Балансировка загрузки в МАС

Н.В. Мальковский

VI Традиционная Молодежная Школа, 2014

Введение

- Ограничение на взаимодействие.
- Распределенное управление и принятие решений.
- Протокол консенсуса: $\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} \alpha_{ij}(x_i(t) - x_j(t))$ – не оптимальный по скорости сходимости.

Балансировка загрузки в МАС: модель

- Имеется сеть из n агентов.
- $Q = \{q_i\}_{i=1}^n, q_i$ – изначальная загрузка агента i.
- Связи имеют ограниченные пропускные способности. Будем считать, что эти связи задаются матрицей $C = \{C_{ij}\}_{i,j=1}^n$, $C_{ij} \geq 0$ отлично от нуля только в том случае, если связь существует и обозначает в этом случае пропускную способность связи от агента i агенту j.
- $P = \{p_i\}_{i=1}^n, p_i$ – производительность агента i.

Балансировка загрузки в МАС: постановка задачи

- Рассматривается задача нахождения оптимального управляющего воздействия: нужно найти пару (τ, F), которая минимизирует τ и удовлетворяет:

$$\begin{cases} Q - \tau((F - F^T)e_n + P) \leq 0_n \\ Q - \tau(F - F^T)e_n \geq 0_n \\ 0 \leq F_{ij} \leq C_{ij} \end{cases} \tag{1}$$

- $F = \{F_{ij}\}_{i,j=1}^{n}$ – управляющее воздействие
- $e_n = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$

Декомпозиция

- При фиксированном τ - нахождение управления F, удовлетворяющего (1), можно свести к задаче нахождения максимального потока в сети.
- Если (τ^*, F^*) удовлетворяют (1), то $\forall \tau \geq \tau^* \quad (\tau, \frac{\tau^*}{\tau} F^*)$ также удовлетворяют (1).
- Таким образом, для минимизации τ можно использовать бинарный поиск с решением задачи максимального потока на каждом шаге.

Сведение к потокам(1-3)

- Рассмотрим взвешенный орграф $G =< V, E >$. Веса соответствуют ограниченным пропускным способностям: $capacity : E \to \mathbb{R}_+$.
- $V = s, t, v_1, v_2, \dots, v_n$, v_i – вершина, соответствующая агенту i, s и t - фиктивные вершины истока и стока соответственно.
- E_c – связи между агентами.
- E_s – множество фиктивных ребер (s, v_i) $\forall i$
- E_t – множество фиктивных ребер (v_i, t) $\forall i$

- При фиксированном τ пропускные способности задаются следующим образом:

$$capacity(e) = \begin{cases} \tau * C_{ij} & , \quad e \in E_c \\ q_i & , \quad e = (s, v_i) \in E_s \\ \tau * p_i & , \quad e = (v_i, t) \in E_t \end{cases} \tag{2}$$

Сведение к потокам(3-3)

- Результатом решения задачи максимального потока является функция f:
  - $\forall\ v \in V \setminus \{s, t\} \quad \sum_{e \in in(v)} f(e) = \sum_{e \in out(v)} f(e)$
  - $\forall\ e \in E\ f(e) \le c(e)$
  - $val(f) = \sum_{e \in out(s)} f(e) - \sum_{e \in in(s)} f(e) = \sum_{e \in in(t)} f(e) - \sum_{e \in out(t)} f(e) \to max$ – величина потока

Эквивалентность управления и функции потока

- Лемма: для данного τ существует управление F, удовлетворяющее (1) $\Longleftrightarrow$ $\exists f: \quad val(f) = \sum_{i=1}^{n} q_i$
- Замечание: такой поток будет максимальным, так как $\sum_{i=1}^{n} q_i = \sum_{e \in out(s)} capacity(e) \geq val(f)$

Алгоритмы preflow-push

- Preflow-push – общая концепция решения задачи максимального потока (A. V. Goldberg, R. E. Tarjan 1986).
- Основная идея: решение можно получить последовательным применением в любом порядке двух основных операций (push, relabel).
- Операция push затрагивает вершину и одного из ее соседей, операция relabel затрагивает только вершину.
- Preflow-push имеет естественную мультиагентную реализацию.

- Количество итераций двоичного поиска оценивается как $\log_2(\frac{1}{\epsilon})$, где ϵ - искомая точность для τ.
- При произвольном выполнении операций push, relabel в худшем случае будет совершено $\mathcal{O}(n^2 m)$ операций, где m - число ребер.
- Общая сложность: $\mathcal{O}(n^2 m \log_2(\frac{1}{\epsilon}))$

Недостатки модели

- Задержки при коммуникации:
    - Задержки канала связи.
    - Время передачи одной задачи.
- Помехи и неопределенности:
    - Неточности в измерениях P, Q и C.

Задержки коммуникации

- В случае небольших задержек коммуникации, решение (1) все еще дает управление, близкое к оптимальному.
- Пусть δ_{ij} – задержка при передачи данных от i-ого агента j-ому. $\Delta = \max \delta_{ij}$.
- Пусть (τ, F) – оптимальное решение (1), тогда при наличии задержек стратегия F ведет к времени выполнения $\tilde{\tau} \leq \tau + n\Delta$

Неточности P, Q, C

- $|\tilde{F}_{ij} - F_{ij}| < \epsilon_c F_{ij}, |\tilde{p}_i - p_i| < \epsilon_p p_i, |\tilde{q}_i - q_i| < \epsilon_q q_i$, τ – оптимальное время для (1)
- $\Rightarrow$: $\tilde{\tau} < \tau \max(\frac{1}{1-\epsilon_c}, \frac{1}{1-\epsilon_p}, 1 + \frac{\epsilon_q}{1-\epsilon_p} \max \frac{q_i}{\tilde{p}_i})$

Спасибо за внимание!

Эквивалентность управления и функции потока

- "$\Rightarrow$": Пусть F – стратегия, удовл. (1), тогда:
  $f(e) =$
  $$\begin{cases} \tau * F_{ij} & , \quad e \in E_c \\ q_i & , \quad e = (s, v_i) \in E_s \\ q_i - \tau(\sum_{j=1}^{n} F_{ij} - \sum_{j=1}^{n} F_{ji}) & , \quad e = (v_i, t) \in E_t \end{cases}$$
  Равенство входящего и исходящего потока гарантируется выбором функции потока.
  Ограничение на функцию потока $0 \leq f(e) \leq capacity(e)$ для ребер из E_c следует из третьего неравенства (1), для ребер из E_t – из первых двух неравенств.

Эквивалентность управления и функции потока

- "$\Leftarrow$": Пусть f – поток, $val(f) = \sum_{i=1}^{n} q_i$, тогда:
  $f(s \to v_i) = capacity(s \to v_i) = q_i$
  При $F_{ij} = \frac{f(v_i \to v_j)}{t}$
  $0 \geq f(v_i \to t) - capacity(v_i \to t) =$
  $q_i - \sum_{e \in out(t) \cap E_c} f(e) + \sum_{e \in in(t) \cap E_c} f(e) - p_i =$
  $[Q - \tau((F - F^T)e_n + P)]^T e_n^i$
  $0 \leq f(v_i \to t) = q_i - \sum_{e \in out(t) \cap E_c} f(e) +$
  $\sum_{e \in in(t) \cap E_c} f(e) = [Q - \tau(F - F^T)e_n]^T e_n^i \blacksquare$

Устойчивость при задержках

- Канал состоит из двух socket'ов, по одному i-ый агент отправляет данные, а с другого j-ый получает с задержкой δ_{ij}.
- Назовем канал $i \to j$ *насыщенным* в момент времени t, если в некоторой окрестности t по каналу передаются данные, при этом j-ый агент получает (а i-ый отправляет) F_{ij} задач за единицу времени.
- *Если в момент времени t по каналу $i \to j$ начинается передача данных, то в момент времени $t + \delta_{ij}$ канал становится насыщенным.

Устойчивость при задержках: поведение агента

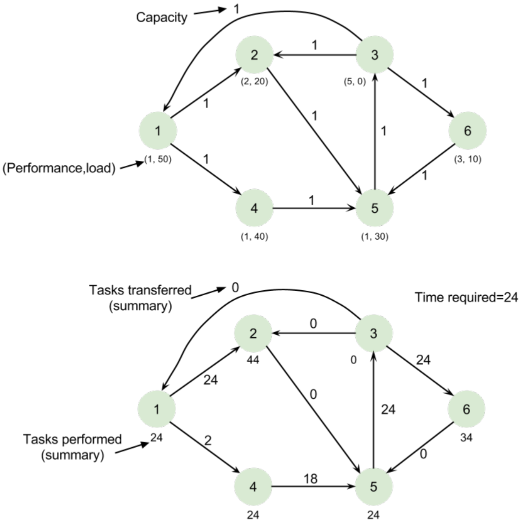
- Стратегия для агента:
  - Каждый агент знает τ.
  - Исходящие каналы насыщаются пропорционально F_{ij} (Существенно при зашумленных измерениях C/P/Q).
  - По каналу $(i \to j)$ должно быть передано не более τF_{ij} задач.

Устойчивость при задержках: определения

- Назовем агента i *стабильным*, если все входящие каналы насыщены или $\sum_{j=1}^{n}(F_{ij} - F_{ji}) > 0$.
- Пусть $E_{sat}(t)$ – множество всех насыщенных каналов в момент времени t, $V_{act}(t)$ – множество всех стабильных агентов в момент времени t.
- *Заметим, что $V_{act}(0) = \{v_i \in V | \sum_{j=1}^{n}(F_{ij} - F_{ji}) > 0\}$, т.е. множество агентов с избытком задач. $E_{sat}(0) = \emptyset$.

Устойчивость при задержках: отсутствие циклов

- Пусть в графе отсутствуют циклы на ребрах $\{e \in E | F_e > 0\}$, тогда к моменту времени Δn все агенты станут стабильными, а каналы – насыщенными.
- Док-во: Отсутствие циклов значит, что относительно данных ребер существует топологическая сортировка. Таким образом, если агент имеет номер k в некоторой топологической сортировке, то к моменту времени Δk он станет стабильным.


Capacity
1
1
1
1
1
1
1
1
1
1
1
2
3
1
6
4
5
(2, 20)
(5, 0)
(1, 50)
(3, 10)
(Performance,load)
(1, 40)
(1, 30)
Tasks transferred (summary)
0
Time required=24
0
2
3
44
0
24
24
24
0
1
6
24
2
34
Tasks performed (summary)
0
18
4
5
24
24