

文章编号:1671-6833(2025)01-0042-09

k -center 问题的算法研究综述

王晓峰^{1,2}, 华盈盈¹, 王军霞¹, 彭庆媛¹, 何飞¹

(1. 北方民族大学 计算机科学与工程学院, 宁夏 银川 750021; 2. 北方民族大学 图像图形智能处理国家民委重点实验室, 宁夏 银川 750021)

摘要: k -center 问题是设施选址的基础问题, 同样是 NP 难问题, 在分配、紧急服务等领域也有着实际的应用。随着问题规模的扩大, 原有的算法已不再适用, 需要进一步优化或者改进。为了找到求解该问题的高效算法, 对现有算法进行研究。对各类求解 k -center 问题的算法进行梳理, 将求解算法划分为精确算法、启发式算法、元启发式算法、近似算法等, 从算法原理、改进思路、性能和精度等方面进行对比综述。精确算法在求解小规模 k -center 问题时可在多项式时间内得到最优解, 但是算法效率低, 不适用于大规模问题; 启发式算法可以在多项式时间内给出相对最优解, 但是没有理论保证, 无法衡量与最优解的关系; 元启发式算法可对目前存在的智能优化算法进行改进, 给出相对最优解, 但是解的质量无法保证; 利用近似算法得到的解具有近似比保证, 有较大的理论研究价值, 但是实用价值较弱。目前求解 k -center 问题的元启发式算法已取得一定的研究成果, 但是在求解时间、求解规模、算法效率等方面仍待突破, 这将是未来 k -center 问题的研究重点。

关键词: k -center 问题; 精确算法; 近似算法; 蜂群优化; 遗传算法

中图分类号: TP301 **文献标志码:** A **doi:** 10.13705/j.issn.1671-6833.2024.01.009

近年来, k -center 问题受到了人们的广泛关注。 k -center 问题是一个经典的 NP 难问题^[1], 广泛应用于设施选址^[2-7]、聚类^[5]、紧急服务^[8-10]、分配^[11]等领域。给定完全无向图 $G = (V, E)$, 它的边费用满足三角不等式。对任意集 $C \subseteq V$ 和顶点 $v \in V$, 定义 $connect(v, C)$ 是 v 和 C 的顶点之间的最便宜边的费用。 k -center 问题是找 $|C| = k$ 的集合 $C \subseteq V$ 使得 $\max_v \{connect(v, C)\}$ 达到最小^[1,12], 其中 k 是整数。

k -center 问题的目标是在图中找到 $k(k > 1)$ 个中心, 使得从最远顶点到最近顶点的距离最小。这些中心沿着任何边定位, 称之为绝对 k -center 问题^[13]; 若局限于顶点, 称之为顶点 k -center 问题^[13]; 若每个顶点的选择都可能有一个代价, 称之为加权顶点 k -center 问题^[13]; 若输入是满足有向三角不等式的图, 则称之为非对称 k -center 问题^[14]; 若每个中心只能参与固定数量的顶点, 则称之为容量限制 k -center 问题^[15]; 若中心点从先定义的直线或多边形中选择, 则称之为对齐 k -center 问题^[16]; 若每个选择中心

点附近都有一组 $\alpha \leq k$ 的中心, 则称之为容错 k -center 问题^[17]; 若中心点达到顶点的最小值, 则称之为具有最小覆盖的 k -center 问题^[18]; 若 $m(m < k)$ 个中心在顶点集合中, 其余的中心可在任意位置, 则称之为混合 k -center 问题^[19]。

现有的解决 k -center 问题的算法有依据给定的 k 值寻找该取值下高效的精确解的精确算法和高效的启发式或元启发式的求解策略等。但是目前存在的求解问题的算法都有其各自的优缺点。例如, 当问题规模较小时精确算法在多项式时间内找到最优解, 但随着问题规模的增大, 时间复杂度呈指数级增长。启发式算法及元启发式算法在多项式时间内给出问题的相对最优解, 但是没有理论的保证, 且无法衡量与最优解的关系。此外, 目前仅有 k -center 问题的某一类算法的综述和对比, 对于 k -center 问题的各种算法的对比和分析较少。因此, 本文对 k -center 问题的求解算法进行归纳总结, 同时进一步指明关于 k -center 问题的研究方向, 以寻找新型高效算法求解 k -center 问题。

收稿日期: 2024-01-10; 修订日期: 2024-03-30

基金项目: 国家自然科学基金资助项目(62062001); 宁夏青年拔尖人才项目(2021)

作者简介: 王晓峰(1980—), 男, 甘肃会宁人, 北方民族大学副教授, 博士, 主要从事算法分析与设计、人工智能等研究, E-mail: xfwang@nmu.edu.cn。

引用本文: 王晓峰, 华盈盈, 王军霞, 等. k -center 问题的算法研究综述[J]. 郑州大学学报(工学版), 2025, 46(1): 42-50, 97. (WANG X F, HUA Y Y, WANG J X, et al. The survey of algorithms for k -center problem[J]. Journal of Zhengzhou University (Engineering Science), 2025, 46(1): 42-50, 97.)

1 精确算法

作为一个经典 NP 难问题, k -center 问题在精确算法、启发式算法、元启发式算法、近似算法等算法中得到广泛研究。尤其是精确算法,由于早期学者将 k -center 问题作为图论问题来解决,与图论相关的算法都是精确算法,因此早期 k -center 问题都是使用精确算法来解决的。

Minieka^[20]提出了求解 k -center 问题的算法,该算法是将 k -center 问题分解为一系列的集合覆盖问题,通过求解集合覆盖问题,得到 k -center 问题的解,但是此算法的效率非常低。因此,Daskin^[2]将 Minieka^[20]提出的算法改进为基于集合覆盖的二分搜索算法,使得算法能够缩紧最优解上界与下界间的距离。不久,Daskin^[21]改进了自己提出的算法,采用了搜索效率更高的最大集合覆盖模型,利用二分法系统地减小了最优解上下界之间的距离从而得到 k -center 问题的解。Ilhan 等^[22]在 Daskin^[21]的最大集合覆盖模型基础上,提出了一种两阶段算法:第一阶段通过求解一系列基于线性规划公式的可行性问题来计算问题最优解的下界;第二阶段,使用可行性问题来检查是否是 k -center 问题的解。Al-Khedhairi 等^[23]对 Daskin^[21]和 Ilhan 等^[22]的算法进行修改,提出了新的整数规划公式、该公式的收紧形式以及基于该公式的连续限制的新方法。新方法的一个特殊化是在二元约束下通过求解一系列结构简单的识别形式的整数规划来获得最优 k -center 的解。该算法更新了大多数数据集上的最优解,并且大大缩短了算法运行时间。

2 启发式算法

精确算法在求解小规模 k -center 问题上有较好的效果,但是通过实验发现用精确算法求解大规模 k -center 问题无论是在计算结果还是运行时间上,都没有很好的效果,因此学者尝试使用一些启发式算法来求解 k -center 问题。Mladenović 等^[24]提出将禁忌搜索算法(tabu search, TS)与变邻域搜索算法(variable neighborhood search, VNS)相结合的启发式算法求解 k -center 问题,实验证明该算法既保证了解的质量又保证了运行时间。Pacheco 等^[25]提出了一种结合了不同策略的分散搜索(scatter search, SS)算法来求解 k -center 问题,这些策略包括局部搜索、随机贪婪自适应搜索和路径重新链接,该算法能够提高运行时间,但是此算法只能用于求解小规模问题。

2.1 禁忌搜索和变邻域搜索算法

Glover^[26]提出的禁忌搜索算法因其在组合优化问题上的优秀表现而广为人知。禁忌搜索算法的核心是禁忌,该算法产生一个禁忌表来确保近期访问过的解在禁忌步长内不会再访问,通过禁忌表驱动算法探索空间的未知区域来避免程序进入局部最优。但是禁忌搜索算法有其局限性,比如刚被禁忌的动作的邻域有更好的解,但是该解被禁忌了,所以禁忌算法存在一种解禁策略。例如当被禁忌的解比没被禁忌的解好,且比当前最优解更好,则算法会选择被禁忌掉的解。同时禁忌步长的选择对算法的性能也有很重要的影响:禁忌步长太短时,算法在较小的范围内进行搜索,容易陷入局部最优;当禁忌步长过大时,会导致解空间内全是被禁忌掉的解,会使得搜索的效率不够高。

变邻域搜索算法^[27]被用于解决组合和全局优化问题。其基本思想是在局部搜索算法中系统地改变邻域结构。该算法保持以相同的解为中心,直到找到比现有解更好的另一个解,然后跳到新的解。在算法开始时需要设计多个邻域结构,并按基数递增的顺序给其编号,算法按照编号在邻域内进行搜索,当算法陷入局部最优无法找到更好的解时,算法选择变换邻域结构来找到更好的解。通过利用适用于大多数组合问题的局部最小值的经验特性,基本的 VNS 算法确保了两个重要的优势:①停留在当前解的邻近区域,搜索是在解空间的一个有吸引力的区域进行的,而且这个区域不会受到禁止移动的干扰;②由于某些解的属性已经处于它们的最优值,局部搜索使用的迭代次数比随机解初始化时少,因此,它可以在相同时间内访问几个高质量的局部最优解,而随机解只能访问一个。

2.2 分散搜索算法

分散搜索是所谓的进化方法的一个例子,与其他进化方法相比区别在于它的搜索机制不仅仅基于随机化^[28]。SS 的特点是使用解的参考集,在每一个步骤中参考解被组合以生成新的解,并根据一些系统规则更新当前的参考集(PefSet)。分散搜索算法的步骤如下。

步骤 1 用多样化生成方法生成 $Ksize$ 解的初始集 K 。

步骤 2 用一种改进方法改进这些解。

步骤 3 使用这些解构建一个初始的 PefSet。

步骤 4 重复以下步骤:从 PefSet 中获取解所对应的所有子集;对这些子集应用组合方法并获得新的解;用改进方法改进这些解;更新 PefSet,考虑

这些新的解,直到 $PefSet$ 稳定(即没有包含新的解)。

步骤 5 如果在 $maxiter$ 次的迭代(步骤 1~4)中没有改进的情况下结束,则停止其他操作,并返回步骤 1。

2.2.1 多样化方法

多样化方法常应用随机贪婪构造启发式。大多数随机贪婪自适应搜索还包括一个局部搜索,用于改进使用随机贪婪函数生成的解。多样化方法由以下步骤组成。

步骤 1 设 $X = \emptyset$ 。

步骤 2 计算 $\Delta_j, \forall j \in V \setminus X$ 。

步骤 3 计算 $\Delta_{\max} = \max\{\Delta_j/j \in V \setminus L\}$ 和 $\Delta_{\min} = \min\{\Delta_j/j \in V \setminus L\}$ 。

步骤 4 定义 $L = \{j \in V/L \leq \alpha \Delta_{\min} + (1 - \alpha) \Delta_{\max}\}$ 。

步骤 5 随机选择 $j^* \in L$ 。

步骤 6 设 $X = X \cup \{j^*\}$ 。

2.2.2 组合法

每对参考解被组合以生成新的解,该组合方法基于路径重链策略。其基本思想是在两个好的解决方案之间的路径中,可能会发现其他质量相似(甚至更好)的解。该算法建立一条连接初始解和指导解的路径,该路径中包含许多中间解,选择中间解,使彼此和极端尽可能相等,然后将该改进方法应用于这些中间解。图 1 显示了这个过程。

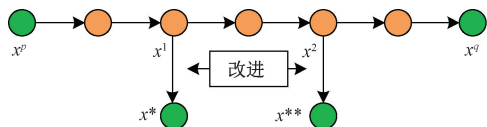


图 1 通过使用路径重新链接生成新的解

Figure 1 Generation of new solutions by using path relinking

3 元启发式算法

随着 k -center 问题概念的逐渐完善,蒋建林等^[29]提出了一种单亲遗传和模拟退火相结合的混合算法 PGASA 求解 k -center 问题,该算法充分结合了单亲遗传算法和模拟退火算法的优点,提高了优化过程中的搜索行为能力,提升了算法全局优化和局部搜索性能。Davidović 等^[30]提出了一种基于蜂群优化(bee colony optimization,BCO)算法的变种算法求解 k -center 问题,该算法与其他方法相比具有较强的竞争力,可以在很短的时间内生成高质量的解。但是蜂群优化算法存在搜索能力不强等缺点,因此 Yurtkuran 等^[31]提出了一种改进的人工蜂群

(modified artificial bee colony, MABC) 算法,该算法使用多种搜索策略和基于随机密钥的编码方案有效地解决了 k -center 问题。此外,包敏泽等^[32]提出了一种将人工蜂群算法与遗传算法相结合的算法来求解 k -center 问题,提高了算法的性能。

3.1 单亲遗传模拟退火算法

单亲遗传模拟退火算法^[29]结合单亲遗传算法和模拟退火算法的优点,简化遗传操作的过程,增强局部优化能力。此外,该算法根据个体的优劣及迭代情况来选择个体,提高了算法的全局优化和局部搜索的能力^[29]。算法步骤如下。

步骤 1 采用实数编码,随机产生初始种群 K , 评价个体适应度,将种群规模设置为 K_c ; 计算初始温度 T_0 。

步骤 2 对于种群 K , 选出个体进行自适应基因重组操作,生成新的种群 K' 。

步骤 3 对于种群 K' , 选出个体进行模拟退火操作,生成新的种群 K'' 。对于要进行模拟退火操作的每个个体,先令 $b = 0, c = 0$, 其中 b 为局部搜索次数, c 为局部搜索中被接受的新个体数量,然后执行如下操作。

对 C 进行局部搜索,得到 C' , 同时令 $b \leftarrow b + 1$ 。如果 $\Delta f(C, C') \leq 0$, 则令 $C = C', c \leftarrow c + 1$; 否则产生一个 $(0, 1)$ 随机数 r 。如果 $\exp(-\Delta f(C, C')/T) > r$, 则令 $C = C', c \leftarrow c + 1$ 。如果 $c > \text{round}(n/10)$ 或 $b > R$, 则执行步骤 4; 否则继续执行局部搜索。

步骤 4 重新评价种群 K'' 中个体的适应度,令 $K = K''$ 。

步骤 5 温度下降,设 $T = T_0/(1+t), t \leftarrow t+1$ 。

步骤 6 设定新初始温度,如果 $t = 6$, 则令 $T_0 = T_0', T = T_0$ 。

步骤 7 产生 $(0, 0.1)$ 的随机数 $rand$, 如果 $T < rand$, 则算法终止,输出当前种群中适应度最小的个体,即产生近似最优解;否则执行步骤 2 继续迭代。

3.2 蜂群优化及其改进算法

3.2.1 蜂群优化算法

蜂群优化算法是探索蜜蜂在采蜜过程中应用集体智能的方法之一,由 Lužić 等^[33-35]提出,迄今已成功地应用于各种实际生活中的优化问题,例如旅行商问题、车辆路径问题等。

BCO 算法的基本思想是建立多智能体系统(人工蜜蜂群体),该算法利用蜜蜂在采蜜过程中使用的原理来寻找各种组合优化问题的最优解。人工蜂

群通常由数量较少的个体(如 5 个或 10 个)组成,通过搜索空间寻找可行解,为了找到越来越好的解,人工蜜蜂自主协作交换信息,利用集体知识和相互分享的信息,将注意力集中在更有希望的领域,并慢慢地放弃不太有希望的领域内的解,逐步改进它们的解。

但是 BCO 算法只能在小规模(50 个节点和 4 个中心)的情况下找到最优解,当问题的规模增大时,BCO 算法不能够找到问题的最优解。此外, k -center 问题具有很大的随机性和组合复杂性,而在构造解时嵌入的启发式概念不能有效地克服这些问题,因此 Davidović 等^[30]提出了 BCO 的一种变体 BCOi,与其他算法相比,BCOi 算法能够在更短的时间内产生高质量的解。算法步骤如下。

步骤 1 预处理,形成矩阵 D_{sort} 和 D_{circles} ,从矩阵 D 开始,对矩阵 D 的每一行进行非递减排序,并给出相应的索引,生成矩阵 D_{radius} 。

步骤 2 生成初始完整解。

步骤 3 单次迭代中通过正向传递修改当前解。

步骤 4 在每次回传开始时,对与每只蜜蜂相关联的解进行评估和比较。

步骤 5 如果在第 $u + 1$ 次开始向前传递时,蜜蜂不想探索它以前的解,那么就需要招募并遵循另一只蜜蜂的解。

3.2.2 改进的人工蜂群算法

由于目前人工蜂群算法(artificial bee colony, ABC)存在局部搜索能力不足等缺点,学者主要从两方面改进这一缺点:一是改进人工蜂群算法的搜索策略;二是将人工蜂群算法与其他启发式算法相结合来提高算法性能。Yurtkuran 等^[31]提出的改进人工蜂群(MABC)算法就是对人工蜂群算法的邻域搜索策略进行改进后得到的。

由于 ABC 算法是最早用于求解连续优化问题的,所以在目标函数的计算步骤中可以直接使用实数编码的解向量^[36-37]。为了将其应用于在 k -center 问题中,MABC 算法使用基于随机密钥的编码的方法将实数编码向量转换为节点排列。由于中心的个数受问题的限制,排序后的索引向量的第一个值被作为中心点。随后,MABC 算法随机使用不同的搜索策略来解决原始算法局部搜索能力不强的问题,利用从候选池中随机选择的搜索方程生成新的邻域解。

3.2.3 BeeGenP 算法

基于人工蜂群算法存在的缺点,另一种改进策略是包敏泽等^[32]提出的一种基于人工蜂群算法和

遗传算法的元启发式算法,称为 BeeGenP 算法。

BeeGenP 算法通过完成初始化蜂群、观察蜂选择蜜源以及侦查蜂搜索新蜜源的规则、解的选择方式、是否舍弃当前蜜源与判断算法是否结束等操作完成迭代过程。但是在采蜜蜂对解的局部搜索过程中,该算法按照一定概率对当前解进行交叉或变异,使算法能获得新的局部解。BeeGenP 算法的主要步骤如下所示。

步骤 1 初始化参数,包括种群数量、算法的最大迭代次数、一个蜜源最大的搜索次数以及交叉和变异的发生概率。

步骤 2 初始化蜂群队列,记录历史局部最优队列,记录当前可搜索蜜源的队列以及步数计数器。

步骤 3 步数计数器加 1,判断是否达到最大迭代次数,若达到最大迭代次数,则执行步骤 6,否则执行步骤 4。

步骤 4 将当前可搜索蜜源的队列中较优的解加入到局部最优解的队列中,遍历蜂群队列中的成员,做交叉或变异运算来获得局部的新解。若新解优于旧解,则替换成新解;否则,舍弃新解,并将蜜源的搜索次数加 1。将蜂群队列中最好的解加入到局部最优解的队列中,并舍弃已到达最大搜索次数的蜜源。被舍弃的蜜源上的蜜蜂,若转换为观察蜂,则根据轮盘赌的方式在当前可搜索的蜜源中选择一个蜜源;若转换为侦查蜂,则根据规则生成新的解。

步骤 5 根据蜂群队列中蜜蜂所拥有的蜜源更新可搜索蜜源的队列,转到步骤 3。

步骤 6 输出历史局部最优队列中的最优解,迭代结束。

3.2.4 算法对比

针对蜂群优化及其改进算法进行对比,分析出算法的改进思路及优缺点,具体的算法分析对比如表 1 所示。

4 近似算法

对于 k -center 问题,目前所提出的 2-近似算法都是以类似贪婪的方式运行的^[38],例如 Shmoys^[39]和 Plesník^[40]分别独立提出的 2-近似算法 Sh 算法。与其他近似算法不同,该算法需要对最优解的大小进行猜测,这是它的一个缺点。为了消除此缺点带来的影响,Gonzalez^[41]以及 Dyer 等^[13]提出了一个新的 2-近似算法,可以取消猜测最优解大小的需要。该算法是一种类似贪婪的最远点算法,在每次迭代时将最远的顶点添加到解中。虽然 2-近似算法在理论上是最好的算法,但在许多特定的情况下实际

表 1 蜂群优化及其改进算法对比			
Table 1 Comparison of bee colony optimization and its improved algorithms			
算法	改进思路	优点	缺点
蜂群优化算法	提出一种基于蜂群算法的构造性变体	具有较强的竞争力,在短时间内生成高质量的解	局部搜索能力不强
改进的人工蜂群算法	对人工蜂群算法的邻域搜索策略进行改进	使用不同的搜索策略改进了蜂群算法搜索能力不强的缺点	算法执行时间相对较长
BeeGenP 算法	人工蜂群算法与遗传算法相结合	具有较强的局部搜索能力,而且更加高效	在大规模 k -center 问题上更加耗时

上是无用的。为此 Garcia-Diaz 等^[42]提出了一个 3-近似算法,尽管近似因子为 3,但在一些小的数据集上,其性能显著优于已知的两种近似算法。此算法在一定程度上平衡了近似算法和其他算法的优点,因为它在多项式时间内运行,保证了解的质量不是任意的,并且生成的解不仅在理论上接近最优,而且在实际应用中也具有一定的竞争力。

4.1 Sh 算法

Sh 算法是一个复杂度为 $O(kn)$ 的 2-近似算法,揭示了 k -center 问题的相关结构性质:在适当的距离上从一组顶点中迭代选择中心,可以保证最优解的每个中心都接近一个选定的中心,因此,所分配给每个中心的所有顶点也接近于最优解的一个中心。图 2 展示了 Sh 算法工作的一个示例。Sh 算法步骤如下。

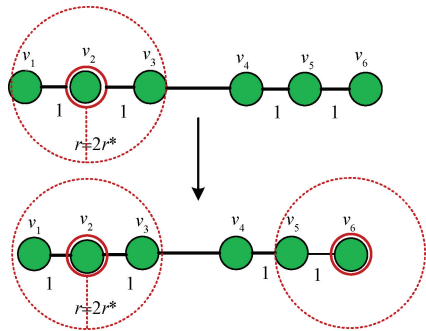


图 2 Sh 算法工作示例

Figure 2 Sh algorithm working example

- 步骤 1 猜测最优解的大小(覆盖半径)作为输入。
- 步骤 2 选择任意顶点 $v \in V$ 作为初始中心 c_1 。
- 步骤 3 在接下来的 $k - 1$ 次迭代中选择距已选中心的距离大于 $2r^*$ 的顶点为中心 c_i 。
- Sh 算法由 k 次迭代组成。在每次迭代中,计算从每个顶点 $v \in V$ 到当前解 C 的距离。通过比较 $d(v, C \setminus \{c_i\})$ 和 $d(v, c_i)$, 并使每一个 $v \in V$ 保持最小值,可分 $O(n)$ 步进行计算。因此,Sh 算法的总体复杂度为 $O(kn)$ 。

4.2 Gon 算法

Sh 算法在执行过程中需要猜测解的大小,因

此 Gonzalez^[41]和 Dyer 等^[13]引入了 Gon 算法,该算法在求解 k -center 问题时不需要对解的大小进行猜测,只需在每次迭代时将最远的顶点添加到解中。和 Sh 算法一样,Gon 算法的总体复杂度为 $O(kn)$ 。

4.3 CDS 算法

CDS 算法^[42]是求解 k -center 问题的复杂度为 $O(n^4)$ 的 3-近似算法。尽管具有次优性能,但它在大量数据集上优于 2-近似算法。关键支配集过程利用了输入图 G 的剪枝图 G_r 。上的一个最小支配集实际上是 k -center 问题的解这一原理^[43]。关键支配集过程的步骤如下。

- 步骤 1 PrunedGraph 子例程构造图 $G_r = V$ 和 E_r , 其中 $E_r = E$ 是成本小于或等于 r 的边的集合;
- 步骤 2 GetInitialScore 子例程评估 G_r 中每个顶点的初始分数,即每个顶点附近的顶点数;
- 步骤 3 在接下来的 k 次迭代的每一次迭代中,算法都会从当前解 C 中选择最远的顶点 f_i ;
- 步骤 4 从 $N(f_i) \cup f_i$ 中选择一个得分最高的顶点 c_i 作为求解的一部分;
- 步骤 5 更新每个顶点的分数。

CDS 算法用不同的 r 执行关键支配集过程, r 取自 $O(n^2)$ 个值的集合。关键支配集过程包括 3 个阶段。首先,从输入图中去除代价大于 r 的边,构造剪枝图 G_r , 这是在 $O(n^2)$ 步中完成的。其次,计算每个顶点的度,这是在 $O(n^2)$ 步中完成的。最后,在 k 次迭代的循环内执行以下步骤。在 $O(n)$ 个步骤中选择最大距离的顶点 f_i , 在 $O(n)$ 个步骤中从 $N(f_i) \cup f_i$ 中选择具有最大分数的顶点 c_i 。对于每个与 c_i 相连的邻居,首次被支配的顶点的邻居的分数将被减少 1 个单位,这是在 $n_i \times n$ 步中完成的,其中 n_i 是 c_i 的邻居数(包括 c_i)减去 C 中的顶点数。由于上一次迭代时 C 包含先前选择的中心 c_j , 其中 $j < i$, 所以在最后一次迭代($i = k$)时, $n_1 + n_2 + \dots + n_k = n$ 。因此,所有 k 次迭代的总体复杂度是 $O(n^2)$ 。

由此可得,CDS 算法的总体复杂度是 $O(n^4)$ 。

由于算法的复杂度,CDS 随着输入的增加而变得不实用,Garcia-Diaz 等^[42]在此基础上提出了复杂度为 $O(n^2 \log n)$ 的更实用的 CDS_h 算法,在大多情况下优于 CDS 算法。此外在 CDS_h 算法的基础上提出了 CDS_h+算法,该算法是重复 N 次的 CDS_h 算法,每次选取一个不同的顶点作为初始中心。

5 各种求解算法对比

5.1 精确算法及其改进算法性能对比分析

表 2 为求解 k -center 问题的各种算法的优缺点。表 3 给出了 Daskin^[21]、Ilhan 等^[22]的算法和 Al-Khedhairi 等^[23]对前两种算法进行改进后在 OR-Lib^[44]实例和 TSP-Lib^[45]实例下求解 k -center 问题的算法执行时间。从表 3 中可以看出改进后的算法在执行时间上大部分优于 Daskin^[21]、Ilhan 等^[22]提出的算法。此外,可以看出精确算法在求解小规模 k -center 问题上在短时间内可以得到最优解,

但是随着问题规模的增大,算法的执行时间无法得到保证。

5.2 蜂群优化算法及其改进算法性能对比分析

将 SS 算法、PGASA 算法、MABC 算法的运行时间、解的质量与 BCOi 算法进行对比分析如表 4 所示。其中,由于 SS 算法仅应用于小规模的问题 ($K \leq 10$) 上,所以 $K > 10$ 的部分在表 4 中不显示。可以看出,BCOi 算法在求解时间及解的质量上大都优于其他算法;而 MABC 算法在解的质量方面优于 BCOi 算法,在算法执行时间上稍慢于 BCOi 算法,却在大多数实例上的执行时间都优于已知的 SS 算法、PGASA 算法等。此外,BCOi 算法不同的参数值会对算法执行产生不同的影响。图 3 显示了在实例 Pmed9 中进行 1 000 次测试时,增加蜜蜂的数量 (B) 和一次迭代中向前(向后)传递的次数 (NC) 对达到最优解的次数的影响。从图 3 中可以看出,增加 B 或 NC 的值不能提高 BCOi 的算法效率。

表 2 求解 k -center 问题的各种算法及优缺点			
Table 2 The advantages and disadvantages of various algorithms for solving k -center problems			
求解算法	优点	缺点	
精确算法	规模较小时在多项式时间内可以得到最优解	效率低,不适用于大规模问题	
启发式算法	多项式时间内给出相对最优解	没有理论保证,无法衡量与最优解的关系	
元启发式算法	对目前存在的智能优化算法进行改进,给出相对最优解	解的质量无法保证	
近似算法	解具有近似比保证,有较大的理论研究价值	实用价值较弱	

表 3 精确算法在不同规模下的执行时间对比分析							
Table 3 Comparative analysis of execution time of exact algorithms at different scales							
数据集	实例	N	K	执行时间/s			
				文献[21]方法		文献[22]方法	
				原始	改进	原始	改进
OR-Lib	Pmed1	100	5	2.99	2.09	5.24	4.05
	Pmed5	100	33	2.07	0.73	1.29	1.49
	Pmed9	200	5	7.41	9.01	11.66	13.53
	Pmed10	200	67	3.58	2.57	3.90	2.76
	Pmed11	300	5	15.73	16.25	22.32	11.67
	Pmed15	300	100	18.05	4.40	8.91	4.43
	Pmed17	400	10	27.75	27.06	137.50	30.88
	Pmed20	400	133	24.04	9.30	13.53	13.35
TSP-Lib	pr439	439	40	39.66	33.03	35.04	32.52
	pr439	439	5	53.13	51.74	53.43	53.08
	lin318	318	20	15.53	14.49	14.60	13.54
	lin318	318	5	19.67	22.06	18.85	18.68
	d493	493	40	73.03	79.97	333.40	45.38
	d493	493	10	82.69	82.25	85.71	60.13
	d657	657	20	215.40	255.80	1047.50	301.15
	d657	657	5	158.60	154.70	117.80	100.56

表 4 蜂群优化算法及其改进算法的性能对比分析

Table 4 Comparative analysis of the performance of bee colony optimization algorithm and its improved algorithms

实例	N	K	最优解	SS		PGASA		BCOi		MABC	
				时间/s	最佳解	时间/s	最佳解	时间/s	最佳解	时间/s	最佳解
Pmed1	100	5	127	0.11	127	1.66	127	0	127	0.00	127
Pmed3	100	10	93	0.49	93	1.58	93	0	93	0.09	93
Pmed7	200	10	64	16.31	64	2.63	64	0	64	0.19	64
Pmed10	200	67	20			2.65	20	0.03	20	1.45	20
Pmed11	300	5	59	7.69	59	4.50	59	0	59	0.33	59
Pmed13	300	30	36			4.59	36	0.01	37	1.20	36
Pmed15	300	100	18			4.52	18	0.04	18	3.69	18
Pmed17	400	10	39	10.49	39	6.86	40	0.01	39	0.96	39
Pmed18	400	40	28			6.45	30	0.15	29	1.63	30
Pmed20	400	133	14			7.88	14	0.09	14	4.33	14
Pmed30	600	200	11			9.67	11	0.35	10	4.98	11

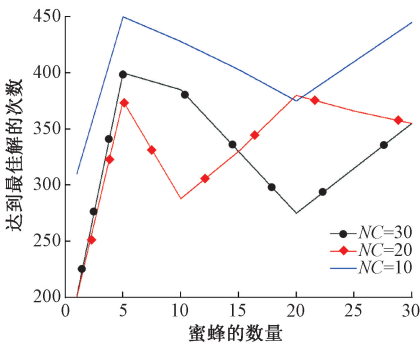


图 3 BCOi 参数对解的影响

Figure 3 Influence of BCOi parameters on the quality of solutions

5.3 近似算法性能对比分析

将近似算法在 OR-Lib 的 Pmed 实例、TSP-Lib 的 d657、nu3496 等实例上解的质量进行对比分析如表 5 所示^[46]。在 OR-Lib 实例中,Gon、Sh 算法的执行时间可忽略不计,而 CDSH 算法的平均执行时间为 0.014 s,但是从表 5 中可以看出,在 Pmed 实例上 CDSH 算法解的质量优于其他算法。在 TSP-Lib 的 d657 等小型实例中,Gon、Sh 算法的执行时间可忽略不计,CDSH 算法总执行时间为 0.13 s,但是 CD-Sh 算法在 d657 实例上解的质量优于 Sh、Gon 算法。在 TSP-Lib 的 nu3496 等大型实例中,Gon、Sh 算法

表 5 近似算法在不同规模下性能对比分析

Table 5 Comparative analysis of performance of approximation algorithm at different scales

数据集	实例	N	K	最优解	最佳值		
					Sh	Gon	CDSH
OR-Lib	Pmed1	100	5	127	167	191	127
	Pmed3	100	10	93	141	142	96
	Pmed7	200	10	64	89	96	64
	Pmed10	200	67	20	28	29	20
	Pmed11	300	5	59	68	91	59
	Pmed13	300	30	36	52	59	37
	Pmed15	300	100	18	24	25	18
	Pmed17	400	10	39	56	59	39
	Pmed18	400	40	28	41	42	29
	Pmed20	400	133	14	18	19	14
	Pmed30	600	200	11	12	14	11
TSP-Lib	d657	657	5	880.90	1 214.30	1352.08	880.90
	d657	657	40	249.51	364.34	333.35	278.35
	u1060	1 060	20	1 580.79	2139.81	2 192.22	1 698.71
	u1060	1 060	100	570.00	828.09	720.92	632.88
	u1060	1 060	150	447.00	699.52	570.01	495.01
	u1979	1 979	10	1 160.69	1 327.48	1 606.55	1 160.69
	u1979	1 979	100	220.32	320.15	295.25	235.08
	u1979	1 979	150	152.02	212.52	197.22	164.56
	nu3496	3 496	20	519.08	703.36	691.64	566.66
	nu3496	3 496	100	194.36	247.77	254.95	222.36
	nu3496	3 496	150	149.07	190.02	186.34	169.96

的平均执行时间分别为 0.001、0.034 s, CDSH 算法的执行时间为 0.314 s, 但是从表 5 中可以看出, 在解的质量上 CDSH 算法优于其他算法。

6 结语

本文对比分析求解 k -center 问题的精确算法、启发式算法、元启发式算法、近似算法等, 总结了各类算法的优缺点及改进思路, 从算法的执行时间、解的质量等方面进行综述。

指出精确算法在小规模的 k -center 问题上有着很好的效果, 但随着规模增大, 变得不再实用。因此未来工作中需要提出更好的启发式产生更紧的上界, 进一步加快寻找最优解的过程。由于启发式算法和元启发式算法没有理论保证, 解的质量难以保证, 因此需要在求解规模、求解效率等方面进行深入研究, 找到更高效的求解算法。近似算法求解 k -center 问题时理论研究价值大于实用价值, 因此未来工作中需要找到不仅理论最优且实用价值更大的算法, 需要继续对 k -center 问题进行深入研究, 找到更加高效的算法求解该问题。

参考文献:

- [1] KARIV O, HAKIMI S L. An algorithmic approach to network location problems. I: the p -centers[J]. SIAM Journal on Applied Mathematics, 1979, 37(3): 513-538.
- [2] DASKIN M. Network and discrete location: models, algorithms and applications[J]. Journal of the Operational Research Society, 1997, 48(7): 763-764.
- [3] ZHANG F Z, HE Y C, OUYANG H B, et al. A fast and efficient discrete evolutionary algorithm for the uncapacitated facility location problem[J]. Expert Systems with Applications, 2023, 213: 118978.
- [4] KAVEH A, NASR H. Solving the conditional and unconditional-center problem with modified harmony search: a real case study[J]. Scientia Iranica, 2011, 18(4): 867-877.
- [5] CONTARDO C, IORI M, KRAMER R. A scalable exact algorithm for the vertex p -center problem[J]. Computers & Operations Research, 2019, 103: 211-220.
- [6] GUI Z P, SUN Y Z, YANG L, et al. LSI-LSTM: an attention-aware LSTM for real-time driving destination prediction by considering location semantics and location importance of trajectory points[J]. Neurocomputing, 2021, 440: 72-88.
- [7] WANG S H, GAO S, FENG X, et al. A context-based geoprocessing framework for optimizing meetup location of multiple moving objects along road networks[J]. International Journal of Geographical Information Science, 2018, 32(7): 1368-1390.
- [8] ZHOU L, WANG S H, XU Z B. A multi-factor spatial optimization approach for emergency medical facilities in Beijing[J]. ISPRS International Journal of Geo-Information, 2020, 9(6): 361.
- [9] ZHU Y S, DU Q Y, TIAN F, et al. Location optimization using a hierarchical location-allocation model for trauma centers in Shenzhen, China[J]. ISPRS International Journal of Geo-Information, 2016, 5(10): 190.
- [10] HAN B, HU M X, ZHENG J M, et al. Site selection of fire stations in large cities based on actual spatiotemporal demands: a case study of Nanjing city[J]. ISPRS International Journal of Geo-Information, 2021, 10(8): 542.
- [11] FARAHANI R Z, HEKMATFAR M. Facility location: concepts, models, algorithms and case studies[M]. Heidelberg: Physica-Verlag HD, 2009.
- [12] HOCHBAUM D S, SHMOYS D B. A best possible heuristic for the k -center problem[J]. Mathematics of Operations Research, 1985, 10(2): 180-184.
- [13] DYER M E, FRIEZE A M. A simple heuristic for the p -centre problem[J]. Operations Research Letters, 1985, 3(6): 285-288.
- [14] PANIGRAHY R, VISHWANATHAN S. An $O(\log^* n)$ approximation algorithm for the asymmetric p -center problem[J]. Journal of Algorithms, 1998, 27(2): 259-268.
- [15] KHULLER S, SUSSMANN Y J. The capacitated K -center problem[J]. SIAM Journal on Discrete Mathematics, 2000, 13(3): 403-418.
- [16] BRASS P, KNAUER C, NA H S, et al. The aligned k -center problem[J]. International Journal of Computational Geometry & Applications, 2011, 21(2): 157-178.
- [17] KHULLER S, PLESS R, SUSSMANN Y J. Fault tolerant K -center problems[J]. Theoretical Computer Science, 2000, 242(1/2): 237-245.
- [18] LIM A, RODRIGUES B, WANG F, et al. k -center problems with minimum coverage[J]. Theoretical Computer Science, 2005, 332(1/2/3): 1-17.
- [19] XU Y, PENG J G, XU Y F. The mixed center location problem[J]. Journal of Combinatorial Optimization, 2018, 36(4): 1128-1144.
- [20] MINIEKA E. The m -center problem[J]. SIAM Review, 1970, 12(1): 138-139.
- [21] DASKIN M S. A new approach to solving the vertex p -center problem to optimality: algorithm and computational results[J]. Communications of the Operations Research Society of Japan, 2000, 45(9): 428-436.
- [22] ILHAN T, PINAR M C. An efficient exact algorithm for the vertex p -center problem[EB/OL]. (2001-09-27) [2023-12-30]. <https://optimization-online.org/2001/>

- 09/376/.
- [23] AL-KHEDHAIRI A, SALHI S. Enhancements to two exact algorithms for solving the vertex P -center problem [J]. *Journal of Mathematical Modelling and Algorithms*, 2005, 4(2): 129–147.
 - [24] MLADENOVIC N, LABBÉ M, HANSEN P. Solving the p -center problem with tabu search and variable neighborhood search[J]. *Networks*, 2003, 42(1): 48–64.
 - [25] PACHECO J A, CASADO S. Solving two location models with few facilities by using a hybrid heuristic: a real health resources case[J]. *Computers & Operations Research*, 2005, 32(12): 3075–3091.
 - [26] GLOVER F. Future paths for integer programming and links to artificial intelligence[J]. *Computers & Operations Research*, 1986, 13(5): 533–549.
 - [27] GLOVER F, LAGUNA M. *Tabu search*[M]. New York: Springer Science+Business Media, 1997.
 - [28] LAGUNA M, MARTÍ R. Scatter search: methodology and implementations in C[J]. *Interfaces*, 2006, 36(6): 610–612.
 - [29] 蒋建林, 徐进澎, 文杰. 基于单亲遗传模拟退火算法的顶点 p -中心问题[J]. *系统工程学报*, 2011, 26(3): 414–420.
JIANG J L, XU J P, WEN J. Solving the vertex p -center problem with a partheno-genetic simulated annealing algorithm[J]. *Journal of Systems Engineering*, 2011, 26(3): 414–420.
 - [30] DAVIDOVIĆ T, RAMLJAK D, ŠELMIĆ M, et al. Bee colony optimization for the p -center problem[J]. *Computers & Operations Research*, 2011, 38(10): 1367–1376.
 - [31] YURTKURAN A, EMEL E. A modified artificial bee colony algorithm for p -center problems[J]. *The Scientific World Journal*, 2014, 2014: 824196.
 - [32] 包敏泽, 胡秀婷, 谢玉莹, 等. 基于人工蜂群算法的 p -center 问题求解算法[J]. *计算机工程与科学*, 2020, 42(6): 1127–1133.
BAO M Z, HU X T, XIE Y Y, et al. A p -center problem solving algorithm based on artificial bee colony algorithm[J]. *Computer Engineering and Science*, 2020, 42(6): 1127–1133.
 - [33] LUČIĆ P, TEODOROVIĆ D. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence[C]//The TRISTAN IV Triennial Symposium on Transportation Analysis. Sao Miguel: TRISTAN, 2001: 441–445.
 - [34] LUČIĆ P, TEODOROVIĆ D. Transportation modeling: an artificial life approach[C]//14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002). Piscataway: IEEE, 2003: 216–223.
 - [35] LUČIĆ P, TEODOROVIĆ D. Computing with bees: attacking complex transportation engineering problems[J]. *International Journal on Artificial Intelligence Tools*, 2003, 12(3): 375–394.
 - [36] 王守娜, 刘弘, 高开周. 一种应用于函数优化问题的多种群人工蜂群算法[J]. *郑州大学学报(工学版)*, 2018, 39(6): 30–35.
WANG S N, LIU H, GAO K Z. A multi-swarm artificial bee colony algorithm for function optimization[J]. *Journal of Zhengzhou University (Engineering Science)*, 2018, 39(6): 30–35.
 - [37] 何尧, 刘建华, 杨荣华. 人工蜂群算法研究综述[J]. *计算机应用研究*, 2018, 35(5): 1281–1286.
HE Y, LIU J H, YANG R H. Survey on artificial bee colony algorithm[J]. *Application Research of Computers*, 2018, 35(5): 1281–1286.
 - [38] RANA R, GARG D. The analytical study of k -center problem solving techniques[J]. *International Journal of Information Technology and Knowledge Management*, 2008, 1(2): 527–535.
 - [39] SHMOYS D B. Computing near-optimal solutions to combinatorial optimization problems[M]//DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Providence: American Mathematical Society, 1995: 355–397.
 - [40] PLESNÍK J. A heuristic for the p -center problems in graphs[J]. *Discrete Applied Mathematics*, 1987, 17(3): 263–268.
 - [41] GONZALEZ T F. Clustering to minimize the maximum intercluster distance[J]. *Theoretical Computer Science*, 1985, 38: 293–306.
 - [42] GARCIA-DIAZ J, SANCHEZ-HERNANDEZ J, MENCHACA-MENDEZ R, et al. When a worse approximation factor gives better performance: a 3-approximation algorithm for the vertex k -center problem[J]. *Journal of Heuristics*, 2017, 23(5): 349–366.
 - [43] MIHELIĆ J, ROBIĆ B. Solving the k -center problem efficiently with a dominating set algorithm[J]. *Journal of Computing and Information Technology*, 2005, 13(3): 225–233.
 - [44] BEASLEY J E. OR-library: distributing test problems by electronic mail[J]. *Journal of the Operational Research Society*, 1990, 41(11): 1069–1072.
 - [45] REINELT G. TSPLIB—A traveling salesman problem library[J]. *ORSA Journal on Computing*, 1991, 3(4): 376–384.
 - [46] GARCIA-DIAZ J, MENCHACA-MENDEZ R, MENCHACA-MENDEZ R, et al. Approximation algorithms for the vertex k -center problem: survey and experimental evaluation[J]. *IEEE Access*, 2019, 7: 109228–109245.

Performance Optimization of Air-cooled Lithium Battery Pack Thermal Management System with Turbulence Structure

HE Chuang, ZHAO Qinxin, LIANG Zhiyuan

(MOE Key Laboratory of Thermo-Fluid Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: The design and simulation of a vehicle-mounted air-cooled lithium battery pack for new energy vehicles were presented. Finite element simulation was employed to analyze the thermal management system of the air-cooled lithium battery pack. The results indicated that with an inlet air velocity of 2 m/s, the maximum temperature differences for the parallel and serpentine structures were 6.96 °C and 6.29 °C, respectively, with a reduction in the maximum temperature difference for the serpentine structure. Introducing flow baffles to create turbulent structures reduced the maximum temperature difference to 5.69 °C compared to the serpentine structure, representing a decrease of 0.60 °C in the maximum temperature difference. When the inlet air velocity exceeded 4 m/s, the maximum temperature difference was below 5 °C, meeting the optimal discharge efficiency of lithium batteries. The battery pack exhibited optimal cooling performance when the flow baffles were arranged in a symmetric structure, with significantly lower maximum temperature and temperature differences compared to other layout arrangements. Furthermore, the inlet air velocity was positively correlated with the cooling performance of the battery pack. The optimal cooling effect was achieved when the inlet air velocity reached 6 m/s, after which the change in cooling performance diminished with further increases in inlet air velocity. In conclusion, the designed 18650 lithium battery pack demonstrated optimal flow disturbance cooling performance with symmetric layout spacing, and controlling the inlet air velocity above 4 m/s enabled the battery to operate at optimal discharge efficiency.

Keywords: lithium battery; turbulent structure; thermal management; inlet air speed; temperature difference

(上接第 50 页)

The Survey of Algorithms for *k*-center Problem

WANG Xiaofeng^{1,2}, HUA Yingying¹, WANG Junxia¹, PENG Qingyuan¹, HE Fei¹

(1. School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China; 2. The Key Laboratory of Images and Graphics Intelligent Processing of State Ethnic Affairs Commission, North Minzu University, Yinchuan 750021, China)

Abstract: The *k*-center problem is fundamental in facility siting, and is also an NP-hard problem with practical applications in the fields of distribution, emergency services, etc. However, with the expansion of the problem scale, the original algorithms were no longer applicable and should be further optimized or improved. In order to find an efficient algorithm to solve the problem, the existing algorithms were studied. The algorithms for the *k*-center problem were classified into exact algorithms, heuristic algorithms, meta-heuristic algorithms, approximation algorithms, etc. The algorithms were compared in terms of their principles, ideas for improvement, performance, accuracy, etc. The exact algorithm obtained the optimal solution in polynomial time when solving small-scale *k*-center problems, but was inefficient and not applicable to large-scale problems. The heuristic algorithm could give the relative optimal solution in polynomial time, but there was no theoretical guarantee to measure the relationship with the most optimal solution. The meta-heuristic algorithm could be improved according to the existing intelligent optimization algorithms to give the relative optimal solution, but the quality of the solution could not be guaranteed. The solution obtained by using the approximation algorithm had the guarantee of the approximation ratio, which was of greater theoretical research value, but the practical value was weaker. At present, the meta-heuristic algorithm for solving the *k*-center problem achieved certain research results, but there were still breakthroughs in the solution time, solution scale and algorithm efficiency, which could be the focus of the future research on the *k*-center problem.

Keywords: *k*-center problem; exact algorithm; approximation algorithm; bee colony optimization algorithm; genetic algorithm