一种用于任意凸形区域高效圆覆盖的改进的准物理动力学算法多边形

易泽平[1], 王永军[1*]， 王宝山[1],刘松毅[1]数学科学学院，北京航空航天大学，中国北京，100191 *通讯作者：wangyj@buaa.edu.cn；贡献作者：yzping@buaa.edu.cn；bwang@buaa.edu.cn；liusongyi@buaa.edu.cn

arXiv:2602.02570v1 [cs.CG] 31 Jan 2026

摘要

最优圆覆盖问题旨在寻找一组圆的构型，使其在给定区域内的覆盖面积最大化。尽管对于简单情形存在理论最优解，但该问题的NP难特性使其在包含大量圆的复杂多边形中计算上难以处理。现有方法主要局限于规则区域，而少数针对不规则多边形设计的算法存在初始化不佳、边界效应缺乏控制以及圆之间重叠过度等问题，导致覆盖效率较低。因此，我们针对任意凸多边形提出了一种改进的准物理动力学（IQPD）算法。我们的核心贡献包括以下三点：(1)提出一种保结构初始化策略，通过缩放和仿射变换将圆的六方密堆积映射到目标多边形中；(2)构建融合摩擦力和半径扩张优化迭代模型的虚拟力场；(3)设计一种基于法向梯度和切向梯度的边界环绕策略，以找回溢出圆。实验结果表明，在各种凸多边形上，针对七项度量，我们的算法显著优于四种最先进的方法。这项工作可为实际应用中的运营优化或资源分配提供更高效的解决方案。

关键词： 覆盖优化、改进的准物理动力学算法、六角密堆积、NP难、计算几何

1 引言

给定一组圆和一个多边形区域，如何排列这些圆，才能最大限度地覆盖给定区域，甚至实现完全覆盖？尽管这个问题看似简单，却在我们的生活中有着广泛的应用。在应急救援、切割制造等实际场景中， [1], 农业灌溉[2], 和设施选址 [3] , 等领域，确定圆的最优排列方式以实现更大的覆盖面积具有重要的实际价值。

这个问题源于一个著名的密堆积问题 [4, 5], ：在给定的多边形内放置已知半径的圆，要求圆之间不重叠，并使圆的数量最多。作为计算几何和运筹学优化领域的经典问题，该问题通常是 NP难 [6–8],的，而已知的可证明最优解仅适用于特定情形（某些简单形状、特定数量或特定半径）。因此，启发式算法成为获得近似最优排列 [9, 10]的主要方法。早在阿基米德时代，学者们就开始研究平面上圆的最优排列，但当时密堆积问题尚未发展成为一套正式的数学理论[11]。17世纪，约翰内斯·开普勒在其著作 $On\ the\ Six-Cornered\ Snowflake$中讨论了平面上全等圆的密堆积，并猜想球的六边形堆积会是三维空间[12], 中最密的构型，但他并未证明二维圆形铺砌的最优性。阿克塞尔·图在 1890 [13], 中首次尝试证明，六边形排列是平面上圆的最密铺砌方式，但即使经过 1910 [14]修订，该证明仍不完整。最终，直到1949年，L´aszl´o Fejes T´oth 才给出了严格证明 [15] ，表明六边形排列是平面上圆形堆积的最优解（密度为 $\frac{\pi}{2\sqrt{3}} \approx 0.9069$）。这一结论为研究不规则多边形的最优覆盖问题 [8, 10, 16]提供了宝贵经验。

关于圆的最优排列的研究主要分为两类典型问题：覆盖与装填问题[16, 17]。覆盖问题通常旨在用一组全等圆完全覆盖给定区域。其目标通常是使覆盖所需的圆的半径或数量最小化，或使完全覆盖的区域面积最大化；一般允许圆之间重叠。相比之下，堆积问题涉及将一组圆（半径不一定相等）放入给定容器中，且彼此不得重叠。其目标通常有两种：要么使固定容器内所装圆的数量或总面积最大化，要么使容纳所有圆所需容器的尺寸最小化（例如圆形容器的半径或矩形的面积）。

圆覆盖与装填问题的求解方法大致可分为两种范式[16]。第一种是解析方法，它运用数学理论推导出可证明为最优的排列[18, 19]。然而，由于计算复杂度较高，这种方法通常仅适用于圆的数量较少或构型高度规则的情况。因此，第二种范式适用范围更广，依赖启发式算法，可按多种策略分类：1）准物理模拟[20, 21],，将圆建模为相互排斥的物体，通过碰撞检测和弹性运动调整其位置，直至达到稳定、密集的状态。2）随机搜索技术，通过引导式随机化探索解空间，例如遗传算法 [10, 22] 和模拟退火算法[23, 24]。3）非线性规划方法，利用几何结构（例如沃罗诺伊图） [16] 将问题构建为优化模型，最终采用上述启发式算法求解。

然而，在农业灌溉和无线通信等实际圆形布局问题中，其复杂性往往超越了纯粹装填或覆盖的经典范式。更常见的情形带来了一个重大挑战：圆的数量和尺寸往往不足以实现无重叠装填，同时也不足以实现完全覆盖。这种情形定义了圆的最优覆盖问题。该问题的核心是在允许重叠的条件下，寻找一种近似最优布局，使目标区域的覆盖率最大化，从而在资源有限的情况下实现最高覆盖效率。

为准确表述这一问题，我们建立如下数学模型。给定一个多边形区域 P、圆半径 r、 n个全等圆以及圆心坐标 $(x_i,\ y_i)$ $(i = 1,$ ……， $n)$，覆盖面积定义为

$$A_{cover} = Area\left(P \cap \left(\bigcup_{i=1}^{n} C_i\right)\right),$$

其中， C_i 表示圆心为 $(x_i,\ y_i)$、半径为 r的圆。圆的最优覆盖问题的数学表述如下：

$$\underset{(x_1,y_1),\cdots,(x_n,y_n)}{maximize}\ A_{cover}/Area(P) \tag{1}$$

$$s.t.\quad \begin{cases} (x_i, y_i) \in P \quad \forall i, \quad dist\left((x_i, y_i), \partial P\right) \geq 0 \\ C_i \cap P \neq \oslash \end{cases}. \tag{2}$$

其中， $dist\left((x_i,\ y_i)\ \partial P\right)$表示圆心 $(x_i,\ y_i)$到多边形边界 ∂P的有符号距离。约束 dist $((x_i,\ y_i)\ \partial P)\ \geq 0$确保所有圆心均位于多边形 P内。第二个约束 $C_i \cap P \neq \emptyset$则保证每个圆都对区域覆盖有所贡献。

目标函数 $A_{cover}/Area(P)$ 是非凸且非线性的，存在大量局部最优。由所有圆心定义的搜索空间（x_i、 y_i）会随着 n 迅速扩展，进一步加剧问题的计算难度。幸运的是，最优圆覆盖问题兼具装箱问题和覆盖问题的特征，因此可以对这些领域中的许多现有方法和策略加以调整和应用。由于问题的复杂性，全局最优解通常只有在小型 n 或简单多边形 P中才能获得。对于更大 n 或更复杂的区域，则采用元启发式算法来逼近最优解。

许多元启发式算法[25–30]能够在正多边形中实现较高的覆盖率，但在处理

复杂多边形时，往往会遇到诸如圆之间重叠过多、圆的数量超出边界、覆盖率较低以及运行时间较长等问题。此外，由于非凸多边形中存在凹角，会产生若干复杂问题：在检查圆心的可行性时必须处理非凸约束；优化圆的布局时必须考虑是否应将圆嵌入凹陷区域；计算覆盖面积时则必须应对更复杂的边界和多个非连通区域。这些因素显著增加了问题的复杂度。为专注于研究核心覆盖机制并建立稳定有效的求解框架，本文将研究范围限定为任意复杂凸多边形中的圆最优覆盖。然而，目前仍存在明显空白：已在圆填充中证明有效的成熟准物理模拟框架，尚未系统地应用于或改进以解决最优覆盖问题。准物理模拟中的智能初始化策略和迭代改进等关键方面仍未得到探索。尽管当前研究十分倾向于将新型元启发式算法应用于圆的最优覆盖问题，但这种侧重可能忽视了成熟策略或方法在经典覆盖与填充问题中的潜在优势。通过充分改进和完善，其中许多方法在求解圆的最优覆盖问题时，可能比元启发式算法获得更好的结果。因此，我们旨在开发一种改进的准物理算法，并提出适用于一般凸多边形情形的新覆盖策略。

本文结构如下：第 2 节回顾与本文相关的工作。第 3 节介绍计算几何中的基本概念。第 4 节详细阐述改进的准物理动力学算法。第 6 节总结全文并提出未来的研究方向。

2 文献综述

2.1 解析方法

解析方法常用于解决覆盖问题。研究人员最初专注于单位圆的最优排列，以覆盖半径尽可能大的 n 圆形区域 [31]。随后，研究进一步确定了特定数量圆的最优排列：Bezdek 针对 $n=5,\ 6$ [7] 1979 年至 1984 年间，Fejes T´oth 针对 $n=8,\ 9,\ 10$ [18] 于 2005 年得出。

与此同时，研究人员还研究了使用单位圆覆盖矩形区域的问题。对于 $n \leq 5$ [32]，最优覆盖方案于 1997 年确定；对于 $n \leq 7$ [19]，于 2000 年确定；最终，对于 $n \leq 12$ [7]，于 2014 年确定。为处理更一般的形状，Gul 等人 [8] 于 2021 年研究了狭长凸区域，提出了基于单行覆盖和不规则六边形网格覆盖的高效策略，并给出了所需圆数量的理论界。最近，Fekete 等人[7] 于 2024 年为 $\lambda \times 1$ 矩形的临界覆盖面积提供了闭式解，给出了一个紧确理论基准。

研究也扩展到了三角形区域。1997年，Melissen为配备 $n \leq 18$ 单位圆[33]的等边三角形提供了最优覆盖方案及猜想。随着 n 增加，确定可证明的最优构型在计算上变得难以承受，因此 Nurmela 等人[34] 在2000年提出并完善了最多 $n=36$ 个圆的猜想。

尽管在覆盖规则区域方面已取得重大进展，但实际应用通常涉及不规则多边形区域。2023年，Choi等人[6] 开发了一种 $O(n\log\ n)$时间算法，可使用两个全等圆以最小半径覆盖任意凸多边形，超越了先前的方法 [35, 36]。更广泛地说，Birgin 等人[37] 研究了使用固定数量的全等圆以最小半径覆盖二维有界区域的问题。对于一类特定的非光滑区域，他们推导出了最优半径的界，并给出了当圆的数量趋于无穷时最优半径界的渐近展开。

Choi等人的研究在理论分析与算法设计之间架起桥梁，显著提升了计算效率。然而，随着问题复杂度增加，无论是覆盖圆的数量增多、维度升高，还是涉及非凸区域，从理论上证明最优性都会变得愈发困难。因此，研究人员转而采用启发式算法来解决这些更复杂的覆盖和堆积问题。

2.2 具有特殊规则的启发式算法

启发式算法主要用于解决 Kravitz 在1967[38]中最初提出的大规模等圆装填问题。该问题要求将 n 个半径为 r 的全等圆无重叠地装入最小的圆形容器中，其中，R_0表示最小圆形容器的半径，而（x_i，y_i）表示第 i个圆圆心的坐标。为简化模型，我们可以假设圆形容器的中心位于原点（0，0）。因此，其数学表达式如下：

$$minimize\ R_0, \tag{3}$$

$$s.t. \begin{cases} \sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \geq 2r, & \forall i \neq j \\ \sqrt{x_i^2+y_i^2} \leq R_0 - r, & \forall i \in \{1,...,n\} \end{cases}. \tag{4}$$

早期的启发式算法基于经验或直觉规则，旨在用可接受的时间找到令人满意的解，通常以无法保证最优性为代价。

2002年，Huang 等人[20] 提出了一种基于人类经验的启发式算法，即最小灵活性优先（Less Flexibility First，LFF）算法。该算法的核心思想是优先填充灵活性最低的角落，其次是灵活性适中的边缘区域，最后填充灵活性最高的中心区域。通过遵循优先处理低灵活性区域的原则，与此前提出的传统随机算法相比，该算法同时提高了装填密度和计算速度。2005年，Mladenovi´c 等人[39] 将紧密圆装填构建为一个非凸优化问题。他们提出了一种基于重构下降法的启发式算法，主要通过切换坐标系（例如笛卡尔坐标和极坐标）来破坏当前解的稳定性，从而提高解的质量。该算法解决了传统算法容易陷入局部最优的缺陷

，增强了全局搜索能力。2006年至2008年，Huang 等人[40, 41] 进一步将等圆装填扩展到异圆装填。他们提出了基于最大孔洞度的 A1.0 算法，随后通过引入自前瞻策略对其进行改进，形成了 A1.5 算法。之后，他们开发了一种将最大空腔度原则与 PERM 策略相结合的混合启发式算法，进一步提高了装填密度和计算效率。

2.3 准物理模拟

准物理模拟方法最早由 Huang 等人于 2011 年提出。该方法最初包含三个关键步骤：(1)准物理下降过程，利用弹性力逐步消除圆之间的重叠，从而收敛至局部最优排列；(2)准物理盆地跳跃过程，引入中心吸引力和非接触排斥力，帮助圆跳出局部最优，实现全局搜索；(3)半径调整过程，在固定容器内应用二分法精确调整圆的半径 R_0，确保找到最优的无重叠装填。在圆的数量 n 从 1 到 150 的测试实例中，该算法改进了 37 个案例的已知最佳装填方案，并在另外 113 个案例中重现了已知最佳结果。通过将准物理方法与全局优化相结合，该方法有效避免了局部最优并提高了解的质量，展现了准物理算法在优化领域的应用潜力。然而，该方法的局限在于依赖初始构型，而初始构型会显著影响收敛速度和结果。因此，Zhu[42]提出了准人类资历顺序（QS）算法，以优化圆填充的初始构型。其核心思想是最大者优先策略：先放置较大的圆，再用较小的圆填充间隙，从而最大限度地提高空间利用率。QS 算法与准物理方法相结合，并与采用随机初始构型的方法进行了基准测试。结果表明，QS 算法显著减少了迭代次数和计算时间，同时降低了过早收敛或陷入无限循环的风险。为解决这些问题，He 等人 [43] 于 2018 年提出了一种用于等圆装填的高效准物理—准人类（QPQH）混合算法。QPQH 将改进的 Broyden-Fletcher-Goldfarb-Shanno（BFGS）算法与准人类盆地跳跃策略相结合，利用每个圆的邻域信息进行加速并收缩容器，以跳出局部最优。该算法在多个基准测试中取得了最先进的结果，展现出处理大规模问题时的高效性和鲁棒性。

2.4 元启发式算法

近年来，已提出多种新颖的元启发式算法[44–47]，用于圆填充或最优圆覆盖问题。目前的主流方法包括改进的群体智能算法，以及结合经典几何结构（如Voronoi 图）的启发式算法。

早期研究倾向于整合多种技术以构建混合优化框架。Ryu等人[16] 于2020年提出了一种基于Voronoi 图的实时圆形填充算法。VOROPACK-D算法采用大圆优先策略，通过Voronoi 图对已放置圆之间的间隙进行建模，从而快速确定后续圆的最优位置。为减轻初始构型对最终结果的影响，他们应用收缩-抖动（S&S）改进算法对初始装填布局进行局部优化，进一步减小容器半径。该方法将时间复杂度从 $O(n^2)$降至 $O(n\log\ n)$。与经典算法[48, 49], 相比，VOROPACK-D算法的求解速度显著更快，与最优解的平均偏差介于2%至5%之间。然而，S&S增强算法的应用会导致计算时间显著增加，尤其是在大规模装填问题中。

2021年，Chowdhury等人[25]提出了沃罗诺伊-萤火虫群优化-K均值（VGSOK）算法，该算法融合了K均值聚类、沃罗诺伊图空间建模和萤火虫群优化。算法通过K均值将圆心划分为若干密集区域，并将聚类中心视为生成点，以构建沃罗诺伊图。随后，算法将每个圆心建模为带有相应荧光素强度的“萤火虫”，驱动它们向各自对应的沃罗诺伊生成点收敛。VGSOK有效消除了随机部署造成的覆盖空洞，使覆盖效率提高了3–10%，降低了实际能耗，并满足了多目标优化需求。

随着应用场景对动态适应能力和收敛速度提出更高要求，研究逐渐转向改进元启发式算法的内部搜索机制。为解决陷入局部最优（导致“覆盖空洞”）以及大规模情况下收敛缓慢等问题，Wang等人 [26] 于2025年提出了多策略改进火烈鸟搜索算法（MIFSA）。MIFSA采用了多项关键策略：（1）在初始化阶段采用精英反向学习策略，以拓宽搜索范围；（2）采用非线性衰减因子控制分阶段步长，以平衡全局探索与局部开发；（3）在后期迭代中引入混沌余弦变异因子，以增强种群多样性并跳出局部最优；（4）在最后阶段采用自适应L´evy飞行机制，通过随机扰动提升解的质量。在 50×50 正方形区域内进行的实验表明，MIFSA优于原始火烈鸟搜索算法：迭代100次时覆盖率提高了7.48%，迭代200次时提高了5.68%，且圆的分布更加均匀，覆盖空洞显著减少。

为进一步解决圆分布不均所导致的低效与覆盖空洞问题，Yang等人 [27] 于2025年提出了多策略改进布谷鸟搜索算法（ICS-MS）。ICS-MS采用：（1）分阶段逐维更新策略（前期全维、后期逐维），以降低计算复杂度；（2）自适应发现概率机制，根据种群适应度方差动态调整；（3）双向搜索与全局最优引导，在保持种群多样性的同时加速收敛性；（4）对精英个体实施反向学习，以便随后在其邻域进行深度搜索。在不同圆数量（20/30）和迭代次数（200/1000）下与元启发式算法相比，ICS-MS将覆盖率提高了2.32%至22.21%，并使圆的分布更加均匀。

与此同时，Shaikh 等人[28]于2025年提出了改进混沌灰狼优化器（ICGWO）。为缓解传统灰狼优化器易陷入局部最优且收敛缓慢的问题，ICGWO将高斯混沌映射融入位置更新过程，利用混沌序列使搜索路径多样化。与需要复杂参数调优的对比算法（如粒子群优化或蚁群优化）不同，ICGWO以极小的参数开销在全局探索与局部开发之间保持平衡。在相同圆数量下进行比较时，ICGWO的覆盖率较现有同类算法高出2.18%–16.41%，在大规模部署中优势尤为显著。该算法展现出较高的可扩展性与鲁棒性，适用于复杂的现实场景。

3 预备知识

3.1 凸集与凸包

本文聚焦于凸多边形的覆盖问题，这一概念源自凸集和凸包。

定义 1（凸集）给定集合 S，若对于任意 $\forall x、\ y \in S$，$(1-\lambda)x+\lambda y \in S$对所有 $\forall \lambda \in [0,1]$均成立，则 S是一个凸集。从几何上直观地说，这意味着凸集包含其中任意两点之间的线段。

定义 2（凸包）集合 S 的凸包 $Conv(S)$ 是包含 S 的最小凸集，也就是所有包含 S 的凸集的交集。

平面上的凸多边形可以定义为有限点集 S 的凸包的边界。通常，从凸包的任意一个顶点出发，按照顺时针（或逆时针）顺序依次连接所有顶点，形成一个封闭区域，以此表示该多边形。因此，构造凸多边形的问题可归结为计算给定点集 S 的凸包。众所周知的凸包计算算法包括 Graham 扫描算法 [50], 、Jarvis 行进算法 [51], 和单调链算法 [52]。

引理 1 *[53]*。给定任意包含 n 个点的平面点集，其凸包可在 $O(n\log\ n)$ 时间内构造完成。

3.2 沃罗诺伊图

沃罗诺伊图是对平面区域的一种划分，它将平面分成 n 个单元。给定一个点集 $S=\{s_1,s_2,\cdots,s_n\}$，这些 n 个互不相同的点称为生成点，每个生成点对应一个划分单元。例如，与生成点 s_i 对应的单元记作 $\mathcal{V}(s_i)$，与点集 S 相关联的沃罗诺伊图记作 $Vor(S)$。

引理 2 *[53]*。点 $p \in \mathcal{V}(s_i)$ 当且仅当 $\delta(p,s_i) < \delta(p,s_j)$ 对所有 $s_j \in S(j \neq i)$均成立，其中 $\delta(p,s_i)$ 表示 p 与 s_i 之间的欧几里得距离。

对于平面内任意两个不同的点 s_i 和 s_j，线段 s_is_j 的垂直平分线将平面划分为两个半平面。包含 s_i 的开半平面记作 $h(s_i,s_j)$，而包含 s_j 的开半平面记作 $h(s_j,s_i)$。由此可知，s_i 的沃罗诺伊单元是由其他生成点所诱导的所有此类半平面的交集：

$$\mathcal{V}(s_i)=\bigcap_{\substack{1\le j\le n\\ j\neq i}} h(s_i,s_j). \tag{5}$$

集合 S 的沃罗诺伊图是所有沃罗诺伊单元的并集：

$$Vor(S)=\bigcup_{i=1}^{n}\mathcal{V}(s_i). \tag{6}$$

这一构造如图1所示。

如果 S 中的所有点都共线，那么 $Vor(S)$ 由 $n-1$ 条平行线组成。否则，Vor(S) 是连通的，且其每条边要么是线段，要么是射线。

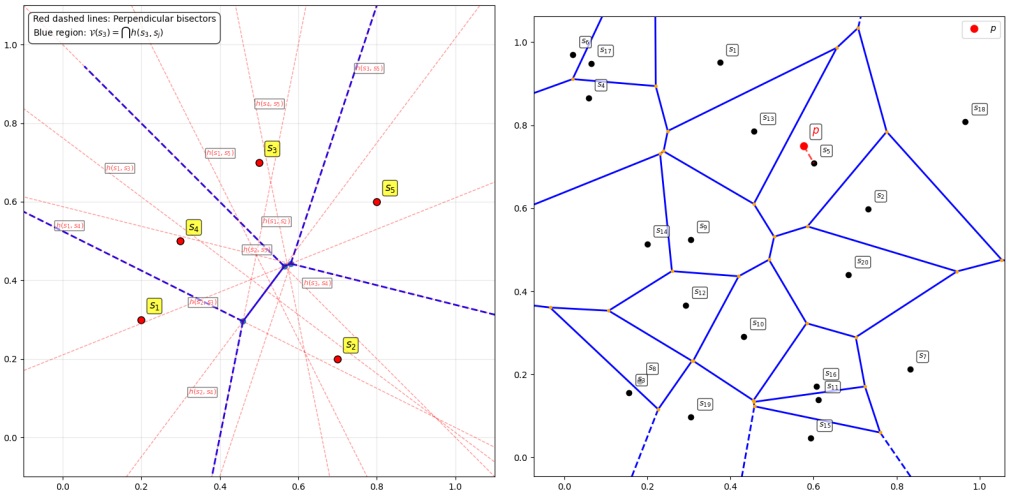
引理 3 *[53]*。如果 $n \geq 3$，则 n 个生成点的沃罗诺伊图至多有 $3n-6$ 条边和 $2n-5$ 个顶点。

定义 3（最大空圆）对于$\mathrm{Vor}(S)$ 中的任意点 q，以 q 为圆心且内部不含 S 中任何生成点的最大圆，称为 q 关于 S 的最大空圆，记作 $C_S(q)$。

定理 1 *[53]*。对于任意点集 S 的沃罗诺伊图 $Vor(S)$，以下性质成立：

(1) 点 q 是 $\mathrm{Vor}(S)$ 的顶点，当且仅当其最大空圆 $C_S(q)$ 的边界上至少存在三个生成点。

(2) s_i 与 s_j 之间的平分线在$\mathrm{Vor}(S)$ 中定义一条边，当且仅当该平分线上存在一点 q，使得 $C_S(q)$的边界经过 s_i 和 s_j，且不包含其他生成点。



(a) 所有半平面的交　　(b) $\mathbb{R}^2$中的沃罗诺伊图

图1：通过半平面交构造Voronoi图

如T所示　　定理1 与图2所示，Voronoi图的顶点和边具有明确的几何解释　　.

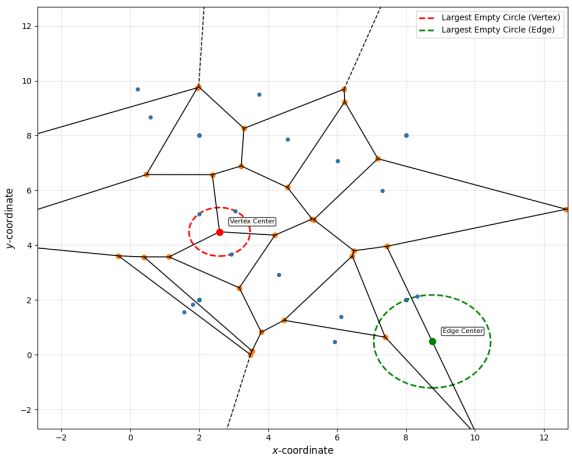


图2：两类最大空圆

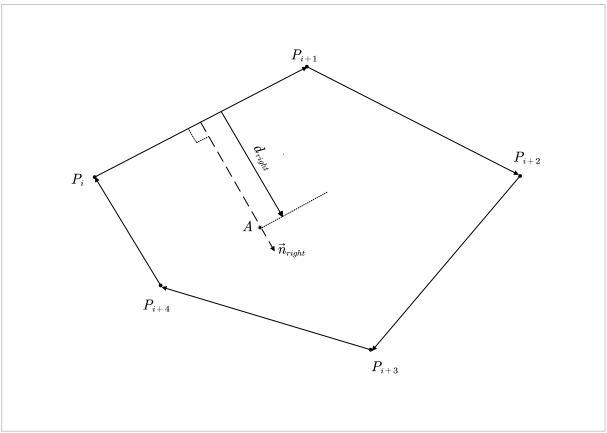


图 3：由顺时针定向线段组成的凸多边形

Voronoi 图通常通过分治方法 [54] 构建，或借助几何对偶性 [55]，间接通过 Delaunay三角剖分构建。我们采用这一框架，通过构造圆心的沃罗诺伊图来评估分布均匀性，详见第 4.2 节。

3.3 几何计算

本文聚焦于凸多边形，因此有必要验证任意多边形的凸性。主要方法是“旋转法” [56], ，该方法将多边形边界视为由方向一致的定向线段构成的闭合序列，如图 3 所示。

引理 4 *[56]*。对于多边形的每个顶点 $P_i = (x_i, y_i)$，依次计算由连续三个顶点 $P_i \to P_{i+1} \to P_{i+2}$形成的向量的叉积。如果所有叉积的符号相同（要么全部为正，要么全部为负），则该多边形为凸多边形。具体而言，设 $\overrightarrow{P_iP_{i+1}} = (x_{i+1} - x_i, y_{i+1} - y_i)$和 $\overrightarrow{P_{i+1}P_{i+2}} = (x_{i+2} - x_{i+1}, y_{i+2} - y_{i+1})$，则叉积 $\overrightarrow{P_iP_{i+1}} \times \overrightarrow{P_{i+1}P_{i+2}} (i = 1, \ldots, n-2)$ 的符号必须保持一致。

为了确定圆的中心是否　　位于凸多边形内部，有必要引入有符号距离函数　　.

定义 4（有符号距离函数）如果 P是度量空间 X 中度量 d下的一个子集，则有符号距离函数 f 定义如下：

$$f_P(x) = \begin{cases} d(x, \partial P), & if\ x \in P \\ -d(x, \partial P), & if\ x \notin P \end{cases} .$$

这里， ∂P表示 P的边界。对于任意 $x \in X$, $d(x, \partial P) := \inf_{y \in \partial P} d(x, y)$。

为简化记号和计算，令 $d(x,y)$表示点 x 和 y之间的欧几里得距离。有向线段的右法向量和右侧距离定义如下。

定义 5（右法向量）给定一条有向线段 $\overrightarrow{P_1P_2}$，其右法向量是从 P_1 到 P_2 [57]运动时指向右侧的单位法向量。形式化地，对于方向向量 $\vec{v}=(v_x,v_y)=\overrightarrow{P_1P_2}$，右法向量表示为 $\vec{n}_{\text{right}}=(v_y,-v_x)$。

定义 6（右侧距离）点 A 到有向线段 $\overrightarrow{P_1P_2}$ 的右侧距离定义如下：如果 A 位于线段右侧，则该距离为正；如果位于左侧，则为负；如果在线段所在直线上，则为零。其计算公式为：$d_{\mathrm{P_1P_2}}(A)=\frac{\overrightarrow{P_1A}\cdot\vec{n}_{\text{right}}}{\|\vec{n}_{right}\|}$，其中 $\vec{n}_{\text{right}}$ 为上述定义的右法向量。

于是我们得到引理5，用于凸多边形中的点包含判定。

引理 5 点 A 位于凸多边形 P内，当且仅当 $f_P(A)=\min_j d_j(A)>0$，其中 j 表示多边形的边索引。

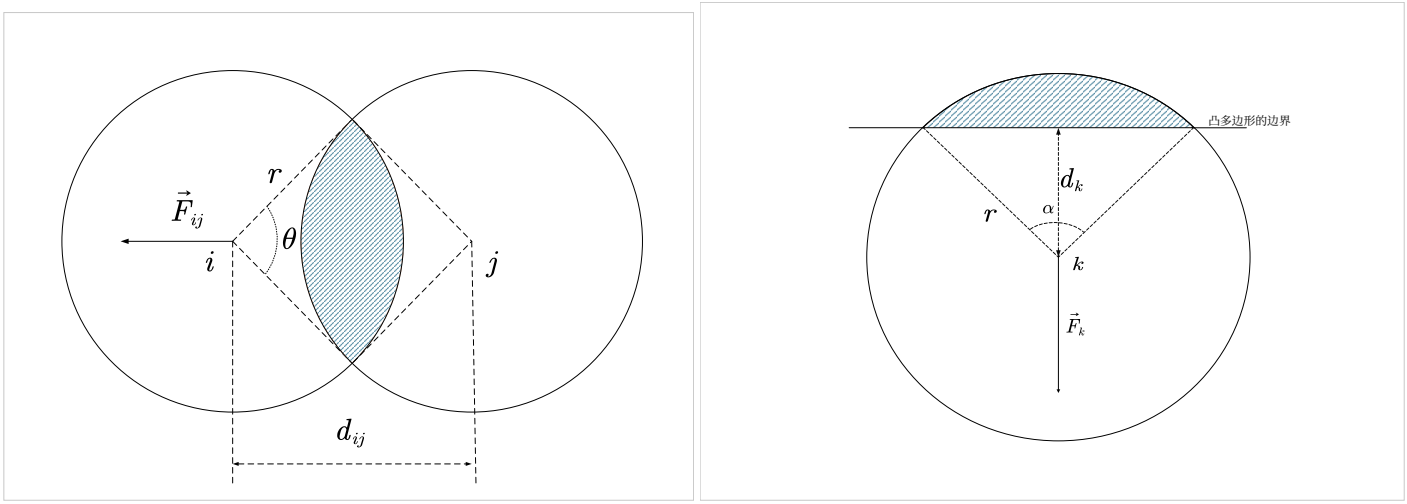
4 改进的准物理动力学算法

Huang 等人最初提出准物理方法，用于求解等圆装填问题。其核心思想是将圆建模为被限制在凸多边形边界内的弹性球，凸多边形边界充当容器壁。通过模拟这些弹性球之间的碰撞和挤压运动，它们会逐渐在多边形内分散开来，从而形成高效的装填构型。这一思路可应用于凸多边形中的圆覆盖优化问题。我们进一步改进该方法，通过优化初始分布、迭代过程和边界覆盖策略，提出了一种改进的准物理动力学算法。

4.1 准物理弹性圆模型

定义 7（构型）任意时刻圆心构成的向量（记为 $\boldsymbol{C}=[x_1,y_1,x_2,y_2,\cdots,x_n,y_n]$）定义了由凸多边形和圆组成的构型。

由凸多边形与圆组成的系统涉及两类弹性力：圆间力以及圆与多边形边界之间的力。设 F_{ij} 表示圆 j 对圆 i施加的弹性力，方向从 j 指向 i，并设 d_{ij} 表示从圆心 j 到圆心 i的位移向量。类似地，设 F_k 表示多边形边界对圆施加的弹性力 k，以及 d_k从多边形边界到圆心的位移向量 k，如图4。



(a) 圆之间的力　　(b) 圆与边界之间的力

图4：两类弹性力

我们将圆与凸多边形边界之间的重叠面积，以及两个圆之间的重叠面积，分别建模为两种弹性力的大小（由阴影区域表示）。重叠面积越大，对应的力越强。因此，图 (a) 和 (b) 中分别记作 S_a 和 S_b 的阴影区域为：

$$\begin{cases}\theta=2\arccos\left(\frac{d_{ij}/2}{r}\right)\\ S_a=2\left(\frac{1}{2}r^2\theta-2r\sin\frac{\theta}{2}r\cos\frac{\theta}{2}\right)=r^2\left(\theta-\sin\theta\right)\end{cases},\tag{7}$$

$$\begin{cases}\alpha=2\arccos\left(\frac{d_k}{r}\right)\\ S_b=\frac{1}{2}r^2\alpha-2\sqrt{r^2-d_k^2}d_k\cdot\frac{1}{2}=\frac{1}{2}r^2\alpha-d_k\sqrt{r^2-d_k^2}\end{cases}.\tag{8}$$

因此，两种弹性力可表示为：

$$\begin{cases} \vec{F}_{ij} = S_a \cdot \frac{\vec{d}_{ij}}{|\vec{d}_{ij}|} \\ \vec{F}_k = S_b \cdot \frac{\vec{d}_k}{|\vec{d}_k|} \end{cases}, \tag{9}$$

其中， S_a 和 S_b 表示先前定义的重叠区域。两个圆之间的弹性力方向始终从施力圆的圆心指向受力圆的圆心，而圆与多边形边界之间的弹性力方向则由有向边的右法向量给出。

由于多个圆可能发生重叠，或一个圆可能与多边形的多条边相交，作用于圆i 上的总弹性力为：

$$\vec{F}_{\text{elastic,i}} = \sum_{j=1, j\neq i}^{n} \vec{F}_{ij} + \sum_{k=1}^{N} \vec{F}_k. \tag{10}$$

4.2 保结构初始化策略

圆的初始构型会显著影响最终近似最优解的质量和计算时间。因此，选择合适的初始构型对于后续获得高质量的优化迭代至关重要。受正六边形密堆积原理的启发[15], 我们提出了一种基于六角密堆积的保结构初始化策略。

首先，在基平面上从中心向外生成一系列同心正六边形。随后，通过缩放和仿射变换将这一构型投影到目标凸多边形中，从而将密堆积的圆排列直接映射到多边形区域内，如图5和6所示。

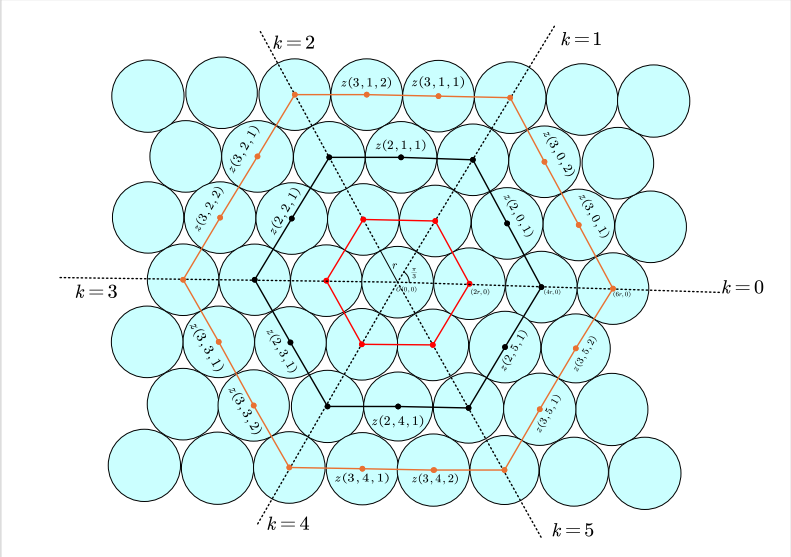


图5：从中心向外生成的圆规则六角密堆积。

这种方法确保多边形内部区域的圆已形成近乎最优的密堆积排列，无需进一步迭代优化。因此，后续优化过程可专注于调整边界附近的圆。这一策略不仅保留了大部分区域的最优堆积结构，还显著降低了迭代优化的计算成本，从而大幅提升了整体算法效率。

定义 8（圆扇形）圆扇形是由从圆心出发的两条半径及其间的圆弧围成的平面区域。根据图5，我们将 $k=i$ 与 $k=i+1$ 之间的区域称为第 i 个扇形。

假设第 0 层由一个圆组成，其圆心位于$(0,\ 0)$。第一层由围绕初始圆沿六个不同方向生成的六个圆组成，如图5所示。该模式递归延续：对于第 l<style id='16"> 层（$l \geq 1$），沿六个方向中的每个方向生成 l 个圆（相邻方向之间的角间隔为 $\frac{\pi}{3}$ 弧度）。该层中每个圆的圆心到原点的距离为 $2lr$，任意两个相邻圆的圆心之间的距离为 $2r$。为便于简洁地表述，使用复数表示圆心坐标。给定旋转角$\theta = k \cdot \frac{\pi}{3}$（其中 $k=0,\ 1,\ \ldots\ldots,\ 5$），第 m个圆在第 k个扇形的第 l层中的复坐标为

–

$$z\,(l,k,m) = 2r \cdot l \cdot e^{i \cdot k \cdot \frac{\pi}{3}} + 2r \cdot m \cdot e^{i \cdot (k+2) \cdot \frac{\pi}{3}}, \tag{11}$$

其中 $k=0,\ \cdots,\ 5,\ m=0,\ \cdots,\ l-1$。

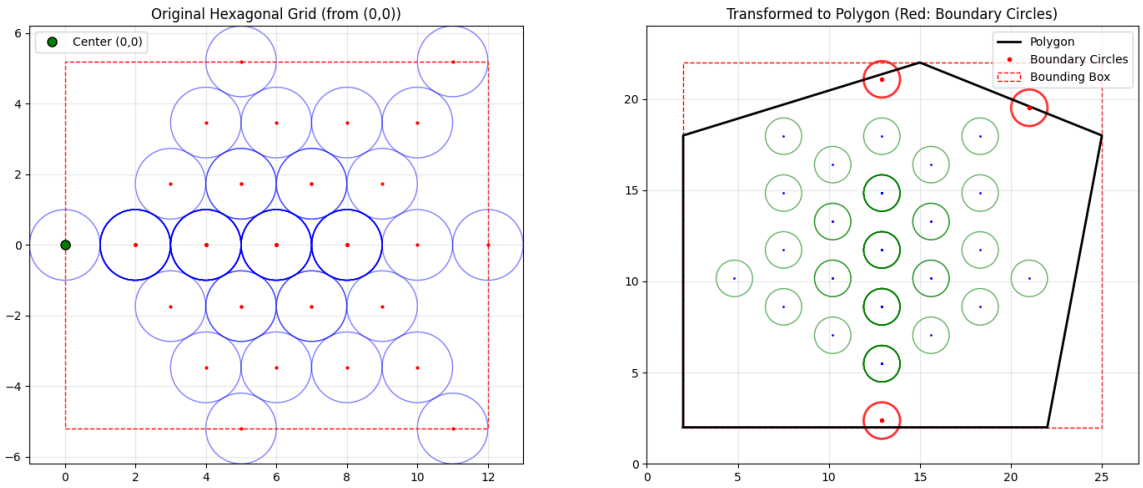


图6: 通过缩放和仿射变换将密集排列的圆投影到凸多边形区域 s

定义 9（最小外接矩形，MBR）多边形的最小外接矩形定义为能够完全包含该多边形且面积最小的矩形。对于任意凸多边形，其最小外接矩形的方向通常用于确定多边形的主方向。

设最小外接矩形的顶点坐标依次记为 $P_1=(x_1,y_1)$ $P_2=(x_2,y_2)$ $P_3=(x_3,y_3)$ $P_4=(x_4,y_4)$（按顺时针方向排列）。不失一般性，设 P_1P_2 为该矩形的最长边。则该矩形的方向向量为 $\vec{d}=(x_2-x_1,y_2-y_1)$，相应的旋转角计算为 $\theta=\arctan\left(\frac{y_2-y_1}{x_2-x_1}\right)$。例如，考虑任意随机凸多边形 $P=\{(0.5,3),(2,1),(5,2),(8,4),(7,7),(4,8),(1,5)\}$。其主方向如图7所示。

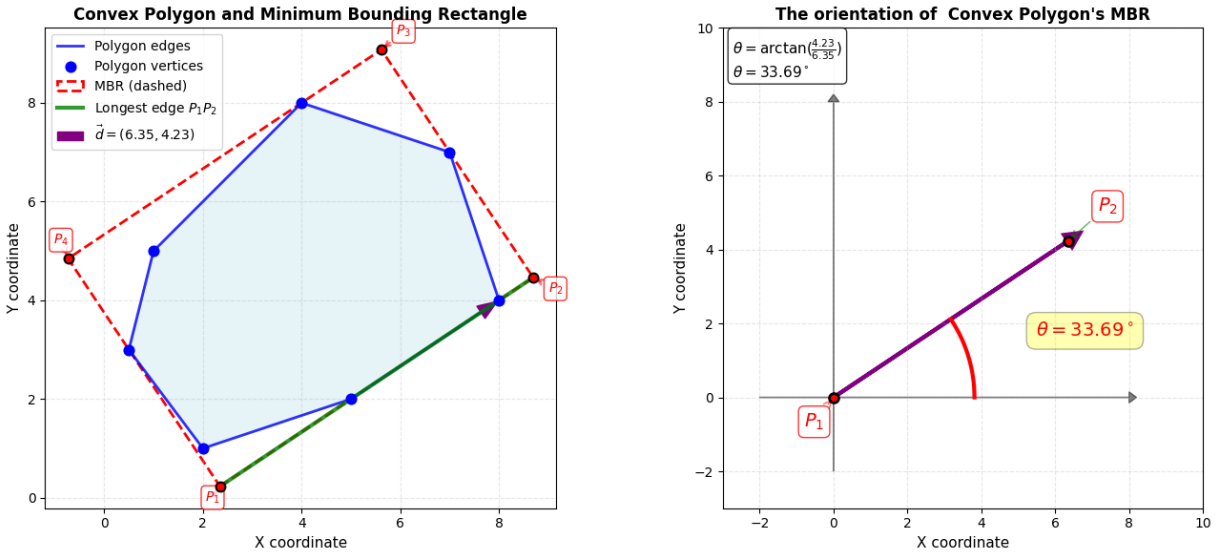


图 7：凸多边形 P 及其最小外接矩形

基于对最小外接矩形方向的分析，我们提出了一种针对紧密排列六边形圆的结构保持型初始化算法。主要步骤如下：

步骤 1： 使用旋转卡尺算法 [59, 60]计算任意凸多边形的最小外接矩形；**步骤 2：** 计算矩形的方向向量 $\vec{d}$ 和旋转角 θ；

Step 3： 将六边形格子的方向与多边形的主方向对齐，如下所示 ：

$$z_{rolated}(l,k,m)=z(l,k,m)\cdot e^{i\theta}; \tag{12}$$

步骤 4： 计算缩放因子 β，将六边形格子缩放至与多边形尺寸相匹配：

$$\begin{cases}\beta=\min\left(\frac{w_p-2r}{w_h},\frac{h_p-2r}{h_h}\right)\cdot\alpha\\ z_{scaled}(l,k,m)=\beta\cdot z_{rotated}(l,k,m)\end{cases}, \tag{13}$$

其中，w_p 和 h_p 分别表示多边形最小外接矩形的长和宽，而 w_h 和 h_h 则表示六边形格子包围盒的长和宽。 r 表示圆的半径。为防止变换后的圆

超出多边形边界，我们将 $\alpha \in (0.9,0.95)$定义为安全系数。$\frac{w_p-2r}{w_h}$ 和 $\frac{h_p-2r}{h_h}$ 分别表示多边形内可用空间与六边形格子原始尺寸在长、宽方向上的比值。引入安全系数 α，以考虑数值计算误差和多边形的不规则形状，从而为后续优化迭代预留调整空间。

步骤 5：将变换后的六边形格子置于多边形内，使格子中心与多边形质心对齐 $\boldsymbol{c}_p$，从而得到大致居中的初始化分布，如图所示。6:

$$\begin{cases} \boldsymbol{c}_p = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}_i = \left(\frac{1}{N}\sum x_i, \frac{1}{N}\sum y_i\right) \\ z_{final}\,(l,k,m) = z_{scaled}\,(l,k,m) + \boldsymbol{c}_p \end{cases}, \tag{14}$$

其中，$\boldsymbol{v}_i = (x_i,\ y_i)$ 表示多边形顶点，N 表示顶点数量。

步骤 6：使用边界感知过滤算法 [61] 对仿射变换后的圆进行边界检测，并滤除超出多边形边界的圆。随后，应用角点占据原理 [20]，在凸多边形内部附近插入圆。

初始化算法使圆能够在多边形内保持六方密堆积特性。相邻圆心之间的距离变为 $2\beta r$，同时保持六边形对称性。当圆的数量较少时，初始分布对应于最优构型，从而将最大面积覆盖问题转化为传统的堆积问题。即使圆的数量较多，该算法也能为后续迭代优化建立高质量的起点，显著提高计算效率。

算法 1 保结构初始化策略

输入：凸多边形 P，圆的数量 N，半径 r，安全系数 $\alpha \in (0.9,\ 0.95)$**输出：**初始圆位置 $\boldsymbol{C}^{(0)} = \{\boldsymbol{c}_1^{(0)}, \ldots, \boldsymbol{c}_N^{(0)}\}$

1: 步骤1：计算最小外接矩形 2: Rect $\vec{d}, \theta \leftarrow$ **旋转卡尺(**P**)**
3: $w_p, h_p \leftarrow$ **宽(Rect)，高(Rect) 4: 步骤2：生成基础六边形格点 5:** $\mathcal{L}_{\text{base}} \leftarrow$ **生成六边形格点(**N**) 根据公式(11) 6:** $w_h, h_h \leftarrow$ **包围盒尺寸(**$\mathcal{L}_{\text{base}}$**) 7: 步骤3：旋转格点 8:** $\mathcal{L}_{\text{rot}} \leftarrow \{z \cdot e^{i\theta} \mid z \in \mathcal{L}_{\text{base}}\}$**9: 步骤4：缩放格点 10:** $\beta \leftarrow$ **min** $\left(\frac{w_p-2r}{w_h}, \frac{h_p-2r}{h_h}\right) \cdot \alpha$**11:** $\mathcal{L}_{\text{scaled}} \leftarrow \{\beta \cdot z \mid z \in \mathcal{L}_{\text{rot}}\}$**12: 步骤5：将格点中心与多边形质心对齐 13:** $\boldsymbol{c}_p \leftarrow \frac{1}{|V|}$**(**$P$**)**$|\sum_{\boldsymbol{v}\in V}$**(**$P$**)** $\boldsymbol{v}$**14:** $\mathcal{L}_{\text{trans}} \leftarrow \{z + \boldsymbol{c}_p \mid z \in \mathcal{L}_{\text{scaled}}\}$**15: 步骤6：过滤并优化 16:** $\mathcal{L}_{\text{filtered}} \leftarrow$ **边界感知过滤(**$\mathcal{L}_{\text{trans}}$**,** P**) 17:** $\mathcal{L}_{\text{final}} \leftarrow$ **角点占据插入(**$\mathcal{L}_{\text{filtered}}$**,** P**) 18: 返回** $\mathcal{L}_{\text{final}}$

4.3 虚拟力与半径迭代过程

初始化所有圆心的坐标后，我们对由所有圆和凸多边形组成的系统进行动力学建模。假设每个圆 C_i 都是一个具有三个属性的动态对象：位置 $\vec{x}_i = (x_i,\ y_i)$、速度 $\vec{v}_i = (v_{ix},\ v_{iy})$ 和合外力 $\vec{F}_{\text{total},\ i} = (\vec{F}_{\text{total},\ ix},\ \vec{F}_{\text{total},\ iy})$。

为防止圆因惯性而无限移动，我们为每个圆引入黏性阻尼项 F_{friction}。设黏性系数为 μ；阻尼力的方向与速度方向相反，且大小与速度大小成正比。因此，每个圆所受的合外力为

$$\begin{cases} \vec{F}_{\text{friction}} = -\mu \cdot \vec{v} \\ \vec{F}_{\text{total,i}} = \sum_{j=1,j\neq i}^{n} \vec{F}_{ij} + \sum_{k=1}^{N} \vec{F}_k + \vec{F}_{\text{friction}} \end{cases}. \tag{15}$$

为简化计算，可将每个圆的质量设为1，时间步长设为 Δt。根据牛顿第二定律，每个圆的加速度、速度和位置的更新方程如下:

$$\begin{cases} \vec{a}_i = \vec{F}_{total,i} \\ \vec{v}_i^{(t+\Delta t)} = \vec{v}_i^{(t)} + \vec{a}_i \Delta t \\ \vec{x}_i^{(t+\Delta t)} = \vec{x}_i^{(t)} + \vec{v}_i^{(t)} \Delta t \end{cases}. \tag{16}$$

在 Pytho 在程序中，速度和位置的更新分别沿 $x-$ 轴和 $y-$ 轴进行，即 ，

$$\begin{cases} v_x^{(t+\Delta t)} = v_x^{(t)} + a_x \Delta t \\ v_y^{(t+\Delta t)} = v_y^{(t)} + a_y \Delta t \end{cases} \tag{17}$$

$$\begin{cases} x^{(t+\Delta t)} = x^{(t)} + v_x \Delta t \\ y^{(t+\Delta t)} = y^{(t)} + v_y \Delta t \end{cases} \tag{18}$$

动力学建模完成后，我们采用渐进式半径扩展策略来优化圆的分布。该策略在半径扩展与虚拟力平衡之间交替进行，以将系统从初始松弛状态驱动至目标构型。具体而言，算法从一组小于目标值的半径开始。在每次迭代中，它首先将所有半径按预定义步长 Δr_k 增大，随后进行虚拟力仿真，使各圆在当前半径下达到近似力学平衡。这一“扩展—平衡”循环持续进行，直至半径恢复到目标值，从而得到均匀且重叠最小的布局。迭代根据第 4.2 节定义的使用率（UR）终止；当连续迭代之间的 UR 变化低于给定阈值时，即认为系统已稳定收敛。

定义 10（扩展系统状态） 对于给定半径 r，所有圆的位置、速度和总外力的集合被定义为扩展系统状态，记为

$$S(r) = \{\boldsymbol{p}_i(r), \boldsymbol{v}_i(r), \boldsymbol{F}_{total,i}(r)\}_{i=1}^{n},$$

其中 $\boldsymbol{p}_i$（r）表示第 i 个圆的位置，$\boldsymbol{v}_i$（r）表示其速度，$\boldsymbol{F}_{total,\ i}$（$r$）表示作用于 *it.*

扩展系统的迭代过程可表述为

$$\begin{cases} S\left(r_{k+1}\right) = \mathcal{O}\left(S\left(r_k\right), r_{k+1}\right) \\ r_{k+1} = r_k + \Delta r_k \end{cases} \tag{19}$$

其中，$\mathcal{O}$ 表示将系统从半径为 r_k 时的状态转换为半径为 r_{k+1}时状态的优化算子。我们采用自适应步长更新圆半径，并以使用率作为半径调整的判据，具体如下:

$$\Delta r_{k+1} = \begin{cases} 2\Delta r_k & \text{if } U\left(r_k + \Delta r_k\right) > U_{\text{th}} \text{ and } C \geq C_{\text{th}} \\ \frac{1}{2}\Delta r_k & \text{if } U\left(r_k + \Delta r_k\right) \leq U_{\text{th}} \\ \Delta r_k & \text{otherwise} \end{cases} \tag{20}$$

其中，$U(r)$表示半径为 r时圆的使用率，U_{th} 为相应的使用率阈值。计数器 C 记录连续成功进行半径扩展的次数，而 C_{th} 定义了步长加倍的加速阈值。当分布呈现较高的使用率时，该设计使算法能够加速扩展；当使用率较低时，则减小步长以进行精细搜索。

算法 2 虚拟力与半径迭代过程

输入： 凸多边形 P，圆的数量 N，目标半径 r_{targe} t

确保： 优化扩展系统 $\mathcal{S}$

$r_{\text{current}} \leftarrow 0.1 \times r_{\text{target}}$ ▷ 初始小半径

$\Delta r_k \leftarrow r_{\text{target}} \times \alpha_{\text{inflate}}$ ▷ 初始膨胀步长

$\mathcal{S} \leftarrow$ 初始化六边形网格(P， N， r_{current})

$C \leftarrow 0$ **当** $\Delta r_k > \epsilon_{\text{inflate}}$ **时** ▷ 主要膨胀循环

$r_{\text{test}} \leftarrow r_{\text{current}} + \Delta r_k$ ▷ 尝试更大的半径

$\mathcal{S}_{\text{test}} \leftarrow \mathcal{S}$.复制() $\mathcal{S}_{\text{test}}$.半径 $\leftarrow r_{\text{test}}$ $\mathcal{S}_{\text{test}} \leftarrow$优化位置($\mathcal{S}_{\text{test}}$) ▷ 在新半径下优化

如果 $\mathcal{S}_{\text{test}}$.使用率 $> U_{\text{th}}$ **则** ▷ 评估系统状态

$\mathcal{S} \leftarrow \mathcal{S}_{\text{test}}$ ▷ 接受新半径

$r_{\text{current}} \leftarrow r_{\text{test}}$

$C \leftarrow C + 1$

如果 $C \geq C_{\text{th}}$ **则**

$\Delta r_k \leftarrow \Delta r_k \times 2$ ▷加速膨胀

$C \leftarrow 0$

结束如果

否则

$\Delta r_k \leftarrow \Delta r_k / 2$ ▷ 缩小步长

$C \leftarrow 0$

结束while

结束循环

返回 $\mathcal{S}$

4.4 边界环绕策略

基于保结构初始化策略和虚拟力的约束，大多数圆集中在凸多边形的内部区域。然而，一些圆不可避免地会溢出多边形边界，因此需要设计一种处理边界溢出的机制。为此，我们提出了一种边界环绕策略。其核心思想是，沿边界梯度方向将溢出圆重新定位到最近的可行位置。

我们将原问题重新表述为最小化负覆盖率，以使其与梯度下降兼容，如下所示：

$$J_{\text{cover}}(\boldsymbol{C}) = -\frac{A_{\text{cover}}(\boldsymbol{C})}{S_P}. \tag{21}$$

定义 11（边界距离项）为阻止圆超出边界，我们定义边界距离项

$$J_{boundary}(\boldsymbol{C}) = \frac{1}{n}\sum_{i=1}^{n}\left(\min_j d_j(\boldsymbol{c}_i) - \alpha r\right)^2.$$

其中 $\alpha \in [1,\infty)$ 控制圆与边界之间的期望距离， $d_j(\boldsymbol{c}_i)$ 表示圆心 $\boldsymbol{c}_i$ 到多边形第 j 条边的有符号距离。条件 $\min_j d_j(\boldsymbol{c}_i) = \alpha r$ 表明圆心恰好位于距其最近多边形边 αr 处。

为收回溢出圆并实现边界环绕优化，我们依次引入两个约束：边界约束和圆心间距约束。对于所有 i, j，必须满足以下不等式约束：

$$\begin{cases} g_j(\boldsymbol{c}_i) = -\left[d_j(\boldsymbol{c}_i) - r\right] \le 0 \\ h_{ij}(\boldsymbol{C}) = 2r - ||\boldsymbol{c}_i - \boldsymbol{c}_j|| \le 0 \end{cases}. \tag{22}$$

随后将增广拉格朗日函数表述如下：

$$\begin{aligned} \mathcal{L}(\boldsymbol{C},\lambda,\mu,\rho) = {} & J_{\text{cover}}(\boldsymbol{C}) + \beta J_{\text{boundary}}(\boldsymbol{C}) \\ & + \sum_{i,j}\left[\mu_{ij}\max\left(0, g_j(\boldsymbol{c}_i)\right) + \frac{\rho_b}{2}\max\left(0, g_j(\boldsymbol{c}_i)\right)^2\right] \\ & + \sum_{i<j}\left[\lambda_{ij}\max\left(0, h_{ij}(\boldsymbol{C})\right) + \frac{\rho_c}{2}\max\left(0, h_{ij}(\boldsymbol{C})\right)^2\right]. \end{aligned} \tag{23}$$

4.4.1 法向梯度

对边界约束求导可得法向梯度：

$$\nabla_{\perp}\mathcal{L} = -\nabla_{\boldsymbol{c}_i}\left[\frac{\rho_b}{2}\max\left(0, g_j(\boldsymbol{c}_i)\right)^2\right] = \begin{cases} \rho_b\big(r - d_j(\boldsymbol{c}_i)\big)\cdot\frac{\boldsymbol{n}_j}{\|\boldsymbol{n}_j\|}, & \text{if } d_j(\boldsymbol{c}_i) < r \\ 0, & \text{otherwise} \end{cases}. \tag{24}$$

该梯 度在圆溢出时提供向内的修正力($d_j(\boldsymbol{c}_i) < r$)，否则为零 .

4.4.2 切向梯度

为驱动溢出圆沿边界移动以实现最优放置，我们引入切向梯度来促进边界环绕。设 k 为最靠近圆心 $\boldsymbol{c}_i$的边的索引，其单位法向量为 $\boldsymbol{n} = \boldsymbol{n}_k/\|\boldsymbol{n}_k\|$。切向量 $\boldsymbol{t}$满足 $\boldsymbol{t}\cdot\boldsymbol{n} = 0$。

对边界距离项求导可得其梯度：

$$\nabla_{\boldsymbol{c}_i} J_{\text{boundary}} = \frac{2}{n}\big(d_k(\boldsymbol{c}_i) - \alpha r\big)\boldsymbol{n}. \tag{25}$$

Si由于 $\nabla_{\boldsymbol{c}_i} J_{\text{boundary}}$ 与法向量 $\boldsymbol{n}$共线，其在切向空间上的投影为零 :

$$\nabla_{\parallel} J_{\text{boundary}} = \boldsymbol{P}_{\parallel}\nabla_{\boldsymbol{c}_i} J_{\text{boundary}} = \boldsymbol{0}, \tag{26}$$

其中 $\boldsymbol{P}_{\parallel} = \boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\top}$ 是切向投影算子。

因此，为了产生 沿边界的驱动力，我们进一步引入边界环绕点的概念。

定义 12（边界环绕点）沿每条多边形边生成均匀分布的边界环绕点：

$$\boldsymbol{b}_{j,l} = \boldsymbol{v}_j + \frac{l - 0.5}{n_j}\left(\boldsymbol{v}_{j+1} - \boldsymbol{v}_j\right) + \alpha r\cdot\boldsymbol{n}_j\,(l = 1,\cdots,n_j), \tag{27}$$

其中，第 j条边上的点数定义为

$$n_j = \max\left(1, \left\lfloor n\cdot\frac{\|\boldsymbol{v}_{j+1} - \boldsymbol{v}_j\|}{L}\right\rfloor\right).$$

其中， n 表示圆的总数， L表示多边形的周长。各符号的定义见表*1*。

表 1：边界环绕点参数说明

符号	含义
$\boldsymbol{b}_{j,l}$	第 l 个边界环绕点在第 j 条边上的坐标。
$\boldsymbol{v}_j, \boldsymbol{v}_{j+1}$	多边形第 j 个和第（$j+1$）个顶点的坐标。
n_j	分配给第 j条边的边界环绕点数量。
$\frac{l-0.5}{n_j}$	取值范围为 (0，1) 的归一化位置参数。
α	边界偏移系数（$\alpha \in [0,1]$）
r	圆的半径。
$\boldsymbol{n}_j$	第 j条边的单位内法向量（指向多边形内部）。
$\mathcal{B}$	所有边界环绕点的集合： $\mathcal{B} = \{\boldsymbol{b}_{j,l} \mid \forall j, l\}$,

$\frac{l-0.5}{n_j}$ 确保点严格位于边界内侧。 α 控制沿法向方向向内偏移的程度。 $\mathcal{B}$ 作为溢出圆的候选目标位置。对于每个圆 C_i，我们应找到离它最近的边界环绕点:

$$\boldsymbol{b}_i^* = \arg\min_{\boldsymbol{b}\in\mathcal{B}} \|\boldsymbol{c}_i - \boldsymbol{b}\|. \tag{28}$$

边界环绕产生的切向吸引力如下:

$$\boldsymbol{F}_{\text{border}}(\boldsymbol{c}_i) = \gamma(\boldsymbol{b}_i^* - \boldsymbol{c}_i), \tag{29}$$

其中， $\gamma > 0$表示吸引系数。然后，我们将切向梯度定义为该吸引力在切向空间上的投影:

$$\nabla_{\parallel}\mathcal{L}_{\text{border}} = \boldsymbol{P}_{\parallel}\boldsymbol{F}_{\text{border}}(\boldsymbol{c}_i) = \gamma\boldsymbol{P}_{\parallel}(\boldsymbol{b}_i^* - \boldsymbol{c}_i). \tag{30}$$

4.4.3 圆间排斥梯度

由圆间约束产生的排斥梯度如下:

$$\begin{aligned}\nabla_{\text{circle}}\mathcal{L} &= \nabla_{\boldsymbol{c}_i}\left[\frac{\rho_c}{2}\max\left(0, h_{ij}(\boldsymbol{C})\right)^2\right] \\ &= \begin{cases}\rho_c\big(2r - \|\boldsymbol{c}_i - \boldsymbol{c}_j\|\big)\cdot\dfrac{\boldsymbol{c}_i - \boldsymbol{c}_j}{\|\boldsymbol{c}_i - \boldsymbol{c}_j\|} & \text{if } \|\boldsymbol{c}_i - \boldsymbol{c}_j\| < 2r, \\ 0 & \text{otherwise.}\end{cases}\end{aligned} \tag{31}$$

综合上述梯度，当发生溢出时，圆心优化的更新公式如下:

$$\boldsymbol{c}_i^{(t+1)} = \boldsymbol{c}_i^{(t)} - \eta\left[\nabla_{\perp}\mathcal{L} + \nabla_{\parallel}\mathcal{L}_{\text{border}} + \sum_{j\neq i}\nabla_{\text{circle}}\mathcal{L}\right], \tag{32}$$

其中， $\eta > 0$ 表示学习率。算法的收敛准则设定为

$$\max_i \|\boldsymbol{c}_i^{(t+1)} - \boldsymbol{c}_i^{(t)}\| < \epsilon, \tag{33}$$

其中， $\epsilon > 0$ 是预先设定的收敛阈值。例如，边界环绕策略如图**??**所示。

5 结论与未来工作

本文提出了一种改进的准物理 算法，用于求解任意凸多边形中的最优圆覆盖。 he improvements are mainly summ概括如下:

（1）设计了一种基于正六边形铺砌、通过缩放和仿射变换实现的保结构初始化策略，有效提升了初始解的质量，并缩短了后续迭代优化所需的时间;

(2) 构建了融入摩擦力的虚拟力场和半径扩张优化迭代模型，以扩大全局搜索空间、减少圆之间的重叠，并最终提升最终解的质量;

(3) 提出了一种基于法向梯度和切向梯度的边界环绕策略，用于找回溢出多边形边界的圆，使原本溢出的圆重新进入多边形内部进行有效覆盖，从而显著提高圆的使用率和覆盖率。

随后，本文针对矩形、正凸多边形和任意不规则凸多边形开展了小规模和大规模实验。实验结果表明，在所有测试条件下，所提出的算法在覆盖率和圆利用率这两项关键度量上均显著优于现有最新元启发式算法（ICGWO、ICM-MS、MIFSA、VGSOK），同时在边界适应性方面也表现出最低水平。尽管较低的边界适应性

对边界附近的覆盖贡献较小，但有助于防止圆溢出边界，从而提高覆盖率和使用率。在所有区域的大规模测试中，除计算时间和最小间隙处于中等水平外，IQPD算法在所有度量上均优于其他四种算法。在小规模测试中，IQPD算法在最小间隙度量上表现出色，而其分布质量和均匀性指标处于中等水平。在所有小规模测试中，IQPD算法的运行时间几乎均为最低；在所有大规模测试中，其运行时间保持在中等水平，且计算时间不会随多边形复杂度的增加而显著上升。因此，在大规模测试中，IQPD算法在多项度量上的综合优势更加明显。

尽管所提出的 IQPD算法在若干度量上优于现有方法，但受篇幅限制，我们尚未深入分析各改进模块对最终结果的贡献大小，也未对各模块的算法复杂度、收敛性和参数敏感性进行更详细的理论分析和消融实验。这将是未来进一步完善本研究的第一步。此外，本文的研究仅限于凸多边形，其凸性简化了问题建模。然而，现实场景往往更加复杂，可能涉及带有孔洞或凹角的非凸区域。因此，构建一种适用于任意二维平面区域的通用圆覆盖机制，是未来研究的重要方向。如果能够成功解决这些问题，还可以进一步研究圆半径和数量动态调整下的覆盖优化问题、非等圆覆盖优化以及三维空间中的最密堆积问题。我们预期，IQPD算法可应用于无线网络覆盖等实际问题。对这一方向的进一步研究，将为运营优化和资源分配提供有价值的解决方案。

6 致谢

本研究得到国家自然科学基金委员会（项目编号：12371016、11871083）和国家重点研发计划（项目编号：2020YFE0204200）的资助。

参考文献

[1] M. Hifi、R. M’Hallah，《受约束圆形切割问题的近似算法》，《计算机与运筹研究》31（2004）675–694。网址：https://doi.org/10.1016/S0305-0548(03)00020-0。[2] I. Castillo、F. J. Kampas、J. D. Pint´er，《通过全局优化解决圆堆积问题：数值结果与工业应用》，《欧洲运筹学杂志》191（2008）786–802。网址：https://doi.org/10.1016/j.ejor.2007.01.054。[3] Y. Kumar、K. Deep，《利用圆堆积问题确定新移动通信塔的位置》，《进化智能》17（2024年）3241–3268。网址：https://doi.org/10.1007/s12065-024-00931-w。[4] R. Kershner，《覆盖一个集合所需的圆的数量》，《美国数学杂志》61（1939）665–671。网址：https: //doi.org/10.2307/2371320。[5] J. Machchhar、G. Elber，《自由形状非凸容器中全等圆的密集堆积》，《计算机辅助几何设计》52-53（2017）13–27。网址：https://doi.org/10.1016/j.cagd.2017.03.006。[6] J. Choi、D. Jeong、H.-K. Ahn，《用两个全等圆盘覆盖凸多边形》，《计算几何》109（2023年）101936。网址：https://doi.org/10.1016/j.comgeo.2022.101936。[7] S. P. Fekete、U. Gupta、P. Keldenich、S. Shah、C. Scheffer，《用圆盘对矩形进行最坏情形最优覆盖》，《离散与计算几何》72（2024）1232–1283。网址：https://doi.org/10.1007/s00454-023-00582-1。[8] S. Gul、R. Cohen，《利用全等圆盘高效覆盖狭长凸区域》，《数学》9（2021年）。网址：https: //doi.org/10.3390/math9233056。[9] W. Akram、S. Saxena，《位似多边形的点包围问题》，《理论计算机科学》1030（2025年）115054。网址：https://doi.org/10.1016/j.tcs.2024.115054。[10] J. J. Flores、J. Mart´ınez、F. Calder´on，《圆堆积问题的进化计算解法》，《软计算》20（2016）1521–1535。网址：https://doi.org/10.1007/s00500-015-1603-y。[11] R. Netz，《希腊数学中的文本界限》，载于：《科学史、文本史》，施普林格，2004年，第161–176页。网址：https://doi.org/10.1007/1-4020-2321-9_8。[12] P. Ball，《回顾：关于六角雪花》，《自然》480（2011）455。网址：https://doi.org/10.1038/480455a。[13] A. Thue，《关于若干几何数论定理》，《斯堪的纳维亚自然科学家会议记录》14（1892）352–353。Scholar.google。.[14] A. Thue，《关于若干几何数论定理》，forhdl. skand. naturforsk. 14（1892），352–353。[2]，《关于平面上全等圆的最密排列》，Norske Vid. Selsk. Skr 1（1910年）1–9。Scholar.google。

[15] L. Fejes Tóth，论最密圆堆积，《数学评论》23 (1962) 342–349。URL：https://doi.org/10.1007/BF02565608。[16] J. Ryu、M. Lee、D. Kim、J. Kallrath、K. Sugihara、D.-S. Kim，Voropack-d：一种使用 Voronoi 图的实时圆盘装填算法，《应用数学与计算》375 (2020) 125076。URL：https://doi.org/10.1016/j.amc.2020.125076。[17] P. G. Szab´o、M. C. Mark´ot、T. Csendes、E. Specht、L. G. Casado、I. Garc´ıa，《正方形内圆装填的新方法：附程序代码》，施普林格优化及其应用丛书，施普林格，2007。URL：https://doi.org/10.1007/ 978-0-387-45676-8。[18] G. F. T´oth，用八个、九个或十个全等圆覆盖圆的最窄方案，《组合与计算几何》52 (2005) 59。URL：https://doi.org/10.1017/9781009701259.019。[19] J. Melissen、P. Schuur，用六个和七个圆覆盖矩形，《离散应用数学》99 (2000) 149–156。URL：https://doi.org/10.1016/S0166-218X(99)00130-4。[20] Y.-L. Wu、W. Huang、S. chung Lau、C. Wong、G. H. Young，一种用于求解矩形装填问题的有效拟人启发式算法，《欧洲运筹学杂志》141 (2002年) 341–358。URL：https://doi.org/10.1016/ S0377-2217(02)00129-7。[21] W. Huang、T. Ye，用于求解等圆装填问题的拟物全局优化方法，《中国科学：信息科学》54 (2011) 1333–1339。URL：https://doi.org/10.1007/s11432-011-4270-3。[22] T. Wang、G. Zhang、X. Yang、A. Vajdi，用于无线传感器网络节能聚类与路由的遗传算法，《系统与软件杂志》146 (2018) 196–214。URL：https://doi.org/10.1016/j.jss.2018.09.067。[23] D. Zhang、W. Huang，一种圆装填问题的模拟退火算法，载于：《计算科学国际会议》，第3036卷，2004年，第206–214页。URL：https://doi.org/10.1007/978-3-540-24685-5_26。[24] K. Tole、R. Moqa、J. Zheng、K. He，一种针对矩形物品的圆形装箱问题的模拟退火方法，《计算机与工业工程》176 (2023) 109004。URL：https://doi.org/10.1016/j.cie.2023.109004。[25] A. Chowdhury、D. De，基于沃罗诺伊图-萤火虫群优化-K均值算法的无线传感器网络节能覆盖优化，《Ad Hoc 网络》122 (2021) 102660。URL：https://doi.org/10.1016/j.adhoc. 2021.102660。[26]S. Wang、Q. Zhang、Y. Zheng、Y. Yue、L. Cao、M. Xiong，一种利用群体智能优化的无线传感器网络覆盖优化方法，《仿生学》10 (2025) 750。URL：https://doi.org/10.3390/ biomimetics10110750。[27] S.-Y. Yang、C.-F. Chen、K.-Q. Zhou、Y. Ou，采用多策略改进 CS 的无线传感器网络覆盖优化，《科学报告》15 (2025) 29668。URL：https://doi.org/10.1038/s41598-025-13247-1。[28] M. S. Shaikh、C. Wang、S. Xie、G. Zheng、X. Dong、S. Qiu、M. A. Ahmad、S. Raj，利用改进的混沌灰狼优化实现无线传感器网络覆盖与连通性最大化，《科学报告》15 (2025) 15706。URL：https: // doi.org/10.1038/s41598-025-00184-2。[29] S. S、S. V. Mallapur，一种用于最大化无线传感器网络区域覆盖的新型混合优化算法，《澳大利亚电气与电子工程杂志》21 (2024) 449–464。URL：https://doi.org/10.1080/1448837X.2024.2337476。[30] J. Li、J. Pang、B. Jiang、Q. Xu、E. Zhang，基于量子遗传算法的室外三维地图中5G基站部署优化，《计算机网络》269 (2025) 111431。URL：https://doi.org/10.1016/j.comnet.2025.111431。[31] E. H. Neville，数值函数方程的求解：通过一个流行谜题及其解法加以说明，《伦敦数学会会刊》s214 (1915) 308–326。URL：https://doi.org/10.1112/plms/s2_14.1.308。[32] A. Heppes、H. Melissen，用等圆覆盖矩形，《匈牙利数学期刊》34 (1997) 65–81。URL：https://doi.org/10.1023/A:1004224507766。[33] Melissen、Hans，等边三角形的最松圆覆盖，《数学杂志》70 (1997) 118–124。URL：https: //doi.org/10.2307/2691433。[34] K. J. Nurmela，关于至多用36个等圆覆盖等边三角形的猜想最优方案，《实验数学》9 (2000) 241–250。URL：https://doi.org/10.1080/10586458.2000.10504649。[35] C.-S. Shin、J.-H. Kim、S. K. Kim、K.-Y. Chwa，凸多边形的双中心问题，载于：《欧洲算法研讨会》，施普林格，1998年，第199–210页。URL：https://doi.org/10.1007/3-540-68530-8_17。

[36] S. K. Kim, C.-S. Shin, 凸多边形双中心问题的高效算法，载于：国际计算与组合学会议，S pringer，2000，pp。299–309。网址：https://doi.org/10.1007/3-540-44968-X_30.

[37] E. G. Birgin、J. L. Gardenghi、A. Laurain，使用相同圆盘覆盖集合时最优半径的界，《数学运筹学》49 (2024) 1855–1889。URL：https://doi.org/10.1287/moor.2022.0104。[38] S. Kravitz，将圆柱体装入圆柱形容器，《数学杂志》40 (1967年) 65–71。URL：https://doi.org/ 10.1080/0025570X.1967.11975768。[39] N. Mladenovi´c、F. Plastria、D. Uro˘sevi´c，将重构下降法应用于圆堆积问题，《计算机与运筹研究》32 (2005年) 2419–2434。URL：https://doi.org/10.1016/j.cor.2004.03.010。[40] W. Q. Huang、Y. Li、C. M. Li、R. C. Xu，将大小不等的圆装入圆形容器的新启发式方法，《计算机与运筹研究》33 (2006) 2125–2142。URL：https://doi.org/10.1016/j.cor.2005.01.003。[41] Z. L¨u，求解圆堆积问题的 Perm 算法，《计算机与运筹研究》35 (2008) 1742–1755。URL：https://doi.org/10.1016/j.cor.2006.10.012。[42] D. Zhu，大小不等圆装填的准人类资历排序算法，《混沌、孤子与分形》89 (2016) 506–517。URL：https://doi.org/10.1016/j.chaos.2016.02.030。[43] K. He、H. Ye、Z. Wang、J. Liu，一种用于将等圆装入圆形容器的高效准物理准人类算法，《计算机与运筹研究》92 (2018) 26–36。URL：https://doi.org/10.1016/j.cor.2017.12.002。[44] F. Carrabs、C. Cerrone、R. Cerulli，一种圆堆积问题的禁忌搜索方法，2014年第17届网络信息系统国际会议 (2014) 165–171。URL：https://doi.org/10.1109/NBiS.2014.28。[45] F. K. Miyazawa、L. L. C. Pedrosa、R. C. S. Schouery、M. Sviridenko、Y. Wakabayashi，圆及其他堆积问题的多项式时间近似方案，《Algorithmica》76 (2016) 536–568。URL：https://doi.org/10.1007/ s00453-015-0052-4。doi:10.1007/s00453-015-0052-4。[46] G. L. Orick、K. Stephenson、C. Collins，一种线性化圆装填算法，《计算几何》64 (2017) 13–29。URL：https://doi.org/10.1016/j.comgeo.2017.03.002。[47] R. Torres-Escobar、J. A. Marmolejo-Saucedo、I. Litvinchev，一种用于在矩形容器中近似装填不全等圆的二进制猴群算法，《无线网络》26 (2020) 4743–4752。URL：https://doi.org/10.1007/ s11276-018-1869-y。[48] C. L´opez、J. Beasley，利用公式空间搜索装填大小不等的圆，《计算机与运筹研究》40 (2013) 1276–1288。URL：https://doi.org/10.1016/j.cor.2012.11.022。[49] I. Al-Mudahka、M. Hifi、R. M'Hallah，在最小圆中装填圆：一种自适应混合算法，《运筹学会杂志》62 (2011) 1917–1930。URL：https://doi.org/10.1057/jors.2010.157。[50] R. Graham，一种用于确定有限平面点集凸包的高效算法，《信息处理快报》1 (1972) 132–133。URL：https://doi.org/10.1016/0020-0190(72)90045-2。[51] R. Jarvis，关于识别平面有限点集的凸包，《信息处理快报》2 (1973) 18–21。URL：https://doi.org/10.1016/0020-0190(73)90020-3。[52] A. Andrew，另一种用于二维凸包的高效算法，《信息处理快报》9 (1979) 216–219。URL：https://doi.org/10.1016/0020-0190(79)90072-3。[53] M. De Berg、O. Cheong、M. Van Kreveld、M. Overmars，《计算几何：算法与应用》，施普林格，2008。[54] A. Okabe、B. Boots、K. Sugihara、S. Chiu，《空间镶嵌：沃罗诺伊图的概念与应用》，第43卷，John Wiley & Sons，2000。URL：https://doi.org/10.2307/2687299。[55] F. P. Preparata、M. I. Shamos，《计算几何：导论》，施普林格科学与商业媒体，2012。[56] P. Schorn、F. Fisher，I.2——检验多边形的凸性，载于：P. S. Heckbert（编），《图形珍宝》，Academic Press，1994年，第7–15页。URL：https://doi.org/10.1016/B978-0-12-336156-1.50011-2。[57] J. H. Hubbard、B. B. Hubbard，《向量微积分、线性代数与微分形式：统一方法》，Matrix Editions，2015。[58] H. Wenqi、K. Yan，一种求解圆盘装填问题的启发式拟物策略，《仿真建模实践与理论》10 (2002年) 195–207。URL：https://doi.org/10.1016/S1569-190X(02)00099-0。[59]C. Alegr´ıa、D. Orden、C. Seara、J. Urrutia，旋转条件下最小面积直角凸包的高效计算及其推广，《全球优化杂志》79 (2020) 687–714。URL：https://doi.org/10.1007/ s10898-020-00953-5。

[60] H. Zhen-hui, 《任意多边形最小面积外接矩形的计算算法》, 工程图学学报 1 (1), 122-126, 2008 (2008)。[61] X. Xu, R. Yang, N. Wang, 基于区域化半径辅助的鲁棒圆检测器, 模式识别 149 (2024) 110256。网址: https://doi.org/10.1016/j.patcog.2024.110256。