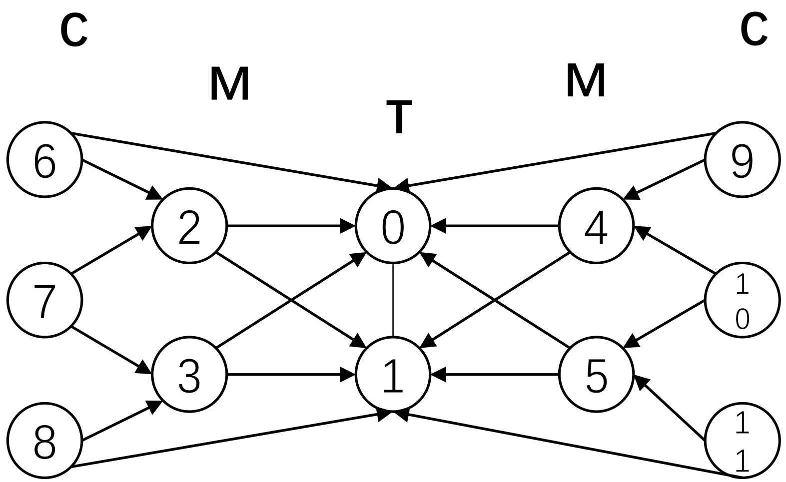
Adjustment on experiment / Distributed

2018 1.31

Network Environment



——> : customer to provider
—— : peer to peer

- Network with 12 nodes and 21 arcs.
- For every node, it has at least two choices for one specific destination.
- The action space for **T** type: 16
  The action space for **M** type: 1024
  The action space for **C** type: 2048

Integer Programming

- Basic setting for linear programming of this problem:

Objective Function $\max(\sum v^f , f \in F)$

$$\sum_{j:(i,j)\in E} x_{ij}^f - \sum_{i:(j,i)\in E} x_{ji}^f = \begin{cases} 0, & for\ all\ i \in V - s^f - t^f \\ -v^f, & for\ i = t^f \\ v^f, & for\ i = s^f \end{cases}$$

$$0 \le \sum_{f\in F} x_{ij}^f \le u_{ij} \ , \ (i,j) \in E$$

- Modify constrains to satisfy determined forwarding path:

$$\sum_{j:(i,j)\in E} x_{ij}^f - \sum_{i:(j,i)\in E} x_{ji}^f = \begin{cases} 0, & for\ all\ i \in V - s^f - t^f \\ 1 & for\ i = t^f \\ -1 & for\ i = s^f \end{cases}$$

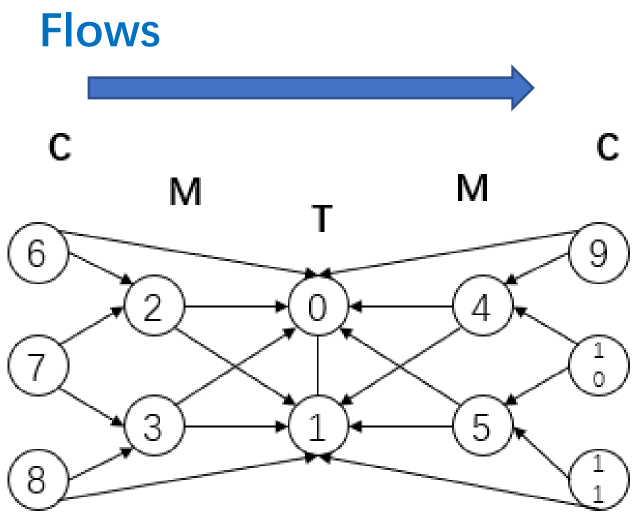
$$x_{ij}^f \in \{0,1\}, \quad for\ \forall f \in F, (i,j) \in E$$

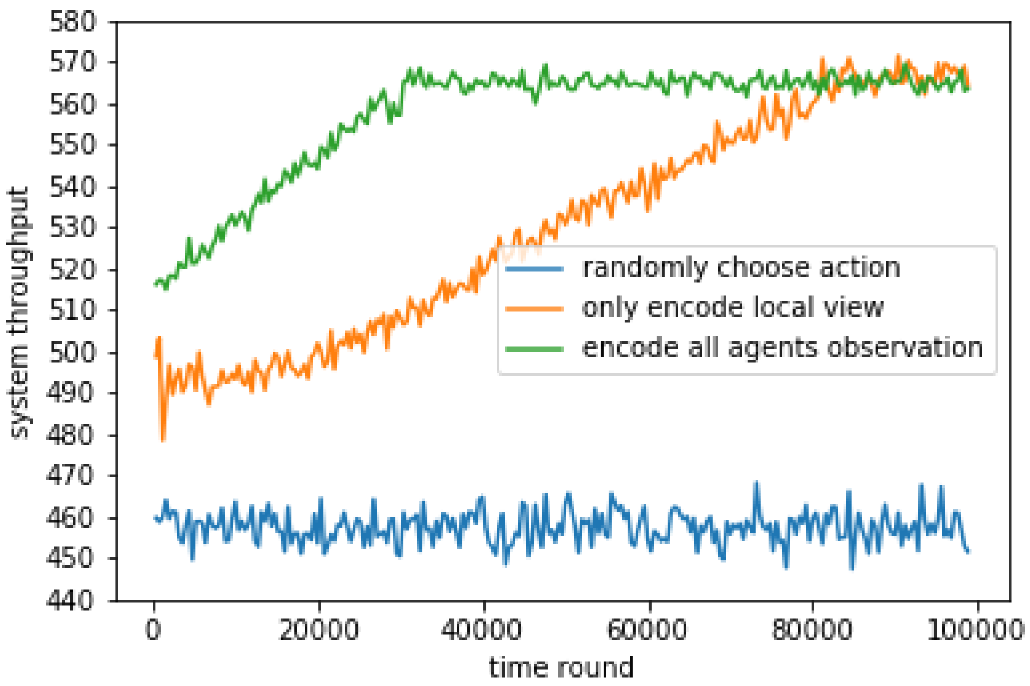
$$0 \le \sum_{f\in F}(\mathrm{v}^f x_{ij}^f) \le u_{ij} \qquad v^f \le \frac{u_{ij}}{\sum_{f\in F} x_{ij}^f}, \qquad \text{for } \forall f \in F$$

Inject Traffic

- **Situation one:**

  Inject 10 flows directly from left to right. Every node in the right side will be chosen as destination twice.



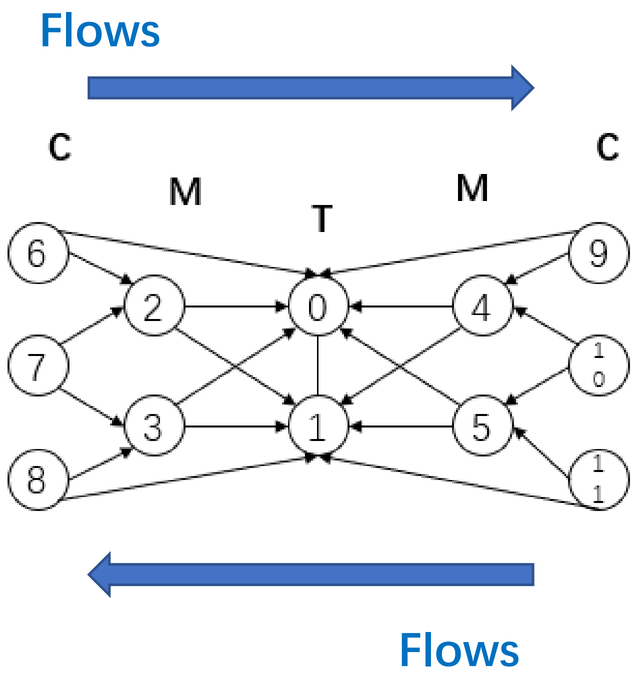


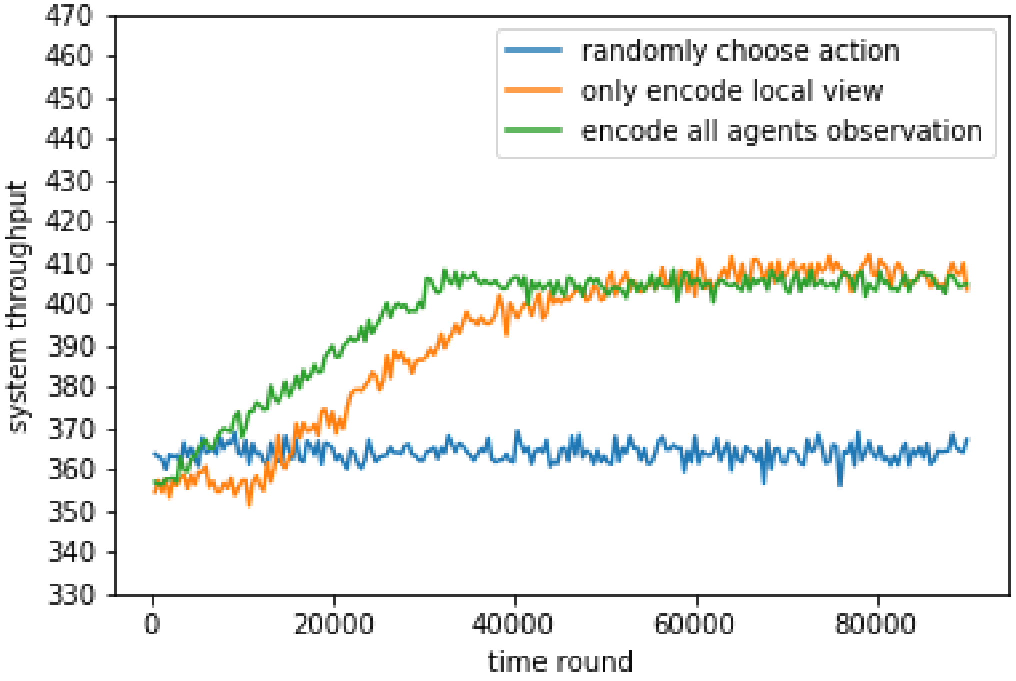
The global optimal : 750
Improvement space : 63.1%
Improvement Achieved: 24.1%

Inject Traffic

- **Situation two:**

  Inject 10 flows with 10 different destinations (**C** and **M** type nodes).

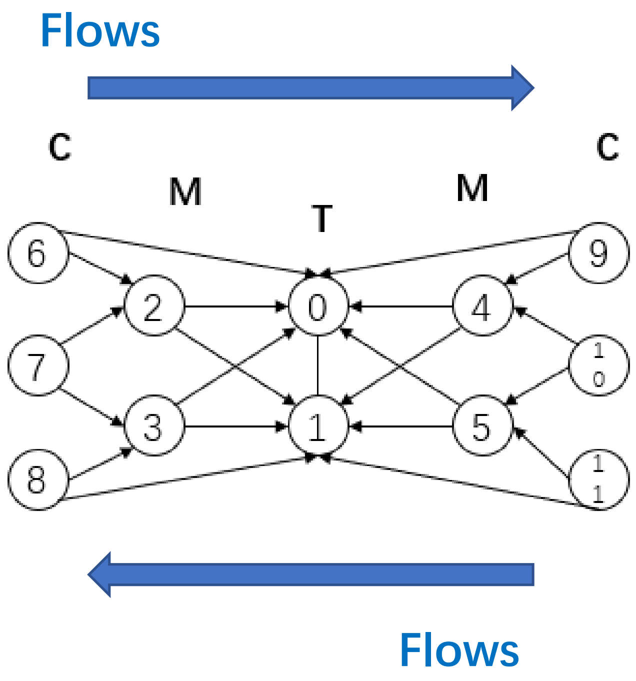


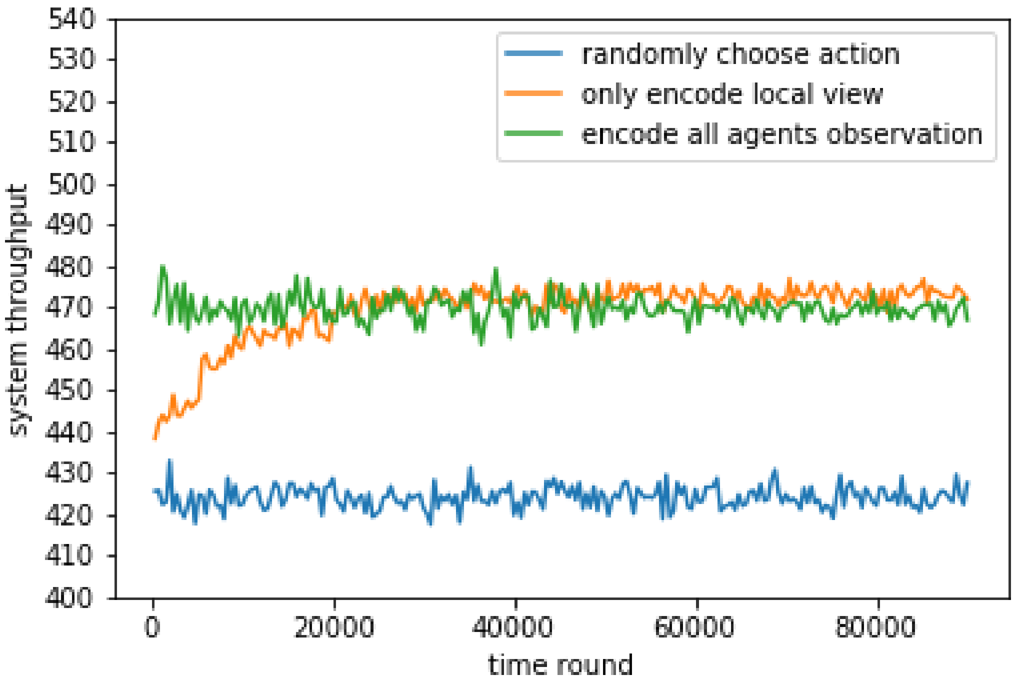


The global optimal : 533
Improvement space : 45.5%
Improvement Achieved: 12.3%

Inject Traffic

- **Situation three:**
  Inject 14 flows(respectively more) with different destinations.



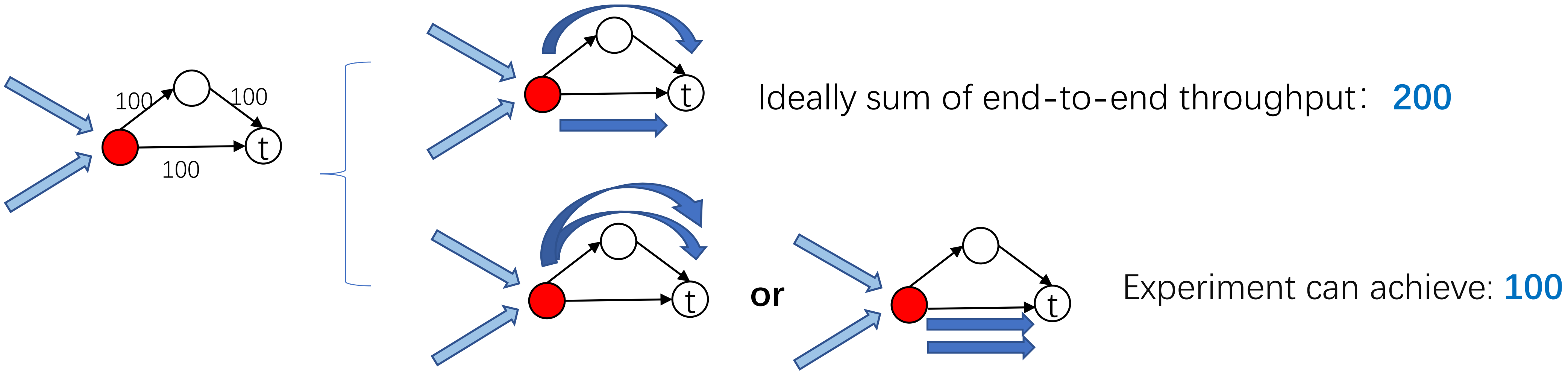


The global optimal : 600
Improvement space : 41.2%
Improvement Achieved: 11.7%

The gap between *computing optimal* and *RL results*

	10 flows with overlapped destinations	10 flows with different destinations	14 flows with different destinations
Global optimal	750	533	600
Improvement space	63.1%	45.5%	41.2%
Improvement achieved	24.1%	12.3%	11.7%

- The computing of global optimal based on ideally choosing path for a specific flow, but for experiment, it will choose the same next-hop for all flows with same destination.



Distributed Algorithms

- Extend basic *Ford-Fulkerson* algorithms or *preflow-push* method to solve Max-flow problem. The messages exchanged are necessary for executing the algorithm, for example in *distributed preflow-push*, messages includes:
  - **INIT HEIGHT** (have a global view of each node's distance)
  - **PUSH REQUEST** (forward volumes from a node)
  - **NEW HEIGHT** (update other nodes' distance information)

  Message exchanging brings an asynchronous executing process.

  Other works divide Graph into sub-graph and apply Genetic Algorithm to find Max flow for each. Or use Map-reduce framework to achieve parallelism. They cause a synchronous process.

- In NUM problem, most distributed manners using dual decomposition and sub-gradient methods. To solve the slow rate of convergence problem, some works try to extend distributed Newton method and prove converge property.