Factor Graph Approach to Distributed Facility Location in Large-Scale Networks

Hung Q. Ngo
Department of Computer Engineering
Kyung Hee University
South Korea, 446-701
Email: nqhung@oslab.khu.ac.kr

Sungyoung Lee
Department of Computer Engineering
Kyung Hee University
South Korea, 446-701
Email: sylee@oslab.khu.ac.kr

Young-Koo Lee
Department of Computer Engineering
Kyung Hee University
South Korea, 446-701
Email: yklee@khu.ac.kr

***Abstract*— In this paper, we present a new approach to solving the distributed facility location problem using the recent modeling and computational methodology of factor graph and message-passing. We first formulate the problem as finding a valid network configuration that minimizes the overall cost. We then represent the problem using a factor graph, and derive simplified, localized, broadcast-based message-passing rules which can elect a near-optimal set of facility nodes in a few iterations. Simulation results for small-world network topologies show that the algorithm is able to achieve good convergence rate and approximation ratio, and scalable to the network size.**

I. Introduction

Facility Location Theory [1] can be used to model a wide range of important application scenarios in large-scale networks concerning the placement of service facilities so as to minimize the overall service costs. For instance, an Internet service provider (ISP) needs to dynamically setup a set of servers or placing caches at dedicated or virtual hosts [2]. On the one hand, every client demands to have access to the servers as close as possible so as to minimize the transmission delay. On the other hand, setting up a server at a hosting autonomous system incurs overhead, bandwidth used to carry the traffic, and maintenance costs, for which that particular host may charge the ISP. In this context, the facility location problem precisely captures the resulting trade-off between the cost (i.e. the number of servers to be installed, as well as their locations) and the efficiency of deploying such services.

The facility location problem can also be applied to wireless sensor networks (WSNs), typically consisting of thousands of tiny nodes deployed to collect and transmit sensing data to external observers. Sensor clustering has been shown to be an effective approach to hierarchically organizing network topology for a wide range of applications [3]. Having a few cluster-heads (CHs) is desirable to maximize the fusion ratio and minimize the relay cost. However, this leads to the increase in transmission power for intra- and inter-cluster communications, since the distances between CHs, and beween each CH and their associated nodes are increased also. By applying the distributed facility location algorithm, we can select the best candidates to act as CHs so as to minimize the overall energy consumption and prolong the network lifetime.

For general network models, it is known that facility location is an NP-hard problem. There are a variety of algorithms with guaranteed constant approximations in the literature. However, they are inherently centralized, and their time and message complexity are not efficient for distributed realization in large-scale networks.

In this paper, we present a simple and highly localized message-broadcast protocol for solving the distributed facility location (DFL) problem. Using behavioral modeling, we first reformulate the DFL problem as finding a valid network configuration that minimizes the overall cost in Section II. We then use the recent modeling and computational methodology of factor graph and message-passing [4], which is very convenient for distributed realization in large-scale networks, to model and solve the formulated min-sum problem in Section III. We provide message simplification, then devise a broadcast version of the algorithm in Section IV, to further reduce the number of message transmissions, resulting in a simpler and more efficient algorithm. Simulation results presented in Section V show that the algorithm is able to achieve good convergence rate and approximation ratio for different network sizes. We give concluding remarks and future extensions in Section VI.

II. Behavioral Modeling for Facility Location Problem

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected undirected graph representing a network defined by a set of nodes $\mathcal{V} = \{1, 2, ..., N\}$ and a set of edges $\mathcal{E}$. For each node $i \in \mathcal{V}$, we assign a weight f_i representing the facility cost for deploying the service at node i, and a weight c_{ij} representing the connection cost for accessing the service at some facility $j \in \mathcal{N}(i)$. Here we denote by $\mathcal{N}(i)$ the *open* neighbor set of i, which includes all nodes within k-hop from i, with k given depending on the scope of the applications. We further define the *closed* neighbor set of i as $\mathcal{N}[i] := \mathcal{N}(i) \bigcup \{i\}$.

The objective of the facility location problem is to select a nonempty subset of nodes $\Omega \subseteq \mathcal{V}$ to act as facilities so as to minimize the overall joint-cost $C(\Omega)$ of deploying the facilities and connecting the clients:

$$C(\Omega) = \sum_{\forall i \in \Omega} f_i + \sum_{\forall i \in \mathcal{V} \setminus \Omega} c_{i\sigma(i)}$$

where $\sigma(i) \in \mathcal{N}(i)$ is the facility that is closer to i. For general network models, it is known that facility location is an NP-hard problem. To the best of our knowledge, there are no distributed algorithms that can solve this problem efficiently.

In this paper we formulate the facility location problem to be a *min-sum labeling*: For each node $i \in \mathcal{V}$, assign a label x_i containing the identity (ID) of the associated facility node, i.e. $x_i \in \mathcal{N}[i]$, in such a way that the total cost in the network is minimized. The cost of each node i now depends on its label x_i, or equivalently, its role in the network:

$$e_i(x_i) = \begin{cases} f_i & \text{if } x_i = i \\ c_{ij} & \text{if } x_i = j \in \mathcal{N}(i) \end{cases}$$

Let $\mathcal{X} := \{x_i : i \in \mathcal{V}\}$ be the set of N labels of the network. Let the n-tuple $\mathbf{x} := (x_1, x_2, ..., x_N)$ denote the *configuration* (or *labeling*) of the whole network. Since the system is specified via its configuration, this approach is known as *behavioral modeling*; and $\mathbf{x}$ can be a valid or *invalid* configuration. For instance, consider a neighbor j of i. If node j selects i as its facility node (i.e., $x_j = i$), while node i itself is not correctly labeled as a facility node (e.g., $x_i = k \neq i$), then this is an invalid configuration. We further denote by $\mathbf{x}_{\mathcal{N}[i]}$ the configuration of nodes in closed neighbor set of i, i.e., $\mathbf{x}_{\mathcal{N}[i]} := (x_j : j \in \mathcal{N}[i])$. We use the constraint function $\theta_i(\mathbf{x}_{\mathcal{N}[i]})$ to enforce valid configurations between the label x_i of node i and the labels of its neighboring nodes. For the min-sum configuration (min-sum semiring [5]) problem, the constraint function gives a penalty of 0 or $+\infty$ for a valid or invalid configuration respectively, defined as follows:

$$\theta_i(\mathbf{x}_{\mathcal{N}[i]}) := \begin{cases} +\infty, & \text{if } x_i \neq i \text{ but } \exists j \in \mathcal{N}(i) : x_j = i \\ 0, & \text{otherwise} \end{cases}$$

Clearly, the configuration of the whole network is valid if and only if, for each and every node $i \in \mathcal{V}$, the configuration of its closed neighbor set is a valid one. The facility location now becomes the problem of finding a min-sum configuration of the network among the valid ones, defined as:

$$\mathbf{x}_{opt} := \arg\min_{\mathbf{x}} \left[\sum_{i \in \mathcal{V}} e_i(x_i) + \sum_{i \in \mathcal{V}} \theta_i(\mathbf{x}_{\mathcal{N}[i]}) \right] \quad (1)$$

To solve this global minimization problem, we use the recent modeling and computational methodology of factor graph and message-passing [4]. A Factor graph can be used to represent a complicated global function, which can be factored into simpler "local" functions, each of which depends on a subset of the variables. In a factor graph, message-passing algorithms can compute, either exactly or approximately, various function marginalization and maximization using simple message passing rules.

III. Factor Graph and Message-Passing Rules

To derive the factor graph $\mathcal{F}$ for the min-sum constraint satisfaction problem given in Eq. (1), we represent each label x_i by a (hidden) variable vertex, each cost function $e_i(x_i)$ by a factor e_i connected to variable vertex x_i, and each constraint $\theta_i(\mathbf{x}_{\mathcal{N}[i]})$ by a factor θ_i connected to the involved variables which are components of $\mathbf{x}_{\mathcal{N}[i]}$. We denote the set of constraint factors connected to x_i as $\Theta(x_i)$, and this set excluding θ_j as $\Theta(x_i)\backslash\theta_j$.

By applying the Min-Sum algorithm on $\mathcal{F}$, we derive the standard message-passing algorithm (MEPA), with messages exchanged between constraint factors and variable vertices defined recursively:

$$\mu^t_{x_i \to \theta_j}(x_i) = e_i(x_i) + \sum_{\theta_k \in \Theta(x_i)\backslash\theta_j} \mu^{t-1}_{\theta_k \to x_i}(x_i) \quad (2)$$

$$\mu^t_{\theta_i \to x_j}(x_j) = \min_{\mathbf{x}_{\mathcal{N}[i]\backslash j}} \left[\begin{array}{c} \theta_i(\mathbf{x}_{\mathcal{N}[j]}|x_j) \\ + \sum_{k \in \mathcal{N}[i]\backslash j} \mu^{t-1}_{x_k \to \theta_i}(x_k) \end{array} \right] \quad (3)$$

At any time instant t, each node x_i can evaluate the optimal label x_i^{opt} by summing up all received messages:

$$x_i^{opt} = \arg\min_{x_i} \left[e_i(x_i) + \sum_{\theta_k \in \Theta(x_i)} \mu^{t-1}_{\theta_k \to x_i}(x_i) \right] \quad (4)$$

In the standard form of messages above, each message is a vector of length $|\mathcal{N}[i]|$. In the sequel we provide a message simplification, inspired from [6], in which each message becomes a scalar, making the derived message passing rules very simple and efficient for practical implementation. The proofs are based on the derivation of data clustering algorithm in [6], which can be considered as a special case of the facility location problem with a *fully* connected graph, i.e., $\mathcal{N}[i] = \mathcal{V}, \forall i \in \mathcal{V}$. We start with the following important observation:

Observation 1: $\forall x_j \neq i$, the elements of vector message $\mu^t_{\theta_i \to x_j}$ in Eq. (3) are identical.

Proof: The observation follows by analyzing in detail the vector message $\mu^t_{\theta_i \to x_j}(x_j | x_j \neq i)$ in two cases: $i = j$ and $i \neq j$. First, if $i = j$ (or $\theta_i \equiv \theta_j$), meaning that node j is not a facility node (since $x_j \neq j$), then the constraint $\theta_i(\mathbf{x}_{\mathcal{N}[i]}|x_j \neq j)$ guarantees that the neighbors of j must not take j as their facility node, or equivalently, $x_k \neq j, \forall k \in \mathcal{N}(j)$. Thus Eq. (3) becomes

$$\mu^t_{\theta_i \to x_j}(x_j \neq i = j) = \sum_{k \in \mathcal{N}(j)} \min_{x_k \neq j} \mu^{t-1}_{x_k \to \theta_j}(x_k) \quad (5)$$

Second, if $i \neq j$ (or $\theta_i \neq \theta_j$), Eq. (3) becomes

$$\mu^t_{\theta_i \to x_j}(x_j \neq i \neq j) = \min \left\{ \begin{array}{l} \mu^{t-1}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i)\backslash j} \min_{x_k} \mu^{t-1}_{x_k \to \theta_i}(x_k), \\ \min_{x_i \neq i} \mu^{t-1}_{x_i \to \theta_i}(x_i) + \sum_{k \in \mathcal{N}(i)\backslash j} \min_{x_k \neq i} \mu^{t-1}_{x_k \to \theta_i}(x_k) \end{array} \right\} \quad (6)$$

Clearly, for both cases, the message $\mu^t_{\theta_i \to x_j}(x_j \mid \forall x_j \neq i)$ does not depend on the values of x_j, and thus they are identical. ■

As a direct result of this observation, we have the following corollaries:

Corrolary 1: The vector message in Eq. (3) can be written as sum of two components, $\mu^t_{\theta_i \to x_j}(x_j) = \widetilde{\mu}_{\theta_i \to x_j}(x_j) + \overline{\mu}_{\theta_i \to x_j}$, which are dependent and independent of x_j respectively, with $\overline{\mu}_{\theta_i \to x_j} = \mu^t_{\theta_i \to x_j}(x_j \neq i)$, and

$$\widetilde{\mu}_{\theta_i \to x_j}(x_j) = \begin{cases} 0, & x_j \neq i \\ \widetilde{\mu}_{\theta_i \to x_j}(i), & x_j = i \end{cases}$$

Corrolary 2:

$$\sum_{\theta_k \in \Theta(x_i)} \widetilde{\mu}_{\theta_k \to x_j}(x_j) = \widetilde{\mu}_{\theta_{x_j} \to x_j}(x_j), \forall x_j \in \mathcal{N}[j] \quad (7)$$

and, similarly:

$$\sum_{\theta_k \in \Theta(x_j) \setminus \theta_i} \widetilde{\mu}_{\theta_k \to x_j}(x_j) = 0, \quad if \quad x_j = i \quad (8)$$

Corollary 2 provides intermediate results used to prove the following lemma:

Lemma 1: To evaluate the optimal label x_{opt} given in Eq. (4), each vector message $\mu^t_{\theta_i \to x_j}$ and $\mu^t_{x_i \to \theta_j}$, can be reduced to a *single*, real-value number.

Proof: Applying the above corollaries, we have:

$$\begin{aligned} x_i^{opt} &= \arg\min_{x_i} \left[e_i(x_i) + \sum_{\theta_k \in \Theta(x_i)} \mu^{t-1}_{\theta_k \to x_i}(x_i) \right] \\ &= \arg\min_{x_i} \left[e_i(x_i) + \sum_{\theta_k \in \Theta(x_i)} \widetilde{\mu}_{\theta_k \to x_i}(x_i) \right] \\ &= \arg\min_{x_i} \left[e_i(x_i) + \widetilde{\mu}_{\theta_{x_i} \to x_i}(x_i) \right] \end{aligned}$$

Hence, each factor θ_j needs to send a single number $\widetilde{\mu}_{\theta_j \to x_i}(x_i = j)$ for variable x_i to evaluate the optimal label. Consequently, each variable x_i needs to send a single number $\widetilde{\mu}_{x_i \to \theta_j}(x_i = j)$ for factor θ_j to update its message. ■

From the result in Lemma 1, we derive the simplified message-passing rules stated in the following theorem. The proof is given in the Appendix.

Theorem 1: The optimal label x_i^{opt} given in Eq. (4) can be evaluated by the simplified message-passing rules:

$$\mu^t_{x_i \to \theta_j} = e_i(j) - \min_{k \in \mathcal{N}[i] \setminus j} \left[e_i(k) + \mu^{t-1}_{\theta_k \to x_i} \right], \; \forall j \in \mathcal{N}[i] \quad (9)$$

$$\mu^t_{\theta_i \to x_j} = \max \left\{ 0 \, , \, \mu^{t-1}_{\theta_i \to x_i} + \mu^{t-1}_{x_i \to \theta_i} - \min \left\{ 0, \mu^{t-1}_{x_j \to \theta_i} \right\} \right\} \quad (10)$$

with

$$\mu^t_{\theta_i \to x_i} = \sum_{k \in \mathcal{N}(i)} \min \left\{ 0, \mu^{t-1}_{x_k \to \theta_i} \right\} \quad (11)$$

The simplified MEPA protocol requires customizing information for each particular neighbor. In wireless communications systems, e.g. WSNs, this means separate transmissions for each and every neighbor, wasting energy and further exacerbating the contention conditions of the shared communication channel. It would be simpler and more energy efficient if each node could fuse the received messages and broadcast a single message without the need for customization. In the next section we will present a broadcast version of this simplified MEPA protocol.

IV. Message-Broadcast Protocol for Distributed Facility Location

A. Message-Broadcast: Reducing Transmissions

We consider next another variance of MEPA in which nodes broadcast the same message to all neighbors. This differs from the standard and simplified MEPA as it does not require updating and sending separate messages to separate neighbors. We start from important observations stated below

Observation 2: Consider a node $i \in \mathcal{G}$. The intra-node messages $\mu^{t-1}_{\theta_i \to x_i}$ (in Eq. (11)) and $\mu^{t-1}_{x_i \to \theta_i}$ (from Eq. (9), with $j = i$) are constants for all neighbors $k \in \mathcal{N}(i)$.

Observation 3: Consider a node i and its neighbor j. Denote the set $S := \left\{ e_i(k) + \mu^{t-1}_{\theta_k \to x_i} : k \in \mathcal{N}[i] \setminus j \right\}$; we are interested in the value $\min S$ in order to evaluate the message in Eq. (9). Denote further the set $S' := \left\{ e_i(k) + \mu^{t-1}_{\theta_k \to x_i} : k \in \mathcal{N}[i] \right\}$. We have $\min S' = \min \left\{ S, e_i(j) + \mu^{t-1}_{\theta_j \to x_i} \right\}$, which implies $e_i(j) + \mu^{t-1}_{\theta_j \to x_i} \geq \min S'$. Thus we have two cases:

1) if $e_i(j) + \mu^{t-1}_{\theta_j \to x_i} > \min S'$, then $\min S = \min S'$.
2) if $e_i(j) + \mu^{t-1}_{\theta_j \to x_i} = \min S'$, then $\min S = \text{second} \min S'$.

From these observations, we are able to derive a broadcast version of MEPA, stated in the following theorem.

Theorem 2: To evaluate the optimal labels at each iteration, each node i needs to broadcast only one message to all neighbors, containing three numbers: 1) $\alpha := \mu^{t-1}_{\theta_i \to x_i} + \mu^{t-1}_{x_i \to \theta_i}$, 2) $\beta_1 := \min S'$, and 3) $\beta_2 := \text{second} \min S'$.

At each neighbor j, the messages in Eq. (9) and Eq. (10) can be reconstructed using its prior messages sent to i, as follows:

$$\mu^t_{x_i \to \theta_j} = \begin{cases} e_i(j) - \beta_1 & \text{if } e_i(j) + \mu^{t-1}_{\theta_j \to x_i} > \beta_1 \\ e_i(j) - \beta_2 & \text{if } e_i(j) + \mu^{t-1}_{\theta_j \to x_i} = \beta_1 \end{cases}$$

$$\mu^t_{\theta_i \to x_j} = \max \left\{ 0 \, , \, \alpha - \min \left\{ 0, \mu^{t-1}_{x_j \to \theta_i} \right\} \right\}$$

B. Protocol Execution

From Theorem 2, we provide the pseudocode of broadcast-based MEPA protocol for each node i in Algorithm 1, including two phases: Facility node selection, then facility node association. Note that each set $\{x_i, e_i, \theta_i\}$ of vertices in factor graph $\mathcal{F}$ corresponds to a node i in network graph $\mathcal{G}$. Thus, we denote by m_j the message $\mu^t_{x_i \to \theta_j}$, and by n_j the message $\mu^t_{\theta_i \to x_j}$ supposed to be sent from i to $j \in \mathcal{N}(i)$. The messages can be initialized arbitrarily [4]. In our implementation we initialize them to zeros.

Algorithm 1: Broadcast MEPA for each node i

Initialization
- $\mathcal{N}[i] \leftarrow$ Neighbor discovery
- estimate and exchange $e_i(j)$ with each $j \in \mathcal{N}[i]$
- $m_j = 0, n_j = 0 \quad \forall j \in \mathcal{N}[i]$
- update and broadcast tuple $< \alpha, \beta_1, \beta_2 >$

end

Facility Node Selection
- **repeat**
  - **for** *each neighbor* $j \in \mathcal{N}(i)$ **do**
    - set *Timer* to receive message from j
    - **if** *MessageReceived* **then**
      - reconstruct *incoming* messages m_j, n_j
    - **else if** *TimeOut* **then**
      - recall previous *incoming* messages m_j, n_j
  - update and broadcast tuple $< \alpha, \beta_1, \beta_2 >$
- **until** *max_iter*

end

Facility Node Association
- $x_{temp} \leftarrow \arg\min_{j \in \mathcal{N}[i]} [e_i(j) + n_j]$
- **if** $x_{temp} = myID$ **then**
  - $x_{final} \leftarrow myID$
  - announceFacilityNode($myID$)
  - collectJoinFacility()
- **else**
  - $x_{candidates} \leftarrow$ collectAnnounce()
  - $x_{final} \leftarrow j \in x_{candidates}$ with *least cost*
  - joinFacility($myID, x_{final}$)

end

In the pseudocode of Algorithm 1, the facility node selection procedure – the main procedure – essentially consists of receiving, updating, and broadcasting messages containing three numbers $< \alpha, \beta_1, \beta_2 >$. The procedure terminates when the maximum number of iterations *max_iter* is reached. We elaborate this upper bound in the subsequent evaluation section.

V. PERFORMANCE EVALUATIONS

In this section, we analyze the minimum cost *approximation rate* of MEPA protocol with respect to the number of iterations. We compare the total cost found by MEPA with the optimal cost found by a centralized algorithm using the Integer Linear Programming solver (ILP) of TOMLAB/CPLEX optimization packet. We evaluate our broadcast MEPA on synthetic Barabasi-Albert small-world graphs generated using the BRITE graph topology generator, with incremental growth model $m = 2$. The number of nodes is from 200 to 1000. We set the connection cost as hop distance, with neighborhood of each node defined in 2 hops, and fixed facility cost as diameter of the largest connected component of the graph. We implement a round-robin message initiation pattern using unique, orderable node IDs, with each node sends out initial messages immediately, and sends out its k^{th} messages only after receiving $(k-1)^{th}$ messages from all of its neighbors. We use a damping factor [6] of 0.3 to avoid numerical oscillations.

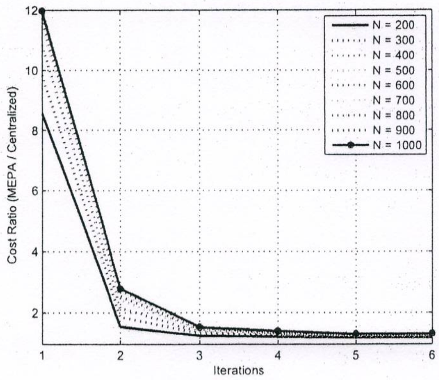

Fig. 1. Approximation ratio of MEPA compared to centralized algorithm using CPLEX MILP solver

Fig. 1 shows the convergence properties of MEPA protocol under different node density, averaged over 100 runs for each configuration. First, the minimum cost found by MEPA quickly converges to the optimal cost found by ILP solver, as the number of iterations increases. After 3 iterations, MEPA can achieve an approximation ratio around 1.5 for different node density, and then gradually improves the approximation ratio to reach 1.2 approximation after more than 5 iterations. Second, the convergence rate and approximation ratio are quite consistent for different node density, showing the scalability of the protocol.

VI. CONCLUSION AND FUTURE WORK

In this paper we proposed a distributed and localized min-sum message-passing algorithm for solving the facility location problem in large-scale networks. We showed that the algorithm is highly localized, able to achieve good convergence rate and approximation ratio, and scalable to network size. Since the factor graph $\mathcal{F}$ contains loops, we are currently working further on the convergence properties of the algorithm.

ACKNOWLEDGMENT

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2009-(C1090-0902-0002)). Corresponding author: Sungyoung Lee.

APPENDIX
DERIVATION OF SIMPLIFIED MESSAGE-PASSING RULES

In this proof we derive the update rules for single value messages $\widetilde{\mu}_{\theta_i \to x_j}(x_j = i)$ and $\widetilde{\mu}_{x_i \to \theta_j}(x_i = j)$. We start by analyzing the message $\mu^t_{\theta_i \to x_j}$ sent from factor θ_i to variable x_j, given in Eq. (3), in case $x_j = i$ (node j takes i as its facility node)

If $i = j$ (or $\theta_i \equiv \theta_j$), then the message becomes

$$\mu^t_{\theta_i \to x_i}(x_i = i = j) = \sum_{k \in \mathcal{N}(i)} \min_{x_k} \mu^{t-1}_{x_k \to \theta_i}(x_k) \tag{12}$$

If $i \neq j$ (or $\theta_i \neq \theta_j$), since $x_j = i$, x_i must be equal to i for a valid configuration, or equivalently, $\min_{x_i} \mu^{t-1}_{x_i \to \theta_i}(x_i) = \mu^{t-1}_{x_i \to \theta_i}(i)$. Thus the message $\mu^t_{\theta_i \to x_j}(x_j)$ becomes

$$\mu^t_{\theta_i \to x_j}(x_j = i \neq j) = \mu^{t-1}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min_{x_k} \mu^{t-1}_{x_k \to \theta_i}(x_k) \tag{13}$$

For vector messages (of length $|\mathcal{N}[i]|$) from variable x_i to factor θ_j, we also rewrite them as the sum of constant and variable (with respect to x_i) components, $\mu^t_{x_i \to \theta_j}(x_i) = \overline{\mu}_{x_i \to \theta_j} + \widetilde{\mu}_{x_i \to \theta_j}(x_i)$. Let $\overline{\mu}_{x_i \to \theta_j} = \min_{x_i \neq j} \mu^{t-1}_{x_i \to \theta_j}(x_i)$, then we have $\min_{x_i \neq j} \widetilde{\mu}_{x_i \to \theta_j}(x_i) = 0$, and it follows that:

$$\min_{x_i} \widetilde{\mu}_{x_i \to \theta_j}(x_i) = \min\{0, \widetilde{\mu}_{x_i \to \theta_j}(x_i = j)\} \tag{14}$$

In the following, by rewriting the messages into two components as described above, then applying the above results, we further simplify the messages:

Applying Eq. (8) to Eq. (2), $\mu^t_{x_i \to \theta_j}(x_i) =$

$$= e_i(x_i) + \sum_{\theta_k \in \Theta(x_i) \setminus \theta_j} \widetilde{\mu}_{\theta_k \to x_i}(x_i) + \sum_{\theta_k \in \Theta(x_i) \setminus \theta_j} \overline{\mu}_{\theta_k \to x_i}$$

$$= \begin{cases} e_i(j) + \sum_{\theta_k \in \Theta(x_i) \setminus \theta_j} \overline{\mu}_{\theta_k \to x_i} & if\, x_i = j \\ e_i(x_i) + \widetilde{\mu}_{\theta_{x_i} \to x_i}(x_i) + \sum_{\theta_k \in \Theta(x_i) \setminus \theta_j} \overline{\mu}_{\theta_k \to x_i} & if\, x_i \neq j \end{cases}$$

Applying Eq. (14) to Eq. (12), $\mu^t_{\theta_i \to x_i}(x_i = i = j) =$

$$\begin{aligned} &= \sum_{k \in \mathcal{N}(i)} \min_{x_k} \widetilde{\mu}_{x_k \to \theta_i}(x_k) + \sum_{k \in \mathcal{N}(i)} \overline{\mu}_{x_k \to \theta_i} \\ &= \sum_{k \in \mathcal{N}(i)} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} + \sum_{k \in \mathcal{N}(i)} \overline{\mu}_{x_k \to \theta_i} \end{aligned} \tag{15}$$

Applying Eq. (14) to Eq. (13), $\mu^t_{\theta_i \to x_j}(x_j = i \neq j) =$

$$\begin{aligned} &= \widetilde{\mu}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min_{x_k} \widetilde{\mu}_{x_k \to \theta_i}(x_k) + \sum_{k \in \mathcal{N}[i] \setminus j} \overline{\mu}_{x_k \to \theta_i} \\ &= \widetilde{\mu}_{x_i \to \theta_i}(i) + \\ &\quad + \sum_{k \in \mathcal{N}(i) \setminus j} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} + \sum_{k \in \mathcal{N}[i] \setminus j} \overline{\mu}_{x_k \to \theta_i} \end{aligned} \tag{16}$$

Applying Eq. (14) to Eq. (5), $\mu^t_{\theta_i \to x_i}(x_i \neq j = i) =$

$$= \sum_{k \in \mathcal{N}(i)} \min_{x_k} \widetilde{\mu}_{x_k \to \theta_i}(x_k) + \sum_{k \in \mathcal{N}(i)} \overline{\mu}_{x_k \to \theta_i} = \sum_{k \in \mathcal{N}(i)} \overline{\mu}_{x_k \to \theta_i} \tag{17}$$

Applying Eq. (14) to Eq. (6), $\mu^t_{\theta_i \to x_j}(x_j \neq i \neq j) =$

$$\begin{aligned} &= \min \left\{ \begin{array}{c} \widetilde{\mu}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min_{x_k} \widetilde{\mu}_{x_k \to \theta_i}(x_k), \\ \min_{x_i \neq i} \widetilde{\mu}_{x_i \to \theta_i}(x_i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min_{x_k \neq i} \widetilde{\mu}_{x_k \to \theta_i}(x_k) \end{array} \right\} \\ &\quad + \sum_{k \in \mathcal{N}[i] \setminus j} \overline{\mu}_{x_k \to \theta_i} \\ &= \min \left\{ 0, \widetilde{\mu}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} \right\} \\ &\quad + \sum_{k \in \mathcal{N}[i] \setminus j} \overline{\mu}_{x_k \to \theta_i} \end{aligned} \tag{18}$$

Now we are ready to derive the simplified update equations for the variable component messages by solving $\widetilde{\mu}_{x_i \to \theta_j}(j) = \mu^t_{x_i \to \theta_j}(j) - \overline{\mu}_{x_i \to \theta_j}$ and $\widetilde{\mu}_{\theta_i \to x_j}(i) = \mu^t_{\theta_i \to x_j}(i) - \overline{\mu}_{\theta_i \to x_j}$.

$$\begin{aligned} &\widetilde{\mu}_{x_i \to \theta_j}(j \mid j \in \mathcal{N}[i]) = \mu^t_{x_i \to \theta_j}(j) - \min_{x_i \neq j} \mu^{t-1}_{x_i \to \theta_j}(x_i) \\ &= e_i(j) + \sum_{\theta_k \in \Theta(x_i) \setminus \theta_j} \overline{\mu}_{\theta_k \to x_i} - \\ &\quad - \min_{x_i \neq j} \left[e_i(x_i) + \widetilde{\mu}_{\theta_{x_i} \to x_i}(x_i) + \sum_{\theta_k \in \Theta(x_i) \setminus \theta_j} \overline{\mu}_{\theta_k \to x_i} \right] \\ &= e_i(j) - \min_{x_i \neq j} \left[e_i(x_i) + \widetilde{\mu}_{\theta_{x_i} \to x_i}(x_i) \right] \end{aligned}$$

$$\begin{aligned} &\widetilde{\mu}_{\theta_i \to x_j}(i \mid i \in \mathcal{N}[j]) \\ &= \mu^t_{\theta_i \to x_j}(i) - \overline{\mu}_{\theta_i \to x_j} = \mu^t_{\theta_i \to x_j}(i) - \mu^t_{\theta_i \to x_j}(x_j \neq i) \end{aligned}$$

If $i = j$, from Eq. (15) and (17) we have $\widetilde{\mu}_{\theta_i \to x_i}(i) =$

$$\begin{aligned} &= \sum_{k \in \mathcal{N}(i)} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} + \sum_{k \in \mathcal{N}(i)} \overline{\mu}_{x_k \to \theta_i} - \sum_{k \in \mathcal{N}(i)} \overline{\mu}_{x_k \to \theta_i} \\ &= \sum_{k \in \mathcal{N}(i)} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} \end{aligned}$$

Similarly, if $i \neq j$, from Eq. (16) and (18) we have:

$$\begin{aligned} &\widetilde{\mu}_{\theta_i \to x_j}(i) = \widetilde{\mu}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} + \\ &\quad - \min \left\{ 0, \widetilde{\mu}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i) \setminus j} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} \right\} \\ &= \max \left\{ \begin{array}{c} 0, \widetilde{\mu}_{x_i \to \theta_i}(i) + \sum_{k \in \mathcal{N}(i)} \min\{0, \widetilde{\mu}_{x_k \to \theta_i}(i)\} \\ - \min\{0, \widetilde{\mu}_{x_j \to \theta_i}(i)\} \end{array} \right\} \\ &= \max\{0, \widetilde{\mu}_{x_i \to \theta_i}(i) + \widetilde{\mu}_{\theta_i \to x_i}(i) - \min\{0, \widetilde{\mu}_{x_j \to \theta_i}(i)\}\} \end{aligned}$$

References

[1] P. Mirchandani and R. Francis, *Discrete location theory*. John Wiley and Sons, 1990.

[2] C. Gkantsidis, T. Karagiannis, and M. VojnoviC, "Planet scale software updates," in *Proc. of ACM SIGCOMM '06*, 2006, pp. 423–434.

[3] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, May/June 2006.

[4] F. R. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Nov. 2001.

[5] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, no. 2.

[6] B. J. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, February 2007.