

半分布式forward-reflected-Douglas-Rachford分裂算法求解广义纳什平衡点

潘晓伟¹, 刘忠信^{1,2†}, 陈增强^{1,2}

(1. 南开大学 人工智能学院, 天津 300350;

2. 南开大学 智能机器人技术重点实验室, 天津 300350)

摘要: 针对多个体参与的广义纳什平衡点的求解问题, 已有算法通常都是基于两算子分裂算法forward-backward splitting. 本文基于三算子分裂算法forward-reflected-Douglas-Rachford (FRDR) splitting, 提出一种半分布式的FRDR算法. 半分布式旨在强调对偶变量的信息交换总是按照分布式的方式进行. 该算法有如下特性: 可以实现邻点映射和投影映射分别计算; 不需要假设伪梯度映射是协强制的或者强单调的; 通过存储上一轮交换的信息, 可以做到所需信息在每一轮迭代中只进行一次交换. 同时, 论文给出了有关迭代残差的收敛速率, 并通过数值仿真验证了所提算法的有效性.

关键词: 广义纳什平衡点; 半分布式; 三算子分裂算法; 多智能体系统

引用格式: 潘晓伟, 刘忠信, 陈增强. 半分布式forward-reflected-Douglas-Rachford分裂算法求解广义纳什平衡点. 控制理论与应用, 2022, 39(10): 1946 – 1951

DOI: 10.7641/CTA.2022.10895

Semi-distributed forward-reflected-Douglas-Rachford splitting method for generalized Nash equilibrium seeking

PAN Xiao-wei¹, LIU Zhong-xin^{1,2†}, CHEN Zeng-qiang^{1,2}

(1. College of Artificial Intelligence, Nankai University, Tianjin 300350, China;

2. Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300350, China)

Abstract: To deal with generalized Nash equilibrium seeking problems for multiple agents, existing methods are designed mainly based on the two-operator splitting approach, e.g. forward-backward splitting. In this paper, a semi-distributed method is proposed by turning to the three-operator splitting procedure, i.e. forward-reflected-Douglas-Rachford splitting (FRDR). It means that the information of dual variables are interchanged in a distributed way. The semi-distributed FRDR method has the following three characteristic. First, the computation of proximal mapping and projection mapping is separated. Second, the pseudo-gradient mapping is no longer assumed to be cocoercive or strongly monotone. Third, with the storage of interchanged information from the last round, all needed information is interchanged only once at each iterating round. In addition, the convergence rate is given in terms of iteration residual. Simulation results are also provided to illustrate the effectiveness of the method.

Key words: generalized Nash equilibrium; semi-distributed; three-operator splitting method; multi-agent systems

Citation: PAN Xiaowei, LIU Zhongxin, CHEN Zengqiang. Semi-distributed forward-reflected-Douglas-Rachford splitting method for generalized Nash equilibrium seeking. *Control Theory & Applications*, 2022, 39(10): 1946 – 1951

1 引言

纳什平衡问题(Nash equilibrium problems, NEP)由John Nash在文献[1–2]中正式提出. 广义纳什平衡问题(generalized Nash equilibrium problems, GNEP)

是NEP的拓展, 由Gerard Debreu在文献[3–4]中正式提出, 其特点是每个参与者的约束集合也依赖于其他参与者. GNEP在许多领域中都有所应用, 如电信系统、环境污染控制等, 参考综述文献[5].

收稿日期: 2021–09–22; 录用日期: 2022–04–14.

[†]通信作者. E-mail: lzhx@nankai.edu.cn; Tel.: +86 22-85358276.

本文责任编辑: 朱善迎.

天津市自然科学基金项目(20JCYBJC01060), 国家自然科学基金项目(62103203, 61973175)资助.

Supported by the Natural Science Foundation of Tianjin (20JCYBJC01060) and the National Natural Science Foundation of China (62103203, 61973175).

近年来, 基于算子分裂框架的求解方法正在得到越来越多的关注, 其基本思路是将GNEP转化为算子的寻零问题, 然后应用不同的算子分裂方法, 可以得到相应的求解算法^[6]. 基于两算子分裂方法的有: forward-backward (half) forward splitting^[7-8, 12], forward-backward splitting^[9-10], forward-reflected-backward splitting^[13]等.

基于 forward-backward-forward splitting 的算法^{[7], [8, 算法2]}, 每一轮迭代需要进行两次信息交换; 基于 forward-backward-half forward splitting 的算法^{[8, 算法3], [12, Eqn. (6.11)]}和基于 forward-backward splitting 的算法^[9], 需要假设伪梯度映射是强单调的, 而另一基于 forward-backward splitting 的算法^[10], 则需要假设伪梯度映射是协强制的; 基于 forward-reflected-backward splitting 的算法^[13, 算法5], 其每一轮迭代只进行一次信息交换, 而且只需要假设伪梯度映射是单调的和李普希茨的.

其他的求解算法有连续时间算法^[11], 邻近点算法(proximal point algorithm, PPA)^{[13, 算法6], [14]}, 交替方向乘子法(alternating direction method of multipliers, ADMM)^[15]等. 文献[11]中的算法要求伪梯度映射是严格单调的; 文献[13]中的算法6只是针对一类具有特殊结构的聚合博弈设计的, 其只需要假设伪次梯度映射是单调的; 文献[14]中的算法需要假设代价函数是可导的, 而在GNEP中会有非光滑项; 文献[15]中的算法4.1经过变换之后, 其实是一种forward-backward算子分裂方法.

根据上述内容可知, 在基于算子分裂设计的方法中, 要么对伪梯度映射均有较强的假设条件, 要么每次迭代需要交换两次信息.

针对前述问题, 本文的主要工作在于, 基于三算子分裂算法forward-reflected-Douglas-Rachford(FRDR)算法^[17], 提出一种半分布式的FRDR算法. 该算法有如下特性: 可以实现邻点映射和投影映射分别进行计算, 因此更适用于分开计算时可以得出具体表达式的情形, 而省去解决一个优化子问题; 不需要假设伪梯度映射是协强制的或者强单调的; 通过存储上一轮交换的信息, 可以做到所需信息在每一轮迭代中只进行一次交换; 同时, 也给出了迭代残差的收敛速率, 即 $o(1/(k+1))$. 跟本文算法比较相似的是基于 forward-reflected-backward splitting 的算法^[13, 算法5], 区别在于FRDR算法中邻点映射和投影映射的计算是分开进行的; 此外, 文献[13]中的算法5是针对聚合博弈设计的, 而本文算法是针对一般的GNEP设计的.

本文的主要结构如下. 第2节包括后续分析所需的预备知识以及对所研究问题的描述. 第3节给出了半分布式FRDR算法的设计及其收敛性分析. 第4节给出了仿真例子用来验证算法的有效性. 最后是全文总结.

2 预备知识和问题描述

令 $\mathbb{R}^m$ 表示 m 维欧式空间, $\langle \cdot, \cdot \rangle$ 是内积. I_m 表示 $m \times m$ 维单位矩阵. $A \otimes B$ 表示矩阵 A 和 B 的克罗内克积. 对于向量 x , $\|x\|$ 表示欧式范数; 对于矩阵 C , $\|C\|$ 表示导出范数. 文中出现的向量, 除非特别说明, 均指列向量. $(x_1, \dots, x_n)$ 表示由列向量 $x_1, \dots, x_n$ 等按列排列生成的列向量. $\text{diag}\{A_1, \dots, A_n\}$ 表示由矩阵 $A_1, \dots, A_n$ 生成的分块对角阵. 0_m 表示 m 维零向量.

2.1 算子理论及图论

集合 $\mathcal{S} \subset \mathbb{R}^m$ 是凸的, 若 $\forall x, y \in \mathcal{S}, \forall 0 < \alpha < 1$, 有 $\alpha x + (1 - \alpha)y \in \mathcal{S}$. $\text{int } \mathcal{S}$ 表示集合 $\mathcal{S}$ 的内点. 非空凸集 $\mathcal{S}$ 在 $x \in \mathcal{S}$ 处的法锥定义为 $N_{\mathcal{S}}(x) := \{\xi \in \mathbb{R}^m | \langle y - x, \xi \rangle \leq 0, \forall y \in \mathcal{S}\}$; 若 $x \notin \mathcal{S}$, 则 $N_{\mathcal{S}}(x) := \emptyset$. 点 $x \in \mathbb{R}$ 到非空闭凸集 $\mathcal{S}$ 的投影映射定义为 $\text{proj}_{\mathcal{S}}(x) := \{p \in \mathcal{S} | \langle y - p, x - p \rangle \leq 0, \forall y \in \mathcal{S}\}$. 函数 $f: \mathbb{R}^m \rightarrow \mathbb{R}$ 是凸的, 若其上图 $\text{epi } f := \{(x, u) \in \mathbb{R}^m \times \mathbb{R} | f(x) \leq u\}$ 是凸集, 其次微分定义为 $\partial f(x) := \{s \in \mathbb{R}^m | f(y) \geq f(x) + \langle y - x, s \rangle, \forall y \in \mathbb{R}^m\}$, 其邻点映射定义为 $\text{prox}_{\gamma f}(x) := \arg \min_{y \in \mathbb{R}^m} \{f(y) + \frac{1}{2\gamma} \|x - y\|^2\}$.

对于集值算子 $F: \mathbb{R}^m \rightrightarrows \mathbb{R}^m$, 其图像定义为 $\text{gph } F := \{(x, u) \in \mathbb{R}^m \times \mathbb{R}^m | y \in F(x)\}$. 算子 F 是单调的, 若 $\forall (x, u), (y, v) \in \text{gph } F$, 有 $\langle x - y, u - v \rangle \geq 0$ 是极大单调的, 若 F 是单调的且不存在单调算子使得其图像能够真包含 $\text{gph } F$ 是严格单调的, 若 $\forall (x, u), (y, v) \in \text{gph } F$ 且 $x \neq y$, 有 $\langle x - y, u - v \rangle > 0$ 是强单调的, 若 $F - \sigma \text{Id}$ 是单调的, 其中 $\sigma > 0$. 算子 F 的预解算子定义为 $J_{\gamma F} := (\text{Id} + F)^{-1}$, 其中 Id 是单位算子. 算子 F 的零点定义为 $\text{zer}(F) := \{x \in \mathbb{R}^m | 0 \in F(x)\}$. 单值算子 $T: \mathbb{R}^m \rightarrow \mathbb{R}^m$ 是协强制的, 若 $\forall x, y \in \mathbb{R}^m$, 有 $\langle x - y, T(x) - T(y) \rangle \geq \eta \|T(x) - T(y)\|^2$, 其中 $\eta > 0$ 是李普希茨的, 若 $\forall x, y \in \mathbb{R}^m$, 有 $\|T(x) - T(y)\| \leq \varsigma \|x - y\|$, 其中 $\varsigma > 0$.

以上内容均参照文献[18]给出. 以下为图论相关内容.

智能体之间的通信拓扑可由 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 描述, 其中 $\mathcal{V} = \{1, \dots, n\}$ 是顶点集合, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ 是连接边的集合. 令 $c_{ij} \geq 0$ 表示边 $(i, j) \in \mathcal{E}$ 上的权重, 若 $c_{ij} = c_{ji}$ 则表示 $\mathcal{G}$ 是无向图. 定义图 $\mathcal{G}$ 的拉普拉斯阵为 $L_c := [l_{ij}]_{i,j \in \mathcal{V}}$, 其中 $l_{ij} = -c_{ij}$, 当 $i \neq j$; $l_{ii} = \sum_{j=1}^n c_{ij}$. 令 $\mu_{\max}$ 表示 L_c 的最大特征值. 图 $\mathcal{G}$ 是连通的当且仅当 L_c 的零特征值是单根.

2.2 问题描述

考虑 n 个智能体参与的广义纳什平衡问题, 智能体之间的通信拓扑为 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. 对于 $\forall i \in \mathcal{V}$, 其模型

描述为

$$\begin{cases} \min_{x_i \in \mathbb{R}^{m_i}} f_i(x_i, x_{-i}) + g_i(x_i) \\ \text{s.t. } \sum_{j=1}^n A_j x_j = \sum_{j=1}^n b_j, x_i \in \mathcal{S}_i, \end{cases} \quad (1)$$

其中: $f_i: \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n} \rightarrow \mathbb{R}$, $g_i: \mathbb{R}^{m_i} \rightarrow \mathbb{R}$, $A_i \in \mathbb{R}^{m \times m_i}$. 为表示方便, 令 $\mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, $x := (x_1, \cdots, x_n)$, $\tilde{A} := (A_1 \cdots A_n)$, $\tilde{b} := \sum_{j=1}^n b_j$, $\mathcal{R}_i(x_{-i}) := \{x_i | \tilde{A}x = \tilde{b}, x \in \mathcal{S}\}$.

定义 1 联合策略 $x^* = (x_1^*, \cdots, x_n^*)$ 是博弈(1)的广义纳什平衡点, 如果对于 $\forall i \in \mathcal{V}$, 都有如下关系成立:

$$x_i^* \in \arg \min_{x_i \in \mathcal{R}_i(x_{-i}^*)} \{f_i(x_i, x_{-i}^*) + g_i(x_i)\}.$$

假设 1 对于 $\forall i \in \mathcal{V}$, 给定 x_{-i} , $f_i(\cdot, x_{-i})$ 是凸的, 连续可导的, 且其导数关于 x 是连续的; g_i 是正常且下半连续的凸函数; 集合 $\mathcal{S}_i$ 是闭凸集.

假设 2 $\text{int } \mathcal{S} \cap \{x \in \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n} | \tilde{A}x = \tilde{b}\} \neq \emptyset$.

本文旨在求解一类特殊的GNE, 即变分平衡点 (variational GNE, v-GNE). 令 $\Psi(x, y) := \sum_{i=1}^n [f_i(x_i, x_{-i}) + g_i(x_i) - f_i(y_i, x_{-i}) - g_i(y_i)]$ 表示GNEP(1)的Nikaido-Isoda函数, 则称满足 $\sup_{y \in \mathcal{R}} \Psi(x^*, y) = 0$ 的 $x^* \in \mathcal{R}$ 为变分平衡点, 其中 $\mathcal{R} := \{x \in \mathcal{S} | \tilde{A}x = \tilde{b}\}$. 根据所有的 f_i , 定义其伪梯度 $\tilde{F}(x): \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n} \rightarrow \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n}$ 为

$$\tilde{F}(x) := \begin{pmatrix} \nabla_{x_1} f_1(x_1, x_{-1}) \\ \vdots \\ \nabla_{x_n} f_n(x_n, x_{-n}) \end{pmatrix}. ==$$

同时, 令 $g(x) := \sum_{i=1}^n g_i(x_i)$, $\tilde{w} := (x, \tilde{\lambda})$, $h(\tilde{w}) := g(x)$, $\tilde{m} := \sum_{i=1}^n m_i$, 可得

$$\partial g(x) = \begin{pmatrix} \partial g_1(x_1) \\ \vdots \\ \partial g_n(x_n) \end{pmatrix}, \quad \partial h(\tilde{w}) = \begin{pmatrix} \partial g(x) \\ \{0_{\tilde{m}}\} \end{pmatrix}.$$

此外, 令

$$\tilde{F}_e(\tilde{w}) := \begin{pmatrix} \tilde{F}(x) \\ \{0_{\tilde{m}}\} \end{pmatrix}, \quad \tilde{C}(\tilde{w}) := \begin{pmatrix} \tilde{A}^T \tilde{\lambda} \\ \tilde{b} - \tilde{A}x \end{pmatrix}.$$

那么, 在假设1和假设2成立的条件下, 由文献[15]中的定理3.3和引理3.5可知, x^* 是v-GNE当且仅当, 存在Lagrange乘子 $\tilde{\lambda}^*$, 满足

$$0 \in \partial h(\tilde{w}^*) + \tilde{F}_e(\tilde{w}^*) + \tilde{C}(\tilde{w}^*) + N_{\mathcal{S} \times \mathbb{R}^{\tilde{m}}}(\tilde{w}^*),$$

其中 $\tilde{w}^* := (x^*, \tilde{\lambda}^*)$.

假设 3 伪梯度映射 $\tilde{F}$ 是单调的.

假设 4 智能体之间的通信连接图 $\mathcal{G}$ 是无向连通的.

令 λ_i 表示智能体 i 对 $\tilde{\lambda}$ 的估计, y_i 为引入的辅助变量. 再令 $\lambda := (\lambda_1, \cdots, \lambda_n)$, $y := (y_1, \cdots, y_n)$, $w := (x, y, \lambda)$, $b := (b_1, \cdots, b_n)$, $A := \text{diag}\{A_1, \cdots, A_n\}$, $L := L_c \otimes I_m$, L_c 表示定义在图 $\mathcal{G}$ 上的拉普拉斯阵. 为了对约束进行解耦, 采用文献[9]中的解耦方式, 定义

$$F(w) := \begin{pmatrix} \tilde{F}(x) \\ \{0_{\tilde{m}}\} \\ L\lambda + b \end{pmatrix}, \quad C(w) := \begin{pmatrix} A^T \lambda \\ L\lambda \\ -Ax - Ly \end{pmatrix},$$

$$G(w) := \begin{pmatrix} \partial g(x) \\ \{0_{\tilde{m}}\} \\ \{0_{\tilde{m}}\} \end{pmatrix}.$$

那么, 可以得到与文献[9]中定理2类似的结论.

引理 1 在假设1-4均成立的条件下, 有如下结论:

1) 若

$$w^* := (x^*, y^*, \lambda^*) \in \text{zer}(G + F + C + N_{\mathcal{S} \times \mathbb{R}^{\tilde{m}} \times \mathbb{R}^{\tilde{m}}}),$$

则有 $\lambda^* = 1_n \otimes \tilde{\lambda}^*$, 且 $(x^*, \tilde{\lambda}^*) \in \text{zer}(\partial h + \tilde{F}_e + \tilde{C} + N_{\mathcal{S} \times \mathbb{R}^{\tilde{m}}})$. 因此, x^* 是GNEP(1)的v-GNE.

2) $\text{zer}(G + F + C + N_{\mathcal{S} \times \mathbb{R}^{\tilde{m}} \times \mathbb{R}^{\tilde{m}}}) \neq \emptyset$.

引理1的证明与文献[16]中引理1的证明类似, 此处不再给出.

3 算法设计与分析

为表示方便, 令 $B := F + C$, $\tilde{\mathcal{S}} := \mathcal{S} \times \mathbb{R}^{\tilde{m}} \times \mathbb{R}^{\tilde{m}}$. 通过引理1, 求解式(1)的v-GNE的问题可转化为求解满足 $0 \in G(w) + B(w) + N_{\tilde{\mathcal{S}}}(w)$ 的 w . 这是关于3个算子的寻零问题, 因此, 可以应用文献[17]中的算法进行求解

假设 5 伪梯度映射 $\tilde{F}$ 是 ρ -Lipshitz连续的.

应用 forward-reflected-Douglas-Rachford 分裂算法[17]可得

$$\begin{cases} w_{k+1} = J_{\gamma G}(w_k - \gamma u_k - \gamma(2B(w_k) - B(w_{k-1}))), \\ v_{k+1} = J_{\beta N_{\tilde{\mathcal{S}}}}(2w_{k+1} - w_k + \beta u_k), \\ u_{k+1} = u_k + \frac{1}{\beta}(2w_{k+1} - w_k - v_{k+1}). \end{cases} \quad (2)$$

令 $v := \begin{pmatrix} v^x \\ v^y \\ v^\lambda \end{pmatrix}$, $u := \begin{pmatrix} u^x \\ u^y \\ u^\lambda \end{pmatrix}$. 进一步整理可得关于 w

的更新律为

$$\begin{cases} x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma u_k^x - \gamma(2(\tilde{F}(x_k) + \\ A^T \lambda_k) - (\tilde{F}(x_{k-1}) + A^T \lambda_{k-1}))), \\ y_{k+1} = y_k - \gamma u_k^y - \gamma(2L\lambda_k - L\lambda_{k-1}), \\ \lambda_{k+1} = \lambda_k - \gamma u_k^\lambda - \gamma(2(L\lambda_k + b - Ax_k - \\ Ly_k) - (L\lambda_{k-1} + b - Ax_{k-1} - Ly_{k-1})). \end{cases}$$

关于 v 的更新律为

$$\begin{cases} v_{k+1}^x = \text{proj}_S(2x_{k+1} - x_k + \beta u_k^x), \\ v_{k+1}^y = 2y_{k+1} - y_k + \beta u_k^y, \\ v_{k+1}^\lambda = 2\lambda_{k+1} - \lambda_k + \beta u_k^\lambda. \end{cases}$$

关于 u 的更新律为

$$\begin{cases} u_{k+1}^x = u_k^x + \frac{1}{\beta}(2x_{k+1} - x_k - v_{k+1}^x), \\ u_{k+1}^y = u_k^y + \frac{1}{\beta}(2y_{k+1} - y_k - v_{k+1}^y), \\ u_{k+1}^\lambda = u_k^\lambda + \frac{1}{\beta}(2\lambda_{k+1} - \lambda_k - v_{k+1}^\lambda). \end{cases}$$

消去 v^y 和 v^λ , 可得

$$\begin{cases} u_{k+1}^y = 0, \\ u_{k+1}^\lambda = 0. \end{cases}$$

因此, 把 u^y 和 u^λ 的初值设为0, 继续化简可得

$$\begin{cases} x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma u_k^x - \gamma(2(\tilde{F}(x_k) + \\ A^T \lambda_k) - (\tilde{F}(x_{k-1}) + A^T \lambda_{k-1}))), \\ y_{k+1} = y_k - \gamma(2L\lambda_k - L\lambda_{k-1}), \\ \lambda_{k+1} = \lambda_k - \gamma(2(L\lambda_k + b - Ax_k - Ly_k) - \\ (L\lambda_{k-1} + b - Ax_{k-1} - Ly_{k-1})), \\ v_{k+1}^x = \text{proj}_S(2x_{k+1} - x_k + \beta u_k^x), \\ u_{k+1}^x = u_k^x + \frac{1}{\beta}(2x_{k+1} - x_k - v_{k+1}^x). \end{cases} \quad (3)$$

注 1 从式(3)中可以看出, 函数 g 的邻点映射与关于 S 的投影映射是分开计算的. 文献[13]中的Algorithm 5需要计算函数 $g_i + \delta_{S_i}$ 的邻点映射, 其中 δ_{S_i} 为集合 S_i 的示性函数. 因此, 当分开计算相较于合起来计算更容易时, 式(3)更为适用. 此处所指为可以得出具体表达式的情形, 而省去解决一个优化子问题. 例如, 令 $\mathcal{P}$ 表示 $\{1, \dots, m\}$ 的一个划分, $x := (\tilde{x}, t) \in \mathbb{R}^{m-1} \times \mathbb{R}$, 函数 $g(x) := \sum_{p \in \mathcal{P}} \|x_{[p]}\|$, 其中, $x_{[p]}$ 表示对应 p 的分量. 根据文献[20, §6.5.4], 可以计算 $g(x)$ 的邻点映射为

$$(\text{prox}_{\gamma g}(x))_{[p]} = \max\{0, 1 - \frac{\gamma}{\|x_{[p]}\|}\}x_{[p]}, \quad \forall p \in \mathcal{P}.$$

对于二阶锥约束 $S := \{(\tilde{x}, t) \in \mathbb{R}^m \mid \|\tilde{x}\| \leq t\}$, 根据文献

[20, §6.3.2], 可以计算其投影映射为

$$\text{proj}_S(\tilde{x}, t) = \begin{cases} 0, & \|\tilde{x}\| \leq -t, \\ (\tilde{x}, t), & \|\tilde{x}\| \leq t, \\ \frac{1}{2}(1 + \frac{t}{\|\tilde{x}\|})(\tilde{x}, \|\tilde{x}\|), & \|\tilde{x}\| \geq |t|. \end{cases}$$

若二者合在一起计算, 则需要解决一个优化子问题.

注 2 得益于FRDR算法的特性, 式(3)所代表的算法, 其收敛不需要假设伪梯度映射是强单调的或协强制的. 后续将会给出证明.

式(3)的伪代码如算法1所示.

算法 1 半分布式forward-reflected-Douglas-Rachford分裂算法

初始化: 任取 $\beta > 0, 0 < \gamma < \beta/(1 + 2\rho\beta)$, $x_1, x_0, y_1, y_0, \lambda_1, \lambda_0, u_1^x$;
对于智能体 $\forall i \in \mathcal{V}$, 从邻居 j 接收 $x_{j,0}, y_{j,0}, \lambda_{j,0}$ 并存储.

迭代更新直到收敛:

$\forall i \in \mathcal{V}$, 从邻居 j 接收 $x_{j,k}, y_{j,k}, \lambda_{j,k}$ 并存储;

$x_{i,k+1} = \text{prox}_{\gamma g_i}(\gamma \nabla_{x_i} f_i(x_{i,k-1}, x_{-i,k-1}) +$

$\gamma A_i^T \lambda_{i,k-1} + x_{i,k} - \gamma u_{i,k}^x -$

$2\gamma A_i^T \lambda_{i,k} - 2\gamma \nabla_{x_i} f_i(x_{i,k}, x_{-i,k})),$

$y_{i,k+1} = y_{i,k} - 2\gamma \sum_{j=1}^n l_{ij} \lambda_{j,k} + \gamma \sum_{j=1}^n l_{ij} \lambda_{j,k-1},$

$\lambda_{i,k+1} = \lambda_{i,k} - 2\gamma \sum_{j=1}^n l_{ij} \lambda_{j,k} - \gamma b_i +$

$2\gamma \sum_{j=1}^n l_{ij} y_{j,k} - \gamma \sum_{j=1}^n l_{ij} y_{j,k-1} -$

$\gamma A_i x_{i,k-1} + \gamma \sum_{j=1}^n l_{ij} \lambda_{j,k-1} +$

$2\gamma A_i x_{i,k},$

$v_{i,k+1}^x = \text{proj}_{S_i}(2x_{i,k+1} - x_{i,k} + \beta u_{i,k}^x),$

$u_{i,k+1}^x = u_{i,k}^x + \frac{1}{\beta}(2x_{i,k+1} - x_{i,k} - v_{i,k+1}^x),$

$k = k + 1.$

定理 1 在假设1–5均成立的情况下, 若 $\beta > 0$, 且 $0 < \gamma < \beta/(1 + 2\rho\beta)$, 则有算法1中的 x_k 收敛到v-GNE.

证

步骤 1 首先, 证明 w_k 收敛到 $\text{zer}(G + B + N_{\tilde{S}})$.

由假设1可知, G 和 $N_{\tilde{S}}$ 都是极大单调的; 由假设3和假设4可知, B 是极大单调的; 由假设5以及 F 和 C 的定义可知, B 是 ρ -Lipshitz连续的, 其中 $\rho = \rho + \mu_{\max} + \|C\|$. 根据文献[17]中的定理5.1可得, w_k 收敛到 $\text{zer}(G + B + N_{\tilde{S}})$.

步骤2 然后, 证明 x_k 收敛到v-GNE.

根据引理1, 由 w_k 收敛到 $\text{zer}(G + B + N_{\tilde{S}})$ 可得, x_k 收敛到v-GNE.

证毕.

注3 算法1之所以叫作半分布式, 是因为当目标函数依赖于所有智能体的 x_i 时, 获取全部的信息需要通过中心结点来实现; 而当其仅受邻居影响时, 该算法就是分布式的. 无论如何, 对偶变量 λ_i 的信息总是通过分布式的方式获取. 所以, 采用半分布式来命名该算法.

注4 从算法1中可以清楚地看到, 每个智能体通过存储上一轮收集到的信息, 从而实现了每一轮迭代中只进行一次信息交换. 另外, 也可以将上一轮计算的伪梯度信息存储下来, 这样的话, 每一轮迭代只需要计算一次伪梯度.

尽管算法的收敛速率并未在文献[17]中直接给出, 但是根据其定理5.1的证明可推出如下结论.

推论1 对于式(2)所产生的序列 $\{(w_k, u_k)\}$, 有如下关系成立:

$$\min_{p \leq k} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 = o\left(\frac{1}{k+1}\right).$$

证 令

$$s_p := \frac{1}{2} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 + \frac{1}{2} \|(w_p, u_p) - (w_{p-1}, u_{p-1})\|_e^2,$$

其中 $\|\cdot\|_e$ 表示定理5.1^[17]中定义的范数. 根据定理5.1^[17]的证明可知 $\sum_{p=0}^{\infty} s_p$ 有上界, 即存在 $\bar{\alpha} > 0$, 使得

$$\sum_{p=0}^{\infty} s_p < \bar{\alpha}. \text{ 因此, 可得}$$

$$\sum_{p=0}^{\infty} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 \leq 2\bar{\alpha}.$$

由

$$\min_{p \leq k} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 \leq \|(w_{k+1}, u_{k+1}) - (w_k, u_k)\|_e^2,$$

可得

$$\sum_{k=0}^{\infty} \min_{p \leq k} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 \leq 2\bar{\alpha}.$$

此外, 有

$$\min_{p \leq k+1} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 \leq \min_{p \leq k} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2.$$

根据文献[19]中的引理1第4部分可得

$$\min_{p \leq k} \|(w_{p+1}, u_{p+1}) - (w_p, u_p)\|_e^2 = o\left(\frac{1}{k+1}\right).$$

利用范数的等价性. 证毕.

4 数值仿真

考虑由编号1, $\dots$, 21共21个公司(智能体)和编

号M1, $\dots$, M6共6个市场组成的网络化古诺博弈, 其网络结构如图1所示, 其中圆形表示公司, 方形表示市场, 箭头表示公司向市场供应产品. 智能体之间的通信连接如下图2所示.

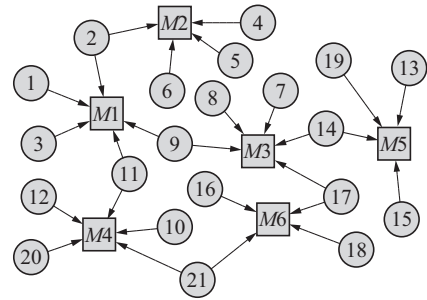


图1 网络化古诺博弈的网络结构

Fig. 1 Structure of networked Cournot game

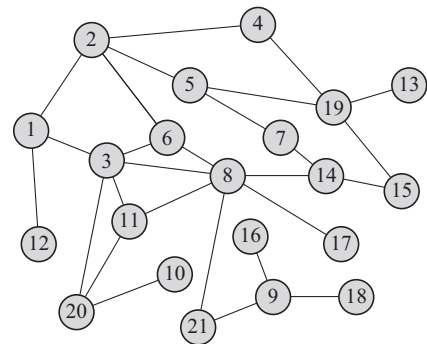


图2 智能体之间的通信拓扑

Fig. 2 Communication topology between agents

给定目标函数为

$$f_i(x_i, x_{-i}) = \sum_{j=1}^{21} (A_j x_j)^T \tilde{P} A_i x_i - \tilde{Q}^T A_i x_i,$$

$$\tilde{P} = \text{diag}\{p_1, \dots, p_6\}, \quad \tilde{Q} = \begin{pmatrix} q_1 \\ \vdots \\ q_6 \end{pmatrix},$$

$$g_i(x_i) = \alpha_i \|x_i\| + a_i^T x_i.$$

给定集合约束为 $\mathcal{S}_i = \{x_i \in \mathbb{R}^{m_i} | \underline{c}_i \leq x_i \leq \bar{c}_i\}$. A_i 可以通过图1获得, L_c 可通过图2获得, 其余常量 $b_i, \underline{c}_i, \bar{c}_i, \tilde{P}, \tilde{Q}$ 均在MATLAB[®]中随机产生. 简单起见, 令 $\alpha_i = 1, a_i = 0_{m_i}, \forall i = 1, \dots, 21$.

图3-5分别表示迭代残差, 一致性误差和约束残差的变化趋势. 从图3-5中可以看出, 迭代残差和约束残差以及一致性误差都呈现出减小的趋势, 说明 $\{x_k\}$ 是逐渐收敛的且趋向于约束集合, 同时 $\{\lambda_{i,k}\}$ 逐渐达到一致.

5 结论

本文基于三算子分裂算法 forward-reflected-Douglas-Rachford (FRDR) 算法, 提出一种半分布式的FRDR算法. 该算法有如下特性: 可以实现邻点映射

和投影映射分别进行计算, 因此在计算上更适用于分开计算简单而合起来计算困难的情形; 而且不需要假设伪梯度映射是协强制的或者强单调的; 通过存储上一轮交换的信息, 可以做到所需信息在每一轮迭代中只进行一次交换. 此外, 数值仿真验证了所提算法的有效性.

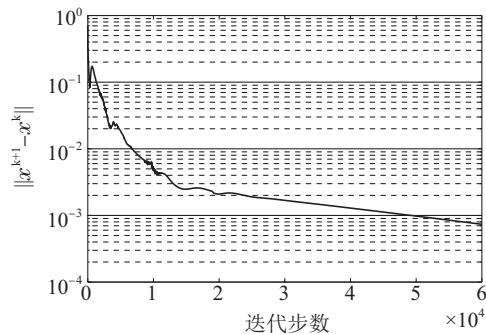


图 3 迭代残差

Fig. 3 Iteration residual

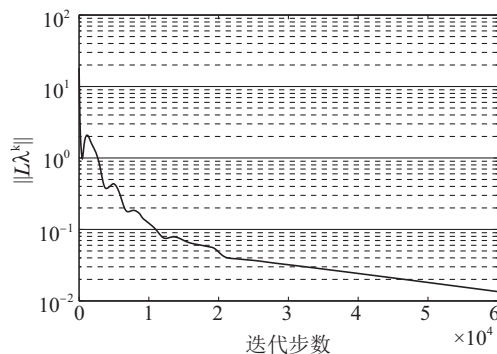


图 4 一致性误差

Fig. 4 Consensus error

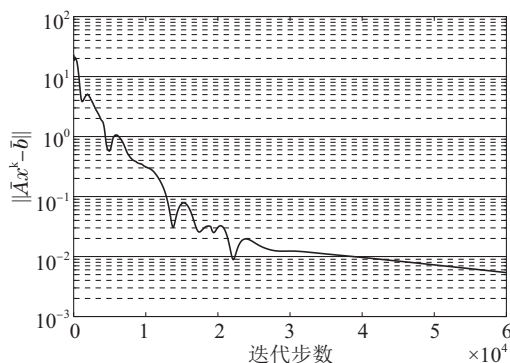


图 5 约束残差

Fig. 5 Constraint residual

参考文献:

- [1] NASH J. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 1950, 36(1): 48 – 49.
- [2] NASH J. Non-cooperative games. *Annals of Mathematics*, 1951, 54(2): 286 – 295.
- [3] DEBREU G. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 1952, 38(10): 886 – 893.
- [4] ARROW K J, DEBREU G. Existence of an equilibrium for a competitive economy. *Econometrica*, 1954, 22(3): 265 – 290.

- [5] FACCHINEI F, KANZOW C. Generalized Nash equilibrium problems. *Annals of Operations Research*, 2010, 175(1): 177 – 211.
- [6] YI P, WANG T. New directions in distributed Nash equilibrium seeking based on monotone operator theory. *Control Theory and Technology*, 2020, 18(3): 333 – 335.
- [7] BELGIOIOSO G, GRAMMATICO S. Semi-decentralized Nash equilibrium seeking in aggregative games with separable coupling constraints and non-differentiable cost functions. *IEEE Control Systems Letters*, 2017, 1(2): 400 – 405.
- [8] FRANCI B, STAUDIGL M, GRAMMATICO S. Distributed forward-backward (half) forward algorithms for generalized Nash equilibrium seeking. *Proceedings of the 18th European Control Conference (ECC)*. St. Petersburg: IEEE, 2020: 1274 – 1279.
- [9] YI P, PAVEL L. An operator splitting approach for distributed generalized Nash equilibria computation. *Automatica*, 2019, 102: 111 – 121.
- [10] BELGIOIOSO G, NEDIĆ A, GRAMMATICO S. Distributed generalized Nash equilibrium seeking in aggregative games on time-varying networks. *IEEE Transactions on Automatic Control*, 2021, 66(5): 2061 – 2075.
- [11] LIANG Yinshan, LIANG Shu, HONG Yiguang. Distributed continuous-time algorithm for Nash equilibrium seeking of nonsmooth aggregative games. *Control Theory & Applications*, 2018, 35(5): 593 – 600
(梁银山, 梁舒, 洪奕光. 非光滑聚合博弈纳什均衡的分布式连续时间算法. *控制理论与应用*, 2018, 35(5): 593 – 600.)
- [12] GAUTAM P, SAHU D R, DIXIT A, et al. Forward-backward-half forward dynamical systems for monotone inclusion problems with application to v-GNE. *Journal of Optimization Theory and Applications*, 2021, 190(2): 491 – 523.
- [13] BELGIOIOSO G, GRAMMATICO S. Semi-decentralized generalized Nash equilibrium seeking in monotone aggregative games, 2021, arXiv: 2003.04031.
- [14] KANNAN A, SHANBHAG U V. Distributed computation of equilibria in monotone Nash games via iterative regularization techniques. *SIAM Journal on Optimization*, 2012, 22(4): 1177 – 1205.
- [15] BÖRGENS E, KANZOW C. A distributed regularized Jacobi-type ADMM-method for generalized Nash equilibrium problems in Hilbert spaces. *Numerical Functional Analysis and Optimization*, 2018, 39(12): 1316 – 1349.
- [16] FRANCI B, GRAMMATICO S. A distributed forward-backward algorithm for stochastic generalized Nash equilibrium seeking. *IEEE Transactions on Automatic Control*, 2020, DOI: 10.1109/TAC.2020.3047369.
- [17] RYU E K, VŨ B C. Finding the forward-Douglas-Rachford-forward method. *Journal of Optimization Theory and Applications*, 2020, 184(3): 858 – 876.
- [18] BAUSCHKE H H, COMBETTES P L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2017.
- [19] DAVIS D, YIN W. Convergence rate analysis of several splitting schemes. GLOWINSKI R, OSHER S J and YIN W. *Splitting Methods in Communication, Imaging, Science, and Engineering*. Cham: Springer, 2016.
- [20] PARIKH N, BOYD S. Proximal algorithms. *Foundations and Trends in Optimization*, 2014, 1(3): 127 – 239.

作者简介:

潘晓伟 博士研究生, 目前研究方向为分布式优化, E-mail: panxw@mail.nankai.edu.cn;

刘忠信 教授, 博士生导师, 目前研究方向为多智能体系统控制与优化、复杂网络、计算机控制与管理, E-mail: lzhx@nankai.edu.cn;

陈增强 教授, 博士生导师, 目前研究方向为智能预测控制、离散事件系统、混沌系统及复杂网络, E-mail: chenzzq@nankai.edu.cn.