Achieving Linear Convergence with Parameter-Free Algorithms in Decentralized Optimization

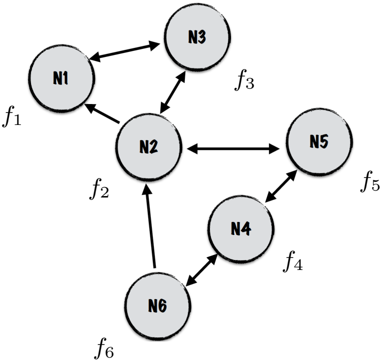
Ilya Kuruzov, Gesualdo Scutari, Alexander Gasnikov

NeurIPS 2024

Decentralized Optimization

$$\min_{x \in \mathbb{R}^d} f(x) \triangleq \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

Mesh Networks (M-Nets)



- Each agent i has access only to f_i
  - f_i is L smooth and μ-strongly convex, $\mu > 0$
- The graph network is connected
  - each agent can communicate only with its immediate neighbors

Decentralized algorithms: each agent interleaves local computations with neighboring communications

On the Choice of the Stepsize

Convergence relays sensibly on the tuning of the stepsize

- **Theory:** Upper bounds
  - require knowledge of global optimization & network parameters, not available locally
  - are quite conservative
- **Practice:**
  - manual tuning is not practical and experiment dependent
  - algorithm performance are quite sensitive to variations of the stepsize

On the Choice of the Stepsize

Convergence relays sensibly on the tuning of the stepsize

- **Theory:** Upper bounds
  - require knowledge of global optimization & network parameters, not available locally
  - are quite conservative
- **Practice:**
  - manual tuning is not practical and experiment dependent
  - algorithm performance are quite sensitive to variations of the stepsize

Open question: Can one perform adaptive stepsize tuning in decentralized algorithms?

Decentralized Setting: Why is Not so Trivial?

Decentralizing the backtracking procedure

Warmup: Backtracking (centralized)

- Algorithm update: $x^{t+1} = x^t + \gamma^t d^t$
- Strict descent direction: $\nabla f(x^t)^\top d^t < 0$
- Backtracking: largest $\gamma^t \in (0, 1] : f(x^t + \gamma^t d^t) \leq f(x^t) + c \cdot \gamma^t \nabla f(x^t)^\top d^t$

Decentralized Setting: Why is Not so Trivial?

Decentralizing the backtracking procedure

Warmup: Backtracking (centralized)

- Algorithm update: $x^{t+1} = x^t + \gamma^t d^t$
- Strict descent direction: $\nabla f(x^t)^\top d^t < 0$
- Backtracking: largest $\gamma^t \in (0, 1]$: $f(x^t + \gamma^t d^t) \leq f(x^t) + c \cdot \gamma^t \nabla f(x^t)^\top d^t$

Decentralized setting:

- How do define such a direction d_i^t at the agent's sides?
- Some dependence of d_i^t on the network is expected – hard to postulate!
- Which *local* surrogate of f for each d_i^t to be strictly descent?

Decentralized Setting: Why is Not so Trivial?

Decentralizing the backtracking procedure

Warmup: Backtracking (centralized)

- Algorithm update: $x^{t+1} = x^t + \gamma^t d^t$
- Strict descent direction: $\nabla f(x^t)^\top d^t < 0$
- Backtracking: largest $\gamma^t \in (0, 1] : f(x^t + \gamma^t d^t) \leq f(x^t) + c \cdot \gamma^t \nabla f(x^t)^\top d^t$

Decentralized setting:

- How do define such a direction d_i^t at the agent's sides?
- Some dependence of d_i^t on the network is expected – hard to postulate!
- Which *local* surrogate of f for each d_i^t to be strictly descent?

Contributions:

- Decentralized *adaptive* method via *operator splitting*
- Adaptive stepsize via local backtracking
- Linear convergence guarantees, compare favorably with nonadaptive methods