that the min-sum objective produces the same optimal result as min-max objective on the original factor-graph. Although this reduction is theoretically appealing, in its simple form it suffers from numerical problems and is not further pursued here.

Section 2 formalizes min-max problem in probabilistic graphical models and provides an inference procedure by reduction to a sequence of CSPs on the factor graph. Section 3 reviews Perturbed Belief Propagation equations (Ravanbakhsh & Greiner 2014) and several forms of high-order factors that allow efficient sum-product inference. Finally Section 4 uses these factors to build efficient algorithms for several important min-max problems with general distance matrices. These applications include problems, such as K-packing, that were not previously studied within the context of min-max or bottleneck problems.

2. Factor Graphs and CSP Reductions

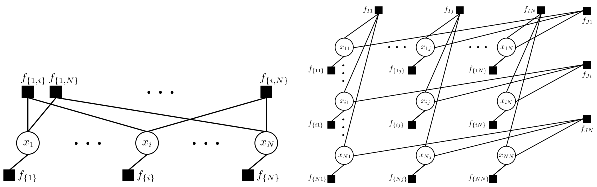
Let $x = \{x_1, \ldots, x_n\}$, where $x \in \mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ denote a set of discrete variables. Each factor $f_I(x_I) : \mathcal{X}_I \to \mathcal{Y}_I \subset \Re$ is a real valued function with range $\mathcal{Y}_I$, over a subset of variables –*i.e.*, $I \subseteq \{1, \ldots, n\}$ is a set of indices. Given the set of factors $\mathcal{F}$, the min-max objective is

$$x^* \quad = \quad \arg_x \min \max_{I \in \mathcal{F}} f_I(x_I) \tag{1}$$

This model can be conveniently represented as a bipartite graph, known as *factor graph* (Kschischang & Frey 2001), that includes two sets of nodes: variable nodes x_i, and factor nodes f_I. A variable node i (note that we will often identify a variable x_i with its index "i") is connected to a factor node I if and only if $i \in I$. We will use ∂ to denote the neighbors of a variable or factor node in the factor graph– that is $\partial I = \{i \ s.t. \ i \in I\}$ (which is the set I) and $\partial i = \{I \ s.t. \ i \in I\}$.

Let $\mathcal{Y} = \bigcup_I \mathcal{Y}_I$ denote the union over the range of all factors. The min-max value belongs to this set $\max_{I \in \mathcal{F}} f_I(x^*_I) \in \mathcal{Y}$. In fact for any assignment x, $\max_{I \in \mathcal{F}} f_I(x_I) \in \mathcal{Y}$.

Example *Bottleneck Assignment Problem:* given a matrix $D \in \Re^{N \times N}$, select a subset of entries of size N that includes exactly one entry from each row and each column, whose largest value is as small as possible. As an application assume the entry $D_{i,j}$ is the time required by worker i to finish task j. The min-max assignment minimizes the maximum time required by any worker to finish his assignment. This problem is also known as bottleneck bipartite matching and belongs to class $\mathcal{P}$ (*e.g.*, Garfinkel 1971). Here we show two factor-graph representations of this problem.



(a) (b)

Figure 1. Factor-graphs for the bottleneck assignment problem

Categorical variable representation: Consider a factor-graph with $x = \{x_1, \ldots, x_N\}$, where each variable $x_i \in \{1, \ldots, N\}$ indicates the column of the selected entry in row i of D. For example $x_1 = 5$ indicates the fifth column of the first row is selected (see Figure 1(a)). Define the following factors: (a) local factors $f_{\{i\}}(x_i) = D_{i,x_i}$ and (b) pairwise factors $f_{\{i,j\}}(x_{\{i,j\}}) = \infty \mathbb{I}(x_i = x_j) - \infty \mathbb{I}(x_i \neq x_j)$ that enforce the constraint $x_i \neq x_j$. Here $\mathbb{I}(.)$ is the indicator function –*i.e.*, $\mathbb{I}(True) = 1$ and $\mathbb{I}(False) = 0$. Also by convention $\infty \ 0 \triangleq 0$. Note that if $x_i = x_j$, $f_{\{i,j\}}(x_{\{i,j\}}) = \infty$, making this choice unsuitable in the min-max solution. On the other hand with $x_i \neq x_j$, $f_{\{i,j\}}(x_{\{i,j\}}) = -\infty$ and this factor has no impact on the min-max value.

Binary variable representation: Consider a factor-graph where $x = [x_{1-1}, \ldots, x_{1-N}, x_{2-1} \ldots, x_{2-N}, \ldots, x_{N-N}] \in \{0,1\}^{N \times N}$ indicates whether each entry is selected $x_{i-j} = 1$ or not $x_{i-j} = 0$ (Figure 1(b)). Here the factors $f_I(x_I) = -\infty \mathbb{I}(\sum_{i \in \partial I} x_i = 1) + \infty \mathbb{I}(\sum_{i \in \partial I} x_i \neq 1)$ ensures that only one variable in each row and column is selected and local factors $f_{i-j}(x_{i-j}) = x_{i-j} D_{i,j} - \infty(1 - x_{i-j})$ have any effect only if $x_{i-j} = 1$.

The min-max assignment in both of these factor-graphs as defined in eq. (1) gives a solution to the bottleneck assignment problem.

For any $y \in \mathcal{Y}$ in the range of factor values, we *reduce* the original min-max problem to a CSP using the following reduction. For any $y \in \mathcal{Y}$, μ_y**-reduction** of the min-max problem eq. (1), is given by

$$\mu_y(x) \quad \triangleq \quad \frac{1}{Z_y} \prod_I \mathbb{I}(f_I(x_I) \leq y) \tag{2}$$

where

$$Z_y \triangleq \sum_{\mathcal{X}} \prod_I \mathbb{I}(f_I(x_I) \leq y) \tag{3}$$

is the normalizing constant and $\mathbb{I}(.)$ is the indicator function.[1] This distribution defines a CSP over $\mathcal{X}$, where $\mu_y(x) > 0$ *iff* x is a satisfying assignment. Moreover, Z_y gives the number of satisfying assignments.

We will use $f_I^y(x_I) \triangleq \mathbb{I}(f_I(x_I) \leq y)$ to refer to reduced factors. The following theorem is the basis of our approach in solving min-max problems.

Theorem 2.1 [2] *Let x^* denote the min-max solution and y^* be its corresponding value –i.e., $y^* = \max_I \; f_I(x_I^*)$. Then $\mu_y(x)$ is satisfiable for all $y \geq y^*$ (in particular $\mu_y(x^*) > 0$) and unsatisfiable for all $y < y^*$.*

This theorem enables us to find a min-max assignment by solving a sequence of CSPs. Let $y^{(1)} \leq \ldots \leq y^{(N)}$ be an ordering of $y \in \mathcal{Y}$. Starting from $y = y^{(\lceil N/2 \rceil)}$, if μ_y is satisfiable then $y^* \leq y$. On the other hand, if μ_y is not satisfiable, $y^* > y$. Using binary search, we need to solve $\log(|\mathcal{Y}|)$ CSPs to find the min-max solution. Moreover at any time-step during the search, we have both upper and lower bounds on the optimal solution. That is $\underline{y} < y^* \leq \overline{y}$, where $\mu_{\underline{y}}$ is the latest unsatisfiable and $\mu_{\overline{y}}$ is the latest satisfiable reduction.

Example *Bottleneck Assignment Problem:* Here we define the μ_y-reduction of the binary-valued factor-graph for this problem by reducing the constraint factors to $f^y(x_I) = \mathbb{I}(\sum_{i \in \partial I} x_i = 1)$ and the local factors to $f^y_{\{i-j\}}(x_{i-j}) = x_{i-j}\mathbb{I}(D_{i,j} \leq y)$. The μ_y-reduction can be seen as defining a uniform distribution over the all valid assignments (*i.e.*, each row and each column has a single entry) where none of the N selected entries are larger than y.

2.1. Reduction to Min-Sum

Kabadi & Punnen (2004) introduce a simple method to transform instances of bottleneck TSP to TSP. Here we show how this results extends to min-max problems over factor-graphs.

Lemma 2.2 *Any two sets of factors, $\{f_I\}_{I \in \mathcal{F}}$ and $\{f'_I\}_{I \in \mathcal{F}}$, have identical min-max solution*

$$\arg_x \min \max_I f_I(x_I) = \arg_x \min \max_I f'_I(x_I)$$

if $\forall I, J \in \mathcal{F}, x_I \in \mathcal{X}_I, x_J \in \mathcal{X}_J$

$$f_I(x_I) < f_J(x_J) \quad \Leftrightarrow \quad f'_I(x_I) < f'_J(x_J)$$

This lemma simply states that what matters in the min-max solution is the *relative ordering* in the factor-values.

[1] To always have a well-defined probability, we define $\frac{0}{0} \triangleq 0$.
[2] All proofs appear in Appendix A.

Let $y^{(1)} \leq \ldots \leq y^{(N)}$ be an ordering of elements in $\mathcal{Y}$, and let $r(f_I(x_I))$ denote the rank of $y_I = f_I(x_I)$ in this ordering. Define the min-sum reduction of $\{f_I\}_{I \in \mathcal{F}}$ as

$$f'_I(x_I) = 2^{r(f_I(x_I))} \quad \forall I \in \mathcal{F}$$

Theorem 2.3

$$\arg_x \min \sum_I f'_I(x_I) \quad = \quad \arg_x \min \max_I f_I(x_I)$$

where $\{f'_I\}_I$ is the min-sum reduction of $\{f_I\}_I$.

Although this allows us to use min-sum message passing to solve min-max problems, the values in the range of factors grow exponentially fast, resulting in numerical problems.

3. Solving CSP-reductions

Previously in solving CSP-reductions, we assumed an ideal CSP solver. However, finding an assignment x such that $\mu_y(x) > 0$ or otherwise showing that no such assignment exists is in general NP-hard (Cooper 1990). However, message passing methods have been successfully used to provide state-of-the-art results in solving difficult CSPs. We use Perturbed Belief Propagation (PBP Ravanbakhsh & Greiner 2014) for this purpose. By using an incomplete solver (Kautz et al. 2009), we lose the lower-bound $\underline{y}$ on the optimal min-max solution, as PBP may not find a solution even if the instance is satisfiable. [3] However the following theorem states that, as we increase y from the min-max value y^*, the number of satisfying assignments to μ_y-reduction increases, making it potentially easier to solve.

Proposition 3.1

$$y_1 < y_2 \quad \Rightarrow \quad Z_{y_1} \leq Z_{y_2} \qquad \forall y_1, y_2 \in \Re$$

where Z_y is defined in eq. (3).

This means that the sub-optimality of our solution is related to our ability to solve CSP-reductions – that is, as the gap $y - y^*$ increases, the μ_y-reduction potentially becomes easier to solve.

PBP is a message passing method that interpolates between Belief Propagation (BP) and Gibbs Sampling. At each iteration, PBP sends a message from variables to factors ($\nu_{i \to I}$) and vice versa ($\nu_{I \to i}$). The factor to variable message is given by

$$\nu_{I \to i}(x_i) \;\propto\; \sum_{x_{I \setminus i} \in \mathcal{X}_{\partial I \setminus i}} f_I^y(x_i, x_{I \setminus i}) \prod_{j \in \partial I \setminus i} \nu_{j \to I}(x_j) \tag{4}$$

[3] To maintain the lower-bound one should be able to correctly assert unsatisfiability.

where the summation is over all the variables in I except for x_i. The variable to factor message for PBP is slightly different from BP; it is a linear combination of the BP message update and an indicator function, defined based on a sample from the current estimate of marginal $\widehat{\mu}(x_i)$:

$$\nu_{i\to I}(x_i) \;\propto\; (1-\gamma) \prod_{J\in\partial i\setminus I} \nu_{J\to i}(x_i) \;+\; \gamma\, \mathbb{I}(\widehat{x}_i = x_i) \tag{5}$$

$$\text{where } \;\widehat{x}_i \sim \widehat{\mu}(x_i) \;\propto\; \prod_{J\in\partial i} \nu_{J\to i}(x_i) \tag{6}$$

where for $\gamma = 0$ we have BP updates and for $\gamma = 1$, we have Gibbs Sampling. PBP starts at $\gamma = 0$ and linearly increases γ at each iteration, ending at $\gamma = 1$ at its final iteration. At any iteration, PBP may encounter a contradiction where the product of incoming messages to node i is zero for all $x_i \in \mathcal{X}_i$. This could mean that either the problem is unsatisfiable or PBP is not able to find a solution. However if it reaches the final iteration, PBP produces a sample from $\mu_y(x)$, which is a solution to the corresponding CSP. The number of iterations T is the only parameter of PBP and increasing T, increases the chance of finding a solution (Only downside is time complexity; nb., no chance of a false positive.)

3.1. Computational Complexity

PBP's time complexity per iteration is identical to that of BP– *i.e.*,

$$\mathcal{O}\left(\sum_I (|\partial I|\; |\mathcal{X}_I|) + \sum_i (|\partial i|\; |\mathcal{X}_i|)\right) \tag{7}$$

where the first summation accounts for all factor-to-variable messages (eq. (4)) [4] and the second one accounts for all variable-to-factor messages (eq. (5)).

To perform binary search over $\mathcal{Y}$ we need to sort $\mathcal{Y}$, which requires $\mathcal{O}(|\mathcal{Y}|\log(|Y|))$. However, since $|\mathcal{Y}_i| \leq |\mathcal{X}_i|$ and $|\mathcal{Y}| \leq \sum_I |\mathcal{Y}_i|$, the cost of sorting is already contained in the first term of eq. (7), and may be ignored in asymptotic complexity.

The only remaining factor is that of binary search itself, which is $\mathcal{O}(\log(|\mathcal{Y}|)) = \mathcal{O}(\log(\sum_I (|\mathcal{X}_I|)))$ – *i.e.*, at most logarithmic in the cost of PBP's iteration (*i.e.*, first term in eq. (7)). Also note that the factors added as constraints only take two values of $\pm\infty$, and have no effect in the cost of binary search.

As this analysis suggests, the dominant cost is that of sending factor-to-variable messages, where a factor may depend on a large number of variables. The next section shows that many interesting factors are sparse, which allows efficient calculation of messages.

3.2. High-Order Factors

The factor-graph formulation of many interesting min-max problems involves sparse high-order factors. In all such factors, we are able to significantly reduce the $\mathcal{O}(|\mathcal{X}_I|)$ time complexity of calculating factor-to-variable messages. Efficient message passing over such factors is studied by several works in the context of sum-product and max-product inference (*e.g.*, Potetz & Lee 2008; Gupta et al. 2007;Tarlow et al. 2010;Tarlow et al. 2012). The simplest form of sparse factor in our formulation is the so-called Potts factor, $f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i = x_j)\phi(x_i)$. This factor assumes the same domain for all the variables ($\mathcal{X}_i = \mathcal{X}_j \;\forall i, j$) and its tabular form is non-zero only across the diagonal. It is easy to see that this allows the marginalization of eq. (4) to be performed in $\mathcal{O}(|\mathcal{X}_i|)$ rather than $\mathcal{O}(|\mathcal{X}_i|\;|\mathcal{X}_j|)$. Another factor of similar form is the inverse Potts factor, $f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i \neq x_j)$, which ensures $x_i \neq x_j$. In fact any pair-wise factor that is a constant plus a band-limited matrix allows $\mathcal{O}(|\mathcal{X}_i|)$ inference (*e.g.*, see Section 4.4).

In Section 4, we use cardinality factors, where $\mathcal{X}_i = \{0, 1\}$ and the factor is defined based on the number of non-zero values –*i.e.*, $f^y_{\mathcal{K}}(x_{\mathcal{K}}) = f^y_{\mathcal{K}}(\sum_{i\in\mathcal{K}} x_i)$. The μ_y-reduction of the factors we use in the binary representation of the bottleneck assignment problem is in this category. Gail et al. (1981) propose a simple $\mathcal{O}(|\partial I|\;\; K)$ method for $f^y_{\mathcal{K}}(x_{\mathcal{K}}) = \mathbb{I}(\sum_{i\in\mathcal{K}} x_i = K)$. We refer to this factor as K-of-N factor and use similar algorithms for at-least-K-of-N $f^y_{\mathcal{K}}(x_{\mathcal{K}}) = \mathbb{I}(\sum_{i\in\mathcal{K}} x_i \geq K)$ and at-most-K-of-N $f^y_{\mathcal{K}}(x_{\mathcal{K}}) = \mathbb{I}(\sum_{i\in\mathcal{K}} x_i \leq K)$ factors (see Appendix B). An alternative for more general forms of high order factors is the clique potential of Potetz & Lee (2008). For large K, more efficient methods evaluate the sum of pairs of variables using auxiliary variables forming a binary tree and use Fast Fourier Transform to reduce the complexity of K-of-N factors to $\mathcal{O}(N\log(N)^2)$ (see Tarlow et al. (2012) and its references).

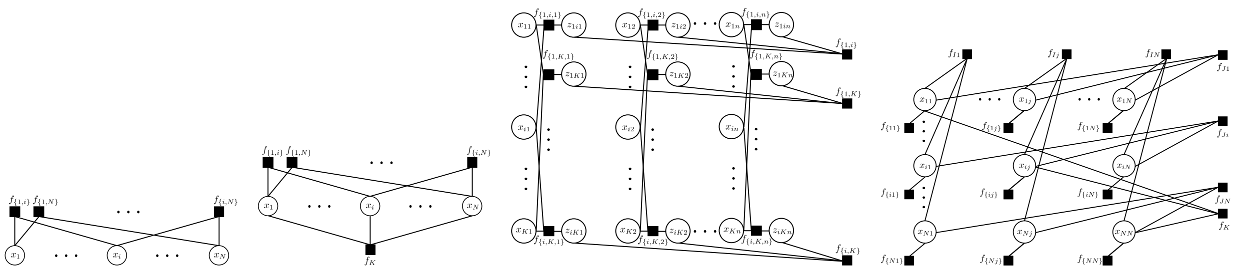
4. Applications

Here we introduce the factor-graph formulation for several NP-hard min-max problems. Interestingly the CSP-reduction for each case is an important NP-hard problem. Table 1 shows the relationship between the min-max and the corresponding CSP and the factor α in the constant factor approximation available for each case. For example, $\alpha = 2$ means the results reported by some algorithm is guaranteed to be within factor 2 of the optimal min-max value $\widehat{y}^*$ when the distances satisfy the triangle inequal-

[4] The $|\partial I|$ is accounting for the number of messages that are sent from each factor. However if the messages are calculated synchronously this factor disappears. Although more expensive, in practice, asynchronous message updates performs better.

Table 1. Min-max combinatorial problems and the corresponding CSP-reductions. The last column reports the best α-approximations when triangle inequality holds. $*$ indicates best possible approximation.

min-max problem	μ_y-reduction	msg-passing cost	α
min-max clustering	clique cover problem	$\mathcal{O}(N^2 K \log(N))$	2^* (Gonzalez 1985)
K-packing	max-clique problem	$\mathcal{O}(N^2 K \log(N))$	N/A
(weighted) K-center problem	dominating set problem	$\mathcal{O}(N^3 \log(N))$ or $\mathcal{O}(NR^2 \log(N))$	$\min(3, 1 + \frac{\max_i w_i}{\min_i w_i})$ (Dyer & Frieze 1985)
asymmetric K-center problem	set cover problem	$\mathcal{O}(N^3 \log(N))$ or $\mathcal{O}(NR^2 \log(N))$	$\log(N)^*$ (Panigrahy & Vishwanathan 1998;Chuzhoy et al. 2005)
bottleneck TSP	Hamiltonian cycle problem	$\mathcal{O}(N^3 \log(N))$	2^* (Parker & Rardin 1984)
bottleneck Asymmetric TSP	directed Hamiltonian cycle	$\mathcal{O}(N^3 \log(N))$	$\log(n)/\log(\log(n))$ (An et al. 2010)



(a) (b) K-packing (binary) (c) sphere packing (d) K-center

Figure 2. The factor-graphs for different applications. Factor-graph (a) is common between min-max clustering, Bottleneck TSP and K-packing (categorical). However the definition of factors is different in each case.

ity. This table also includes the complexity of the message passing procedure (assuming asynchronous message updates) in finding the min-max solution. See Appendix C for details.

4.1. Min-Max Clustering

Given a symmetric matrix of pairwise distances $D \in \Re^{N \times N}$ between N data-points, and a number of clusters K, min-max clustering seeks a partitioning of data-points that minimizes the maximum distance between all the pairs in the same partition.

Let $x = \{x_1, \ldots, x_N\}$ with $x_i \in \mathcal{X}_i = \{1, \ldots, K\}$ be the set of variables, where $x_i = k$ means, point i belongs to cluster k. The Potts factor $f_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i = x_j) D_{i,j} - \infty \mathbb{I}(x_i \neq x_j)$ between any two points is equal to $D_{i,j}$ if points i and j belong the same cluster and $-\infty$ otherwise (Figure 2(a)). When applied to this factor graph, the min-max solution x^* of eq. (1) defines the clustering of points that minimizes the maximum inter-cluster distances.

Now we investigate properties of μ_y-reduction for this factor-graph. The *y-neighborhood graph* for distance matrix $D \in \Re^{N \times N}$ is defined as graph $\mathcal{G}(D, y) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ and $\mathcal{E} = \{(i,j) |\ D_{i,j} \leq y\}$. Note than this definition is also valid for an asymmetric adjacency matrix D. In such cases, the y-neighborhood graph is a directed-graph.

The *K-clique-cover* $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a partitioning of $\mathcal{V}$ to at most K partitions such that $\forall i, j, k \quad i, j \in \mathcal{C}_k \Rightarrow (i,j) \in \mathcal{E}$.

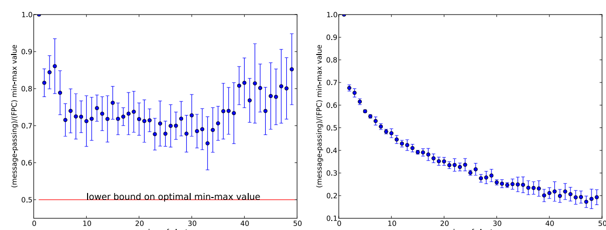


Figure 3. Min-max clustering of 100 points with varying numbers of clusters (x-axis). Each point is an average over 10 random instances. The y-axis is the ratio of the min-max value obtained by message passing ($T = 50$ iterations for PBP) over the min-max value of FPC. **(left)** Clustering of random points in 2D Euclidean space. The red line is the lower bound on the optimal result based on the worst case guarantee of FPC. **(right)** Using symmetric random distance matrix where $D_{i,j} = D_{j,i} \sim U(0,1)$.

Proposition 4.1 *The μ_y-reduction of the min-max clustering factor-graph defines a uniform distribution over the K-clique-covers of $\mathcal{G}(D, y)$.*

Figure 3 compares the performance of min-max clustering using message passing to that of Furthest Point Clustering (FPC) of Gonzalez (1985) which is 2-optimal when the triangle inequality holds. Note that message passing solutions are superior even when using Euclidean distance.

4.2. K-Packing

Given a symmetric distance matrix $D \in \Re^{N \times N}$ between N data-points and a number of code-words K, the K-packing problem is to choose a subset of K points such that the minimum distance $D_{i,j}$ between any two code-words is maximized. Here we introduce two different factor-graph formulations for this problem.

4.2.1. First Formulation: Binary Variables

Let binary variables $x = \{x_1, \ldots, x_N\} \in \{0,1\}^N$, indicate a subset of variables of size K that are selected as code-words (Figure 2(b)). Use the factor $f_{\mathcal{K}}(x) = \infty \mathbb{I}(\sum_i x_i \neq K) - \infty \mathbb{I}(\sum_i x_i = K)$ (here $\mathcal{K} = \{1, \ldots, N\}$) to ensure this constraint. The μ_y-reduction of this factor for any $-\infty < y < +\infty$ is a K-of-N factor as defined in Section 3.2. Furthermore, for any two variables x_i and x_j, define factor $f_{x_i, x_j}(x_{i,j}) = -D_{i,j} x_i x_j - \infty(1 - x_i x_j)$. This factor is effective only if both points are selected as code-words. The use of $-D_{i,j}$ is to convert the initial max-min objective to min-max.

4.2.2. Second Formulation: Categorical Variables

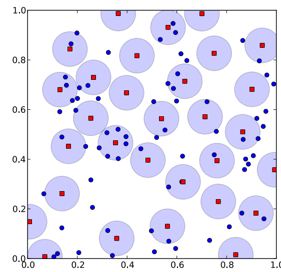
Define the K-packing factor-graph as follows: Let $x = \{x_1, \ldots, x_K\}$ be the set of variables where $x_i \in \mathcal{X}_i = \{1, \ldots, N\}$ (Figure 2(a)). For every two distinct points $1 \leq i < j \leq K$ define the factor $f_{\{i,j\}}(x_i, x_j) = -D_{x_i, x_j} \mathbb{I}(x_i \neq x_j) + \infty \mathbb{I}(x_i = x_j)$. Here each variable represents a code-word and the last term of each factor ensures that code-words are distinct.

Proposition 4.2 *The μ_y-reduction of the K-packing factor-graph for the distance matrix $D \in \Re^{N \times N}$ defines a uniform distribution over the cliques of $\mathcal{G}(-D, -y)$ that are larger than K.*

The μ_y-reduction of our second formulation is similar to the factor-graph used by Ramezanpour & Zecchina (2012) to find non-linear binary codes. The authors consider the Hamming distance between all binary vectors of length n (*i.e.*, $N = 2^n$) to obtain binary codes with known minimum distance y. As we saw, this method is $\mathcal{O}(N^2 K^2) = \mathcal{O}(2^{2n} K^2)$ –*i.e.*, grows exponentially in the number of bits n. In the following section, we introduce a factor-graph formulation specific to categorical variables with Hamming distance that have message passing complexity $\mathcal{O}(n^2 K^2 y)$ –*i.e.*, not exponential in n. Using this formulation we find optimal binary codes where both n and y are large.

4.2.3. Sphere packing with Hamming Distance

Our factor-graph defines a distribution over the K binary vectors of length n such that the distance between every pair of binary vectors is at least y.[5] Finding so-called "nonlinear binary codes" is a fundamental problem in information theory and a variety of methods have been used to find better codes, trying either to maximize the number of keywords K or their minimum distance y (*e.g.*, see Litsyn et al. 1999 and its references). Let $x = \{x_{1-1}, \ldots, x_{1-n}, x_{2-1}, \ldots, x_{2-n}, \ldots, x_{K-n}\}$ be the set of our binary variables, where $x_i = \{x_{i-1}, \ldots, x_{i-n}\}$ represents the i^{th} binary vector or code-word. Additionally for each $1 \leq i < j \leq K$, define an auxiliary binary vector $z_{i,j} = \{z_{i,j,1}, \ldots, z_{i,j,n}\}$ of length n (Figure 2(c)).



2	1	2	1	0	1	1	2	2	1	2	1	2	1	0	2
1	1	1	2	1	0	0	2	2	1	1	1	0	2	1	0
0	0	1	2	0	1	2	0	2	1	2	2	1	0	2	0
0	0	0	2	1	0	1	2	1	0	1	0	2	1	0	1
2	2	0	2	2	2	1	1	2	0	2	2	1	2	1	2
0	2	0	0	0	2	0	2	0	0	2	1	0	0	0	0
1	1	2	0	0	1	2	2	0	0	1	0	1	2	2	1
2	0	2	0	2	1	0	1	1	2	0	1	1	1	2	0
0	2	1	1	1	1	1	1	0	1	0	0	0	0	1	2
1	0	0	1	0	2	0	0	1	2	0	0	2	2	1	2
1	2	2	1	1	0	2	1	1	2	2	2	2	0	2	1
1	0	1	0	2	2	2	1	0	1	1	2	2	1	0	2

Figure 4. **(left)** Using message passing to choose $K = 30$ out of $N = 100$ random points in the Euclidean plane to maximize the minimum pairwise distance (with $T = 500$ iterations for PBP). Touching circles show the minimum distance. **(right)** Example of an optimal ternary code ($n = 16, y = 11, K = 12$), found using the K-packing factor-graph of Section 4.2.3. Here each of $K = 12$ lines contains one code-word of length 16, and every pair of code-words are different in at least $y = 11$ digits.

For each distinct pair of binary vectors x_i and x_j, and a particular bit $1 \leq k \leq n$, the auxiliary variable $z_{i,j,k} = 1$ *iff* $x_{i-k} \neq x_{j-k}$. Then we define an at-least-y-of-n factor over $z_{i,j}$ for every pair of code-words, to ensure that they differ in at least y bits.

In more details, define the following factors on x and z:
(a) *z-factors:* For every $1 \leq i < j \leq K$ and $1 \leq k \leq n$, define a factor to ensure that $z_{i,j,k} = 1$ *iff* $x_{i-k} \neq x_{j-k}$:

$$f(x_{i-k}, x_{j-k}, z_{i,j,k}) = \mathbb{I}(x_{i-k} \neq x_{j-k}) \mathbb{I}(z_{i,j,k} = 1) + \mathbb{I}(x_{i-k} = x_{j-k}) \mathbb{I}(z_{i,j,k} = 0).$$

This factor depends on three binary variables, therefore we can explicitly define its value for each of $2^3 = 8$ possible inputs.

(b) *distance-factors:* For each $z_{i,j}$ define at-least-y-of-n factor:

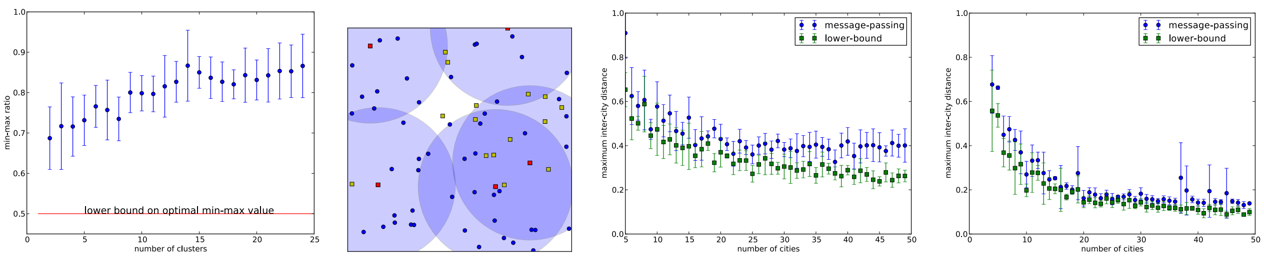
$$f_{\mathcal{K}}(z_{i,j}) = \mathbb{I}\Big(\sum_{1 \leq k \leq n} z_{i,j,k} \geq y\Big)$$

Table 2 reports some optimal codes including codes with large number of bits n, recovered using this factor-graph. Here Perturbed BP used $T = 1000$ iterations.

4.3. (Asymmetric) K-center Problem

Given a pairwise distance matrix $D \in \Re^{N \times N}$, the K-center problem seeks a partitioning of nodes, with one center per

[5] For convenience we restrict this construction to the case of binary vectors. A similar procedure may be used to find maximally distanced ternary and q-ary vectors, for arbitrary q.



(a) K-center (Euclidean) (b) Facility Location (c) BTSP (Euclidean) (d) BTSP (Random)

Figure 5. **(a)** K-center clustering of 50 random points in a 2D plane with various numbers of clusters (x-axis). The y-axis is the ratio of the min-max value obtained by message passing ($T = 500$ for PBP) over the min-max value of 2-approximation of Dyer & Frieze (1985). **(b)** Min-max K-facility location formulated as an asymmetric K-center problem and solved using message passing. Squares indicate 20 potential facility locations and circles indicate 50 customers. The task is to select 5 facilities (red squares) to minimize the maximum distance from any customer to a facility. The radius of circles is the min-max value. **(c,d)** The min-max solution for Bottleneck TSP with different number of cities (x-axis) for 2D Euclidean space as well as asymmetric random distance matrices ($T = 5000$ for PBP). The error-bars in all figures show one standard deviation over 10 random instances.

Table 2. Some optimal binary codes from Litsyn et al. 1999 recovered by K-packing factor-graph in the order of increasing y. n is the length of the code, K is the number of code-words and y is the minimum distance between code-words.

n	K	y	n	K	y	n	K	y	n	K	y
8	4	5	11	4	7	14	4	9	16	6	9
17	4	11	19	6	11	20	8	11	20	4	13
23	6	13	24	8	13	23	4	15	26	6	15
27	8	15	28	10	15	28	5	16	26	4	17
29	6	17	29	4	19	33	6	19	34	8	19
36	12	19	32	4	21	36	6	21	38	8	21
39	10	21	35	4	23	39	6	23	41	8	23
39	4	25	43	6	23	46	10	25	47	12	25
41	4	27	46	6	27	48	8	27	50	10	27
44	4	29	49	6	29	52	8	29	53	10	29

partition such that the maximum distance from any node to the center of its partition is minimized. This problem is known to be NP-hard, even for Euclidean distance matrices (Masuyama et al. 1981). Frey & Dueck (2007) use max-product message passing to solve the min-sum variation of this problem –*a.k.a.* K-median problem. A binary variable factor-graph for the same problem is introduced in Givoni & Frey (2009). Here we introduce a binary variable model for the asymmetric K-center problem. Let $x = \{x_{1-1}, \ldots, x_{1-N}, x_{2-1}, \ldots, x_{2-N}, \ldots, x_{N-N}\}$ denote N^2 binary variables, where $x_{i-j} = 1$ indicates that point i participates in the partition that has j as its center. Now define the following factors:

A. *local factors:* $\forall i \neq j \quad f_{\{i-j\}}(x_{i-j}) = D_{i,j} x_{i-j} - \infty(1 - x_{i-j})$.

B. *uniqueness factors:* every point is associated with exactly one center (which can be itself). For every i consider $I = \{i - j \mid 1 \leq j \leq N\}$ and define $f_I(x_I) = \infty\mathbb{I}(\sum_{i-j \in \partial I} x_{i-j} \neq 1) - \infty\mathbb{I}(\sum_{i-j \in \partial I} x_{i-j} = 1)$.

C. *consistency factors:* when j is selected as a center by any node i, node j also recognizes itself as a center. $\forall j, i \neq j$ define $f(x_{\{j-j, i-j\}}) = \infty\mathbb{I}(x_{j-j} = 0 \wedge x_{i-j} = 1) - \infty\mathbb{I}(x_{j-j} = 1 \vee x_{i-j} = 0)$.

D. *K-of-N factor:* only K nodes are selected as centers. Letting $\mathcal{K} = \{i - i \mid 1 \leq i \leq N\}$, define $f_{\mathcal{K}}(x_{\mathcal{K}}) = \infty\mathbb{I}(\sum_{i-i \in \mathcal{K}} x_{i-i} \neq K) - \infty\mathbb{I}(\sum_{i-i \in \mathcal{K}} x_{i-i} = K)$.

For variants of this problem such as the capacitated K-center, additional constraints on the maximum/minimum points in each group may be added as the at-least/at-most K-of-N factors.

We can significantly reduce the number of variables and the complexity (which is $\mathcal{O}((N^3 \log(N)))$ by bounding the distance to the center of the cluster $\overline{y}$. Given an upper bound $\overline{y}$, we may remove all the variables x_{i-j} for $D_{i,j} > \overline{y}$ from the factor-graph. Assuming that at most R nodes are at distance $D_{i-j} \leq \overline{y}$ from every node j, the complexity of min-max inference drops to $\mathcal{O}(NR^2 \log(N))$.

Figure 5(a) compares the performance of message-passing and the 2-approximation of Dyer & Frieze (1985) when triangle inequality holds. The min-max facility location problem can also be formulated as an asymmetric K-center problem where the distance *to* all customers is ∞ and the distance from a facility to another facility is $-\infty$ (Figure 5(b)).

The following proposition establishes the relation between the K-center factor-graph above and dominating set problem as its CSP reduction. *The K-dominating set* of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a subset of nodes $\mathcal{D} \subseteq \mathcal{V}$ of size $|\mathcal{D}| = K$ such that any node in $\mathcal{V} \setminus \mathcal{D}$ is adjacent to at least one member of D – *i.e.*, $\forall i \in \mathcal{V} \setminus \mathcal{D} \quad \exists j \in \mathcal{D}$ *s.t.* $(i, j) \in \mathcal{E}$.

Proposition 4.3 *For symmetric distance matrix $D \in \Re^{N \times N}$, the μ_y-reduction of the K-center factor-graph* *above, is non-zero (i.e., $\mu_y(x) > 0$)* iff *x defines a K-dominating set for $\mathcal{G}(D, y)$.*

Note that in this proposition (in contrast with Propositions 4.1 and 4.2) the relation between the assignments x and K-dominating sets of $\mathcal{G}(D, y)$ is not one-to-one as several assignments may correspond to the same dominating set. Here we establish a similar relation between asymmetric K-center factor-graph and set-cover problem.

Given universe set $\mathcal{V}$ and a set $\mathcal{S} = \{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ *s.t.* $\mathcal{V}_m \subseteq \mathcal{V}$, we say $\mathcal{C} \subseteq \mathcal{S}$ *covers* $\mathcal{V}$ *iff* each member of $\mathcal{V}$ is present in at least one member of $\mathcal{C}$ *–i.e.*, $\bigcup_{\mathcal{V}_m \in \mathcal{C}} \mathcal{V}_m = \mathcal{V}$. Now we consider a natural set-cover problem induced by any directed-graph. Given a directed-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for each node $i \in \mathcal{V}$, define a subset $\mathcal{V}_i = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$ as the set of all nodes that are connected *to* i. Let $\mathcal{S} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$ denote all such subsets. An *induced K-set-cover* of $\mathcal{G}$ is a set $\mathcal{C} \subseteq \mathcal{S}$ of size K that covers $\mathcal{V}$.

Proposition 4.4 *For a given asymmetric distance matrix $D \in \Re^{N \times N}$, the μ_y-reduction of the K-center factor-graph as defined above, is non-zero (i.e., $\mu_y(x) > 0$)* iff *x defines an induced K-set-cover for $\mathcal{G}(D, y)$.*

4.4. (Asymmetric) Bottleneck Traveling Salesman Problem

Given a distance matrix $D \in \Re^{N \times N}$, the task in the Bottleneck Traveling Salesman Problem (BTSP) is to find a tour of all N points such that the maximum distance between two consecutive cities in the tour is minimized (Kabadi & Punnen 2004). Any constant-factor approximation for arbitrary instances of this problem is $\mathcal{NP}$-hard (Parker & Rardin 1984).

Let $x = \{x_1, \ldots, x_N\}$ denote the set of variables where $x_i \in \mathcal{X}_i = \{0, \ldots, N-1\}$ represents the time-step at which node i is visited. Also, we assume modular arithmetic (module N) on members of $\mathcal{X}_i$ *–e.g.*, $N \equiv 0 \mod N$ and $1 - 2 \equiv N - 1 \mod N$. For each pair x_i and x_j of variables, define the factor (Figure 2(a))

$$f_{\{i,j\}}(x_i, x_j) = \infty \mathbb{I}(x_i = x_j) - \infty \mathbb{I}(|x_i - x_j| > 1) \quad (8)$$
$$+ D_{i,j} \mathbb{I}(x_i = x_j - 1) + D_{j,i} \mathbb{I}(x_i = x_j - 1)$$

where the first term ensures $x_i \neq x_j$ and the second term means this factor has no effect on the min-max value when node i and j are not consecutively visited in a path. The third and fourth terms express the distance between i and j depending on the order of visit. Figure 6 shows the tabular form of this factor. In Appendix B.2 we show an $\mathcal{O}(N)$ procedure to marginalize this type of factor.

Here we relate the min-max factor-graph above to a uniform distribution over Hamiltonian cycles.

$$\begin{bmatrix} \infty & D_{i,j} & -\infty & \cdots & -\infty & -\infty & D_{j,i} \\ D_{j,i} & \infty & D_{i,j} & \cdots & -\infty & -\infty & -\infty \\ -\infty & D_{j,i} & \infty & \cdots & -\infty & -\infty & -\infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -\infty & -\infty & -\infty & \cdots & \infty & D_{i,j} & -\infty \\ -\infty & -\infty & -\infty & \cdots & D_{j,i} & \infty & D_{i,j} \\ D_{i,j} & -\infty & -\infty & \cdots & -\infty & D_{j,i} & \infty \end{bmatrix}$$

Figure 6. The tabular form of $f_{\{i,j\}}(x_i, x_j)$ used for the bottleneck TSP.

Proposition 4.5 *For any distance matrix $D \in \Re^{N \times N}$, the μ_y-reduction of the BTSP factor-graph (shown above), defines a uniform distribution over the (directed) Hamiltonian cycles of $\mathcal{G}(D, y)$.*

Figure 5(c,d) reports the performance of message passing (over 10 instances) as well as a lower-bound on the optimal min-max value for tours of different length (N). Here we report the results for random points in 2D Euclidean space as well as asymmetric random distance matrices. For symmetric case, the lower-bound is the maximum over j of the distance of two closest neighbors to each node j. For the asymmetric random distance matrices, the maximum is over all the minimum length incoming edges and minimum length outgoing edges for each node.[6]

5. Conclusion

This paper introduces the problem of min-max inference in factor-graphs and provides a general methodology to solve such problems. We use Factor-graphs to represent several important combinatorial problems such as min-max clustering, K-clustering, bottleneck TSP and K-packing and use message passing to find near optimal solutions to each problem. In doing so, we also suggest a message passing solution to several $\mathcal{NP}$-hard decision problems including the clique-cover, max-clique, dominating-set, set-cover and Hamiltonian path problem. For each problem we also analyzed the complexity of message passing and established its practicality using several relevant experiments.

References

Aissi, Hassene, Bazgan, Cristina, and Vanderpooten, Daniel. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438, 2009.

An, Hyung-Chan, Kleinberg, Robert D, and Shmoys, David B. Approximation algorithms for the bottleneck asymmetric traveling salesman problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 1–11. Springer, 2010.

[6] If for one node the minimum length in-coming and out-going edges point to the same city, the second minimum length in-coming and out-going edges are also considered in calculating a tighter bound.

Averbakh, Igor. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90(2):263–272, 2001.

Bayati, Mohsen, Shah, Devavrat, and Sharma, Mayank. Maximum weight matching via max-product belief propagation. In *ISIT*, pp. 1763–1767. IEEE, 2005.

Chuzhoy, Julia, Guha, Sudipto, Halperin, Eran, Khanna, Sanjeev, Kortsarz, Guy, Krauthgamer, Robert, and Naor, Joseph Seffi. Asymmetric k-center is log* n-hard to approximate. *Journal of the ACM (JACM)*, 52(4):538–551, 2005.

Cooper, Gregory F. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2):393–405, 1990.

Dyer, Martin E and Frieze, Alan M. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.

Edmonds, Jack and Fulkerson, Delbert Ray. Bottleneck extrema. *Journal of Combinatorial Theory*, 8(3):299–306, 1970.

Frey, Brendan J and Dueck, Delbert. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

Fulkerson, DR. Flow networks and combinatorial operations research. *The American Mathematical Monthly*, 73(2):115–138, 1966.

Gail, Mitchell H, Lubin, Jay H, and Rubinstein, Lawrence V. Likelihood calculations for matched case-control studies and survival studies with tied death times. *Biometrika*, 68(3):703–707, 1981.

Garfinkel, Robert S. An improved algorithm for the bottleneck assignment problem. *Operations Research*, pp. 1747–1751, 1971.

Givoni, Inmar E and Frey, Brendan J. A binary variable model for affinity propagation. *Neural computation*, 21(6):1589–1600, 2009.

Gonzalez, Teofilo F. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

Gross, O. The bottleneck assignment problem. Technical report, DTIC Document, 1959.

Gupta, Rahul, Diwan, Ajit A, and Sarawagi, Sunita. Efficient inference with cardinality-based clique potentials. In *Proceedings of the 24th international conference on Machine learning*, pp. 329–336. ACM, 2007.

Hochbaum, Dorit S and Shmoys, David B. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM (JACM)*, 33(3):533–550, 1986.

Huang, Bert and Jebara, Tony. Approximating the permanent with belief propagation. *arXiv preprint arXiv:0908.1769*, 2009.

Ibrahimi, Morteza, Javanmard, Adel, Kanoria, Yashodhan, and Montanari, Andrea. Robust max-product belief propagation. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 43–49. IEEE, 2011.

Kabadi, Santosh N and Punnen, Abraham P. The bottleneck tsp. In *The Traveling Salesman Problem and Its Variations*, pp. 697–735. Springer, 2004.

Kautz, Henry A, Sabharwal, Ashish, and Selman, Bart. Incomplete algorithms., 2009.

Kearns, Michael, Littman, Michael L, and Singh, Satinder. Graphical models for game theory. In *Proceedings of the Seventeenth conference on UAI*, pp. 253–260. Morgan Kaufmann Publishers Inc., 2001.

Khuller, Samir and Sussmann, Yoram J. The capacitated k-center problem. *SIAM Journal on Discrete Mathematics*, 13(3):403–418, 2000.

Kschischang, FR and Frey, BJ. Factor graphs and the sum-product algorithm. *Information Theory, IEEE*, 2001.

Litsyn, Simon, M., Rains E., and A., Sloane N. J. The table of nonlinear binary codes. `http://www.eng.tau.ac.il/~litsyn/tableand/`, 1999. [Online; accessed Jan 30 2014].

Masuyama, Shigeru, Ibaraki, Toshihide, and Hasegawa, Toshiharu. The computational complexity of the m-center problems on the plane. *IEICE TRANSACTIONS (1976-1990)*, 64(2):57–64, 1981.

Mezard, M., Parisi, G., and Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582): 812–815, August 2002. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1073287.

Panigrahy, Rina and Vishwanathan, Sundar. An o(log*n) approximation algorithm for the asymmetric p-center problem. *Journal of Algorithms*, 27(2):259–268, 1998.

Parker, R Gary and Rardin, Ronald L. Guaranteed performance heuristics for the bottleneck travelling salesman problem. *Operations Research Letters*, 2(6):269–272, 1984.

Potetz, Brian and Lee, Tai Sing. Efficient belief propagation for higher-order cliques using linear constraint nodes. *Computer Vision and Image Understanding*, 112(1):39–54, 2008.

Ramezanpour, Abolfazl and Zecchina, Riccardo. Cavity approach to sphere packing in hamming space. *Physical Review E*, 85(2): 021106, 2012.

Ravanbakhsh, Siamak and Greiner, Russell. Perturbed message passing for constraint satisfaction problems. *arXiv preprint arXiv:1401.6686*, 2014.

Schrijver, Alexander. *Min-max results in combinatorial optimization*. Springer, 1983.

Tarlow, Daniel, Givoni, Inmar E, and Zemel, Richard S. Hopmap: Efficient message passing with high order potentials. In *International Conference on Artificial Intelligence and Statistics*, pp. 812–819, 2010.

Tarlow, Daniel, Swersky, Kevin, Zemel, Richard S, Adams, Ryan Prescott, and Frey, Brendan J. Fast exact inference for recursive cardinality models. *arXiv preprint arXiv:1210.4899*, 2012.

A. Proofs

Proof 2.1 *(A) μ_y for $y \geq y^*$ is satisfiable:* It is enough to show that for any $y \geq y^*$, $\mu_y(x^*) > 0$. But since

$$\mu_y(x^*) \quad = \quad \frac{1}{Z_y} \prod_I \mathbb{I}(f_{\mathbb{I}}(x_I^*) \leq y)$$

and $f_I(x_I^*) \leq y^* \leq y$, all the indicator functions on the rhs evaluate to 1, showing that $\mu_y(x^*) > 0$.

(B) μ_y for $y < y^*$ is not satisfiable: Towards a contradiction assume that for some $\underline{y} < y^*$, $\mu_{\underline{y}}$ is satisfiable. Let $\underline{x}$ denote a satisfying assignment –*i.e.*, $\mu_{\underline{y}}(\underline{x}) > 0$. Using the definition of μ_y-reduction, this implies that $\mathbb{I}(f_I(\underline{x}_I) \leq \underline{y}) > 0$ for all $I \in \mathcal{F}$. However this means that $\max_I f_I(\underline{x}_I) \leq \underline{y} < y^*$, which means y^* is not the min-max value.

Proof 2.2 Assume they have different min-max assignments[7] – *i.e.*, $x^* = \arg_x \min \max_I f_I(x_I)$, $x'^* = \arg_x \min \max_I f'_I(x_I)$ and $x^* \neq x'^*$. Let y^* and y'^* denote the corresponding min-max values.

Claim A.1

$$y^* > \max_I f_I(x_I'^*) \quad \Leftrightarrow \quad y'^* < \max_I f'_I(x_I^*)$$
$$y^* < \max_I f_I(x_I'^*) \quad \Leftrightarrow \quad y'^* > \max_I f'_I(x_I^*)$$

This simply follows from the condition of Lemma 2.2. But in each case above, one of the assignments y^* or y'^* is not an optimal min-max assignment as an alternative assignment has a lower maximum over all factors.

Proof 2.3 First note that since $g(x) = 2^z$ is a monotonically increasing function, the rank of elements in the range of $\{f'_I\}_I$ is the same as their rank in the range of $\{f_I\}_I$. Using Lemma 2.2, this means

$$\arg_x \min \max_I f'_I(x_I) \quad = \quad \arg_x \min \max_I f_I(x_I). \qquad (9)$$

Since $2^z > \sum_{l=0}^{z-1} 2^l$, by definition of $\{f'_I\}$ we have

$$\max_{I \in \mathcal{F}} f'_I(x_I) > \sum_{I \in \mathcal{F} \setminus I^*} f'_I(x_I) \quad \text{where} \quad I^* = \arg_I \max f'_I(x_I)$$

It follows that for $x^1, x^2 \in \mathcal{X}$,

$$\max_I f'_I(x_I^1) < \max_I f'_I(x_I^2) \quad \Leftrightarrow \quad \sum_I f'_I(x_I^1) < \sum_I f'_I(x_I^2)$$

Therefore

$$\arg_x \min \max_I f'_I(x_I) \quad = \quad \arg_x \min \sum_I f'_I(x_I).$$

This equality, combined with eq. (9), prove the statement of the theorem.

[7]For simplicity, we are assuming each instance has a single min-max assignment. In case of multiple assignments there is a one-to-one correspondence between them. Here one starts with the assumption that there is an assignment x^* for the first factor-graph that is different from all min-max assignments in the second factor-graph.

Proof 3.1 Recall the definition $Z_y \triangleq \sum_{\mathcal{X}} \prod_I \mathbb{I}(f_I(x_I) \leq y)$. For $y_1 < y_2$ we have:

$$\begin{aligned}
&f_I(x_I) \leq y_1 \quad \rightarrow \quad f_I(x_I) \leq y_2 && \Rightarrow\\
&\mathbb{I}(f_I(x_I) \leq y_1) \leq \mathbb{I}(f_I(x_I) \leq y_2) && \Rightarrow\\
&\sum_{\mathcal{X}} \prod_I \mathbb{I}(f_I(x_I) \leq y_1) \leq \sum_{\mathcal{X}} \prod_I \mathbb{I}(f_I(x_I) \leq y_2) && \Rightarrow\\
&Z_{y_1} \leq Z_{y_2}
\end{aligned}$$

Proof 4.1 First note that μ_y defines a uniform distribution over its support as its unnormalized value is only zero or one.

(A) *Every K-clique-cover over $\mathcal{G}(D, y)$, defines K unique assignments x with $\mu_y(x) > 0$:* Let $x \in \mathcal{X}$ denote a K-clique-cover over $\mathcal{G}$, such that $x_i \in \{1, \ldots, K\}$ indicates the clique assignment of node $i \in \mathcal{V}$. Since all the permutations of cliques produces a unique assignment x, there are K assignments per K-clique-cover. For any clique-cover of $G(D, y)$, i and j are connected only if $D_{i,j} \leq y$. Therefore two nodes can belong to the same clique $\mathcal{C}_k$ only if $D_{i,j} \leq y$. On the other hand, the μ_y-reduction of the Potts factor for min-max clustering is

$$f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i \neq x_j) + \mathbb{I}(x_i = x_j \wedge D_{i,j} \leq y) \qquad (10)$$

Here either i and j belong to different cliques where $f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i \neq x_j) > 0$ or they belong to the same clique in the clique-cover x: $f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i = x_j \wedge D_{i,j} \leq y) > 0$. Therefore for any such x, $\mu_y(x) > 0$.

(B) *Every x for which $\mu_y(x) > 0$, defines a unique K-clique-cover over $\mathcal{G}(D, y)$:* Let i and j belong to the same cluster *iff* $x_i = x_j$. $\mu_y(x) > 0$ implies all its factors as given by eq. (10) have non-zero values. Therefore, for every pair of nodes i and j, either they reside in different clusters (*i.e.*, $x_i \neq x_j$), or $D_{i,j} \leq y$, which means they are connected in $\mathcal{G}(D, y)$. However if all the nodes in the same cluster are connected in $\mathcal{G}(D, y)$, they also form a clique. Since $\forall i \;\; x_i \in \{1, \ldots, K\}$ can take at most K distinct values, every assignment x with $\mu_y(x) > 0$ is a K-clique-cover.

Proof 4.2 Here we prove Proposition 4.2 for the factor-graph of Section 4.2.2. The proof for the binary variable model follows the same procedure. Since μ_y defines a uniform distribution over its support, it is enough to show that any clique of size K over $\mathcal{G}(-D, -y)$ defines a unique set of assignments all of which have nonzero probability ($\mu_y(x) > 0$) and any assignment x with $\mu_y(x) > 0$ defines a unique clique of size at least K on $\mathcal{G}(-D, -y)$. First note that the basic difference between $\mathcal{G}(D, y)$ and $\mathcal{G}(-D, -y)$ is that in the former all nodes that are connected have a distance of at most y while in the later all nodes that have a distance of at least y are connected to each other. Consider the μ_y-reduction of the pairwise factors of the factor-graph defined in Section 4.2.2:

$$f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i \neq x_j \wedge -D_{x_i,x_j} \leq y) \qquad (11)$$

(A) Any clique of size K in $\mathcal{G}(-D, -y)$, defines K unique assignments, such that for any such assignment x, $\mu_y(x) > 0$: For a clique $\mathcal{C} = \{c_1, \ldots, c_K\}$ of size K, define $x_i = c_{\pi(i)}$, where $\pi : \{1, \ldots, K\} \rightarrow \{1, \ldots, K\}$ is a permutation of nodes in clique $\mathcal{C}$. Since there are K such permutations we may define as many assignments x. Now consider one such assignment x. For every two nodes x_i and x_j, since they belong to the clique $\mathcal{C}$ over $\mathcal{G}(-D, -y)$, they are connected and $D_{x_i,x_j} \geq y$. This

means that all the pairwise factors defined by eq. (11) have non-zero values and therefore $\mu_y(x) > 0$.

(B) Any assignment x with $\mu_y(x) > 0$ corresponds to a unique clique of size K in $\mathcal{G}(-D, -y)$: Let $\mathcal{C} = \{x_1, \ldots, x_K\}$. Since $\mu_y(x) > 0$, all pairwise factors defined by eq. (11) are non-zero. Therefore $\forall i, j \neq i \;\; D_{x_i x_j} \geq y$, which means all x_i and x_j are connected in $\mathcal{G}(-D, -y)$, forming a clique of size K.

Proof 4.3 The μ_y-reduction of the factors of factor-graph of Section 4.3 are:

A. *local factors* $\forall i \neq j \quad f^y_{\{i-j\}}(x_{i-j}) = \mathbb{I}(D_{i,j} \leq y \vee x_{i-j} = 0)$.

B. *uniqueness factors* For every i consider $I = \{i - j \mid 1 \leq j \leq N\}$, then $f^y_I(x_I) = \mathbb{I}(\sum_{i-j \in \partial I} x_{i-j} = 1)$.

C. *consistency factors* $\forall j, i \neq j$, $f^y(x_{\{j-j, i-j\}}) = \mathbb{I}(x_{j-j} = 0 \wedge x_{i-j} = 1)$.

D. *K-of-N factor* Let $\mathcal{K} = \{i-i \mid 1 \leq i \leq N\}$, then $f^y_{\mathcal{K}}(x_{\mathcal{K}}) = \mathbb{I}(\sum_{i-i \in \mathcal{K}} x_{i-i} = K)$.

(A) Any assignment x with $\mu_y(x) > 0$ defines a K-dominating set of $\mathcal{G}(D, y)$: Since $\mu_y(x) > 0$, all the factors above have a non-zero value for x. Let $\mathcal{D} = \{i \mid x_{i-i} = 1\}$.

Claim A.2 *$\mathcal{D}$ defines a K-dominating set of graph $\mathcal{G}$.*

The reason is that first (a) Since the K-of-N factor is nonzero $|D| = K$. (b) For any node $j \in \mathcal{V} \setminus \mathcal{D}$, the uniqueness factors and consistency factors, ensure that they are associated with a node $i \in \mathcal{D}$ – $\exists i \in \mathcal{D} \mid x_{j,i} = 1$. (c) Local factors ensure that if $x_{j,i} = 1$ then $D_{j,i} \leq y$, therefore i and j are connected in the neighborhood graph –*i.e.*, $(i, j) \in \mathcal{E}$. (a), (b) and (c) together show that if all factors above are non-zero, x defines a K-dominating set for $\mathcal{G}$.

(B) Any K-dominating set of $\mathcal{G}(D, y)$ defines an x with nonzero probability $\mu_y(x)$: Define x as follows: For all $i \in \mathcal{D}$ set $x_{i-i} = 1$. For any j with $x_{j-j} = 1$, select and $(j, i) \in \mathcal{E}$ where $x_{i-i} = 1$ and set $x_{j-i} = 1$. Since $\mathcal{D}$ is a dominating set, the existence of such an edge is guaranteed. It is easy to show that for an assignment x constructed this way, all μ_y-reduced factors above are non-zero and therefore $\mu_y(x) > 0$.

Proof 4.4 Here we re-enumerate the μ_y-reduction of factors for factor-graph of Section 4.3.

A. *local factors* $\forall i \neq j \quad f^y_{\{i-j\}}(x_{i-j}) = \mathbb{I}(D_{i,j} \leq y \vee x_{i-j} = 0)$.

B. *uniqueness factors* For every i consider $I = \{i - j \mid 1 \leq j \leq N\}$, then $f^y_I(x_I) = \mathbb{I}(\sum_{i-j \in \partial I} x_{i-j} = 1)$.

C. *consistency factors* $\forall j, i \neq j$, $f^y(x_{\{j-j, i-j\}}) = \mathbb{I}(x_{j-j} = 0 \wedge x_{i-j} = 1)$.

D. *K-of-N factor* Let $\mathcal{K} = \{i-i \mid 1 \leq i \leq N\}$, then $f^y_{\mathcal{K}}(x_{\mathcal{K}}) = \mathbb{I}(\sum_{i-i \in \mathcal{K}} x_{i-i} = K)$.

(A) Any assignment x with $\mu_y(x) > 0$ defines an induced K-set-cover for $\mathcal{G}$: Let $\mathcal{C} = \{\mathcal{V}_i \mid x_{i-i} = 1\}$, where $\mathcal{V}_i = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$, as in the definition of an induced set-cover. Note that $\mathcal{E}$ refers to the edge-set of $\mathcal{G}(D, y)$.

Claim A.3 *$\mathcal{C}$ defines an induced K-set-cover for $\mathcal{G}(D, y)$.*

First note that $\mathcal{C}$ as defined here is a subset of $\mathcal{S}$ as defined in the definition of an induced set-cover. Since $\mu_y(x) > 0$, all the μ_y-reduced factors above have a non-zero value for x. (a) Since the K-of-N factor is nonzero, by definition of $\mathcal{C}$ above, $|\mathcal{C}| = K$. (b) Since uniqueness factors are non-zero for every node i, either $x_{i-i} = 1$, in which case i is covered by $\mathcal{V}_i \in \mathcal{C}$, or $x_{i-j} = 1$ for some $j \neq i$, in which case non-zero consistency factors imply that $\mathcal{V}_j \in \mathcal{C}$. (c) It only remains to show that if $x_{i-j} = 1$, then $(i - j) \in \mathcal{E}$. The non-zero local factors imply that for every $x_{i-j} = 1$, $D_{i-j} \leq y$. However by definition of $\mathcal{G}(D, y)$, this also means that $(i, j) \in \mathcal{E}$. Therefore for any assignment x with $\mu_y(x) > 0$, we can define a unique $\mathcal{C}$ (an induced K-set-cover for $\mathcal{G}(D, y)$).

(B) Any induced K-set-cover for $\mathcal{G}$, defines an assignment x with , $\mu_y(x) > 0$: Here we need to build an assignment x from an induced K-set-cover $\mathcal{C}$ and show that all the factors in μ_y-reduction, have non-zero value and therefore $\mu_y(x) > 0$. To this end, for each $\mathcal{V}_i \in \mathcal{C}$ set $x_{i-i} = 1$. Since $|\mathcal{C}| = K$, the K-of-N factor above will have a non-zero value. For any node j such that $x_{j-j} = 0$, select a cover $\mathcal{V}_i$ and set $x_{j-i} = 1$. Since $\mathcal{C}$ is a set-cover the existence of at least one such $\mathcal{V}_i$ is guaranteed. Since we have selected only a single cover for each node, the uniqueness factor is non-zero. Also $x_{j-i} = 1$ only if $\mathcal{V}_i$ is a cover (and therefore $x_{i-i} = 1$), the consistency factors are non-zero. Finally since $\mathcal{C}$ is an induced cover for $G(D, y)$, for any $j \in \mathcal{V}_i$, $D_{j,i} \leq y$ and therefore $x_{j,i} = 1 \;\Rightarrow D_{j,i} \;\leq y$. This ensures that local factors are non-zero. Since all factors in the μ_y-reduced factor-graph are non-zero, $\mu_y(x) > 0$.

Proof 4.5 First note that μ_y defines a uniform distribution over its support as its unnormalized value is only zero or one. Here w.l.o.g we distinguish between two Hamiltonian cycles that have a different starting point but otherwise represent the same tour. Consider the p_y-reduction of the pairwise factor of eq. (8)

$$f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(f_{\{i,j\}}(x_i, x_j) \leq y) \tag{12}$$

$$= \mathbb{I}(|x_i - x_j| > 1) + \; \mathbb{I}(x_i = x_j - 1 \wedge D_{i,j} \leq y) \tag{13}$$

$$+ \, \mathbb{I}(x_i = x_j + 1 \wedge D_{j,i} \leq y) \tag{14}$$

(A) *Every Hamiltonian cycle over $\mathcal{G}(D, y)$, defines a unique assignments x with $\mu_y(x) > 0$:* Given the Hamiltonian cycle $H = h_0, h_2, \ldots, h_{N-1}$ where $h_i \in \{1, \ldots, N\}$ is the i^{th} node in the path, for each i define $x_i = j \;\, s.t. \;\, h_j = i$. Now we show that all pairwise factors of eq. (12) are non-zero for x. Consider two variables x_i and x_j. If they are not consecutive in the Hamiltonian cycle then $f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(|x_i - x_j| > 1) > 0$. Now w.l.o.g. assume i and j are consecutive and x_i appears before x_j. This means $(i, j) \in \mathcal{E}$ and therefore $D_{i,j} \leq y$, which in turn means $f^y_{\{i,j\}}(x_i, x_j) = \mathbb{I}(x_i = x_j - 1 \wedge D_{i,j} \leq y) > 0$ Since all pairwise factors are non-zero, $\mu_y(x) > 0$.

(B) *Every x for which $\mu_y(x) > 0$, defines a unique Hamiltonian path over $\mathcal{G}(D, y)$:* Given assignment x, construct $H = h_0, \ldots, h_{N-1}$ where $h_i = j \;\, s.t. x_j = i$. Now we show that if $\mu(x) > 0$, H defines a Hamiltonian path. If $\mu(x) > 0$, for every two variables x_i and x_i, one of the indicator functions of eq. (12) should evaluate to one. This means that first of all, $x_i \neq x_j$ for $i \neq j$, which implies H is well-defined and $h_i \neq h_j$ for $i \neq j$. Since all $x_i \in \{0, \ldots, N-1\}$ values are distinct, for each $x_i = s$ there are two variables $x_j = s - 1$ and $x_k = s + 1$ (recall that we are using modular arithmetic) for which the pairwise factor of eq. (12) is non-zero. This means $D_{j,i} \leq y$ and $D_{i,k} \leq y$ and therefore $(j,i),(i,k) \in \mathcal{E}$ (the edge-set of $\mathcal{G}(D,y)$). But by definition of H, $h_s = i$, $h_{s-1} = j$ and $h_{s+1} = k$ are consecutive nodes in H and therefore H is a Hamiltonian path.

B. Efficient Messages Passing

B.1. K-of-N Factors

For binary variables, it is convenient to assume all variable-to-factor messages are normalized such that $\nu_{j\to I}(0) = 1$. Now consider the task of calculating $\nu_{I\to i}(0)$ and $\nu_{I\to i}(1)$ for at-least-K-of-N factors. Let $\mathcal{A}(K)$ denote the subsets of $\mathcal{A}$ with at least K members. Then

$$\nu_{I\to i}(0) \quad = \quad \sum_{\mathcal{X}_{\partial I\setminus i}} \mathbb{I}(\sum_{j\in\partial I\setminus i} x_j \geq K) \prod \nu_{j\to I}(x_j) = \sum_{(\partial I\setminus i)(K)} \prod_{j\in(\partial I\setminus i)(K)} \nu_{j\to I}(1) \tag{15}$$

where in deriving eq. (15) we have used the fact that $\nu_{j\to I}(0) = 1$. To calculate $\nu_{I\to i}(1)$ we follow the same procedure, except that here the factor is replaced by $\mathbb{I}(\sum_{j\in\partial I\setminus i} x_j \geq K-1)$. This is because here we assume $x_i = 1$ and therefore it is sufficient for $K-1$ other variables to be nonzero.

To evaluate eq. (15), we use the dynamic programming recursion where another variable x_l for some $l \in \partial I$ is either zero or one:

$$\sum_{(\partial I\setminus i)(K)} \prod_{j\in(\partial I\setminus i)(K)} \nu_{j\to I}(1) = \sum_{(\partial I\setminus i,l)(K)} \prod_{j\in(\partial I\setminus i,l)(K)} \nu_{j\to I}(1) + \nu_{l\to I}(1) \sum_{(\partial I\setminus i,l)(K)-1} \prod_{j\in(\partial I\setminus i,l)(K-1)} \nu_{j\to I}(1)$$

This allows us to calculate these messages in $\mathcal{O}(NK)$.

B.2. Bottleneck TSP Factors

The μ_y-reduction of the min-max factors of BTSP is given by:

$$f^y_{\{i,j\}}(x_i,x_j) = \mathbb{I}(f_{\{i,j\}}(x_i,x_j) \leq y) \tag{16}$$
$$= \mathbb{I}(|x_i - x_j| > 1) + \mathbb{I}(x_i = x_j - 1 \wedge D_{i,j} \leq y) \tag{17}$$
$$+ \mathbb{I}(x_i = x_j + 1 \wedge D_{j,i} \leq y) \tag{18}$$

The matrix-form of this factor (depending on the order of $D_{i,j}$, $D_{j,i}, y$) takes several forms all of which are band-limited. Assuming the variable-to-factor messages are normalized (*i.e.*, $\sum_{x_i} \nu_{j\to I}(x_i) = 1$) the factor-to-variable message is given by

$$\nu_{\{i,j\}\to i}(x_i) = 1 - \nu_{j\to\{i,j\}}(x_i) + \mathbb{I}(D_{i,j} \leq y)(1 - \nu_{j\to\{i,j\}}(x_i - 1)) + \mathbb{I}(D_{j,i} \leq y)(1 - \nu_{j\to\{i,j\}}(x_i + 1))$$

C. Analysis of Complexity

Here for different factor-graphs used in the Section 4, we provide a short analysis of complexity.

Min-max Clustering:
This formulation includes N^2 pairwise factors – one for every pair of variables – and the cost of sending each factor to variable message is $\mathcal{O}(K)$, resulting in $\mathcal{O}(N^2K)$ complexity for all factor-to-variable messages. The cost of sending variable-to-factor messages is also $\mathcal{O}(N^2K)$. This gives $\mathcal{O}(N^2K\log(N))$ cost for finding the approximate min-max solution. The $\log(N)$ factor reflects the complexity of binary search, which depends on the diversity of range of factors –*i.e.*, $\log(|\mathcal{Y}|) = \log(N^2) = 2\log(N)$.

K-Packing (binary variable):
The time complexity of Perturbed BP iterations in this factor-graph is dominated by factor-to-variable messages sent from $f_{\mathcal{K}}(x)$ to all variables. Assuming asynchronous update, the complexity for min-max procedure is $\mathcal{O}(N^2K\log(N))$.

K-Packing (categorical variable):
Since the factors are not sparse, the complexity of factor-to-variable messages are $\mathcal{O}(N^2)$, resulting in $\mathcal{O}(N^2K^2)$ cost per iteration for μ_y-reduction. Since the diversity of pairwise distances is $|\mathcal{Y}| = \mathcal{O}(N^2)$, the general cost of finding an approximate min-max solution by message passing is $\mathcal{O}(N^2K^2\log(N))$.

Sphere-Packing (Hamming distance):
The total number of variables (including x and z) in this factor-graph is $N = Kn + \frac{K(K-1)}{2}n = \mathcal{O}(nK^2)$. The factor-graph contains $\frac{K(K-1)}{2}n$ z-factors and $\frac{K(K-1)}{2}$ distance factors. Each x_{i-k} variable receives messages from $K-1$ z-factors while each $z(i,j)_k$ variable is connected to one z-factor and one distance factor. Assuming an asynchronous message update the cost of factor-to-variable messages from distance-factors (*i.e.*, $\mathcal{O}(n^2y)$) dominates the complexity of Perturbed BP's iterations. Since there are K^2 such factors the complexity of message passing per iteration is $\mathcal{O}(K^2n^2y)$.

K-Center Problem:
Assuming an asynchronous update schedule, the cost of sending messages from uniqueness factors dominates the complexity of Perturbed BP's iteration resulting in $\mathcal{O}(N^3)$ complexity for μ_y-reduction and $\mathcal{O}(N^3\log(N)$ for the min-max problem.

Bottleneck Traveling Salesman Problem:
In Section B.2 we show that the μ_y-reduction of BTSP factors of Figure 6 allows efficient marginalization in $\mathcal{O}(N)$. Since there are N^2 factors in this factor-graph and the cost sending factor-to-variable messages is $\mathcal{O}(N)$, the general cost of min-max inference is $\mathcal{O}(N^3\log(N))$.