arXiv:2603.04231v1 [math.NA] 4 Mar 2026

线性子空间基于图的分裂算法的比较数值研究

Francisco J. Aragón-Artacho* Rubén Campoy* Irene López-Larios
César López-Pastor*

摘要

本文对六种基于图的分裂方法的算法进行了性能测试 [*SIAM*最优化杂志, 34 (2024), pp. 1569–1594] ，这些算法专门针对线性子空间的法锥。为此，我们首先进行了若干数值实验，以确定每种算法的最佳松弛参数。然后，使用此前确定的最佳松弛参数，比较各算法达到给定停止准则所需的迭代次数。数值结果使我们能够识别一些相关模式，并提供数值证据，为进一步的理论分析提供参考。

1 引言

设 $U_i \subseteq \mathbb{R}^p$ 为 $i = 1, \dots, n$的线性子空间。我们关注以下可行性问题的求解:

$$\text{Find } x \in \bigcap_{i=1}^{n} U_i. \tag{1}$$

对于 $n = 2$的情形，这个问题可以使用著名的*Douglas–Rachford*（*DR*）算法 [10]进行数值求解。给定初始点 $v^0 \in \mathbb{R}^p$ 和一个松弛参数 $\theta \in]0,\ 2[$, 算法按照以下递推关系迭代

$$\begin{cases} x_1^{k+1} = P_{U_1}\big(v^k\big), \\ x_2^{k+1} = P_{U_2}\big(2x_1^{k+1} - v^k\big), \\ v^{k+1} = v^k + \theta\big(x_2^{k+1} - x_1^{k+1}\big), \end{cases}$$

其中， P_U 表示到子空间 $U \subseteq \mathbb{R}^p$上的投影。生成的序列满足$v^k \to v^*$，其中某个 $v^* \in \mathbb{R}^p$，且 $x_i^k \to x^* = P_{U_1}(v^*) = P_{U_2}(2x^* - v^*) \in U_1 \cap U_2$，对于 $i = 1,\ 2$。此外，文献 [6]证明，极限点可以表示为

$$v^* = P_{U_1 \cap U_2}(v^0) + P_{U_1^\perp \cap U_2^\perp}(v^0). \tag{2}$$

因此可得 $x^* = P_{U_1 \cap U_2}(v^0)$，由此可知，Douglas–Rachford 算法不仅能求解原始可行性问题，还能求解一个最佳逼近问题。

当 DR 应用于子空间时，其线性收敛性已在 [5, 6] 中得到分析。作者证明，收敛速率同时取决于松弛参数 θ 以及 U_1 与 U_2之间的*Friedrichs* 角的余弦，其表达式为

$$c_F(U_1, U_2) := \max_{\|u_1\|, \|u_2\| \le 1} \left\{ \langle u_1, u_2 \rangle \;\middle|\; u_1 \in U_1 \cap (U_1 \cap U_2)^\perp,\ u_2 \in U_2 \cap (U_1 \cap U_2)^\perp \right\}.$$

*系 of 数学系 大学 of 阿利坎特， 西班牙。 电子邮件: francisco.aragon@ua.es, ruben.campoy@ua.es, cesar.lopez@ua.es

特别地，研究证明松弛参数的最优选择为 $\theta = 1$，此时可达到最小收敛速率，即 $c_F(U_1, U_2)$。

该算法自然推广到两个以上子空间时，通常无法保证收敛（例如参见 [1, § 3。3]）。克服这一困难的一种标准方法最初由 Pierra [12], 提出，即在适当的乘积空间中重新表述可行性问题。更确切地说，问题 (1) 可以等价地写成

$$\text{Find } \mathbf{x} \in \mathcal{U} \cap \Delta_n,$$

其中

$$\begin{aligned} \mathcal{U} &:= U_1 \times \cdots \times U_n \subseteq (\mathbb{R}^p)^n, \\ \Delta_n &:= \{(x, \ldots, x) \in (\mathbb{R}^p)^n \mid x \in \mathbb{R}^p\}. \end{aligned} \tag{3}$$

这种重构使得可以再次应用 DR算法，但代价是需要在更高维的空间中进行运算。

近年来，若干种无需依赖乘积空间重构、将 Douglas–Rachford算法推广到任意数量集合的方法被分别提出（例如参见 [8, 9, 11, 13]）。在开创性工作 [7], 中，这些方法通过引入 基于图的分裂方法这一类，在统一的理论框架下得到了整合。该方法针对线性子空间的情形在 第 2 节 中进行了说明，为设计分裂算法提供了一种灵活且通用的方法论。

本文旨在对若干基于图的 DR 方法实例在应用于线性子空间上的可行性问题时的性能进行数值研究，相关方法详见 第 3 节。具体而言，我们在 第 4 节 研究松弛参数的影响，并在 第 5 节比较不同图诱导算法的性能。最后，在第 6 节讨论一些结论和未解决的问题。

2 投影算法的图框架

文献 [7] 中介绍的框架基于对两个有向图 G 和 G'的选择，其中$G = (\{1, \ldots, n\}, \mathcal{E})$ 是一个保持自然顺序的连通图（即，$(i, j) \in \mathcal{E}$蕴含 $i < j$），而 $G' = (\{1, \ldots, n\}, \mathcal{E}') \subseteq G$ 是一个包含所有节点的连通子图。

给定图 G和节点 $i \in \{1, \ldots, n\}$，并用 $|\cdot|$表示集合的基数，我们定义

$$d_i^{\text{in}} := |\{h \mid (h, i) \in \mathcal{E}\}| \quad \text{and} \quad d_i^{\text{out}} := |\{h \mid (i, h) \in \mathcal{E}\}|,$$

分别表示节点 i 的入度和出度，并设

$$d_i := d_i^{\text{in}} + d_i^{\text{out}}$$

表示节点 i 的总度。另一方面，对于图 G'，我们构造其拉普拉斯矩阵 $L \in \mathbb{R}^{n \times n}$，定义为

$$L_{i,j} = \begin{cases} d'_i & \text{if } i = j, \\ -1, & \text{if } (i, j) \in \mathcal{E}' \text{ or } (j, i) \in \mathcal{E}', \\ 0, & \text{otherwise}, \end{cases}$$

其中，d'_i 表示子图 G'中节点 i 的总度数。拉普拉斯矩阵可分解为$L = ZZ^*$，其中 Z是一个满秩的 $n \times$（$n-1$）矩阵。此类分解总能通过奇异值分解得到（参见 [3, 命题 2.16]）。

采用此记号，给定初始点 v_1^0、……、 $v_{n-1}^0 \in \mathbb{R}^p$ 以及松弛参数 $\theta \in]0,\ 2[$,文献 [7] 中介绍的基于图的 DR 算法按以下方式迭代：

$$\begin{cases} x_i^{k+1} = P_{U_i}\left(\dfrac{2}{d_i}\displaystyle\sum_{(h,i)\in\mathcal{E}} x_h^{k+1} + \dfrac{1}{d_i}\sum_{j=1}^{n-1} Z_{i,j} v_j^k\right),\ \forall i = 1,\dots,n \\ v_j^{k+1} = v_j^k - \theta\displaystyle\sum_{i=1}^{n} Z_{i,j} x_i^{k+1},\ \forall j = 1,\dots,n-1. \end{cases}$$

若想进一步了解图 G 和 G' 在该算法中的作用，感兴趣的读者可参阅 [3, 备注 3.8]。

与经典 DR 算法类似，所生成的序列满足： $v_j^k \to v_j^*$ 对所有 $j = 1,\ \dots,\ n-1$ 以及 $x_i^k \to x^*$，均有

$$x^* = \frac{2d_i^{\text{in}}}{d_i} P_{U_i}(x^*) + \frac{1}{d_i}\sum_{j=1}^{n-1} P_{U_i}\left(Z_{i,j} v_j^*\right) \in \bigcap_{i=1}^{n} U_i, \qquad \text{for } i = 1,\dots,n,$$

参见 [7, 定理 3.2]。尽管该方程没有明确说明极限点 x^* 和 $\mathbf{v}^*$的信息，但近期在 [2, 定理 4.8]中得到了一个将 (2) 推广的闭式表达式。更确切地说，令 $\boldsymbol{\delta} \in \mathbb{R}^n$ 表示图 G的度平衡向量，其定义为 $\delta_i := d_i^{\text{in}} - d_i^{\text{out}}$，其中 $i = 1,\ \dots,\ n$；并令 $\boldsymbol{\alpha} = (\alpha_1,\ \dots,\ \alpha_{n-1}) \in \mathbb{R}^{n-1}$ 为方程组 $Z\boldsymbol{\alpha} = \boldsymbol{\delta}$的唯一解。再考虑由下式给出的子空间 $E \subseteq (\mathbb{R}^p)^{n-1}$

$$E := \left\{ (e_1,\dots,e_{n-1}) \in (\mathbb{R}^p)^{n-1} \;\middle|\; \textstyle\sum_{j=1}^{n-1} Z_{i,j}\, e_j \in U_i^{\perp},\ \forall i = 1,\dots,n \right\}.$$

那么，极限点 $x^* \in \mathbb{R}^p$ 和$\mathbf{v}^* = (v_1^*,\ \dots\dots,\ v_{n-1}^*) \in (\mathbb{R}^p)^{n-1}$ 可表示为显式形式

$$x^* = P_{\cap_{i=1}^n U_i}\left(\frac{1}{\|\boldsymbol{\alpha}\|^2}\sum_{j=1}^{n-1} \alpha_j v_j^0\right), \tag{4a}$$

$$\mathbf{v}^* = (\alpha_1 x^*,\dots,\alpha_{n-1} x^*) + P_E(\mathbf{v}^0), \tag{4b}$$

其中，$\mathbf{v}^0 := (v_1^0,\ \dots\dots,\ v_{n-1}^0)$。

3 所选算法与实验设置

在我们的计算研究中，将考虑以下六种基于图的 DR 方法系列中的算法：

- 顺序（见 [7]）。
- 完全 （针对 $n = 3$由 [7] 提出，针对任意 n由 [3] 推广，并允许使用前向项）。
- 向下并行（参见 [7]）。
- 向上并行（研究见 [8, 9]）。
- *Malitsky–Tam* （参见 [11]）。
- 广义 *Ryu* （在 [13] 中针对 $n = 3$ 提出，并在 [7]中推广到任意 n ）。

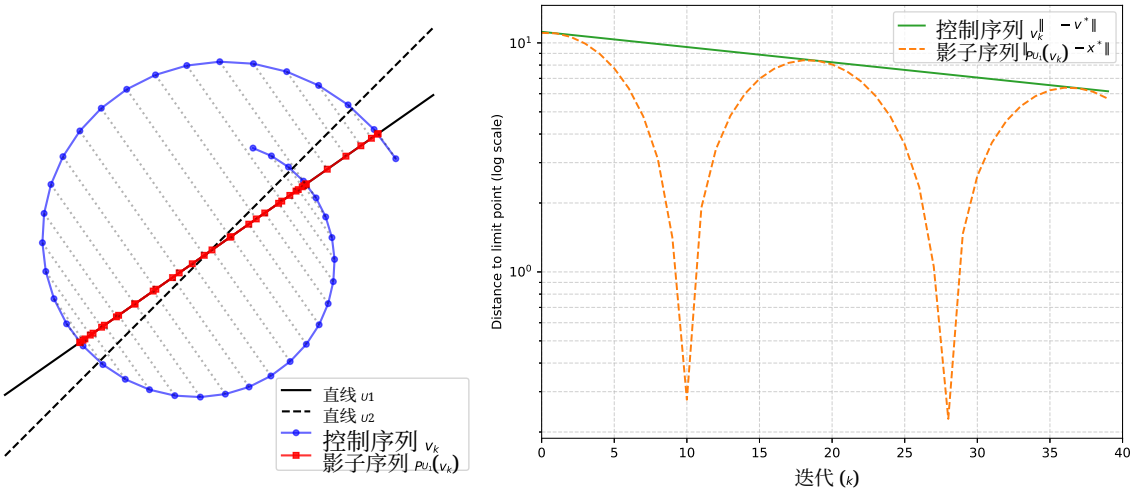
所有这些方法都可以视为第 2 节所述基于图的框架的特例，区别仅在于底层图对$(G,\ G')$。每种算法的具体图配置详见表 1。

算法	图 G	子图 G'
顺序	顺序 $\mathcal{E}:=\{(i,\ i+1)\ \mid i=1,\ \ldots,\ n-1\}$	顺序 $\mathcal{E}':=\{(i,\ i+1)\ \mid i=1,\ \ldots,\ n-1\}$
完全	完全 $\mathcal{E}:=\{(i,j)\mid 1\le i<j\le n\}$	完全 $\mathcal{E}':=\{(i,j)\mid 1\le i<j\le n\}$
向下并行	向下并行 $\mathcal{E}:=\{(i,\ n)\ \mid i=1,\ \ldots\ldots,\ n-1\}$	向下并行 $\mathcal{E}':=\{(i,n)\mid i=1,\ldots,n-1\}$
向上并行	向上并行 $\mathcal{E}:=\{(1,i)\mid i=2,\ldots,n\}$	向上并行 $\mathcal{E}':=\{(1,i)\mid i=2,\ldots,n\}$
Malitsky–Tam	Ring $\mathcal{E}:=\{(i,\ i+1)\ \mid i=1,\ \ldots\ldots,\ n-1\}\cup\{(1,\ n)\}$	顺序 $\mathcal{E}':=\{(i,i+1)\ \mid i=1,\ldots,n-1\}$
广义 Ryu	完全 $\mathcal{E}:=\{(i,j)\mid 1\le i<j\le n\}$	向下并行 $\mathcal{E}':=\{(i,\ n)\ \mid i=1,\ \ldots,\ n-1\}$

表 1：数值实验中考察的图生成算法。

在所有数值实验中，环境空间固定为 $\mathbb{R}^{50}$。所有子空间 $U_i \subseteq \mathbb{R}^{50}$和初始点均随机生成，并且对于每个问题实例，所有算法都使用相同的一组初始点进行初始化。

尽管影子序列会收敛到问题的解 x^*，但在数值实验中，我们监测的是控制序列的收敛情况。之所以这样选择，是因为即使序列尚未收敛，影子迭代点也可能恰好接近极限点。经典的 Douglas–Rachford 算法中已知这种现象：控制迭代点可能呈现绕极限点螺旋运动的特征，从而导致影子迭代点与解之间的距离呈非单调变化（见 图 1）。



(a) 迭代点轨迹　　(b) 到极限点的距离

Fig图 1：用于求解涉及 $\mathbb{R}$ 中两条直线的可行性问题的 Douglas–Rachford算法

由于可以通过 (4)显式计算极限点，我们在所有数值实验中采用了以下停止准则：

$$\|\mathbf{v}^k - \mathbf{v}^*\| < 10^{-6},$$

其中 $\mathbf{v}^k := (v_1^k, \ldots, v_{n-1}^k)$ 为控制序列。这一选择提供了可靠的收敛度量，并确保在不同算法和参数设置之间对收敛行为进行公平且一致的比较。

4 最佳松弛参数的数值研究

对于每个 $n \in \{3, 4, \ldots, 12\}$，我们生成了20个随机可行性问题，每个问题涉及 n个子空间。对于每个问题实例，所有算法均从相同的10个随机选取的初始点开始运行，并对结果取平均。这些算法在从网格 $\{0.1, 0.2, \ldots, 1.9\}$中选取的不同松弛参数 θ 下进行了测试。

令 $k_{i,\,\theta}$ 表示给定算法使用松弛参数 θ 求解第 i 个可行性问题时所需的平均迭代次数（在 10 个初始点中）。为评估参数 θ_0 的某一特定选择的性能，我们计算比值

$$\tau_{i,\theta_0} := \frac{k_{i,\theta_0}}{\min_{\theta \in \{0.1,\ldots,1.9\}} k_{i,\theta}},$$

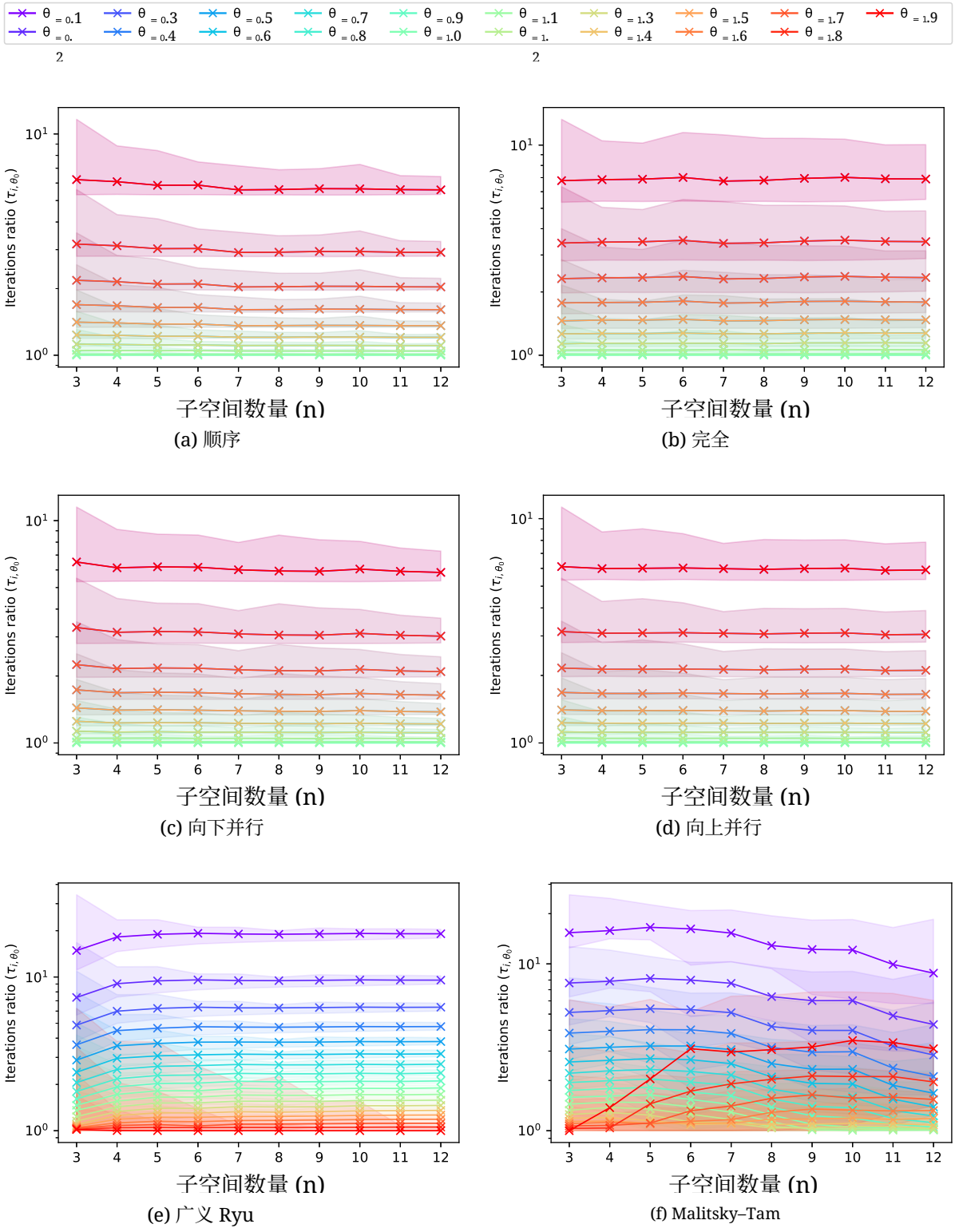
该比值将算法使用 θ_0 时所需的迭代次数与表现最佳的参数在该实例中所需的迭代次数进行比较。该测试的结果如图 2所示。对于每种算法，均绘制一幅图，比较子空间数量 n与 $\tau_{i,\,\theta_0}$ 的取值。不同颜色表示 θ_0 的不同选择。阴影区域表示全部 20 个实例所有可能比值的范围，而由线段连接的叉号表示这些比值的中位数。

结果讨论 前四种算法在 图 2中的表现非常相似，因此我们将它们一并分析。显然，最佳松弛参数为 $\theta = 1$ （以浅绿色标示）。还有另一个有趣的事实。乍看之下，可以观察到冷色似乎缺失，即 $\theta \in]0,\ 1[$取值所对应的结果。然而，这些结果实际上恰好出现在暖色的后面。不仅如此，实验还表明，较小与较大的 θ取值之间存在对称性：对于 θ 与对于 $2 - \theta$ 所需的迭代次数相同。在图中，这一点体现在例如以下选择0.1 与 1.9完全重叠，以及 0.2与 1.8，依此类推。这四种算法得到相同的结果，表明内部图结构解释了这些相似性。值得注意的是，这四种算法都具有性质 $G = G'$。

接下来，在考察广义 Ryu 算法中松弛参数的性能时，我们观察到了一些差异。事实上，此时最佳选择是 $\theta = 1.9$。另一方面，前面示例中观察到的对称性在这里不再适用。实际上，这里呈现出单调变化的特征： θ越大，性能越好。

最后，到目前为止，我们已经看到，松弛参数的最佳选择与子空间数量无关。但 Malitsky–Tam 并非如此：当 n较小时，较大的松弛参数是最佳选择。然而，随着子空间数量增加，性能最佳的参数值会减小，最终达到与 $\theta = 1$ 相同的值，此时共有四种情况满足 $G = G'$。

概述 在 图3中，针对所研究的六种算法，分别绘制了每个子空间数量对应的性能最佳参数。此前所述的三类情况得到了清晰呈现：满足 $G = G'$的四种算法，其最佳参数始终等于 $\theta = 1$；广义 Ryu 算法，其最佳参数始终为 $\theta = 1.9$；以及 Malitsky–Tam，其最佳参数取决于n松弛参数。更准确地说，我们可以观察到，当 n 较小时，Malitsky–Tam 的最佳参数接近于（甚至等同于）广义 Ryu 算法的最佳参数；而随着 n增大，它最终趋近于其他算法的最佳参数。这一行为的理论解释也留待未来研究。



(a) 顺序 (b) 完全

(c) 向下并行 (d) 向上并行

(e) 广义 Ryu (f) Malitsky–Tam

图 2：比较各算法在不同松弛参数值下性能的数值实验结果 θ

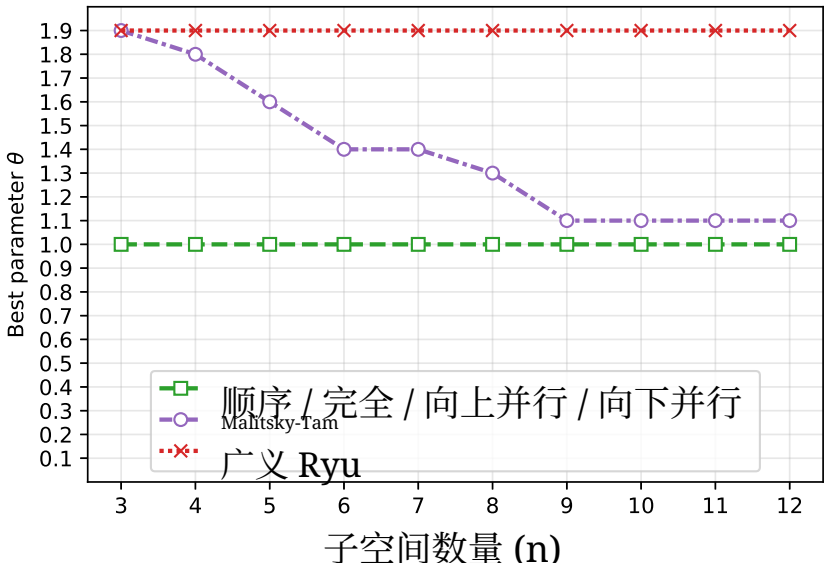


图3：关于子空间数量的最佳松弛参数 θ

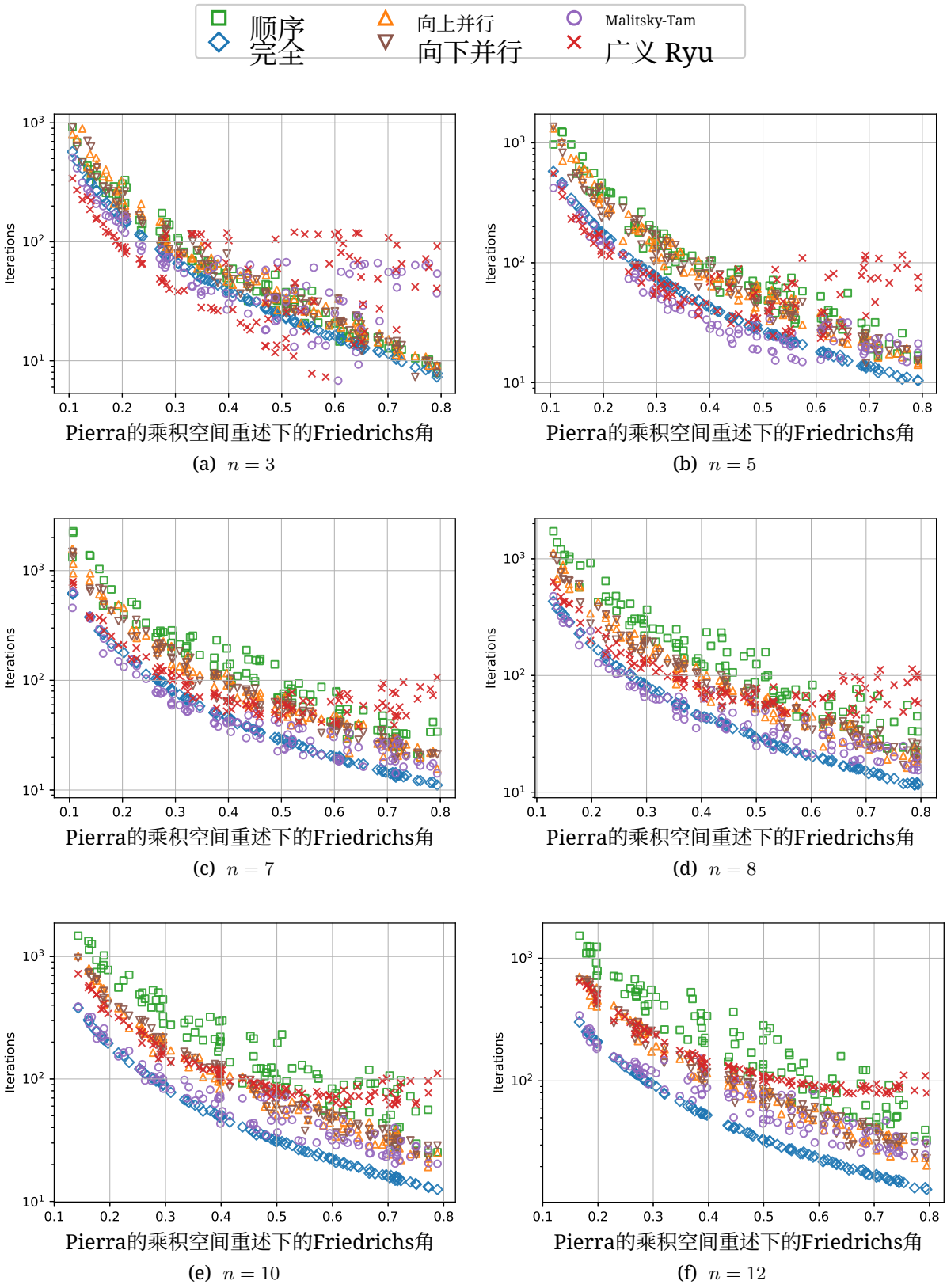
5 算法性能

根据上一节得到的实证结果，现在我们可以公平比较表 1中详述的六种算法的性能，并依据图3为每种算法设置性能最佳的参数。为此，我们进行了类似的实验。对于每个 $n \in \{3, 4, \ldots, 12\}$（表示子空间数量），我们生成了 100 个随机可行性问题。与上一次实验一样，我们随机生成了 10 个初始点，并在这些初始点上测试六种算法，最后对结果取平均值。

已知 Douglas–Rachford算法的收敛速率取决于 Friedrichs角 [5, 6], ，因此也希望量化多于两个子空间时这一几何量的大小。Friedrichs角向多于两个子空间的推广称为*Friedrichs* 数，其定义见 [4, 定义 3。2]。根据 Pierra 的乘积空间重构，它可以通过相关子空间之间的Friedrichs角明确确定，见(3)，以及 [4, 命题 3.6(f)]。为简便起见，在数值结果中，我们展示 Pierra 重构下的 Friedrichs角。尽管没有理论保证这一量能够反映原始子空间的几何结构，但数值结果表明，它确实提供了有意义的度量。

对于选取的 $n \in \{3, 5, 7, 8, 10, 12\}$值，本次测试的结果如图4所示。每个图展示了Friedrichs角（横轴）与平均迭代次数（纵轴）之间的关系，每个标记表示某种算法应用于特定问题实例时对应的数值。因此，每个图中每种算法都有100个形状和颜色相同的点。在实验中，我们根据主角计算 Friedrichs角，更多信息请参见 [5]。

结果讨论从图4可以看出，迭代次数与 Friedrichs角密切相关。尤其是对于完全算法，这种关系非常明显。对于每个子空间数量 n，迭代次数似乎都遵循一条直接取决于 Friedrichs角的曲线。其他算法似乎也遵循与完全算法相似的曲线模式，不过迭代次数的分布更分散，二者之间的关系也不那么明显。例如，与完全算法不同，随着 n 增大，顺序算法受到这种效应的影响，因此与 Friedrichs角之间的关系变得不那么明显。



(a) $n = 3$ (b) $n = 5$ (c) $n = 7$ (d) $n = 8$ (e) $n = 10$ (f) $n = 12$

图4：在 Pierra 的乘积空间重构下，比较各算法在不同子空间 Friedrichs角条件下性能的数值实验结果

Malitsky–Tam 算法的性能与完全算法不相上下。对于较小的 n 值和较小的角度，前者略快一些；但在角度较大时，前者则较慢，曲线的变化范围也更大，且对曲线的适应性较差。另一方面，当 n值较大时，完全算法在几乎所有问题实例中都是性能最佳的方法。

图中还显示，向上并行和向下并行几乎无法区分，在每个问题上使用的迭代次数大致相同。这可能是由这两种算法的图配置所致。事实上，尽管它们由不同的算法图定义，向上并行图和向下并行图具有相同的图拓扑：不考虑方向时，它们在拓扑上同构。

最后，可以观察到广义 Ryu 算法的一种有趣表现。与其他算法不同，角度较大时，其典型曲线会逐渐消失。对于 $n = 3$，这一现象尤为明显：当角度大于 0.5 时，迭代次数从不到 10 次到超过 100 次不等。另一方面，随着 n 增大，曲线会变得更加明显。然而，与其他曲线相比，它更加平坦，这意味着该算法的表现不如其他算法。

概述 我们在图 5中总结了前一项实验的信息。对于每种算法和每个 n，图中都显示了所有迭代次数的平均值。这张图清楚地展现了我们先前分析中最重要的几个方面。首先，它表明顺序算法速度最慢，且其性能随着 n 的增加而下降。接下来是向上并行和向下并行算法，两者的性能几乎相同。然后是广义 Ryu 算法，其性能随着 n 的增加而变差。最后，这张图还展示了完全算法和 Malitsky–Tam 算法的性能几乎相同。当 n较小时，Malitsky–Tam 所需的迭代次数较少；而当 $n \geq 8$时，完全算法所需的迭代次数保持不变，而 Malitsky–Tam 则逐渐需要更多迭代次数才能达到停止准则。

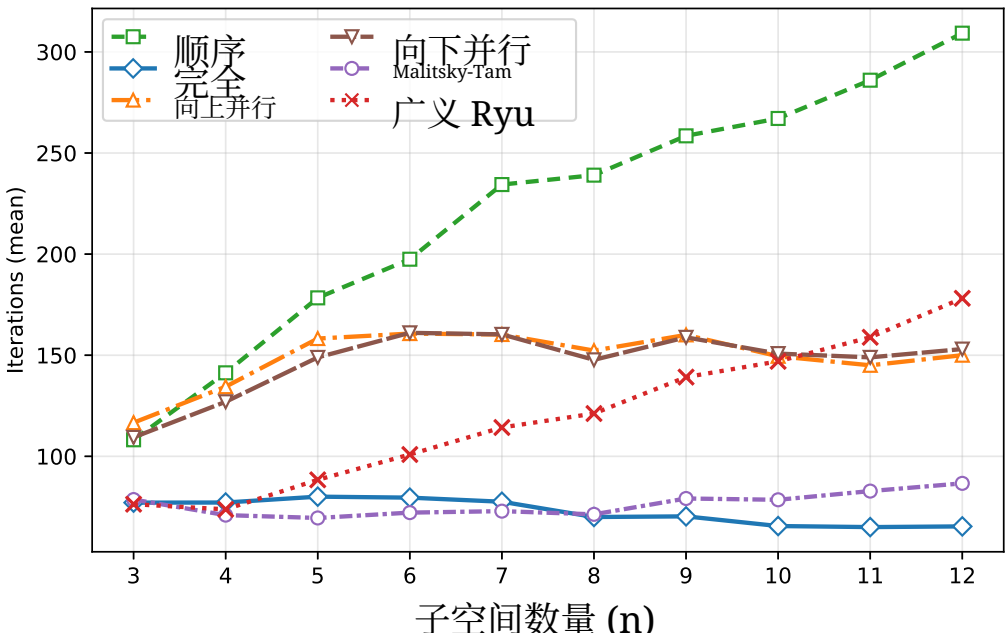


图 5：比较所有算法在不同子空间数量下性能的数值实验结果

6 结论与开放问题

我们研究了近期提出的、应用于线性子空间的基于图的分裂方法 [7], 这一系列方法中六种特征算法的数值性能。我们通过数值计算得出了每种算法的最佳松弛参数，并比较了子空间数量变化时各算法的表现。由此引出了以下问题：

- 当图 G 等于子图 G'时， $\theta = 1$ 是最优松弛参数吗？为什么此时 θ 和 $2 - \theta$ 表现相同？
- 当 $G \neq G'$ 时，是否存在最优松弛参数的一般表达式？
- 是否可以根据乘积空间重构中子空间的 Friedrichs角（或者等价地，根据 Friedrichs 数）得到收敛速率的表达式？

我们认为，这些问题值得从理论上加以解释，并计划在未来的工作中对其进行研究。

致谢 感谢一位匿名审稿人提出的宝贵意见。FJAA、RC 和 CLP 获得了由 ERDF/EU 资助的 PID2022-136399NB-C21 项目的部分支持。CLP 获得了由 MICIU/AEI/10.13039/501100011033 资助的 PREP2022-000118 项目以及“ESF 投资你的未来”项目的支持。

参考文献

[1] F. J. Aragón Artacho、R. Campoy 与 M. K. Tam：面向凸与非凸可行性问题的 Douglas–Rachford 算法。《数学方法与运筹研究》91（2020），第2期，201–240。[2] F. J. Aragón-Artacho、H. H. Bauschke、R. Campoy 与 C. López-Pastor：图分裂方法：线性子空间的不动点与强收敛。arXiv：2505.16564（2025）。[3] F. J. Aragón-Artacho、R. Campoy 与 C. López-Pastor：由图设计的前向-后向算法。SIAM最优化杂志 35（2025），第4期，2423–2451。[4] C. Badea、S. Grivaux 与 V. Müller：交替投影法的收敛速率。《圣彼得堡数学杂志》，23（2012），第3期，413–434。[5] H. H. Bauschke、J. Y. Bello Cruz、T. T. A. Nghia、H. M. Phan 与 X. Wang：松弛交替投影与广义 Douglas-Rachford 方法针对两个子空间的最优线性收敛速率。《数值算法》73（2016），第1期，33–76。[6] H. H. Bauschke、J. Y. Bello Cruz、T. T. A. Nghia、H. M. Phan 与 X. Wang：Douglas–Rachford 算法在子空间上的线性收敛速率为 Friedrichs 角的余弦。《逼近理论杂志》185（2014），63–79。[7] K. Bredies、E. Chenchene 与 E. Naldi：Douglas–Rachford 方法的图与分布式扩展。SIAM最优化杂志 34（2024），第2期，1569–1594。[8] R. Campoy：面向分裂算法的降维乘积空间重构。《计算优化与应用》83（2022），第1期，319–348。[9] L. Condat、D. Kitahara、A. Contreras 与 A. Hirabayashi：凸优化的近端分裂算法：近期进展巡礼，兼论新的变化。《SIAM 评论》65（2023），第2期，375–435。

[10] P.-L. Lions 与 B. Mercier：两个非线性算子之和的分裂算法。*SIAM* 数值分析期刊 **16**（1979），第6期，964–979。[11] Y. Malitsky 与 M. K. Tam：具有最小提升的单调算子和的预解算子分裂。 数学规划 **201** （2023），第1期，231–262。[12] G. Pierra：通过乘积空间中的形式化进行分解。数学规划 **28**（1984），第1期，96–115。[13] E. K. Ryu：DRS 作为双算子预解分裂的唯一性以及三算子预解分裂的不可能性。数学规划 **182** （2020），第1期，233–273。