Distributed Message Passing for Large Scale Graphical Models

Yan Shen
University at Buffalo

Outline

- Algorithm
- Implementation Detail
- Experiment evaluation
- Conclusion

MAP as Convex optimization

- Maximum a posteriori (MAP) assignment

$$\mathrm{x}^* = \operatorname*{argmax}_{\mathrm{x}} \prod_{i=1}^{n} \psi_i(x_i) \prod_{\alpha=1}^{m} \psi_\alpha(\mathrm{x}_\alpha). \qquad (2)$$

- Reformulate the MAP problem as *convex optimization*

$$\max \sum_{\alpha,\mathrm{x}_\alpha} b_\alpha(\mathrm{x}_\alpha)\theta_\alpha(\mathrm{x}_\alpha) + \sum_{i,x_i} b_i(x_i)\theta_i(x_i) \qquad (5)$$

$$+\epsilon\left(\sum_\alpha c_\alpha H(\mathrm{b}_\alpha) + \sum_i c_i H(\mathrm{b}_i)\right)$$

subject to:

$$\forall i, x_i, \alpha \in N(i), \sum_{\mathrm{x}_\alpha \backslash x_i} b_\alpha(\mathrm{x}_\alpha) = b_i(x_i)$$

Distributed convex belief propagation

- Breaking into sub-problems on Different Nodes

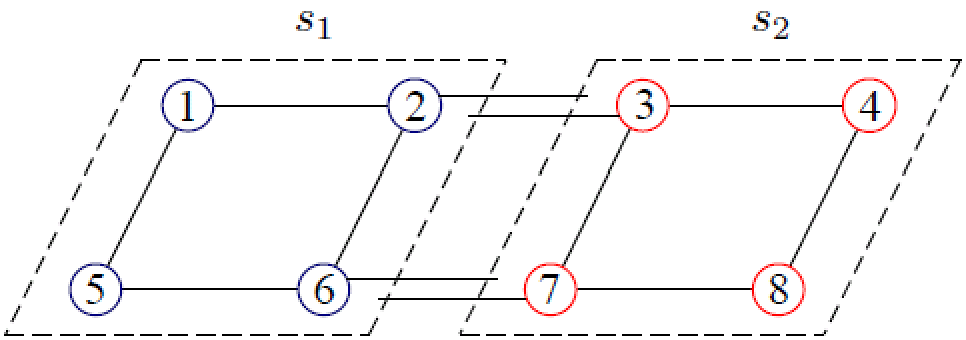
$$\max \sum_{s \in G_{\mathcal{P}}} \sum_{\alpha \in G_s, \mathbf{x}_\alpha} b_\alpha^s(\mathbf{x}_\alpha) \hat{\theta}_\alpha(\mathbf{x}_\alpha) + \sum_{i \in G_s, x_i} b_i^s(x_i) \theta_i(x_i) \quad (6)$$

$$+ \epsilon \sum_{s \in G_{\mathcal{P}}} \left(\sum_{\alpha \in G_s} \hat{c}_\alpha H(\mathbf{b}_\alpha^s) + \sum_{i \in G_s} c_i H(\mathbf{b}_i^s) \right)$$

subject to:

$$\forall s, i, x_i, \alpha \in N(i), \quad \sum_{\mathbf{x}_\alpha \setminus x_i} b_\alpha^s(\mathbf{x}_\alpha) = b_i^s(x_i)$$

$$\forall s, \alpha \in N_{\mathcal{P}}(s), \mathbf{x}_\alpha, \quad b_\alpha^s(\mathbf{x}_\alpha) = b_\alpha(\mathbf{x}_\alpha)$$



Distributed convex belief propagation

Algorithm 1 (Distributed Convex Belief Propagation) *Set* $\hat{\psi}_\alpha(\mathrm{x}_\alpha) = \exp(\hat{\theta}_\alpha(\mathrm{x}_\alpha))$, $\psi_i(x_i) = \exp(\theta_i(x_i))$. *Set* $n_{i\to\alpha}(\mathrm{x}_\alpha) = 1$ *and set* $n_{s\to\alpha}(\mathrm{x}_\alpha) = 1$. *Repeat until convergence:*

1. *For* $s \in \mathcal{P}$ *in parallel: Iterate over* $i \in s$

$$\forall x_i\ \forall \alpha \in N(i), \quad m_{\alpha\to i}(x_i) = \left(\sum_{\mathrm{x}_\alpha \setminus x_i} \left(\hat{\psi}_\alpha(\mathrm{x}_\alpha) \prod_{j \in N(\alpha)\cap s \setminus i} n_{j\to\alpha}(x_j) \cdot n_{s\to\alpha}(\mathrm{x}_\alpha) \right)^{1/\epsilon\hat{c}_\alpha} \right)^{\epsilon\hat{c}_\alpha}$$

$$\forall \alpha \in N(i)\ \forall x_i, \quad n_{i\to\alpha}(x_i) \propto \left(\psi_i(x_i) \prod_{\beta \in N(i)} m_{\beta\to i}(x_i) \right)^{\hat{c}_\alpha/\hat{c}_i} \Big/ m_{\alpha\to i}(x_i)$$

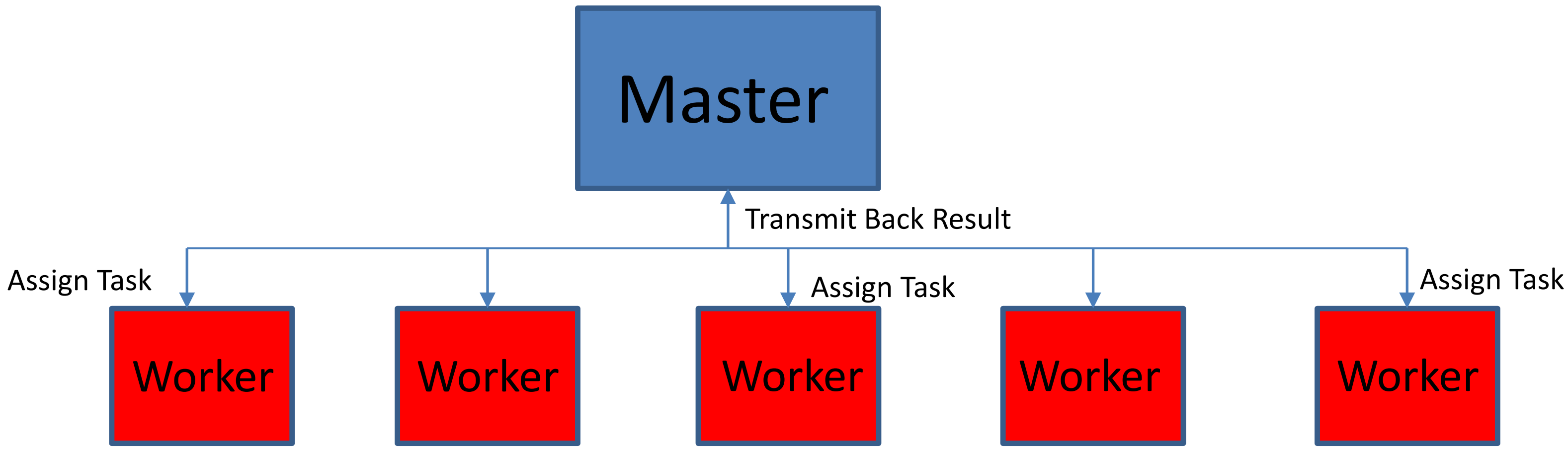
2.

$$\forall s \in G_{\mathcal{P}} \quad \forall \alpha: \ \alpha \text{ is edge in } G_{\mathcal{P}} \qquad n_{s\to\alpha}(\mathrm{x}_\alpha) = \frac{1}{|N_{\mathcal{P}}(\alpha)|} \prod_{i\in N(\alpha)} n_{i\to\alpha} \Big/ \prod_{i \in s\cap N(\alpha)} n_{i\to\alpha}(x_i)$$

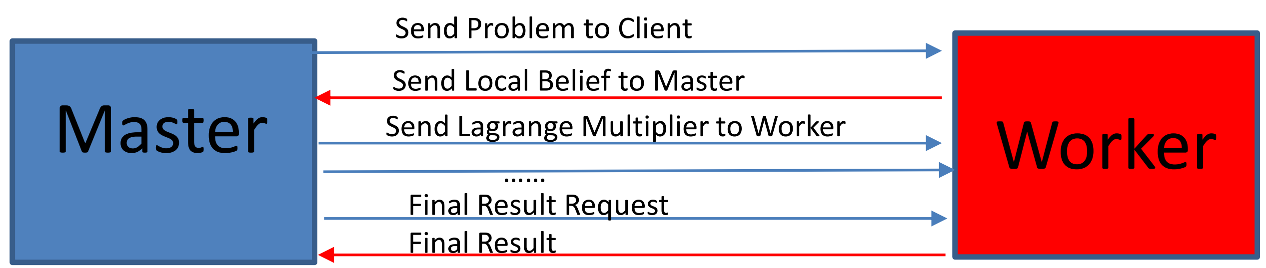
Figure 3. Our distributed convex belief propagation algorithm extends the sequential convex belief-propagation by adding messages $n_{s\to\alpha}(\mathrm{x}_\alpha)$ to maintain consistency between the distributed executions.

Implementation Detail

- Master and Worker Mode

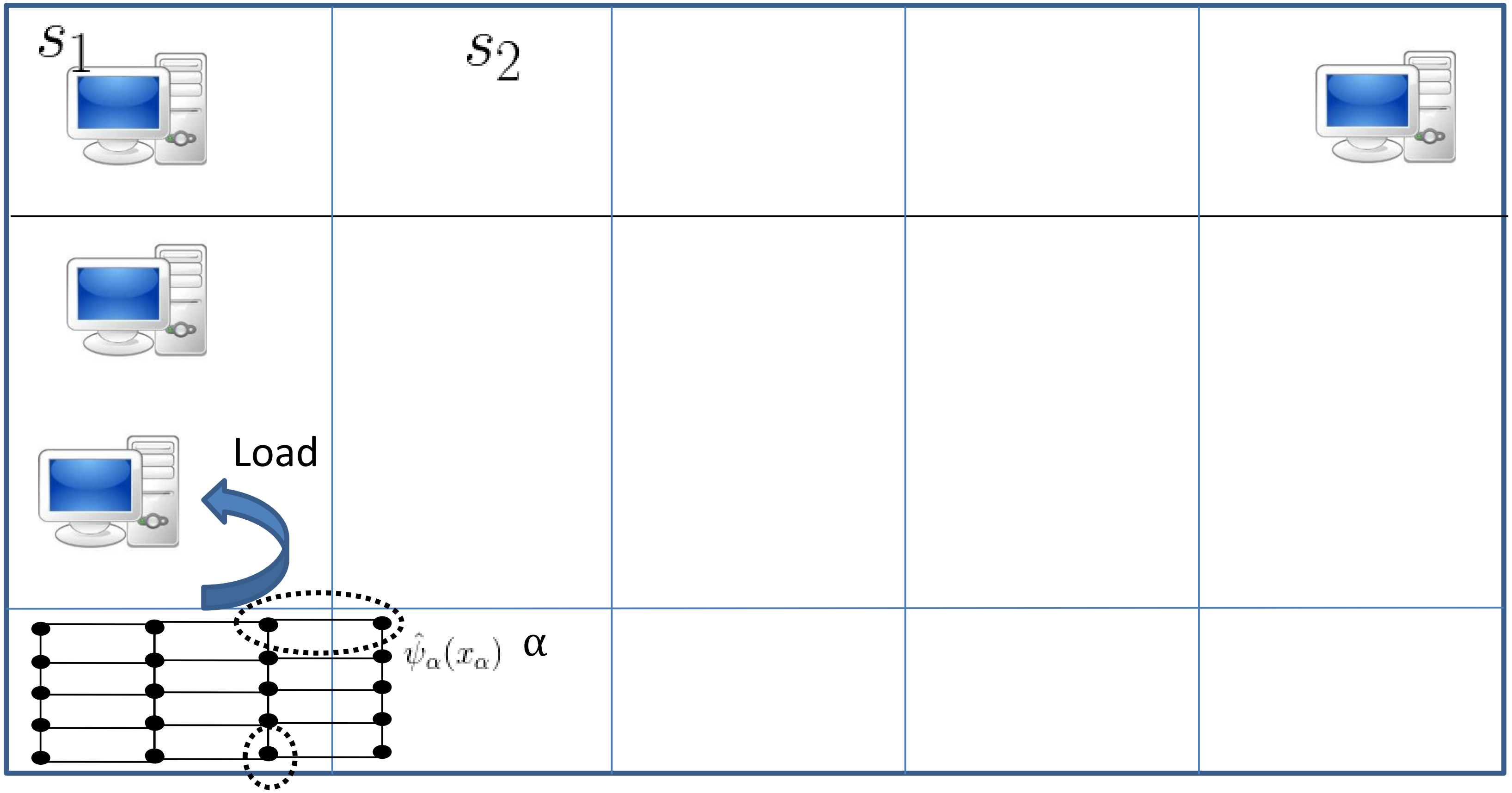


- Round Communication



Implementation Detail

- Distributed Loading Potential Data
  - Graph Structure and Node Assignment in Master
  - Worker load their subgraph and local potential from file in parallel



Implementation Detail

- Hybrid use of MPI and OpenMP
  - Worker updates their Local Beilief in (Implemented in OpenMP)
  - Worker and Master run on their Local Data and use MPI to communicate.

Worker Node

\# pragma omp parrallel on $i \in s$

$$\forall x_i \; \forall \alpha \in N(i), \quad m_{\alpha \to i}(x_i) = \left(\sum_{\mathrm{x}_\alpha \setminus x_i} \left(\hat{\psi}_\alpha(\mathrm{x}_\alpha) \prod_{j \in N(\alpha) \cap s \setminus i} n_{j \to \alpha}(x_j) \cdot n_{s \to \alpha}(\mathrm{x}_\alpha) \right)^{1/\epsilon \hat{c}_\alpha} \right)^{\epsilon c_\alpha}$$

$$\forall \alpha \in N(i) \; \forall x_i, \quad n_{i \to \alpha}(x_i) \propto \left(\psi_i(x_i) \prod_{\beta \in N(i)} m_{\beta \to i}(x_i) \right)^{\hat{c}_\alpha / \hat{c}_i} \Big/ m_{\alpha \to i}(x_i)$$

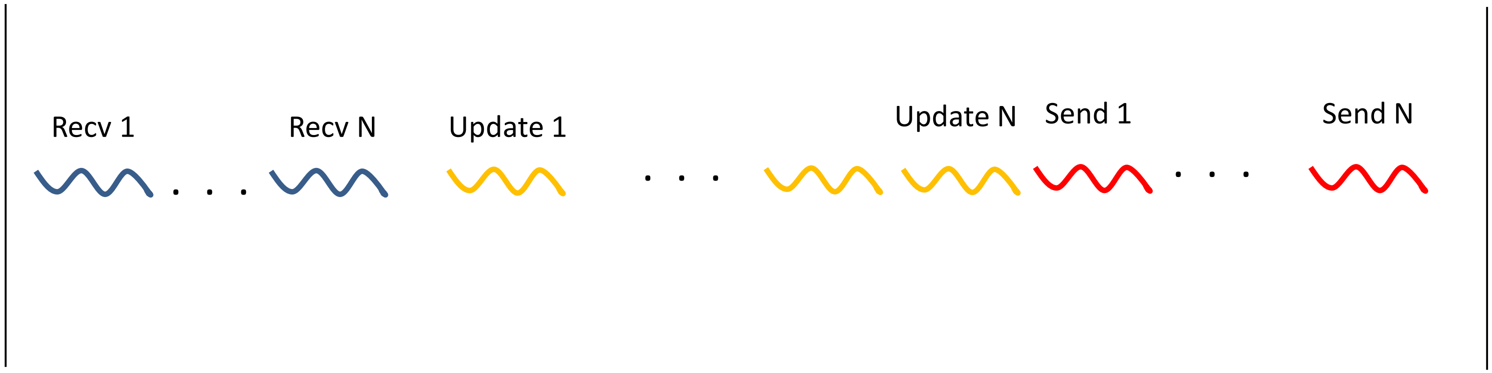
Master Node

\# pragma omp parrallel on s

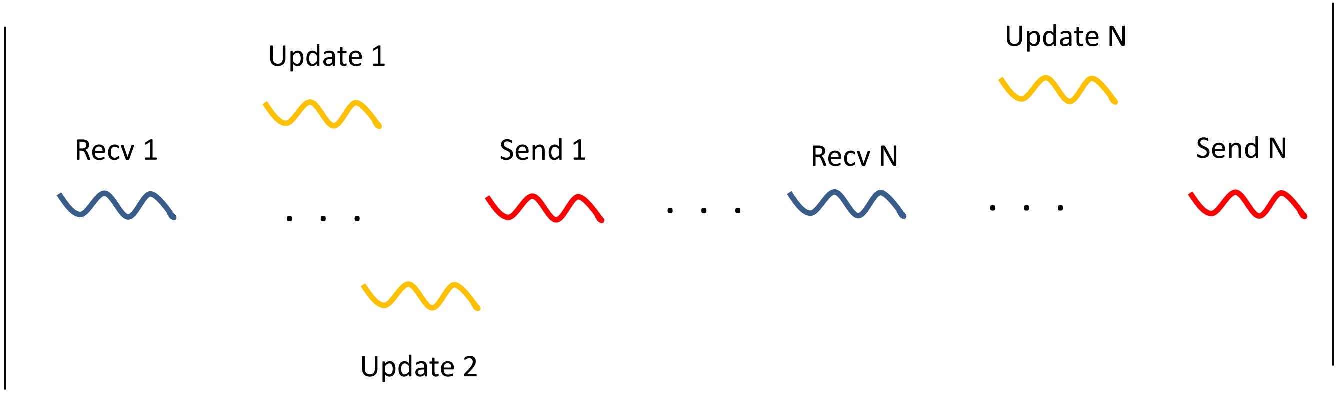
$$n_{s \to \alpha}(\mathrm{x}_\alpha) = \frac{1}{|N_{\mathcal{P}}(\alpha)|} \prod_{i \in N(\alpha)} n_{i \to \alpha} \Big/ \prod_{i \in s \cap N(\alpha)} n_{i \to \alpha}(x_i)$$

Implementation Detail

- Serial and Parrallel Mode on Master Updates
  - Serial Mode



  - Parallel Mode



Experiment evaluation

- Result and Speedup of Parrallelism

Method	Run Time	duality gap
cen. cBP	70	0.043
dis. cBP(S)	25	0.0449
dis.cBP(P)	20	0.045

(a) Tsukuba

(d) cBP $\epsilon = 0$ (e) cBP $\epsilon = 0.01$ (f) cBP $\epsilon = 0.1$

Experiment evaluation

- Convergence Result

Sever Iteration Time	Primal Value	Dual Value
1	28672	38672
2	32021	35621
3	34781	35583
4	35473	35581
5	35498	35581
6	35543	35580
7	35570	35579
8	35573	35578
9	35575	35578

- Convergence Result as number of master iteration
  (Master iteration times= 10， Worker iteration times= 10)

Experiment evaluation

- Larger Data on Bigger clusters
  - 1GB data of Local Belief
  - Partition Local Belief Data in 16 nodes in Grid
  - 17 Total Nodes (16 workers + 1 master)

sbatch.sh

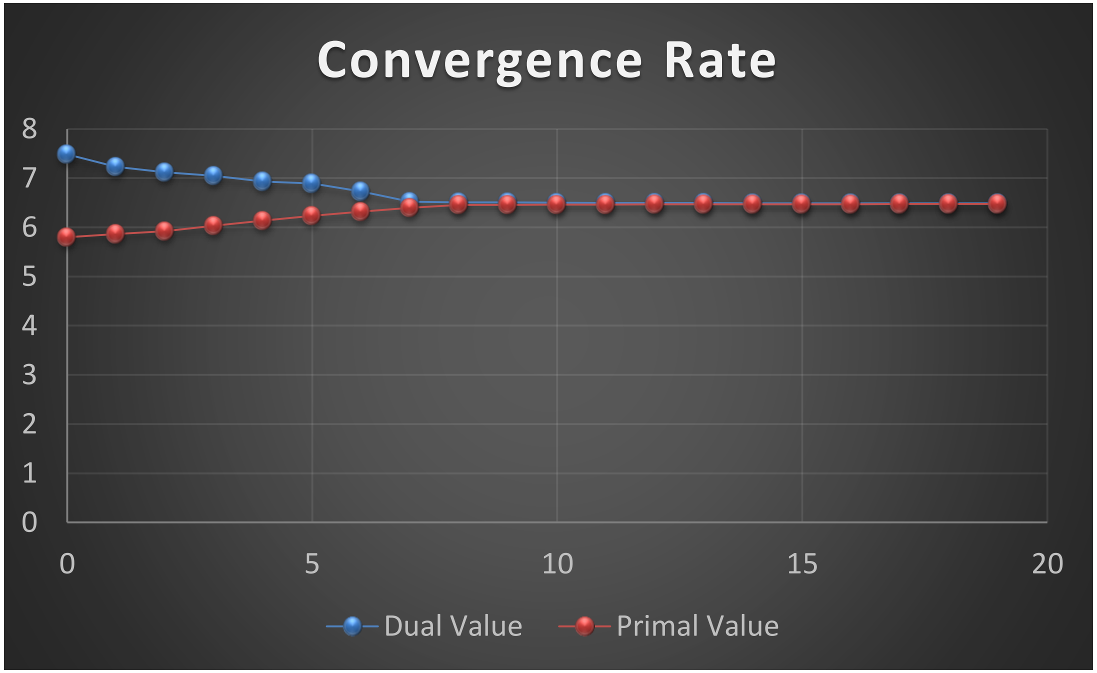
```
#!/bin/bash
#SBATCH  --nodes=17
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=3
#SBATCH --time=05:00:00
#SBATCH --mail-type=END
#SBATCH --mail-user=yshen22@buffalo.edu
#SBATCH --output=slurmQ.out
#SBATCH --job-name=mpi-testmodule
load intel-mpi intel-mpi
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
srun ./dcBP -f tsukuba.cbp -c 10 -s 20
```

Runtime Report

Job ID: 8804896
Cluster: ub-hpc
User/Group: yshen22/cse633s18
State: COMPLETED (exit code 0)
Nodes: 17
Cores per node: 3
CPU Utilized: 1-11:45:21
CPU Efficiency: 60.42% of 2-11:10:27 core-walltime
Memory Utilized: 4.76 GB (estimated maximum)
Memory Efficiency: 3.42% of 139.45 GB (2.73 GB/core)

Experiment evaluation

- Convergence Result on Clusters



- Running Times

3329.66s for total 200 iterations

Conclusion

- Large scale graphical models by dividing the computation and memory requirements into multiple machines.
- Convergence and optimality guarantees are preserved.
- Main benefit : the use of multiple computers.