Message-passing in stochastic processing networks

Devavrat Shah[a,b]

[a] *Laboratory for Information and Decision Systems, Department of EECS*
Massachusetts Institute of Technology
[b] devavrat@mit.edu

Abstract

Simple, distributed and iterative algorithms, popularly known as *message-passing*, have become the architecture of choice for emerging infrastructure networks and the canonical behavioral model for natural networks. Therefore designing, as well as understanding, message-passing algorithms has become important.

The purpose of this survey is to describe the state-of-art of message-passing algorithms in the context of dynamic resource allocation in the presence of uncertainty, a problem that is central to operations research (OR) and management science (MS). Various directions for future research are described in this context as well as connections beyond OR and MS are explained. Through this survey, we hope to convey the opportunity presented to the OR and MS community to benefit from and contribute to the growing inter-disciplinary area of message-passing algorithms.

Keywords: message-passing, markov random field, duality, loss networks, variational characterization, belief propagation, maximum weight policy, utility maximization policy

1. Introduction

We are ushering into an era of everything *networked*: social networks connect people across the globe, everything is instantly searchable and advertiseable, shopping is a 'click away', 'moods' in stock-market are instantly observable through micro-blog sites, and so on. All such systems utilize as their backbone computation, storage and communication infrastructure networks. To operate these infrastructure networks in an efficient and scalable manner, simple, iterative, and distributed or otherwise called “message-passing algorithms”, have emerged as the architecture of choice. On the other hand, such message-passing mechanisms provide a canonical behavioral model to describe and understand the interactive evolution of natural networked systems. For example, they model how humans react to changes in a policy, and they subsequently assess the success or failure of the policy; they also model interaction of financial agencies in a market, or the spread of opinions in a social scenario. Therefore, it is of utmost importance to develop new methods as well as understand existing methods for designing message-passing algorithms so as to be able to design large-scaled networked system or to provide/refute behavioral models for naturally networked system.

In this survey, motivated by this impending need, we shall discuss the state-of-the-art as well as directions for future research of message-passing algorithms in the context of dynamic resource allocation which is one of the fundamental problems in operations research, management science and more broadly any engineering discipline. Usually, two types of operational questions arise in this context. One: capacity planning, i.e. how much of a resource should be planned for to meet desired performance criteria. Two: contention resolution or scheduling, i.e. how should contentions be resolved among various entities competing for access to given resources. The capacity planning problem is essentially an *offline* question, in the sense that it needs updating on a longer time scale. It arises naturally in many contexts. Examples include the design of a communication network such as the telephone network, with a desired quality of service in terms of the call-drop rate; decisions about staffing of skilled workers in a consulting firm to maximize revenue; maintenance of product repositories such as air-tickets or hotel-rooms in a travel agency; or planning for the number of disks and servers needed to operate a data center facility. In contrast to the planning problem, the task of scheduling is an *online* question, where decisions are made at a much shorter (operational) time scale. For example, scheduling calls in a telephone network, sharing bandwidth in the Internet among users accessing it, assigning skilled staff to a project in a consulting firm, generating deal for a given travel request using existing repository in a travel agency, or allocating disks and servers to an arriving job in a data center.

Both questions, capacity planning and scheduling, are algorithmic in nature. These algorithms need to be implemented, usually in a large networked system, while respecting application dependent and technological constraints. For this reason, in most applications of interest, the only type of algorithms that can be implemented are the ones that are iterative, distributed and perform simple computation per iteration while maintaining little data structure. The motivation for considering such *message-passing* algorithms could be different across applications. For example, in the context of the communication network such as the Internet, the only feasible as well as scalable solution for sharing bandwidth is the one in which individual users find their appropriate share by iteratively reacting to feedback obtained through the network in terms of drop of packets or acknowledgment of successful transmission. Similarly, in generating travel deals for a travel query it is essential to use distributed, iterative algorithm to plough through the massive repository in a cloud computation facility. For similar reasons message-passing algorithms provide a canonical solution for deciding revenue maximizing availability of skilled staff in a consulting firm. In summary, efficient message-passing algorithms are essential and urgently needed for solving variety of operational problems including those of capacity planning and scheduling.

As mentioned earlier, the primary purpose of this paper is to survey various message-passing algorithms for capacity planning and scheduling in the context of stochastic processing networks. The stochastic processing network model was introduced by Harrison (2000). It has emerged as a canonical model to capture dynamic resource allocation problems including the capacity planning and scheduling faithfully across a variety of applications. In such a model, a collection (or network) of queues are served from a prespecified set of actions (or schedules). Each of these queues may receive exogenous demand. Further, servicing of demand at one queue may lead to creation of demand at another queue, or the serviced demand may leave the network. In this survey, we shall focus on a scenario where the collection of actions are described through a finite set of linear inequalities over discrete or continuous space. Specifically, given a network of N queues, one could allocate non-negative service rates $\mathbf{x} = [x_i]_{1 \le i \le N}$ such that

$$\begin{aligned} A\mathbf{x} &\le \mathbf{C}, \\ \mathbf{x} &\in \Sigma^N. \end{aligned} \tag{1}$$

Here $A = [A_{ij}]$ is a non-negative valued $M \times N$ matrix, $\mathbf{C} = [C_i]$ is a non-negative valued vector of length M, and $\Sigma \subset \mathbb{R}_+$ is either finite, discrete set or continuous set of type $\Sigma = [a, b]$ for $0 \le a < b \le \infty$ including $\Sigma = \mathbb{R}_+$.

As discussed later, the stochastic processing network model with constraints described as per (1) captures problems arising in variety of applications: this includes those arising in communication networks such as telephone networks and the Internet as well as network revenue maximization.

The basic question of interest in the context of capacity planning problem concerns finding out whether a given set of resources are sufficient to meet a given set of demands. Equivalently, it requires evaluation of the fraction of demands that can not be satisfied using a given set of resources. In a sense, this is a *first order* performance metric. Therefore, to evaluate a good approximation of the rate at which demands are lost (the loss rate), a bufferless stochastic processing network model is utilized. In the literature, it is popularly known as a stochastic loss network. This model and associated problem have been classically well studied with the motivation of designing telephone networks. However, as explained in an excellent survey by Kelly (1991), this model provides the means to address variety of network revenue maximization questions such as those mentioned through examples earlier in this section. Formally, the model as well as the problem of capacity planning are described in Section 3. We also describe known message-passing algorithmic solutions. Roughly speaking, they are divided into two classes. One, the optimization based algorithm and a closely associated algorithm, which is known as the *Erlang approximation*. Two, the variational approximation based algorithms, specifically the mean-field and belief propagation. It has been well understood that the optimization based algorithm (as well as the Erlang approximation) provides an excellent approximation in the large network limit, cf. Kelly (1991). In a sense, such an algorithm captures the effect induced by *mode* of the underlying stationary distribution quite well. The mean-field and belief propagation algorithms, on the other hand, seem to capture the higher order effect (specifically, that induced by the *entropy* term in the variational characterization of the stationary distribution) in addition to that of the mode. While these algorithms seem to be developed in the right direction, questions related to their properties as well as development of their further refinement suggest exciting directions for future work. This is well summarized through concrete open questions stated near the end of Section 3.

In the context of scheduling problems, the performance metric of interest is finiteness of queue-sizes (smaller the better). In contrast to the capacity planning problem, this is a *second order* performance metric. In Section 4, we introduce the problem of scheduling in the context of stochastic processing networks with infinite buffers. Interest is in two high-performance myopic scheduling policies: the maximum weight by Tassiulas and Ephremides (1992) and α-fair sharing by Kelly et al. (1998); Mo and Walrand (1998). Section 4 discusses message-passing implementation of these policies in the context of two examples in detail: bandwidth sharing in the Internet and scheduling in an input-queued switch of an Internet router. Bandwidth sharing in the Internet requires solving a strictly concave maximization with convex constraint set in a continuous domain. Given the nature of 'message-passing' constraints and the form of optimization problems, the primal-dual algorithm provides the appropriate solution. This message-passing algorithmic solution is quite general and provides implementation of α-fair policy for any instance of stochastic processing network considered in this survey. On the other hand, implementing the maximum weight in the context of input-queued switch requires solving a combinatorial optimization problem. In the context of input-queued switch, this problem reduces to finding a maximum weight assignment or matching in an edge weighted bipartite graph. The belief propagation heuristic, in this specific context, provides an excellent message-passing implementation to solve this problem exactly. The belief propagation, for this problem, happens to be closely related to the auction algorithm by Bertsekas (1992). The auction algorithm, a variant of the classical dual co-ordinate descent algorithm, is known to solve the maximum weight matching or assignment problem in a bipartite graph in a message-passing manner. However, such exact message-passing solutions are not known to provide exact implementation of the maximum weight policy for generic instance of stochastic processing network considered in this survey. The associated challenges and exciting directions for future work in this context are summarized near the end of Section 4.

We state our conclusions in Section 5 where we summarize the key messages of the survey supported by explanations provided in Sections 3 and 4: message-passing algorithms are essential for designing and understanding future networked systems; this is a topic of broad interest with a lot to be done; and dynamic resource allocation in stochastic networks could provide an excellent fertile ground for this future development.

1.1. Related work, some connections

Here we start by describing some of the closely related works on resource allocation in form of surveys or monographs followed by various connections that message-passing algorithms bring to OR and MS.

For capacity planning (stochastic loss network), the survey by Kelly (1991) is a must read. It provides a detailed overview of illustrious development that took place in the 1980s. It also discusses the role of stochastic loss networks in the context of network revenue maximization. For the problem of scheduling, which is currently actively researched, there are excellent sources to understand various aspects. Specifically, the book by Srikant (2004) provides a detailed overview of bandwidth sharing model, relation between TCP protocol and primal-dual algorithm for an associated optimization problem and the stability of congestion control protocols. The book chapter by Shah (2008) provides an overview of switched network model, message-passing implementation of scheduling algorithms and associated performance trade-offs. The more recent monographs by Georgiadis et al. (2006) and Shakkottai and Srikant (2007) provides a unified architectural view of scheduling and congestion control policies for an end-to-end design of communication networks. Finally, the monograph by Shah (2009) discusses strengths and limitations of a class of extremely simple, randomized message-passing algorithms, also known as the *Gossip* algorithms.

As mentioned earlier, the message-passing algorithms discussed in this survey can be roughly classified based on two methods. The first method is based on theory of optimization. In a nutshell, all algorithms utilize the structure of the Lagrangian dual associated with the primal optimization problem of interest, with the constraints of the form (1), to obtain message-passing implementation. This method is classical and has roots in work by Rockafellar (1998) on monotropic programs. An interested reader can find, for example a variety of refinements as well as applications in the book by Bertsekas and Tsitsiklis (1997) on parallel and distributed computation.

The second method uses structural properties of constraints (1) directly to design an approximate dynamic programming based message-passing implementation. As it happens, the resulting algorithm (belief propagation) ends up solving an appropriate, related problem which can be thought of an approximation of the original problem in the so called *variational* form. It is somewhat amusing to note that the belief propagation algorithm has been discovered and re-discovered over years in variety of different contexts with different perspectives: early on, Bethe (1935) used similar approximations to evaluate the free energy of a certain statistical physics model, Gallager (1962) used it as a meaningful heuristic in place of the maximum likelihood decoding for low-density parity check codes and Pearl (1988) stated it as a heuristic for inference in a generic probabilistic graphical model or Markov Random Field (MRF). The relation between variational approximation and belief propagation was first observed by Yedidia et al. (2001). An interested reader can find, for example, a detailed discussion on relation between message-passing algorithms and variational approximations in the book by Wainwright and Jordan (2008). The treatment of belief propagation from the statistical physics perspective is detailed in the book by Mezard and Montanari (2009). The use of belief propagation in designing communication systems is detailed in the book by Richardson and Urbanke (2008).

2. Preliminaries

2.1. Markov Random Field

The Markov Random Field (MRF) is a succinct way to represent the joint distribution of a collection of random variables by means of graphical model. It shall provide a common framework to discuss both capacity planning and scheduling problems. This shall allow use of MRF based message-passing algorithms, like belief propagation, for solving capacity planning and scheduling problem.

In this survey, the MRF of interest will be of the following type. Consider a collection of N random variables $\mathbf{X} = [X_i]_{1\leq i\leq N}$ taking values over the subset of Σ^N defined through constraints given by (1). Specifically, for any $\mathbf{x} \in \Sigma^N$

$$\begin{aligned}\mathbb{P}\Big(\mathbf{X} = \mathbf{x}\Big) &\propto \exp\Big(\sum_i \phi_i(x_i)\Big) \prod_{1\leq j\leq M} \mathbf{1}_{\{\sum_k A_{jk}x_k \leq C_j\}} \\ &= \frac{1}{Z}\exp\Big(\sum_i \phi_i(x_i)\Big) \prod_{1\leq j\leq M} \mathbf{1}_{\{\sum_k A_{jk}x_k \leq C_j\}}. \qquad (2)\end{aligned}$$

In the above, $\phi_i : \Sigma \to \mathbb{R}$ is a real-valued function for all i; $\mathbf{1}_{\{\cdot\}}$ represents indicator with $\mathbf{1}_{\{\text{true}\}} = 1$ and $\mathbf{1}_{\{\text{false}\}} = 0$; and Z is the normalization constant (also called partition function). In (2), $\mathbb{P}(\cdot)$ should be treated as density if Σ is continuous. The corresponding graphical model (also known as the factor graph) is given by a bipartite graph $G = (U \cup V, E)$ with partition $U = \{u_1, \ldots, u_N\}$ where node u_i corresponds to a random variable X_i; partition $V = \{v_1, \ldots, v_M\}$ where node v_j corresponds to the jth constraint $\sum_k A_{jk}x_k \leq C_k$; and $E \subset U \times V$ with $E = \{(u_i, v_j) : A_{ji} \neq 0\}$. Note that even though the size of the support space of the distribution could be exponentially large in N, the description is only polynomially large in N. Therefore, such a representation (MRF) has become quite useful. For example, application MRFs in hierarchical bayesian models is explained in Gilks et al. (1996) and in bioinformatics is explained in Pevzner (2000). In the context of communication systems they were first used by Gallager (1962); for representing satisfiability and combinatorial optimization they have been used by Geman and Geman (1984); and Nemhauser and Wolsey (1999) used them for image processing. In a sense, this survey describes application of MRFs and related algorithms in the context of resource allocation in stochastic networks. The book by Lauritzen (1996) provides a good introduction to the topic of Markov Random Fields. Finally, it should be noted that the relation between Markov Random Field and Graphical Models is fundamental as explained through the celebrated result by Hammersley and Clifford (see book by Lauritzen (1996)).

2.2. Two problems

Given an MRF, in general as well as in this survey, there are two algorithmic or computational problems of interest. The first problem is computing marginal distributions of all random variables. That is, finding $\mathbb{P}(X_i = \sigma)$ for all $\sigma \in \Sigma$ and for all $1 \leq i \leq N$. The second problem is computing the mode or maximum a posteriori (MAP) assignment of the distribution. That is, finding $\mathbf{x}^* \in \Sigma^N$ so that $\mathbb{P}(\mathbf{X} = \mathbf{x}^*)$ is maximum: $\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x} \in \Sigma^N} \mathbb{P}(\mathbf{X} = \mathbf{x})$. We shall denote the first problem by MARG and the second problem by MAP. The problem of MARG is precisely needed to evaluate loss rates in the context of stochastic loss network. The problem of MAP, which can represent an optimization problem with constraints given by (1), is precisely required to be solved for scheduling as per the maximum weight and α-fair sharing policies in a stochastic processing network. Thus, both capacity planning and scheduling problems can be casted as questions (MARG and MAP respectively) on an appropriate MRF. This abstraction, though may seem like a formal excercise, allows one to utilize message-passing algorithms (like belief propagation) developed for solving MARG and MAP for the purpose of computing loss rate and scheduling efficiently.

It is worth noting that both problems, MARG and MAP are computationally hard in general. For example, with choice of $\Sigma = \{0, 1\}$, appropriate $A \in \{0, 1\}^{M \times N}$ (and $M \leq N^2$) and $\phi_i(\sigma) = 0$ for $\sigma \in \{0, 1\}$ for all i, it is possible to model uniform distribution over the space of independent sets of a given undirected graph $\mathcal{G}$ of N vertices as an MRF of the form (2). In that case, the normalization constant Z becomes number of independent sets of $\mathcal{G}$. Counting the number of independent sets of any graph is a #P-complete problem, see Valiant (1979). It is also well known that a polynomial (in N) time algorithm for solving MARG for any such resulting MRF will lead to polynomial (in N) time algorithm for computing Z, see for example Weitz (2006). Therefore, computing MARG in a general MRF is a #P-complete problem. To see that MAP is computationally hard (NP-hard), consider the same setup. Then MAP corresponds to finding an independent set of $\mathcal{G}$ with the largest cardinality. This is known to be NP-hard problem and hard to approximate. See for example Arora and Lund (1996). That is, MAP in a general MRF is NP-hard.

2.3. Message-passing algorithm

Even though MARG and MAP are hard problems in general, they need to be addressed in practice (e.g. for resource allocation in stochastic networks). An ideal algorithmic solution is the one that provides exact or reasonably accurate answers when the underlying graphical structure is *easy* and reasonable heuristic answer when it is *hard*. As discussed earlier, in most settings of interest, it is preferred that such algorithms are of a message-passing nature. That is, the algorithms are iterative, distributed with respect to the graphical structure of the MRF and perform simple computation per iteration while maintaining little data structure. In the formalism described above, each node of G maintains a minimal state based on which solution is estimated – in case of MARG, the marginal probability of the corresponding random variable and in case of MAP, an assignment of the particular random variable. This per-node state evolves iteratively using simple computation and based on information exchanged or messages passed between neighbors with respect to G.

Message-passing algorithms have been well studied in the past and there has been an emergence of interest in recent years. This survey attempts to review various types of message-passing algorithms relevant for capacity planning and scheduling in stochastic networks. Specifically, the algorithms considered in this paper fall in roughly two categories. One, the optimization based algorithms. They utilize the structure of the dual induced by constraints of the type (1) to obtain message-passing dual or primal-dual algorithms. Two, variational approximation of Markov Random Field leading to message-passing algorithms such as the belief propagation and mean-field. These algorithms explicitly utilize the graphical structure induced by the constraints (1). The belief propagation algorithm is of particular interest here. The term belief propagation is used as an umbrella heuristic for both problems: MARG and MAP. Its versions for solving MARG and MAP problems are known as the sum-product and max-product (also min-sum) respectively. We shall utilize the term belief propagation to refer to both versions in this paper; the context should make it clear which version we are referring to. The general form of belief propagation algorithm for the Markov Random Field of the type described above is presented in the context of capacity planning problem for MARG near the end of Section 3 and in the context of scheduling for MAP near the end of Section 4. An interested reader is referred to, for example book by Wainwright and Jordan (2008), for the most general form of such an algorithm.

2.4. Notation

Let $\mathbb{N}$ be the set of natural numbers $\{1, 2, \ldots\}$, let $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$, let $\mathbb{R}$ be the set of real numbers, let $\mathbb{R}_+ = \{x \in \mathbb{R} : x \geq 0\}$ and $\mathbb{R}_{++} = \{x \in \mathbb{R} : x > 0\}$. Let $\mathbf{1}_{\{\cdot\}}$ be the indicator function, $\mathbf{1}_{\text{true}} = 1$ and $\mathbf{1}_{\text{false}} = 0$. Let $x \wedge y = \min(x, y)$ and $x \vee y = \max(x, y)$ and $[x]^+ = x \vee 0$. When x is a vector, the maximum is taken componentwise.

We will reserve bold letters for vectors in $\mathbb{R}^N$, for example $\mathbf{x} = [x_n]_{1 \leq n \leq N}$. Let $\mathbf{0}$ be the vector of all 0s, and $\mathbf{1}$ be the vector of all 1s. Use $|\mathbf{x}|$ to represent the ℓ_∞ norm, $\max_n |x_n|$ and $\|\mathbf{x}\|$ to represent the ℓ_2 norm, $\left(\sum_n x_n^2\right)^{1/2}$. For a set $S \subset \mathbb{R}_+^N$ and $\mathbf{x} \in \mathbb{R}_+^N$, define

$$d(\mathbf{x}, S) = \inf\{\|\mathbf{x} - \mathbf{y}\| : \mathbf{y} \in S\}.$$

For vectors $\mathbf{u}$ and $\mathbf{v}$ and functions $f : \mathbb{R} \to \mathbb{R}$, let

$$\mathbf{u} \cdot \mathbf{v} = \sum_{n=1}^{N} u_n v_n, \quad \mathbf{u}\mathbf{v} = [u_n v_n]_{1 \leq n \leq N}, \quad \text{and} \quad f(\mathbf{u}) = \big[f(u_n)\big]_{1 \leq n \leq N}$$

and let matrix multiplication take precedence over dot product so that

$$\mathbf{u} \cdot A\mathbf{v} = \sum_{n=1}^{N} u_n \left(\sum_{m=1}^{N} A_{nm} v_m \right).$$

Let A^T be the transpose of matrix A. For a set $\mathcal{S} \subset \mathbb{R}^N$, denote its convex hull by $\langle \mathcal{S} \rangle$.

Finally, log in this paper is used to represent $\log_e$, unless stated otherwise.

3. Capacity planning

This section describes message-passing algorithms for capacity planning problems. We start by describing the bufferless stochastic loss network model. Then we show that the evaluation of the primary performance metric of interest, the loss rates given demands, reduces to that of MARG for the stationary distribution of the loss network. The product-form nature of the stationary distribution leads to its representation through an appropriate Markov Random Field. We develop a variational characterization of such product-form stationary distribution. This characterization provides a unifying framework to understand performance of various message-passing algorithms developed over years with seemingly very different perspective. Specifically, the variational characterization suggests that the stationary distribution can be thought of as optimization over space of all feasible distribution with objective that has two terms: maximizing the first term leads to the identification of the mode of the distribution and the classical Erlang approximation as well as related optimization based algorithms essentially try to capture the effect of the mode on the performance; the second term corresponds to the entropy term and capturing its effect on performance leads to the higher order correction which is precisely what the recent message-passing algorithms, mean-field and belief propagation, try to capture by means of approximation of entropy. We end this section by summarizing the outstanding issues and directions for future research.

3.1. Stochastic loss network

Model. Consider a network with M links, labeled $1, 2, \ldots, M$. Each link j has C_j units of capacity. There is a set of N distinct routes, denoted by $\mathcal{R} = \{1, \ldots, N\}$. A call on route i requires A_{ji} units of capacity on link j, $A_{ji} \geq 0$. Calls on route i arrive according to an independent Poisson process of rate ν_i with $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_N)$ denoting the vector of these rates. The dynamics of the network is such that an arriving call on route i is admitted to the network if sufficient capacity is available on all links used by route i; else, the call is dropped. To simplify the exposition, we will assume that the call service times are i.i.d. exponential random variables with unit mean. It is important to note, however, that all results discussed in this survey depend on the characterization of stationary distribution which remain unchanged for a much larger class of service distributions due to the well-known *insensitivity* property of the stochastic loss networks, for example see survey by Kelly (1991).

Let $\mathbf{n}(t) = (n_1(t), \ldots, n_N(t)) \in \mathbb{Z}_+^N$ be the vector of the number of active calls in the network at time t. By definition, we have that $\mathbf{n}(t)$ satisfies constraints of the type (1). Equivalently, $\mathbf{n}(t) \in \mathcal{S}(\mathbf{C})$ where

$$\mathcal{S}(\mathbf{C}) = \left\{\mathbf{n} \in \mathbb{Z}_+^N : A\mathbf{n} \leq \mathbf{C}\right\},$$

and $\mathbf{C} = (C_1, \ldots, C_M)$ denotes the vector of link capacities.

Within this framework, $\mathbf{n}(t),\ t \geq 0$ is a *reversible* multidimensional Markov process with a product-form stationary distribution; cf. Kelly (1986). Namely, there is a unique stationary distribution $\boldsymbol{\pi}$ on the state space $\mathcal{S}(\mathbf{C})$ such that for $\mathbf{n} \in \mathcal{S}(\mathbf{C})$

$$\boldsymbol{\pi}(\mathbf{n}) = \frac{1}{Z(\mathbf{C})} \prod_{i \in \mathcal{R}} \frac{\nu_i^{n_i}}{n_i!}, \tag{3}$$

where $Z(\mathbf{C})$ is the normalization constant, i.e.

$$Z(\mathbf{C}) = \sum_{\mathbf{n} \in \mathcal{S}(\mathbf{C})} \prod_{i \in \mathcal{R}} \frac{\nu_i^{n_i}}{n_i!}. \tag{4}$$

Note that $\boldsymbol{\pi}$ is of exactly the same form as (2) and thus can be viewed as MRF with graphical model G.

Performance metric. The performance metric of interest is the loss rate L_i of calls of a given type i, for any $i \in \mathcal{R}$. Equivalently, the stationary probability that a call arriving on route i is lost. Since arrival process is an independent Poisson process for each route and the property of Poisson sampling known as Poisson Arrivals See Time Average (PASTA), it can be expressed using the stationary distribution $\boldsymbol{\pi}$. Specifically, it can be checked that

$$L_i = 1 - \frac{Z(\mathbf{C} - A\mathbf{e}_i)}{Z(\mathbf{C})}, \tag{5}$$

where $\mathbf{e}_i$ is the unit vector corresponding to a single active call on route i. An alternative characterization of L_i can be obtained as follows. Given i, consider calls arriving on route i. It is a stable system, by definition. The average delay experienced by a call that is admitted is 1 (the service requirement) and by a call that is not admitted 0 (immediate departure). Therefore, the average delay experienced by calls on route i given by

$$D_i = (1 - L_i) \times 1 + L_i \times 0 \; = \; (1 - L_i).$$

Upon applying Little's Law (see Little (1961)) to this stable system (with respect to route i), we obtain

$$\nu_i D_i = \mathbb{E}[n_i]$$

which yields

$$1 - L_i = \frac{\mathbb{E}[n_i]}{\nu_i}. \tag{6}$$

Thus, computing L_i is equivalent to computing the expected value of the number of active calls on route i with respect to the stationary distribution $\boldsymbol{\pi}$. This can be achieved by solving the corresponding MARG problem. That is, computing $\mathbb{P}(n_i = \ell)$ for all $\ell \in \mathbb{N}$, will lead to finding $\mathbb{E}[n_i]$ and hence L_i.

3.2. Large network scaling

The problem of computing loss probabilities by solving MARG problem for the stationary distribution $\boldsymbol{\pi}$ is hard in general. The problem becomes unmanageable for large networks and indeed, in many practical scenarios the interest is in large networks. Here we consier a scaling of the stochastic loss network to model the type of large networks that arise in applications such as the telephone networks (cf. Kelly (1986)). This scaling will be useful to evaluate the performance of different message-passing algorithms for computing loss probabilities as well as guide their design.

To this end, we define a scaled version with scaling parameter $r \in \mathbb{N}$ of a given stochastic loss network with capacities $\mathbf{C}$, routing matrix A and arrival rates $\boldsymbol{\nu}$, as follows: scale capacities as

$$\mathbf{C}^r = r\mathbf{C} = (rC_1, \ldots, rC_M),$$

and scale the arrival rates as

$$\boldsymbol{\nu}^r = r\boldsymbol{\nu} = (r\nu_1, \ldots, r\nu_N).$$

The corresponding feasible region of calls is given by $\mathcal{S}(r\mathbf{C})$. Now consider a normalized version of this region defined as

$$\mathcal{S}^r(\mathbf{C}) = \left\{ \frac{1}{r}\mathbf{n} : \mathbf{n} \in \mathcal{S}(r\mathbf{C}) \right\}.$$

Then the following *continuous approximation* of $\mathcal{S}^r(\mathbf{C})$ emerges in the large r limit:

$$\mathcal{S}^*(\mathbf{C}) = \{\mathbf{n} \in \mathbb{R}_+^N : A\mathbf{n} \leq \mathbf{C}\}. \tag{7}$$

3.2.1. Variational characterization of $\boldsymbol{\pi}$

To understand the structure of $\boldsymbol{\pi}$ for the scaled loss network with large r, we shall utilize a variational characterization of $\boldsymbol{\pi}$. This provides an alternative (and possibly simpler in exposition) approach to that utilized by Kelly (1986) to study $\boldsymbol{\pi}$ for the scaled network with large r.

Recall that the stationary distribution $\boldsymbol{\pi}$ of unscaled system

$$\boldsymbol{\pi}(\mathbf{n}) = \frac{1}{Z(\mathbf{C})} \exp\Big(Q(\mathbf{n})\Big), \quad \text{for } \ \mathbf{n} \in \mathcal{S}(\mathbf{C}),$$

where

$$Q(\mathbf{n}) = \sum_i n_i \log \nu_i - \log(n_i!).$$

Define $\mathcal{M}(\mathbf{C})$ as the space of probability distributions on $\mathcal{S}(\mathbf{C})$. Clearly, $\boldsymbol{\pi} \in \mathcal{M}(\mathbf{C})$. For $\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})$, define

$$\begin{aligned} F(\boldsymbol{\mu}) &\triangleq \sum_{\mathbf{n} \in \mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n}) Q(\mathbf{n}) - \sum_{\mathbf{n} \in \mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n}) \log \boldsymbol{\mu}(\mathbf{n}) \\ &= \mathbb{E}_{\boldsymbol{\mu}}[Q] \ + \ H_{\mathrm{ER}}(\boldsymbol{\mu}). \end{aligned} \tag{8}$$

Now we state the so called *variational* characterization of $\boldsymbol{\pi}$, which will be quite useful. This characterization essentially states that $\boldsymbol{\pi}$ is characterized uniquely as the maximizer of $F(\cdot)$ over $\mathcal{M}(\mathbf{C})$.

Lemma 1 (Variational characterization) *For all* $\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})$,

$$F(\boldsymbol{\pi}) \geq F(\boldsymbol{\mu})$$

where equality holds iff $\boldsymbol{\mu} = \boldsymbol{\pi}$*. Further,* $F(\boldsymbol{\pi}) = \log Z(\mathbf{C})$.

Proof. From the definition of $Q(\cdot)$, we have

$$Q(\mathbf{n}) = \log \boldsymbol{\pi}(\mathbf{n}) + \log Z(\mathbf{C}).$$

Consider the following sequence of inequalities, which essentially use Jensen's inequality together with the above definition of $Q(\cdot)$:

$$\begin{aligned}
F(\boldsymbol{\mu}) &= \sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n})Q(\mathbf{n}) - \sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n}) \log \boldsymbol{\mu}(\mathbf{n}) \\
&= \sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n})(\log \boldsymbol{\pi}(\mathbf{n}) + \log Z(\mathbf{C})) - \sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n}) \log \boldsymbol{\mu}(\mathbf{n}) \\
&= \sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n}) \left(\log \frac{\boldsymbol{\pi}(\mathbf{n})}{\boldsymbol{\mu}(\mathbf{n})}\right) + \log Z(\mathbf{C}) \\
&\le \log \left[\sum_{\mathbf{n}\in\mathcal{S}(\mathbf{C})} \boldsymbol{\mu}(\mathbf{n}) \frac{\boldsymbol{\pi}(\mathbf{n})}{\boldsymbol{\mu}(\mathbf{n})}\right] + \log Z(\mathbf{C}) \\
&= \log 1 + \log Z(\mathbf{C}) \\
&= F(\boldsymbol{\pi}).
\end{aligned}$$

The only inequality above is tight iff $\boldsymbol{\mu} = \boldsymbol{\pi}$. This concludes Lemma 1. □

3.2.2. Large network approximation

Now we use variational characterization of $\boldsymbol{\pi}$ (cf. Lemma 1) to establish its concentration property for large r. We shall start with some calculations about $\boldsymbol{\pi}$ for the scaled network with large r.

$\boldsymbol{\pi}$ *for large* r. To this end, recall that for the scaled system with parameter r, the feasibility region is $\mathcal{S}(r\mathbf{C})$. Equivalently, $\frac{1}{r}\mathbf{n} \in \mathcal{S}^r(\mathbf{C})$. Then, the stationary distribution of the scaled system with parameter r is equivalent to the distribution $\boldsymbol{\pi}^r$ on $\mathcal{S}^r(\mathbf{C})$ defined as: for $\mathbf{n} \in \mathcal{S}^r(\mathbf{C})$,

$$\begin{aligned}
\boldsymbol{\pi}^r(\mathbf{n}) &= \boldsymbol{\pi}(r\mathbf{n}) \\
&= \frac{1}{Z(r\mathbf{C})} \exp\Big(Q(r\mathbf{n})\Big). \qquad (9)
\end{aligned}$$

Now

$$\begin{aligned}
\exp\Big(Q(r\mathbf{n})\Big) &= \exp\Big(\sum_i rn_i \log r\nu_i - \sum_i \log((rn_i)!)\Big) \\
&= \exp\Big(r \log r \sum_i n_i + r \sum_i n_i \log \nu_i - \sum_i \log((rn_i)!)\Big). \qquad (10)
\end{aligned}$$

Stirling's approximation suggests that for any $m \in \mathbb{N}$

$$\log(m!) = m \log m - m + 0.5 \log m + 0.5 \log 2\pi + O\Big(1/m\Big). \tag{11}$$

Therefore,

$$\log((rn_i)!) = r \sum_i n_i \log rn_i - r \sum_i n_i + 0.5 \sum_i \log rn_i + O(1). \tag{12}$$

From (10) and (12), it follows that

$$\begin{aligned} \frac{1}{r} Q\big(r\mathbf{n}\big) &= \sum_i n_i \log \frac{\nu_i e}{n_i} + \frac{0.5}{r} \Big[\sum_i \log(rn_i)\Big] + O\big(1/r\big) \\ &= q(\mathbf{n}) + O\left(\frac{\log r}{r}\right), \end{aligned} \tag{13}$$

where

$$q(\mathbf{n}) = \sum_i n_i \log \frac{\nu_i e}{n_i}. \tag{14}$$

The $O(\cdot)$ term in (13) holds uniformly for all $\mathbf{n} \in \mathcal{S}^r(\mathbf{C})$ or more generally for any $\mathbf{n} \in \mathcal{S}^*(\mathbf{C})$.

Concentration of $\boldsymbol{\pi}^r$*.* Given above calculations, we obtain the following concentration property of $\boldsymbol{\pi}^r$ for large r.

Lemma 2 *Given* $\varepsilon > 0$*, define the set*

$$A_\varepsilon = \Big\{\mathbf{n} \in \mathcal{S}^*(\mathbf{C}) : \|\mathbf{n} - \mathbf{n}^*\| > \varepsilon\Big\},$$

where $\mathbf{n}^* = \operatorname{argmax}_{\mathbf{n} \in \mathcal{S}^*(\mathbf{C})} q(\mathbf{n})$*. Then*

$$\boldsymbol{\pi}^r(A_\varepsilon) = O\Big(\varepsilon^{-2} \frac{\log r}{r}\Big). \tag{15}$$

Lemma 2 suggests that most of the probability mass under $\boldsymbol{\pi}^r$ for large r is concentrated around the maximal element with respect to q. As per (13), this is *close* to the mode of the distribution $\boldsymbol{\pi}^r$ for large r.

Proof. (*Lemma 2*). From the definition of $q(\cdot)$, it can be verified that this is a strictly concave function on the set $\mathcal{S}^*(\mathbf{C})$. Moreover, the constraint set $\mathcal{S}^*(\mathbf{C})$ is closed and convex. Hence, there exists a unique optimal solution $\mathbf{n}^*$ of the optimization problem

$$\text{maximize } q(\mathbf{n}) \quad \text{over} \quad \mathbf{n} \in \mathcal{S}^*(\mathbf{C}).$$

By optimality and uniqueness of $\mathbf{n}^*$, it follows that for any $\mathbf{n} \in \mathcal{S}^*(\mathbf{C})$

$$\nabla q(\mathbf{n}^*)^T(\mathbf{n} - \mathbf{n}^*) \le 0. \tag{16}$$

By a Taylor expansion, the value of $q(\cdot)$ at $\mathbf{n} \in \mathcal{S}^*(\mathbf{C})$ around $\mathbf{n}^*$ can be represented as:

$$q(\mathbf{n}) = q(\mathbf{n}^*) + \nabla q(\mathbf{n}^*)^T(\mathbf{n} - \mathbf{n}^*) + (\mathbf{n} - \mathbf{n}^*)^T\nabla^2 q(\mathbf{z})(\mathbf{n} - \mathbf{n}^*), \tag{17}$$

where $\mathbf{z} = \alpha\mathbf{n}^* + (1-\alpha)\mathbf{n}$, for some $\alpha \in [0,1]$. Using the optimality condition, we have

$$q(\mathbf{n}) \le q(\mathbf{n}^*) + (\mathbf{n} - \mathbf{n}^*)^T\nabla^2 q(\mathbf{z})(\mathbf{n} - \mathbf{n}^*). \tag{18}$$

Next, in order to evaluate the RHS in (18), we shall compute the Hessian $\nabla^2 q(\mathbf{z})$. For this, recall that

$$q(\mathbf{n}) = \sum_i n_i \log \frac{\nu_i e}{n_i}.$$

Therefore, the Hessian $\nabla^2 q(\cdot)$ is a diagonal matrix of the form

$$\begin{aligned} \nabla^2 q(\mathbf{n}) &= \left[\frac{\partial^2 q(\mathbf{n})}{\partial n_i \partial n_j}\right] \\ &= \operatorname{diag}\left[-\frac{1}{n_1}, \ldots, -\frac{1}{n_N}\right]. \end{aligned} \tag{19}$$

For any $\mathbf{z} \in \mathcal{S}^*(\mathbf{C})$, we have the bound that

$$|\mathbf{z}| \le |\mathbf{C}| \quad \text{and} \quad \mathbf{z} \in \mathbb{R}_+^N.$$

Using this bound, the definition of the Hessian (19) and (18), we obtain

$$\begin{aligned} q(\mathbf{n}) &\le q(\mathbf{n}^*) - \sum_i \frac{(n_i - n_i^*)^2}{z_i} \\ &\le q(\mathbf{n}^*) - \frac{1}{|\mathbf{C}|}\left(\sum_i (n_i - n_i^*)^2\right) \\ &= q(\mathbf{n}^*) - \frac{1}{|\mathbf{C}|}\left\|\mathbf{n} - \mathbf{n}^*\right\|^2. \end{aligned} \tag{20}$$

Now, by Markov's inequality and (20), we have

$$\begin{aligned}\varepsilon^2 \boldsymbol{\pi}^r(A_\varepsilon) &\le \mathbb{E}_{\boldsymbol{\pi}^r}\left[\|\mathbf{n}-\mathbf{n}^*\|^2\right] \\ &\le |\mathbf{C}|\Big(q(\mathbf{n}^*) - \mathbb{E}_{\boldsymbol{\pi}^r}[q(\mathbf{n})]\Big), \qquad (21)\end{aligned}$$

which together with (13) yields

$$\varepsilon^2 \boldsymbol{\pi}^r(A_\varepsilon) \le \frac{|\mathbf{C}|}{r}\Big(Q(r\mathbf{n}^*) - \mathbb{E}_{\boldsymbol{\pi}^r}[Q(r\mathbf{n})]\Big) + O\left(\frac{\log r}{r}\right). \qquad (22)$$

Consider a dirac distribution $\boldsymbol{\mu}$ on $\mathcal{S}(r\mathbf{C})$ that has all its mass on $r\mathbf{n}^*$. Then, by Lemma 1, it follows that

$$\begin{aligned}Q(r\mathbf{n}^*) &= F(\boldsymbol{\mu}) \\ &\le F(\boldsymbol{\pi}^r) \\ &= \mathbb{E}_{\boldsymbol{\pi}^r}[Q(r\mathbf{n})] + H_{\mathrm{ER}}(\boldsymbol{\pi}^r). \qquad (23)\end{aligned}$$

In the above, with an abuse of notation, $\boldsymbol{\pi}^r$ is thought of as defined on $\mathcal{S}(r\mathbf{C})$ rather than the normalized version $\mathcal{S}^r(\mathbf{C})$. Now the support of $\boldsymbol{\pi}^r$ is over at most $O(r^N)$ elements. Therefore, by the standard bounds on entropy, $H_{\mathrm{ER}}(\boldsymbol{\pi}^r) = O(\log r)$. Therefore, we can conclude Lemma 2 using (22) and (23).

We remark that use of $\mathbf{n}^*$ in (23) assumes that it belong to $\mathcal{S}^r(\mathbf{C})$. However, $\mathbf{n}^*$ is only known to belongs to $\mathcal{S}^*(\mathbf{C})$. To fix this, one must use $\tilde{\mathbf{n}}^*$ which is closest to $\mathbf{n}^*$ in $\mathcal{S}^r(\mathbf{C})$ so that $\|\mathbf{n}^* - \tilde{\mathbf{n}}^*\| = O(1/r)$. For such an $\tilde{\mathbf{n}}^*$ essentially the same argument holds since $|q(\mathbf{n}^*) - q(\tilde{\mathbf{n}}^*)|$ can be easily shown to be $O(1/r)$.□

3.3. Message-passing: optimization

The concentration property established in Lemma 2 suggests that most of the mass of the scaled system concentrates around $\mathbf{n}^*$, the maximizer of $q(\cdot)$ for large r. Our interest is in computing loss probabilities, L_i or equivalently $\mathbb{E}[n_i]$ for $1 \le i \le N$. First, we state a result that suggests that L_i can be computed by knowing n_i^*. Then, we present a message-passing algorithm based on dual co-ordinate descent method to compute $\mathbf{n}^*$.

3.3.1. Loss probability for large network

The following result, which states that the loss rates are well approximated through scaled mode of the distribution, was established by Kelly (1986). Here we provide an alternative derivation by Jung et al. (2008) using variational characterization of $\boldsymbol{\pi}$.

Theorem 1 (Large network approximation) *Consider stochastic loss network with scaling parameter r. Let L_i^r be the exact loss probability of route $i \in \mathcal{R}$. Then*

$$\left|(1 - L_i^r) - \frac{n_i^*}{\nu_r}\right| = O\left(\sqrt{\frac{\log r}{r}}\right). \tag{24}$$

Proof. The proof follows immediately using Lemma 2. Specifically, in Lemma 2, use $\varepsilon_k = k\sqrt{\frac{\log r}{r}}$ for the value of ε. Then, we obtain from (15) that

$$\boldsymbol{\pi}^r\left(|n_i - n_i^*| > \varepsilon_k\right) \;=\; O\left(\frac{1}{k^2}\right), \tag{25}$$

which immediately implies

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi}^r}\left[|n_i - n_i^*|\right] &= O\left(\sqrt{\frac{\log r}{r}}\right) \times O\left(\sum_k \frac{1}{k^2}\right)\\ &= O\left(\sqrt{\frac{\log r}{r}}\right).\end{aligned} \tag{26}$$

Due to scaling of $\boldsymbol{\nu}$ and $\mathbf{C}$ by r under the large network scaling and by linearity of expectation, it can be argued that

$$1 - L_i^r = \frac{\mathbb{E}_{\boldsymbol{\pi}^r}[n_i]}{\nu_i}.$$

Therefore, it follows that

$$\begin{aligned}\left|(1 - L_i^r) - \frac{n_i^*}{\nu_i}\right| &\leq \frac{\mathbb{E}_{\boldsymbol{\pi}^r}[|n_i - n_i^*|]}{\nu_i}\\ &= O\left(\frac{1}{\nu_i}\sqrt{\frac{\log r}{r}}\right).\end{aligned} \tag{27}$$

This completes the proof of Theorem 1. □

3.3.2. Message-passing via dual co-ordinate descent

Theorem 1 suggests that for large r, good approximation of L_i^r can be obtained by knowing n_i^* for $1 \leq i \leq N$. This requires solving the optimization problem:

$$\begin{aligned} &\text{maximize} \quad q(\mathbf{n}) = \sum_i n_i \log \frac{\nu_i e}{n_i} \quad \text{over} \quad \mathbf{n} \in \mathbb{R}_+^N \\ &\text{subject to} \quad A\mathbf{n} \leq \mathbf{C}. \end{aligned} \tag{28}$$

As discussed earlier, the objective of optimization problem (28) is strictly concave over its convex feasible region. Therefore, unique optimum $\mathbf{n}^*$ is achieved. The Lagrangian dual of this optimization problem is given by:

$$\text{minimize} \quad \sum_i \nu_i \exp\Big(-\sum_j y_j A_{ji}\Big) + \sum_j y_j C_j \tag{29}$$

$$\text{subject to} \quad \mathbf{y} \in \mathbb{R}_+^M. \tag{30}$$

In above, vector of dual variables $\mathbf{y} \in \mathbb{R}_+^M$ with y_j corresponding to constraint $\sum_i A_{ji} n_i \leq C_j$ for $1 \leq j \leq M$. Define dual objective function as $g : \mathbb{R}_+^M \to \mathbb{R}$ as

$$g(\mathbf{y}) = \sum_i \nu_i \exp\Big(-\sum_j y_j A_{ji}\Big) + \sum_j y_j C_j. \tag{31}$$

By Slater's condition, strong duality holds and hence the optimal cost of primal optimization (28) and dual optimization (29) are the same. The Karush-Kuhn-Tucker conditions suggest that the pair of primal and dual optima, $(\mathbf{n}^*, \mathbf{y}^*)$, satisfy the following:

(a) For each link j,

$$\frac{\partial g(\mathbf{y}^*)}{\partial y_j} = 0 \quad \text{or} \quad y_j^* = 0 \;\&\; \frac{\partial g(\mathbf{y}^*)}{\partial y_j} \leq 0.$$

Equivalently,

$$\sum_i A_{ji} \nu_i \exp\Big(-\sum_\ell y_\ell^* A_{\ell i}\Big) = C_j \;\&\; y_j^* > 0,$$

$$\text{or,} \quad \sum_i A_{ji} \nu_i \exp\Big(-\sum_j y_\ell^* A_{\ell i}\Big) \leq C_j \;\&\; y_j^* = 0.$$

(b) For each route $i \in \mathcal{R}$,

$$n_i^* = \nu_i \exp\Big(-\sum_j y_j^* A_{ji}\Big).$$

Here we note that since the optimal solution $\mathbf{n}^*$ of optimization problem (28) is unique, any dual optimum $\mathbf{y}^*$ will give the same $\mathbf{n}^*$ by (b). The above conditions suggest the following approach: obtain a dual optimal solution, $\mathbf{y}^*$ and use it to obtain $\mathbf{n}^*$; eventually this will lead to loss probability as

$$\begin{aligned} 1 - L_i &= \frac{n_i^*}{\nu_i} \\ &= \exp\Big(-\sum_j y_j^* A_{ji}\Big). \end{aligned} \tag{32}$$

Next, we describe message-passing algorithm based on dual co-ordinate descent algorithm for obtaining $\mathbf{y}^*$.

The basic idea behind a co-ordinate descent algorithm is quite simple. In an iterative manner, each co-ordinate's value is changed one at a time, so as to minimize the value of objective function as much as possible. Such a myopic algorithm may not even converge in general. However, in the setup considered here, the algorithm always converges to an optimal solution. The precise description of the algorithm is as follows.

Dual Co-ordinate Descent.

---

1. Denote by t the iteration of the algorithm. Initially, $t = 0$, $y_j^{(0)} = 1$ for all $1 \leq j \leq M$.

2. In iteration $t + 1$, determine $\mathbf{y}^{(t+1)}$ as follows:

   (a) Choose coordinates from $1, \ldots, M$ in a round-robin manner.

   (b) Update $y_j^{(t+1)}$ as the solution of the following equation: let x be such that

$$g_j^{(t)}(x) = \min\left\{C_j, g_j^{(t)}(1)\right\},$$

where $g_j^{(t)}(x) = \sum_i A_{ji}\nu_i \exp\Big(-\sum_\ell A_{\ell i} y_\ell\Big)$ with

$$y_\ell = \begin{cases} y_\ell^{(t+1)} & \text{for } \ell < j, \\ x & \text{for } \ell = j, \\ y_\ell^{(t)} & \text{for } \ell > j. \end{cases}$$

Thus, $g_j^{(t)}(x)$ is evalatuion of partial derivative of $g(\cdot)$ with respect to the jth co-ordinate with values of components $< j$ being from iteration $t+1$, values of components $> j$ from iteration t, and component j being x.

3. Upon convergence (per appropriate stopping conditions), denote the resulting values by y_j^* for $1 \le j \le M$. Compute the loss probabilities L_i^*, for $i \in \mathcal{R}$, as

$$1 - L_i^* = \exp\Big(-\sum_j A_{ji} y_j^*\Big).$$

The algorithm described is iterative and in each iteration, the components of $\mathbf{y}$ are updated one-by-one. To update variable associated with a link j, y_j, the information needed is the value of all the variables y_ℓ such that there is a flow i that passes through link j and link ℓ simultaneously, i.e. $A_{ji}, A_{\ell i} \neq 0$. This creates an *edge* between components of variables $\mathbf{y}$. More formally, consider a graph with M nodes with node j, $1 \le j \le M$ correspond to y_j. In this graph, there is an edge between nodes corresponding to variables y_j and y_ℓ if there is a flow i that passes through links j and ℓ. Then the above described algorithm can be imagined as passing 'messages' between the nodes of this graph to perform iterative computation. Therefore, we call it a message-passing algorithm.

3.3.3. Convergence, correctness of dual co-ordinate descent

Theorem 2 (Convergence & Correctness) *Given a loss network with routing matrix A, link capacities $\mathbf{C}$ and rate vector $\boldsymbol{\nu}$, let $\mathbf{y}^{(t)}$ be the vector of dual variables produced by the dual co-ordinate descent algorithm at the end of iteration t. Assume that each link is utilized by some route in the matrix A, that is $\sum_i A_{ji} > 0$ for all j. Let $\mathcal{Y}^*$ be the set of dual optimal solutions. Then,*

$$d(\mathbf{y}^{(t)}, \mathcal{Y}^*) \le \alpha \exp\left(-\beta t\right),$$

where α, β are positive constants that depend on the problem parameters, and $d(\cdot, \mathcal{Y}^*)$ is distance to the set $\mathcal{Y}^*$. Further, for the primal optimal $\mathbf{n}^*$

$$\left\| \left(\nu_i \exp\Big(-\sum_j y_j^{(t)} A_{ji} \Big) \right)_{1 \le i \le N} - \mathbf{n}^* \right\|_2 \le \alpha' \exp\left(-\beta' t\right),$$

for some positive constants α' and β'.

The proof of the convergence and correctness of the "round-robin" coordinate descent algorithm follows from a result of Luo and Tseng (1992). We first recall their precise result and then show how it implies Theorem 2.

In order to state the result in Luo and Tseng (1992), some additional notation needs to be introduced. Consider a real valued function $\phi : \mathbb{R}^n \to \mathbb{R}$ defined as

$$\phi(\mathbf{x}) = \psi(E\mathbf{x}) + \sum_{i=1}^{n} w_i x_i, \tag{33}$$

where $E \in \mathbb{R}^{m \times n}$ is an $m \times n$ matrix with no zero column (i.e., all coordinates of $\mathbf{x}$ are useful), $\mathbf{w} = (w_i) \in \mathbb{R}^n$ is a given fixed vector, and $\psi : \mathbb{R}^m \to \mathbb{R}$ is a strongly convex function on its domain

$$D_\psi = \{\mathbf{z} \in \mathbb{R}^m : \psi(\mathbf{z}) \in (-\infty, \infty)\}.$$

We have D_ψ being open and let ∂D_ψ denote its boundary. We also have that, along any sequence $\mathbf{z}_k$ such that $\mathbf{z}_k \to \partial D_\psi$ (i.e., approaches the boundary of D_ψ), $\psi(\mathbf{z}_k) \to \infty$. The goal is to solve the optimization problem

$$\begin{aligned} \text{minimize} \quad & \phi(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{34}$$

where we assume that $\mathcal{X}$ is of box-type, i.e.,

$$\mathcal{X} = \prod_{i=1}^{n} [\ell_i, u_i], \quad \ell_i, u_i \in \mathbb{R} \cup \infty.$$

Let $\mathcal{X}^*$ be the set of all optimal solutions of the problem (34). The "round-robin" or "cyclic" co-ordinate descent algorithm for this problem has the following convergence property, as proved in Theorem 6.2 of Luo and Tseng (1992).

Lemma 3 *There exist constants α_0 and β_0 which may depend on the problem parameters in terms of $\psi, E, \mathbf{w}$ such that starting from the initial value $\mathbf{x}^0$, we have in iteration t of the algorithm*

$$d(\mathbf{x}^t, \mathcal{X}^*) \leq \alpha_0 \exp\left(-\beta_0 t\right) d(\mathbf{x}^0, \mathcal{X}^*).$$

Here, $d(\cdot, \mathcal{X}^*)$ denotes distance to the optimal set $\mathcal{X}^*$.

Proof. (of Theorem 2) Note that the optimization problem of interest (29) has very similar form to that of (34) including a box-type domain set: $\mathbb{R}^N$. The dual objective function $g(\cdot)$ has the form

$$g(\mathbf{y}) = \sum_i \nu_i \exp\left(-\sum_j y_j A_{ji}\right) + \sum_j y_j C_j.$$

Assuming that each link is utilized by some route in the matrix A, it can be easily verified that $g(\cdot)$ can be written in the desired form (33) of the cost function of the optimization problem (34). Therefore, the setup of Theorem 2 satisfies the conditions of Lemma 3 and it follows that

$$d(\mathbf{y}^{(t)}, \mathcal{Y}^*) \leq \alpha \exp\left(-\beta t\right),$$

for some positive constants α, β.

As discussed earlier, due to strict concavity of objective and bounded convex domain, the primal optimization problem (28) has a unique solution, $\mathbf{n}^*$. And for $1 \leq i \leq N$,

$$n_i^* = \nu_i \exp\Big(-\sum_j y_j^* A_{ji}\Big).$$

The map $\mathbf{y} \to \Big(\exp\Big(-\sum_j y_j A_{ji}\Big)\Big)_{1\leq i \leq N}$ is from $\mathbb{R}_+^M \to [0,1]^N$ which is uniformly Lipschitz continous. Therefore, it immediately follows that

$$\left\| \left(\nu_i \exp\Big(-\sum_j y_j^{(t)} A_{ji}\Big)\right)_{1\leq i\leq N} - \mathbf{n}^* \right\|_2 \leq \alpha' \exp\left(-\beta' t\right),$$

for some positive constants α' and β'. □

3.4. Message-passing: Erlang approximation

The dual co-ordinate descent message-passing algorithm essentially obtains asymptotically correct loss probabilities with respect to large network scaling. In order to do so, it essentially makes the approximation that average number of calls on each route is equal to the number of calls with respect to its mode, i.e. $\mathbb{E}[n_i] \approx n_i^*$. A classical Erlang approximation, derived from very different considerations, does have very similar properties that we shall discuss here.

The Erlang formula for a single-link, single-route network with capacity C and arrival rate ν states that the loss probability, denoted by $E(\nu, C)$, is given by

$$E(\nu, C) = \frac{\nu^C}{C!}\left[\sum_{k=0}^{C} \frac{\nu^k}{k!}\right]^{-1}. \tag{35}$$

Based on this simple formula, the Erlang fixed-point approximation for multi-link, multi-route networks arose from the hypothesis that calls are lost due to *independent* blocking events on each link in the route. More formally, this hypothesis implies that the loss probabilities of routes $\mathbf{L} = (L_1, \ldots, L_N)$ and blocking probabilities of links $\mathbf{E} = (E_1, \ldots, E_M)$ satisfy the set of *fixed-point* equations

$$\begin{aligned} E_j &= E(\rho_j, C_j), \\ \rho_j &= \frac{1}{1 - E_j}\left[\sum_i \nu_i A_{ji} \prod_{\ell} (1 - E_\ell)^{A_{\ell i}}\right], \\ 1 - L_i &= \prod_j (1 - E_j)^{A_{ji}}, \end{aligned} \tag{36}$$

for $j = 1, \ldots, M$ and $i \in \mathcal{R}$. Here recall that a call on route i requires A_{jr} units of capacity on link j. A natural iterative algorithm that attempts to obtain a solution to the above fixed-point equations is as follows:

Erlang fixed-point approximation.

---

1. Denote by t the iteration of the algorithm, with $t = 0$ initially. Start with $E_j^{(0)} = e^{-1}$ for all $1 \leq j \leq M$.

2. In iteration $t+1$, update $E_j^{(t+1)}$ according to

$$E_j^{(t+1)} \;=\; E(\rho_j^{(t)}, C_j),$$

where

$$\rho_j^{(t)} \;=\; (1-E_j^{(t)})^{-1}\sum_i \nu_i A_{ji}\prod_\ell (1-E_\ell^{(t)})^{A_{\ell i}}.$$

3. Upon convergence per appropriate stopping conditions, denote the resulting values by $E_j^{\mathcal{E}}$ for $1 \leq j \leq M$. Compute the loss probabilities from the Erlang fixed-point approximation, $L_i^{\mathcal{E}}$, $i \in \mathcal{R}$, as

$$1 - L_i^{\mathcal{E}} = \prod_j (1-E_j^{\mathcal{E}})^{A_{ji}}.$$

3.4.1. Existence, correctness of Erlang approximation

Apriori it is not clear if there exists a solution to Erlang fixed-point approximation (36) for any given loss network. And even if it exists, it may not provide good approximation to loss probabilities. Both of these questions have affirmative answers (see Kelly, 1986, Theorems 5.1, 5.3). To start with, each iteration of the iterative procedure to find Erlang fixed point approximation can be viewed as a continuous mapping from $[0,1]^M \to [0,1]^M$. Therefore, by an application of Brouwer's fixed point theorem, existence of fixed point, i.e. solution to (36) follows. However, this is hardly insightful in studying its properties. The following result of Kelly (1986) provides it's relation to a useful optimization problem.

Theorem 3 (Existence & Uniqueness) *There exists a unique vector* $\mathbf{E} \in [0,1]^M$ *that satisfies equations* (36).

Proof. The existence and uniqueness of $\mathbf{E} \in [0,1]^M$ satisfying (36) will be established by showing it to be a unique solution of certain convex minimization problem. To that end, define a function $U : \mathbb{R}_+^2 \to \mathbb{R}$ as follows:

$$\begin{aligned} &U(y,C) = \nu \exp(-y), \\ &\quad \text{where } \;\nu\; \text{ is s.t. } \; 1 - E(\nu, C) = \exp(-y). \end{aligned} \tag{37}$$

Note that U is well defined because, $E(\cdot, C) : \mathbb{R}_+ \to [0,1]$ is a strictly decreasing function ranging from 0 to 1 as its argument varies from 0 to ∞. Therefore, for each $y \geq 0$, there exists a unique value of corresponding ν (given C). Also observe that $U(\cdot, C)$ is a strictly increasing function. Therefore, $\int_0^y U(z,C)dz$ is a strictly convex function. Consider the optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_i \nu_i \exp\Big(-\sum_j A_{ji} y_j\Big) + \sum_j \int_0^{y_j} U(z, C_j)dz \qquad (38) \\ \text{subject to} \quad & \mathbf{y} \in \mathbb{R}_+^M. \end{aligned}$$

The objective function of this optimization problem is a strictly convex function over a convex set. Further, as $|\mathbf{y}| \to \infty$, the value of the objective function goes to ∞. Therefore, unique minimum must be achieved. Since the objective function is differentiable, stationarity conditions can be obtained as

$$\sum_i A_{ji} \nu_i \exp\big(-\sum_\ell y_\ell A_{\ell i}\big) = U(y_j, C_j), \quad \text{for all } \ 1 \leq j \leq M. \qquad (39)$$

Now suppose there exists a solution, say $\mathbf{E}$, to (36). We shall show that under one-to-one transformation $E_j = 1 - \exp(-y_j)$, the corresponding $\mathbf{y}$ satisfies (39). This will imply the desired claim that $\mathbf{E}$ uniquely satisfies (36).

Now with this transformation of $\mathbf{E} \to \mathbf{y}$, (36) becomes

$$E\Big(\exp(y_j)\Big[\sum_i A_{ji} \exp\Big(-\sum_\ell y_\ell A_{\ell i}\Big)\Big], C_j\Big) = 1 - \exp(-y_j), \quad \text{for all } 1 \leq j \leq M. \qquad (40)$$

For given j, in defining function $U(y_j, C_j)$ set

$$\nu = \exp(y_j)\Big[\sum_i A_{ji} \exp\Big(-\sum_\ell y_\ell A_{\ell i}\Big)\Big].$$

Then it immediately follows that $E(\nu, C_j) = 1 - \exp(-y_j)$ and collectively they satisfy (39). This completes the proof of Theorem 3. □

As per proof of Theorem 3, the solution to the Erlang fixed point equations, say $\mathbf{E}$, is related to the solution of optimization problem (38), say $\mathbf{y}$, through the transformation $1 - E_j = \exp(-y_j)$ for $1 \leq j \leq M$. Notice the syntactic similarity between estimation of loss probability as per dual co-ordinate descent (step 3) and Erlang approximation (step 3) under this transformation. Therefore, to establish asymptotic correctness of Erlang approximation, it is sufficient to establish relation between $\mathbf{E}$ (or transformed $\mathbf{y}$) and the optimal solution, $\mathbf{y}^*$, of dual optimization problem (29). This is precisely established by Kelly (1986). Here we state some key intuition.

Note that the only difference between optimization problems (29) and (38) is in the second part of the objective function: in one case, it is $\sum_j y_j C_j$ while in the other case it is $\sum_j \int_0^{y_j} U(z, C_j) dz$. If $U(z, C_j) \approx C_j$, then indeed both of them become identical. This approximation is established for large values of C_j (see Kelly, 1986, Lemma 5.2). Therefore, in the large network scaling, the Erlang approximation becomes similar to the approximation based on dual co-ordinate descent.

3.5. Message-passing: variational approximation

The primary reason for dual co-ordinate descent or Erlang approximation to a provide good estimate for the loss probability arises from the approximation $\mathbb{E}[n_i] \approx n_i^*$ based on large network scaling, where $\mathbf{n}^*$ is the mode of the distribution. This means that in the variational representation of stationary distribution given in Lemma 1, these approaches completely ignore the effect of the *entropy* term. Indeed the error induced by doing so is of order $O(\log r/r)$ for large network scaling. However, for relatively smaller networks or for scenarios where more accurate evaluation of the loss probability is desired, one needs to account for the effect of the entropy term on the loss probability. Of course, as discussed in the beginning, precise evaluation (solving MARG exactly) is computationally hard. Therefore, the goal is to balance the complexity of the algorithm and quality of solution (especially in its ability to capture the effect of entropy term). This is precisely done by algorithms based on variational approximation methods – they approximate entropy term in the variational characterization (cf. Lemma 1) by utilizing tractable surrogates of entropy. In what follows, we shall discuss two such approximations: the mean-field method and belief propagation. Both of them utilize tractable approximations that yield to message-passing evaluation of the loss probabilities.

3.5.1. Mean-field method

The variational characterization (Lemma 1) suggests that the approximation by mode, $\mathbf{n}^*$, leads to essentially a lower bound on $F(\boldsymbol{\pi}) = \log Z(\mathbf{C})$. We start with a naive method, which we shall call the mean-field method after the classical statistical physics approach, to obtain an upper bound on $F(\boldsymbol{\pi})$ with essentially the same computational cost compared to the dual co-ordinate descent or Erlang approximation. As we shall see, in a nutshell, the mean-field method boils down to running a dual co-ordinate descent algorithm for an optimization problem (43) which is syntactically very similar to (29) but with a different objective function.

Consider the following: using Lemma 1 with $Q(\mathbf{n}) = \sum_i n_i \log \nu_i - \log(n_i!)$,

$$\begin{aligned}
\log Z(\mathbf{C}) &= F(\boldsymbol{\pi}) \\
&= \mathbb{E}_{\boldsymbol{\pi}}[Q(\mathbf{n})] + H_{\mathrm{ER}}(\boldsymbol{\pi}) \\
&= \sup_{\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})} \mathbb{E}_{\boldsymbol{\mu}}[Q(\mathbf{n})] + H_{\mathrm{ER}}(\boldsymbol{\mu}) \\
&\leq \sup_{\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})} \mathbb{E}_{\boldsymbol{\mu}}[Q(\mathbf{n})] + \sum_i H_{\mathrm{ER}}(\boldsymbol{\mu}_i), \qquad (41)
\end{aligned}$$

where by $\boldsymbol{\mu}_i$ we mean marginal distribution of n_i with respect to $\boldsymbol{\mu}$; the last inequality follows because entropy of a joint distribution is no more than the sum of the entropies of the individual marginals. Indeed, the approximation is tight when $\boldsymbol{\pi}$ itself satisfies indepedence across its marginals, e.g. under assumption that $\boldsymbol{\pi}$ is dirac with mode as its support. In (41), the choice of $\boldsymbol{\mu}$ is restricted to $\mathcal{M}(\mathbf{C})$, the space of all feasible distributions, which is intractable to work with. Now the approximate variational characterization in (41) utilizes only marginal distributions induced by a given $\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})$. Therefore one way to achieve tractability is to consider a (tractable) relaxation of the 'marginal polytope' induced by $\mathcal{M}(\mathbf{C})$. Here by marginal polytope we mean the set of all marginal distributions $(\boldsymbol{\mu}_i)_{1 \leq i \leq N}$ that are induced by feasible distributions $\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})$. In what follows, we shall define such a relaxation by means of inequality constraints utilizing only marginal information.

Let $\hat{\mu}_{(i,k)}$ denote the probability of route i having k active calls for $0 \leq k \leq |\mathbf{C}|$. Note that at most K_i calls can be active for route i, where

$$K_i \triangleq \left\lfloor \left(\min_j \frac{C_j}{A_{ji}} \right) \right\rfloor.$$

Define a matrix $\hat{A}$ as an extension of A as follows. For each A_{ji}, create entries $\hat{A}_{j(i,k)}$ with $1 \leq k \leq |\mathbf{C}|$ as

$$\hat{A}_{j(i,k)} = k A_{ji}.$$

Thus, $\hat{A}$ is $M \times N|\mathbf{C}|$ matrix. For simplicity, we shall assume that columns are organized so that all $|\mathbf{C}|$ columns corresponding to a given route i are adjacent. Now consider a relaxation $\mathcal{M}^1(\mathbf{C})$ of $\mathcal{M}(\mathbf{C})$ defined as

$$\mathcal{M}^1(\mathbf{C}) = \{\hat{\boldsymbol{\mu}} = (\mu_{(i,k)}) : \sum_k \hat{\mu}_{(i,k)} = 1,\ \hat{\mu}_{(i,k)} \geq 0,\ \hat{A}\hat{\boldsymbol{\mu}} \leq \mathbf{C}\}.$$

In above, $\mu_{(i,k)}$ represents the marginal probability of route i having k active calls. Now $\hat{A}\hat{\boldsymbol{\mu}} \leq \mathbf{C}$ represents that the feasibility constraints must be satisfied on average (this should explain our reason for defining $\hat{A}$). Indeed, for any distribution $\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})$, its corresponding marginals $\hat{\boldsymbol{\mu}}$ must satisfy this constraint. However, given $\hat{\boldsymbol{\mu}} \in \mathcal{M}^1(\mathbf{C})$, it is not clear if there exists $\boldsymbol{\mu} \in \mathcal{M}(\mathbf{C})$ with $\hat{\boldsymbol{\mu}}$ as its marginals. The answer is always yes when all the extreme points of the marginal polytope $\mathcal{M}^1(\mathbf{C})$ belong to $\mathcal{S}(\mathbf{C})$. That is, the marginal constraints induced polytope (here $\mathcal{M}^1(\mathbf{C})$) and the marginal polytope are the same if the extreme points of the earlier are contained in the support of the space of distributions of interest. In such cases, $\mathcal{M}^1(\mathbf{C})$ is not a relaxation; but in general it is a relaxation (hence leads to weaker upper bound on $\log Z(\mathbf{C})$). Finally, for ease of notations, define

$$w_{ik} = \begin{cases} \log \frac{\nu_i^k}{k!}, & 0 \leq k \leq K_i \\ -\infty, & K_i < k \leq |\mathbf{C}|. \end{cases}$$

With these notations, the following optimization serves as an upper bound on $\log Z(\mathbf{C})$ for any loss network:

$$\begin{aligned} \text{maximize} \quad & \sum_{1 \leq i \leq N} \sum_{0 \leq k \leq |\mathbf{C}|} w_{ik}\hat{\mu}_{(i,k)} - \hat{\mu}_{(i,k)} \log \hat{\mu}_{(i,k)} \\ \text{over} \quad & \hat{\boldsymbol{\mu}} = (\hat{\mu}_{(i,k)}) \in \mathbb{R}_+^{N(|\mathbf{C}|+1)}, \\ \text{subject to} \quad & \hat{A}\hat{\boldsymbol{\mu}} \leq \mathbf{C}, \\ & \sum_k \hat{\mu}_{(i,k)} = 1, \quad \text{for all} \quad 1 \leq i \leq N. \end{aligned} \tag{42}$$

Observe that optimization problem (42) has strictly concave objective with bounded convex feasible region. Therefore, it achieves a unique optimum. Let us consider the Lagrangian dual of the optimization problem (42) with dual variables $\mathbf{y} = (y_j) \in \mathbb{R}_+^M$ corresponding to capacity constraints. It can be checked that its dual objective $\hat{g} : \mathbb{R}_+^M \to \mathbb{R}$ is

$$\hat{g}(\mathbf{y}) \;=\; \sum_i \log\Big(\sum_k \exp\Big(w_{ik} - \sum_\ell y_\ell \hat{A}_{\ell ik}\Big)\Big) + \sum_j y_j C_j.$$

And the dual optimization problem is given by

$$\text{minimize} \quad \hat{g}(\mathbf{y}) \quad \text{over} \quad \mathbf{y} \in \mathbb{R}_+^M. \tag{43}$$

By definition, the dual objective $\hat{g}(\cdot)$ is a convex function. By Slater's condition strong duality holds. Therefore, the pair of optimal solutions $\hat{\boldsymbol{\mu}}^*$ and $\mathbf{y}^*$ satisfy the Karush-Kuhn-Tucker conditions. Specifically, given $\mathbf{y}^*$, the $\hat{\boldsymbol{\mu}}^*$ can be recovered as follows: for each i,

$$\hat{\mu}^*_{(i,k)} \propto \exp\Big(w_{ik} - \sum_\ell y^*_\ell \hat{A}_{\ell ik}\Big); \quad \sum_k \hat{\mu}^*_{(i,k)} = 1. \tag{44}$$

To obtain a message-passing algorithm for finding $\mathbf{y}^*$, as before we can use dual co-ordinate descent. The algorithm will be very similar to the one described to solve optimization (29) with only one difference: use of $\hat{g}$ in place of g. Indeed, this makes the iterative steps somewhat more involved. However, it still retains the distributed, iterative structure. Further, due to similarity of structure the convergence property of dual co-ordinate descent carries over as well. In summary, the mean-field method can lead to an upper bound on partition function $\log Z(\mathbf{C})$ in contrast to lower bound obtained through approximation based on the mode of the distribution.

Some remarks are in order. Observe similarities as well as difference with the earlier approaches. The optimization problem (43) has very similar 'form' compared to the optimization problem (29). The primary difference arises from the fact that mean-field approach inherently allows for richer parametrization for estimating marginal of each call type compared to single parametrization in earlier approaches. However, it should be noted that in both of the dual formulations (29) and (43), the effective parameterization is the same and their objective functions are quite similar as well. This leads us to speculate that mean-field method may be getting *more out of* similar effort. Finally, we believe that under large network scaling the mean-field method based on dual co-ordinate descent should be asymptotically correct in estimating loss rates.

3.5.2. Belief propagation: pair-wise loss network

The mean-field approach described earlier provides a valid upper bound by optimizing approximate variational form over a relaxation of the marginal polytope. The approximate variational form used independence hypothesis to obtain tractable surrogate for entropy which involved only marginal entropies. As the next step, it is reasonable to utilize entropy of higher-order marginals. A belief propagation algorithm precisely does that and allows for its evaluation by means of message-passing. In what follows, we shall describe the belief propagation algorithm and its relation to the corresponding variational approximation for a restricted instance of loss network, the pair-wise loss network, for ease of exposition. The belief propagation algorithm for general loss network will be described near the end. We note that the belief propagation algorithm for the loss network model was first used by Ni and Tatikonda (2007).

We shall start by introducing the restricted, pair-wise loss network. Under this restriction on the loss network model, we shall require that each link is shared by at most two routes. That is, $|\{i : A_{ji} > 0\}| \leq 2$ for each j. We shall assume that $A_{ji} \in \{0, 1\}$ for all $i,\ j$ without loss of generality. Further, we shall assume that a given pair of routes will share at most one edge. In this situation, we can effectively represent the bipartite factor graph (graphical model) $G = (U \cup V, E)$ by an undirected graph G' defined as follows: $G' = (U, E')$ with vertices $U = \{1, \ldots, N\}$ corresponding to N different routes and edges E' so that $(i, j) \in E'$ if and only if routes i and j share a link. Let C_{ij} represent capacity of the link shared by routes i and j: if $(i, j) \in E'$ then C_{ij} is finite, else set it to ∞. Call this graph $G' = (U, E')$. Define neighbors of node i as $\mathcal{N}(i) = \{j \in U : (i, j) \in E'\}$. We call this pair-wise loss network model as the constraints imposed on routes through link capacities can be represented by means of the undirected graph G'.

Now we describe the belief propagation (BP) algorithm. The belief propagation is an iterative procedure that exchanges messages along each edge of G' in both directions. The messages exchanged try to incorporate the philosophy behind the dynamic programing in a local manner. More precisely, as part of BP, each node (here route) $i \in U$ wishes to estimate its marginal distribution, $\hat{\mu}_{(i,k)},\ 0 \leq k \leq K_i$. Initially, each node has no prior information about how to estimate it. It refines its estimate over time by means of exchanging messages with its neighbors (as per G'). The information a node $i \in U$ sends to its neighbor $j \in U$ (i.e. $(i, j) \in E'$) represents 'belief' of node i about j's marginal distribution. This 'belief' is generated based on node i's own estimate of its marginal distribution and the estimation of conditional probability of node j given node i's marginal through the constraint imposed by link joining them. In a sense, generation of a message is a naive execution of Bayes's rule (or total probability theorem). Finally, the estimation of each node's marginal is updated based on messages or beliefs received from its neighbors using another application of Bayes's rule assuming that the messages generated by all neighbors are independent of each other. In a sense, philosophically BP is quite similar to the Erlang approximation – both try to iteratively reach fixed point equations that are obtained assuming some form of independence between quantities of interest. The precise description of BP is as follows.

Belief Propagation: Pair-wise Loss Network.

---

1. Denote by t the iteration of the algorithm, with $t = 0$ initially. For each $(i, j) \in E'$, let $m^{(t)}_{i\to j}(k)$ (resp. $m^{(t)}_{j\to i}(k)$) denote message from node i to j (resp. j to i) in iteration t about marginal probability of node j (resp. i) taking value k. Initially, for all $(i, j) \in E'$

$$m^{(0)}_{i\to j}(k) = \mathbf{1}_{\{k\le K_j\}}, \quad \text{and} \quad m^{(0)}_{j\to i}(k) = \mathbf{1}_{\{k\le K_i\}}.$$

2. In iteration $t+1$, update messages as follows: with notation $w_{ik} = \log \frac{\nu_i^k}{k!}$

$$\begin{aligned} m^{(t+1)}_{i\to j}(k) &\propto \mathbf{1}_{\{k\le K_j\}} \Big(\sum_{0\le k'\le C_{ij}-k} \exp(w_{ik'}) \prod_{\ell\in\mathcal{N}(i)\setminus\{j\}} m^{(t)}_{\ell\to i}(k') \Big), \\ m^{(t+1)}_{j\to i}(k) &\propto \mathbf{1}_{\{k\le K_i\}} \Big(\sum_{0\le k'\le C_{ij}-k} \exp(w_{jk'}) \prod_{\ell\in\mathcal{N}(j)\setminus\{i\}} m^{(t)}_{\ell\to j}(k') \Big). \end{aligned} \tag{45}$$

Each message vector (from $i \to j$, $j \to i$) is normalized to sum to 1.

3. Each node (route) i estimates marginal distribution at the end of iteration $t + 1$ as

$$\hat{\mu}^{(t+1)}_{(i,k)} \propto \mathbf{1}_{\{k\le K_i\}} \exp(w_{ik}) \prod_{j\in\mathcal{N}(i)} m^{(t+1)}_{j\to i}(k), \tag{46}$$

and the pair-wise marginal for $(i, j) \in E'$ is estimated as

$$\begin{aligned} \hat{\mu}^{(t+1)}_{(ij,kk')} \propto \mathbf{1}_{\{k+k'\le C_{ij}\}} \ \exp\big(w_{ik} + w_{jk'}\big) \Big(\prod_{\ell\in\mathcal{N}(i)\setminus\{j\}} m^{(t+1)}_{\ell\to i}(k) \Big) \\ \times \Big(\prod_{\ell'\in\mathcal{N}(j)\setminus\{i\}} m^{(t+1)}_{\ell'\to j}(k') \Big). \end{aligned} \tag{47}$$

The belief propagation algorithm as described above is an iterative procedure with its messages being finite dimensional non-negative valued vectors normalized to sum to 1. It's iteration defined by (45) is a continuous map from a bounded convex set to a bounded convex set. Therefore, by an application of the Brouwer's fixed point theorem, it does have a fixed point. The basic question is, whether the fixed points are meaningful. As mentioned earlier, the following result of Yedidia et al. (2001) shows that the belief propagation fixed points are intimately related to the variational characterization (of Lemma 1) through an approximation. Specifically, consider the following variational approximation known as the *Bethe Variational Problem* (BVP) :

$$\begin{aligned}
\text{maximize} \quad & \sum_i \sum_{0\le k\le K_i} w_{ik}\hat{\mu}_{(i,k)} + \sum_i H_{\mathrm{ER}}(\hat{\boldsymbol{\mu}}_i) - \sum_{(i,j)\in E'} I(\hat{\boldsymbol{\mu}}_{ij}) && (48)\\
\text{over} \quad & \hat{\mu}_{(i,k)} \ge 0, \quad \hat{\mu}_{(ij;kk')} \ge 0, \quad \text{for all} \;\; i,\; (i,j)\in E' \\
\text{subject to} \quad & \sum_k \hat{\mu}_{(i,k)} = 1, \quad \text{for all} \;\; i, && (49)\\
& \sum_{k'} \hat{\mu}_{(ij;kk')} = \hat{\mu}_{(i,k)}, \quad \text{for all } (i,j)\in E,\; k\le K_i. && (50)
\end{aligned}$$

In above $\hat{\mu}_{(i,k)}$ represent node (route) i's marginal distribution – probability of node i taking value k; $\hat{\mu}_{(ij,k\ell)}$ represent pair-wise marginal distributions – probability of nodes i and j taking values k and ℓ respectively. Equation (49) is a normalization constraint while (50) is the consistency constraint. Notation $I(\hat{\boldsymbol{\mu}}_{ij})$ represents mutual information between nodes i and j based on pair-wise marginal distribution $\hat{\boldsymbol{\mu}}_{ij}$ defined as

$$\begin{aligned}
I(\hat{\boldsymbol{\mu}}_{ij}) &= H_{\mathrm{ER}}(\hat{\boldsymbol{\mu}}_i) + H_{\mathrm{ER}}(\hat{\boldsymbol{\mu}}_j) - H_{\mathrm{ER}}(\hat{\boldsymbol{\mu}}_{ij}) \\
&= -\sum_k \hat{\mu}_{(i,k)} \log \hat{\mu}_{(i,k)} - \sum_k \hat{\mu}_{(j,k)} \log \hat{\mu}_{(j,k)} + \sum_{k,k'} \hat{\mu}_{(ij,kk')} \log \hat{\mu}_{(ij,kk')}.
\end{aligned} \tag{51}$$

Observe that the BVP replaces the entropy term in variational characterization by the so called *Bethe entropy*, $H_{\mathrm{Bethe}}(\boldsymbol{\mu}) = \sum_i H_{\mathrm{ER}}(\boldsymbol{\mu}_i) - \sum_{(i,j)\in E'} I(\boldsymbol{\mu}_{ij})$ with $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_{ij}$ representing marginals of node i and node pair (i,j) with respect to $\boldsymbol{\mu}$.

The fixed points of belief propagation algorithm are related to the zero gradient points (or stationary points) of the Lagrangian of the Bethe Variational Problem. To define Lagrangian, introduce variables y_i for constraint (49) for each i; variables $z_{j\to i}(k)$ for constraint (50) for each $(i,j) \in E'$ and $k \le K_i$ (similarly $z_{i\to j}(k)$ for $(i,j) \in E'$ and $k \le K_j$). Then the Lagrangian of BVP is

$$\begin{aligned}\mathcal{L}(\hat{\boldsymbol{\mu}}, \mathbf{y}, \mathbf{z}) = \; & \sum_i \sum_{0\le k\le K_i} w_{ik}\hat{\mu}_{(i,k)} + H_{\text{Bethe}}(\hat{\boldsymbol{\mu}}) + \sum_i y_i \Big(1 - \sum_k \hat{\mu}_{(i,k)}\Big) \\ & + \sum_{(i,j)\in E'} \Big[\sum_{k\le K_i} z_{j\to i}(k) \Big(\hat{\mu}_{(i,k)} - \sum_{k'\le K_j} \hat{\mu}_{(ij,kk')}\Big) \\ & \qquad + \sum_{k\le K_j} z_{i\to j}(k) \Big(\hat{\mu}_{(j,k)} - \sum_{k'\le K_i} \hat{\mu}_{(ij,k'k)}\Big)\Big]. \qquad (52)\end{aligned}$$

Theorem 4 (Fixed point characterization) *Let $(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)$ be a zero gradient point of $\mathcal{L}$, i.e.*

$$\begin{aligned}\frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)}{\partial \hat{\mu}_{(i,k)}} &= 0, \quad \textit{for all} \quad i, \; k \le K_i, \\ \frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)}{\partial \hat{\mu}_{(ij,kk')}} &= 0, \quad \textit{for all} \quad (i,j) \in E', \; k \le K_i, k' \le K_j, \\ \frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)}{\partial z_{i\to j}(k)} &= 0, \quad \textit{for all} \quad (i,j) \in E', \; k \le K_j, \\ \frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)}{\partial z_{j\to i}(k)} &= 0, \quad \textit{for all} \quad (i,j) \in E', \; k \le K_i, \\ \frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)}{\partial y_i} &= 0, \quad \textit{for all} \quad i. \qquad (53)\end{aligned}$$

Define vector of messages, $\mathbf{m}^*$ as follows: for each $(i,j) \in E'$,

$$\log m^*_{i\to j}(k) \propto z_{i\to j}(k), \quad k \le K_j, \quad \textit{and} \quad \log m^*_{j\to i}(k) \propto z_{j\to i}(k), \quad k \le K_i.$$

Then $\mathbf{m}^*$ is a fixed point of message updates under belief propagation and $\hat{\boldsymbol{\mu}}^*$ is the node and pair-wise marginals based on $\mathbf{m}^*$ obtained as per (46)-(47).

Proof. Since the partial derivatives of $\mathcal{L}$ with respect to y_i, $z_{i\to j}(k)$ and $z_{j\to i}(k)$ are 0 at $(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)$, it must be that $\hat{\boldsymbol{\mu}}^*$ satisfies the constraints (49) and (50). Let us consider partial derivatives of $\mathcal{L}$ with respect to $\hat{\mu}_{(i,k)}$ and $\hat{\mu}_{(ij;kk')}$. It can be checked that

$$\begin{aligned}\frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}, \mathbf{z}, \mathbf{y})}{\partial \hat{\mu}_{(i,k)}} &= w_{ik} + y_i - 1 - \log \hat{\mu}_{(i,k)} + \sum_{j \in \mathcal{N}(i)} z_{j \to i}(k) \\ \frac{\partial \mathcal{L}(\hat{\boldsymbol{\mu}}, \mathbf{z}, \mathbf{y})}{\partial \hat{\mu}_{(ij,kk')}} &= -z_{j \to i}(k) - z_{i \to j}(k') + 1 + \log \tilde{\mu}_{(i,k)} + \log \tilde{\mu}_{(j,k')} - \log \hat{\mu}_{(ij,kk')},\end{aligned} \tag{54}$$

where we have used the notation $\tilde{\mu}_{(i,k)} = \sum_{k'} \hat{\mu}_{(ij;kk')}$ and $\tilde{\mu}_{(j,k')} = \sum_k \hat{\mu}_{(ij,kk')}$. Now using hypothesis of theorem that at $(\hat{\boldsymbol{\mu}}^*, \mathbf{z}^*, \mathbf{y}^*)$ these equal to 0, the fact we observed earlier that $\hat{\boldsymbol{\mu}}^*$ satisfies (49)-(50) and minor manipulation leads to the following:

$$\begin{aligned}\log \hat{\mu}^*_{(i,k)} &\propto w_{ik} + \sum_{\ell \in \mathcal{N}(i)} z^*_{\ell \to i}(k) \\ \log \hat{\mu}^*_{(j,k')} &\propto w_{jk'} + \sum_{\ell' \in \mathcal{N}(j)} z^*_{\ell' \to j}(k') \\ \log \hat{\mu}^*_{(ij;kk')} &\propto w_{ik} + w_{jk'} + \sum_{\ell \in \mathcal{N}(i) \setminus \{j\}} z^*_{\ell \to i}(k) + \sum_{\ell' \in \mathcal{N}(j) \setminus \{i\}} z^*_{\ell' \to j}(k').\end{aligned} \tag{55}$$

Let the vector of messages, $\mathbf{m}^*$, be defined based on $\mathbf{z}^*$ as stated in the statement of the theorem, i.e.

$$\log m^*_{i \to j}(k) \propto z_{i \to j}(k), \quad k \le K_j, \quad \text{and} \quad \log m^*_{j \to i}(k) \propto z_{j \to i}(k), \quad k \le K_i.$$

Then, (55) along with the constraint (50) implies that $\mathbf{m}^*$ is indeed a fixed point of the message update iteration of belief propagation algorithm (45). This completes the proof of Theorem 4. □

Some remarks are in order. Theorem 4 states that stationary or zero-gradient points of Bethe Variational Problem (BVP) are fixed points of the message-passing belief propagation algorithm. However, it is does not claim all fixed points are of this form. Such a result can be established for distributions of the Exponential family (the loss network distributions are not such due to hard inequality constraints). An interested reader is referred to monograph by Wainwright and Jordan (2008). It should also be noted that unlike dual coordinate descent algorithm for mean-field approximation, belief propagation is not known to converge to a fixed point.

3.6. Belief propagation: General loss network

Now we provide description of belief propagation for the general loss network setup. For this, we shall utilize the bipartite factor graph notation, $G = (U \cup V, E)$. We shall utilize notation $\mathcal{N}_u(i) = \{j \in V : (i,j) \in E\}$ for $i \in U$ and $\mathcal{N}_v(j) = \{i \in U : (i,j) \in E\}$ for $j \in V$. In the description of the algorithm that follows, we will stick to use of i (i' etc) for routes and j (j' etc) for links.

Belief Propagation: General Loss Network.

---

1. Denote by t the iteration of the algorithm, with $t = 0$ initially. For each $(i,j) \in E$, $m^{(t)}_{i\to j}(k)$ (resp. $m^{(t)}_{j\to i}(k)$) denote message from node i to j (resp. j to i). Message $m^{(t)}_{i\to j}(k)$ represents belief of node i about marginal probability of route j having k active calls. They are always normalized to 1. Initially, for all $(i,j) \in E$
$$m^{(0)}_{i\to j}(k) \propto \mathbf{1}_{\{k\le K_i\}} \quad \text{and} \quad m^{(0)}_{j\to i}(k) \propto \mathbf{1}_{\{k\le K_i\}}.$$

2. In iteration $t+1$, update messages as follows: with notation $w_{ik} = \log \frac{\nu_i^k}{k!}$, for $k \le K_i$
$$\begin{aligned} m^{(t+1)}_{i\to j}(k) &\propto \exp\big(w_{ik}\big) \Big(\prod_{j'\in\mathcal{N}_u(i)\backslash\{j\}} m^{(t)}_{j'\to i}(k)\Big), \\ m^{(t+1)}_{j\to i}(k) &\propto \Big(\sum_{(k_{i'})\in S(C_j,i,k)} \prod_{i'\in\mathcal{N}_v(j)\backslash\{i\}} m^{(t)}_{i'\to j}(k_{i'})\Big), \end{aligned} \tag{56}$$
where $S(C_j,i,k) = \{(k_{i'}) : \sum_{i'\in\mathcal{N}_v(j)\backslash\{i\}} A_{ji'}k_{i'} \le C_j - A_{ji}k\}$. All message vectors (from $i \to j$ and $j \to i$) are normalized to 1.

3. Each node (route) i estimates marginal distribution at the end of iteration $t+1$ as
$$\hat{\mu}^{(t+1)}_{(i,k)} \propto \mathbf{1}_{\{k\le K_i\}} \exp\big(w_{ik}\big) \prod_{j\in\mathcal{N}_u(i)} m^{(t+1)}_{j\to i}(k). \tag{57}$$

3.7. Discussion, future directions

In this section, we discussed various message-passing algorithms for performance evaluation for capacity planning problem modeled as a stochastic loss networks. Primarily, we discussed algorithms based on optimization and variational approximation. The optimization based algorithms (include Erlang approximation) tend to capture the dominant effect on loss rates induced by the mode of the stationary distribution. On the other hand, the variational approximation based algorithms, the mean-field and belief propagation, attempt to obtain a correction over them by utilizing surrogates for the 'entropy' term that is entirely ignored in the optimization based method (or Erlang approximation). While these algorithms are progress in the right direction, there are various issues that remain open. In what follows, we summarize a set of concrete questions.

To start with, it would be good to establish that the loss rate evaluation under the mean-field algorithm is asymptotically correct under the large network limit. We believe that this should be relatively easier compared to establishing a similar result for the belief propagation algorithm. While the dual co-ordinate descent based implementation of mean-field provides a convergent method, establishing convergence property of belief propagation (or designing a convergent modification of belief propagation) would be of interest. It is worth remarking that belief propagation always converges when the underlying graphical model of the corresponding Markov Random Field is a tree, see for example the book by Pearl (1988). In general, certain sufficient conditions for convergence are implied by the conditions for uniqueness of Gibbs distribution on infinite structure by Dobrushin and Simon (see book by Georgii (1988) for these conditions and work by Tatikonda and Jordan (2002) on how they imply convergence of belief propagation). In general, the convergence and correctness property of belief propagation remain open and it is very much likely that stochastic loss network model can provide fertile ground for developing such an understanding.

More ambitiously, as an ideal solution it would make sense to develop a sequence of approximations that successively try to provide better evaluation of the loss rates for generic loss network model by gracefully increasing the complexity of the solution. Indeed, an initial attempt has been made towards this by Jung et al. (2008) where the authors propose a modification of optimization based method by utilizing it as a subroutine to refine the estimation of loss rates. They show its effectiveness especially when the network is 'critically' loaded.

Towards achieving the above mentioned ideal solution, it is reasonable to speculate the possibility of utilizing some of the recent BP-like algorithms for the MARG problem. For example, works by Weitz (2006), Gamarnik and Katz (2007) and Bayati et al. (2007a) provide a dynamic programing and cavity method of Parisi (1988), Mezard et al. (2002) and Mezard et al. (1987) based approach for efficient approximation of MARG problem in order to obtain a method for counting of combinatorial objects for a restricted class of graphs and problems. The work by Jung et al. (2009) provides a simple, local BP-like linear time algorithm that utilizes the *geometry* of graphical model to obtain MAP as well as compute the logarithm of partition function ($\log Z(\mathbf{C})$). The sophisticated variational approximations like the Tree Reweighted Algorithm by Wainwright et al. (2005a,b) provides efficient convex relaxation based solutions for MAP and MARG. Finally, an entirely different line of approach based on the classical Markov Chain Monte Carlo (MCMC) method seem to be little explored in the context of loss network. Given that under the large network scaling, the distribution is well approximated through a unimodular distribution with concave density, we strongly believe that MCMC based approaches for large loss networks could be quite effective. This comment is particularly motivated by works of Dyer et al. (1991), Lovász and Vempala (2003), Bertsimas and Vempala (2004) and Kalai and Vempala (2006) where authors have managed to design provably efficient MCMC based methods for both MARG and MAP problem when underlying distribution has concave density with convex support.

4. Scheduling

This section discusses message-passing algorithms for scheduling or contention resolution in a queueing network modeled as a stochastic processing network with constraints of the type (1). We start with the description of the queueing model. We shall be interested in two classes of efficient, myopic policies: the maximum weight policy by Tassiulas and Ephremides (1992) and the fair bandwidth sharing policy proposed by Kelly et al. (1998); Mo and Walrand (1998). We shall describe message-passing implementations for both of these policies. Specifically, the primal-dual algorithm provides message-passing implementation for the bandwidth sharing model; the maximum weight policy is implemented using belief propagation. The description of these algorithms is restricted to examples of bandwidth sharing or congestion control in the Internet (a la TCP) and scheduling in an input-queued switch in an Internet router. The primal-dual algorithm for fair bandwidth sharing policy extends easily for generic stochastic processing network considered here. However, the belief propagation for the maximum weight policy in general is only heuristic. The associated challenges and directions for future work are summarized near the end of the section.

4.1. Model

We define a general queueing network model for resource allocation. Here our interest is in two contention resolution or scheduling policies. We shall consider two example scenarios: scheduling in an input-queued switch and bandwidth sharing in the Internet. The dynamics of queueing network model is described by means of the fluid model. The precise stochastic models that are well approximated by the fluid model can be found in the literature; see for example Dai and Prabhakar (2000); Kelly and Williams (2004); Shah and Wischik (2011); Dai and Lin (2005, 2008); Gromoll and Williams (2009).

As stated earlier, we have a collection of N queues with infinite buffers. Queues can be served as per an action that is feasible with respect to constraints (1). That is, at any time a queue can receive service as per $\mathbf{x} \in \mathbb{R}_+^N$ where $\mathbf{x} \in \mathcal{S}$ and $\mathcal{S}$ is the space of all feasible solutions to (1), i.e.

$$\mathcal{S} = \{\mathbf{x} \in \Sigma^N : A\mathbf{x} \leq \mathbf{C}\},$$

where $\mathbf{C} \in \mathbb{R}_+^M$ and Σ is either a discrete or continuous subset of $\mathbb{R}_+$. We shall assume that every queue is serviceable, i.e. for every i there exists some $\mathbf{x} \in \mathcal{S}$ such that $x_i > 0$, and the empty schedule is feasible, i.e. $\mathbf{0} \in \mathcal{S}$. Let $\langle \mathcal{S} \rangle$ represent the convex hull of the set of all feasible schedules $\mathcal{S}$ defined as

$$\langle \mathcal{S} \rangle = \Big\{ \mathbf{y} \in \mathbb{R}_+^N : \ \mathbf{y} \leq \sum_{k \geq 1} \beta_k \mathbf{x}_k, \ \sum_{k \geq 1} \beta_k \leq 1, \text{ and } \beta_k \geq 0, \ \mathbf{x}_k \in \mathcal{S} \text{ for all } k \Big\}.$$

As we shall see, the effective service rates provided to queues over time belong to $\langle \mathcal{S} \rangle$. In $\mathcal{S}$, if Σ is a finite discrete set then by definition $\mathcal{S}$ has finitely many elements and hence $\langle \mathcal{S} \rangle$ defines the convex set obtained by all sorts of convex combinations of elements of $\mathcal{S}$. Else, Σ is a continuous interval. Since $\mathbf{0} \in \mathcal{S}$ is a requirement, we must have $\Sigma = [0, b]$ for some $b > 0$ or $\Sigma = \mathbb{R}_+$. In this case, $\mathcal{S}$ is a polytope with finitely many extreme points. Therefore, $\langle\mathcal{S}\rangle$ can be thought of as obtained by convex combinations of the extreme points of $\mathcal{S}$. We shall assume a single-hop network, i.e. once work is served from a queue, it leaves the network. Each queue i receives work as per exogeneous arrival process at rate $\lambda_i \in \mathbb{R}_+$. Let $\boldsymbol{\lambda} \in \mathbb{R}_+^N$ denote the arrival rate vector. For any $D \in \mathbb{N}$, let $\mathcal{AC}^D$ denote the space of absolutely continuous trajectories in $\mathbb{R}^D$. We describe the queueing dynamics of interest through a fluid model.

Definition 1 (Fluid queueing dynamics) *Let $\boldsymbol{\lambda} \in \mathbb{R}_+^N$. Say that the triple $\mathbf{q}(\cdot) \in \mathcal{AC}^N$, $\mathbf{z}(\cdot) \in \mathcal{AC}^N$, $\mathbf{s}(\cdot) \in \mathcal{AC}^N$ is a fluid model solution for the queueing dynamics with arrival rate vector $\boldsymbol{\lambda}$ if $s(0) = \mathbf{0}$, $\mathbf{z}(0) = \mathbf{0}$, and the following equations are satisfied for all t:*

$$\mathbf{q}(t) = \mathbf{q}(0) + \boldsymbol{\lambda} t - \mathbf{s}(t) + \mathbf{z}(t) \tag{58}$$

$$\frac{1}{t}\mathbf{s}(t) \in \langle\mathcal{S}\rangle \tag{59}$$

$$\textit{each } s_i(\cdot) \textit{ and } z_i(\cdot) \textit{ are non-decreasing for all } i \tag{60}$$

$$\textit{for all } i \textit{ and all regular time } t,\ \dot{z}_i(t) = 0 \textit{ if } q_i(t) = 0 \tag{61}$$

$$\mathbf{z}(t) \leq \mathbf{s}(t). \tag{62}$$

In above, by regular time t we mean any t at which all components of triple $(\mathbf{q}, \mathbf{z}, \mathbf{s})$ are differentiable. Since they are all absolutely continuous, almost all t are regular.

Here $\mathbf{q}(t)$ represents the vector of queue sizes at time t, $\mathbf{z}(t)$ represents the cumulative idleness up to time t, and $\mathbf{s}(t)$ represents the total amount of service provided to queues up to time t.

Next, we define two contention resolution or scheduling policies in terms of fluid model. Given that $\mathbf{s}(\cdot)$ is absolutely continuous, the derivative of $\mathbf{s}(\cdot)$, that is the instantaneous rate vector allocated to queues, is well defined for almost all (regular) t. Therefore, we shall define these two policies in terms of properties of these instantaneous rate allocation.

Definition 2 (Max-weight policy) *Given $\alpha > 0$, we say that $(\mathbf{q}, \mathbf{z}, \mathbf{s})$ is a fluid model solution for the α max-weight policy, denoted as MW-α, if $(\mathbf{q}, \mathbf{z}, \mathbf{s})$ is a fluid model solution for the queueing dynamics and in addition for any regular t,*

$$\frac{d}{dt}\mathbf{s}(t) \cdot \mathbf{q}^\alpha(t) = \max_{\boldsymbol{\sigma} \in \langle\mathcal{S}\rangle} \boldsymbol{\sigma} \cdot \mathbf{q}^\alpha(t). \tag{63}$$

Definition 3 ($\boldsymbol{\alpha}$-fair policy) *Given $\alpha \in \mathbb{R}_+ \backslash \{1\}$, we say that $(\mathbf{q}, \mathbf{z}, \mathbf{s})$ is a fluid model solution for the α-fair policy if $(\mathbf{q}, \mathbf{z}, \mathbf{s})$ is a fluid model solution for the queueing dynamics and in addition for any regular t,*

$$\frac{d}{dt}\mathbf{s}(t) \in \operatorname*{argmax}_{\boldsymbol{\rho} \in \langle \mathcal{S} \rangle} \mathbf{q}^{\alpha}(t) \cdot \frac{\boldsymbol{\rho}^{1-\alpha}}{1-\alpha}. \tag{64}$$

Implementing the maximum weight MW-α policy or α-fair policy requires solving the following type of network-wide problem: given vector of queue-sizes $\mathbf{q}$, the schedule is a solution of

$$\begin{aligned} \text{maximize} \quad & F(\mathbf{q}, \mathbf{x}) \\ \text{over} \quad & \mathbf{x} \in \Sigma^N \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{C}. \end{aligned} \tag{65}$$

For the maximum weight MW-α policy

$$F(\mathbf{q}, \mathbf{x}) = \sum_i q_i^{\alpha} x_i;$$

and for the α-fair policy

$$F(\mathbf{q}, \mathbf{x}) = \sum_i q_i^{\alpha} \frac{x_i^{1-\alpha}}{1-\alpha}.$$

Therefore, the goal is to obtain a message-passing algorithm to solve optimization problem (65) with linear objective for MW-α and concave objective for α-fair policy. Next, we describe two examples of practical interest for which we shall describe such message-passing algorithms.

4.1.1. Model application: input-queued switch

The role of a switch in an Internet router is to transfer (or switch) packets arriving at the input (or ingress) ports of the router to their corresponding output (or egress) ports through a switch fabric. The input-queued switch is a specific switch architecture in which all packets that are waiting to be transfered are buffered (or queued) at the input ports with separate queues at each input port for different output ports. Usually, each port in a router acts as both input and output port. Therefore, logically a switch has, say n

input ports and n output ports. Thus an input-queued switch has total of $N = n^2$ queues: n input ports, each with n queues. In such an architecture, the switch fabric imposes constraints that at any given time instance, each input port can transfer data at unit rate to at most one other output port and each output port can receive data at unit rate from at most one other input port. Thus, the contention resolution or scheduling constraint corresponds to finding a 'matching' of n input and n output ports at each time instance so that each input (respectively output) port is matched to a distinct output (respectively input) port. Equivalently, at each time instance the scheduling algorithm needs to pick a permutation of n inputs with i permuted to j if and only if input i is matched to output j.

Such an input-queued switch can be effectively modeled as a collection of $N = n^2$ queues operating in slotted time; in this application it is most natural to consider the queue lengths to be a matrix in $\mathbb{R}_+^{n\times n}$ rather than a vector in $\mathbb{R}_+^N$; the (i, j)th component corresponds to the length of queue of packets waiting at input i for output j. At the beginning of each timeslot, a (random) integer number of packets arrive, and there may be arrivals to any queue. Then a service action $\mathbf{x}$ is chosen from the set $\mathcal{S} \subset \{0, 1\}^{n\times n}$ consisting of all n! permutation matrices, chosen according a scheduling policy. During the timeslot, $\mathbf{x}$ is the offered service to each queue, and served work leaves the system at the end of the timeslot. Interest is in utilizing the maximum weight MW-α policy for selecting the service action or permutation. Then, the MW-α policy will require solving the following optimization problem at each time instance t: given queue-sizes $\mathbf{q}(t) \in \mathbb{R}_+^{n\times n}$,

$$\begin{aligned} \text{maximize} \quad & \sum_{1\le i,j\le n} x_{ij} q_{ij}(t) \\ \text{over} \quad & x_{ij} \in \{0, 1\}, \quad \text{for all} \;\; 1 \le i, j \le n \\ \text{subject to} \quad & \sum_k x_{ik} \le 1, \quad \sum_k x_{kj} \le 1, \quad \text{for all} \;\; 1 \le i, j \le n. \end{aligned} \tag{66}$$

We note that the maximum weight policy for input-queued switch was first studied by McKeown et al. (1996) and its fluid model was introduced by Dai and Prabhakar (2000).

4.1.2. Model application: bandwidth sharing in the Internet

Roberts and Massoulie (2000) introduced a model for bandwidth-sharing in the Internet. They took there to be a finite set M of links, and for each link

j an associated capacity $C_j \geq 0$, and a finite set $\mathcal{R} = \{1, \ldots, N\}$ of routes where each route i being a subset of M links. At every instant in time t, there is a certain number $q_i(t)$ of active flows on route i. These flows receive service at a certain rate, which depends only on the number of active flows: let $x_i(\mathbf{q}(t))/q_i(t)$ be the service rate for each flow on route i. We can think of the Internet's congestion control algorithm (TCP) as selecting a service rate vector $\mathbf{x}(\mathbf{q}(t))$ that satisfies the capacity constraint $A\mathbf{x}(\mathbf{q}(t)) \leq \mathbf{C}$ where $A_{ji} = 1$ if route i utilizes link j and 0 otherwise. Let $\mathcal{S}$ be the set of all feasible rate allocations $\{\mathbf{x} \in \mathbb{R}_+^N : A\mathbf{x} \leq \mathbf{C}\}$. Then the constraint can be written as $\mathbf{x}(\mathbf{q}(t)) \in \langle \mathcal{S} \rangle$. The α-fair bandwidth-sharing policy suggests that $\mathbf{x}(\mathbf{q}(t))$ is obtained by solving the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_i q_i^\alpha(t) \frac{x_i^{1-\alpha}}{1-\alpha} \\ \text{over} \quad & \mathbf{x} \in \mathbb{R}_+^N \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{C}. \end{aligned} \tag{67}$$

Kelly et al. (1998) suggested that in a stationary flow-level model, the bandwidth sharing or congestion control (TCP) protocol can be viewed as attempting to reach network-wide rate allocation as per α-fair policy for certain value of α. Based on this, the notion of α-fairness for general α was introduced by Mo and Walrand (1998). Bonald and Massoulie (2001) and Kelly and Williams (2004) introduced and analyzed fluid model equations for this system for the case that each route i has flows arriving as per Poisson process of rate λ_i with flow size having exponential distribution of mean 1.

4.1.3. Network stability

An important reason for attractiveness of MW-α and α-fair bandwidth sharing policy is their throughput optimality property despite being myopic, i.e. utilize only current network state in terms of $\mathbf{q}(t)$ to make scheduling decisions. Roughly speaking, by throughput optimality we mean that as long as the network is not overloaded then queues remain finite. Precise definition is as follows.

Definition 4 (Throughput optimal) *A policy is called throughput optimal if for every fluid model solution,* $\mathbf{q}(t) = \mathbf{0}$ *for all* $t > 0$ *if* $\mathbf{q}(0) = \mathbf{0}$ *as long as* $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$*, the interior of* $\langle \mathcal{S} \rangle$*, defined as*

$$\langle \mathcal{S} \rangle^o = \Big\{ \mathbf{y} \in \mathbb{R}_+^N : \ \mathbf{y} \leq \sum_{k \geq 1} \beta_k \mathbf{x}_k, \ \sum_{k \geq 1} \beta_k < 1, \ \textit{and } \beta_k \geq 0, \ \mathbf{x}_k \in \mathcal{S} \textit{ for all } k \Big\}.$$

In what follows, throughput optimality of MW-α and α-fair policies are established using Lyapunov functions. The appropriate Lyapunov functions were introduced by Tassiulas and Ephremides (1992) and Bonald and Massoulie (2001) as well as de Veciana et al. (2001) for MW-α and α-fair policies respectively.

Lemma 4 *Given $\alpha > 0$, define Lyapunov function $L(\mathbf{q}) = \mathbf{1} \cdot \mathbf{q}^{1+\alpha} = \sum_i q_i^{1+\alpha}$. Let $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$. Then for any fluid model solution $\mathbf{q}(\cdot)$ under MW-α policy,*

$$\frac{d}{dt} L(\mathbf{q}(t)) < 0, \quad \textit{if} \ \ \mathbf{q}(t) \neq \mathbf{0},$$

for almost all t. Therefore, $\mathbf{q}(t) = \mathbf{0}$ if $\mathbf{q}(0) = \mathbf{0}$.

Proof. Consider a fluid model solution operating under MW-α policy with $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$. Let t be such that $\mathbf{x}(t) = \frac{d}{dt}\mathbf{s}(t)$ is well defined. Then, $\mathbf{x}(t)$ maximizes $\mathbf{q}^\alpha(t) \cdot \mathbf{x}$ over $\mathbf{x} \in \langle \mathcal{S} \rangle$. Since $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$, there exists $\varepsilon > 0$ so that

$$\boldsymbol{\lambda} \leq \sum_{\boldsymbol{\rho} \in \mathcal{S}} \beta_{\boldsymbol{\rho}} \boldsymbol{\rho}, \quad \text{such that} \quad \sum_{\boldsymbol{\rho} \in \mathcal{S}} \beta_{\boldsymbol{\rho}} = 1 - \varepsilon, \ \beta_{\boldsymbol{\rho}} \geq 0. \tag{68}$$

From the fluid model solutions, it follows that

$$\begin{aligned} \frac{d}{dt} L(\mathbf{q}(t)) &= (1+\alpha) \sum_i q_i^\alpha(t) \frac{d}{dt} q_i(t) \\ &= (1+\alpha) \sum_i q_i^\alpha(t) (\lambda_i - x_i(t) + z_i(t)) \\ &= (1+\alpha) \mathbf{q}^\alpha(t) \cdot (\boldsymbol{\lambda} - \mathbf{x}(t) + \mathbf{z}(t)) \\ &= (1+\alpha) \mathbf{q}^\alpha(t) \cdot (\boldsymbol{\lambda} - \mathbf{x}(t)) \qquad \text{because of (61)} \\ &\leq (1+\alpha) \mathbf{q}^\alpha(t) \Big(\sum_{\boldsymbol{\rho} \in \mathcal{S}} \beta_{\boldsymbol{\rho}} \boldsymbol{\rho} - \mathbf{x}(t) \Big) \\ &= (1+\alpha) \Big(\sum_{\boldsymbol{\rho} \in \mathcal{S}} \beta_{\boldsymbol{\rho}} \mathbf{q}^\alpha(t) \cdot \boldsymbol{\rho} - \mathbf{q}^\alpha(t) \cdot \mathbf{x}(t) \Big) \\ &\leq (1+\alpha) \Big(\sum_{\boldsymbol{\rho} \in \mathcal{S}} \beta_{\boldsymbol{\rho}} - 1 \Big) \Big(\mathbf{q}^\alpha(t) \cdot \mathbf{x}(t) \Big) \\ &< 0, \quad \text{if} \ \ \mathbf{q}(t) \neq \mathbf{0}. \end{aligned} \tag{69}$$

In the above, we have used the fact that for any $\boldsymbol{\rho} \in \langle \mathcal{S} \rangle$, $\mathbf{q}^{\alpha}(t)\cdot\boldsymbol{\rho} \leq \mathbf{q}^{\alpha}(t)\cdot\mathbf{x}(t)$. Now $L(\mathbf{q}(t)) = 0$ if and only if $\mathbf{q}(t) = \mathbf{0}$. From (69), it follows that

$$\begin{aligned} L^2(\mathbf{q}(t)) - L^2(\mathbf{q}(0)) &= 2\int_0^t L(\mathbf{q}(s))\frac{d}{ds}L(\mathbf{q}(s))ds \\ &\leq 0. \end{aligned} \tag{70}$$

Therefore, if $\mathbf{q}(0) = \mathbf{0}$ then $L(\mathbf{q}(t)) = 0$ for all $t > 0$. Therefore, $\mathbf{q}(t) = \mathbf{0}$. □

Lemma 5 *Given $\alpha \in \mathbb{R}_+ \backslash \{1\}$, define Lyapunov function $G(\mathbf{q}) = \boldsymbol{\lambda}^{-\alpha} \cdot \mathbf{q}^{1+\alpha} = \sum_i \lambda_i^{-\alpha} q_i^{1+\alpha}$. Let $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$. Then for any fluid model solution $\mathbf{q}(\cdot)$ under α-fair bandwidth sharing policy,*

$$\frac{d}{dt}G(\mathbf{q}(t)) < 0, \quad \textit{if} \;\; \mathbf{q}(t) \neq \mathbf{0},$$

for almost all t. Therefore, $\mathbf{q}(t) = \mathbf{0}$ if $\mathbf{q}(0) = \mathbf{0}$.

Proof. Define function $F(\mathbf{q}, \mathbf{x}) = \sum_i q_i^{\alpha} \frac{x_i^{1-\alpha}}{1-\alpha}$. Consider a fluid model solution operating under α-fair policy with $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$. And let t be such that $\mathbf{x}(t) = \frac{d}{dt}\mathbf{s}(t)$ is well defined. Then, $\mathbf{x}(t)$ maximizes $F(\mathbf{q}(t), \mathbf{x})$ over $\mathbf{x} \in \langle \mathcal{S} \rangle$. By definition $F(\mathbf{q}(t), \cdot)$ is a concave function and $\mathbf{x}(t)$ is its maximizer over a convex feasible set. Therefore, it follows that for any feasible $\boldsymbol{\rho} \in \langle \mathcal{S} \rangle$,

$$\nabla_{\mathbf{x}} F(\mathbf{q}(t), \mathbf{x}(t))\cdot(\boldsymbol{\rho} - \mathbf{x}(t)) \leq 0. \tag{71}$$

By concavity of function $F(\mathbf{q}(t), \cdot)$,

$$\nabla_{\mathbf{x}} F(\mathbf{q}(t), \boldsymbol{\rho})\cdot(\boldsymbol{\rho} - \mathbf{x}(t)) \leq \nabla_{\mathbf{x}} F(\mathbf{q}(t), \mathbf{x}(t))\cdot(\boldsymbol{\rho} - \mathbf{x}(t)). \tag{72}$$

Now there exists $\varepsilon > 0$ so that $(1+\varepsilon)\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle$ since $\boldsymbol{\lambda} \in \langle \mathcal{S} \rangle^o$. Therefore, $\boldsymbol{\rho} = (1+\varepsilon)\boldsymbol{\lambda}$ is feasible and from (71)-(72), we have

$$\begin{aligned} \nabla_{\mathbf{x}} F(\mathbf{q}(t), (1+\varepsilon)\boldsymbol{\lambda})\cdot((1+\varepsilon)\boldsymbol{\lambda} - \mathbf{x}(t)) &= \sum_i \frac{q_i^{\alpha}(t)}{(1+\varepsilon)^{\alpha}\lambda_i^{\alpha}}((1+\varepsilon)\lambda_i - x_i(t)) \\ &\leq 0. \end{aligned} \tag{73}$$

Therefore

$$\begin{aligned} \sum_i \frac{q_i^{\alpha}(t)}{\lambda_i^{\alpha}}(\lambda_i - x_i(t)) &\leq -\varepsilon \sum_i q_i^{\alpha}(t)\lambda_i^{1-\alpha} \\ &< 0, \;\; \text{if} \;\; \mathbf{q}(t) \neq \mathbf{0}. \end{aligned} \tag{74}$$

From fluid model solutions, it follows that

$$\begin{aligned}\frac{d}{dt}G(\mathbf{q}(t)) &= (1+\alpha)\sum_i \lambda_i^{-\alpha} q_i^{\alpha}(t)\frac{d}{dt}q_i(t) \\ &= (1+\alpha)\sum_i \lambda_i^{-\alpha} q_i^{\alpha}(t)(\lambda_i - x_i(t) + z_i(t)) \\ &= (1+\alpha)\sum_i \lambda_i^{-\alpha} q_i^{\alpha}(t)(\lambda_i - x_i(t)) \qquad \text{because of (61)} \\ &< 0, \end{aligned} \tag{75}$$

where the last inequality uses (74). From (75), using an argument similar to that used in Lemma 4 it follows that $\mathbf{q}(t) = \mathbf{0}$ if $\mathbf{q}(0) = \mathbf{0}$. This completes the proof of Lemma 5. □

4.2. Message-passing implementation

Implementation of α-fair bandwidth sharing and the MW-α policy requires solving network wide optimization problem (65) with appropriate objective function at each time instance. This is equivalent to solving a MAP problem over Markov Random Field of the type discussed in Section 2.1. To see this, consider the following. Given queue-size vector $\mathbf{q}$, let a collection of N random variables $\mathbf{X}$ have their joint distribution defined as (like (2))

$$\mathbb{P}\Big(\mathbf{X} = \mathbf{x}\Big) \propto \exp\Big(F(\mathbf{q}, \mathbf{x})\Big) \prod_{1\le j\le M} \mathbf{1}_{\{\sum_k A_{jk}x_k \le C_j\}}. \tag{76}$$

Then, solution of the MAP problem for this MRF solves the optimization problem (65). As discussed earlier, the MAP problem is hard in general. Therefore, implementing these scheduling policies in general require solving hard problem.

Now in the context of bandwidth sharing in the Internet or scheduling in an input-queued switch, it is essential to design message-passing implementation of such policies so as to have practical, scalable architectures. In general, message-passing solution is unlikely since the MAP problem is hard. As it happens, the examples of bandwidth sharing in the Internet and scheduling in input-queued switch are instances where it is indeed possible to design such message-passing solutions. The message-passing algorithm for bandwidth sharing in the Internet is based on primal-dual algorithm for concave maximization problem while that for input-queued switch scheduling is based on a belief propagation algorithm. In the case of input-queued switch, the auction algorithm by Bertsekas (1992) can be used as well. The auction algorithm is based on a modification (known as the ϵ-relaxation method) of dual co-ordinate descent algorithm. In contrast, belief propagation is a graphical model based dynamic programming approximation. Somewhat surprisingly, these two algorithms turn out to be essentially the same (at least syntactically). We shall explain this in detail later in this section.

In general, it is unlikely to have an exact message-passing implementation and best one can hope for is a reasonable heuristic. Since scheduling as per the policy of interest is merely a MAP in an MRF, we can utilize belief propagation, which is a heuristic for the MAP problem in MRF, as a heuristic for scheduling in general network. This belief propagation based scheduling heuristic is explained near the end of this section.

4.2.1. Bandwidth sharing using primal-dual

Given $\alpha \in \mathbb{R}_+ \backslash \{1\}$, the α-fair bandwidth sharing policy requires finding aggregate rate vector $\mathbf{x}(t)$ at time t such that it solves optimization problem (67) given the vector of number of backlogged flows $\mathbf{q}(t)$. In what follows, we shall drop notation of t for convenience. Now the optimization problem (67) has strictly concave objective over a bounded convex domain. Therefore, it has a unique solution, the $\mathbf{x}^* = \mathbf{x}^*(t)$. Now observe close similarity of the optimization problem (67) with the optimization problem (28) that appeared in the context of stochastic loss network under large network scaling. Both optimization problems have exactly the same feasible region and have strictly concave objective that is separable[1]. Subsequently, they have very similar Lagrangian dual. As before, let $\mathbf{y} \in \mathbb{R}_+^M$ denote the dual variables associated with constraint $A\mathbf{x} \le \mathbf{C}$. Then the Lagrangian dual of (67) can be presented as

$$\begin{aligned} \text{minimize} \quad & \frac{\alpha}{1-\alpha} \sum_i q_i \Big(\sum_j y_j A_{ji} \Big)^{1-\frac{1}{\alpha}} + \sum_j y_j C_j \qquad (77) \\ \text{over} \quad & \mathbf{y} \in \mathbb{R}_+^M. \end{aligned}$$

In Section 3, it was established that dual co-ordinate descent provides message-passing algorithm to solve the optimization problem (29) exactly. Given the

[1] A function $f : \mathbb{R}^N \to \mathbb{R}$ is called separable if it can be represented as $f(\mathbf{x}) = \sum_i f_i(x_i)$ with $f_i : \mathbb{R}^N \to \mathbb{R}$.

similarity of (77) and (29), similar dual co-ordinate descent algorithms would provide a message-passing implemention for finding optimal solution $\mathbf{y}^*$ of (77) as well. The Karush-Kuhn-Tucker conditions will imply that the pair of primal and dual optimal solutions $(\mathbf{x}^*, \mathbf{y}^*)$ should satisfy

$$x_i^* = q_i\Big(\sum_j A_{ji} y_j^*\Big)^{-1/\alpha} \quad \text{for} \quad 1 \le i \le N. \tag{78}$$

Therefore, given a dual optimal $\mathbf{y}^*$, the unique primal optimal $\mathbf{x}^*$ could be recovered. Like Theorem 2 and Lemma 3, such an algorithm will have excellent convergence properties (linear rate of convergence). However, implementing such a dual co-ordinate descent algorithm in the context of bandwidth sharing in the Internet does not seem feasible. Recall that each dual variable, say y_j, is associated with the link j of capacity C_j. The dual co-ordinate descent requires updates to be done in a partially synchronized (round-robin) manner which could be hard to achieve in a distributed system like the Internet. Further, the updates of co-ordinate descent could possibly change value of variables drastically. In a large distributed system like the Internet, drastic changes could lead to undesirable behavior like oscillations in the presence of noisy information. Finally, in an autonomous system like the Internet made of collection of decentralized entities, it unlikely to expect all links to maintain and update corresponding dual variables. In a sense, implementing bandwidth sharing in the context of Internet is a lot more constrained than the standard requirements faced in designing generic message-passing algorithm.

Somewhat miraculously, through an interpretation of queue-sizes as dual variables, an appropriate primal-dual algorithm can be designed where dual variables (interpreted as queues at links) are updated depending upon their loading and link capacity while the primal variables (rates allocated) are updated with the aim of maximizing individual objective value penalized appropriately through dual variables. Such a message-passing implementation provides an intuitively pleasing interpretation of the TCP protocol that is currently implemented to achieve bandwidth sharing. Such an interpretation was first explained by Kelly et al. (1998).

In the primal-dual algorithm, the allocated rates and value of dual variables are updated iteratively. With an abuse of notation, we shall use t to represent the algorithm iteration index. The backlogged number of flows, $\mathbf{q}$, should be thought of as constant. Equivalently, it is assumed that the time scale at which the primal-dual algorithm operates is much faster than the time-scale over which flows arrive and depart. The primal-dual updates are given by

$$\frac{d}{dt}x_i(t) = K_1(t)\Big(q_i^\alpha x_i^{-\alpha} - \sum_j y_j(t)A_{ji}\Big), \tag{79}$$

$$\frac{d}{dt}y_j(t) = K_2(t)\Big(\sum_k A_{jk}x_k(t) - C_j\Big)^+_{y_j(t)}. \tag{80}$$

In the above, $K_1(t), K_2(t)$ are possibly time varying strictly positive valued quantities; $[a]_b^+$ equals a if $b > 0$ and equals $\max(a, 0)$ if $b = 0$. The (79) is attempting to reach the relation between primal-dual optimal solutions implied by Karush-Kuhn-Tucker condition (78). The (80) is attempting to satisfy the complementary slackness condition related to the constraint $\sum_k A_{jk}x_k \leq C_j$.

An interpretation of the update equations (79) and (80) is as follows. The variable $y_j(\cdot)$ is like the buffer size at link j and equation (80) represents its rate level dynamics; as per (79) the rate variable $x_i(\cdot)$ tries to increase itself inversely proportional to $x_i(\cdot)$ and decrease its value proportional to the round-trip 'delay', $\sum_j y_j(t)A_{ji}$. An interested reader is referred to the monography by Srikant (2004) for a detailed account on relation between primal-dual algorithm for α-fair policy and the TCP proptocol.

Finally, we take note of the convergence and correctness property of the primal-dual algorithm. It's proof can be found in a standard text on optimization or in this context in the monograph by Srikant (2004).

Theorem 5 *Under the primal-dual algorithm described as per* (79)-(80),

$$\mathbf{x}(t) \to \mathbf{x}^*, \quad \textit{as} \quad t \to \infty,$$

where $\mathbf{x}^*$ *is the unique solution of optimization problem* (67).

4.2.2. Switch scheduling using belief propagation

We describe message-passing implementation of MW-α policy using belief propagation for input-queued switch scheduling. In the context of input-queued switch, the MW-α algorithm is required to solve the optimization problem (66) each time instance. The optimization problem (66) corresponds to finding a maximum weight matching in a complete weighted bipartite with n nodes in each partition for a switch with n input and output ports. That is, find a matching of inputs and outputs so that the sum of weights of the matched input-output pairs is maximized; here weight of an edge between input i and output j is q_{ij}^{α} (which is $q_{ij}^{\alpha}(t)$ at time t; for convenience we shall drop notation of t hence forth). Equivalently, each input i (respectively output j) wishes to find which of the n possible outputs (respectively inputs) it ought to connect to so that overall weight of all connections is maximized. In what follows, we shall use notation I_i to denote input node i and O_j to denote output node j for $1 \leq i, j \leq n$.

The belief propagation algorithm, in general is a heuristic, to find these connections iteratively by exchanging messages between all input-output nodes. The basic idea behind the belief propagation algorithm is to iteratively reach the fixed point equations that are induced by the Hamilton-Jacobi-Bellman (HJB) equation of dynamic programing with respect to the underlying problem (graph) structure (see book by Bertsekas (1995) for detailed account on dynamic programing). To understand this, let us consider the bipartite maximum weight matching problem. Node O_j wishes to decide which input to connect to as per optimal assignment. Let $m_{I_i \to O_j}$ be the weight of the optimal matching among all matchings in which O_j is connected to I_i. If node O_j has access to $m_{I_i \to O_j}$ for all i, then under the maximum weight matching, it should connect to i^* such that

$$i^* \in \operatorname*{argmax}_i m_{I_i \to O_j}.$$

The above conclusion remains true even if $m_{I_i \to O_j}$ does not represent the weight of optimal assignment conditioned on I_i connecting to O_j, but instead it represents the difference between the weights of the optimal assignments in which O_j connects to I_i and O_j does not connect to I_i. Now the basic question is how to find such $m_{I_i \to O_j}$. That is, the difference between the weights of optimal assignment with conditions that O_j connects to I_i and O_j does not connect to I_i.

Belief propagation is a heuristic to find these $m_{I_i \to O_j}$ values iteratively assuming that the underlying graph is tree. Imagine that $m_{I_i \to O_j}$ is the value that input I_i sends to output O_j based on similar values that it has received from other outputs $\{O_k : k \neq j\}$. Then dynamic programing for a tree graph would suggest that

$$m_{I_i \to O_j} = q_{ij}^{\alpha} - \max_{k \neq j} m_{O_k \to I_i}.$$

This is essentially because the difference between weights of optimal matching under which O_j connects to I_i and does not connect to I_i includes the gain in terms of the weight of edge (i, j), i.e. q_{ij}^{α}, and the loss in terms of I_i not able to connect to the best of the other outputs in $\{O_k : \ k \neq j\}$. This suggests a natural recursion to find $m_{I_i \to O_j}$ for all i, j. Based on this, we arrive at the following belief propagation algorithm.

Belief propagation: input-queued switch.

1. Denote by t the iteration of the algorithm. For each input I_i and output O_j, let $m_{I_i \to O_j}^{(t)}$ and $m_{O_j \to I_i}^{(t)}$ denote messages from nodes I_i to O_j and O_j to I_i respectively. Initially, $t = 0$ and $m_{I_i \to O_j}^{(0)} = m_{O_j \to I_i}^{(0)} = q_{ij}^{\alpha}$.
2. In iteration $t + 1$, update messages as follows: for $1 \leq i, j \leq n$,
$$\begin{aligned} m_{I_i \to O_j}^{(t+1)} &= q_{ij}^{\alpha} - \max_{k \neq j} \ m_{O_k \to I_i}^{(t)}, \\ m_{O_j \to I_i}^{(t+1)} &= q_{ij}^{\alpha} - \max_{k \neq i} \ m_{I_k \to O_j}^{(t)}. \end{aligned} \tag{81}$$
3. Each node estimates its assignment as follows: for $1 \leq i, j \leq n$,
$$\begin{aligned} I_i &\to \operatorname*{argmax}_{O_k : 1 \leq k \leq n} \ m_{O_k \to I_i}^{(t+1)}, \\ O_j &\to \operatorname*{argmax}_{I_k : 1 \leq k \leq n} \ m_{I_k \to O_j}^{(t+1)}. \end{aligned} \tag{82}$$

The belief propagation as described above is merely a heuristic. It is an iterative procedure. Apriori it is not clear if its estimates converge and if so, whether they are correct. The case of matching in bipartite graph is special and indeed belief propagation finds the correct solution. The precise statement is as follows.

Theorem 6 *Given $\alpha > 0$ and queue-sizes $\mathbf{q}$, let there be a unique optimal solution to optimization problem* (66)*. Then the estimation of belief propagation algorithm described above equals the optimal solution for all $t > \frac{2n|\mathbf{q}^{\alpha}|}{\delta}$ for all nodes. Here δ is the difference between weight of the optimal solution and the weight of second optimal solution of* (66)*.*

Theorem 6 was established by Bayati et al. (2008b). The use of belief propagation for switch scheduling was proposed by Bayati et al. (2007b).

4.2.3. Belief propagation, co-ordinate descent and auction algorithm

The optimization problem (66) is an integer program. All the feasible integral solutions of this optimization problem are perfect matchings in a complete bipartite graph or equivalently a permutation matrix. That is, $\mathbf{x} = [x_{ij}] \in \{0,1\}^{n\times n}$ where each row and column sums upto 1 (equality is assumed since the weights are non-negative and hence optimal will be achieved when all inequalities are tight). By Birkhoff and Von Neumann's result, these are precisely the extreme points of the space of all doubly stochastic matrices, i.e. $\mathbf{z} = [z_{ij}] \in [0,1]^{n\times n}$ so that all row-sums and column-sums are equal to 1. Therefore, solving (66) is equivalent to solving the following linear program:

$$\begin{aligned} \text{maximize} \quad & \sum_{1\le i,j\le n} z_{ij} q_{ij}^{\alpha} \\ \text{over} \quad & z_{ij} \in [0,1], \quad \text{for all} \;\; 1 \le i,j \le n \\ \text{subject to} \quad & \sum_k z_{ik} = 1, \quad \sum_k z_{kj} = 1, \quad \text{for all} \;\; 1 \le i,j \le n. \end{aligned} \tag{83}$$

To obtain a message-passing algorithm to solve this linear program, we turn to its dual:

$$\begin{aligned} \text{minimize} \quad & \sum_i r_i + \sum_j p_j \\ \text{subject to} \quad & r_i + p_j \ge q_{ij}^{\alpha}, \quad \text{for} \;\; \text{all } 1 \le i,j \le n. \end{aligned} \tag{84}$$

The co-ordinate descent algorithm for this optimization problem will iterative over r_is and p_js by updating their values as per following rule:

$$\begin{aligned} r_i^{\text{new}} &= \max_j \; \left(q_{ij}^{\alpha} - p_j^{\text{old}}\right), \\ p_j^{\text{new}} &= \max_i \; \left(q_{ij}^{\alpha} - r_i^{\text{old}}\right). \end{aligned} \tag{85}$$

While this is a quite simple and intuitively pleasing algorithm, due to the form of inequality constraints, it may not converge to the optimal solution of (85) in general. Further, even if it converged to the optimal dual solution, it is not clear how one can recover optimal primal solution. To overcome these issues, a clever method was proposed by Bertsekas (1992) known as the ϵ-relaxation method. The resulting algorithm is known as the auction algorithm. It maintains a feasible primal solution and modifies the dual co-ordinate descent algorithm:

in an (85) like update equation, an appropriate $\epsilon > 0$ is added as a penalty and such updates are performed in an appropriate order. Indeed for $\epsilon = 0$, it reduces to the update of (85). Bertsekas (1992) established that such a modification of co-ordinate descent algorithm, the auction algorithm, will always converge for any $\epsilon > 0$ in $O(n|\mathbf{q}^\alpha|/\epsilon)$ iterations; the resulting matching will have weight that is no less than the weight of the optimal matching by ϵn.

The auction algorithm or dual co-ordinate descent algorithm is tantalizingly closely connected to the belief propagation. To explain this, consider update of dual co-ordinate descent as per (85). The update of r_i and p_j are based on the following 'messages': define

$$\hat{m}_{I_i \to O_j} = q^\alpha_{ij} - r_i.$$

With this new definition and assuming that all updates of (85) are done in parallel, then we can re-write all the updates only in terms of $\hat{m}_{I_i \to O_j}$ for all $1 \leq i, j \leq n$ as follows: for iteration $t+1$,

$$\begin{aligned} \hat{m}^{(t+1)}_{I_i \to O_j} &= q^\alpha_{ij} - r^{(\mathrm{t+1})}_i \\ &= q^\alpha_{ij} - \big(\max_{j'} q^\alpha_{ij'} - p^{(\mathrm{t})}_{j'}\big) \\ &= q^\alpha_{ij} - \big(\max_{j'} q^\alpha_{ij'} - \big(\max_{i'} \hat{m}^{(t-1)}_{I_{i'} \to O_{j'}}\big)\big). \end{aligned} \tag{86}$$

Now let us re-write the message updates as per (81) under belief propagation only in terms of $m^{(\cdot)}_{I_i \to O_j}$ for all $1 \leq i, j \leq n$ (we shall ignore the index t for iteration):

$$\begin{aligned} m^{(t+1)}_{I_i \to O_j} &= q^\alpha_{ij} - \big(\max_{j' \neq j} m^{(t)}_{O_{j'} \to I_i}\big) \\ &= q^\alpha_{ij} - \big(\max_{j' \neq j} q^\alpha_{i'j'} - \big(\max_{i' \neq i} m^{(t-1)}_{I_{i'} \to O_{j'}}\big)\big). \end{aligned} \tag{87}$$

Note that (86) and (87) are essentially the same with the only exception that in belief propagation $j' \neq j$ and $i' \neq i$ is required in message update while such is not the case in dual co-ordinate descent. This syntactic similarity suggests a deeper connection between the belief propagation, dual co-ordinate descent and auction algorithm. It is worth noting that the overall complexity bound known for belief propagation and auction algorithm are essentially the same (but are based on very different proof methods). This relation between dual co-ordinate descent and belief propagation was made by Bayati et al. (2008b) for matching and by Sanghavi et al. (2009) for the maximum weight independent set.

4.2.4. Belief propagation: general stochastic network

The MW-α policy or α-fair policy in the context of stochastic network of general sort described through constraint (1) requires solving optimization of the following form:

$$\begin{aligned} \text{maximize} \quad & \sum_i F(q_i, x_i) \\ \text{over} \quad & \mathbf{x} \in \Sigma^N \\ \text{subject to} \quad & A\mathbf{x} \le \mathbf{C}. \end{aligned} \tag{88}$$

In above the function $F(q_i, \cdot)$ is linear or concave. In so far, we have explained message-passing algorithms for two special cases: a primal-dual algorithm for bandwidth sharing in the Internet, and belief propagation (as well as auction algorithm) for scheduling in an input-queued switch. In both cases, these algorithms provably solve the optimization problem of interest with reasonable complexity. When $\Sigma = \mathbb{R}_+$, the problem (88) is essentially solvable using algorithm described in the context of bandwidth sharing in the Internet. The problem becomes particularly hard when Σ is a discrete set. In such a case, it is not reasonable to expect message-passing algorithm to be able to solve the problem exactly in an efficient manner. Next we describe belief propagation based heuristic for this problem. Specifically, we shall assume that $\Sigma = \{1, \dots, K\}$. We shall end with remarks on conditions under which it is known to solve the problem exactly.

As described in Section 4.2.2, the basic idea behind belief propagation is to design an iterative procedure that mimics the evoluation of dynamic programing on tree graph (through HJB equation) at each node in the graph. In general, the graph is defined through the graphical model of Markov Random Field of the optimization problem (88) as described in Section 2.1. Specifically, $G = (U \cup V, E)$ is the graphical model where $U = \{1, \dots, N\}$ corresponds to variables $x_1, \dots, x_N$ and $V = \{1, \dots, M\}$ corresponds to inequality constraints $\sum_i A_{ji} x_i \le C_j$ for $1 \le j \le M$. And $E = \{(i, j) : i \in U,\ j \in V,\ A_{ji} \neq 0\}$. Let $\mathcal{N}_u(i) = \{j \in V : (i, j) \in E\}$ and $\mathcal{N}_v(j) = \{i \in U : (i, j) \in E\}$. The belief propagation algorithm for optimization problem (88) based on reasoning similar to that explained in Section 4.2.2 is described as follows.

Belief propagation: general stochastic network

1. Denote by t the iteration of the algorithm, with $t = 0$ initially. For each $(i, j) \in E$, $m^{(t)}_{i\to j}(k)$ (resp. $m^{(t)}_{j\to i}(k)$) denote message from node i to j (resp. j to i). The message $m^{(t)}_{i\to j}(k)$ (resp. $m^{(t)}_{j\to i}(k)$) represents belief of node i (resp. j) regarding cost of optimal assignment subject to $x_j = k$ (resp. $x_i = k$) for $1 \le k \le K$. Initially, for all $(i, j) \in E$

$$m^{(0)}_{i\to j}(k) = 0 \quad \text{and} \quad m^{(0)}_{j\to i}(k) = 0.$$

2. In iteration $t + 1$, update messages as follows: for $1 \le k \le K$

$$\begin{aligned} m^{(t+1)}_{i\to j}(k) &= F(q_i, k) + \sum_{j' \in \mathcal{N}_u(i)\backslash\{j\}} m^{(t)}_{j'\to i}(k), \\ m^{(t+1)}_{j\to i}(k) &= \max_{(k_{i'}) \in S(C_j, i, k)} \sum_{i' \in \mathcal{N}_v(j)\backslash\{i\}} m^{(t)}_{i'\to j}(k_{i'}), \end{aligned} \tag{89}$$

where $S(C_j, i, k) = \{(k_{i'}) : \sum_{i' \in \mathcal{N}_v(j)\backslash\{i\}} A_{ji'} k_{i'} \le C_j - A_{ji} k\}$.

3. Each node i estimates its optimal assignment at the end of iteration $t+1$ as

$$x^{(t+1)}_i \in \operatorname*{argmax}_{1\le k\le K} F(q_i, k) + \sum_{j \in \mathcal{N}_u(i)} m^{(t+1)}_{j\to i}(k). \tag{90}$$

4.3. Discussion, future direction

In this section, we discussed message-passing implementation of the fair bandwidth sharing policy and the maximum weight policy in the context of congestion control in the Internet and scheduling in a input-queued switch respectively. The primal-dual algorithm provides exact solutions for the fair bandwidth sharing policy for generic instance of stochastic processing network considered here as the underlying optimization problem of interest is concave maximization over a continuous, convex domain. The belief propagation provides exact solutions for the maximum weight policy in the context of input-queued switch. However, in general the maximum weight policy requires solving a combinatorial optimization problem and this is computationally hard in general. Therefore, it is unlikely to have an efficient, message-passing implementation for the exact maximum weight problem in general. And for that reason, belief propagation is unlikely to provide an exact solution.

The quest of understanding strengths and limitations of belief propagation heuristics is an active area of research. We quickly summarize the state-of-art. As mentioned in the section, the belief propagation is closely related to the auction algorithm, a modified dual co-ordinate descent algorithm, for the matching or assignment problem in the weighted bipartite graph. This suggests a strong relation between belief propagation and linear programing relaxation of combinatorial optimization problem as first observed by Bayati et al. (2008b). In the context of matching for general graph, this relation was made precise by Sanghavi et al. (2007) and Bayati et al. (2008a): belief propagation solves the maximum weight matching in a given graph if and only if the corresponding edge-based linear programming relaxation has unique integral solution. This relation, while tempting to conjecture to be true in the context of general combinatorial optimization, is in fact not true. This was established by producing a simple counter-example by Sanghavi et al. (2009) in the context of finding maximum weight independent set. Now on the other hand, it is true that whenever belief propagation solves the problem exactly, linear programming relaxation is likely to have integral solution (this was established in the context of independent set by Sanghavi et al. (2009) and is believed to be true more generally). Thus belief propagation solvable problems are roughly speaking contained in the set of problems solvable by linear programming relaxation. However, this containment does not seem too restrictive. Specifically Gamarnik et al. (2009) have shown that BP solves all network flow problems (in polynomial time), an important and large class of linear programming solvable problems (it includes matching in bipartite graph as a special instance). Further, a minor modification of belief propagation solves the maximum weight independent set in a bipartite graph as well. In summary, it seems that for a large class of combinatorial optimization problems, the belief propagation seems as powerful as their linear programming relaxation. Precise strengths and limitations of belief propagation remains an outstanding problem going forward.

It is worth remarking on related important results in the context of continuous optimization. Specifically, in a sequel of works, Moallemi and Van Roy (2008, 2009) showed that a class of unconstrained convex optimization problems can be solved efficiently by belief propagation. These results are related to analysis of belief propagation for what is known as the Gaussian Graphical Model, for example see work by Johnson et al. (2006) which provides interesting sets of convergence conditions for belief propagation in that context. These results have been recently improved by Ruozzi and Tatikonda (2010).

Now while it may seem that belief propagation is only as powerful as linear programming and may be one may use other linear programming based methods for solving the optimization problem arising in the context of scheduling, it is worth noting that belief propagation *does not* require any problem specific fine tuning – it is a one fixed recipe that applies to all sorts of problems; when problem is not too complicated, it seem to solve them exactly or else it provides a reasonable approximation. It would be interesting to understand how well does belief propagation performs as an approximation method. For example, an initial empirical study in the context of maximum weight independent set (in the context of wireless scheduling) suggests that when the linear programming relaxation does not have integral solution, the linear programming based methods provide very poor answers while the belief propagation (and its variants) seem to provide reasonable approximation, see Giaccone and Shah (2010) for details.

Finally, closer to the topic of this survey, in the context of designing scheduling in wireless networks, also known as the medium access protocol, where node can not even exchange messages explicitly, building upon insights provided by variational characterization of an appropriate product-form distribution, an efficient queue-based randomized algorithm has been proposed recently by Rajagopalan and Shah (2008); Rajagopalan et al. (2009); Shah and Shin (2009). In effect, this algorithm is a combination of variational approximation along with the Markov Chain Monte Carlo based method. Such a possibility of existence of an efficient algorithm without explicit message-passing has created a lot of excitement including other recent such works, for example by Jiang and Walrand (2008); Jiang et al. (2010). While these implementations provide efficiency in terms of long term throughput, they do suffer from large latency or delay. In general, designing algorithm with high throughput and low latency is impossible Shah et al. (2009). Therefore, an ideal solution would be an algorithm that utilizes given per-node budget of message-passing and computation in order to provide as high throughput as possible with a constraint on the finiteness of buffer-size.

Indeed, such an ideal solution is quite ambitious and will require us to take many steps towards it before even getting in its vicinity. Some initial steps are taken towards this: specifically, Shah and Shin (2010) utilize the *network geometry* to design low delay, high throughput medium access protocol with minimal message-passing in the context of wireless networks deployed in a geographic area in a reasonable manner.

5. Conclusions

In this survey, we have provided an overview of message-passing algorithms in the context of stochastic processing networks through two prototypical problems: estimation of loss rate which is equivalent to MARG in a Markov Random Field and scheduling as per myopic MW-α and α-fair policies which are equivalent to MAP problem in a Markov Random Field. The message-passing algorithms discussed in this survey were primarily based upon theory of optimization and variational approximation. While the message-passing algorithmic solutions described provide reasonable (exact or approximate) answers, there seem to be a long way before we can find ideal solution for both problems. Various concrete open problems were discussed in detail near the end of Sections 3 and 4.

Now the quest of efficient message-passing algorithms for the problems of MARG and MAP in a Markov Random Field, which is precisely of interest in the context of stochastic processing networks, has been of interest much more broadly and is very actively pursued in recent years. Therefore, we foresee that interaction between stochastic networks and message-passing through the interface of dynamic resource allocation problem can lead to a rich intellectual development. It is no surprise that such an interplay has started handsomely rewarding various other disciplines including de-randomization in computer science, e.g. Weitz (2006), Bandyopadhyay and Gamarnik (2006), Gamarnik and Katz (2007), Bayati et al. (2007a); combinatorial optimization, e.g. Gamarnik et al. (2006), Salez and Shah (2009), Gamarnik et al. (2010); statistical inference and computational geometry, e.g. Jung et al. (2009), high dimensional statistics Lu et al. (2008), Donoho et al. (2009), Chandar et al. (2010); and convex analysis Wainwright et al. (2005a,b). Even practically, these algorithms are seeing day of light. For example, message-passing algorithms have become crucial in designing codes for high bandwidth modern communication systems and have found place in IEEE standards (see book by Richardson and Urbanke (2008)).

In summary, message-passing algorithms will play an instrumental role in architecting large complex networked systems in the near future. Therefore, understanding strengths and limitations of existing algorithms as well as developing novel designs are urgent and of utmost importance. We believe that stochastic processing networks through interface of resource allocation will provide a fertile ground for their development.

Acknowledgment

The author wishes to thank all the collaborators he has interacted with over years on the topic of stochastic networks and message-passing algorithms. He wishes to thank Bert Zwart for encouragement to write this survey as well as providing detailed comments on the earlier draft of the manuscript to improve readability. He also wishes to thank anonymous reviewers for their careful reading and suggestions. Finally he wishes to acknowledge support of funding agencies NSF (through CNS CAREER, EMT and TF projects) and AFORS (through complex networks project) as well as the generous support of Jamieson family through the Career Development Chair at MIT.

References

Arora, S., Lund, C., 1996. Hardness of approximations. In Approximation Algorithms for NP-hard Problems, Ed. Dorit Hochbaum.

Bandyopadhyay, A., Gamarnik, D., 2006. Counting without sampling: new algorithms for enumeration problems using statistical physics. In: Proceedings of ACM-SIAM SODA. pp. 890–899.

Bayati, M., Borgs, C., Chayes, J., Zecchina, R., 2008a. On the exactness of the cavity method for weighted b-matchings on arbitrary graphs and its relation to linear programs. Journal of Statistical Mechanics: Theory and Experiment.

Bayati, M., Gamarnik, D., Katz, D., Nair, C., Tetali, P., 2007a. Simple deterministic approximation algorithms for counting matchings. In: Proceedings ACM STOC. pp. 122–127.

Bayati, M., Prabhakar, B., Shah, D., Sharma, M., 2007b. Iterative scheduling algorithm. In: Proceedings of IEEE Infocom. pp. 445–453.

Bayati, M., Shah, D., Sharma, M., 2008b. Max-product for maximum weight matching: convergence, correctness and lp duality. IEEE Transactions on Information Theory 54 (3), 1241–1251.

Bertsekas, D., 1995. Dynamic programming and optimal control. Athena Scientific Belmont, MA.

Bertsekas, D., Tsitsiklis, J., 1997. Parallel and distributed computation: Numerical methods. Athena Scientific.

Bertsekas, D. P., 1992. Auction algorithms for network flow problems: A tutorial introduction. Computational Optimization and Applications 1, 7–66.

Bertsimas, D., Vempala, S., 2004. Solving convex programs by random walks. Journal of the ACM (JACM) 51 (4), 5–56.

Bethe, H., 1935. Statistical theory of superlattices. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 150 (871), 552–575.

Bonald, T., Massoulie, L., 2001. Impact of fairness on internet performance. In: Proceedings of ACM Sigmetrics. pp. 91–102.

Chandar, V., Shah, D., Wornell, G., 2010. A simple message-passing algorithm for compressed sensing. In: Proceedings of IEEE ISIT. pp. 1968–1972.

Dai, J., Prabhakar, B., 2000. The throughput of switches with and without speed-up. In: Proceedings of IEEE Infocom. pp. 556–564.

Dai, J. G., Lin, W., 2005. Maximum pressure policies in stochastic processing networks. Operations Research 53 (2), 197–218.

Dai, J. G., Lin, W., 2008. Asymptotic optimality of maximum pressure policies in stochastic processing networks. Annals of Applied Probability 18 (6), 2239–2299.

de Veciana, G., Lee, T., Konstantopoulos, T., 2001. Stability and performance analysis of networks supporting elastic services. IEEE/ACM Transactions on Networking 9 (1), 2–14.

Donoho, D., Maleki, A., Montanari, A., 2009. Message-passing algorithms for compressed sensing. Proceedings of the National Academy of Sciences 106 (45), 18–26.

Dyer, M., Frieze, A., Kannan, R., 1991. A random polynomial-time algorithm for approximating the volume of convex bodies. Journal of the ACM (JACM) 38 (1), 1–17.

Gallager, R., 1962. Low-density parity-check codes. IRE Transactions on Information Theory 8 (1), 21–28.

Gamarnik, D., Goldberg, D., Weber, T., 2010. Average-case complexity of Maximum Weight Independent Set with random weights in bounded degree graphs. In: Proceedings of ACM-SIAM SODA.

Gamarnik, D., Katz, D., 2007. Correlation decay and deterministic FPTAS for counting list-colorings of a graph. In: Proceedings of ACM-SIAM SODA.

Gamarnik, D., Nowicki, T., Swirszcz, G., 2006. Maximum weight independent sets and matchings in sparse random graphs. Exact results using the local weak convergence method. Random Structures & Algorithms 28 (1), 76–106.

Gamarnik, D., Shah, D., Wei, Y., 2009. Belief Propagation for Min-cost Network Flow: Convergence & Correctness. In: Proceedings of ACM-SIAM SODA.

Geman, S., Geman, D., 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 721–741.

Georgiadis, L., Neely, M., Tassiulas, L., 2006. Resource allocation and cross layer control in wireless networks. Foundations and Trends in Networking, Now Publishers Inc.

Georgii, H. O., 1988. Gibbs Measures and Phase Transitions. New York: De Gruyter.

Giaccone, P., Shah, D., 2010. Message-passing for wireless scheduling: an experimental study. In: IEEE International Conference on Computer Communication Networks (ICCN 2010).

Gilks, W., Richardson, S., Spiegelhalter, D., 1996. Markov Chain Monte Carlo in Practice. New York: Chapman and Hall.

Gromoll, H. C., Williams, R. J., 2009. Fluid limits for networks with bandwidth sharing and general document size distributions. Annals of Applied Probability 19 (1).

Harrison, J. M., 2000. Brownian models of open processing networks: canonical representation of workload. Annals of Applied Probability 10, 75–103.

Jiang, L., Shah, D., Shin, J., Walrand, J., 2010. Distributed random access algorithm: Scheduling and congesion control. IEEE Transactions on Information Theory.

Jiang, L., Walrand, J., 2008. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. In: Proceedings of Allerton conference on Communication, Control, and Computing. pp. 1511–1519.

Johnson, J., Malioutov, D., Willsky, A., 2006. Walk-sum interpretation and analysis of Gaussian belief propagation. Advances in Neural Information Processing Systems 18.

Jung, K., Kohli, P., Shah, D., 2009. Local Rules for Global MAP: When Do They Work? In: Proceedings of Neural Information Processing Systems.

Jung, K., Lu, Y., Shah, D., Sharma, M., Squillante, M., 2008. Revisiting stochastic loss networks: Structures and algorithms. In: Proceedings of ACM Sigmetrics.

Kalai, A., Vempala, S., 2006. Simulated annealing for convex optimization. Mathematics of Operations Research 31 (2), 253–266.

Kelly, F., 1986. Blocking probabilities in large circuit-switched networks. Advances in Applied Probability 18 (2), 473–505.

Kelly, F., 1991. Loss networks. Annals of Applied Probability 1 (3), 319–378.

Kelly, F., Maulloo, A., Tan, D., 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research society 49 (3), 237–252.

Kelly, F. P., Williams, R. J., 2004. Fluid model for a network operating under a fair bandwidth-sharing policy. Annals of Applied Probability 14, 1055–1083.

Lauritzen, S. L., 1996. Graphical Models. Oxford: Oxford University Press.

Little, J. D. C., 1961. A proof of the queueing formula $l = \lambda$ w. Operations Research 9, 383–387.

Lovász, L., Vempala, S., 2003. Logconcave functions: Geometry and efficient sampling algorithms. In: Proceedings of IEEE FOCS. pp. 640–649.

Lu, Y., Montanari, A., Prabhakar, B., Dharmapurikar, S., Kabbani, A., 2008. Counter braids: a novel counter architecture for per-flow measurement. In: Proceedings ACM Sigmetrics. pp. 121–132.

Luo, Z., Tseng, P., 1992. On the convergence of the coordinate descent method for convex differentiable minimization. Journal of Optimization Theory and Applications 72 (1), 7–35.

McKeown, N., Anantharam, V., Walrand, J., 1996. Achieving 100% throughput in an input-queued switch. In: Proceedings of IEEE Infocom. pp. 296–302.

Mezard, M., Montanari, A., 2009. Information, physics, and computation. Oxford University Press, USA.

Mezard, M., Parisi, G., Virasoro, M., 1987. Spin glass theory and beyond. World scientific Singapore.

Mezard, M., Parisi, G., Zecchina, R., 2002. Analytic and algorithmic solution of random satisfiability problems. Science 297.

Mo, J., Walrand, J., 1998. Fair end-to-end window-based congestion control. In: Internation Symposium on Voice, Video and Data Communications (SPIE).

Moallemi, C., Van Roy, B., 2008. Convergence of the min-sum algorithm for convex optimization. In: Proceedings of Allerton Conference on Communication, Control and Computing.

Moallemi, C., Van Roy, B., 2009. Convergence of min-sum message passing for quadratic optimization. IEEE Transactions on Information Theory 55 (5), 2413–2423.

Nemhauser, G. L., Wolsey, L. A., 1999. Integer and Combinatorial Optimization. New York: Wiley-Interscience.

Ni, J., Tatikonda, S., 2007. Analyzing product-form stochastic networks via factor graphs and the sum-product algorithm. IEEE Transactions on Communications 55 (8), 1588–1597.

Parisi, G., 1988. Statistical Field Theory. New York: Addison-Wesley.

Pearl, J., 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

Pevzner, P., 2000. Computational Molecular Biology: An Algorithmic Approach. Cambridge, MA: MIT Press.

Rajagopalan, S., Shah, D., 2008. Distributed algorithm and reversible network. In: Proceedings of Annual Conference on Information Sciences and Systems (CISS). pp. 498–502.

Rajagopalan, S., Shah, D., Shin, J., 2009. Network adiabatic theorem: An efficient randomized protocol for contention resolution. In: Proceedings ACM Sigmetrics/Performance. pp. 133–144.

Richardson, T., Urbanke, R., 2008. Modern coding theory. Cambridge University Press.

Roberts, J., Massoulie, L., 2000. Bandwidth sharing and admission control for elastic traffic. Telecommunication Systems 15, 185–201.

Rockafellar, R. T., 1998. Network flow and monotropic optimization. Athena Scientific.

Ruozzi, N., Tatikonda, S., 2010. Convergent and Correct Message Passing Schemes for Optimization Problems over Graphical Models. Arxiv preprint arXiv:1002.3239.

Salez, J., Shah, D., 2009. Belief propagation: an asymptotically optimal algorithm for the random assignment problem. Mathematics of Operations Research 34 (2), 468–480.

Sanghavi, S., Malioutov, D., Willsky, A., 2007. Belief propagation and lp relxation for weighted matching in general graphs. In: Proceedings of Neural Information Processing System.

Sanghavi, S., Shah, D., Willsky, A., 2009. Message passing for maximum weight independent set. IEEE Transactions on Information Theory 55 (11), 4822–4834.

Shah, D., 2008. Network Scheduling and Message-passing. Performance Modeling and Engineering, 147–184.

Shah, D., 2009. Gossip algorithms. Foundations and Trends in Networking, Now Publishers Inc.

Shah, D., Shin, J., 2009. Randomized Scheduling Algorithm for Queueing Networks. Arxiv preprint arXiv:0908.3670.

Shah, D., Shin, J., 2010. Delay optimal queue-based csma. In: ACM Sigmetrics/Performance.

Shah, D., Tse, D. N. C., Tsitsiklis, J. N., 2009. Hardness of low delay scheduling. IEEE Transactions on information theory, under revision.

Shah, D., Wischik, D., 2011. Switched networks with maximum weight policies: Fluid approximation and state space collapse. Accepted to appear in Annals of Applied Probability.

Shakkottai, S., Srikant, R., 2007. Network optimization and control. Foundations and Trends in Networking, Now Publishers Inc.

Srikant, R., 2004. The mathematics of Internet congestion control. Birkhauser.

Tassiulas, L., Ephremides, A., 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Transactions on Automatic Control 37, 1936–1948.

Tatikonda, S., Jordan, M. I., 2002. Loopy belief propagation and gibbs measures. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence. pp. 493–500.

Valiant, L., 1979. The complexity of computing the permanent. Theoretical computer science 8 (2), 189–201.

Wainwright, M., Jordan, M., 2008. Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning, Now Publishers Inc. 1 (1-2), 1–305.

Wainwright, M. J., Jaakkola, T. S., Willsky, A. S., 2005a. Exact map estimates via agreement on (hyper)trees: Linear programming and message-passing. IEEE Transactions on Information Theory 51 (11), 3697–3717.

Wainwright, M. J., Jaakkola, T. S., Willsky, A. S., 2005b. A new class of upper bounds on the log partition function. IEEE Transactions on Information Theory 51 (7), 2313–2335.

Weitz, D., 2006. Counting down the tree. In: Proceedings of ACM STOC.

Yedidia, J., Freeman, W., Weiss, Y., 2001. Generalized belief propagation. Proceedings of Neural Information Processing System.