arXiv:2304.02458v1 [cs.NE] 5 Apr 2023

用于分布估计算法的双随机矩阵模型

Valentino Santucci
佩鲁贾外国人大学 意大利佩鲁贾
valentino.santucci@unistrapg.it

Josu Ceberio
巴斯克地区大学 西班牙圣塞巴斯蒂安 josu.ceberio@ehu.eus

摘要

在组合优化中，以排列表示解的问题十分常见。因此，近几十年来，人们提出了许多进化算法来求解这类问题，其中基于概率模型的算法备受关注。相关研究大多致力于引入适用于解决排序或排名类问题的算法。然而，在针对指派问题提出基于概率的进化算法方面，现有研究尚未超越简单且多数情况下为单变量的模型。本文探讨将双随机矩阵（双随机矩阵）用于优化匹配和指派类排列问题。为此，我们研究若干学习与采样方法，以便将双随机矩阵有效纳入进化算法框架。具体而言，我们采用分布估计算法框架，并将双随机矩阵与针对排列问题的一些现有方法进行比较。对二次分配问题实例开展的初步实验验证了这一研究方向，并表明双随机矩阵有望取得极具竞争力的结果，但计算成本问题仍有待进一步研究。

1 引言

排列问题一直是组合优化领域的重要研究课题。尽管排列问题属于组合问题这一大类，但多年来的发展针对这类问题提出了专门的方法，这是因为它们具有一个独特之处：排列编码。正如 [1], 所指出的，排列可能是最丰富的组合结构之一。排列之所以受到广泛关注，主要是因为其用途广泛；它们出现在许多领域，例如图论、数学心理学和生物信息学，尤其是在路径规划 [2], 调度 [3] 或分配 [4]等物流问题中。

文献中介绍了大量不同的排列问题，但可以根据其所表示对象的性质对它们进行分类。Santucci 等人 [5], 将排列问题分为排序问题和匹配问题。第一类问题旨在确定给定一组项目的顺序或排名；第二类则旨在匹配两个大小相同的给定项目集。显然，由于排列在这两类问题中的含义不同，它们需要采用不同的算法方法。
为了解决排列问题，进化计算领域提出了许多方法，并特别致力于开发基于概率的策略（涵盖多种范式）。

举例来说，Ceberio 等人研究了用于描述排列上的概率分布的概率模型，例如基于 Kendall-τ [7] 距离和 Ulam [8] 距离的 Plackett-Luce [6] 和 Mallows 模型，并在分布估计算法（EDA）的框架下评估了它们的性能。Ayodele 等人 [9]提出了一种名为“随机键”的实数编码向量与排列之间的转换方法，并使用 EDA 处理排列流水车间调度问题。Santucci 等人 [10] 研究了在 Plackett-Luce 模型下采用基于模型的梯度搜索来优化线性排序问题。

深入的文献分析表明，在处理排列问题时，研究者一直专注于设计与排列编码 [11, 12]兼容的策略。然而，显而易见的是，不存在适用于所有排列问题的通用模型。事实上，我们发现，上述模型主要侧重于对排列中各项的顺序／排名进行建模，而能够捕捉问题分配特性的模型则较为少见。沿着这一研究方向，Irurozki 等人 [13] 提出了基于 Cayley 距离的 Mallows 模型，并指出该模型适合对第二类问题中的排列进行建模。文献 [14]和 [15]也对所谓的“交换”或“互换”操作是否适用于分配／匹配问题进行了类似探讨。

本文旨在延续这一趋势，提出适用于匹配问题中排列建模的概率模型，并以此开发基于概率的进化算法。为此，我们研究将双随机矩阵（DSM）用作排列空间上的概率模型。已知排列可以表示为一个 0/1 矩阵，其中每行和每列恰有一个元素为 1（其余元素均为 0）。因此，排列是双随机矩阵的一种特殊形式；双随机矩阵则将约束放宽为“每行和每列都由总和为 1 的正实数组成”。这些性质（将在第 3 节中阐明）使我们可以将双随机矩阵视为多个排列的综合，从而能够根据排列学习双随机矩阵，并从双随机矩阵中采样排列。双随机矩阵规定了某个元素出现在特定位置的概率。直观而言，它们似乎适合用于匹配／指派问题的建模，例如广为人知的二次分配问题（QAP） [4]。

基于以往关于离散状态模型的文献（这些文献大多来自进化计算领域之外），我们研究了学习和采样过程，并分析了其基本理论性质。随后，我们构建了若干以离散状态模型为模型的 EDA 方案。最后，我们在选定的一组二次分配问题基准实例上开展实验，并比较了所提出的不同策略之间的结果，以及这些策略与针对二次分配问题的相关 EDA 方案的结果。此外，我们还对收敛进行了实证分析。结果表明，离散状态模型在处理置换匹配问题方面具有良好的应用潜力。
本文其余部分安排如下。第 2 节介绍排列问题的背景、所用编码及其性质，并对问题类别加以区分。接着，第 3 节介绍双随机矩阵的理论方面，第 4 节则探讨学习和采样过程。第 5 节介绍所采用的 EDA 算法框架并讨论其特征。第 6 节总结初步实验。第 7 节对本文的贡献进行总结，并提出若干未来研究方向。

2 排列问题的表示与编码

形式上，排列是集合 $[n] = \{1, \ldots, n\}$ 到其自身的双射函数。通常使用希腊字母 σ, π 或 ρ 表示排列。 $\sigma(i)$（为便于阅读，也记作 σ_i）表示位置 i上的元素，而 $\sigma^{-1}(i)$表示元素 i的位置。所有大小为 n 的排列组成的集合记作 $\mathbb{S}_n$，在复合运算下构成一个群，在代数中称为对称群。

排列的经典描述是通常的向量表示，其中集合 $[n]$ 中的元素按特定顺序排列。然而，除了有序元素集合之外，排列还可以唯一地表示为若干互不相交的循环、对换、两两优先关系、0/1 矩阵，甚至图（即向量表示与以下表示之间存在双射）。排列有无数种编码方式，这或许正是它们被用于表示各种不同组合问题的解的原因。

Santucci 等人 [5]将现有问题分为两类（另见图 1）：

- 排序问题，目标是为给定的一组项目找到最优排序（例如排列流水车间调度问题），以及
- 匹配问题，要求以尽可能好的方式匹配两个给定的、规模相同的项目集合（例如二次分配问题）。

对于排序问题和匹配问题，排列是前 n 个整数的双射这一事实，只应被视为一种基因型编码。在排序问题中，给定一组需要根据所提供的目标函数进行最优排序的 n 个项目 A。因此， A 中项目的排序可以用两种不同的方式表示：将位置映射到项目（排序表示），或将项目映射到位置（秩表示）。需要注意的是，对于这两种表示（排序或秩），其语义解释（项目的排序）完全相同。选定一种表示后，下一步是寻找一种基因型编码。显然，位置是

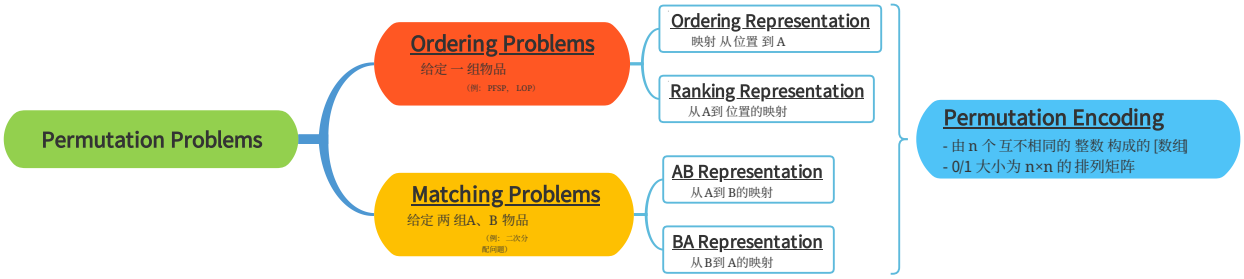

图 1：排列问题中语义解释、表示与编码之间的关系。

$[n]$ 中的整数，而 A 中的项目可以任意分配给 $[n]$ 中互不相同的标识编号。因此，位置和项目都编码为 $[n]$ 的元素，从而可以轻松地用 $\mathcal{S}_n$ 中的一个排列对它们之间的映射进行编码。换言之，排序表示和秩表示采用相同的基因型编码，但它们仍然是同一语义解释的不同表示。需要注意的是，在排序问题中，采用排序表示时，解的关键信息由序列中项目之间的先后关系提供（而采用秩表示时，则由秩大小的比较提供）。

匹配问题的基因型描述了不同的概念。具体而言，排列描述了一个双射函数，用于匹配两个大小相等的离散集合 A 和 B n中的元素。与排序问题类似，匹配问题也有两种表示：(1) *AB* 表示 和 (2) *BA* 表示。在第一种表示中，$[n]$ 中的每个位置对应于集合 A中的一个元素，而该位置上的标识编号表示与之匹配的 B 中的元素。顾名思义，BA 表示将集合 A和 B 分别从位置和标识编号的角色互换。

对于这两类问题，无论是排序还是匹配，都可以通过求逆操作轻松地将一种表示转换为另一种表示。

如本节引言所述，排列可以处理常见向量表示之外的不同描述。本工作相关的两种排列编码方案是：(1) 由 $[n]$ 到 n 的所有互不相同整数构成的数组（向量表示），以及 (2) $n\times n$ 置换矩阵，即每行和每列恰好有一个元素为 1 的 0/1 矩阵（0/1 矩阵表示）。这两种编码方案显然等价，可以通过简单的转换过程相互转换。下文分别用 $\mathbb{S}_n$ 和 $\mathbb{P}_n$ 表示线性编码的 n长度排列集合，以及 $n\times n$ 置换矩阵集合。因此，这些转换过程在 $\mathbb{S}_n$ 和 $\mathbb{P}_n$之间建立了一个同构。

3 双随机矩阵

双随机矩阵（DSM）是一个由非负实数构成的矩阵 $D = [d_{ij}]_{n\times n}$，其每一行和每一列的元素之和均为 1，即 $d_{ij} \geq 0$ 和 $\sum_{i=1}^{n} d_{ij} = \sum_{j=1}^{n} d_{ij} = 1$，对所有项 i，$j \in [n]$均成立。我们用 $\mathbb{D}_n$ 表示所有 n 阶双随机矩阵的集合。

显然，任何置换矩阵也是双随机矩阵，但反之不成立，因此 $\mathbb{P}_n \subset \mathbb{D}_n$。特别地，根据比尔霍夫–冯·诺依曼定理（BvN） [16]， $\mathbb{D}_n$ 定义了一个多面体，它是 $\mathbb{P}_n$ 的凸包。因此，任何双随机矩阵都可以表示为置换矩阵的凸组合。形式化地，给定 $D \in \mathbb{D}_n$，存在 k 个置换矩阵 $P_1, \ldots, P_k \in \mathbb{P}_n$ 以及 k 个权重 $w_1, \ldots, w_k \in (0, 1]$，其总和为 1（即 $\sum_{i=1}^{k} w_i = 1$），使得

$$D = w_1 P_1 + w_2 P_2 + \ldots + w_k P_k. \tag{1}$$

式 (1) 右侧称为 D 的分解，可通过所谓的 Birkhoff 算法 [17], 计算得出。众所周知，该算法返回的分解长度 k 至多为 $n^2 - 2n + 2$，即 $k = O(n^2)$。

根据定义，双随机矩阵的任意一行和任意一列都是定义在行/列索引集合 $[n]$ 上的多项分布[1]。这一性质使离散状态模型特别适合在置换匹配问题的背景下，为 AB（或 BA）表示构建模型。

[1]在 [18] 中也称为“多项分布”。

作为匹配问题的一个典型例子，我们来考虑二次分配问题（QAP）：它要求将一组 n 设施与一组 n 位置进行匹配（即 AB表示中的集合*A* 和*B* ）。设施和位置都可以用 $[n]$的元素编码。不过请注意，QAP 作为一个匹配问题，并未规定设施集合或位置集合中的任何顺序，因此 $[n]$ 中的元素虽然是基数，但应视为标识编号，不能依据它们之间的任何顺序关系来理解。在此背景下，我们可以设计一个双随机矩阵 $D \in \mathbb{D}_n$，使其满足：

- 的行和列分别表示 D 设施和位置；
- 的第 i行 D 是一个多项分布，表示将任意位置分配给设施 i的概率；
- 的第 j列 D 是一个多项分布，表示将任意设施分配到位置 j的概率。

以上几点清楚地表明，离散状态模型适合作为匹配问题的概率模型，能够一致地编码设施和位置的分布，而无需依赖它们之间的任何顺序关系。

4 离散状态模型的学习与采样

前文介绍的 Birkhoff-von Neumann定理明确指出，双随机矩阵可以看作多个排列（以置换矩阵的形式编码）的汇总，其中只考虑排列的匹配或指派性质。这一特性使双随机矩阵特别适合作为置换匹配问题的模型。因此，下面我们将分析如何从一组排列中学习双随机矩阵，以及反过来如何从双随机矩阵中采样排列。

4.1 学习离散状态模型

为了从一组排列中学习双随机矩阵，我们设计了两种不同的策略：精确学习 和平滑学习，后者是前者的推广。

精确学习。 给定 m 个排列 $P_1, \ldots, P_m \in \mathbb{P}_n$，则可以将双随机矩阵 $D \in \mathbb{D}_n$ 学习为它们的凸组合。形式上，

$$D = w_1 P_1 + w_2 P_2 + \ldots + w_m P_m, \tag{2}$$

其中，$w_1, \ldots, w_m$ 是 m 非负权重，且总和为 1；这些权重可以统一设为 $1/m$，或者根据相应排列的某种重要性度量按比例设定。
根据 Birkhoff-von Neumann定理（见第 3 节），D 也可以解释为 Birkhoff 多面体中 P_1、.....、 P_n 的（加权）质心。

根据凸组合的一个简单推论，如果所有置换矩阵P_1、.....、 P_m中的一般元素（i， j）均为零，那么 D 中的元素（i， j）也为零，反之亦然。正如下一小节将要说明的，这一性质使得无法从 D中采样出将设施 i 分配到位置 j的排列。

平滑学习。 为了解决精确学习 方案的“零概率”缺陷，我们对公式 (2) 稍作修改，引入均匀双随机矩阵 $U = [u_{ij}]_{n \times n}$ 作为凸组合的一个附加项，使得$u_{ij} = 1/n$ 对每个 i， $j \in [n]$。形式上，现在可以按如下方式学习双随机矩阵 D 。

$$D = w_1 P_1 + w_2 P_2 + \ldots + w_m P_m + \alpha U, \tag{3}$$

其中 $\alpha \in (0, 1]$ 是一个平滑因子，用于调节 U 在凸组合中的重要性，而此时 $\sum_{i=1}^{m} w_i = 1 - \alpha$。因此，在均匀设置下，权重设为 $(1 - \alpha)/m$。
公式 (3) 的学习方案生成的双随机矩阵中不含 0 项，因此可以对包含训练集中未出现的项目分配的排列进行采样。由于这一方面对于避免基于 DSM 模型的分布估计算法过早收敛至关重要，因此本文将采用平滑学习方案。

4.2 对离散状态模型进行采样

为了从给定的双随机矩阵中采样排列，我们设计了三种不同的策略：

- 概率采样，这允许定义一个适当的概率质量函数；

- 代数采样，它利用双随机矩阵与随机向量的乘法；
- 几何采样，其基于 Birkhoff-von Neumann定理。

以下将介绍这些策略。

概率采样（PS）。为简单起见，为了描述这一采样过程，我们使用线性置换编码。因此，给定 $D \in \mathbb{D}_n$，按照以下方式从 D 中采样得到一个排列 $\sigma \in \mathbb{S}_n$。

1. 从 D 中均匀随机地选择一行或一列，并分别用 $i \in [n]$ 和 $p \in [0,1]^n$ 表示其索引及对应的概率向量。2. 由于 p 是一个多项概率分布，根据 p采样一个元素 $j \in [n]$。3. 如果 p 是 D的一行，则设定 $\sigma_i := j$；否则，设定$\sigma_j := i$。4. 当 p 是行（或列）时，从 D 中删除行 i （或 j）以及列 j （或 i），然后对剩余的行和列重新归一化。5. 如果尚未删除 D 的所有行和列，则返回步骤（1）。

不难看出，该过程的每次迭代都会将 σ 中恰好一个元素置为 1，并从 D 中恰好移除一行和一列。这保证了经过 n 次迭代后，σ 是 $\mathbb{S}_n$ 的一个合法排列。此外，由于单次迭代的计算成本为 $\Theta(n)$，使用 PS 对一个排列进行采样需要 $\Theta(n^2)$ 个时间步。

对于给定的 $n\times n$ 双随机矩阵 $D = [d_{ij}]$，通过 PS 采样得到排列 $\sigma \in \mathbb{S}_n$ 的概率为

$$\Pr(\sigma|D) = \frac{\prod_{i=1}^n d_{i,\sigma_i}}{\sum_{\pi\in\mathbb{S}_n}\prod_{i=1}^n d_{i,\pi_i}} = \frac{\prod_{i=1}^n d_{i,\sigma_i}}{\mathrm{Perm}(D)}, \tag{4}$$

其中，分母是矩阵 D [19] 的永久式。

式（4）的概率质量函数可以导出 PS 的两个有趣性质。回顾一下，置换矩阵是双随机矩阵的特殊情形；当当前双随机矩阵是置换矩阵 P 时，只能采样得到 P。与此相反，当当前双随机矩阵是均匀双随机矩阵 U（见第 4.1 节）时，所有排列的概率都相同。

代数采样（AS）。这种采样策略基于 [20] 中提出的“随机舍入”方法（该方法又扩展了 [21] 和 [22] 中介绍的方法）。

其思路是：要从 $D \in \mathbb{D}_n$ 中采样一个置换矩阵 $P \in \mathbb{P}_n$，我们先均匀随机地生成一个向量 $v \in [0,1]^n$，然后通过求解方程得到 P

$$P \cdot \mathrm{rank}(v) = \mathrm{rank}(D \cdot v), \tag{5}$$

其中，$\cdot$ 表示通常的矩阵-向量乘法，而向量 rank(v) 定义为 rank(v)$_i = j$，其中 v_i 是 v 中第 j 小的值。例如，若 v =(0.1，0.5，0.8，0.2)，则 rank(v) =(1，3，4，2)。实际上，rank(v) 返回 argsort(v) 的逆排列，这就是众所周知的用于排列的随机键变换 [23]。

因此，通过使用 th e 线性置换编码，可从双随机矩阵 $D \in \mathbb{D}_n$ 中采样一个排列 $\sigma \in \mathbb{S}_n$，如下所示 s.

1. 均匀随机地生成一个向量 $v \in [0,1]^n$。2. 计算排列 π = rank($D \cdot v$) 和 ρ = argsort(v)。3. 返回排列 $\sigma = \pi \circ \rho$ （其中 $\circ$ 表示通常的置换复合）。

在 [20] 中已经证 明如上计算的线性排列 σ对应于置换矩阵 P *
其求解以下 优化问题：

$$P^* = \mathrm{argmin}_{P\in\mathbb{P}_n} ||D \cdot v - P \cdot v||_F^2, \tag{6}$$

其中 $||\cdot||_F$ 是矩阵的通常弗罗贝尼乌斯范数。

式(6)为AS提供了直观的理由。此外，它还表明，与PS情况一样，当 v 条目全部不同[2] 且当前双随机矩阵是置换矩阵时，则只能采样此排列。相反，当双随机矩阵是均匀的时，所有排列被采样的概率相同。

[2]Sin由于 v 个条目是独立且均匀随机抽取的，因此它们以概率 1 全都不同 .

最后，该过程的计算复杂度为 $\Theta(n^2)$，由步骤 (2) 中的矩阵-向量乘法决定。不过值得注意的是，矩阵-向量乘法可以采用非常快速的实现方式，因此 AS 在实际应用中效率很高。

几何采样（GS）。给定一个双随机矩阵 $D \in \mathbb{D}_n$，可以通过以下步骤对一个排列 $P \in \mathbb{P}_n$ 进行采样。为便于描述，该排列以置换矩阵的形式表示。

1. 执行 Birkhoff 算法 [17]，并获得长度为 $k = O(n^2)$ 的 D 分解。 2. 该分解具有公式 (1) 的形式，因此根据分解权重 $w_1, \ldots, w_k$ 对索引 $i \in [k]$ 进行采样。 3. 从计算得到的分解中选取并返回置换矩阵 P_i。

实质上，返回的排列是从 Birkhoff 多面体的顶点中采样得到的，其隐含地考虑了与 D 对应的点到各顶点的几何距离的倒数概率。

与 PS 和 AS 情形一样，当双随机矩阵是一个置换矩阵时，其 Birkhoff 分解仅由该矩阵自身构成，因此它是唯一可能被采样的排列。然而，当双随机矩阵是均匀双随机矩阵时，由于 Birkhoff 算法不是随机化的（或者至少目前还不是），并非所有排列都具有相同的概率。事实上，GS 一个显而易见的缺点是，采样范围受 Birkhoff-von Neumann定理限制，即最多只能采样 $n^2 - 2n + 2$ 个不同的排列。
此外，GS 的效率也很低。实际上，GS 的计算复杂度主要由 Birkhoff 算法决定。正如 [17], 所述，该算法是一种贪心迭代方法，每次迭代都会在由 $2n$ 个顶点构成的二分图中计算一个完美匹配。最常用于求解二分图匹配问题的算法是 Hopcroft–Karp 算法 [24], , 其计算复杂度为 $O(n^{2.5})$。还需注意，由于 Birkhoff-von Neumann定理，Birkhoff 算法需要进行 $O(n^2)$ 次迭代，因此 GS 的计算复杂度非常高，为 $O(n^{4.5})$。

5 所提出的 EDA

本节介绍一种用于置换匹配问题的分布估计（EDA），其概率模型被设计为双随机矩阵。

我们采用第 4.1 节介绍的平滑学习 方案，以及第 4.2 节介绍的三种采样策略。因此，我们基于所采用的采样方案，将 EDA 实现为三种不同的形式，分别称为 DSM-PS、DSM-AS 和 DSM-GS。

除采样策略外，所有实现均共享同一基础算法，其伪代码见算法 1。

算法 1所提出 EDA 的算法流程

输入： f: $\mathbb{S}_n \to \mathbb{R}$, 采样策略 $\in$ {PS, AS,GS}

1: $\lambda \leftarrow 10n$ ▷样本量
2: $\mu \leftarrow n$ ▷ 选择规模
3: $\alpha \leftarrow 1/n^2$ ▷平滑因子
4: $t \leftarrow 0$
6: $X_0 \leftarrow$ 个 λ 随机均匀抽取的排列
6: **当**评估预算未耗尽**时**
7: $t \leftarrow t + 1$
8: $Y \leftarrow$ 从 X_{t-1}中选择最佳的 μ 排列
9: $D \leftarrow$ 从 Y中学习得到的、平滑因子为 α的双随机矩阵
10: $Z \leftarrow$ 使用采样策略从 D 中采样得到的一组 λ 排列
11: 对所有 $\sigma \in Z$评估 f（σ）
12: $X_t \leftarrow Y \cup Z$
13: **结束 while 循环**
14: **返回** X_t中的最佳排列

算法 1 接收待优化的目标函数，以及从 PS、AS 或 GS 中选择的所需采样策略。

在第1–3行中，根据问题规模 n 并根据估计分布算法领域的先前经验 [7, 8]设置了三个算法参数。即，通过设置采样规模 λ来考虑高选择压力

以及将选择规模 μ 分别设置为 $10n$ 和 n。平滑因子 α 设置为 $1/n^2$，以便能够逃离停滞状态，但不会使搜索完全随机。
然后，在第 5 行用 λ 个随机排列初始化一组 X 解，并在第 6–13 行的主循环中迭代更新。在循环的每次迭代中，使用 X 中最优的 μ 个解（第 8 行）来学习 DSM 模型 D（第 9 行），并从中采样得到 λ 个新解（第 10 行）。对采样得到的解进行评估（第 11 行），然后将先前选出的解与新生成的解合并，以更新 X（第 12 行）。

请注意，在学习步骤中，除 α 外，训练集中的所有解均以统一的方式设置权重。
此外，迭代更新方案确保算法具有精英行为。事实上，集合 X 始终包含迄今访问过的最优解，并在第 14 行返回该解。此外，精英策略保证在任意迭代 t 中，DSM 模型 D 始终根据算法迄今访问过的最优 μ 个解进行学习。

精英行为和较高的选择压力通过平滑学习得以平衡；平滑学习使未见过的项目匹配也能以非零概率生成。

如前所述，学习和采样分别通过平滑学习方案和所选采样策略进行。因此，一次迭代的复杂度主要由采样和评估步骤决定。

6 实验研究

为分析所探讨的双随机矩阵的采样方法和学习方法的有效性与效率，下面将进行全面的实验研究。

6.1 实验设置

如引言所述，本研究面向匹配/分配类问题，因此在实验中，我们选择二次分配问题（QAP）作为基准问题。在 QAP 中，给定两个参数方阵$\mathbf{B} = [b_{ij}]_{n\times n}$和$\mathbf{H} = [h_{ij}]_{n\times n}$，目标是找到使目标函数最小化的排列 $\sigma \in \mathbb{S}_n$

$$f(\sigma) = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{i,j}h_{\sigma(i)\sigma(j)}.$$

已从 QAPLIB [25] 中选取了十二个规模各异的实例用于展示实验结果；在观察任何性能结果之前，就已确定实例的选择。具体而言，所选实例包括：*tai15{a/b}, tai20{a/b}, tai30{a/b}, tai50{a/b}, tai80{a/b}*和 *tai100{a/b}*。

为评估离散状态模型在优化中的适用性，我们考虑第 4.2 节所述的三种采样过程，即 PS、AS 和 GS。为了真实了解它们的性能，我们还将考虑文献中的另外三种竞争性 EDA 算法：

- 基于 Cayley 距离的 Mallows估计算法分布算法 [13]。这是本文所提算法的主要竞争算法。
- 基于 Kendall 距离的 Mallows估计算法分布算法 [7]。由于所采用度量的性质，该算法不适用于处理匹配问题。
- Plackett-Luce EDA [6] 已被公认为处理排序问题的算法，因此与之前的算法一样，在匹配问题中应不会非常有效。

用于表示它们的标签分别是EDA-MC、EDA-MK和EDA-PL。类似地，我们根据采样方法区分DSM算法，即DSM-PS和DSM-AS。
关于几何采样，初步运行表明DSM-GS方法在此框架中不可行。事实上，观察到即使平滑因子 α取值非常大，GS采样策略大多仍产生用于学习双随机矩阵的相同排列。这反过来使DSM-GS在迄今最佳解中产生非常罕见的改进。因此，同时考虑到GS的计算复杂度非常高（见第4.2节），我们决定在实验结果的呈现中省略它，将更准确的分析推迟到未来工作。

参照文献中 EDA 竞赛算法的指南，为所有算法确定了一套通用参数设置。因此，正如第 5 节所述，样本量和选择规模分别设为 $\lambda = 10n$ 和 $\mu = n$。

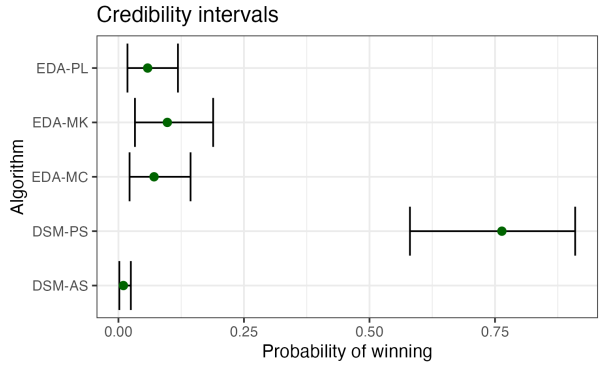

图2：各算法在基准实例集上的可信区间。该区间根据提供的实验数据，描述了每种算法成为最佳选择的概率。

6.2 有效性

为了总体评估 DSM类算法与上述估计分布算法相比的有效性，我们在选定的基准实例集上对每种算法进行了 20 次重复运行，每次运行允许进行 $100n^2$ 次评估。结果以相对于 QAPLIB 网站[3]所报告的已知最优结果的中位数相对偏差 (MRD) 汇总于表1。令人惊讶的是，结果显示有一种算法的表现远优于其他算法：DSM-PS。其他算法似乎都不如 DSM-PS，且彼此之间的表现相近。

表1：EDA和DSM算法在QAPLIB基准测试中的12个QAP实例（来自Taillard数据集）上的结果。报告了在20次重复实验中，相对于已知最佳结果所得值的中位数相对偏差（MRD）。粗体结果标出了MRD最低的算法。每种算法最多进行了 $100n^2$ 次评估。

实例	已知最佳值	EDA-PL	EDA-MK	EDA-MC	DSM-PS	DSM-AS
tai15a	388214	0.09283	0.09039	0.06565	**0.04145**	0.09659
tai15b	51765268	0.01379	0.00960	0.00510	**0.00455**	0.01520
tai20a	703482	0.13623	0.13740	0.13293	**0.05471**	0.13805
tai20b	122455319	0.11341	0.05344	0.04272	**0.01807**	0.11903
tai30a	1818146	0.12511	0.12666	0.13055	**0.04783**	0.13016
tai30b	637117113	0.27583	0.15849	0.31548	**0.07563**	0.31814
tai50a	4938796	0.14096	0.14025	0.13820	**0.04965**	0.14048
tai50b	458821517	0.35262	0.09779	0.36341	**0.03915**	0.37153
tai80a	13499184	0.12401	0.12492	0.12404	**0.03945**	0.12495
tai80b	818415043	0.34160	0.31773	0.34531	**0.03729**	0.34593
tai100a	21052466	0.11712	0.11668	0.11664	**0.03480**	0.11713
tai100b	1185996137	0.32555	0.12570	0.33081	**0.01785**	0.33082

为从统计学角度评估不同备选方案（参数组合）之间的差异，我们进行了贝叶斯性能分析 [26]。分析结果以可信区间的形式汇总于图 2 中。

图中 y 轴列出了不同算法，并为每种算法绘制了一个可信区间。该区间由一个绿色圆点（期望值）和一段数值范围构成，用于描述该算法在所比较的算法中成为最佳选择的概率。构建这些区间所需的数值，是通过对 [4] 计算得到的贝叶斯模型的后验分布进行采样获得的。

[3]https://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/ [4]分析的详细流程见 [26]。

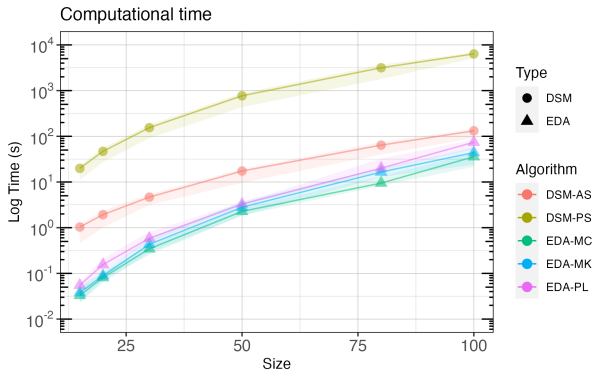

图 3：EDA-PL、EDA-MK、EDA-MC、DSM-AS 和 DSM-PS 算法每次运行 $100n^2$ 次评估的平均执行时间（对数秒）。

分析结果证实，DSM-PS 比其他算法领先一步；然而，区间的宽度，尤其是 EDA-PL 和 DSM-PS 的区间宽度，表明上述结论仍存在很大的不确定性。考虑到分析仅使用了 12 个实例，显然还需要更多数据才能得出更可靠的结论。不过，这些结果已表明，离散状态模型在优化方面可能非常有价值，至少在匹配问题的背景下是如此。

6.3 效率

我们已经看到，双随机矩阵模型的性能优于文献中的某些竞争算法，但它们的计算效率如何？本节旨在评估这些算法运行$100n^2$ 次评估所需的计算时间，以及该时间如何随问题规模变化。此处需要指出，估计分布算法是用 C++ 编写的，而双随机矩阵模型的各个版本则是用 Python 编写的，因此在分析[5]时需要考虑这一因素。

图 3 展示了各个对比算法消耗的计算时间的平均结果。由于需要执行的评估次数是问题规模的函数，因此我们针对实例集中的不同问题规模，为每种算法绘制了一条曲线。结果表明，估计分布算法（均使用 C++实现）的计算成本远低于所提出的基于双随机矩阵模型的算法（注意， y轴采用对数刻度）。事实上，上一节中表现最佳的算法 DSM-PS 是耗时最多的方法。双随机矩阵算法目前只是用 Python 实现的原型版本，因此仍有很大的改进空间，也确实有望接近估计分布算法的性能。

图 3 显示，各算法的可扩展性相近，因为折线图的斜率相当，这进一步佐证了最后这一结论。

6.4 采样收敛

为了分析不同算法在对问题的候选解进行建模并捕获其相关信息时的效果，本节分析算法每次迭代中所采样解的质量。我们不关注种群本身，因为其质量通常会单调提升；我们旨在考察采样得到的解。在合适的模型中，我们预期采样解的质量会随着迭代推进而提升。

为此，我们针对实例 *tai50a* 开展了另一项实验，并记录了各算法在执行过程中采样的每个解。随后，对每次迭代中样本的目标值取平均。为避免随机性，每种算法均进行了 5 次重复实验。结果以散点图和平滑曲线图的形式呈现在图 4 中。

一方面，我们可以观察到，估计分布算法几乎没有收敛，而且其（归一化）目标值始终停留在 $[0.19, 0.2]$ 范围内。至于离散状态模型，结果差异很大。DSM-PS 呈现出一种

[5] [27]对 27 种编程语言的能效进行了比较。分析表明，C 和 C++ 通常位居排名前列，而 Python 往往排名靠后，甚至可能垫底。

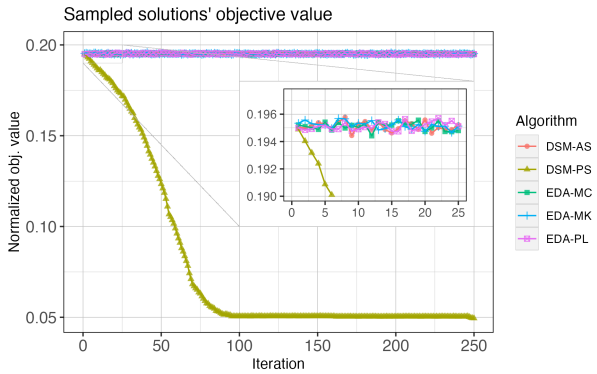

图 4：在运行 $100n^2$ 次评估期间，EDA-PL、EDA-MK、EDA-MC、DSM-AS 和 DSM-PS 算法每次迭代所采样解的目标函数平均值。每种算法均重复运行 5 次。此图使用的实例为*tai50a*。 x 轴的范围限制为 250 次迭代。

非常特殊的形态：几乎从最初几次迭代开始，算法就逐步改善结果。这一现象并不适用于所有离散状态模型，因为 DSM-AS 的表现似乎与其他估计分布算法相当。

7 结论与未来工作

本文研究了双随机矩阵（DSM）在进化算法框架中的应用，并重点关注用于求解基于排列的组合优化问题的分布估计算法（EDA）。

我们认为，现有的估计分布算法并不适合求解匹配或指派类排列问题，因为其中大多数算法都假设排列具有顺序性。在这种情况下，我们认为双随机矩阵可能表现良好。为此，我们通过探索不同的学习方法和采样方法，分析了双随机矩阵在估计分布算法中的潜在应用。具体而言，我们利用代数领域中一项著名结果——伯克霍夫–冯诺依曼定理，设计了一种简单的学习过程。对于从双随机矩阵中采样候选解（排列），我们发现了多种实现方式。本文研究了三种采样策略——概率式、代数式和几何式；它们各具特点，时间复杂度也各不相同。

针对一组二次分配问题实例开展的实验揭示了一个有趣的现象。与已发表的用于排列问题的估计分布算法相比，我们发现，基于双随机矩阵的估计算法，尤其是在采用概率采样时，取得了非常好的结果，甚至更胜一筹。

本研究旨在探索双随机矩阵在分布估计算法框架中的潜力，因此，其设计、集成和微调仍有很大的改进空间。一方面，这些算法所需的计算时间仍需进一步研究，以提升其竞争力。基于双随机矩阵的算法使用 Python 编写，而竞争算法则以 C++ 实现，这表明观察到的效率差距有望缩小。另一方面，可以通过深入分析参数设置（目前在算法中硬编码），并研究一种新的加权方案（目前采用均匀方案），进一步改进基于双随机矩阵的算法。此外，探索其他采样方法也很有意义，例如近期在机器学习领域采用的、基于 Sinkhorn-Knopp 算法的方法 [28]。

最后，本文仅专注于实现估计分布算法，因为这相对简单；但双随机矩阵还有其他用途，例如，在遗传算法中，可用于设计针对置换匹配问题的概率性交叉算子；或者在基于模型的梯度搜索算法中，可用于设计可微的类 Mallows 模型，将众数排列放宽为双随机矩阵。

致谢

Josu Ceberio 获得了巴斯克政府资助的研究团队 2022-2025（IT1504-22）和 Elkartek（KK- 2021/00065、KK-2022/00106）的部分支持，以及西班牙经济、工业和竞争力部资助的研究项目 PID2019-106453GA-I00 的部分支持。Valentino Santucci 获得了以下项目的部分支持：

“佩鲁贾外国人大学——大学研究项目资助计划（PRA 2022）”、“佩鲁贾外国人大学——面向教育、社会科学和人文科学的人工智能”，以及“佩鲁贾外国人大学——远程教育数字工具的设计与开发”。

参考文献

[1] Richard P. Stanley. 组合计数. Wadsworth 出版公司，美国加利福尼亚州贝尔蒙特，1986.[2] Paolo Toth 和 Daniele Vigo. 车辆路径问题. 工业与应用数学学会，2001.[3] Jatinder N. D. Gupta 和 Edward F. Stafford. 五十年来的流水车间调度研究. 欧洲运筹学杂志, 169(3):699–711, 2006.[4] Tjalling C. Koopmans 和 Martin J. Beckmann. 指派问题与经济活动的选址. Cowles 基金会讨论论文 4，耶鲁大学经济学研究中心，1955.[5] Valentino Santucci 和 Marco Baioletti. 一种用于黑盒排列问题低预算优化的快速随机局部搜索. 载于*2022*年*IEEE*进化计算大会（*CEC*），第1–8页，2022.[6] Josu Ceberio、Alexander Mendiburu 和 Jose A. Lozano. 基于排列的优化问题中的 Plackett-Luce 排名模型. 载于*2013*年*IEEE*进化计算大会（*CEC*），墨西哥坎昆，第494–501页。IEEE，2013.[7] Josu Ceberio、Ekhine Irurozki、Alexander Mendiburu 和 Jose A. Lozano. 用于流水车间调度问题的基于距离的排名模型分布估计算法. *IEEE*进化计算汇刊, 18(2):286–300，2014年4月.[8] Ekhine Irurozki、Josu Ceberio、Borja Calvo 和 Jose A. Lozano. Ulam 距离下的 Mallows 模型：一种可行的组合方法. 载于*2014*年*NIPS*会议研讨会论文集，加拿大魁北克省蒙特利尔，2014年12月.[9] Mayowa Ayodele、John McCall 和 Olivier Regnier-Coudert. RK-EDA：一种基于随机键的新型分布估计算法. 载于 Julia Handl、Emma Hart、Peter R. Lewis、Manuel López-Ibáñez、Gabriela Ochoa 和 Ben Paechter 编《自然问题求解——第十四届*PPSN*会议》，第849–858页，瑞士 Cham，2016年。Springer 国际出版公司.[10] Valentino. Santucci、Josu Ceberio 和 Marco Baioletti. 排列空间中的梯度搜索：在线性排序问题中的应用. 载于*2020*年遗传与进化计算大会伴随会议论文集，第1704–1711页，2020.[11] Marco Baioletti、Alfredo Milani 和 Valentino Santucci. 可变邻域代数微分进化：其在具有累积成本的线性排序问题中的应用. 信息科学, 507:37–52, 2020.[12] Arthur Guijt、Ngoc Hoang Luong、Peter AN Bosman 和 Mathijs de Weerdt. 链接学习、基因池最优混合与非冗余编码对排列优化的影响. 群体与进化计算, 70:101044, 2022.[13] Ekhine Irurozki、Borja Calvo 和 Jose A Lozano. Cayley 距离下 Mallows 模型和广义 Mallows 模型的采样与学习. 应用概率方法与计算, 20(1):1–35, 2018.[14] Marco Baioletti、Alfredo Milani、Valentino Santucci 和 Marco Tomassini. 排列空间局部最优网络中的搜索移动：二次分配问题案例. 载于遗传与进化计算大会伴随会议论文集，第1535–1542页，2019.[15] Marco Baioletti、Gabriele Di Bari、Alfredo Milani 和 Valentino Santucci. 排列问题代数交叉算子的实验比较. 计算机基础, 174(3-4):201–228, 2020.[16] Fanny Dufossé 和 Bora Uçar. 关于双随机矩阵的 Birkhoff–von Neumann 分解的说明.线性代数及其应用, 497:108–115, 2016.[17] Diane M Johnson、AL Dulmage 和 NS Mendelsohn. 关于 G. Birkhoff 的一个双随机矩阵算法. 加拿大数学通报, 3(3):237–242, 1960.[18] Ian Goodfellow、Yoshua Bengio 和 Aaron Courville. 深度学习. MIT 出版社，2016. http://www. deeplearningbook.org.[19] David G Glynn. 方阵的永年式. 欧洲组合学杂志, 31(7):1887–1891, 2010.[20] RJ Wolstenholme 和 Andrew T Walden. 一种用于在图匹配问题中投影到排列的采样策略. *arXiv* 预印本 *arXiv:1604.04235*, 2016.

[21] Fajwel Fogel、Rodolphe Jenatton、Francis Bach 和 Alexandre d'Aspremont. 排列问题的凸松弛. 神经信息处理系统进展, 26, 2013.[22] Alexander Barvinok. 用置换矩阵逼近正交矩阵. 纯粹与应用数学季刊, 2(4):943–961, 2006.[23] Valentino Santucci、Marco Baioletti 和 Alfredo Milani. 组合优化中群体算法与进化算法的代数框架. 群体与进化计算, 55:100673, 2020.[24] John E Hopcroft 和 Richard M Karp. 求解二分图最大匹配的 nˆ5/2 算法. *SIAM*计算杂志, 2(4):225–231, 1973.[25] Rainer E. Burkard、Stefan E. Karisch 和 Franz Rendl. QAPLIB——二次分配问题库.全球优化杂志, 10(4):391–403, 1997.[26] Jairo Rojas-Delgado、Josu Ceberio、Borja Calvo 和 Jose A. Lozano. 算法排名比较的贝叶斯性能分析. *IEEE*进化计算汇刊，第1–1页，2022.[27] Rui Pereira、Marco Couto、Francisco Ribeiro、Rui Rua、Jácome Cunha、João Paulo Fernandes 和 João Saraiva. 按能效对编程语言进行排名. 科学计算机编程, 205:102609, 2021.[28] Gonzalo Mena、David Belanger、Gonzalo Munoz 和 Jasper Snoek. Sinkhorn 网络：运用最优传输技术学习排列. 载于*NIPS*最优传输与机器学习研讨会，第3卷，2017。