Supplementary Information Appendix
From the Physics of Interacting Polymers to Optimizing Routes on the London Underground

Chi Ho Yeung, David Saad and K. Y. Michael Wong

June 20, 2013

Contents

1 Analytical Solution by the Replica Approach **2**
1.1 The partition function 2
1.2 Replica calculation and the saddle point equations 3
1.3 The functional order parameter ansatze at finite temperature 10
1.3.1 The expression for $\hat{P}(\mathbf{S}, \hat{\underline{\lambda}})$ 10
1.3.2 The expression for$\hat{P}(\mathbf{S}, \hat{\underline{\lambda}})$ 13
1.3.3 Recursive equation for $w(\vec{h})$ 15
1.3.4 Further simplification of the equations 16
1.3.5 The derivation of $\rho(\vec{\lambda}^*|\vec{h})$ 19
1.3.6 Summary of the solution for finite M 24
1.4 The zero temperature solution 25
1.5 The directed formulation 26

2 The algorithm **29**
2.1 Introducing quenched random bias 31
2.2 Computational complexity 32
2.3 Convergence 32
2.4 Comparison with the conventional cavity approach 33
2.5 Discrepancy between simulations and analytic solution 36

3 Generalized formulation and algorithms **37**
3.1 The algorithm for weighted communications 37
3.2 The algorithm for costs defined on edges 39
3.3 The algorithm for separate costs defined on each direction of edges 42

4 Comparison with existing multi-commodity flow optimization algorithms **44**

5 The emergence of replica symmetric breaking (RSB) **49**

6 The optimal path length in ER and SF graphs **50**

7 The fraction of idle nodes and the phase transition at $\gamma = 1$ **51**

1 Analytical Solution by the Replica Approach

Here we employ the replica approach [S1, S2] and a method from polymer physics [S3] to solve the system of interacting polymers, which resembles the interaction of communication paths on a sparse graph. Although the derivation is rather involved, we are able to simplify the final equations and obtain a simple recursive equation resembling the sparse graph cavity equation in disordered systems [S1, S2]. We suggest readers who are only interested in the final solution to jump directly to Sections S1.3.6 and S1.4 where we provide a short summary for the finite temperature and zero-temperature (optimized) solutions, respectively. For readers who would like to follow the whole derivation, a glance at these final results may clarify the main objectives of the calculation. In Section S1.3.6 we also discuss the possible application of the cavity approach, which is easier to be converted to an algorithm, and its limitations.

1.1 The partition function

We first write down the partition function $\mathcal{Z}$ of a system of M polymers interacting on a network of N nodes, subject to the Hamiltonian $\mathcal{H} = M\sum_i \phi(I_i/M)$ with I_i the number of polymers/routes passing through i. We will make use a method developed in polymer science, termed the 0-vector model [S3], commonly employed to analyze self-avoiding walk [S4, S5]. Denote $\int_\odot d\vec{S}$ as the angular integration over an n-component vector $\vec{S}$ of length $|\vec{S}| = \sqrt{n}$ and the normalization factor $C_n = \int_\odot d\vec{S}$, it was shown in [S3] that all positive moments of any component S_a vanishes in the limit $n \to 0$ except for the term $\frac{1}{C_n}\int_\odot d\vec{S} S_a^2 = 1$ for each component a in $\vec{S}$. It then implies when $n \to 0$ all the terms which contribute in

$$\prod_{i=1}^{N}\left(\frac{1}{C_n}\int_\odot d\vec{S}_i\right) S_{x,a}S_{y,a}\prod_{(kl)}\left(1 + A_{kl}\vec{S}_k\cdot\vec{S}_l\right) \ , \tag{S1}$$

where parenthesis (k,l) represent ordered variables, such that $k < l$. The only surviving terms are of the form $A_{xk_1}A_{k_1k_2}\cdots A_{k_ly}S_{x,a}^2S_{k_1,a}^2S_{k_2,a}^2\cdots S_{k_l,a}^2S_{y,a}^2$, that represent a self-avoiding path $(x, k_1, k_2, \ldots, k_l, y)$ joining nodes x and y [S3]. Each surviving path contributes exactly 1 so that Eq. (S1) counts the number of heterogeneous route between x and y. We make use of this method to write the partition function of our polymer model as

$$\mathcal{Z} = \prod_{i=1}^{N} \left[\int \frac{d\lambda_i d\hat{\lambda}_i}{2\pi} e^{iM\lambda_i\hat{\lambda}_i - \beta M\phi(\lambda_i)} \right] \prod_{i\nu} \left[\frac{1}{C_n} \int_{\odot} d\vec{S}_i^{\nu} \right] \\ \times \prod_{\nu=1}^{M} \left[\prod_{(ij)} \left(e^{-\frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}} S_{i,a}^{\nu} S_{j,a}^{\nu} \right)^{\Lambda_{(ij)}^{\nu}} \prod_{(kl)} \left(1 + A_{kl} e^{-\frac{i\hat{\lambda}_k + i\hat{\lambda}_l}{2}} \vec{S}_k^{\nu} \cdot \vec{S}_l^{\nu} \right) \right] \tag{S2}$$

where each factor in the product over ν is identical to Eq. (S1) except the additional factors $e^{-\frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}}$ and $e^{-\frac{i\hat{\lambda}_k + i\hat{\lambda}_l}{2}}$ (note that the superindex ν in $S_{i,a}^{\nu}$ is the polymer/communication index and not a power). It implies for each polymer/route ν the path choice $(i, k_1, \ldots, k_l, j)$ contributes a factor $e^{-i\hat{\lambda}_i - i\hat{\lambda}_{k_1} - \cdots - i\hat{\lambda}_{k_l} - i\hat{\lambda}_j}$, and the product over ν leads to the term $e^{-i\sum_{i=0}^{N} I_i\hat{\lambda}_i}$ where I_i is the number of polymers/routes passing through i. Finally, the integration over $\hat{\lambda}_i$ corresponds to a delta function which replaces λ_i in $\phi(\lambda_i)$ by I_i/M, giving rise to the correct Boltzmann factor $e^{-\beta M\sum_i \phi(I_i/M)}$. We choose to write $\mathcal{Z}$ in a relatively complicated way as it allows for a generic cost function ϕ throughout the calculation.

1.2 Replica calculation and the saddle point equations

Here we employ the replica approach, which makes use of the trick $\overline{\log \mathcal{Z}} = \lim_{m\to 0} \frac{\overline{\mathcal{Z}^m} - 1}{m}$ to compute the average of $\overline{\log \mathcal{Z}}$ by evaluating $\mathcal{Z}^m$ averaged over the quenched disorders A_{ij} and $\Lambda_{(ij)}^{\nu}$ and finally take the limit of $m \to 0$. Here $\overline{\cdots}$ corresponds to the average over the quenched disorder.

We first write $\mathcal{Z}^m$

$$\mathcal{Z}^m = \prod_{i\alpha} \left[\int \frac{d\lambda_i^{\alpha} d\hat{\lambda}_i^{\alpha}}{2\pi} e^{iM\lambda_i^{\alpha}\hat{\lambda}_i^{\alpha} - \beta M\phi(\lambda_i^{\alpha})} \right] \prod_{i\nu a} \left[\frac{1}{C_n} \int_{\odot} d\vec{S}_{i\nu}^{\alpha} \right] \\ \times \prod_{\nu\alpha} \left[\prod_{(ij)} \left(e^{-\frac{i\hat{\lambda}_i^{\alpha} + i\hat{\lambda}_j^{\alpha}}{2}} S_{i\nu,a}^{\alpha} S_{j\nu,a}^{\alpha} \right)^{\Lambda_{(ij)}^{\nu}} \prod_{(kl)} \left(1 + A_{kl} e^{-\frac{i\hat{\lambda}_k^{\alpha} + i\hat{\lambda}_l^{\alpha}}{2}} \vec{S}_{k\nu}^{\alpha} \cdot \vec{S}_{l\nu}^{\alpha} \right) \right] \tag{S3}$$

where α is the replica index, and subscript ν of S the communication/polymer index. Since we aim to analyze the ground state behavior of the system on a sparse graph with arbitrary degree distribution $\rho(k)$, we will mainly follow [S6] in the first half of the derivation to decouple the interaction of nodes on sparse graphs. The distribution of the disorder variables A_{ij} and $\Lambda^{\nu}_{(ij)}$ are given by

$$p\left(\{\Lambda^{\nu}_{(ij)}\}\right) = \prod_{\nu}\left[\frac{2}{N(N-1)}\sum_{(ij)}\delta_{\Lambda^{\nu}_{(ij)},1}\prod_{(kl)\neq(ij)}\delta_{\Lambda^{\nu}_{(kl)},0}\right] \tag{S4}$$

$$\begin{aligned} p\left(\{A_{ij}\}\right) &= \frac{1}{\mathcal{D}_A}\prod_{(ij)}\left[\left(1-\frac{\langle k\rangle}{N}\right)\delta_{A_{ij},0}+\frac{\langle k\rangle}{N}\delta_{A_{ij},1}\right]\prod_{i}\left[\delta\left(\sum_{j}A_{ij}-k_i\right)\right] \\ &= \frac{1}{\mathcal{D}_A}\prod_{(ij)}\left[\left(1-\frac{\langle k\rangle}{N}\right)\delta_{A_{ij},0}+\frac{\langle k\rangle}{N}\delta_{A_{ij},1}\right]\prod_{i}\left[\oint\frac{dZ_i}{2\pi i}Z_i^{-(k_i+1)}\right]\prod_{(ij)}\left(Z_iZ_j\right)^{A_{ij}} \end{aligned} \tag{S5}$$

where $\langle k\rangle = \sum_{k=1}^{\infty}k\rho(k)$ is the average node degree and the integration over Z_i is an integral representation of the delta function $\delta\left(\sum_j A_{(ij)}-k_i\right)$. The average over the degree distribution $\rho(k)$ will be introduced later. The resulting representation of $p\left(\{A_{ij}\}\right)$ with the denominator $\mathcal{D}_A$

$$\mathcal{D}_A = \prod_{(ij)}\sum_{A_{ij}=0,1}\left[\left(1-\frac{\langle k\rangle}{N}\right)\delta_{A_{ij},0}+\frac{\langle k\rangle}{N}\delta_{A_{ij},1}\right]\prod_{i}\left[\oint\frac{dZ_i}{2\pi i}Z_i^{-(k_i+1)}\right]\prod_{(ij)}\left(Z_iZ_j\right)^{A_{ij}} \tag{S6}$$

gives rise to the correct degree probability distribution $\prod_i \frac{\langle k\rangle^{k_i}}{k_i!}e^{-\langle k\rangle}$, which follows the Poisson distribution of mean $\langle k\rangle$. We then average $\mathcal{Z}^m$ over the above distributions to obtain

$$\begin{aligned} \overline{\mathcal{Z}^m} \propto &\prod_{i\alpha}\left[\int\frac{d\lambda_i^{\alpha}d\hat{\lambda}_i^{\alpha}}{2\pi}e^{iM\lambda_i^{\alpha}\hat{\lambda}_i^{\alpha}-\beta M\phi\left(\lambda_i^{\alpha}\right)}\right]\prod_{i\nu\alpha}\left[\int_{\odot}d\vec{S}^{\alpha}_{i\nu}\right]\prod_{i}\left[\oint\frac{dZ_i}{2\pi i}Z_i^{-(k_i+1)}\right] \\ &\times\prod_{\nu}\left\{\left[\sum_{(ij)}\prod_{\alpha}\left(e^{-\frac{i\hat{\lambda}_i^{\alpha}+i\hat{\lambda}_j^{\alpha}}{2}}S^{\alpha}_{i\nu,a}S^{\alpha}_{j\nu,a}\right)\right]\prod_{(kl)}\left[1-\frac{\langle k\rangle}{N}+\frac{\langle k\rangle}{N}Z_kZ_l\prod_{\nu\alpha}\left(1+e^{-\frac{i\hat{\lambda}_k^{\alpha}+i\hat{\lambda}_l^{\alpha}}{2}}\vec{S}^{\alpha}_{k\nu}\cdot\vec{S}^{\alpha}_{l\nu}\right)\right]\right\} \end{aligned} \tag{S7}$$

To define macroscopic order parameters we then introduce delta functions as follows:

$$
\begin{aligned}
\overline{\mathcal{Z}^m} \propto \frac{1}{\mathcal{D}_A} \prod_{i\alpha} \left[\int \frac{d\lambda_i^\alpha d\hat{\lambda}_i^\alpha}{2\pi} e^{iM\lambda_i^\alpha \hat{\lambda}_i^\alpha - \beta M \phi(\lambda_i^\alpha)} \right] \prod_{i\nu a} \left[\int_\odot d\vec{S}_{i\nu}^\alpha \right] \prod_i \left[\oint \frac{dZ_i}{2\pi i} Z_i^{-(k_i+1)} \right] \\
\times \prod_\nu \Bigg\{ \sum_{(ij)} \prod_\alpha \Bigg[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu'^\alpha \int d\hat{\lambda}_\alpha \int d\hat{\lambda}_\alpha' \\
\delta(S_{\nu,a}^\alpha - S_{i\nu,a}^\alpha) \delta(S_{\nu,a}'^\alpha - S_{j\nu,a}^\alpha) \delta(\hat{\lambda}_\alpha - \hat{\lambda}_i^\alpha) \delta(\hat{\lambda}_\alpha' - \hat{\lambda}_j^\alpha) \left(e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}_\alpha'}{2} S_{\nu,a}^\alpha S_{\nu,a}'^\alpha} \right) \Bigg] \Bigg\} \\
\times \exp \Bigg\{ \frac{\langle k \rangle}{2N} \sum_{k,l} Z_k Z_l \prod_{\nu\alpha} \left[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu'^\alpha \right] \prod_\alpha \left[\int d\hat{\lambda}_\alpha \int d\hat{\lambda}_\alpha' \right] \\
\prod_{\nu\alpha} [\delta(\vec{S}_\nu^\alpha - \vec{S}_{k\nu}^\alpha) \delta(\vec{S}_\nu'^\alpha - \vec{S}_{l\nu}^\alpha)] \prod_\alpha [\delta(\hat{\lambda}_\alpha - \hat{\lambda}_k^\alpha) \delta(\hat{\lambda}_\alpha' - \hat{\lambda}_l^\alpha)] \prod_{\nu\alpha} \left(1 + e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}_\alpha'}{2} \vec{S}_\nu^\alpha \cdot \vec{S}_\nu'^\alpha} \right) \Bigg\}
\end{aligned}
\tag{S8}
$$

where we arrive at the last exponential factor by (i) assuming $\langle k \rangle / N$ is small, which is justified in sparse graph and (ii) neglecting a term $k = l$ which is order N, compared to the summation over k and l which is of order N^2. We then introduce the so-called *functional order parameters* [S6] given by

$$
P(\mathbf{S}, \underline{\hat{\lambda}}) = \frac{1}{N} \sum_i Z_i \prod_{\nu\alpha} [\delta(\vec{S}_\nu^\alpha - \vec{S}_{i\nu}^\alpha)] \prod_\alpha [\delta(\hat{\lambda}_\alpha - \hat{\lambda}_i^\alpha)] \tag{S9}
$$

$$
Q(\underline{S}_{\nu,a}, \underline{\hat{\lambda}}) = \frac{1}{N} \sum_i \prod_\alpha [\delta(S_{\nu,a}^\alpha - S_{i\nu,a}^\alpha) \delta(\hat{\lambda}_\alpha - \hat{\lambda}_i^\alpha)], \tag{S10}
$$

where $\mathbf{S}$ corresponds to a vector of the variables $\vec{S}_\nu^\alpha$ over the labels ν and α, $\underline{\hat{\lambda}}$ and $\underline{S}_{\nu,a}$ correspond to the vectors of $\hat{\lambda}_\alpha$ and $S_{\nu,a}^\alpha$ over the label α. One then arrives at the following expression

$$
\begin{aligned}
\overline{\mathcal{Z}^m} \propto \frac{1}{\mathcal{D}_A} \int \prod_{\{\mathbf{S},\underline{\hat{\lambda}}\}} dP(\mathbf{S},\underline{\hat{\lambda}}) d\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \prod_{\{\underline{S}_{\nu,a},\underline{\hat{\lambda}}\}} dQ(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) d\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \\
\times \prod_{i\alpha} \left[\int \frac{d\lambda_i^\alpha d\hat{\lambda}_i^\alpha}{2\pi} e^{iM\lambda_i^\alpha \hat{\lambda}_i^\alpha - \beta M \phi(\lambda_i^\alpha)} \right] \prod_{i\nu a} \left[\int_\odot d\vec{S}_{i\nu}^\alpha \right] \prod_i \left[\oint \frac{dZ_i}{2\pi i} Z_i^{-(k_i+1)} \right] \\
\times \prod_\nu \exp \left\{ \prod_\alpha \left[\int_\odot d\vec{S}_\nu^\alpha \int d\hat{\lambda}_\alpha \right] Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \right. \\
\left. - \frac{1}{N} \prod_\alpha \left[\int_\odot d\vec{S}_\nu^\alpha \int d\hat{\lambda}_\alpha \right] \sum_i \prod_\alpha [\delta(S_{\nu,a}^\alpha - S_{i\nu,a}^\alpha) \delta(\hat{\lambda}_\alpha - \hat{\lambda}_i^\alpha)] \hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \right\} \\
\times \exp \left\{ \prod_{\nu\alpha} \left[\int_\odot d\vec{S}_\nu^\alpha \right] \prod_\alpha \left[\int d\hat{\lambda}_\alpha \right] P(\mathbf{S},\underline{\hat{\lambda}}) \hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \right. \\
\left. - \frac{1}{N} \prod_{\nu\alpha} \left[\int_\odot d\vec{S}_\nu^\alpha \right] \prod_\alpha \left[\int d\hat{\lambda}_\alpha \right] \sum_i Z_i \prod_{\nu\alpha} [\delta(\vec{S}_\nu^\alpha - \vec{S}_{i\nu}^\alpha)] \prod_\alpha [\delta(\hat{\lambda}_\alpha - \hat{\lambda}_i^\alpha)] \hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \right\} \\
\times \prod_\nu \left\{ \frac{N^2}{2} \prod_\alpha \left[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu'^\alpha \int d\hat{\lambda}_\alpha d\hat{\lambda}'_\alpha \right] Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) Q(\vec{S}'_{\nu,a},\underline{\hat{\lambda}}') \prod_\alpha \left(e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}'_\alpha}{2}} S_{\nu,a}^\alpha S_{\nu,a}'^\alpha \right) \right\} \\
\times \exp \left\{ \frac{N\langle k\rangle}{2} \prod_{\nu\alpha} \left[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu'^\alpha \right] \prod_\alpha \left[\int d\hat{\lambda}_\alpha d\hat{\lambda}'_\alpha \right] \right. \\
\left. P(\vec{S},\underline{\hat{\lambda}}) P(\mathbf{S}',\underline{\hat{\lambda}}') \prod_{\nu\alpha} \left(1 + e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}'_\alpha}{2}} \vec{S}_\nu^\alpha \cdot \vec{S}_\nu'^\alpha \right) \right\}
\end{aligned}
\tag{S11}
$$

Although Eq. (S11) looks complicated, one can easily integrate the delta functions on the 4th, and the 6th lines. After making the changes of variables

$$\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \to -N\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \tag{S12}$$

$$\hat{Q}(\vec{S}_\nu,\underline{\hat{\lambda}}) \to -N\hat{Q}(\vec{S}_\nu,\underline{\hat{\lambda}}), \tag{S13}$$

we arrive at

$$
\begin{aligned}
\overline{\mathcal{Z}^m} \propto & \int \prod_{\{\mathbf{S},\underline{\hat{\lambda}}\}} dP(\mathbf{S},\underline{\hat{\lambda}}) d\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \prod_{\{\underline{S}_{\nu,a},\underline{\hat{\lambda}}\}} dQ(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) d\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \\
& \times \exp\Bigg\{ -N \sum_\nu \prod_\alpha \left[\int_\odot d\vec{S}_\nu^\alpha \int d\hat{\lambda}_\alpha \right] Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \\
& \qquad -N \prod_{\nu\alpha} \left[\int_\odot d\vec{S}_\nu^\alpha \right] \prod_\alpha \left[\int d\hat{\lambda}_\alpha \right] P(\mathbf{S},\underline{\hat{\lambda}}) \hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \Bigg\} \\
& \times \frac{1}{\mathcal{D}_A} \Bigg\{ \prod_{i\alpha} \left[\int \frac{d\lambda_i^\alpha d\hat{\lambda}_i^\alpha}{2\pi} e^{iM\lambda_i^\alpha \hat{\lambda}_i^\alpha - \beta M \phi(\lambda_i^\alpha)} \right] \prod_{i\nu a} \left[\int_\odot d\vec{S}_{i\nu}^\alpha \right] \prod_i \left[\oint \frac{dZ_i}{2\pi i} Z_i^{-(k_i+1)} \right] \\
& \qquad \times \prod_i e^{Z_i \hat{P}(\vec{S}_i,\underline{\hat{\lambda}}_i) + \sum_\nu \hat{Q}_\nu(\vec{S}_i^\nu,\underline{\hat{\lambda}}_i)} \Bigg\} \\
& \times \exp\Bigg\{ N \frac{1}{N} \sum_\nu \log \Bigg\{ \frac{N^2}{2} \prod_\alpha \left[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu'^\alpha \int d\hat{\lambda}_\alpha \int d\hat{\lambda}_\alpha' \right] \\
& \qquad \qquad \times Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) Q(\underline{S}'_{\nu,a},\underline{\hat{\lambda}}') \prod_\alpha \left(e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}_\alpha'}{2}} S_{\nu,a}^\alpha S_{\nu,a}'^\alpha \right) \Bigg\} \Bigg\} \\
& \times \exp\Bigg\{ \frac{N\langle k \rangle}{2} \prod_{\nu\alpha} \left[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu'^\alpha \right] \prod_\alpha \left[\int d\hat{\lambda}_\alpha d\hat{\lambda}_\alpha' \right] \\
& \qquad \qquad \times P(\vec{S},\underline{\hat{\lambda}}) P(\mathbf{S}',\underline{\hat{\lambda}}') \prod_{\nu\alpha} \left(1 + e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}_\alpha'}{2}} \vec{S}_\nu^\alpha \cdot \vec{S}_\nu'^\alpha \right) \Bigg\} . \qquad \text{(S14)}
\end{aligned}
$$

One can now integrate the variable Z_i which involves only the 4th and the 5th line. With the normalization factor $\mathcal{D}_A = \prod_i \frac{\langle k \rangle^{k_i}}{k_i!} e^{-\langle k \rangle}$ given by Eq. (S6), the 4th and the 5th line become

$$
\prod_i \left\{ \prod_\alpha \left[\int \frac{d\lambda_i^\alpha d\hat{\lambda}_i^\alpha}{2\pi} e^{iM\lambda_i^\alpha \hat{\lambda}_i^\alpha - \beta M \phi(\lambda_i^\alpha)} \right] \prod_{\nu a} \left[\int_\odot d\vec{S}_{i\nu}^\alpha \right] \frac{\left[\hat{P}(\vec{S}_i,\underline{\hat{\lambda}}_i) \right]^{k_i}}{\langle k \rangle^{k_i}} e^{\sum_\nu \hat{Q}_\nu(\vec{S}_i^\nu,\underline{\hat{\lambda}}_i)} \right\} . \qquad \text{(S15)}
$$

One can see that except for the first line in Eq. (S14), all the terms in Eq. (S14) are either exponential in N or factorized in node index i, such that

$$
\overline{\mathcal{Z}^m} \propto \int \prod_{\{\mathbf{S},\underline{\hat{\lambda}}\}} dP(\mathbf{S},\underline{\hat{\lambda}}) d\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \prod_{\{\underline{S}_{\nu,a},\underline{\hat{\lambda}}\}} dQ(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) d\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) e^{N\Psi} \qquad \text{(S16)}
$$

with

$$
\begin{aligned}
\Psi = &-\sum_{\nu}\prod_{\alpha}\left[\int_{\odot} d\vec{S}_{\nu}^{\alpha}\int d\hat{\lambda}_{\alpha}\right] Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}})\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) \\
&-\prod_{\nu\alpha}\left[\int_{\odot} d\vec{S}_{\nu}^{\alpha}\right]\prod_{\alpha}\left[\int d\hat{\lambda}_{\alpha}\right] P(\mathbf{S},\underline{\hat{\lambda}})\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) \\
&+\log\left\{\prod_{\alpha}\left[\int\frac{d\lambda_{\alpha}d\hat{\lambda}_{\alpha}}{2\pi}e^{iM\lambda_{\alpha}\hat{\lambda}_{\alpha}-\beta M\phi(\lambda_{\alpha})}\right]\prod_{\nu a}\left[\int_{\odot} d\vec{S}_{\nu}^{\alpha}\right]\frac{\left[\hat{P}(\mathbf{S},\underline{\hat{\lambda}})\right]^{k}}{\langle k\rangle^{k}}e^{\sum_{\nu}\hat{Q}_{\nu}(\mathbf{S}_{\nu},\underline{\hat{\lambda}})}\right\} \\
&+\frac{1}{N}\sum_{\nu}\log\left\{\frac{N^2}{2}\prod_{\alpha}\left[\int_{\odot} d\vec{S}_{\nu}^{\alpha}\int_{\odot} d\vec{S}_{\nu}^{\prime\alpha}\int d\hat{\lambda}_{\alpha}\int d\hat{\lambda}_{\alpha}^{\prime}\right]\right. \\
&\qquad\left. Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}})Q(\vec{S}_{\nu,a}^{\prime},\underline{\hat{\lambda}}^{\prime})\prod_{\alpha}\left(e^{-\frac{i\hat{\lambda}_{\alpha}+i\hat{\lambda}_{\alpha}^{\prime}}{2}S_{\nu,a}^{\alpha}S_{\nu,a}^{\prime\alpha}}\right)\right\} \\
&+\frac{\langle k\rangle}{2}\prod_{\nu\alpha}\left[\int_{\odot} d\vec{S}_{\nu}^{\alpha}\int_{\odot} d\vec{S}_{\nu}^{\prime\alpha}\right]\prod_{\alpha}\left[\int d\hat{\lambda}_{\alpha}d\hat{\lambda}_{\alpha}^{\prime}\right] P(\vec{S},\underline{\hat{\lambda}})P(\mathbf{S}^{\prime},\underline{\hat{\lambda}}^{\prime})\prod_{\nu\alpha}\left(1+e^{-\frac{i\hat{\lambda}_{\alpha}+i\hat{\lambda}_{\alpha}^{\prime}}{2}\vec{S}_{\nu}^{\alpha}\cdot\vec{S}_{\nu}^{\prime\alpha}}\right).
\end{aligned}
\tag{S17}
$$

The free energy $\mathcal{F}$ of the system is given by $\mathcal{F} = \lim_{m\to 0}\frac{1}{\beta m N}\log\overline{\mathcal{Z}^m} \approx \lim_{m\to 0}\frac{1}{m}\Psi^*$ where Ψ^* is the extremization of the exponent Ψ with the saddle point of $P(\mathbf{S},\underline{\hat{\lambda}})$, $\hat{P}(\mathbf{S},\underline{\hat{\lambda}})$, $Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}})$ and $\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}})$. We can now average $\mathcal{F}$ with respect to the degree distribution $\rho(k)$ such that $\sum_{k=1}^{\infty}\rho(k)\mathcal{F}\propto\sum_{k=1}^{\infty}\rho(k)\Psi^*$, where Ψ is the same as in Eq. (S17) except for the term in the 3rd line that becomes

$$
\sum_{k=1}^{\infty}\rho(k)\log\left\{\prod_{\alpha}\left[\int\frac{d\lambda_{\alpha}d\hat{\lambda}_{\alpha}}{2\pi}e^{iM\lambda_{\alpha}\hat{\lambda}_{\alpha}-\beta M\phi(\lambda_{\alpha})}\right]\prod_{\nu a}\left[\int_{\odot} d\vec{S}_{\nu}^{\alpha}\right]\frac{\left[\hat{P}(\mathbf{S},\underline{\hat{\lambda}})\right]^{k}}{\langle k\rangle^{k}}e^{\sum_{\nu}\hat{Q}_{\nu}(\mathbf{S}_{\nu},\underline{\hat{\lambda}})}\right\}
\tag{S18}
$$

The saddle point of P, $\hat{P}$, Q and $\hat{Q}$ are given by the solution of

$$P(\mathbf{S},\underline{\hat{\lambda}})=\sum_{k}\rho(k)\frac{\prod_\alpha\left[\int\frac{d\lambda_\alpha}{2\pi}e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]k[\hat{P}(\mathbf{S},\underline{\hat{\lambda}})]^{k-1}e^{\sum_\nu\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}})}}{\prod_\alpha\left[\int\frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi}e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_{\nu\alpha}\left[\int_\odot d\vec{S}^\alpha_\nu\right][\hat{P}(\mathbf{S},\underline{\hat{\lambda}})]^k e^{\sum_\nu\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}})}} \tag{S19}$$

$$\hat{P}(\mathbf{S},\underline{\hat{\lambda}})=\langle k\rangle\prod_{\nu\alpha}\left[\int_\odot d\vec{S}'^\alpha_\nu\right]\prod_\alpha\left[\int d\hat{\lambda}'_\alpha\right]P(\mathbf{S}',\underline{\hat{\lambda}}')\prod_{\nu\alpha}\left(1+e^{-\frac{i\hat{\lambda}_\alpha+i\hat{\lambda}'_\alpha}{2}}\vec{S}^\alpha_\nu\cdot\vec{S}'^\alpha_\nu\right) \tag{S20}$$

$$Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}})=\sum_k\rho(k)\frac{\prod_\alpha\left[\int\frac{d\lambda_\alpha}{2\pi}e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_{\alpha,\mu\neq\nu}\left[\int_\odot d\vec{S}^\alpha_\mu\right][\hat{P}(\mathbf{S},\underline{\hat{\lambda}})]^k e^{\sum_\mu\hat{Q}_\mu(\vec{S}_{\mu,a},\underline{\hat{\lambda}})}}{\prod_\alpha\left[\int\frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi}e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_{\mu\alpha}\left[\int_\odot d\vec{S}^\alpha_\mu\right][\hat{P}(\mathbf{S},\underline{\hat{\lambda}})]^k e^{\sum_\mu\hat{Q}_\mu(\vec{S}_{\mu,a},\underline{\hat{\lambda}})}} \tag{S21}$$

$$\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}})=\frac{2}{N\mathcal{N}_Q}\prod_\alpha\left[\int_\odot d\vec{S}^\alpha_\nu\int d\hat{\lambda}'_\alpha\right]Q(\vec{S}'_{\nu,a},\underline{\hat{\lambda}}')\prod_\alpha\left(e^{-\frac{i\hat{\lambda}_\alpha+i\hat{\lambda}'_\alpha}{2}}S^\alpha_{\nu,a}S'^\alpha_{\nu,a}\right) \tag{S22}$$

with the normalization factor

$$\mathcal{N}_Q=\prod_\alpha\left[\int_\odot d\vec{S}'^\alpha_\nu\int_\odot d\vec{S}''^\alpha_\nu\int d\hat{\lambda}'_\alpha\int d\hat{\lambda}''_\alpha\right]Q(\vec{S}'_{\nu,a},\underline{\hat{\lambda}}')Q_\nu(\vec{S}''_{\nu,a},\underline{\hat{\lambda}}'')\prod_\alpha\left(e^{-\frac{i\hat{\lambda}'_\alpha+i\hat{\lambda}''_\alpha}{2}}S'^\alpha_{\nu,a}S''^\alpha_{\nu,a}\right)$$

One can see that the substitution of $\hat{P}$ and $\hat{Q}$ (Eqs. (S20) and (S22)) into P and Q (Eqs. (S19) and (S21)) leads to a pair of coupled recursive equations in terms of P and Q only. To solve these equations, one needs to find a self-consistent closed form for the equations of P and Q. We will thus apply a symmetry ansatz for P and Q and show that they satisfy Eqs. (S19) and (S21) self-consistently.

Before introducing the specific ansatz chosen, let us summarize the physical interpretation of the existing calculation and provide a sketch of the subsequent procedure following the choice of symmetry ansatz. Equation (S16) and the corresponding set of saddle point order parameters P, $\hat{P}$, Q and $\hat{Q}$ decouple the interaction between nodes on a sparse graph to provide mean field macroscopic distributions. We will proceed to decouple the communications, which are fully connected in their own space (since each communication interacts with all other communications on each node). We will see that the variable $\hat{\lambda}_\alpha$, defined as the conjugate of the normalized traffic $\lambda_\alpha = I_\alpha/M$ on a node, plays a crucial role in decoupling the communications in the limit $M\to\infty$, resembling the role of the magnetization m_α in the conventional fully connected spin glass replica calculation as $N\to\infty$ [S1, S2, S7]. Since we are also interested in the regime of finite M, we will obtain approximate equations that are applicable for any value of M. In summary, to solve the system of equations, we apply the first set of order parameter to decouple the *sparsely* connected nodes, and then a second set of (order parameter-like) variables to decouple the *fully* connected communications.

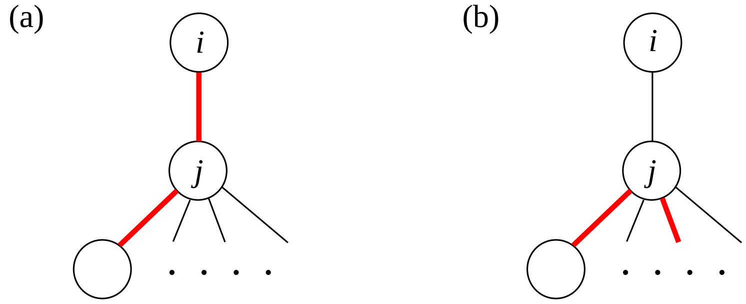

Fig. S1: Communication routes are indicated by thick red lines. (a) Communication passes through node j and its ancestor i, and (b) communication route passes node j but not its ancestor i.

1.3 The functional order parameter ansatze at finite temperature

1.3.1 The expression for $\hat{P}(\mathbf{S},\, \underline{\hat{\lambda}})$

We apply the following ansatz for $\hat{P}(\mathbf{S}, \underline{\hat{\lambda}})$

$$P(\mathbf{S}, \underline{\hat{\lambda}}) = \prod_{\alpha} \left[\int \frac{d\lambda_\alpha}{2\pi} e^{iM\lambda_\alpha \hat{\lambda}_\alpha - \beta M \phi(\lambda_\alpha)} \right] \int d\vec{h}\ w(\vec{h}) \prod_{\nu\alpha} \left[h_0^\nu + h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}} S_{\nu,a}^\alpha + h_2^\nu e^{-i\hat{\lambda}_\alpha} (S_{\nu,a}^\alpha)^2 \right], \tag{S23}$$

which is a power series of $S_{\nu,a}^\alpha$ up to the second order, since all higher order moments vanish due to the 0-vector method of polymer science [S3]. For readers familiar to the techniques of disordered system, the above ansatz relates to the cavity approach [S1, S2], where $h_0^\nu, h_1^\nu, h_2^\nu$ resemble the *cavity fields* where communication ν does not pass through the node (h_0^ν), passes through the node via the cavity to an ancestor as in Fig. S1(a) (h_1^ν) or passes the node without going through the cavity as in Fig. S1(b) (h_2^ν), respectively. Due to this connection with the cavity approach, this ansatz can be converted into a successful algorithm for optimizing real instances.

Inserting ansatz (S23) into Eq. (S20) gives $\hat{P}(\mathbf{S}, \underline{\hat{\lambda}})$ by keeping only terms in the order

$(S'^{\alpha}_{\nu,a})^0$ or $(S'^{\alpha}_{\nu,a})^2$, due to the angular integration $\int_{\odot} d\vec{S}'^{\alpha}_{\nu}$ of the 0-vector method, yielding

$$\begin{aligned}
\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) &= \langle k\rangle \int d\vec{h}\; w(\vec{h}) \prod_{\nu\alpha}\left[\int_{\odot} d\vec{S}'^{\alpha}_{\nu}\right] \prod_{\alpha}\left[\int \frac{d\lambda'_{\alpha} d\hat{\lambda}'_{\alpha}}{2\pi} e^{iM\lambda'_{\alpha}\hat{\lambda}'_{\alpha} - \beta M\phi\left(\lambda'_{\alpha}\right)}\right] \\
&\quad \times \prod_{\nu\alpha}\left[h^{\nu}_0 + h^{\nu}_2 e^{-i\hat{\lambda}'_{\alpha}}(S'^{\alpha}_{\nu,a})^2 + h^{\nu}_1 e^{-\frac{i\hat{\lambda}_{\alpha}}{2} - i\hat{\lambda}'_{\alpha}} S^{\alpha}_{\nu,a}(S'^{\alpha}_{\nu,a})^2\right] \\
&= \langle k\rangle (C_n)^{mM} \int d\vec{h}\; w(\vec{h}) \prod_{\alpha}\left[\int \frac{d\lambda'_{\alpha} d\hat{\lambda}'_{\alpha}}{2\pi} e^{iM\lambda'_{\alpha}\hat{\lambda}'_{\alpha} - \beta M\phi\left(\lambda'_{\alpha}\right)}\right] \\
&\quad \times \prod_{\nu\alpha}\left[h^{\nu}_0 + h^{\nu}_2 e^{-i\hat{\lambda}'_{\alpha}} + h^{\nu}_1 e^{-\frac{i\hat{\lambda}_{\alpha}}{2} - i\hat{\lambda}'_{\alpha}} S^{\alpha}_{\nu,a}\right] \qquad \because \int_{\odot} d\vec{S}'^{\alpha}_{\nu}(S'^{\alpha}_{\nu,a})^2 = \int_{\odot} d\vec{S}'^{\alpha}_{\nu} = C_n \\
&= \langle k\rangle \int d\vec{h}\; w(\vec{h}) \prod_{\alpha}\left[\int \frac{d\lambda'_{\alpha} d\hat{\lambda}'_{\alpha}}{2\pi} e^{iM\lambda'_{\alpha}\hat{\lambda}'_{\alpha} - \beta M\phi\left(\lambda'_{\alpha}\right)}\right] \\
&\quad \times \prod_{\nu\alpha}\left[h^{\nu}_0 + h^{\nu}_2 e^{-i\hat{\lambda}'_{\alpha}} + h^{\nu}_1 e^{-\frac{i\hat{\lambda}_{\alpha}}{2} - i\hat{\lambda}'_{\alpha}} S^{\alpha}_{\nu,a}\right] \qquad \because \text{replica } m \to 0
\end{aligned} \tag{S24}$$

where the integration on λ'_{α} and $\hat{\lambda}'_{\alpha}$ can be computed by the saddle point method in the limit $M \to \infty$. In other words, we compute the saddle point with respect to λ'^{*}_{α} and $\hat{\lambda}'^{*}_{\alpha}$ (related by $i\hat{\lambda}'^{*}_{\alpha} = \beta\phi'(\lambda'^{*}_{\alpha})$ as explained by Section S1.3.5) and use them to evaluate the integral over λ'_{α} and $\hat{\lambda}'_{\alpha}$. For finite values of M, we found in Section S1.3.5 that if we estimate λ'^{*}_{α} individually per communication, i.e. obtain an estimated solution for $\lambda'^{*}_{\nu\alpha}$, then the analytic solution obtained for Eq. (S24) provides physically meaningful results in a broad range of M values. In the subsequent derivation, we will denote the saddle point value of $\lambda'^{*}_{\nu\alpha}$ by a function $F_{\nu}(\vec{h}, \mathbf{S}, \underline{\hat{\lambda}})$ to emphasize its dependence on the variables $\vec{h}, \mathbf{S}$ and $\underline{\hat{\lambda}}$. The explicit form of $F_{\nu}(\vec{h}, \mathbf{S}, \underline{\hat{\lambda}})$ is derived in Section S1.3.5.

By using the saddle point value of $\lambda'^{*}_{\nu\alpha}$ in Eq. (S24), and assuming replica symmetry (RS) such that $\lambda'^{*}_{\nu\alpha} = \lambda'^{*}_{\nu}$ for all α, we obtain

$$\begin{aligned}
\hat{P}(\mathbf{S},\underline{\hat{\lambda}}) =& \langle k\rangle \int d\vec{h}\; w(\vec{h}) \prod_{\nu}\left[\int d\lambda'^{*}_{\nu} \delta[\lambda'^{*}_{\nu} - F_{\nu}(\vec{h}, \mathbf{S}, \underline{\hat{\lambda}})]\right] \left(\frac{e^{\beta\sum_{\nu}\left[\lambda'^{*}_{\nu}\phi'(\lambda'^{*}_{\nu}) - \phi(\lambda'^{*}_{\nu})\right]}}{2\pi}\right)^{m} \\
& \times \prod_{\nu\alpha}\left[h^{\nu}_0 + h^{\nu}_2 e^{-\beta\phi'(\lambda'^{*}_{\nu})} + h^{\nu}_1 e^{-\frac{i\hat{\lambda}_{\alpha}}{2} - \beta\phi'(\lambda'^{*}_{\nu})} S^{\alpha}_{\nu,a}\right] \\
=& \langle k\rangle \int d\vec{h}\; w(\vec{h}) \prod_{\nu}\left[\int d\lambda'^{*}_{\nu} \delta[\lambda'^{*}_{\nu} - F_{\nu}(\vec{h}, \mathbf{S}, \underline{\hat{\lambda}})]\right] \\
& \times \prod_{\nu\alpha}\left[h^{\nu}_0 + h^{\nu}_2 e^{-\beta\phi'(\lambda'^{*}_{\nu})} + h^{\nu}_1 e^{-\frac{i\hat{\lambda}_{\alpha}}{2} - \beta\phi'(\lambda'^{*}_{\nu})} S^{\alpha}_{\nu,a}\right] \quad \because \text{replica } m \to 0
\end{aligned} \tag{S25}$$

Having the expression for $\hat{P}(\mathbf{S}, \underline{\hat{\lambda}})$, the product $[\hat{P}(Z, \vec{S}, \underline{\hat{\lambda}})]^{k-1}$ in the numerator of

Eq. (S19) (and hence similarly for $[\hat{P}(Z,\vec{S},\underline{\hat{\lambda}})]^k$ in the denominator) can be expressed as

$$
\begin{aligned}
[\hat{P}(\mathbf{S},\underline{\hat{\lambda}})]^{k-1} &= \prod_{l=1}^{k-1} \hat{P}_l(\vec{S},\underline{\hat{\lambda}}) \\
&= \langle k\rangle^{k-1} \prod_{l=1}^{k-1} \left[\int d\vec{h}_l \; w(\vec{h}_l) \prod_{\nu} \int d\lambda^*_{\nu l} \delta[\lambda^*_{\nu l} - F_\nu(\vec{h}_l,\mathbf{S},\underline{\hat{\lambda}})]\right] \\
&\times \prod_{\nu\alpha} \prod_{l=1}^{k} \left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda^*_{\nu l})} + h_1^{\nu l} e^{-\frac{i\hat{\lambda}_\alpha}{2} - \beta\phi'(\lambda^*_{\nu l})} S^\alpha_{\nu,a}\right].
\end{aligned}
\tag{S26}
$$

We then expand the final product up to the second order $(S^\alpha_{\nu,a})^2$ as the integration of all higher order of $S^\alpha_{\nu,a}$ vanishes when we later complete the integration over $S^\alpha_{\nu,a}$. This leads to

$$
\begin{aligned}
[\hat{P}(\mathbf{S},\underline{\hat{\lambda}})]^{k-1} =& \langle k\rangle^{k-1} \prod_{l=1}^{k-1} \left[\int d\vec{h}_l \; w(\vec{h}_l) \prod_{\nu} \int d\lambda^*_{\nu l} \delta[\lambda^*_{\nu l} - F_\nu(\vec{h}_l,\mathbf{S},\underline{\hat{\lambda}})]\right] \prod_{\nu\alpha} \left[\left(\prod_{l=1}^{q} (h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda^*_{\nu l})})\right)\right. \\
&+ \left(\sum_{l=1}^{q} h_1^{\nu l} e^{-\beta\phi'(\lambda^*_{\nu l})} \prod_{j\neq l} (h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda^*_{\nu j})})\right) e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} \\
&+ \left.\left(\sum_{(lr)} h_1^{\nu l} h_1^{\nu r} e^{-\beta\phi'(\lambda^*_{\nu l}) - \beta\phi'(\lambda^*_{\nu r})} \prod_{j\neq l,w} (h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda^*_{\nu j})})\right) e^{-i\hat{\lambda}_\alpha} (S^\alpha_{\nu,a})^2\right] \\
=& \langle k\rangle^{k-1} \prod_{l=1}^{k-1} \left[\int d\vec{h}_l \; w(\vec{h}_l)\right] \prod_{\nu\alpha} \left[g^\nu_{0,k-1} + g^\nu_{1,k-1} e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} + g^\nu_{2,k-1} e^{-i\hat{\lambda}_\alpha} (S^\alpha_{\nu,a})^2\right]
\end{aligned}
\tag{S27}
$$

such that

$$
g^\nu_{0,k-1} = \prod_{l=1}^{k-1} \left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda^*_{\nu l})}\right] \tag{S28}
$$

$$
g^\nu_{1,k-1} = \sum_{l=1}^{k-1} h_1^{\nu l} e^{-\beta\phi'(\lambda^*_{\nu l})} \prod_{j\neq l} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda^*_{\nu j})}\right] \tag{S29}
$$

$$
g^\nu_{2,k-1} = \sum_{(lr)} h_1^{\nu l} h_1^{\nu r} e^{-\beta\phi'(\lambda^*_{\nu l}) - \beta\phi'(\lambda^*_{\nu r})} \prod_{j\neq l,r} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda^*_{\nu j})})\right] \tag{S30}
$$

Expression (S27) already shares a similar form with the ansatz (S23). Next we will show that after applying a similar ansatz to $\hat{Q}(\vec{S},\underline{\hat{\lambda}})$, we will get a self consistent equation in terms of $w(\vec{h}_l)$ and the fields h_0^ν, h_1^ν and h_2^ν only.

1.3.2 The expression for $\hat{P}(\mathbf{S},\ \underline{\hat{\lambda}})$

The ansatz for $Q(\vec{S},\underline{\hat{\lambda}})$ takes a similar form to (S23)

$$Q(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) = \prod_{\alpha}\left[\int \frac{d\lambda_\alpha}{2\pi} e^{iM\lambda_\alpha\hat{\lambda}_\alpha - \beta M\phi(\lambda_\alpha)}\right] \prod_{\alpha,\mu\neq\nu}\left[h_0^\mu + h_2^\mu e^{-i\hat{\lambda}_\alpha}\right]$$
$$\times \int d\vec{h}\ w(\vec{h}) \prod_{\alpha}\left[h_0^\nu + h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}} S_{\nu,a}^\alpha + h_2^\nu e^{-i\hat{\lambda}_\alpha}(S_{\nu,a}^\alpha)^2\right] \tag{S31}$$

except that it is applied individually for each communication ν and the dependence on $S_{\nu,a}^\alpha$ is accordingly. The above ansatz is plugged into Eq. (S21) to give $\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}})$ as

$$\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) = \frac{2}{N\mathcal{D}_{\hat{Q}}} \prod_{\alpha}\left[\int \frac{d\lambda'_\alpha d\hat{\lambda}'_\alpha}{2\pi} e^{iM\lambda'_\alpha\hat{\lambda}'_\alpha - \beta M\phi\left(\lambda'_\alpha\right)}\right] \prod_{\alpha,\mu\neq\nu}\left[h_0^\mu + h_2^\mu e^{-i\hat{\lambda}_\alpha}\right]$$
$$\times \int d\vec{h}\ w(\vec{h}) \prod_{\alpha}\left[h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2} - i\hat{\lambda}'_\alpha} S_{\nu,a}^\alpha\right] \tag{S32}$$

where $\mathcal{D}_{\hat{Q}}$ is a normalization constant. As in the case for $\hat{P}$, we evaluate the integration of λ'_α and $\hat{\lambda}'_\alpha$ by finding saddle points when $M \to \infty$

$$\hat{Q}(\underline{S}_{\nu,a},\underline{\hat{\lambda}}) = \frac{2}{N\mathcal{D}_{\hat{Q}}} \int d\vec{h}\ w(\vec{h}) \prod_{\alpha}\left[\int \frac{d\lambda'_\alpha d\hat{\lambda}'_\alpha}{2\pi}\right] \prod_{\alpha}\left[h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}} S_{\nu,a}^\alpha\right] e^{M\Omega_\nu(\underline{\hat{\lambda}})} \tag{S33}$$

with

$$\Omega_\nu(\underline{\hat{\lambda}}) = -\frac{i\hat{\lambda}'_\alpha}{M} + \sum_\alpha i\lambda'_\alpha\hat{\lambda}'_\alpha - \sum_\alpha \beta\phi\left(\lambda'_\alpha\right) + \frac{1}{M}\left[\sum_\alpha\sum_{\mu\neq\nu}\log\left(h_0^\mu + h_2^\mu e^{-i\hat{\lambda}'_\alpha}\right)\right]. \tag{S34}$$

This allows one to obtain the saddle point of λ'_α and $\hat{\lambda}'_\alpha$ by $\partial\Xi/\partial\lambda'_\alpha = 0$ and $\partial\Xi/\partial\hat{\lambda}'_\alpha = 0$ which give

$$i\hat{\lambda}'_\alpha = \beta\phi'\left(\lambda'_\alpha\right) \tag{S35}$$

$$i\lambda'_\alpha = \frac{i}{M}\left[1 + \sum_{\mu\neq\nu}\frac{h_2^\mu e^{-i\hat{\lambda}'_\alpha}}{h_0^\mu + h_2^\mu e^{-i\hat{\lambda}'_\alpha}}\right] \tag{S36}$$

implying

$$\lambda'_\alpha = \frac{1}{M}\left[1 + \sum_{\mu\neq\nu}\frac{h_2^\mu e^{-\beta\phi'(\lambda'_\alpha)}}{h_0^\mu + h_2^\mu e^{-\beta\phi'(\lambda'_\alpha)}}\right], \tag{S37}$$

which is an equation in λ'_α. Unlike the saddle point $\lambda_\nu^* = F_\nu(\vec{h},\mathbf{S},\underline{\hat{\lambda}})$ in the expression of $\hat{P}(\mathbf{S},\underline{\hat{\lambda}})$ (in Eq. (S25)), the above equation (S37) is independent of $\underline{S}_{\nu,a}$ and $\underline{\hat{\lambda}}$ which make the solution λ'^*_α dependent only on h_0^μ, h_1^μ and h_2^μ. By assuming replica symmetry such that $\lambda'^*_\alpha = \lambda'^*$ for all α, Eq. (S32) becomes

$$\begin{aligned}
\hat{Q}(\underline{S}_{\nu,a}, \underline{\hat{\lambda}}) =& \frac{2}{N\mathcal{D}_{\hat{Q}}} \int d\vec{h}\; w(\vec{h}) \left(\frac{e^{\beta M\left[\lambda'^*\phi'(\lambda'^*)-\phi(\lambda'^*)\right]}}{2\pi} \right)^m \prod_{\alpha,\mu\neq\nu} \left[h_0^\mu + h_2^\mu e^{-\beta\phi'(\lambda'^*)} \right] \\
& \prod_\alpha \left[h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2} - \beta\phi'(\lambda'^*)} S^\alpha_{\nu,a} \right] \\
=& \frac{2}{N\mathcal{D}_{\hat{Q}}} \int d\vec{h}\; w(\vec{h}) \left[\left(\frac{e^{\beta M\left[\lambda'^*\phi'(\lambda'^*)-\phi(\lambda'^*)\right]}}{2\pi} \right) \prod_{\mu\neq\nu} \left(h_0^\mu + h_2^\mu e^{-2\beta\lambda} \right) h_1^\nu e^{-2\beta\lambda} \right]^m \\
& \prod_\alpha \left[e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} \right] \\
=& \frac{2}{N\mathcal{D}_{\hat{Q}}} \prod_\alpha \left[e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} \right] \qquad \because \text{replica } m \to 0
\end{aligned} \tag{S38}$$

given that $\int d\vec{h}\; w(\vec{h}) = 1$. We now compute the denominator $\mathcal{D}_{\hat{Q}}$, which can be done in the same manner yielding

$$\begin{aligned}
\mathcal{D}_{\hat{Q}} =& \left\{ \int d\vec{h}\; w(\vec{h}) \left[\left(\frac{e^{\beta M\left[\lambda'^*\phi'(\lambda'^*)-\phi(\lambda'^*)\right]}}{2\pi} \right) \prod_{\mu\neq\nu} \left(h_0^\mu + h_2^\mu e^{-2\beta\lambda} \right) h_1^\nu e^{-2\beta\lambda} \right]^m \right\}^2 \\
=& 1 \qquad \because \text{replica } m \to 0
\end{aligned} \tag{S39}$$

Thus

$$\hat{Q}(\underline{S}_{\nu,a}, \underline{\hat{\lambda}}) = \frac{2}{N} \prod_\alpha \left[e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} \right] \tag{S40}$$

Finally we expand the exponential factor in $\prod_\nu e^{\hat{Q}(\vec{S}, \underline{\hat{\lambda}})}$ to first order of $S^\alpha_{\nu,a}$. The second order of $S^\alpha_{\nu,a}$ has a factor $\frac{4}{N^2}$ which corresponds to the same node being both source and destination of a communication and will be neglected. This gives

$$\begin{aligned}
\prod_\nu e^{\hat{Q}(\vec{S}, \underline{\hat{\lambda}})} =& \prod_\nu \left[1 + \frac{2}{N} \prod_\alpha \left(e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} \right) \right] \\
=& \prod_\nu \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N} \delta_{\Lambda_\nu,1} \right] \prod_{\nu\alpha} \left[1 - \Lambda_\nu + \Lambda_\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}} S^\alpha_{\nu,a} \right]
\end{aligned} \tag{S41}$$

1.3.3 Recursive equation for $w(\vec{h})$

Finally we substitute the two expressions (S27) and (S41) into Eq. (S19) to obtain a self consistent equation for P

$$
\begin{aligned}
P(\mathbf{S},\hat{\underline{\lambda}}) \propto & \sum_{k=1}^{\infty}\frac{k\rho(k)}{\langle k\rangle}\frac{1}{\mathcal{D}_{P,k}}\prod_{l=1}^{k-1}\left[\int d\vec{h}_l\; w(\vec{h}_l)\prod_{\nu}\int d\lambda^*_{\nu l}\delta[\lambda^*_{\nu l}-F_\nu(\vec{h}_l,\mathbf{S},\hat{\underline{\lambda}})]\right]\prod_{\nu}\sum_{\Lambda_\nu=0}^{1}\left[\delta_{\Lambda_\nu,0}+\frac{2}{N}\delta_{\Lambda_\nu,1}\right] \\
\times & \prod_{\alpha}\left[\int\frac{d\lambda_\alpha}{2\pi}e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_{\nu\alpha}\Big\{\left[(1-\Lambda_\nu)g^{\nu}_{0,k-1}\right]+\left[(1-\Lambda_\nu)g^{\nu}_{1,k-1}+\Lambda_\nu g^{\nu}_{0,k-1}\right]e^{-\frac{i\hat{\lambda}_\alpha}{2}}S^{\alpha}_{\nu,a} \\
& +\left[(1-\Lambda_\nu)g^{\nu}_{2,k-1}+\Lambda_\nu g^{\nu}_{1,k-1}\right]e^{-i\hat{\lambda}_\alpha}(S^{\alpha}_{\nu,a})^2\Big\} \qquad \text{(S42)}
\end{aligned}
$$

The above expression does not yield a self-consistent ansatz since $\lambda^*_{\nu l}$ still depends on $\mathbf{S}$ and $\hat{\underline{\lambda}}$. We thus approximate the above expression by

$$
\begin{aligned}
P(\mathbf{S},\hat{\underline{\lambda}}) \approx & \sum_{k=1}^{\infty}\frac{k\rho(k)}{\langle k\rangle}\frac{1}{\mathcal{D}_{P,k}}\prod_{l=1}^{k-1}\left[\int d\vec{h}_l\; w(\vec{h}_l)\int_{\odot} d\mathbf{S}'_l\int d\hat{\underline{\lambda}}'_l P(\mathbf{S}'_l,\hat{\underline{\lambda}}'_l)\prod_{\nu}\int d\lambda^*_{\nu l}\delta[\lambda^*_{\nu l}-F_\nu(\vec{h}_l,\mathbf{S}'_l,\hat{\underline{\lambda}}'_l)]\right] \\
\times & \prod_{\nu}\sum_{\Lambda_\nu=0}^{1}\left[\delta_{\Lambda_\nu,0}+\frac{2}{N}\delta_{\Lambda_\nu,1}\right] \\
\times & \prod_{\alpha}\left[\int\frac{d\lambda_\alpha}{2\pi}e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_{\nu\alpha}\Big\{\left[(1-\Lambda_\nu)g^{\nu}_{0,k-1}\right]+\left[(1-\Lambda_\nu)g^{\nu}_{1,k-1}+\Lambda_\nu g^{\nu}_{0,k-1}\right]e^{-\frac{i\hat{\lambda}_\alpha}{2}}S^{\alpha}_{\nu,a} \\
& +\left[(1-\Lambda_\nu)g^{\nu}_{2,k-1}+\Lambda_\nu g^{\nu}_{1,k-1}\right]e^{-i\hat{\lambda}_\alpha}(S^{\alpha}_{\nu,a})^2\Big\} \qquad \text{(S43)}
\end{aligned}
$$

where the normalization factor $\mathcal{D}_{P,k}=1$ as the number of replica $m\to 0$. This approximation of $F_\nu(\vec{h}_l,\mathbf{S},\hat{\underline{\lambda}})$ by $F_\nu(\vec{h}_l,\mathbf{S}'_l,\hat{\underline{\lambda}}'_l)$, with $\mathbf{S}'_l$ and $\hat{\underline{\lambda}}'_l$ independently drawn from the *same distribution* $P(\mathbf{S},\hat{\underline{\lambda}})$, is particularly justified when $M\to\infty$ as $\lambda^*_{\nu l}$ is obtained from a saddle point, where the distribution approaches a delta function and is primarily dependent on the *form* of the distribution $P(\mathbf{S},\hat{\underline{\lambda}})$. We remark that the above approximation is the main approximation utilized for the derivation. One can now identify a recursive relation for $w(\vec{h})$ by comparing the above expression with Eq. (S23), such that

$$
\begin{aligned}
w(\vec{h}) = & \sum_{k=1}^{\infty}\frac{k\rho(k)}{\langle k\rangle}\prod_{l=1}^{k-1}\left[\int d\vec{h}_l\; w(\vec{h}_l)\int_{\odot} d\mathbf{S}'_l\int d\hat{\underline{\lambda}}'_l P(\mathbf{S}'_l,\hat{\underline{\lambda}}'_l)\prod_{\nu}\int d\lambda^*_{\nu l}\delta[\lambda^*_{\nu l}-F(\vec{h}_l,\mathbf{S}'_l,\hat{\underline{\lambda}}'_l)]\right] \\
& \times\prod_{\nu}\sum_{\Lambda_\nu=0}^{1}\left[\delta_{\Lambda_\nu,0}+\frac{2}{N}\delta_{\Lambda_\nu,1}\right]\prod_{\nu}\left\{\delta\left[h^{\nu}_0-(1-\Lambda_\nu)g^{\nu}_{0,k-1}\right]\right. \\
& \left.\times\delta\left[h^{\nu}_1-\left[(1-\Lambda_\nu)g^{\nu}_{1,k-1}+\Lambda_\nu g^{\nu}_{0,k-1}\right]\right]\delta\left[h^{\nu}_2-\left[(1-\Lambda_\nu)g^{\nu}_{2,k-1}+\Lambda_\nu g^{\nu}_{1,k-1}\right]\right]\right\} \qquad \text{(S44)}
\end{aligned}
$$

where the g^{ν}'s are given in terms of $h^{\nu l}$ by Eqs. (S28) to (S30). In other words, combining the above expression and Eqs. (S28) to (S30), we obtain a recursion relation for the fields

h as

$$h_0^\nu = (1-\Lambda_\nu)\prod_{l=1}^{k-1}\left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)}\right] \tag{S45}$$

$$h_1^\nu = (1-\Lambda_\nu)\sum_{l=1}^{k-1} h_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \prod_{j\neq l}\left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)}\right] + \Lambda_\nu \prod_{l=1}^{k-1}\left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)}\right] \tag{S46}$$

$$\begin{aligned} h_2^\nu =& (1-\Lambda_\nu)\sum_{(lr)} h_1^{\nu l} h_1^{\nu r} e^{-\beta\phi'(\lambda_{\nu l}^*)-\beta\phi'(\lambda_{\nu r}^*)} \prod_{j\neq l,r}\left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)}\right] \\ &+ \Lambda_\nu \sum_{l=1}^{k-1} h_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \prod_{j\neq l}\left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)}\right] \end{aligned} \tag{S47}$$

The general form of this equation resembles the replica symmetry cavity equations obtained elsewhere. After deriving these equations, we can now focus on the variable h's instead of the more detailed polymer variables $S_{\nu,a}^\alpha$. A further simplification is shown in 1.3.4. By using the same procedure as above, one can show that the substitution of (S27) and (S41) into Eq. (S21) lead to a self-consistent form of the ansatz Eq. (S31).

1.3.4 Further simplification of the equations

Further simplification of equation (S44) is based on simplifying the notation in Eqs. (S28) to (S30) using

$$z_l^\nu = \frac{h_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)}}{h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)}} \tag{S48}$$

and similarly

$$z^\nu = \frac{h_1^{\nu} e^{-\beta\phi'(\lambda_{\nu}^*)}}{h_0^{\nu} + h_2^{\nu} e^{-\beta\phi'(\lambda_{\nu}^*)}} \tag{S49}$$

such that Eqs. (S28) to (S30) become

$$\frac{g_{1,k-1}^\nu}{g_{0,k-1}^\nu} = \sum_{l=1}^{k-1} z_l^\nu \tag{S50}$$

$$\frac{g_{2,k-1}^\nu}{g_{0,k-1}^\nu} = \sum_{(lr)} z_l^\nu z_r^\nu . \tag{S51}$$

Using Eq. (S44), we now rewrite Eq. (S49) as

$$
\begin{aligned}
z^{\nu} &= \frac{h_1^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}{h_0^{\nu} + h_2^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}} \\
&= (1-\Lambda_\nu)\frac{g_{1,k-1}^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}{g_{0,k-1}^{\nu} + g_{2,k-1}^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}} + \Lambda_\nu \frac{g_{0,k-1}^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}{g_{1,k-1}^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}} \\
&= (1-\Lambda_\nu)\frac{e^{-\beta\phi'(\lambda_\nu^*)}\sum_{l=1}^{k-1} z_l^{\nu}}{1 + e^{-\beta\phi'(\lambda_\nu^*)}\sum_{(lr)}^{k-1} z_l^{\nu} z_r^{\nu}} + \Lambda_\nu \frac{1}{\sum_{l=1}^{k-1} z_l^{\nu}},
\end{aligned} \tag{S52}
$$

where $\lambda_\nu^* = f_\nu^{-1}(0)$, and the function $f_\nu(x)$ depends on the descendent field $\vec{z}$ in the manner explained in the next section. With this notation the equation can be simplified based on a distribution $w(\vec{z})$ which characterizes each communication ν by a single value z^ν, instead of h_0^ν, h_1^ν and h_2^ν as is the case in Eq. (S44). In other words, we preform the change of variables by defining the distribution $w(\vec{z})$ as

$$
w(\vec{z}) = \int d\vec{h}\; w(\vec{h}) \int_{\odot} d\mathbf{S}' \int d\hat{\underline{\lambda}}' P(\mathbf{S}', \hat{\underline{\lambda}}') \prod_{\nu} \delta\left(z^{\nu} - \frac{h_1^{\nu} e^{-\beta\phi'(F_\nu(\vec{h},\mathbf{S}',\hat{\underline{\lambda}}'))}}{h_0^{\nu} + h_2^{\nu} e^{-\beta\phi'(F_\nu(\vec{h},\mathbf{S}',\hat{\underline{\lambda}}'))}} \right), \tag{S53}
$$

To perform the change of variables, we start from the recursion of $w(\vec{h})$ in Eq. (S44) and introduce the delta function $\int dz_l^{\nu} \delta\left(z_l^{\nu} - \frac{h_1^{\nu l} e^{-\beta\phi'(F_\nu(\vec{h},\mathbf{S}',\hat{\underline{\lambda}}'))}}{h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(F_\nu(\vec{h},\mathbf{S}',\hat{\underline{\lambda}}'))}} \right)$ for each l and ν on the right hand side of Eq. (S44). This effectively makes the variable g's depend solely on $\vec{z}_1, \ldots, \vec{z}_{k-1}$ (see Eqs. (50) and (51)), i.e.

$$
\begin{aligned}
w(\vec{h}) &= \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k \rangle} \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N}\delta_{\Lambda_\nu,1} \right] \\
&\times \prod_{l=1}^{k-1} \left[\int d\vec{h}_l w(\vec{h}_l) \int_{\odot} d\mathbf{S}_l' \int d\hat{\underline{\lambda}}'_l P(\mathbf{S}_l', \hat{\underline{\lambda}}'_l) \int d\vec{z}_l \prod_{\nu} \delta\left(z_l^{\nu} - \frac{h_1^{\nu l} e^{-\beta\phi'(F(\vec{h}_l,\mathbf{S}_l',\hat{\underline{\lambda}}'_l))}}{h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(F(\vec{h}_l,\mathbf{S}_l',\hat{\underline{\lambda}}'_l))}} \right) \right] \\
&\quad \times \prod_{\nu} \left\{ \delta\left[h_0^{\nu} - (1-\Lambda_\nu) g_{0,k-1}^{\nu}(\vec{z}_1,\ldots,\vec{z}_{k-1}) \right] \right. \\
&\qquad \times \delta\left[h_1^{\nu} - \left[(1-\Lambda_\nu) g_{1,k-1}^{\nu}(\vec{z}_1,\ldots,\vec{z}_{k-1}) + \Lambda_\nu g_{0,k-1}^{\nu}(\vec{z}_1,\ldots,\vec{z}_{k-1}) \right] \right] \\
&\qquad\quad \left. \times \delta\left[h_2^{\nu} - \left[(1-\Lambda_\nu) g_{2,k-1}^{\nu}(\vec{z}_1,\ldots,\vec{z}_{k-1}) + \Lambda_\nu g_{1,k-1}^{\nu}(\vec{z}_1,\ldots,\vec{z}_{k-1}) \right] \right] \right\}
\end{aligned} \tag{S54}
$$

The integration over $\vec{h}_l$, $\mathbf{S}'_l$ and $\hat{\underline{\lambda}}'_l$ can be replaced by $w(\vec{z}_l)$ as defined in Eq. (S53), leading to

$$\begin{aligned} w(\vec{h}) = \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k\rangle} \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N}\delta_{\Lambda_\nu,1}\right] \prod_{l=1}^{k-1} \left[\int d\vec{z}_l \; w(\vec{z}_l)\right] \\ \times \prod_{\nu} \left\{\delta\left[h_0^\nu - (1-\Lambda_\nu) g_{0,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1})\right]\right. \\ \times \delta\left[h_1^\nu - [(1-\Lambda_\nu) g_{1,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1}) + \Lambda_\nu g_{0,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1})]\right] \\ \left.\times\delta\left[h_2^\nu - [(1-\Lambda_\nu) g_{2,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1}) + \Lambda_\nu g_{1,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1})]\right]\right\} \end{aligned} \quad \text{(S55)}$$

To obtain a recursion relation for $w(\vec{z})$, we introduce $w(\vec{z})$ on the left hand side by following the definition of (S53), and multiply $P(\mathbf{S},\hat{\underline{\lambda}})\prod_\nu \delta\left(z^\nu - \frac{h_1^\nu e^{-\beta\phi'(F(\vec{h},\mathbf{S},\hat{\underline{\lambda}}))}}{h_0^\nu + h_2^\nu e^{-\beta\phi'(F(\vec{h},\mathbf{S},\hat{\underline{\lambda}}))}}\right)$ followed by the integration of the variables $\vec{h}$, $\mathbf{S}$ and $\hat{\underline{\lambda}}$ on both sides, which leads to

$$\begin{aligned} w(\vec{z}) = \int d\vec{h} \int_{\odot} d\vec{S} \int d\hat{\underline{\lambda}} P(\mathbf{S},\hat{\underline{\lambda}}) \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k\rangle} \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N}\delta_{\Lambda_\nu,1}\right] \prod_{l=1}^{k-1} \left[\int d\vec{z}_l w(\vec{z}_l)\right] \\ \times \prod_{\nu} \delta\left(z^\nu - \frac{h_1^\nu e^{-\beta\phi'(F_\nu(\vec{h},\mathbf{S},\hat{\underline{\lambda}}))}}{h_0^\nu + h_2^\nu e^{-\beta\phi'(F_\nu(\vec{h},\mathbf{S},\hat{\underline{\lambda}}))}}\right) \\ \times \prod_{\nu} \left\{\delta\left[h_0^\nu - (1-\Lambda_\nu) g_{0,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1})\right]\right. \\ \times \delta\left[h_1^\nu - [(1-\Lambda_\nu) g_{1,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1}) + \Lambda_\nu g_{0,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1})]\right] \\ \left.\times\delta\left[h_2^\nu - [(1-\Lambda_\nu) g_{2,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1}) + \Lambda_\nu g_{1,k-1}^\nu(\vec{z}_1,\ldots,\vec{z}_{k-1})]\right]\right\} \end{aligned} \quad \text{(S56)}$$

Finally, we integrate over the variables $\vec{h}$ by making use of the delta functions of the last three lines in Eq. (S56), i.e. replacing the variables h_0^ν, h_1^ν, h_2^ν by the g variables in the delta function of z^ν. This is equivalent to the simplifications obtained in Eq. (S52), leading to

$$\begin{aligned} w(\vec{z}) = & \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k\rangle} \prod_{l=1}^{k-1} \left[\int d\vec{z}_l \; w(\vec{z}_l)\right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N}\delta_{\Lambda_\nu,1}\right] \\ & \times \int_{\odot} d\vec{S} \int d\hat{\underline{\lambda}} P(\mathbf{S},\hat{\underline{\lambda}}) \int d\vec{\lambda}^* \prod_{\nu} \delta[\lambda_\nu^* - F_\nu(\vec{h}(\vec{z}_1,\ldots,\vec{z}_{k-1}),\mathbf{S},\hat{\underline{\lambda}})] \\ & \times \prod_{\nu} \delta\left\{z^\nu - (1-\Lambda_\nu) \frac{e^{-\beta\phi'(\lambda_\nu^*)}\sum_{l=1}^{k-1} z_l^\nu}{1 + e^{-\beta\phi'(\lambda_\nu^*)}\sum_{(lr)}^{k-1} z_l^\nu z_r^\nu} - \Lambda_\nu \frac{1}{\sum_{l=1}^{k-1} z_l^\nu}\right\} \end{aligned} \quad \text{(S57)}$$

where $\vec{\lambda}^*$ is a vector of λ_ν^* over the label ν. We further denote

$$\rho(\vec{\lambda}^*|\vec{h}(\vec{z}_1,\ldots,\vec{z}_{k-1})) = \int_{\odot} d\vec{S} \int d\hat{\underline{\lambda}} P(\mathbf{S},\hat{\underline{\lambda}}) \prod_{\nu} \delta[\lambda_\nu^* - F_\nu(\vec{h}(\vec{z}_1,\ldots,\vec{z}_{k-1}),\mathbf{S},\hat{\underline{\lambda}})] \quad \text{(S58)}$$

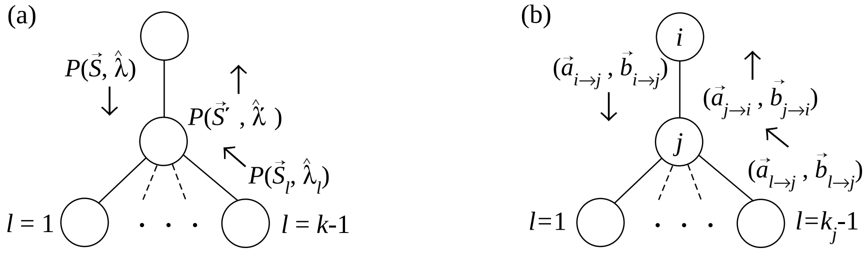

Fig. S2: (a) A forward field $P(\mathbf{S}', \hat{\underline{\lambda}}')$ from the central node to the ancestor, and a backward field $P(\mathbf{S}, \hat{\underline{\lambda}})$ from ancestor to the central node. (b) A corresponding picture of the messages in the derived algorithm.

such that the recursion of $w(\vec{z})$ is given by

$$w(\vec{z}) = \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k\rangle} \prod_{l=1}^{k-1} \left[\int d\vec{z}_l \; w(\vec{z}_l) \right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N} \delta_{\Lambda_\nu,1} \right] \int d\vec{\lambda}^* \rho(\vec{\lambda}^* | \vec{h}(\vec{z}_1, \ldots, \vec{z}_{k-1}))$$
$$\times \prod_{\nu} \delta \left\{ z^\nu - (1-\Lambda_\nu) \frac{e^{-\beta\phi'(\lambda^*_\nu)} \sum_{l=1}^{k-1} z_l^\nu}{1 + e^{-\beta\phi'(\lambda^*_\nu)} \sum_{(lr)}^{k-1} z_l^\nu z_r^\nu} - \Lambda_\nu \frac{1}{\sum_{l=1}^{k-1} z_l^\nu} \right\} \tag{S59}$$

1.3.5 The derivation of $\rho(\vec{\lambda}^*|\vec{h})$

To derive $\rho(\vec{\lambda}^*|\vec{h})$, we first note that the function $F_\nu(\vec{h}, \mathbf{S}, \hat{\underline{\lambda}})$ is defined as the saddle point of the integration (S20), i.e.

$$\hat{P}(\mathbf{S}, \hat{\underline{\lambda}}) = \langle k\rangle \prod_{\nu\alpha} \left[\int_{\odot} d\vec{S}'^{\alpha}_{\nu} \right] \prod_{\alpha} \left[\int d\hat{\lambda}'_\alpha \right] P(\mathbf{S}', \hat{\underline{\lambda}}') \prod_{\nu\alpha} \left(1 + e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}'_\alpha}{2}} \vec{S}^\alpha_\nu \cdot \vec{S}'^\alpha_\nu \right) .$$

From the definition of $\rho(\vec{\lambda}^*|\vec{h})$ in Eq. (S58), we compute λ^*_ν by evaluating the saddle point in the following integration

$$\int_{\odot} d\vec{S} \int d\hat{\underline{\lambda}} P(\mathbf{S}, \hat{\underline{\lambda}}) \hat{P}(\mathbf{S}, \hat{\underline{\lambda}}) =$$
$$\langle k\rangle \prod_{\nu\alpha} \left[\int_{\odot} d\vec{S}^\alpha_\nu \int_{\odot} d\vec{S}'^\alpha_\nu \right] \prod_{\alpha} \left[\int d\hat{\lambda}_\alpha d\hat{\lambda}'_\alpha \right] P(\vec{S}, \hat{\underline{\lambda}}) P(\mathbf{S}', \hat{\underline{\lambda}}') \prod_{\nu\alpha} \left(1 + e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}'_\alpha}{2}} \vec{S}^\alpha_\nu \cdot \vec{S}'^\alpha_\nu \right) . \tag{S60}$$

Equation (S60) can be interpreted as the interaction between two vertices sharing an edge, where variables denoted by a prime give rise to a *backward field* with respect to the one *forward field* generated by non-primed variables, as shown schematically in Fig. S2 (a), to acknowledge the central node of the current state of the ancestor node.

To obtain λ_α^*, we follow the same procedure as before by inserting the ansatz (S23) into Eq. (S60)

$$\begin{aligned}
&\prod_{\nu\alpha}\left[\int_\odot d\vec{S}_\nu^\alpha \int_\odot d\vec{S}_\nu^{\prime\alpha}\right]\prod_\alpha\left[\int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi} e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_\alpha\left[\int \frac{d\lambda_\alpha^\prime d\hat{\lambda}_\alpha^\prime}{2\pi} e^{iM\lambda_\alpha^\prime\hat{\lambda}_\alpha^\prime-\beta M\phi(\lambda_\alpha)}\right]\\
&\quad\times\int d\vec{h}\ w(\vec{h})\int d\vec{h}^\prime\ w(\vec{h}^\prime)\prod_{\nu\alpha}\left[h_0^\nu+h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}}S_{\nu,a}^\alpha+h_2^\nu e^{-i\hat{\lambda}_\alpha}(S_{\nu,a}^\alpha)^2\right]\\
&\qquad\times\prod_{\nu\alpha}\left[h_0^{\nu^\prime}+h_1^{\nu^\prime}e^{-\frac{i\hat{\lambda}_\alpha^\prime}{2}}S_{\nu,a}^{\prime\alpha}+h_2^{\nu^\prime}e^{-i\hat{\lambda}_\alpha^\prime}(S_{\nu,a}^{\prime\alpha})^2\right]\prod_{\nu\alpha}\left(1+e^{-\frac{i\hat{\lambda}_\alpha+i\hat{\lambda}_\alpha^\prime}{2}}\vec{S}_\nu^\alpha\cdot\vec{S}_\nu^{\prime\alpha}\right).\\
&=\int d\vec{h}\ w(\vec{h})\int d\vec{h}^\prime\ w(\vec{h}^\prime)\\
&\quad\times\prod_\alpha\left[\int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi} e^{iM\lambda_\alpha\hat{\lambda}_\alpha-\beta M\phi(\lambda_\alpha)}\right]\prod_\alpha\left[\int \frac{d\lambda_j^\alpha d\hat{\lambda}_\alpha^\prime}{2\pi} e^{iM\lambda_\alpha^\prime\hat{\lambda}_\alpha^\prime-\beta M\phi\left(\lambda_\alpha^\prime\right)}\right]\\
&\qquad\times\prod_{\nu\alpha}\left[(h_0^\nu+h_2^\nu e^{-i\hat{\lambda}_\alpha})(h_0^{\nu^\prime}+h_2^{\nu^\prime}e^{-i\hat{\lambda}_\alpha^\prime})+h_1^\nu h_1^{\nu^\prime}e^{-i\hat{\lambda}_\alpha-i\hat{\lambda}_\alpha^\prime}\right]
\end{aligned}\tag{S61}$$

We can now evaluate the saddle point of $\lambda_\alpha^\prime$ when $M\to\infty$ by writing the above as

$$\int d\vec{h}\ w(\vec{h})\int d\vec{h}^\prime\ w(\vec{h}^\prime)\prod_\alpha\left[\int\frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi}\right]\prod_\alpha\left[\int\frac{d\lambda_\alpha^\prime d\hat{\lambda}_\alpha^\prime}{2\pi}\right]e^{M\Xi}\tag{S62}$$

where Ξ is

$$\begin{aligned}
\Xi=&\sum_\alpha\left[\lambda_\alpha\hat{\lambda}_\alpha-\beta\phi\left(\lambda_\alpha\right)+\lambda_\alpha^\prime\hat{\lambda}_\alpha^\prime-\beta\phi\left(\lambda_\alpha^\prime\right)\right]\\
&+\frac{1}{M}\sum_{\nu\alpha}\left[(h_0^\nu+h_2^\nu e^{-i\hat{\lambda}_\alpha})(h_0^{\nu^\prime}+h_2^{\nu^\prime}e^{-i\hat{\lambda}_\alpha^\prime})+h_1^\nu h_1^{\nu^\prime}e^{-i\hat{\lambda}_\alpha-i\hat{\lambda}_\alpha^\prime}\right]
\end{aligned}\tag{S63}$$

which leads us to the following symmetry coupled equations in λ_α and $\lambda_\alpha^\prime$

$$\lambda_\alpha=\frac{1}{M}\sum_\nu\frac{h_2^\nu e^{-\beta\phi^\prime(\lambda_\alpha)}\left[h_0^{\nu^\prime}+h_2^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha^\prime)}\right]+h_1^\nu h_1^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha)-\beta\phi^\prime(\lambda_\alpha^\prime)}}{\left[h_0^\nu+h_2^\nu e^{-\beta\phi^\prime(\lambda_\alpha)}\right]\left[h_0^{\nu^\prime}+h_2^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha^\prime)}\right]+h_1^\nu h_1^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha)-\beta\phi^\prime(\lambda_\alpha^\prime)}}\tag{S64}$$

$$\lambda_\alpha^\prime=\frac{1}{M}\sum_\nu\frac{h_2^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha^\prime)}\left[h_0^\nu+h_2^\nu e^{-\beta\phi^\prime(\lambda_\alpha)}\right]+h_1^\nu h_1^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha)-\beta\phi^\prime(\lambda_\alpha^\prime)}}{\left[h_0^\nu+h_2^\nu e^{-\beta\phi^\prime(\lambda_\alpha)}\right]\left[h_0^{\nu^\prime}+h_2^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha^\prime)}\right]+h_1^\nu h_1^{\nu^\prime}e^{-\beta\phi^\prime(\lambda_\alpha)-\beta\phi^\prime(\lambda_\alpha^\prime)}}.\tag{S65}$$

To obtain $\lambda_\alpha^{\prime*}$, the exact way is to solve the above coupled equations simultaneously for both $\lambda_\alpha^{\prime*}$ and λ_α^*, which is computationally complicated. An alternative way which yields a simple yet reasonable estimate of $\lambda_\alpha^{\prime*}$ is to assume that λ_α^* is known, in this case we focus on solving Eq. (S65) and using the simplification of z^ν from Eq. (S48) we rewrite Eq. (S65) as

$$\lambda_{\alpha}^{'} = \frac{1}{M}\sum_{\nu} \frac{h_2^{\nu'} e^{-\beta\phi'(\lambda_{\alpha}^{'})} + z^{\nu} h_1^{\nu'} e^{-\beta\phi'(\lambda_{\alpha}^{'})}}{\left[h_0^{\nu'} + h_2^{\nu'} e^{-\beta\phi'(\lambda_{\alpha}^{'})}\right] + z^{\nu} h_1^{\nu'} e^{-\beta\phi'(\lambda_{\alpha}^{'})}}. \tag{S66}$$

Furthermore, the fields $h_0^{\nu'}$, $h_1^{\nu'}$ and $h_2^{\nu'}$ can be expressed in terms of the z_l^{ν} of the neighboring nodes $l = 1, \ldots, k_j - 1$ (see Fig. S2(a) for schematic diagram) via the relation between h and g^{ν}'s in Eq. (S44) and the relation between g^{ν}'s and z's in Eqs. (S50) and (S51). Thus, we arrive at the following equation for $\lambda_{\alpha}^{'}$

$$\lambda_{\alpha}^{'} = \frac{1}{M}\sum_{\nu} \left\{ \Lambda_{\nu} + (1-\Lambda_{\nu}) \frac{e^{-\beta\phi'(\lambda_{\alpha}^{'})} \sum_{(lr)}^{k-1} z_l^{\nu} z_r^{\nu} + e^{-\beta\phi'(\lambda_{\alpha}^{'})} z^{\nu} \sum_{l=1}^{k-1} z_l^{\nu}}{\left[1 + e^{-\beta\phi'(\lambda_{\alpha}^{'})} \sum_{(lr)}^{k-1} z_l^{\nu} z_r^{\nu}\right] + e^{-\beta\phi'(\lambda_{\alpha}^{'})} z^{\nu} \sum_{l=1}^{k-1} z_l^{\nu}} \right\} \tag{S67}$$

which constitutes the function $f(x; z_1^{\nu}, \ldots, z_k^{\nu})$ given by

$$f(x; z_1^{\nu}, \ldots, z_k^{\nu}) = x - \frac{1}{M}\sum_{\nu} \left\{ \Lambda_{\nu} + (1-\Lambda_{\nu}) \frac{e^{-\beta\phi'(x)} \sum_{(lr)}^{k} z_l^{\nu} z_r^{\nu}}{1 + e^{-\beta\phi'(x)} \sum_{(lr)}^{k} z_l^{\nu} z_r^{\nu}} \right\}. \tag{S68}$$

such that $\lambda_{\alpha}^{'*} = f^{-1}(0; z_1^{\nu}, \ldots, z_k^{\nu})$. This gives rise to

$$\begin{aligned}
\rho(\vec{\lambda}^{*}|\vec{h}(\vec{z}_1, \ldots, \vec{z}_{k-1})) &= \int_{\odot} d\mathbf{S}' \int d\hat{\underline{\lambda}}' P(\mathbf{S}', \hat{\underline{\lambda}}') \prod_{\nu} \delta[\lambda_{\nu}^{*} - F_{\nu}(\vec{h}(\vec{z}_1, \ldots, \vec{z}_{k-1}), \mathbf{S}', \hat{\underline{\lambda}}')] \\
&= \int d\vec{h}' w(\vec{h}') \int_{\odot} d\mathbf{S}' \int d\hat{\underline{\lambda}}' P(\mathbf{S}', \hat{\underline{\lambda}}') \prod_{\nu} \delta[\lambda_{\nu}^{*} - F_{\nu}(\vec{h}(\vec{z}_1, \ldots, \vec{z}_{k-1}), \mathbf{S}', \hat{\underline{\lambda}}')] \\
&= \int d\vec{h}' w(\vec{h}') \int_{\odot} d\mathbf{S}' \int d\hat{\underline{\lambda}}' P(\mathbf{S}', \hat{\underline{\lambda}}') \int d\vec{z}' \\
&\times \prod_{\nu} \left\{ \delta\left(z^{\nu\prime} - \frac{h_1^{\nu\prime} e^{-\beta\phi'(F_{\nu}(\vec{h}', \mathbf{S}', \hat{\underline{\lambda}}'))}}{h_0^{\nu\prime} + h_2^{\nu\prime} e^{-\beta\phi'(F_{\nu}(\vec{h}', \mathbf{S}', \hat{\underline{\lambda}}'))}} \right) \delta\left[\lambda_{\nu}^{*} - f^{-1}(0; \vec{z}_1, \ldots, \vec{z}_{k-1}, \vec{z}')\right] \right\}
\end{aligned} \tag{S69}$$

where we have introduced the delta function for the variable $z^{\nu\prime}$ and at the same time replace $F_{\nu}\vec{h}(\vec{z}_1, \ldots, \vec{z}_{k-1}), \mathbf{S}', \hat{\underline{\lambda}}')$ by $f^{-1}(0; \vec{z}_1, \ldots, \vec{z}_{k-1}, \vec{z}')$. Apply the change of variable (S53) we arrive at

$$\rho(\vec{\lambda}^{*}|\vec{h}(\vec{z}_1, \ldots, \vec{z}_{k-1})) = \int d\vec{z}' w(\vec{z}') \prod_{\nu} \delta\left[\lambda_{\nu}^{*} - f^{-1}(0; \vec{z}_1, \ldots, \vec{z}_{k-1}, \vec{z}')\right] \tag{S70}$$

Substitute of this expression into Eq. (S59), we obtain

$$\begin{aligned}
w(\vec{z}) = &\sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k \rangle} \prod_{l=1}^{k} \left[\int d\vec{z}_l \; w(\vec{z}_l) \right] \prod_{\nu} \sum_{\Lambda_{\nu}=0}^{1} \left[\delta_{\Lambda_{\nu},0} + \frac{2}{N} \delta_{\Lambda_{\nu},1} \right] \\
&\times \prod_{\nu} \left\{ \int d\lambda_{\nu}^{*} \delta\left[\lambda_{\nu}^{*} - f^{-1}(0; \vec{z}_1, \ldots, \vec{z}_{k-1}, \vec{z}_k)\right] \right\} \\
&\times \prod_{\nu} \delta \left\{ z^{\nu} - (1-\Lambda_{\nu}) \frac{e^{-\beta\phi'(\lambda_{\nu}^{*})} \sum_{l=1}^{k-1} z_l^{\nu}}{1 + e^{-\beta\phi'(\lambda_{\nu}^{*})} \sum_{(lr)} z_l^{\nu} z_r^{\nu}} - \Lambda_{\nu} \frac{1}{\sum_{l=1}^{k-1} z_l^{\nu}} \right\}
\end{aligned} \tag{S71}$$

which resembles an ordinary RS cavity equation except old the product for $w(\vec{z}_l)$ runs from $l = 1$ to k, instead of from $l = 1$ to $k - 1$. The origin of the difference is due to the approximation we employed in Eq. (S43).

We note that Eq. (S67) gives the average number of polymers/communications that pass through the node, as each term ν in the summation on the right corresponds to the probability that polymer ν passes through the node. We will see in the next section that the above equation is used to compute the flow through a node.

Equation (S68) is only valid at large M while we are also interested in cases when M is small or intermediate. To derive an approximate expression suitable for finite M, we start from Eq. (S61) and instead of writing it as an exponent of $e^{M\Xi}$ as in Eq. (S62), we isolate one communication ν and write it as

$$\begin{aligned}
&\int d\vec{h}\; w(\vec{h}) \int d\vec{h}'\; w(\vec{h}') \\
&\quad \times \prod_{\alpha} \left[\int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi} e^{iM\lambda_\alpha \hat{\lambda}_\alpha - \beta M \phi(\lambda_\alpha)} \right] \prod_{\alpha} \left[\int \frac{d\lambda'_\alpha d\hat{\lambda}'_\alpha}{2\pi} e^{iM\lambda'_\alpha \hat{\lambda}'_\alpha - \beta M \phi\left(\lambda'_\alpha\right)} \right] \\
&\quad \times \prod_{\alpha} \left[h_0^\nu h_0^{\nu'} + h_2^\nu h_0^{\nu'} e^{-i\hat{\lambda}_\alpha} + h_0^\nu h_2^{\nu'} e^{-i\hat{\lambda}'_\alpha} + (h_2^\nu h_2^{\nu'} + h_1^\nu h_1^{\nu'}) e^{-i\hat{\lambda}_\alpha - i\hat{\lambda}'_\alpha} \right] \\
&\quad \times \prod_{\mu \neq \nu, \alpha} \left[(h_0^\mu + h_2^\mu e^{-i\hat{\lambda}_\alpha})(h_0^{\mu'} + h_2^{\mu'} e^{-i\hat{\lambda}'_\alpha}) + h_1^\mu h_1^{\mu'} e^{-i\hat{\lambda}_\alpha - i\hat{\lambda}'_\alpha} \right]
\end{aligned} \tag{S72}$$

The rationale behind this approach is similar to that of the cavity approach, but now in *the space of M communications*, where we isolate one communication/polymer and evaluate how it is influenced by all other communications through the variable λ'_α. This process will give us an expression for λ'^{*}_α which is exact when $M = 1$, i.e. only communication ν exists. We continue our derivation by writing the integrand in the above expression in the form of $e^{M\Xi}$, i.e.

$$\begin{aligned}
&\int d\vec{h}\; w(\vec{h}) \int d\vec{h}'\; w(\vec{h}') \left\{ \prod_{\alpha} \left[h_0^\nu h_0^{\nu'} \int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha d\lambda'_\alpha d\hat{\lambda}'_\alpha}{(2\pi)^2} e^{M\Xi_{\nu,1}} \right.\right. \\
&\quad + h_0^\nu h_2^{\nu'} \int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha d\lambda'_\alpha d\hat{\lambda}'_\alpha}{(2\pi)^2} e^{M\Xi_{\nu,2}} + h_2^\nu h_0^{\nu'} \int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha d\lambda'_\alpha d\hat{\lambda}'_\alpha}{(2\pi)^2} e^{M\Xi_{\nu,3}} \\
&\quad \left.\left. + (h_2^\nu h_2^{\nu'} + h_1^\nu h_1^{\nu'}) \int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha d\lambda'_\alpha d\hat{\lambda}'_\alpha}{(2\pi)^2} e^{M\Xi_{\nu,4}} \right] \right\}.
\end{aligned} \tag{S73}$$

All the Ξ exponents carry a subscript ν to represent the isolation of communication ν. As only the last two terms carry the coefficient $e^{-i\hat{\lambda}'_\alpha}$, we will only look for the saddle points of $\Xi_{\nu,3}$ and $\Xi_{\nu,4}$ with respect to $\hat{\lambda}'_\alpha$. This is clearly an approximation as the saddle point does not dominate the integral at finite M values. They are given by

$$
\begin{aligned}
\Xi_{\nu,3}(\vec{S},\underline{\hat{\lambda}}) =& -\frac{i\hat{\lambda}_\alpha^{'}}{M} + i\lambda_\alpha^{'}\hat{\lambda}_\alpha^{'} + i\lambda_\alpha\hat{\lambda}_\alpha - \beta\phi\left(\lambda_\alpha^{'}\right) - \beta\phi\left(\lambda_\alpha\right) \\
&+ \frac{1}{M}\sum_{\mu\neq\nu}\log\left[(h_0^\mu + h_2^\mu e^{-i\hat{\lambda}_\alpha})(h_0^{\mu'} + h_2^{\mu'} e^{-i\hat{\lambda}_\alpha^{'}}) + h_1^\mu h_1^{\mu'} e^{-i\hat{\lambda}_\alpha - i\hat{\lambda}_\alpha^{'}}\right] \qquad \text{(S74)}\\
\Xi_{\nu,4}(\vec{S},\underline{\hat{\lambda}}) =& -\frac{i\hat{\lambda}_\alpha^{'}}{M} - \frac{i\hat{\lambda}_\alpha}{M} + i\lambda_\alpha^{'}\hat{\lambda}_\alpha^{'} + i\lambda_\alpha\hat{\lambda}_\alpha - \beta\phi\left(\lambda_\alpha^{'}\right) - \beta\phi\left(\lambda_\alpha\right) \\
&+ \frac{1}{M}\sum_{\mu\neq\nu}\log\left[(h_0^\mu + h_2^\mu e^{-i\hat{\lambda}_\alpha})(h_0^{\mu'} + h_2^{\mu'} e^{-i\hat{\lambda}_\alpha^{'}}) + h_1^\mu h_1^{\mu'} e^{-i\hat{\lambda}_\alpha - i\hat{\lambda}_\alpha^{'}}\right]. \qquad \text{(S75)}
\end{aligned}
$$

Differentiating with respect to $\lambda_\alpha^{'}$ we find that the saddle point equation obtained from both $\Xi_{\nu,3}$ and $\Xi_{\nu,4}$ are identical and given by

$$
\lambda_\alpha^{'} = \frac{1}{M}\left[1 + \sum_{\mu\neq\nu}\frac{h_2^{\mu'} e^{-\beta\phi'(\lambda_\alpha^{'})}\left[h_0^\mu + h_2^\mu e^{-\beta\phi'(\lambda_\alpha)}\right] + h_1^\mu h_1^{\mu'} e^{-\beta\phi'(\lambda_\alpha)-\beta\phi'(\lambda_\alpha^{'})}}{\left[h_0^\mu + h_2^\mu e^{-\beta\phi'(\lambda_\alpha)}\right]\left[h_0^{\mu'} + h_2^{\mu'} e^{-\beta\phi'(\lambda_\alpha^{'})}\right] + h_1^\mu h_1^{\mu'} e^{-\beta\phi'(\lambda_\alpha)-\beta\phi'(\lambda_\alpha^{'})}}\right]. \qquad \text{(S76)}
$$

We express $h_0^{\mu'}$, $h_1^{\mu'}$ and $h_2^{\mu'}$ in terms of the z_l^ν's as for deriving Eq. (S67). These give the following equation in $\lambda_\alpha^{'}$

$$
\lambda_\alpha^{'} = \frac{1}{M} + \frac{1}{M}\sum_{\mu\neq\nu}\left\{\Lambda_\nu + (1-\Lambda_\nu)\frac{e^{-\beta\phi'(\lambda_\alpha^{'})}\sum_{(lr)}^{k-1} z_l^\nu z_r^\nu + e^{-\beta\phi'(\lambda_\alpha^{'})} z^\nu \sum_{l=1}^{k-1} z_l^\nu}{\left[1 + e^{-\beta\phi'(\lambda_\alpha^{'})}\sum_{(lr)}^{k-1} z_l^\nu z_r^\nu\right] + e^{-\beta\phi'(\lambda_\alpha^{'})} z^\nu \sum_{l=1}^{k-1} z_l^\nu}\right\} \qquad \text{(S77)}
$$

which is very similar to Eq. (S67) except for the first term $\frac{1}{M}$ and the summation which runs over all communications other than ν. This expression leads to $\lambda_\alpha^{'*} = 1$ when $M = 1$, which is equivalent to multiplying the fields h_1^ν and h_2^ν by a constant factor $e^{-\beta\phi'(1)}$ and an energy increase whenever ν passes the node, thus an exact counting of energy when $M = 1$. On the other hand, when $M \to \infty$, $1/M \to 0$ and $\lambda_\alpha^{'}$ gives the correct counting of the normalized traffic through the node as in Eq. (S67). While we expect Eq. (S67) and Eq. (S77) to become identical as $M \to \infty$, Eq. (S67) provides an additional term for low M values.

As Eq. (S77) depends on the communication ν, we will define a communication dependent function $f_\nu(x;\vec{z}_1,\ldots,\vec{z}_k)$ where $\lambda_\nu^* = f_\nu^{-1}(0;\vec{z}_1,\ldots,\vec{z}_k)$ gives an estimate of λ_ν^* for each individual communication ν. The function $f_\nu(x)$ is given by

$$
f_\nu(x;\vec{z}_1,\ldots,\vec{z}_k) = x - \frac{1}{M} - \frac{1}{M}\sum_{\mu\neq\nu}\left\{\Lambda_\nu + (1-\Lambda_\nu)\frac{e^{-\beta\phi'(x)}\sum_{(lr)}^{k} z_l^\mu z_r^\mu}{1 + e^{-\beta\phi'(x)}\sum_{(lr)}^{k} z_l^\mu z_r^\mu}\right\}. \qquad \text{(S78)}
$$

The final recursion equation for $w(\vec{z})$ is identical to Eq. (S71) except for $f^{-1}(0;\vec{z}_1,\ldots,\vec{z}_k)$ which is replaced by $f_\nu^{-1}(0;\vec{z}_1,\ldots,\vec{z}_k)$ in the delta function of λ_ν^*. The equation is given by Eq. (S79) in the next subsection.

1.3.6 Summary of the solution for finite M

Here we summarize the obtained solution to highlight the main results. After the replica calculation, we obtain a self-consistent equation for the distribution $w(\vec{z})$, where $\vec{z}$ is an M-dimensional field vector. The self-consistent equation is given by Eq. (S71),

$$w(\vec{z}) = \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k \rangle} \prod_{l=1}^{k} \left[\int d\vec{z}_l \; w(\vec{z}_l) \right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N} \delta_{\Lambda_\nu,1} \right]$$
$$\times \prod_{\nu} \left\{ \int d\lambda_\nu^* \delta \left[\lambda_\nu^* - f_\nu^{-1}(0; \vec{z}_1, \ldots, \vec{z}_k) \right] \right\}$$
$$\times \prod_{\nu} \delta \left\{ z^\nu - (1 - \Lambda_\nu) \frac{e^{-\beta\phi'(\lambda_\nu^*)} \sum_{l=1}^{k-1} z_l^\nu}{1 + e^{-\beta\phi'(\lambda_\nu^*)} \sum_{(lr)}^{k-1} z_l^\nu z_r^\nu} - \Lambda_\nu \frac{1}{\sum_{l=1}^{k-1} z_l^\nu} \right\}. \quad \text{(S79)}$$

where $f_\nu(x; \vec{z}_1, \ldots, \vec{z}_k)$ is given by Eq. (S78),

$$f_\nu(x; \vec{z}_1, \ldots, \vec{z}_k) = x - \frac{1}{M} - \frac{1}{M} \sum_{\mu \neq \nu} \left\{ \Lambda_\mu + (1 - \Lambda_\mu) \frac{e^{-\beta\phi'(x)} \sum_{(lr)}^{k} z_l^\mu z_r^\mu}{1 + e^{-\beta\phi'(x)} \sum_{(lr)}^{k} z_l^\mu z_r^\mu} \right\}. \quad \text{(S80)}$$

This recursion equation for $w(\vec{z})$ resembles an ordinary RS cavity equation except that the product for $w(\vec{z}_l)$ runs from $l = 1$ to k, instead of from $l = 1$ to $k - 1$. The origin of the difference is the approximation we employed in Eq. (S43). The analytic solution yields results that are in good agreement with simulations obtained using the suggested algorithm in most of the parameter space. Moreover, comparison of the results with those obtained by the full cavity equations, which are computationally feasible only for systems with small M values described in Sec. S2.4 show excellent agreement between the two. These numerical results provide a justification for using this approximation, especially since the conventional cavity equations are exact but infeasible for realistic systems. More detailed discussions are also found in Section S2.5.

Note that the suggested approach is highly efficient computationally in comparison with the conventional cavity method. If we were to use the conventional cavity approach, a cavity field spanning the domain of all the possible 2^M states of each node would had to be established, representing each of the M possible communications that potentially go though any of the nodes in the graph. We have derived the corresponding equations and numerically compared them in Sec. S2.4 for small M values only, due to the prohibitive computational cost; the results obtained are consistent with those obtained via the approximate algorithm. The suggested replica approach incorporates the interaction of the M communications into the variable λ_ν^*, which results in an equation of M-dimensional fields which is feasible to solve even for large M.

To compute physical quantities of interest, we first obtain a stable $w(z)$ by iterating Eq. (S79) until convergence and use it to compute the distribution $P(I)$, where I denotes the number of communications passing through a node and is defined by Eq. (S67). The equation for $P(I)$ is given by

$$P(I) = \sum_{k=1}^{\infty} \rho(k) \prod_{l=1}^{k} \left[\int d\vec{z}_l \ w(\vec{z}_l) \right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N} \delta_{\Lambda_\nu,1} \right]$$
$$\times \prod \left\{ \int d\lambda_\nu^* \delta \left[\lambda_\nu^* - f_\nu^{-1}(0; \vec{z}_1, \ldots, \vec{z}_k) \right] \right\}$$
$$\times \prod_{\nu} \delta \left\{ I - \sum_{\nu} \left[\Lambda_\nu - (1-\Lambda_\nu) \frac{e^{-\beta\phi'(\lambda_\nu^*)} \sum_{(lr)}^{k} z_l^\nu z_r^\nu}{1 + e^{-\beta\phi'(\lambda_\nu^*)} \sum_{(lr)}^{k} z_l^\nu z_r^\nu} \right] \right\}, \tag{S81}$$

where $f_\nu(x; \vec{z}_1, \ldots, \vec{z}_k)$ is given by Eq. (S80). With $\langle \ldots \rangle$ denoting the average over $P(I)$, the average energy is given by $\langle E \rangle = MN \langle \phi(I/M) \rangle$, the average path length is the average occupancy per node and is given by $\langle L \rangle = \langle I \rangle / M$ and the fraction of idle node is given by $f_{\text{idle}} = \langle \delta(I) \rangle$.

1.4 The zero temperature solution

After obtaining the solution at finite temperature, the zero-temperature solution is straightforward. In all previous derivations, we can substitute

$$h_0^\nu = e^{-\beta E_0^\nu} \tag{S82}$$
$$h_1^\nu = e^{-\beta E_1^\nu} \tag{S83}$$
$$h_2^\nu = e^{-\beta E_2^\nu}. \tag{S84}$$

We then define $z^\nu = e^{-\beta u^\nu}$ such that from Eqs. (S49) and (S52), we obtain the following recursion of u's as $\beta \to \infty$

$$u^\nu = E_1^\nu + \phi'(\lambda_\nu^*) - \min[E_0^\nu, E_2^\nu + \phi'(\lambda_\nu^*)]$$
$$= (1-\Lambda_\nu) \left\{ \phi'(\lambda_\nu^*) + \min_{l \le k-1}[u_l^\nu] - \min \left[0, \min_{\substack{l,r \le k-1 \\ l \ne r}} [\phi'(\lambda_\nu^*) + u_l^\nu + u_r^\nu] \right] \right\} - \Lambda_\nu \min_{l \le k-1}[u_l^\nu]$$
$$= (1-\Lambda_\nu) \left\{ \min_{l \le k-1}[u_l^\nu] - \min \left[-\phi'(\lambda_\nu^*), \min_{\substack{l,r \le k-1 \\ l \ne r}} [u_l^\nu + u_r^\nu] \right] \right\} - \Lambda_\nu \min_{l \le k-1}[u_l^\nu]. \tag{S85}$$

The self-consistent equation for the distribution $w(\vec{u})$ is given by the zero-temperature limit of Eq. (S79)

$$
\begin{aligned}
w(\vec{u}) = &\sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k\rangle} \prod_{l=1}^{k} \left[\int d\vec{u}_l \; w(\vec{u}_l)\right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N}\delta_{\Lambda_\nu,1}\right] \\
&\times \prod_{\nu} \left\{\int d\lambda_\nu^* \delta\left[\lambda_\nu^* - f_\nu^{-1}(0; \vec{u}_1, \ldots, \vec{u}_k)\right]\right\} \\
&\times \prod_{\nu} \delta\left\{u^\nu - (1-\Lambda_\nu)\left(\min_{l\le k-1}[u_l^\nu] - \min\left[-\phi'(\lambda_\nu^*), \min_{\substack{l,r\le k-1\\ l\neq r}}[u_l^\nu + u_r^\nu]\right]\right) - \Lambda_\nu \min_{l\le k-1}[u_l^\nu]\right\}.
\end{aligned}
\tag{S86}
$$

where $f_\nu(x; \vec{u}_1, \ldots, \vec{u}_k)$ is given by the zero-temperature limit of Eq. (S80), i.e.

$$
f_\nu(x) = x - \frac{1}{M} - \frac{1}{M}\sum_{\mu\neq\nu}\left\{\Lambda_\mu + (1-\Lambda_\mu)\Theta\left(-\phi'(x) - \min_{\substack{l,r\le k\\ l\neq r}}[u_l^\mu + u_r^\mu]\right)\right\}. \tag{S87}
$$

One may solve the Eq. (S87) by setting $x = I/M$ and testing integer values of I from $I = 0$ until a solution is found; if no solution is found we use the two x values that give the smallest positive and the largest negative value of $f_\nu(x)$ and extrapolate to obtain a fractional x^* which gives $f_\nu(x^*) = 0$. Methods to reduce computational complexity in solving Eq. (S87) are discussed in Section S2.2.

Similarly, the distribution $P(I)$, of the number of communications passing through a node is given by

$$
\begin{aligned}
P(I) = &\sum_{k=1}^{\infty} \rho(k) \prod_{l=1}^{k} \left[\int d\vec{u}_l \; w(\vec{u}_l)\right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N}\delta_{\Lambda_\nu,1}\right] \\
&\times \prod_{\nu} \left\{\int d\lambda_\nu^* \delta\left[\lambda_\nu^* - f_\nu^{-1}(0; \vec{u}_1, \ldots, \vec{u}_k)\right]\right\} \\
&\times \prod_{\nu} \delta\left\{I - \sum_{\nu}\left[\Lambda_\nu - (1-\Lambda_\nu)\Theta\left(-\phi'(\lambda_\nu^*) - \min_{\substack{l,r\le k\\ l\neq r}}[u_l^\nu + u_r^\nu]\right)\right]\right\},
\end{aligned}
\tag{S88}
$$

where $f_\nu(x; \vec{u}_1, \ldots, \vec{u}_k)$ is given by Eq. (S87). Physical quantities such as $\langle E\rangle = MN\langle\phi(I/M)\rangle$, $\langle L\rangle = \langle I\rangle/M$ and f_{idle} are computed as in the case of finite temperature (section 1.3.6).

1.5 The directed formulation

In this subsection, we show that the present framework accommodates directed communications and will lead to the algorithm expressed by Eqs. (S103) to (S106) in Section S2; it involves passing a pair of messages $a_{j\to i}^\nu$ and $b_{j\to i}^\nu$, that correspond to the corresponding energy terms when communication ν passes from i towards j and from j towards i, respectively. Since the calculation is rather involved and greatly resembles the derivation in previous Section S1.1 to S1.3, we will only outline the main steps of the derivation. We start with the partition function

$$\begin{aligned}\mathcal{Z} =& \prod_{i=1}^{N} \left[\int \frac{d\lambda_i d\hat{\lambda}_i}{2\pi} e^{iM\lambda_i\hat{\lambda}_i - \beta M\phi(\lambda_i)} \right] \prod_{i\nu} \left[\frac{1}{C_n^2} \int_{\odot} d\vec{S}_i^{\nu} \int_{\odot} d\vec{Z}_{i\nu} \right] \prod_{i\nu} \left[\sum_{\sigma_{i\nu}=0,1} \left(1 - \sigma_{i\nu} + \sigma_{i\nu} \vec{S}_i^{\nu} \cdot \vec{Z}_{i\nu} \right) \right] \\ & \times \prod_{\nu=1}^{M} \left\{ \prod_{(ij)} \left[\left(\sigma_{i\nu}\sigma_{j\nu} e^{-\frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}} Z_{i,a}^{\nu} S_{j,a}^{\nu} \right)^{\Lambda_{ij}^{\nu}(1-\Lambda_{ji}^{\nu})} + \left(\sigma_{i\nu}\sigma_{j\nu} e^{-\frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}} S_{i,a}^{\nu} Z_{j,a}^{\nu} \right)^{\Lambda_{ji}^{\nu}(1-\Lambda_{ij}^{\nu})} \right] \right. \\ & \left. \times \prod_{(kl)} \left[1 + A_{kl}\sigma_{k\nu}\sigma_{l\nu} e^{-\frac{i\hat{\lambda}_k + i\hat{\lambda}_l}{2}} \left(\vec{S}_k^{\nu} \cdot \vec{Z}_l^{\nu} + \vec{Z}_k^{\nu} \cdot \vec{S}_l^{\nu} \right) \right] \right\}, \end{aligned} \tag{S89}$$

which differs from the partition function in Eq. (S2) in the following:

1. Additional 0-vector variables $\vec{Z}_{i\nu}$ are introduced such that $\vec{S}_i^{\nu}$ and $\vec{Z}_{i\nu}$ of node i correspond to its *outgoing* and *incoming* cavity for communication ν. As we can see from the last line of the partition function, the edge (kl) either connect the outgoing cavity of node k to the incoming cavity of node l, or the incoming cavity of node k to the outgoing cavity of node l when $A_{kl} = 1$.

2. The variable $\sigma_{i\nu} = 0, 1$ and the factor $\left(1 - \sigma_{i\nu} + \sigma_{i\nu}\vec{S}_i^{\nu} \cdot \vec{Z}_{i\nu}\right)$ are introduced for each node i and communication ν; $\sigma_{k\nu} = \sigma_{l\nu} = 1$ when ν passes the edge (kl), such that the factor $\left(1 - \sigma_{k\nu} + \sigma_{k\nu}\vec{S}_k^{\nu} \cdot \vec{Z}_k^{\nu}\right)$ connects one of the incoming cavities of node k to one of its outgoing cavities when ν passes through k.

3. The variable Λ_{ij}^{ν} is directed, such that $\Lambda_{ij}^{\nu} = 1$ ($\Lambda_{ji}^{\nu} = 1$) when ν corresponds to a communication from source i to sink j (source j to sink i) and $\Lambda_{ij}^{\nu} = \Lambda_{ji}^{\nu} = 0$ otherwise; the two terms $Z_{i,a}^{\nu}S_{j,a}^{\nu}$ and $S_{i,a}^{\nu}Z_{j,a}^{\nu}$ thus correspond to communications in the respective directions.

One can proceed with the derivation as in Section S1 to arrive at the corresponding saddle point equations (similar to Eqs. (S19) and (S20)) for the functional order parameter $P(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma})$ and $\hat{P}(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma})$

$$\begin{aligned} P(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma}) =& \sum_k \frac{\rho(k)}{D_k} \prod_{\alpha} \left[\int \frac{d\lambda_\alpha}{2\pi} e^{iM\lambda_\alpha\hat{\lambda}_\alpha - \beta M\phi(\lambda_\alpha)} \right] \\ & \times k[\hat{P}(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma})]^{k-1} e^{\sum_\nu \hat{Q}(\underline{S}_{\nu,a}, \underline{Z}_{\nu,a}, \underline{\hat{\lambda}}, \sigma_\nu^\alpha)} \prod_{\nu\alpha} \left(1 - \sigma_\nu^\alpha + \sigma_\nu^\alpha \vec{S}_\nu^\alpha \cdot \vec{Z}_\nu^\alpha \right) \end{aligned} \tag{S90}$$

$$\begin{aligned} \hat{P}(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma}) =& \langle k \rangle \prod_{\nu\alpha} \left[\sum_{\sigma_\nu^{\alpha\prime}=0,1} \right] \prod_{\nu\alpha} \left[\int_{\odot} d\vec{S}_\nu^{\prime\alpha} \int_{\odot} d\vec{Z}_\nu^{\prime\alpha} \right] \prod_{\alpha} \left[\int d\hat{\lambda}_\alpha' \right] \\ & \times P(\mathbf{S}', \vec{Z}', \underline{\hat{\lambda}}', \vec{\sigma}) \prod_{\nu\alpha} \left(1 + \sigma_\nu^\alpha \sigma_\nu^{\alpha\prime} e^{-\frac{i\hat{\lambda}_\alpha + i\hat{\lambda}_\alpha'}{2}} (\vec{S}_\nu^\alpha \cdot \vec{Z}_\nu^{\prime\alpha} + \vec{Z}_\nu^\alpha \cdot \vec{S}_\nu^{\prime\alpha}) \right) \end{aligned} \tag{S91}$$

where D_k is given by

$$D_k = \prod_{\alpha} \left[\int \frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi} e^{iM\lambda_\alpha \hat{\lambda}_\alpha - \beta M \phi(\lambda_\alpha)} \right] \prod_{\nu\alpha} \left[\int_{\odot} d\vec{S}_\nu^\alpha \int_{\odot} d\vec{Z}_\nu^\alpha \right] \prod_{\nu\alpha} \left[\sum_{\sigma_\nu^{\alpha\prime}=0,1} \right]$$
$$\times [\hat{P}(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma})]^k e^{\sum_\nu \hat{Q}(\underline{S}_{\nu,a}, \underline{Z}_{\nu,a}, \underline{\hat{\lambda}}, \sigma_\nu^\alpha)} \prod_{\nu\alpha} \left(1 - \sigma_\nu^\alpha + \sigma_\nu^\alpha \vec{S}_\nu^\alpha \cdot \vec{Z}_\nu^\alpha \right) \quad \text{(S92)}$$

To solve the above saddle point equations, we employ an ansatz for $\hat{P}(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma})$ similar to the ansatz for $\hat{P}(\mathbf{S}, \underline{\hat{\lambda}})$ in Eq. (S23), given by

$$P(\mathbf{S}, \mathbf{Z}, \underline{\hat{\lambda}}, \vec{\sigma}) = \prod_{\alpha} \left[\int \frac{d\lambda_\alpha}{2\pi} e^{iM\lambda_\alpha \hat{\lambda}_\alpha - \beta M \phi(\lambda_\alpha)} \right] \int d\vec{h}\ w(\vec{h}) \quad \text{(S93)}$$
$$\times \prod_{\nu\alpha} \left[(1-\sigma_\nu^\alpha) h_0^\nu + \sigma_\nu^\alpha h_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}} (Z_{\nu,a}^\alpha)^2 S_{\nu,a}^\alpha + \sigma_\nu^\alpha \tilde{h}_1^\nu e^{-\frac{i\hat{\lambda}_\alpha}{2}} (S_{\nu,a}^\alpha)^2 Z_{\nu,a}^\alpha + \sigma_\nu^\alpha h_2^\nu e^{-i\hat{\lambda}_\alpha} (S_{\nu,a}^\alpha)^2 (Z_{\nu,a}^\alpha)^2 \right].$$

Similarly, this ansatz relates to the cavity approach [S1, S2], where $h_0^\nu, h_1^\nu, \tilde{h}_1^\nu, h_2^\nu$ resemble the cavity fields where communication ν does not pass through the node (h_0^ν), passes through the node via the *outgoing* cavity to the ancestor (h_1^ν), passes through the node via the *incoming* cavity to the ancestor ($\tilde{h}_1^\nu$), or passes the node without going through the cavity (h_2^ν), respectively. To derive the recursion relation involving only the cavity fields, one can first show that

$$e^{\hat{Q}(\underline{S}_{\nu,a}, \underline{Z}_{\nu,a}, \underline{\hat{\lambda}}, \sigma_\nu^\alpha)} \quad \text{(S94)}$$
$$= \sum_{\Lambda_\nu=-1}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{1}{N} \delta_{\Lambda_\nu,1} + \frac{1}{N} \delta_{\Lambda_\nu,-1} \right] \prod_{\nu\alpha} \left[\delta_{\Lambda_\nu,0} + \delta_{\Lambda_\nu,1} e^{-\frac{i\hat{\lambda}_\alpha}{2}} Z_{\nu,a}^\alpha + \delta_{\Lambda_\nu,-1} e^{-\frac{i\hat{\lambda}_\alpha}{2}} S_{\nu,a}^\alpha \right],$$

which is similar to Eq. (S41). Inserting Eqs. (S93) and (S94) into Eq. (S90), one can derive the following recursion relation in terms of cavity fields $h_0^\nu, h_1^\nu, \tilde{h}_1^\nu$ and h_2^ν (as in Eqs. (S45) to (S47)),

$$h_0^\nu = \delta_{\Lambda_\nu,0} \prod_{l=1}^{k-1} \left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \right] \quad \text{(S95)}$$

$$h_1^\nu = \delta_{\Lambda_\nu,0} \sum_{l=1}^{k-1} h_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \prod_{j\neq l} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)} \right] + \delta_{\Lambda_\nu,1} \prod_{l=1}^{k-1} \left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \right] \quad \text{(S96)}$$

$$\tilde{h}_1^\nu = \delta_{\Lambda_\nu,0} \sum_{l=1}^{k-1} \tilde{h}_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \prod_{j\neq l} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)} \right] + \delta_{\Lambda_\nu,-1} \prod_{l=1}^{k-1} \left[h_0^{\nu l} + h_2^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \right] \quad \text{(S97)}$$

$$h_2^\nu = \delta_{\Lambda_\nu,0} \sum_{(lr)} h_1^{\nu l} \tilde{h}_1^{\nu r} e^{-\beta\phi'(\lambda_{\nu l}^*) - \beta\phi'(\lambda_{\nu r}^*)} \prod_{j\neq l,r} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)} \right]$$
$$+ \delta_{\Lambda_\nu,1} \sum_{l=1}^{k-1} \tilde{h}_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \prod_{j\neq l} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)} \right] + \delta_{\Lambda_\nu,-1} \sum_{l=1}^{k-1} h_1^{\nu l} e^{-\beta\phi'(\lambda_{\nu l}^*)} \prod_{j\neq l} \left[h_0^{\nu j} + h_2^{\nu j} e^{-\beta\phi'(\lambda_{\nu j}^*)} \right]$$
$$\text{(S98)}$$

Following Eq. (S49), we denote

$$z^{\nu} = \frac{h_1^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}{h_0^{\nu} + h_2^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}, \qquad \tilde{z}^{\nu} = \frac{\tilde{h}_1^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}{h_0^{\nu} + h_2^{\nu} e^{-\beta\phi'(\lambda_\nu^*)}}, \tag{S99}$$

which lead to the following recursion relations, similar to Eq. (S52),

$$z^{\nu} = \delta_{\Lambda_\nu,0} \frac{e^{-\beta\phi'(\lambda_\nu^*)} \sum_{l=1}^{k-1} z_l^{\nu}}{1 + e^{-\beta\phi'(\lambda_\nu^*)} \sum_{(lr)}^{k-1} z_l^{\nu} \tilde{z}_r^{\nu}} + \delta_{\Lambda_\nu,1} \frac{1}{\sum_{l=1}^{k-1} \tilde{z}_l^{\nu}}, \tag{S100}$$

$$\tilde{z}^{\nu} = \delta_{\Lambda_\nu,0} \frac{e^{-\beta\phi'(\lambda_\nu^*)} \sum_{l=1}^{k-1} \tilde{z}_l^{\nu}}{1 + e^{-\beta\phi'(\lambda_\nu^*)} \sum_{(lr)}^{k-1} z_l^{\nu} \tilde{z}_r^{\nu}} + \delta_{\Lambda_\nu,-1} \frac{1}{\sum_{l=1}^{k-1} z_l^{\nu}}, \tag{S101}$$

such that $\lambda_\nu^* = f^{-1}(0)$ as in Eq. (S80), where $f(x)$ is given by

$$f_\nu(x) = x - \frac{1}{M} - \frac{1}{M} \sum_{\mu \neq \nu} \left\{ |\Lambda_\mu| + (1 - |\Lambda_\mu|) \frac{e^{-\beta\phi'(x)} \sum_{(lr)}^{k} z_l^{\mu} \tilde{z}_r^{\mu}}{1 + e^{-\beta\phi'(x)} \sum_{(lr)}^{k} z_l^{\mu} \tilde{z}_r^{\mu}} \right\}. \tag{S102}$$

To derive the corresponding optimization algorithm, we further substitute $z^{\nu} = e^{-\beta a^{\nu}}$ and $\tilde{z}^{\nu} = e^{-\beta b^{\nu}}$. In the zero-temperature limit, these give rise to the message passing algorithm of Eqs. (S103) to (S106) we are going to describe in Section S2 except that we have set the message $b_{j\to i}^{\nu} = 0$ when $\Lambda_j^{\nu} = -1$ to prevent indefinite looping of messages.

2 The algorithm

The analytic solution gives rise to a message passing algorithm, which identifies the path configuration that minimizes a given form of $\mathcal{H}$ on real instances. In contrast to the local tree structure assumed by the cavity approach, short loops are present in real instances which makes the direct algorithmic implementation of the cavity equation (S85) non-convergent. As the direct algorithmic interpretation does not distinguish between the two ends of a polymer, it happens that the iterative cavity equation does not converge since a shorter loop back to the *starting point* is preferred over one leading to the destination. We thus adopt the slightly modified formulation in Section S1.5 to identify one end of the polymer as the source and the other as the destination; this gives rise to a message passing algorithm the involves two messages a^{ν} and b^{ν}, which correspond to the cases where polymer ν passes via an *outgoing* and an *incoming* node-cavity en route to its ancestor, respectively. This also enables one to accommodate *directed polymers* or communications. The relation between the un-directed and directed formulation can be seen by the resemblance between Eq. (S85) and the zero-temperature limit of Eqs. (S100) and (S101) in the Section S1.5. To prevent infinite looping of messages, we set the message from the destination to zero which greatly improved convergence. From Eqs. (S100) and

(S101), the derived algorithm reads (Eq. (3) in the paper):

$$a_{j\to i}^{\nu} = \begin{cases} \min_{l\in\mathcal{L}_j\backslash\{i\}} \left[a_{l\to j}^{\nu}\right] - \min\left[-\phi'(\lambda_j^{\nu*}), \min_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}} \left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right], & \Lambda_j^{\nu} = 0 \\ -\min_{l\in\mathcal{L}_j\backslash\{i\}} \left[b_{l\to j}^{\nu}\right], & \Lambda_j^{\nu} = 1 \\ \infty, & \Lambda_j^{\nu} = -1 \end{cases} \tag{S103}$$

$$b_{j\to i}^{\nu} = \begin{cases} \min_{l\in\mathcal{L}_j\backslash\{i\}} \left[b_{l\to j}^{\nu}\right] - \min\left[-\phi'(\lambda_j^{\nu*}), \min_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}} \left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right], & \Lambda_j^{\nu} = 0 \\ \infty, & \Lambda_j^{\nu} = 1 \\ 0, & \Lambda_j^{\nu} = -1 \end{cases} \tag{S104}$$

where $\mathcal{L}_j$ denotes the neighboring nodes of j, $\Lambda_j^{\nu} = 1, -1, 0$ corresponds respectively to node j being the source, destination or otherwise. The value of $\lambda_j^{\nu*}$ is given by $\lambda_j^{\nu*} = g_{\nu}^{-1}(0)$ where $g_{\nu}(x)$ is given by Eq. (S102) (Eq.(4) of the manuscript)

$$g_{\nu}(x) = x - \frac{1}{M} - \frac{1}{M}\sum_{\mu\neq\nu}\left\{|\Lambda_{\mu}| + (1-|\Lambda_{\mu}|)\Theta\left(-\phi'(x) - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a_{l\to j}^{\mu} + b_{r\to j}^{\mu}\right]\right)\right\}, \tag{S105}$$

where $\Theta(x)$ is the step function with $\Theta(x) = 0, 0.5, 1$ for $x < 0, x = 0, x > 0$, respectively. The correspondence between the algorithm and the analytic solution is illustrated by comparing Fig. S2(a) and (b). Finally, after the messages converge for a real instance, we use an equation that resembles the definition of I in Eq. (S88) to obtain the optimal path configuration (Eq.(5) of the paper)

$$\sigma_j^{\nu} = |\Lambda_{\nu}| + (1-|\Lambda_{\nu}|)\Theta\left(-\phi'\left(\lambda_j^{\nu*}\right) - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right), \tag{S106}$$

where $\lambda_i^{\nu*} = g_{\nu}^{-1}(0)$ and $\sigma_i^{\nu} = 1$ if the communication ν passes i and otherwise $\sigma_i^{\nu} = 0$. The traffic I_i on node i is given by $I_i = \sum_{\nu}\sigma_i^{\nu}$.

We remark that when $\gamma = 1$ with the cost function defined by $\phi(x) = x^{\gamma}$, the above algorithm becomes exact and reduces to a shortest path algorithm. One can see this from Eqs. (S103) and (S104), when the cost function is $\phi(\lambda_j^{\nu*}) = \lambda_j^{\nu*}$ (i.e. $\gamma = 1$), the message passing of any individual communication ν becomes independent of all other communications. This is because the original interaction between communications is embedded in $\lambda_j^{\nu*}$ given by the solution of Eq. (S105); when $\gamma = 1$, $\phi'(\lambda_j^{\nu*}) = 1$ becomes independent of $\lambda_j^{\nu*}$, as well as of the other communications. In this case, the messages a^{ν} or b^{ν} increase by 1 unit (since $\phi'(\lambda_j^{\nu*}) = 1$) every time communication ν passes a node, effectively counting the distance from the source or sink of ν. One can further show that the algorithm is equivalent to a cavity algorithm which aims to find the shortest path.

2.1 Introducing quenched random bias

As discussed in the main article, the RSB-like behavior hinders algorithmic convergence in some instances. Convergence is improved by assigning a random bias ϵ_i to each node [S10], akin to an external field, guiding the system to one of the local minima. These biases can be easily incorporated in the present formalism by replacing $\phi(x)$ with $\phi_i(x)$ for each node i such that $\phi_i(x) = \phi(x) + x\epsilon_i$. While sub-optimal, the obtained local minima typically provide close-to-optimal solutions. Details of how RSB is examined in the present system are given in Section S5.

Cases of variable size communications, where a large number of unit source-destination pairs are identical, give rise to degeneracy among communications; to break the degeneracy in this case, brought by Eq. (S105), we replace ϵ_i by ϵ_i^ν for each communication ν. These give rise to a slight modification of Eqs. (103) - (106), where the messages a^ν and b^ν are updated by

$$a_{j\to i}^\nu = \begin{cases} \min\limits_{l\in\mathcal{L}_j\backslash\{i\}} \left[a_{l\to j}^\nu\right] - \min\left[-\phi'(\lambda_j^{\nu *}) - \epsilon_j^\nu, \min\limits_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}} \left[a_{l\to j}^\nu + b_{r\to j}^\nu\right]\right], & \Lambda_j^\nu = 0 \\ -\min\limits_{l\in\mathcal{L}_j\backslash\{i\}} \left[b_{l\to j}^\nu\right], & \Lambda_j^\nu = 1 \\ \infty, & \Lambda_j^\nu = -1 \end{cases} \tag{S107}$$

$$b_{j\to i}^\nu = \begin{cases} \min\limits_{l\in\mathcal{L}_j\backslash\{i\}} \left[b_{l\to j}^\nu\right] - \min\left[-\phi'(\lambda_j^{\nu *}) - \epsilon_j^\nu, \min\limits_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}} \left[a_{l\to j}^\nu + b_{r\to j}^\nu\right]\right], & \Lambda_j^\nu = 0 \\ \infty, & \Lambda_j^\nu = 1 \\ 0, & \Lambda_j^\nu = -1 \end{cases} \tag{S108}$$

and $\lambda_j^{\nu *}$ is given by $\lambda_j^{\nu *} = g_\nu^{-1}(0)$, such that $g_\nu(x)$ is given by

$$g_\nu(x) = x - \frac{1}{M} - \frac{1}{M}\sum_{\mu\neq\nu}\left\{|\Lambda_\mu| + (1-|\Lambda_\mu|)\Theta\left(-\phi'(x) - \epsilon_j^\mu - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a_{l\to j}^\mu + b_{r\to j}^\mu\right]\right)\right\}. \tag{S109}$$

The optimized state σ_j^ν of communication ν on node j is obtained by

$$\sigma_j^\nu = |\Lambda_\nu| + (1-|\Lambda_\nu|)\Theta\left(-\phi'\left(\lambda_j^{\nu *}\right) - \epsilon_j^\nu - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a_{l\to j}^\nu + b_{r\to j}^\nu\right]\right). \tag{S110}$$

We found that ϵ_i^ν with magnitude of the order $O[0.01 \cdot \phi'(1/M)]$ is usually sufficient to break the degeneracy in paths in the presence of identical source-destination pairs. In cases where a large number of source-destination pairs are identical, larger ϵ_i^ν would be beneficial.

2.2 Computational complexity

As mentioned in the paper, the computation complexity of the algorithm where costs are defined on vertices is $O\left(N(M + M^2\langle I\rangle)\right)$. The factor N comes from the passing of messages among the N nodes until convergence, while the factor M comes from the update of messages a and b in Eqs. (103) and (104), and the factor M^2 comes from the computation of $\lambda_i^{\nu *}$ for each of the M communications, each of which involves a summation of M; and finally, the average flow factor $\langle I\rangle$ comes from solving Eq. (S105) by setting $x = I/M$ and increasing the integer I from $I = 0$ until a solution x for $f_\nu(x) = 0$ is found. We remark that if costs are defined on edges only, the algorithm has a much lower complexity $O(NM)$ making it more suitable for very large systems. We refer readers to Section S3.2 for details.

To reduce the computational complexity of the algorithm with vertex cost, we note that for a node j, all communications ν with $-\phi'(1/M) - \min_{l,r\in\mathcal{L}_j|l\neq r}\left[a^\nu_{l\to j} + b^\nu_{r\to j}\right] < 0$ do not contribute to the summation over $\mu \neq \nu$ in Eq. (S105). In other words, by denoting the set of all the other communications as $\mathcal{M}$, the summation in Eq. (S105) runs over the set $\mathcal{M}\backslash\{\nu\}$. We also note that all communications $\nu \notin \mathcal{M}$ have the same $\lambda_i^{\nu *}$ given by $\lambda_i^{\nu *} = g^{-1}(0)$ where $g(x)$ is given by

$$g(x) = x - \frac{1}{M} - \frac{1}{M}\sum_{\mu\in\mathcal{M}}\left\{|\Lambda_\mu| + (1-|\Lambda_\mu|)\Theta\left(-\phi'(x) - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a^\mu_{l\to j} + b^\mu_{r\to j}\right]\right)\right\}. \tag{S111}$$

This reduces the computational complexity for obtaining $\lambda_i^{\nu *}$ to $\langle I\rangle^2$, and thus reduces the complexity of the algorithm to $O(N(M + \langle I\rangle^3))$. This can also be used to solve Eq. (S86) for obtaining an analytic solution. A further simplification is noted in the cases of convex $\phi(x)$, since the summations in both $g_\nu(x)$ and $g(x)$ (Eqs. (S105) and (S111), respectively) are decreasing functions of x, and thus the solutions of $g_\nu^{-1}(0)$ and $g^{-1}(0)$ differ at most by $1/M$.

Another way to reduce computational complexity is to adopt a decimation procedure similar to [S8, S9], i.e. fix a communication path and thus the messages a and b, which are already biased towards a particular choice of routes. Detailed results are shown in the next subsection.

2.3 Convergence

We ran the algorithm (S107) to (S110) on random regular graphs with various values of N and M values, and show in Fig. S3(a) the fraction of instances where the algorithm converges; we also monitor the validity of solutions obtained in cases when the algorithm does not converge. As we can see, the convergence ratio slightly decreases with increasing M and N. As suggested by the higher fraction of valid solutions compared with the convergence ratio, when the iteration procedure is terminated, the algorithm is able to establish complete paths for all communications in many instances even for which it does not converge. Indeed, the algorithm finds a valid solution for more than 80% of the instances even at large M and N without decimation. The results shown in Fig. S3 are obtained with $\gamma = 2$ in $\phi(x) = x^\gamma$ whereas the cases with $\gamma = 0.5$ show similar behaviors but a higher fraction of convergence and validity.

In Fig. S3(b), we show that the average convergence time increases for small M values and becomes saturated at large M for various values of N, and increases slightly when N increases from $N = 100$ to $N = 1000$. By employing a decimation procedure to fix communications that are strongly biased towards a particular path, a much faster convergence is obtained as shown by the dashed lines in Fig. S3(b). We remark that (i) all obtained solutions are valid when decimation is added to the algorithm, and (ii) the trade-off in energy increase (with respect to the optimal solution) is smaller than 1% for almost all cases shown in Fig. S3(b). For graphs with a large number of short loops, convergence is more difficult and decimation is useful. Nevertheless, most of our results were obtained without the decimation procedure, with the iterations terminated after messages converge or if the state of all communication paths stabilizes for a certain number of steps.

In Fig. S3(c), we show that convergence is greatly improved by running the algorithm (S129) to (S133) derived in Section S3.2 where the cost is defined on edges only. A similar picture is obtained when separate costs are defined on each direction of the edges, where optimal solutions are obtained by the algorithm defined by equations (S134) to (S138) derived in Section S3.3. We remark that these two algorithms also have a much lower computational complexity as discussed in Section S3.2 and 3.3.

2.4 Comparison with the conventional cavity approach

In this section, we apply the cavity approach directly to derive an algorithm without the backward messages from nodes uplink and compare the results obtained to those obtained via our algorithm, where backward messages originate from the approximation of Eq. (S43). As will be shown below, the conventional cavity method involves 3^M states, but its performance is matched by the simplified approach involving only $2M$ states. This justifies the approximation of Eq. (S43). Since the conventional cavity approach involves 3^M states, we can only compare the algorithmic results up to a small value of M, e.g. $M \approx 7$, due to the prohibitive computational cost, in both time and memory.

To write the conventional cavity equation, we first denote $\sigma^\nu_{j\to i} = +1, 0, -1$ to represent the cases where communication ν goes from node j to i, does not go through the edge between i and j, and goes from i to j, respectively. We denote the vector of $\sigma^\nu_{j\to i}$ over the index ν by a vector $\vec{\sigma}_{j\to i}$ of 3^M states. We then define the cavity energy $E_{j\to i}(\vec{\sigma}_{j\to i})$ to be the *optimized* energy of the tree terminated at node j without the edge connecting it to i, as a function of the state $\vec{\sigma}_{j\to i}$ at j. One can then write a recursion equation relating $E_{j\to i}(\vec{\sigma}_{j\to i})$ to the neighboring nodes cavity energies $E_{k\to j}(\vec{\sigma}_{k\to j})$ as

$$E_{j\to i}(\vec{\sigma}_{j\to i}) = \min_{\{\{\vec{\sigma}_{k\to j}\}|\vec{\Lambda}_j - \vec{\sigma}_{j\to i} + \sum_{k\in\mathcal{L}_j\backslash\{i\}} \vec{\sigma}_{k\to j} = \vec{0}\}} \left[\phi(I_j) + \sum_{k\in\mathcal{L}_j\backslash\{i\}} E_{k\to j}(\vec{\sigma}_{k\to j}) \right], \quad \text{(S112)}$$

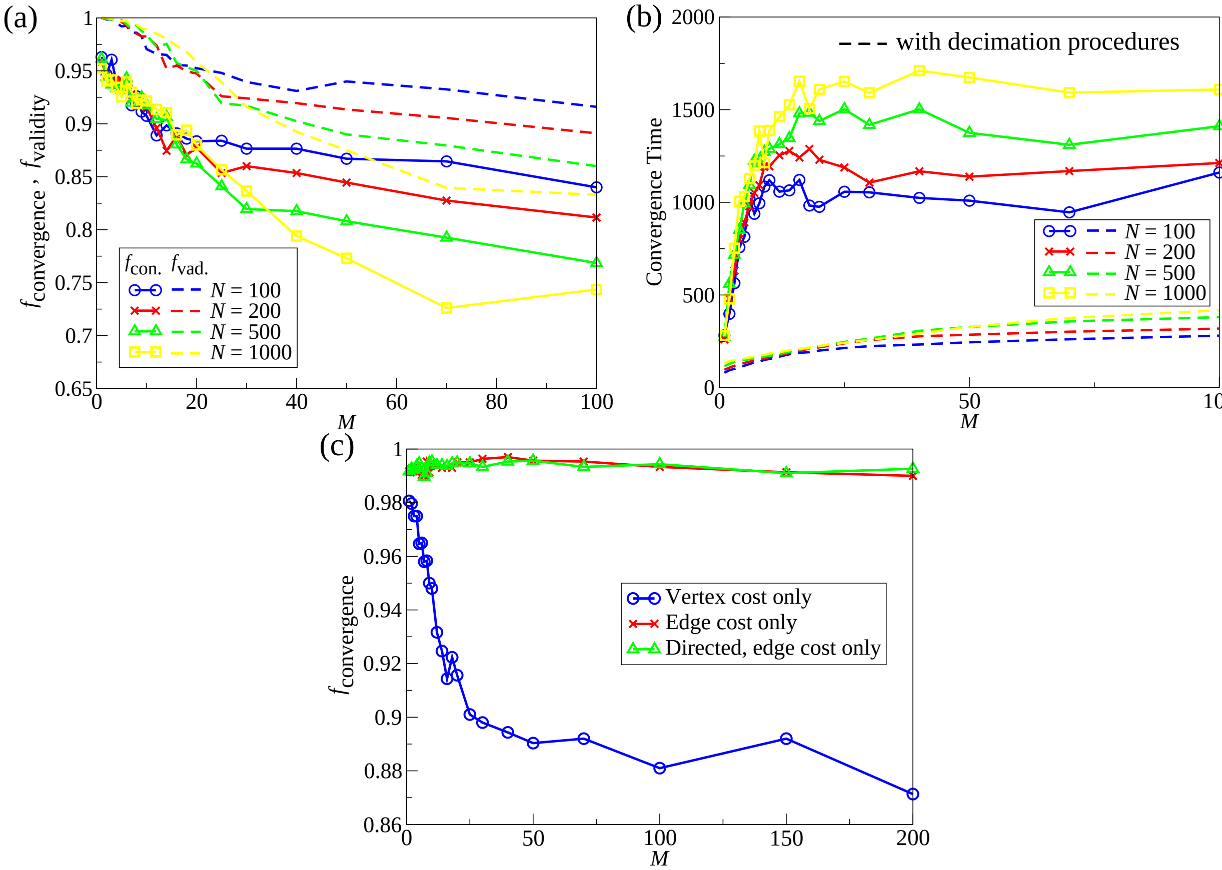

Fig. S 3: (a) The fraction of convergent (solid lines with symbols) and valid solution for instances (dashed lines) obtained by the algorithm (S107) to (S110) (without decimation) within $10^4 \times N$ iterations on random regular graphs of average degree $\langle k \rangle = 3$ of size $N = 100, 200, 500$ and 1000, averaged over 2000 instances. *Valid solutions for instances* are defined as instances where complete paths are established for all communications when the iteration terminates. (b) Average convergence times with and without decimation, represented by dashed lines and solid lines with symbols, respectively. Solutions for all instances are valid when decimation is added to the algorithm. (c) The fraction of convergent instances where costs are defined on vertices (algorithm (S107) to (S110)), on edges only (algorithm (S129) to (S133) with random bias), and where separate costs are defined on each direction of the edges (algorithm (S134) to (S138) with random bias).

where $\vec{\Lambda}_j$ denotes the vector $(\Lambda_j^1, \cdots, \Lambda_j^M)$ with each $\Lambda_j^\nu = \{\pm 1, 0\}$ defined as in Eqs. (S103) and (S104), and $I_j = \sum_{\nu=1}^M \left[|\Lambda_j^\nu| + (1 - |\Lambda_j^\nu|)\Theta(|\sigma_{j\to i}^\nu| + \sum_{k\in\mathcal{L}_j\setminus\{i\}} |\sigma_{k\to j}^\nu|) \right]$ the number of communications/polymers which pass through node j; $\Theta(x) = 0$ with $x \le 0$ and otherwise $\Theta(x) = 1$.

One can now optimize real instances by passing messages consisting of the cavity energy function $E_{j\to i}(\vec{\sigma}_{j\to i})$ until convergence, and evaluate the optimized state $\vec{\sigma}_j^*$ of node j by

$$\sigma_j^{\nu *} = |\Lambda_j^\nu| + (1 - |\Lambda_j^\nu|)\Theta\left(\sum_{k\in\mathcal{L}_j\setminus\{i\}} |\sigma_{k\to j}^{\nu *}| \right) \tag{S113}$$

with $\sigma_{k\to j}^{\nu *}$ obtained by

$$\vec{\sigma}_{k\to j}^* = \underset{\{\{\vec{\sigma}_{k\to j}\} | \vec{\Lambda}_j + \sum_{k\in\mathcal{L}_j} \vec{\sigma}_{k\to j} = \vec{0}\}}{\text{argmin}} \left[\sum_{k\in\mathcal{L}_j} E_{k\to j}(\vec{\sigma}_{k\to j}) \right]. \tag{S114}$$

Equation (S112)-(S114) constitute an algorithm, which we refer to as the *conventional cavity algorithm*, for solving the system of interacting polymers/communications. In general, this algorithm does not always find optimal solutions due to the presence of degenerate paths. One can then adopt the same method as in our algorithm and introduce a small random bias ϵ_i on each node, akin to an external field, which breaks the path degeneracy. Moreover, due to the presence of short loops or possibly the influence of RSB-like behavior in finite systems, the conventional cavity algorithm does not always converge. In our algorithm we found that the influence of short loops can be reduced by setting messages from destinations to be zero (i.e. the last message in Eq. (S104)); this contributes to a much improved convergence rate. In the present case of Eq. (S112) we do not have a trivial way to implement a similar trick. Nevertheless, the state of communication paths usually stabilizes after a number of iteration, resulting in convergence of a (sub-)optimal state in spite the non-converging messages.

As mentioned before, the 3^M-dimension of the cavity energy $E_{j\to i}(\vec{\sigma}_{j\to i})$ limits the usefulness of the conventional cavity algorithm. In addition, the constrained integer optimization of Eq. (S112) requires an exhaustive search in state space, implying an $O((3^M)^{k_j-1})$ operations for each iteration on node j with degree k_j. Due to these limitations, we have only tested the conventional cavity algorithm on random regular networks with $k = 3$.

We compare results obtained by our algorithm with those obtained by the conventional cavity algorithm by running them on instances of identical quenched disorders (i.e. identical topology and source-destination communicating pairs). The corresponding optimized energy E is shown in Fig. S4. The difference in optimized energy is extremely small and of the order of less than 1%. Remarkably, our algorithm performs slightly better than the conventional cavity algorithm in most of the cases. We remark that such a small differences may not be representative due to the small number of communications; it may result from the influence of small loops or RSB-like behavior, such that different local minima have

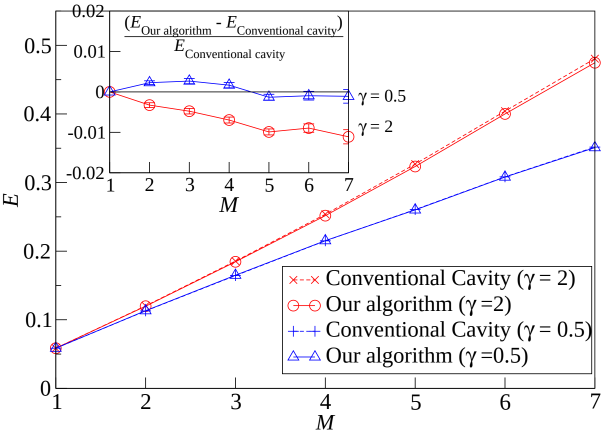

Fig. S 4: The optimized energy $E = \frac{1}{N}\sum_i I_i^\gamma$ obtained by the conventional cavity algorithm in Eqs (S112)-(S114) compared to that obtained by our approximate algorithm using Eqs. (S103)-(S106). Simulations are conducted by running the two algorithms on 2000 pairs of identical instances of random regular graphs with $N = 100$ and $k = 3$, from $M = 1$ to $M = 5$, 1000 pairs for $M = 6$ and 400 pairs for $M = 7$. Error bars are smaller than symbol size for all curves. Inset: the percentage difference of the optimized energy.

slightly different energies. In general, we found that running any of the algorithms separately on the same instance but with different update sequences may result in different optimized states with a slight difference in energy (despite these variations, we show in Sec. S4 that our algorithm always provides a lower bound of the optimized energy with respect to the other congestion-aware algorithm we tested). Nevertheless, the results suggest that the difference between our algorithm and the conventional cavity algorithm is negligible while our algorithm is much more efficient computationally with much shorter computing time, better convergence and easier implementation. These results also support the approximation Eq. (S43) used in deriving the algorithm.

2.5 Discrepancy between simulations and analytic solution

To validate the ansätze used we examine the agreement between analytical and simulation results.

For the Hamiltonian $\mathcal{H} \propto \sum_i I_i^\gamma$, i.e. $\phi(x) = x^\gamma$, we found a good agreement of average energy $\langle E\rangle$ and path length $\langle L\rangle$ between simulations and analytical solution is good for $\gamma \geq 1$ and improving with the increase in system size. For $\gamma \leq 1$, larger discrepancy in $\langle E\rangle$ and $\langle L\rangle$ are observed for larger N and M and small γ. Results presented in Fig. S5, show good agreement between simulations and the analytical predictions of $\langle L\rangle$ for all N and M with $\gamma = 2$, and for all M at $N = 100$ with $\gamma = 0.5$, but discrepancy starts to appear at $M > 100$ with $N = 200$ and $N = 500$ with $\gamma = 0.5$. In this case the simulation results still look reasonable, but the analytical results show an abrupt increase at $M > 100$. A possible reason for this discrepancy is that for small γ, the flow pattern is tree-like (as mentioned

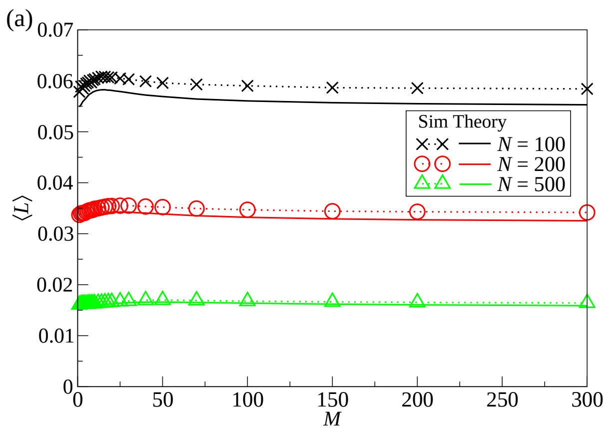

Fig. S5: The analytical and the simulation results of average path length $\langle L\rangle$ as a function of M in the optimized solution of $\mathcal{H} \propto \sum_i I_i^\gamma$ on random regular graphs, with (a) $\gamma = 2$ and (b) $\gamma = 0.5$.

in the paper), and when N and M are large the inaccuracies due to the approximation made influence the collective choices of paths.

3 Generalized formulation and algorithms

In this section, we generalize the previous formulation to derive algorithms which accommodate other communication scenarios: when communications are weighted, costs are defined on edges, and separate costs are defined on each direction of edges. The resulting algorithms for the various cases were tested on random regular graphs and yield favorable results. In particular, as we will see in the following subsections, the computation complexity of the algorithm is greatly reduced when costs are defined on edges only. We remark that better convergence is also obtained for these algorithms as discussed in Section S2.3.

3.1 The algorithm for weighted communications

We first derive the algorithm when communications are weighted. This is in particular relevant when a large number communications share the same source and destination and can be grouped during transmission and in cases where different communications have different corresponding priorities. More specifically, we define w_ν to be the weight of communication ν such that the weighted sum of traffic on node i is given by $I_i = \sum_\nu w_\nu \sigma_i^\nu$, where $\sigma_i^\nu = 1$ when communication ν passes node i and $\sigma_i^\nu = 0$ otherwise. We can then write down the partition function to accommodates *weighted* communications, given by

$$\mathcal{Z} = \prod_{i=1}^{N}\left[\int \frac{d\lambda_i d\hat{\lambda}_i}{2\pi} e^{iM\lambda_i\hat{\lambda}_i - \beta M\phi(\lambda_i)}\right] \prod_{i\nu}\left[\frac{1}{C_n}\int_{\odot} d\vec{S}_i^{\nu}\right] \\ \times \prod_{\nu=1}^{M}\left[\prod_{(ij)}\left(e^{-w_\nu \frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}} S_{i,a}^{\nu} S_{j,a}^{\nu}\right)^{\Lambda_{(ij)}^{\nu}} \prod_{(kl)}\left(1 + A_{kl} e^{-w_\nu \frac{i\hat{\lambda}_k + i\hat{\lambda}_l}{2}} \vec{S}_k^{\nu} \cdot \vec{S}_l^{\nu}\right)\right], \tag{S115}$$

which is identical to Eq. (S2) except the presence of weight factors w_ν in the exponentials $e^{-w_\nu \frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}}$ and $e^{-w_\nu \frac{i\hat{\lambda}_k + i\hat{\lambda}_l}{2}}$. The integration of $\hat{\lambda}_i$ leads to $\lambda_i = \sum_\nu w_\nu \sigma_i^\nu / M$; λ_i is thus the weighted sum of traffic through node i. One can then follow the same derivations as in Section S1 to arrive at the corresponding Eqs. (S85) and (S87) given by

$$u^{\nu} = (1 - \Lambda_\nu)\left\{\min_{l \le k-1}[u_l^{\nu}] - \min\left[-w_\nu \phi'(\lambda_\nu^*), \min_{\substack{l,r \le k-1 \\ l \ne r}}[u_l^{\nu} + u_r^{\nu}]\right]\right\} - \Lambda_\nu \min_{l \le k-1}[u_l^{\nu}], \tag{S116}$$

$$f_\nu(x) = x - \frac{w_\nu}{M} - \sum_{\mu \ne \nu} \frac{w_\mu}{M}\left\{\Lambda_\mu + (1 - \Lambda_\mu)\Theta\left(-w_\mu \phi'(x) - \min_{\substack{l,r \le k \\ l \ne r}}[u_l^{\mu} + u_r^{\mu}]\right)\right\}. \tag{S117}$$

By similar arguments suggested in the Section S1.5, the above equations give rise to an algorithm capable of optimizing weighted communications,

$$a_{j\to i}^{\nu} = \begin{cases} \min\limits_{l \in \mathcal{L}_j \setminus \{i\}}\left[a_{l\to j}^{\nu}\right] - \min\left[-w_\nu \phi'(\lambda_j^{\nu *}), \min\limits_{\substack{l,r \in \mathcal{L}_j \setminus \{i\} \\ l \ne r}}\left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right], & \Lambda_j^{\nu} = 0 \\ -\min\limits_{l \in \mathcal{L}_j \setminus \{i\}}\left[b_{l\to j}^{\nu}\right], & \Lambda_j^{\nu} = 1 \\ \infty, & \Lambda_j^{\nu} = -1 \end{cases} \tag{S118}$$

$$b_{j\to i}^{\nu} = \begin{cases} \min\limits_{l \in \mathcal{L}_j \setminus \{i\}}\left[b_{l\to j}^{\nu}\right] - \min\left[-w_\nu \phi'(\lambda_j^{\nu *}), \min\limits_{\substack{l,r \in \mathcal{L}_j \setminus \{i\} \\ l \ne r}}\left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right], & \Lambda_j^{\nu} = 0 \\ \infty, & \Lambda_j^{\nu} = 1 \\ 0, & \Lambda_j^{\nu} = -1 \end{cases} \tag{S119}$$

such that the value of $\lambda_j^{\nu *}$ is given by $\lambda_j^{\nu *} = g_\nu^{-1}(0)$ where $g_\nu(x)$ is defined by

$$g_\nu(x) = x - \frac{w_\nu}{M} - \sum_{\mu \ne \nu} \frac{w_\mu}{M}\left\{|\Lambda_\mu| + (1 - |\Lambda_\mu|)\Theta\left(-w_\mu \phi'(x) - \min_{\substack{l,r \in \mathcal{L}_j \\ l \ne r}}\left[a_{l\to j}^{\mu} + b_{r\to j}^{\mu}\right]\right)\right\}, \tag{S120}$$

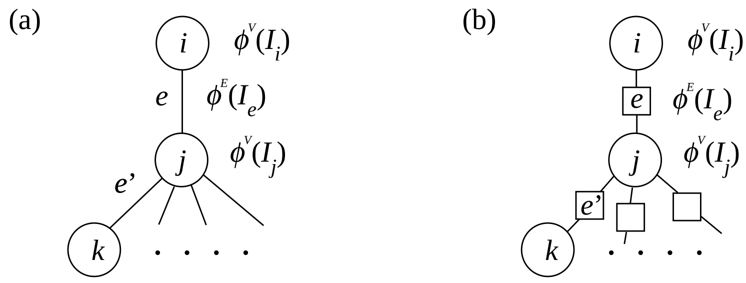

Fig. S 6: (a) The graph with cost $\phi^V(I)$ defined on nodes and $\phi^E(I)$ defined on edges, and (b) the factor graph representation of the corresponding network where edges are represented square factor nodes.

and the optimized state is given by

$$\sigma_j^\nu = |\Lambda_\nu| + (1-|\Lambda_\nu|)\Theta\left(-w_\nu \phi'\left(\lambda_j^{\nu *}\right) - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a_{l\to j}^\nu + b_{r\to j}^\nu\right]\right). \tag{S121}$$

One can see that when $w_\nu = 1$ for all ν, the above algorithm reduces to Eqs. (S103)-(S106). To break path degeneracy, small quenched biases ϵ_i can be introduced as in Section S2.1 by replacing $\phi(x)$ with $\phi_i(x) + x\epsilon_i$.

3.2 The algorithm for costs defined on edges

We will now derive the algorithm when costs are defined on both vertices and edges. As shown in Fig. S6(a), we denote $\phi_V(I)$ and $\phi_E(I)$ as the cost function on vertices and edges respectively, and represent the network as a factor graph by considering edges as *factor nodes* as shown in Fig. S6(b). We can then write the corresponding partition function where edges are labeled by e, given by

$$\begin{aligned}\mathcal{Z} =& \prod_{i=1}^{N}\left[\int \frac{d\lambda_i d\hat{\lambda}_i}{2\pi} e^{iM\lambda_i\hat{\lambda}_i - \beta M\phi_V(\lambda_i)}\right]\prod_{i\nu}\left[\frac{1}{C_n}\int_{\odot} d\vec{S}_i^\nu\right]\\ &\times\prod_{e=1}^{E}\left[\int \frac{d\lambda_e d\hat{\lambda}_e}{2\pi} e^{iM\lambda_e\hat{\lambda}_e - \beta M\phi_E(\lambda_e)}\right]\prod_{e\nu}\left[\frac{1}{C_n}\int_{\odot} d\vec{S}_e^\nu\right]\\ &\times\prod_{\nu=1}^{M}\left[\prod_{(ij)}\left(e^{-\frac{i\hat{\lambda}_i + i\hat{\lambda}_j}{2}} S_{i,a}^\nu S_{j,a}^\nu\right)^{\Lambda_{(ij)}^\nu}\prod_{(ke)}\left(1 + A_{ke}e^{-\frac{i\hat{\lambda}_k + i\hat{\lambda}_e}{2}}\vec{S}_k^\nu\cdot\vec{S}_e^\nu\right)\right].\end{aligned} \tag{S122}$$

One can then proceed with the calculation following Section S1.

Alternatively, we can use Fig. S6(b) and Eqs. (S103)-(S106) to derive the corresponding algorithm. We first write the messages from a vertex j to an edge node e, which are identical to the previous case and are given by

Messages from a vertex to an edge

$$a^{\nu}_{j\to e}=\begin{cases}\min\limits_{e'\in\mathcal{L}_j\backslash\{i\}}\left[a^{\nu}_{e'\to j}\right]-\min\left[-\phi'_V(\lambda^{\nu *}_j),\min\limits_{\substack{e',e''\in\mathcal{L}_j\backslash\{i\}\\ e'\neq e''}}\left[a^{\nu}_{e'\to j}+b^{\nu}_{e''\to j}\right]\right], & \Lambda^{\nu}_j=0\\ -\min\limits_{e'\in\mathcal{L}_j\backslash\{i\}}\left[b^{\nu}_{e'\to j}\right], & \Lambda^{\nu}_j=1\\ \infty, & \Lambda^{\nu}_j=-1\end{cases}\tag{S123}$$

$$b^{\nu}_{j\to e}=\begin{cases}\min\limits_{e'\in\mathcal{L}_j\backslash\{i\}}\left[b^{\nu}_{e'\to j}\right]-\min\left[-\phi'_V(\lambda^{\nu *}_j),\min\limits_{\substack{e',e''\in\mathcal{L}_j\backslash\{i\}\\ e'\neq e''}}\left[a^{\nu}_{e'\to j}+b^{\nu}_{e''\to j}\right]\right], & \Lambda^{\nu}_j=0\\ \infty, & \Lambda^{\nu}_j=1\\ 0, & \Lambda^{\nu}_j=-1\end{cases}\tag{S124}$$

such that the value of $\lambda^{\nu *}_j$ is given by $\lambda^{\nu *}_j=g^{-1}_{\nu}(0)$ where $g_{\nu}(x)$ is

$$g_{\nu}(x)=x-\frac{1}{M}-\frac{1}{M}\sum_{\mu\neq\nu}\left\{|\Lambda_{\mu}|+(1-|\Lambda_{\mu}|)\Theta\left(-\phi'_V(x)-\min_{\substack{e',e''\in\mathcal{L}_j\\ e'\neq e''}}\left[a^{\mu}_{e'\to j}+b^{\mu}_{e''\to j}\right]\right)\right\}.\tag{S125}$$

We can then write the messages from an edge node to a vertex by using Eqs. (S103)-(S106). Since all edge nodes are of degree two and they are neither source nor destination, the messages from edge node e to node i in Fig. S6(b) are given by

Messages from an edge to a vertex

$$a^{\nu}_{e\to i}=a^{\nu}_{j\to e}+\phi'_E(\eta^{\nu *}_e)\tag{S126}$$

$$b^{\nu}_{e\to i}=b^{\nu}_{j\to e}+\phi'_E(\eta^{\nu *}_e)\tag{S127}$$

such that $\eta^{\nu *}_e$ is given by $\eta^{\nu *}_e=h^{-1}_{\nu}(0)$ where $h_{\nu}(x)$ is

$$h_{\nu}(x)=x-\frac{1}{M}-\frac{1}{M}\sum_{\mu\neq\nu}\Theta\left(-\phi'_E(x)-\min\left[a^{\mu}_{j\to e}+b^{\mu}_{i\to e},a^{\mu}_{i\to e}+b^{\mu}_{j\to e}\right]\right).\tag{S128}$$

Indeed, one can use the above relations to devise a message passing algorithm involving vertices only. We let $a^{\nu}_{j\to i}=a^{\nu}_{e\to i}$ and $b^{\nu}_{j\to i}=b^{\nu}_{e\to i}$ such that Eqs. (S123), (S124), (S126) and (S127) combine to become

<u>*Combined message passing which involves vertices only*</u>

$$a^{\nu}_{j\to i} = \begin{cases} \min\limits_{l\in\mathcal{L}_j\backslash\{i\}}\left[a^{\nu}_{l\to j}\right] + \phi'_E(\eta^{\nu*}_{ji}) - \min\left[-\phi'_V(\lambda^{\nu*}_j), \min\limits_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}}\left[a^{\nu}_{l\to j} + b^{\nu}_{r\to j}\right]\right], & \Lambda^{\nu}_j = 0 \\ -\min\limits_{l\in\mathcal{L}_j\backslash\{i\}}\left[b^{\nu}_{l\to j}\right] + \phi'_E(\eta^{\nu*}_{ji}), & \Lambda^{\nu}_j = 1 \\ \infty, & \Lambda^{\nu}_j = -1 \end{cases} \tag{S129}$$

$$b^{\nu}_{j\to i} = \begin{cases} \min\limits_{l\in\mathcal{L}_j\backslash\{i\}}\left[b^{\nu}_{l\to j}\right] + \phi'_E(\eta^{\nu*}_{ji}) - \min\left[-\phi'_V(\lambda^{\nu*}_j), \min\limits_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}}\left[a^{\nu}_{l\to j} + b^{\nu}_{r\to j}\right]\right], & \Lambda^{\nu}_j = 0 \\ \infty, & \Lambda^{\nu}_j = 1 \\ \phi'_E(\eta^{\nu*}_{ji}), & \Lambda^{\nu}_j = -1 \end{cases} \tag{S130}$$

such that the value of $\lambda^{\nu*}_j$ is given by $\lambda^{\nu*}_j = g^{-1}_{\nu}(0)$ where $g_{\nu}(x)$ is

$$g_{\nu}(x) = x - \frac{1}{M} - \frac{1}{M}\sum_{\mu\neq\nu}\left\{|\Lambda_{\mu}| + (1-|\Lambda_{\mu}|)\Theta\left(-\phi'(x) - \min_{\substack{l,r\in\mathcal{L}_j\\ l\neq r}}\left[a^{\mu}_{l\to j} + b^{\mu}_{r\to j}\right]\right)\right\}. \tag{S131}$$

The value of $\eta^{\nu*}_{ji}$ is given by

$$\eta^{\nu*}_{ji} = \frac{1}{M} + \sum_{\mu\neq\nu}\sigma^{\mu}_{ji}, \tag{S132}$$

such that σ^{ν}_{ji} is given by

$$\begin{aligned}\sigma^{\nu}_{ji} = {} & \delta_{\Lambda_\nu,1}\Theta\left(\min_{l\in\mathcal{L}_j\backslash\{i\}}\left[b^{\nu}_{l\to j}\right] - b^{\nu}_{i\to j}\right) + \delta_{\Lambda_\nu,-1}\Theta\left(\min_{l\in\mathcal{L}_j\backslash\{i\}}\left[a^{\nu}_{l\to j}\right] - a^{\nu}_{i\to j}\right) \\ & + \delta_{\Lambda_\nu,0}\Theta\left(\min\left[-\phi'_V(\lambda^{\nu*}_j), \min_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}}\left[a^{\nu}_{l\to j} + b^{\nu}_{r\to j}\right]\right]\right. \\ & \left. - \min\left[a^{\nu}_{i\to j} + \min_{l\in\mathcal{L}_j\backslash\{i\}}\left[b^{\nu}_{l\to j}\right], b^{\nu}_{i\to j} + \min_{l\in\mathcal{L}_j\backslash\{i\}}\left[a^{\nu}_{l\to j}\right]\right]\right),\end{aligned} \tag{S133}$$

which also corresponds to the optimized state of the edge (ij) after the convergence of messages: $\sigma^{\nu}_{ji} = 1$ if the communication ν passes the edge and $\sigma^{\nu}_{ji} = 0$ otherwise. We remark that when no cost is defined on vertices, i.e. $\phi_V(x) = 0$, the computation of $\lambda^{\nu*}_j$ by Eq. (S131) can be omitted. The computation of $\eta^{\nu*}_{ji}$ for all ν in Eq. (S132) involves only

$O(M)$ operations since one can first compute the sum $\sum_{\mu=1}^{M} \sigma_{ji}^{\mu}$ and then compute each $\eta_{ji}^{\nu*}$ by $\eta_{ji}^{\nu*} = (\sum_{\mu=1}^{M} \sigma_{ji}^{\mu}) - \sigma_{ji}^{\nu} + 1/M$; the computational complexity of the algorithm is thus greatly reduced from $O(N(M+M^2\langle I\rangle)$ to $O(NM)$.The optimized path configurations obtained by this algorithm for the London subway data are shown in Fig. S7.

3.3 The algorithm for separate costs defined on each direction of edges

Finally, we derive the algorithm when separate costs are defined on each direction of the edges. This is in particular relevant to transportation networks where congestion may appear only in one direction. Since traffic direction is in general not well-defined on vertices, we will derive the algorithm assuming only edge costs are present, i.e. $\mathcal{H} = \sum_{(ij)}[\phi_E(I_{i\to j}/M) + \phi_E(I_{j\to i}/M)]$ where $I_{i\to j}$ and $I_{j\to i}$ correspond to the traffic from i towards j, and from j towards i, respectively. As discussed in Section S1.5 and the paragraph above Eq. (S103), our algorithm already accommodates directed communications; we can then write an algorithm for two-way traffic based on Eqs. (S129) and (S130) assuming a vertex cost function $\phi_V(x) = 0$

$$a_{j\to i}^{\nu} = \begin{cases} \min\limits_{l\in\mathcal{L}_j\backslash\{i\}} \left[a_{l\to j}^{\nu}\right] + \phi_E'(\eta_{j\to i}^{\nu*}) - \min\left[0, \min\limits_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}} \left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right], & \Lambda_j^{\nu} = 0 \\ -\min\limits_{l\in\mathcal{L}_j\backslash\{i\}} \left[b_{l\to j}^{\nu}\right] + \phi_E'(\eta_{j\to i}^{\nu*}), & \Lambda_j^{\nu} = 1 \\ \infty, & \Lambda_j^{\nu} = -1 \end{cases} \tag{S134}$$

$$b_{j\to i}^{\nu} = \begin{cases} \min\limits_{l\in\mathcal{L}_j\backslash\{i\}} \left[b_{l\to j}^{\nu}\right] + \phi_E'(\eta_{i\to j}^{\nu*}) - \min\left[0, \min\limits_{\substack{l,r\in\mathcal{L}_j\backslash\{i\}\\ l\neq r}} \left[a_{l\to j}^{\nu} + b_{r\to j}^{\nu}\right]\right], & \Lambda_j^{\nu} = 0 \\ \infty, & \Lambda_j^{\nu} = 1 \\ \phi_E'(\eta_{i\to j}^{\nu*}), & \Lambda_j^{\nu} = -1 \end{cases} \tag{S135}$$

where $a_{j\to i}^{\nu}$ and $b_{j\to i}^{\nu}$ correspond to the energy when communication ν passes from j towards i and from i towards j, respectively. The variables $\eta_{j\to i}^{\nu*}$ and $\eta_{i\to j}^{\nu*}$ correspond to the normalized traffic from j towards i and from i towards j, respectively; they are given by

$$\eta_{j\to i}^{\nu*} = \frac{1}{M} + \sum_{\mu\neq\nu} \sigma_{j\to i}^{\mu}, \qquad \eta_{i\to j}^{\nu*} = \frac{1}{M} + \sum_{\mu\neq\nu} \sigma_{i\to j}^{\mu}, \tag{S136}$$

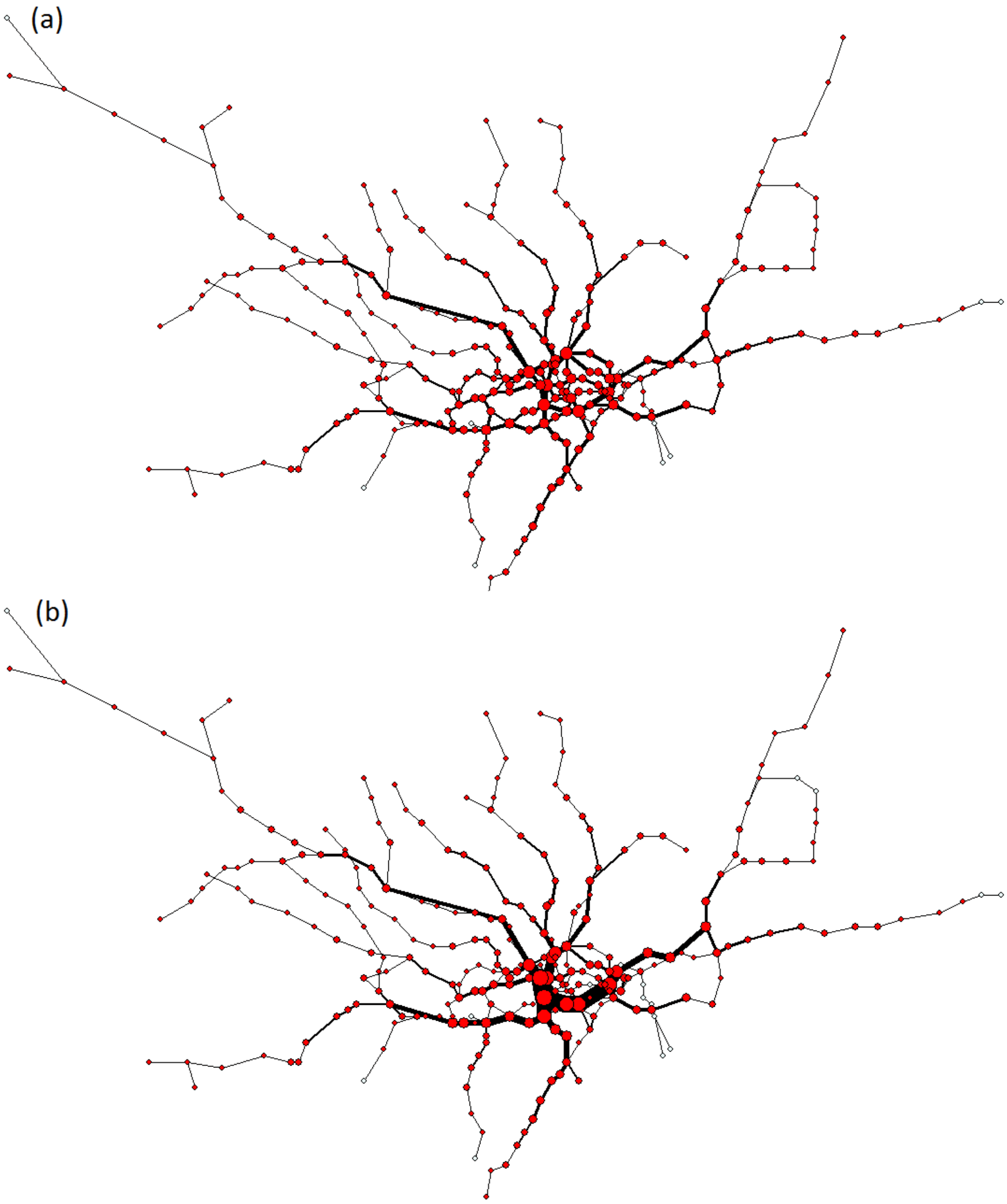

Fig. S 7: Optimized traffic for the London underground data obtained by the algorithm defined in (S129) - (S133). A total of 2316 real passenger source-destination pairs were optimized, corresponding to 5% of data recorded by the Oyster card system between 8:30am - 8:39am on one Wednesday in November 2009. The corresponding costs are (a) $\mathcal{H} \propto \sum_i {I_i}^2$, and (b) $\mathcal{H} \propto \sum_i I_i^{0.5}$. Red nodes correspond to stations with non-zero traffic. The size of each node and the thickness of each edge are proportional to traffic through them. By comparing the results obtained by our algorithms with those obtained by Djisktra's shortest path algorithm as in Table 1 of the main paper, we find $(E_P - E_D)/E_D$ and $(L_P - L_D)/L_D$ to be 23.7% and -4.6% for $\gamma = 2$ and 7.0% and -6.5% for $\gamma = 0.5$, respectively.

such that

$$\sigma^{\nu}_{j\to i} = \delta_{\Lambda_\nu,1}\Theta\left(\min_{l\in\mathcal{L}_j\setminus\{i\}}\left[b^{\nu}_{l\to j}\right] - b^{\nu}_{i\to j}\right) \tag{S137}$$
$$+\,\delta_{\Lambda_\nu,0}\Theta\left(\min\left[0, b^{\nu}_{i\to j}+\min_{l\in\mathcal{L}_j\setminus\{i\}}\left[a^{\nu}_{l\to j}\right], \min_{\substack{l,r\in\mathcal{L}_j\setminus\{i\}\\ l\neq r}}\left[a^{\nu}_{l\to j}+b^{\nu}_{r\to j}\right]\right] - \left[a^{\nu}_{i\to j}+\min_{l\in\mathcal{L}_j\setminus\{i\}}\left[b^{\nu}_{l\to j}\right]\right]\right)$$

$$\sigma^{\nu}_{i\to j} = \delta_{\Lambda_\nu,-1}\Theta\left(\min_{l\in\mathcal{L}_j\setminus\{i\}}\left[a^{\nu}_{l\to j}\right] - a^{\nu}_{i\to j}\right) \tag{S138}$$
$$+\,\delta_{\Lambda_\nu,0}\Theta\left(\min\left[0, a^{\nu}_{i\to j}+\min_{l\in\mathcal{L}_j\setminus\{i\}}\left[b^{\nu}_{l\to j}\right], \min_{\substack{l,r\in\mathcal{L}_j\setminus\{i\}\\ l\neq r}}\left[a^{\nu}_{l\to j}+b^{\nu}_{r\to j}\right]\right] - \left[b^{\nu}_{i\to j}+\min_{l\in\mathcal{L}_j\setminus\{i\}}\left[a^{\nu}_{l\to j}\right]\right]\right)$$

We remark that the computational complexity of the above algorithm is also $O(MN)$.

4 Comparison with existing multi-commodity flow optimization algorithms

In this section we discuss the principal differences between our algorithm and existing routing algorithms, especially those devised in the study of multi-commodity flow. We will also give a quantitative comparison of the optimized energy obtained by our algorithm and by a state-of-the-art multi-commodity flow algorithm.

Multi-commodity flow problem corresponds to the minimization of a weighted linear cost of the form

$$C = \sum_{ij} w_{ij} I_{ij} \tag{S139}$$

where I_{ij} represents the flow on the edge ij and w_{ij} is the corresponding weight, which can be interpreted as the local cost of using edge ij for transportation/communication; the minimization is typically subject to constraints on edge capacity c_{ij} and flow conservation, to satisfy M source-destination pairs (also called commodities) kl with demand D_{kl}. If we denote each commodity by ν, the flow I_{ij} on an edge ij is the sum of the flow of each commodity, i.e. $I_{ij} = \sum_{\nu=1}^{M} I^{\nu}_{ij}$.

Conventional multi-commodity flow problems allow for the transportation of each commodity via multiple paths, thus considering the I^{ν}_{ij} as a real variable. The linear cost in Eq. (S139) and the real-valued flow I^{ν}_{ij} facilitates the use of linear programming to solve the optimization problem. However, as linear programming for large systems can be time consuming, large effort has been devoted to the derivation of controlled approximations, which gave rise in efficient algorithms [S11, S12, S14, S13, S15]. Many of these algorithms follow the rationale of [S11] and assign a distance on edges, which is dynamically updated to satisfy the required constraints while minimizing the cost. Thus, most of these algorithms rely on centralized computations to find shortest path according to the (weighted) distance function.

While the majority of multi-commodity flow studies focus on real-valued variables, integer multi-commodity flow problems are also studied being more suitable to a range of applications from airline fleet assignment and schedule planning [S17, S16] to optical networks [S18]. Integer multi-commodity flow problems are considered more difficult than their real-valued counterparts and more advanced techniques for integer programming, such as branch-and-price restricted column generation [S19, S20] and primal partitioning [S21], have to be employed. They usually require more complicated implementations and simplifications are restricted to linear costs.

To solve the problem we have introduced, namely a system of interacting polymers/ communications, an algorithm capable to solve an *integer programming problem with non-linear cost* is required (let alone concave cost functions); while most multi-commodity flow algorithms are limited to the case of linear cost. Non-linear cost optimization has been studied in the context of transportation and flow networks but usually with a single commodity only [S22, S23, S24]. An integer problem with non-linear cost is studied in optical networks but only on circular networks with a specific setting of source and destination [S18]. To the best of our knowledge, there is no simple distributive algorithm capable of solving an integer multi-commodity flow problem with a generic non-linear cost function (especially concave).

Nevertheless, it is useful to compare the performance of our algorithm with those of existing state-of-the-art algorithms. We thus adapt the multi-commodity flow optimization algorithm [S11], designed for real-valued variables, and then for each commodity identify the path with the largest flow to be the one connecting the source and destination, resembling the idea of linear programming relaxation. We remark that this approach [S11] is the basis of many existing congestion-aware algorithms, but congestion is reduced only in the sense that edge capacity restrictions are satisfied. In addition, to optimize the non-linear (quadratic) cost used in our set-up, and straightforwardly minimized by our algorithm, we introduced a tunable parameter α to the algorithm [S11] and optimize the cost by an extensive search over all α values. Without such parameter search, the multi-commodity flow algorithm performs very unfavorably compared to our algorithm.

The algorithm of Ref. [S11] maximizes the concurrent flow given a set of edge capacities, or equivalently, minimizes the edge capacity such that the specified demand is satisfied (which is called the *minimum capacity utilization problem* [S11]). In this sense, the algorithm mitigates congestion by uniform distribution of traffic. As only linear cost is originally considered [S11], we slightly modify this algorithm to minimize a quadratic traffic cost. We term it as the *min-cap* algorithm, characterized by a weight parameter α; the algorithm is determined as follows:

1. Assign a distance $d_i = 1/N$ for each node. For each source-destination pair i and j, we set the flow $I(p^*_{ij}) = 1$, where p^*_{ij} is one of the shortest paths connecting i and j.
2. Normalize $I(p)$ subject to the constraint $\sum_{p \in P'} I(p) = 1$, where P' is the set of active

paths, i.e. paths with non-zero flow. Normalize d_i subject to $\sum_i d_i = 1$. Evaluate $I_i = \sum_{p \in P_i} I(p)$, where P_i is the set of active paths passing through i.

3. Update d_i by the formula:

$$d_i = \frac{e^{\alpha I_i}}{\sum_j e^{\alpha I_j}}. \tag{S140}$$

4. For all source-destination pair i and j, compute the shortest path (with the new set of distances d_i) and denote it by p^*_{ij}. Among all the non-zero flow paths for a source-destination pair i and j, denote its longest one by p^L_{ij}. Find the source-destination pair k, l such that $(k.l) = \mathrm{argmax}[\sum_{i' \in p^L_{ij}} d_{i'} - \sum_{i' \in p^*_{ij}} d_{i'}]$. Compute

$$\sigma = \frac{1}{2\alpha} \log \frac{\sum_{i \in p^*_{kl} \setminus p^L_{kl}} d_i}{\sum_{i \in p^L_{kl} \setminus p^*_{kl}} d_i} \tag{S141}$$

where $p^*_{kl} \setminus p^L_{kl}$ denotes the nodes in path p^*_{kl} but not in p^L_{kl}. If $I(p^L_{kl}) > \sigma$, re-route σ units of flow from path p^L_{kl} to p^*_{kl}; otherwise, re-route all flow from p^L_{kl} to p^*_{kl}.

5. Repeat step 2 to step 4 until $\sigma = 0$. For each source-destination pair i and j, identify the dominant path, i.e. a single path with maximum flow among all active paths connecting i and j. Compute I_i as the number of dominant paths passing node i. Compute energy $E = \frac{1}{N} \sum_i I_i^2$.

In Fig. S8, we show three examples of the optimized energy $E = \frac{1}{N} \sum_i I_i^2$ as a function of parameter α, compared to the optimized energy obtained by our algorithm and Dijkstra algorithm. As we can see, an extensive search over α is required to achieve an optimized energy close to the optimized energy obtained by our algorithm, especially for the global airport network and random regular graph where E does not show a smooth behavior. In addition, the optimal α is not universal and has to be found individually for different instances and is problem-dependent. Without such a search over α, for example, if a small α value is used as suggested by Ref. [S11], the optimized energy found will be much higher than the one obtained by our algorithm.

The average optimized energies at individual optimal α^* obtained by the min-cap algorithm is compared to those obtained by our algorithm in Table S1. The energy gain is modest with respect to those achieved with respect to the Dijkstra's shortest path algorithm as shown in Table 1 of the article. Nevertheless, our algorithm always provides the lower bound of energy, which is not achievable by the min-cap algorithm (see Fig. S5). Our algorithm also results in shorter average path lengths L and lower energy E in random regular graphs, used as a controlled benchmark problem.

To further compare the performance of our algorithm with that of the min-cap algorithm, we tested both on two sets of multi-commodity flow benchmark graph instances, namely the *Mnetgen* and the *PDS* instances [S25]. We remark that our algorithm has not been designed to consider edge capacities (although an extension in this direction is

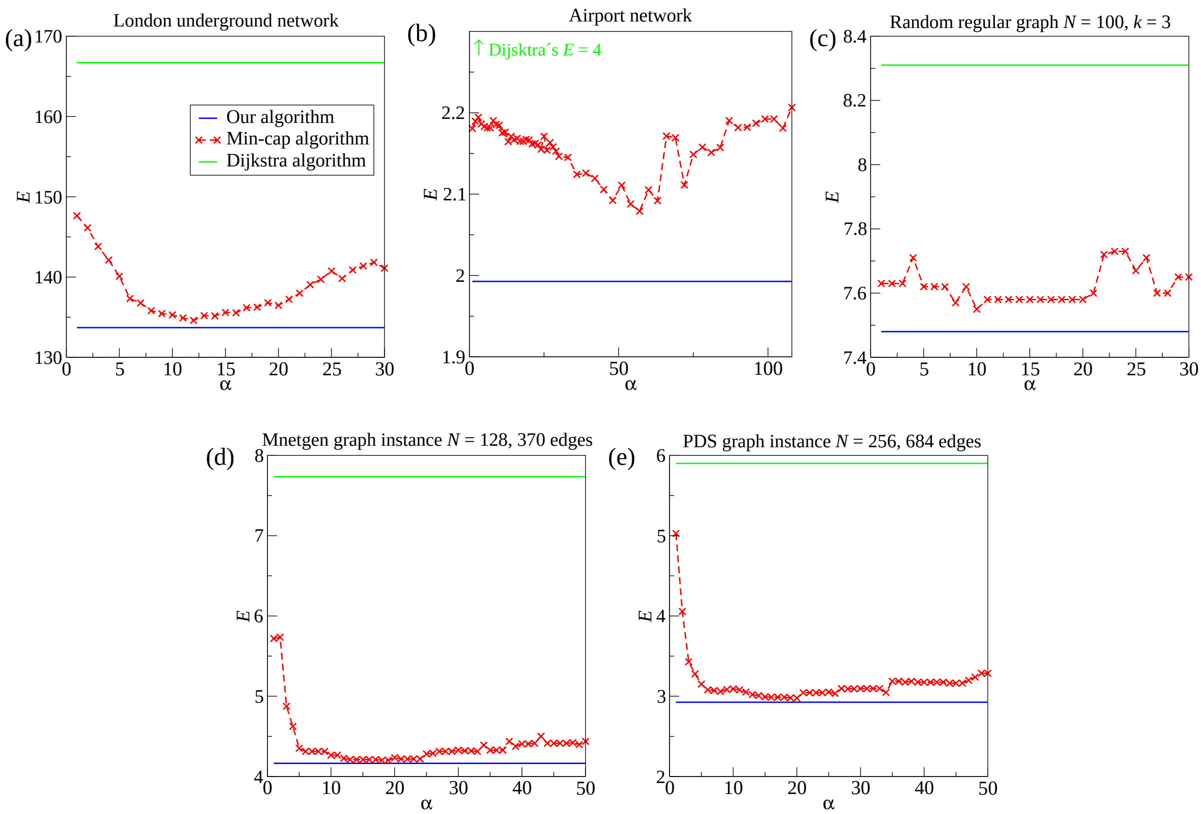

Fig. S 8: The optimized energy $E = \frac{1}{N}\sum_i I_i^2$ obtained by (top to bottom) the Dijkstra algorithm, the min-cap algorithm and our routing algorithm, on (a) the London subway network with 246 real passenger source-destination pairs, (b) the global airport network with 300 random source-destination pairs, (c) a random regular network with $N = 100$, $k = 3$ and $M = 40$, (d) a Mnetgen graph instance with $N = 128$ and $M = 50$, and (e) a PDS graph instance with $N = 256$ and $M = 50$.

feasible and will be studied elsewhere) and they were therefore ignored in this comparison. Moreover, the demand in these instances, e.g. distributed source and sink nodes of the same commodity over the network, are different from our setting of individual source-destination pairs; we thus randomly draw source-destination pairs instead of implementing the demand as stated in the instances. The optimized results are shown in Table S1. Results are qualitatively similar to all previous examples, except that the min-cap algorithm slightly outperforms our algorithm in energy by 0.1% on the Mnetgen instances, although this result is not statistically significant. We note that our algorithm provides lower energy E as well as shorter average path length L for the PDS graph instances, similar to that observed for random regular graphs. Figure 8(d) and (e) show similar plots of the extensive search over α to optimize the required energy on both Mnetgen and PDS instances, separately.

In addition to the reduced energy obtained, we emphasize that our algorithm is distributive rather than centralized, principled rather than heuristic, and does not rely on free parameters. Most importantly, this comparison is only limited to congestion-aware algorithms while our algorithm is generic and can accommodate any non-pathological cost function designated to suit the specific objectives and characteristics of the problem. For instance, we cannot find any efficient algorithm which for optimizing path-sharing as obtained by our algorithm for $\gamma < 1$.

	$\gamma = 2$	
	$\frac{E_{\rm P}-E_{\rm MC}(\alpha^*)}{E_{\rm MC}(\alpha^*)}$	$\frac{L_{\rm P}-L_{\rm MC}(\alpha^*)}{L_{\rm MC}(\alpha^*)}$
London subway network	$-0.70 \pm 0.05\%$	$+0.72 \pm 0.10\%$
Global airport network	$-3.09 \pm 0.59\%$	$+0.90 \pm 0.64\%$
Random regular graphs ($N = 100$, $k = 3$)		
$M = 10$	$-0.01 \pm 0.06\%$	$-2.53 \pm 0.09\%$
$M = 20$	$-1.02 \pm 0.04\%$	$-0.80 \pm 0.05\%$
$M = 40$	$-0.66 \pm 0.02\%$	$-0.20 \pm 0.03\%$
Mnetgen graphs ($N = 128, M = 50$)	$+0.11 \pm 0.17\%$	$-0.78 \pm 0.16\%$
PDS graphs ($M = 50$)	$-1.85 \pm 0.15\%$	$-1.35 \pm 0.21\%$

Table S 1: A comparison of optimized energy $E = \sum_i I_i^2$ and path length $L = \frac{1}{M} \sum_i I_i$ obtained by our algorithm (P) and the min-cap algorithm (MC) at individual optimal α^* for each instance. Results are averaged over sets of source-destination pairs recorded in each 1 minute interval between 8:30 am – 9:00 am on the London subway network; 5 sets of 300 randomly drawn source-destination pairs for the global airport network; 2000 sets of random regular graphs with $N = 100$ and $k = 3$; 100 sets of 50 randomly drawn pairs on each of the first 12 graph instances of the Mnetgen generator [S25] with $N = 128$; and 100 sets of 50 randomly drawn pairs on each of the first 5 graph instances of the PDS problem [S25] with N ranging from 126 to 686.

5 The emergence of replica symmetric breaking (RSB)

Replica symmetric breaking (RSB) [S1, S2] is a phenomenon which describes the emergence of numerous local minima separated by high energy barriers. Although RSB is a phenomenon that primarily occurs in infinite systems, it may hinder algorithmic convergence in finite systems and is potentially one of the reasons for non-convergence we have experienced in some real instances.

To test the emergence of RSB, we examine the Hamming distance between two infinite systems which share the same quenched disorders, i.e. identical topology and source and destination for each communication, but different initial conditions. We denote the cavity field $\vec{u}$ in the two systems by $\vec{u}_\alpha$ and $\vec{u}_\beta$ respectively, and start with a random initial condition, iterate the following equation to obtain the distribution $w(\vec{u}_\alpha, \vec{u}_\beta)$

$$
\begin{aligned}
w(\vec{u}_\alpha, \vec{u}_\beta) = & \sum_{k=1}^{\infty} \frac{k\rho(k)}{\langle k \rangle} \prod_{l=1}^{k} \left[\int d\vec{u}_{l\alpha} d\vec{u}_{l\beta}\; w(\vec{u}_{l\alpha}, \vec{u}_{l\beta}) \right] \prod_{\nu} \sum_{\Lambda_\nu=0}^{1} \left[\delta_{\Lambda_\nu,0} + \frac{2}{N} \delta_{\Lambda_\nu,1} \right] \\
& \times \prod_{\nu} \delta \left\{ u_\alpha^\nu - (1-\Lambda_\nu) \left(\min_{l \le k-1} [u_{l\alpha}^\nu] - \min \left[-\phi'(\lambda_{\nu\alpha}^*), \min_{\substack{l,r \le k-1 \\ l \ne r}} [u_{l\alpha}^\nu + u_{r\alpha}^\nu] \right] \right) - \Lambda_\nu \min_{l \le k-1} [u_{l\alpha}^\nu] \right\} . \\
& \times \prod_{\nu} \delta \left\{ u_\beta^\nu - (1-\Lambda_\nu) \left(\min_{l \le k-1} [u_{l\beta}^\nu] - \min \left[-\phi'(\lambda_{\nu\beta}^*), \min_{\substack{l,r \le k-1 \\ l \ne r}} [u_{l\beta}^\nu + u_{r\beta}^\nu] \right] \right) - \Lambda_\nu \min_{l \le k-1} [u_{l\beta}^\nu] \right\} .
\end{aligned}
\tag{S142}
$$

where $\lambda_{\nu\alpha}^* = f_{\nu\alpha}^{-1}(0)$ and $\lambda_{\nu\beta}^* = f_{\nu\beta}^{-1}(0)$, with $f_{\nu\alpha}(x)$ and $f_{\nu\beta}(x)$ given by

$$
f_{\nu\alpha}(x) = x - \frac{1}{M} - \frac{1}{M} \sum_{\mu \ne \nu} \left\{ \Lambda_\mu + (1-\Lambda_\mu) \Theta \left(-\phi'(x) - \min_{\substack{l,r \le k \\ l \ne r}} [u_{l\alpha}^\mu + u_{r\alpha}^\mu] \right) \right\}
\tag{S143}
$$

$$
\begin{aligned}
f_{\nu\beta}(x) = & x - \frac{1}{M} \\
& - \frac{1}{M} \sum_{\mu \ne \nu} \left\{ \Lambda_\mu + (1-\Lambda_\mu) \Theta \left(-\phi'(x) - \min \left[\min_{\substack{l,r \le k-1 \\ l \ne r}} [u_{l\beta}^\mu + u_{r\beta}^\mu], \min_{l \le k-1} [u_{l\beta}^\mu + u_{l=k,\alpha}^\mu] \right] \right) \right\}
\end{aligned}
\tag{S144}
$$

where the slightly more complicated definition of $f_{\nu\beta}(x)$ is introduced to make sure that the additional k-th field arising from the approximation (S43) is the same for both systems α and β.

To examine the evolution of Hamming distance between the two systems as the corresponding fields propagate, we define R as the ratio of Hamming distances (between $\vec{u}_\alpha$ and $\vec{u}_\beta$) at the current and previous levels, given by

$$
R = \frac{\sqrt{\sum_{\nu=1}^{M} (u_\alpha^\nu - u_\beta^\nu)^2}}{\frac{1}{k-1} \sum_{l=1}^{k-1} \sqrt{\sum_{\nu=1}^{M} (u_{l\alpha}^\nu - u_{l\beta}^\nu)^2}}
\tag{S145}
$$

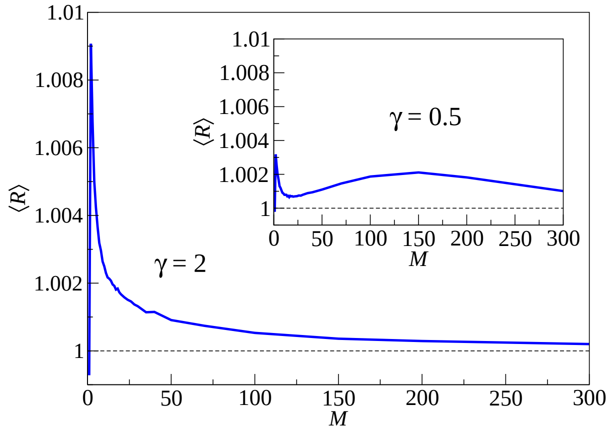

Fig. S9: The average ratio $\langle R \rangle$ (defined by Eq. (S145)) between Hamming distances at the current and previous levels as a function of M, at $N = 100$, $\gamma = 2$ for $\mathcal{H} \propto \sum_i I_i^\gamma$. Inset: the same plot for $\gamma = 0.5$.

where u_α^ν and u_β^ν are computed by the delta function in Eq. (S142) in terms of $u_{l\alpha}^\nu$'s and $u_{l\beta}^\nu$'s. After a stable solution $w(\vec{u}_\alpha, \vec{u}_\beta)$ is obtained, we evaluate the average value $\langle R \rangle$ by computing R in every iteration of Eq. (S142). In cases of $\langle R \rangle < 1$, the Hamming distance vanishes as $N \to \infty$ which corresponds to the replica symmetry (RS) behavior. On the other hand, when $\langle R \rangle > 1$ the hamming distance diverges as $N \to \infty$, corresponding to the RSB behavior. The results are shown in Fig. S9, where for both cases of $\gamma = 2$ and $\gamma = 0.5$, $\langle R \rangle$ is slightly larger than 1 for all M except $M = 1$, suggesting the system is characterized by the RSB behavior in most of the parameter space.

We understand that incorporating the RSB ansatz into the algorithm may lead to different energy solutions in both artificial and real single instances. However, we do not expect the true physical picture to be qualitatively different from the results shown in Fig. 6 of the main paper. In extreme traffic conditions the optimized path length shown in Fig. 6(a) agrees well with our expectation: when $M = 1$, we obtain the shortest path; when M is large, the path length approaches the shortest path again as communications are expected to go through the shortest path when traffic become homogeneous in the large M limit. In regimes with intermediate M values we expect a peak in path length, which may be marginally lower if RSB would be used, based on our experience with other hard problems of this type. Thus, we expect the true physical pictures to agree qualitatively with the results shown in Fig. 6 of the main text.

6 The optimal path length in ER and SF graphs

In Fig. S10 we show the rescaled path length $(\langle L \rangle - L_1)(N/\log N)$ in Erdös-Rényi (ER) graphs as a function of rescaled number of communication $M/(N \log N)$. As we can see, $\langle L \rangle$ increases as M increases and then slowly decrease as M increases further, which is similar to what has been observed in regular graphs. We also observe a similar data collapse for

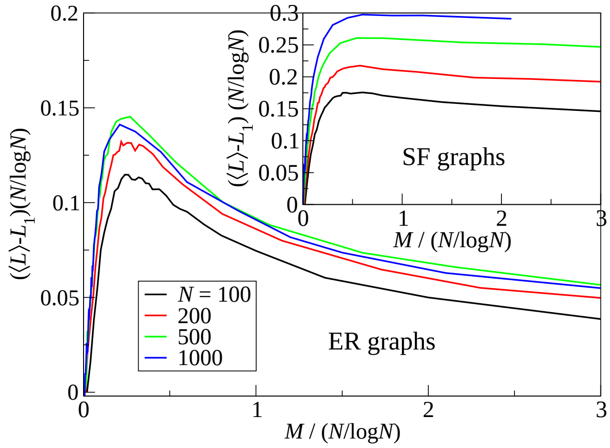

Fig. S10: The rescaled path length $(\langle L\rangle - L_1)(N/\log N)$ as a function of rescaled number of communication $M/(N\log N)$, for Erdös-Rényi (ER) graphs with $N = 100, 200, 500, 1000$ and $\langle k\rangle = 3$, in results obtained for $\mathcal{H} \propto \sum_i {I_i}^2$. The value of L_1 corresponds to the value of $\langle L\rangle$ when $M = 1$ for the corresponding system, i.e. shortest path. Inset: $(\langle L\rangle - L_1)(N/\log N)$ as a function of $M/(N\log N)$ for scale-free (SF) graphs.

systems with different N values in ER graphs after rescaling with $N/\log N$. A similar plot for the scale-free (SF) graphs is shown in the inset of Fig. S10. The decrease of $\langle L\rangle$ is much slower after the peak when compared to regular and ER graphs, possibly due to the intrinsic node degree inhomogeneity in SF graphs which leads to traffic inhomogeneity even at large M. In addition, the data scaling with $N\log N$ is not as accurate as that observed in regular and ER graphs.

7 The fraction of idle nodes and the phase transition at $\gamma = 1$

In addition to the dependence on M, we examined the influence of the choice of cost exponent γ on the optimized states. As we can see from the inset of Fig. S11, the minimum of $\langle L\rangle$ is observed at $\gamma = 1$ which corresponds to the case where all communications are routed through the shortest paths, while both cases of $\gamma > 1$ and $\gamma < 1$ show a longer average path $\langle L\rangle$ to avoid congestion or aggregate traffic, respectively. Interestingly, the fraction $f_{\rm idle}$ of idle nodes shows a prominent jump at $\gamma = 1$ in simulations, resembling a phase transition. While noting the deviation between theory and simulations, one also observes a small jump at $\gamma = 1$ for the curves obtained from the theoretical equations, which is visible at a larger scale (inset). Discrete jumps and large deviations of this type are not observed for $\langle L\rangle$ or average cost $\langle E\rangle$. A similar phase transition at $\gamma = 1$ is reported in [S26] with regard to flow patterns of electric currents in resistor networks. In the present context the implication is that even a small change at $\gamma = 1$ is sufficient to effectively power down unnecessary routers or close redundant subway stations (higher

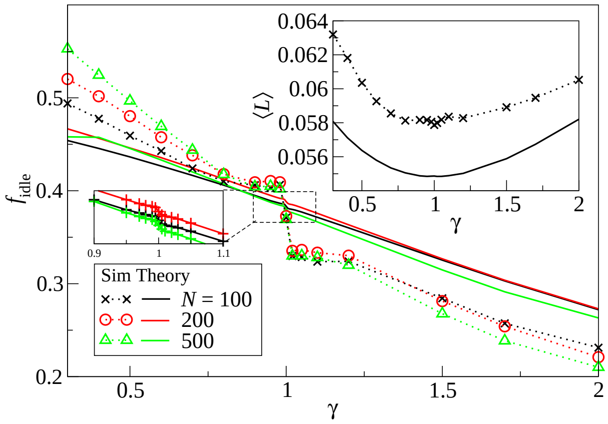

Fig. S 11: Dependence of the optimized state on the cost exponent. The fraction f_{idle} of idle nodes as a function of γ for random regular graphs of sizes $N = 100, 200, 500$ and $k = 3$, with $M \approx N/(k \log N)$ communications that approximate peak positions in Fig. 6 of the main article. Inset: $\langle L \rangle$ as a function of γ for the networks with $N = 100$. Small inset: enlarged plot for the analytical results of f_{idle} as a function of γ around $\gamma = 1$. The error bars for simulation results are of the order of the symbol size. All simulation results are averaged over 1000 realizations.

f_{idle}), with little impact on the cost or average route length $\langle L \rangle$.

References

[S1] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).

[S2] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing* Oxford University Press, Oxford, UK, 2001.

[S3] M. Daoud, J. P. Cotton, B. Farnoux, G. Jannink, G. Sarma, H. Benoit, R. Duplessix, C. Picot, and P. G. de Genne, Marcomolecules 8, 804 (1975)

[S4] M. T. Batchelor, B. Nienhuis, and S. O. Warnaar, Phys. Rev. Lett. 62, 2425-2428 (1989).

[S5] J. F. Stilck, K. D. Machado and P. Serra Phys. Rev. Lett. 76, 2734-2737 (1996)

[S6] M. Leone, A. Vázquez, A. Vespignani, and R. Zecchina, Eur. Phys. J. B 28, 191-197 (2002).

[S7] D. Sherrington and S. Kirkpatrick, Phys. Rev. Lett. 16, 1792 (1975).

[S8] M. Mezard and R. Zecchina, Phys. Rev. E 66, 056126 (2002).

[S9] C. H. Yeung and K. Y. M. Wong, J. Stat. Mech P04017 (2010).

[S10] C. H. Yeung and D. Saad, Phys. Rev. Lett. 108, 208701 (2012).

[S11] F. Shahrokhi and D. W. Matula, The Maximum Concurrent Flow Problem, Journal of the Association for Computing Machinery 37, 318-334 (1990).

[S12] T. Leighton and S. Rao, An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms, Proceedings of Foundations of Computer Science 29th Annual Symposium, 422 - 431 (1988).

[S13] N. Garg and J. Könemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, Proceedings of Foundations of Computer Science 39th Annual Symposium (1998).

[S14] B. Awerbuch, Y. Azar and S. Plotkin, Throughput-competitive on-line routing, Proceedings of Foundations of Computer Science 34th Annual Symposium. 32-40 (1993).

[S15] B. Awerbuch and R. Khandekar, Greedy distributed optimization of multicommodity flows, Distributed Computing 21, 317-329 (2009).

[S16] P. Cappanera and G. Gallo, A Multicommodity Flow Approach to the Crew Rostering Problem, Operations Research 52, 583-596 (2004).

[S17] B. Verweij, K. Aardal, and G. Kant. On an integer multicommodity flow problem from the airplane industry. Technical Report UU-CS-1997-38, Utrecht University, Department of Computer Science (1997).

[S18] A. E. Ozdaglar and D. P. Bertsekas, Optimal solution of integer multicommodity flow problems with application in optical networks, Proc. of symposium on Global Optimization, 2003.

[S19] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh and P. H. Vance, Branch-and-Price: Column Generation for Solving Huge Integer Programs Operations Research 46, 316-329 (1998).

[S20] C. Barnhart, C. A. Hane and P. H. Vance, Using branch-and-price-and-cut to solve origin destination integer multicommodity flow problems, Operations Research 48, 318-326 (2000).

[S21] J. Castro and N. Nabona, An implementation of linear and nonlinear multicommodity network flows, European Journal of Operational Research 92, 37-53 (1996).

[S22] K. Scott and D. Bernstein, Solving a best path problem when the value of time function is nonlinear. Technical report, New Jersey TIDE Center, New Jersey Institute of Technology 11, 1997.

[S23] G. Tsaggouris and C. Zaroliagis, Multiobjective optimization: improved FPTAS for shortest paths and non-linear objectives with applications, Algorithms and Computation, Lecture Notes in Computer ScienceVolume 4288/2006, 389-398 (2006).

[S24] R. P. Agdeppa, N. Yamashita and M. Fukushima, The traffic equilibrium problem with nonadditive costs and its monotone mixed complementarity problem formulation, Transportation Research Part B: Methodological 41, 862 - 874 (2007).

[S25] The Mnetgen and the PDS instances are downloaded from the website http://www.di.unipi.it/optimize/Data/MMCF.html.

[S26] S. Bohn, M. O. Magnasco, Phys. Rev. Lett. 98, 088702 (2007).