引用格式：余修武, 周颖, 尹文辉, 等. 融合 VFM 与 D*算法的 WSN 动态栅栏覆盖优化 [J]. 电子科技大学学报, 2026, 55(3): 385-397.
YU X W, ZHOU Y, YIN W H, et al. WSN dynamic barrier coverage optimization integrating VFM and D* algorithm[J]. Journal of University of Electronic Science and Technology of China, 2026, 55(3): 385-397.

融合 VFM 与 D*算法的 WSN 动态栅栏覆盖优化

余修武，周 颖*，尹文辉，叶 莱

（南华大学资源环境与安全工程学院，衡阳 421001）

摘要：为了解决无线传感器网络（WSN）的覆盖优化问题，提出了一种基于动态参数调控和虚拟力增强的动态栅栏覆盖 PSO 算法（DPVF-PSO）。首先，初始化网络参数和粒子群参数，随机生成初始群体的位置和速度，并计算每个粒子的适应度值。随后，在迭代过程中动态调整 PSO 参数，计算节点间的虚拟力，选择目标栅栏区域，并融合 D*算法寻找最短路径，更新粒子的速度和位置，同时进行适应度评估。每隔若干代进行一次局部搜索，以进一步优化全局最优解。尤其在含障碍物的检测区域内，DPVF-PSO 通过虚拟力机制引导节点避开障碍物，同时利用 D*算法规划最短路径，优化节点的移动轨迹，减少传输延迟和能量消耗，确保网络覆盖的有效性和通信连通性。实验结果表明，在有障碍物的环境中，DPVF-PSO 的覆盖率比 Chaos ABC、IIC-CS、IHPO 和 HHO 算法分别高出 3.623%、5.762%、10.643% 和 4.385%，适用于智能交通管理、环境监测等实际应用场景，具有显著的实用价值。

关键词：无线传感器网络；粒子群优化；D*算法；虚拟力算法；动态参数调整；栅栏覆盖
中图分类号：TP393 **文献标志码：**A DOI: 10.12178/1001-0548.2025023

WSN dynamic barrier coverage optimization integrating VFM and D* algorithm

YU Xiuwu, ZHOU Ying*, YIN Wenhui, and YE Lai

(School of Resource Environment and Safety Engineering, University of South China, Hengyang 421001, China)

Abstract: To solve the coverage optimization problem of wireless sensor networks (WSN), this paper proposes the dynamic parameter-controlled virtual force enhanced particle swarm optimization (DPVF-PSO) algorithm for region fence coverage. First, the network parameters and particle swarm parameters are initialized, and the initial positions and velocities of the particles are randomly generated. The fitness values of each particle are then calculated. During the iteration process, the PSO parameters are dynamically adjusted, virtual forces between nodes are computed, target fence regions are selected, and the shortest path is determined using D* algorithm. The particles' velocities and positions are updated, and fitness evaluations are performed. Every few generations, a local search is conducted to further optimize the global best solution. In particular, within detection areas containing obstacles, DPVF-PSO uses a virtual force mechanism to guide nodes around obstacles, while also leveraging D* algorithm to plan the shortest path, optimize node movement trajectories, reduce transmission delays, and conserve energy, ensuring the effectiveness of network coverage and communication connectivity. Experimental results show that, in environments with obstacles, the coverage rate of DPVF-PSO exceeds that of the Chaos ABC, IIC-CS, IHPO, and HHO algorithms by 3.623%, 5.762%, 10.643%, and 4.385%, respectively. This makes it highly applicable to practical scenarios such as intelligent traffic management and environmental monitoring, demonstrating significant practical value.

Key words: wireless sensor network (WSN); particle swarm optimization (PSO); D* algorithm; virtual force algorithm; dynamic parameter adjustment; barrier coverage

---

收稿日期：2025−01−20
基金项目：湖南省自然科学基金（2024JJ5338）
作者简介：余修武，博士，教授，主要从事无线传感器网络方面的研究。
*通信作者 E-mail: zhouying20020221@163.com

无线传感器网络（wireless sensor network, WSN）是一种由大量微型传感器节点通过无线通信方式组成的自组织网络系统[1-2]，广泛应用于环境监测、军事侦察、智能交通等领域[3-4]。然而，WSN 节点通常具有通信和感知范围受限、能量有限的特点，在实际应用中，如何在复杂环境下实现高效的覆盖部署成为一个关键挑战[5-6]。尤其是在存在障碍物的环境中，传统算法易陷入局部最优，难以实现高覆盖率与低能耗的平衡[7]。因此，如何有效地提高 WSN 的动态覆盖率[8]，成为当前研究的热点。

为了应对这一挑战，近年来，群体智能算法[9]在 WSN 覆盖优化中表现出显著的效果。如文献 [10] 提出了基于人工蜂群算法的优化方法，通过引入随机因素和随机搜索，能够有效地避免陷入局部极值，但该方法仍存在收敛速度较慢、局部搜索能力不足和节点冗余聚集等问题。文献 [11] 提出了一种改进的猎人猎物优化（improved hunter-prey optimization, IHPO）算法，通过差分进化、自适应 α 变异和动态反向学习平衡节点能耗，虽然提升了能耗均衡性，但在早期收敛、参数敏感性和冗余控制方面仍存在不足。文献 [12] 提出了多策略混合改进的哈里斯鹰算法，通过 Fuch 无限折叠混沌初始化减少了节点冗余，延长了网络寿命，但该算法计算复杂度较高，容易陷入局部最优，全局适应性较差。

为了克服上述问题，本文提出了一种基于动态参数调控和虚拟力增强的动态栅栏覆盖的粒子群优化算法（dynamic parameter-controlled virtual force enhanced particle swarm optimization, DPVF-PSO）。该算法旨在应对复杂环境中的覆盖优化问题，尤其是在具有障碍物和动态环境变化的情况下，表现出显著的节能效果和较高的覆盖率。通过引入虚拟力模型和 D*路径规划，DPVF-PSO 能够灵活地调整节点部署，并动态优化路径，以确保在不同环境下实现高效覆盖和最短路径选择。

1 系统模型与目标函数

1.1 感知覆盖模型

感知覆盖模型用于描述传感器节点在其监测范围内的覆盖能力[13-14]。每个传感器节点拥有 1 个感知半径 R_{sense}，即节点能够有效覆盖的最大范围。

如图 1 所示，红色圆点表示传感器节点的位置，蓝色圆形区域表示该节点的感知范围。感知半径内的所有区域都能被节点监测覆盖，而超出感知半径的区域则无法被节点感知。

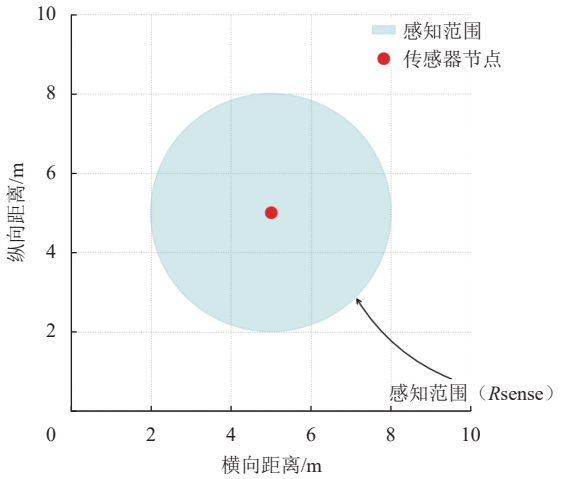


图 1 感知覆盖模型

若节点 S_i 位于位置 (x_i, y_i)，则其感知覆盖区域 A_{sense} 定义为：

$$A_{\text{sense}} = \{(x,y) | \sqrt{(x-x_i)^2+(y-y_i)^2} \leqslant R_{\text{sense}}^2\} \quad (1)$$

即，在节点 S_i 不超过 R_{sense} 的范围内，所有点 (x_i, y_i) 都能被该节点监测覆盖。

1.2 通信覆盖模型

通信覆盖模型保证了传感器节点之间的数据传输链路的畅通性，从而实现监控信息的高效传输[15]。每个传感器节点具有 1 个通信半径 R_{comm}，即节点之间的最大通信距离。如果两个节点间的距离小于其通信半径 R_{comm}，则它们可以直接通信；否则，需要通过中继节点进行数据传输。

如图 2 所示，红色点代表传感器节点的位置，蓝色圆表示每个节点的通信半径。当两个节点的距离不超过通信半径时，它们之间建立通信连接（用绿色连线表示）。

节点间的通信条件为：若节点 S_i 和 S_j 的位置分别为 (x_i, y_i) 和 (x_j, y_j)，则它们之间的距离 $d(i,j)$ 为：

$$d(i,j) = \sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \quad (2)$$

当满足条件 $d(i,j) \leqslant R_{\text{comm}}$ 时，节点 S_i 和 S_j 可以直接通信。当网络中的所有节点之间直接或间接存在路径连接，即任意两个节点间的信息可以通过多跳传输到达汇聚节点或控制中心，则认为该网络是连通的。

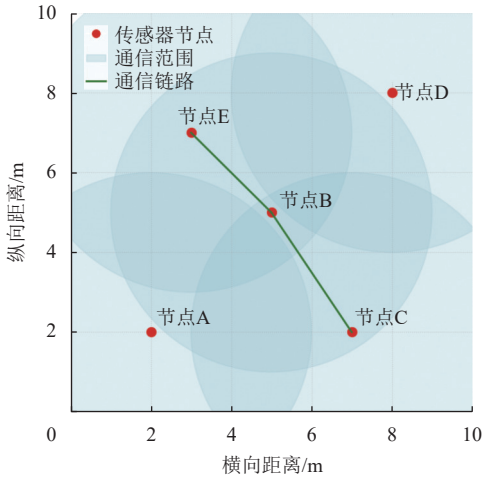


图 2　通信覆盖模型

1.3　区域栅栏覆盖模型

区域栅栏覆盖问题[16-17]是 WSN 中的一种特殊覆盖问题，旨在确保部署在特定区域的传感器节点能够有效监测穿越该区域的目标或事件。与传统区域覆盖不同，栅栏覆盖（barrier coverage）的重点在于在边界或指定路径上形成一条或多条感知“栅栏”，以阻止或检测特定方向上的穿越行为。栅栏覆盖将监测区域分割为若干相同大小的网格，以便管理和计算。

图 3 展示了在区域栅栏覆盖中，如何使用虚拟力模型和 D*算法来优化传感器节点的移动。其中，红点代表传感器节点的初始位置，负责监控并向未覆盖的栅栏单元移动。蓝点表示栅栏中的未覆盖单元，是传感器节点的目标位置。绿点代表中继节点，帮助节点逐步移动到目标位置，保证通信连通性。灰色实线代表虚拟力，灰色箭头表示虚拟力的方向，推动节点向目标栅栏单元移动，以减少监控盲区。蓝色虚线代表 D*路径，展示了通过 D*算法计算的最短路径，确保传感器节点之间保持通信连接。

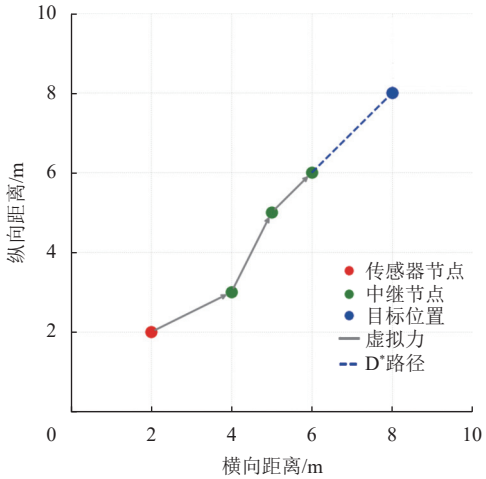


图 3　虚拟力和栅栏覆盖路径图

1.4　目标函数

在 WSN 中，提升网络覆盖效果是核心目标之一。然而，覆盖效果不仅仅依赖于覆盖率的提高，还需要综合考虑冗余度和能量消耗等因素[18-19]。因此，本文在评估算法性能时引入 3 个关键评价指标：覆盖率、冗余度和能量消耗。

1）覆盖率（coverage rate）表示实际被覆盖的区域占总监测区域的百分比，用符号C_{r}表示，计算公式为：

$$C_{\mathrm{r}}=\frac{S_{\mathrm{covered}}}{S_{\mathrm{total}}}\times 100\% \tag{3}$$

式中，S_{covered}是所有节点感知范围的并集面积（不重复计入重叠部分）；S_{total}是整个监控区域的总面积。

2）冗余度（redundancy）用于衡量节点的覆盖冗余情况，体现网络资源的利用效率，用符号C_{e}表示，其计算公式为：

$$C_{\mathrm{e}}=\frac{C_{\mathrm{r}}\times lm}{n\pi R^2} \tag{4}$$

式中，C_{r}为覆盖率；l、m 为检测区域的长和宽；n 为节点个数；R 为通信半径。

3）实际总能量消耗（E_{actual}）表示节点在数据传输、接收和处理过程中消耗的总能量，用符号E_{a}表示，计算公式为：

$$E_{\mathrm{a}}=E_{\mathrm{transmission}}+E_{\mathrm{reception}}+E_{\mathrm{processing}} \tag{5}$$

式中，$E_{\mathrm{transmission}}$为数据传输能耗；$E_{\mathrm{reception}}$为接收能耗；$E_{\mathrm{processing}}$为处理能耗。

特别地，定义能量消耗比值（E_{c}）为实际能耗除以最大能耗，该值越小越好：

$$E_{\mathrm{c}}=\frac{E_{\mathrm{a}}}{E_{\max}} \tag{6}$$

式中，$E_{\max}$是节点的最大能量消耗（理论上的上限），通常是在最极端条件下（最大通信距离、最大感知负荷、最大移动距离）所消耗的能量。

这 3 个指标综合考虑了网络的覆盖效果、资源利用效率和能量消耗情况，是评估 WSN 覆盖优化算法性能的基础。通过对这些指标的分析，可以全面评估算法在不同应用场景中的实际表现。

为了实现 WSN 在覆盖率、冗余度和能量消耗 3 方面的综合优化，本文采用了权重形式的目标函数。目标函数定义如下：

$$\boldsymbol{H}_{\mathrm{total}}=\omega_1 C_{\mathrm{r}}+\omega_2 C_{\mathrm{e}}+\omega_3(1-E_{\mathrm{c}}) \tag{7}$$

权重值ω_1、ω_2、ω_3的选择将直接影响算法的优化方向，其具体取值将在实验部分通过敏感性分析进行讨论。

2 改进的粒子群算法

传统的粒子群优化算法[20]在解决无线传感器网络覆盖问题时具有简单、易实现和全局搜索能力较强的优势，适合处理连续优化问题。然而，其在后期往往容易陷入局部最优，且缺乏足够的局部搜索能力，导致收敛精度不足。以下为改进的 DPVF-PSO 算法的主要步骤。

2.1 栅栏覆盖的网格划分

2.1.1 数学定义

栅栏划分的核心目标是将监测区域有效分解为若干网格单元，以便对每个单元的覆盖状态进行评估和优化。假设监测区域为 1 个边长为 L 的正方形区域，可划分为fenceRows×fenceColumns个均匀网格，每个栅栏单元的面积定义为：

$$\text{fenceSize} = \frac{L}{\text{fenceRows}} \times \frac{L}{\text{fenceColumns}} \tag{8}$$

2.1.2 栅栏单元与节点覆盖的关系

若传感器节点 S_i 的感知范围为 1 个以节点位置为圆心、半径为 R_{sense} 的圆形区域 A_{sense}，则其覆盖的栅栏单元集合 C 可以定义为：

$$C = \{\text{cell}_j | A_{\text{sense}} \cap \text{cell}_j \neq \varnothing\} \tag{9}$$

式中，cell_j表示第 j 个栅栏单元。如果某个栅栏单元未被任何传感器节点覆盖，则称该单元为“盲区”。

2.1.3 均匀划分的优点

1）覆盖状态可控

在均匀划分中，每个栅栏单元的面积相等，覆盖盲区的计算可以简化为：

$$A_{\text{uncovered}} = \sum_{j \in \text{Uncovered}} A_{\text{cell}_j} \tag{10}$$

式中，A_{cell_j}为未覆盖单元的面积。而在非均匀划分方式中，盲区面积难以通过简单加总获得，增加了计算复杂性。

2）优化效率高

在均匀划分中，栅栏单元的中心点位置规则明确，能够快速生成目标点的坐标：

$$(x_{\text{target}}, y_{\text{target}}) = \left(\frac{(i-0.5)L}{\text{fenceRows}}, \frac{(j-0.5)L}{\text{fenceColumns}}\right) \tag{11}$$

这大大减少了盲区检测和路径规划的计算复杂度。假设监测区域包含 n 个栅栏单元，均匀划分的复杂度为 $O(n)$，而非均匀划分需要对每个栅栏单元单独计算边界位置，复杂度为 $O(n\log n)$。因此，均匀划分在大规模场景下更具优势。

2.2 动态参数优化机制

2.2.1 惯性权重的自适应调整

惯性权重 W 是控制粒子速度更新的重要参数，它决定了粒子在下一步中是继续沿用当前的速度方向，还是更多地依赖于个体和群体的经验。在 PSO 中，惯性权重通常会随着迭代次数的变化而动态调整，以实现全局搜索和局部搜索的平衡。本文中，惯性权重 W 根据迭代次数逐渐减小，其调整公式为：

$$W_i = W_{\max} - \frac{(W_{\max} - W_{\min})i}{\text{maxgen}} \tag{12}$$

式中，$W_{\max}$ 和 $W_{\min}$ 分别表示惯性权重的最大值和最小值；i 是当前迭代次数；maxgen 是最大迭代次数。

惯性权重的选择对算法性能有显著影响。较大的惯性权重有助于粒子在搜索空间中进行广泛探索，增加跳出局部最优的可能性；而较小的惯性权重则增强了粒子的局部搜索能力，加快了算法的收敛速度。式 (12) 中，W 随着迭代次数的增加而线性递减，这种线性递减策略被广泛应用于 PSO 算法中，因为它在早期阶段提供较大的搜索范围，后期阶段则聚焦于优化细节。

2.2.2 学习因子 c_1 和 c_2 的自适应调整

学习因子 c_1 和 c_2 决定了粒子在下一步中对个体最优解和群体最优解的依赖程度，分别表示粒子对自身经验（pbest）和对群体经验（gbest）的信任。在传统 PSO 算法中，学习因子通常为定值，但在动态环境（如节点频繁移动的无线传感器网络场景）中，最优解可能随着时间变化，因此固定学习因子会影响算法对变化的适应性。

为提升算法的适应性，本文对学习因子进行动态调整：

$$c_1 = 2 - 1.5\frac{i}{\text{maxgen}} \tag{13}$$

$$c_2 = 0.5 + 1.5\frac{i}{\text{maxgen}} \tag{14}$$

学习因子的动态调整策略基于以下考虑：在初期阶段，较大的 c_1 和较小的 c_2 使粒子更多依赖自

身经验，鼓励探索新的解，增强全局搜索能力。在后期阶段，较小的 c_1 和较大的 c_2 引导粒子依赖群体经验，加快收敛速度，提升局部搜索精度。

2.3　虚拟力模型的集成

虚拟力模型（virtual force model, VFM）通过模拟物理世界中的引力和斥力相互作用，来优化节点的部署和移动，从而提升网络的覆盖率和能量效率[21-22]。在本文提出的 DPVF-PSO 算法中，VFM 通过吸引力和排斥力机制来优化节点的部署。

具体来说，吸引力 $\boldsymbol{F}_{\mathrm{a}}$ 用于引导节点向未覆盖区域或目标区域移动，其大小与节点到目标位置的距离成正比，计算公式为：

$$\boldsymbol{F}_{\mathrm{a}} = k_{\mathrm{a}}(\boldsymbol{P}_t - \boldsymbol{P}_i) \tag{15}$$

式中，k_{a} 为吸引力系数；$\boldsymbol{P}_t$ 为目标位置（如未覆盖区域或目标点）。

排斥力 $\boldsymbol{F}_{\mathrm{r}}$ 则用于避免节点之间过于靠近，防止节点冗余覆盖区域，其大小与节点间距离成反比，其公式为：

$$\boldsymbol{F}_{\mathrm{r}} = \frac{k_{\mathrm{r}}(\boldsymbol{P}_j - \boldsymbol{P}_i)}{d_{ij}^2} \tag{16}$$

式中，k_{r} 是排斥力系数；$\boldsymbol{P}_i$ 和 $\boldsymbol{P}_j$ 分别为节点 i 和节点 j 的位置；d_{ij} 为节点 i 和节点 j 之间的距离。排斥力方向由节点 j 指向节点 i。

当 d_{ij} 小于一定阈值（如通信半径 R）时，节点之间会产生排斥力，迫使节点互相远离，从而避免节点过于集中。

每个节点 i 受到多个节点的排斥力以及 1 个目标点的吸引力，其合力可以表示为：

$$\boldsymbol{F}_i = \sum_{j \in N_i} \boldsymbol{F}_{\mathrm{r}}^{ij} + \boldsymbol{F}_{\mathrm{a}}^{i} \tag{17}$$

式中，N_i 表示与节点 i 相邻的节点集合；合力 $\boldsymbol{F}_i$ 决定了节点下一步移动的方向和幅度。

为了实现节点部署过程中对覆盖性能与收敛速度的兼顾，本文在虚拟力模型中引入动态参数调节机制。具体而言，通过引力系数 ω_{A} 与斥力系数 ω_{R} 的比值动态调整节点间的相互作用强度。初始阶段，设定较高的 ω_{R}，以提升节点间的排斥性，促使其快速扩散覆盖监测区域；而在迭代中后期，则逐步提升 ω_{A} 以增强收敛能力，防止节点过度发散无法聚集至关键区域。

同时，为避免不必要的虚拟力作用，算法设置了感知阈值距离 d_{th}：仅当节点间距离小于此阈值时施加斥力，距离大于阈值则视为互不影响。该阈值依据节点感知半径确定，使得虚拟力仅作用于局部范围，有效降低计算冗余。当所有节点在连续两次迭代中的移动距离均小于预设的微小阈值 ε，即可认为系统已收敛，无需继续迭代。

虚拟力触发机制流程图如图 4 所示。

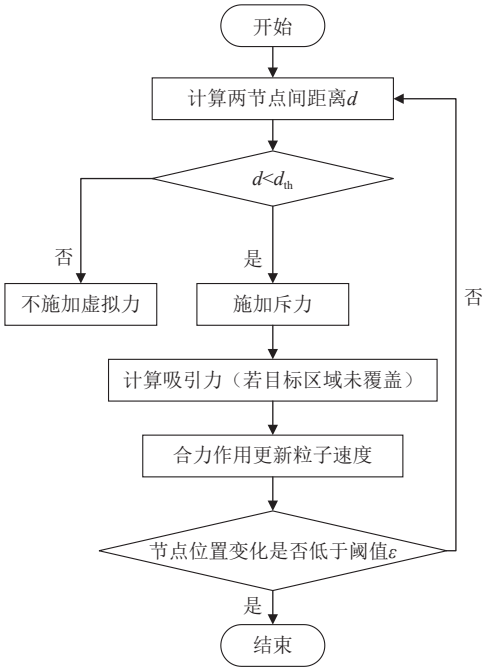


图 4　虚拟力触发机制流程图

2.4　D*算法介绍

D*（dynamic A*）[23] 算法是一种用于路径规划的优化算法，尤其适用于在环境变化的情况下需要实时更新路径的场景。传统的 A*算法在固定环境下非常高效，但在动态或变化的环境中，它会面临重新计算路径的高昂代价。D*通过智能地更新部分路径信息，从而避免了重新计算整个路径，显著提高了效率。

2.4.1　A*算法

A*算法[24] 是一种基于启发式搜索的最短路径算法。它在寻找从起点到目标的最短路径时，结合了路径成本（g 值）和启发式估计（h 值）因素。A*算法的核心思想是通过每个节点的代价（$f(n)$）来评估其最优性，公式如下：

$$f(n) = g(n) + h(n) \tag{18}$$

式中，$g(n)$ 是从起点到当前节点 n 的实际代价（即路径长度）；$h(n)$ 是从当前节点 n 到目标节点的预估代价（启发式函数），通常采用欧几里得距

离、曼哈顿距离等度量方式。A*算法通过优先队列的方式逐步展开搜索，直到找到最短路径或确认无解。

2.4.2 D*算法的引入

D*算法是 A*算法的扩展，专门用于应对动态变化的环境。它最初由 Steven M. LaValle 于 1998 年提出，适用于机器人导航和自动驾驶等领域。它的核心思想在于，计算最短路径时，不是从头开始，而是当环境发生变化时，仅更新受影响的部分路径，从而避免重新计算整个路径。这使得 D*算法比传统的 A*算法更适合处理动态场景。

2.4.3 D*算法的工作原理

1）初始化

首先，D*算法会像 A*算法一样，计算从起点到目标的最短路径。然后，构建 1 个 cost map（代价图），其中包含了每个节点的 g 值和 h 值。

2）环境变化检测

在动态环境中，障碍物或其他因素可能会发生变化。D*算法会监测这些变化，特别是那些会影响当前路径的变化。每当环境发生变化时，D*算法只会更新发生变化的区域，并通过重新规划来调整路径。

3）更新路径

D*算法在计算最短路径时，会利用已计算出的路径信息进行局部修正。算法会对受影响的节点进行动态调整，通过重新评估与目标的路径代价，并根据新的障碍信息来调整路径。

4）局部最优路径更新

与 A*算法不同，D*算法的每次路径更新不仅会考虑到起点到目标的路径，还会考虑到局部优化，即路径的细节更新，而不是完全依赖于启发式函数。

为提升节点在动态环境中的路径规划能力，本文将 D*算法嵌入 DPVF-PSO 框架中，作为节点位置更新阶段的局部路径重构模块。具体而言，D*算法的触发条件包括：节点在移动过程中遇到障碍物阻断原有路径或环境地图出现拓扑变化（如新增障碍或感知区域失效）时，系统即刻调用 D*算法对受影响路径段进行局部增量式更新。

相较于传统 A*算法需全局重算路径的方式，D*通过启发式增量重规划机制，仅对环境变化影响的子路径进行调整，从而显著降低了计算开销，提升了适应性与实时性。该机制特别适用于资源受限的无线传感器网络，能够在动态多变的环境中持续提供最优或近似最优的路径方案，增强了系统的稳健性与部署效率。

路径重构触发机制流程图如图 5 所示。

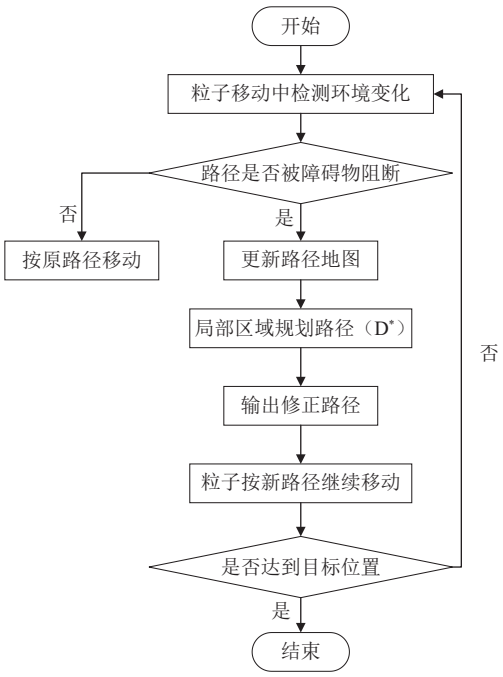


图 5 D*算法路径更新触发机制流程图

2.5 障碍物对 WSN 覆盖的影响与模型构建

在实际应用中，无线传感器网络经常部署在存在多种障碍物的复杂环境中。障碍物会对 WSN 的覆盖性能产生如信号衰减与阻断、覆盖盲区增加、覆盖冗余与能耗增加、节点部署和优化策略更为复杂等显著影响。

为简化分析并突出障碍物的影响，本研究中仅考虑两类典型几何形状的障碍物：正方形与等腰直角三角形。这两种障碍物具有明确的几何特征和简单的参数化表示方式，便于数学建模与仿真实现。

1）正方形障碍物

正方形障碍物常用于模拟建筑物、墙体或其他规则人工构造所形成的遮挡。数学模型如下。

假设正方形障碍物的中心坐标为(x_s,y_s)，边长为L_s，则其 4 个顶点坐标为：$(x_s-\frac{L_s}{2}, y_s-\frac{L_s}{2})$、$(x_s-\frac{L_s}{2}, y_s+\frac{L_s}{2})$、$(x_s+\frac{L_s}{2}, y_s+\frac{L_s}{2})$、$(x_s+\frac{L_s}{2}, y_s-\frac{L_s}{2})$，在实际计算中，可以利用多边形判定方法（如射线法或多边形点判断算法）来确定某一点是否在正方形障碍物内部，从而判断该点与传感器间的信号是否被阻断。

2）等腰直角三角形障碍物

等腰直角三角形障碍物可代表斜坡、拐角或某些不规则但仍相对简单的自然或人工场景。数学模型如下。

假设等腰直角三角形的直角顶点坐标为(x_t, y_t)，直角边长为 L_t，则 3 个顶点坐标为：(x_t, y_t)、(x_t+L_t, y_t)、(x_t, y_t+L_t)，同样，通过多边形点判断算法可确定任一点是否位于三角形障碍物内部。

在计算覆盖时，可将传感器节点与目标点（或感知网格点）间的连线与障碍物的多边形区域进行检测，若连线与障碍物内部区域重叠，即判定该点的覆盖受到阻断，从而无效。

2.6　算法复杂度分析

本文提出的 DPVF-PSO 算法结合了 PSO 算法和 D*路径规划算法，优化了节点部署和路径规划。下面对该算法的整体复杂度进行分析。

时间复杂度主要来自粒子更新的计算，假设有 n 个粒子和 m 个维度，每次粒子更新的计算复杂度为 $O(mn)$。在每次粒子更新后，D*算法需要对路径进行更新，其复杂度为 $O(N\log N)$，其中 N 为路径图中的节点数量。因此，DPVF-PSO 的总时间复杂度为 $O(mn + N\log N)$，其中 $O(mn)$ 表示 PSO 的计算复杂度，$O(N\log N)$ 表示 D*算法的路径规划复杂度。

空间复杂度由 PSO 粒子的存储和 D*算法的路径图存储决定。PSO 需要存储每个粒子的当前位置、速度和最优位置，因此空间复杂度为 $O(n)$，而 D*算法则需要存储路径图和节点状态，空间复杂度同样为 $O(n)$。因此，DPVF-PSO 的总空间复杂度为 $O(n)$。

通过这种设计，DPVF-PSO 算法在时间和空间复杂度上保持合理。与传统的 D*算法或 PSO 算法相比，DPVF-PSO 算法更高效地结合了路径规划和动态参数调整，减少了计算负担。其低内存消耗使其适合资源受限的无线传感器网络，并能灵活应对动态环境和障碍物，展现出优越的适应性。

2.7　算法实施流程

本文提出的 DPVF-PSO 算法在惯性权重动态调整、未覆盖区域选择、虚拟力引导和局部搜索优化方面增强了传统 PSO 算法的性能。DPVF-PSO 算法的实施流程如图 6 所示。

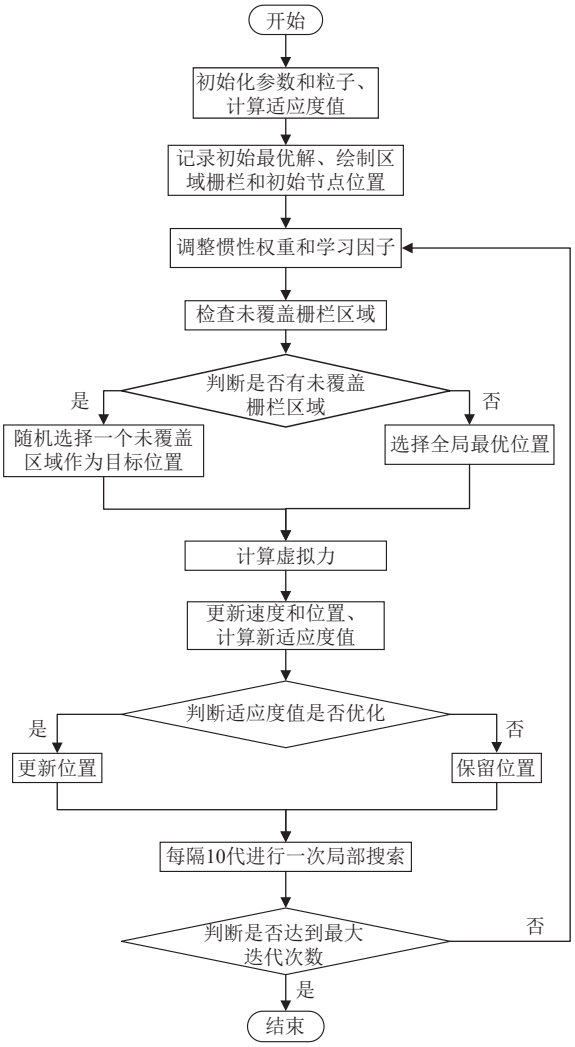


图 6　DPVF-PSO 算法流程

3　仿真实验与分析

在实验设计中，引入不同参数配置的敏感性分析，以探讨 DPVF-PSO 算法在不同场景下的适用性。实验在 MATLAB R2021b 环境中进行，设置不同的迭代次数以及节点数量，观察算法在不同配置下的表现。同时，与传统 PSO 算法和其他优化算法进行对比，主要评价指标为覆盖率、冗余度和能量消耗。结果显示，DPVF-PSO 算法在多种配置下均表现出较高的适应性，特别是在较高节点密度和复杂障碍物环境中，DPVF-PSO 算法能够更好地保持低能耗与高覆盖率的平衡。

3.1　目标函数性能验证与参数敏感性分析

3.1.1　实验设计

为了确定动态参数选择依据，验证目标函数在不同权重组合下的优化能力以及探讨低节点数量对算法性能的影响，本文设计了以下实验。

1）惯性权重敏感性分析

为了选择合适的 W_{max} 和 W_{min}，本文进行了参数敏感性分析。通过实验比较不同 W_{max} 和 W_{min} 组合下算法在覆盖率、冗余度和能耗方面的表现。

2）学习因子敏感性分析

为了确定动态调整策略的有效性，本文进行了不同的学习因子调整策略，包括动态调整策略和固定依赖策略。

3）目标函数权重分析

固定节点数为 40，设置覆盖率、冗余度和能耗的权重组合$\omega_1:\omega_2:\omega_3$为不同比例（如 0.6∶0.2∶0.2，0.4∶0.4∶0.2，0.3∶0.3∶0.4），假定最大能耗为 5 000 J。对比综合目标函数值 F_{total} 的变化，考察权重对覆盖率、冗余度和能量消耗的具体影响。

4）低节点数量的性能分析

设置节点数量为 20、30 和 40，假定对应的最大能耗分别为 2 000、2 500 和 3 000 J，用于模拟稀疏覆盖场景。$\omega_1:\omega_2:\omega_3$取 0.6∶0.2∶0.2 的权重，比较目标函数的优化效果，以验证低节点数量是否会导致某些指标表现不佳。

3.1.2 实验参数

实验参数如表 1 所示。

表 1 实验参数

参数	值
区域边长/m	50
通信半径/m	5
感知半径/m	7
粒子规模/个	20
最大迭代次数	500
障碍物数量	无/少量随机分布

3.1.3 实验结果与分析

图 7 和图 8 展示了在不同惯性权重和学习因子组合下，DPVF-PSO 算法的性能表现。实验通过对这些参数进行敏感性分析，探讨了它们对算法优化效果的影响。具体来说，本文通过设置多个惯性权重和学习因子的组合，运行 DPVF-PSO 算法，记录其在不同参数配置下的性能表现（如覆盖率和能效）。

图 9 和图 10 展示了目标函数权重设置对网络覆盖率、冗余度和能量消耗的影响。实验通过设置不同的目标函数权重组合，并在多个不同场景下进行评估，分析了权重调整对算法优化的影响。这两幅图通过对比不同权重配置下的结果，展示了在不同目标下优化效果的变化。

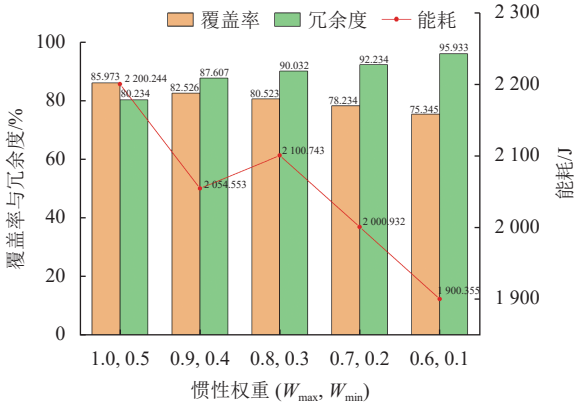


图 7 不同惯性权重对比

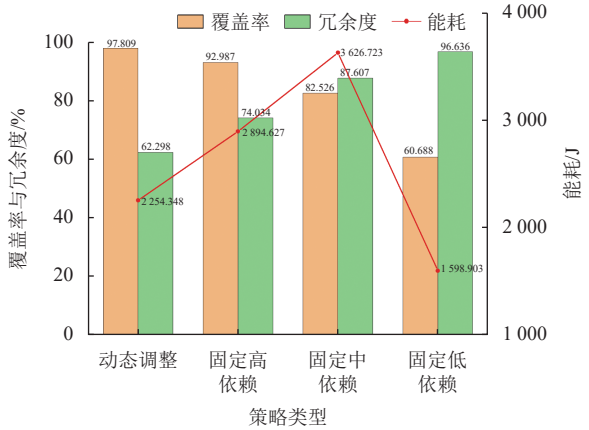


图 8 不同调整策略对比

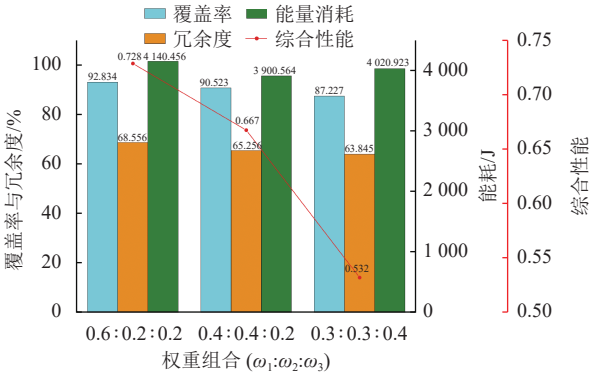


图 9 目标函数权重敏感性对比

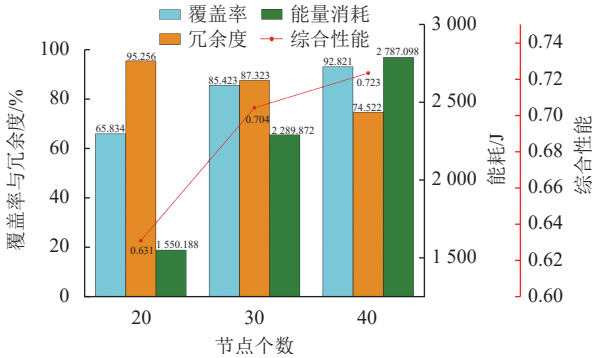


图 10 低节点数量分析对比

1）惯性权重敏感性分析结果

由图 7 可知，随着 W_{max} 和 W_{min} 的递减，覆盖率和冗余度呈现下降趋势，而能耗也相应减少。这表明较大的惯性权重有助于提高覆盖率和冗余度，但同时也会增加能耗。组合 2（W_{max}=0.9, W_{min}=0.4）在覆盖率和冗余度之间取得了较好的平衡，同时能耗控制在合理范围内，是本研究选定的最优惯性权重组合。

2）学习因子敏感性分析结果

图 8 显示，动态调整 c_1 和 c_2 相较于固定学习因子的策略，能有效平衡全局搜索和局部搜索，提升覆盖率和减少冗余，同时能耗也控制在合理范围内，是本研究选定的最优调整策略。

3）目标函数权重分析结果

从图 9 中可以看出，当覆盖率的权重ω_1较大时（如 0.6∶0.2∶0.2），覆盖率优先优化，但冗余度略高。能耗权重ω_3较大时（如 0.3∶0.3∶0.4），能耗显著降低，但覆盖率下降，可见目标函数权重需在实际场景中权衡选择。

4）低节点数量的性能分析结果

从图 10 中可以观察到，在节点数量较低时（如 20 个节点），覆盖率显著降低，冗余度升高，表明网络资源不足对 DPVF-PSO 算法的综合性能的负面影响。随着节点数量增加，综合性能逐渐提高，尤其在 40 个节点时达到最佳值。

3.2　DPVF-PSO 算法对比仿真

为验证对传统 PSO 算法改进策略的有效性和优越性，本文对比了传统 PSO 算法、加入动态参数调整的 PSO 算法、加入局部搜索的 PSO 算法、加入虚拟力的 PSO 算法、融合栅栏覆盖和 D*算法的 PSO 算法，以及融合了上述动态参数调整、局部搜索、虚拟力、栅栏覆盖与 D*算法这 4 种方法的 DPVF-PSO 算法。其实验参数如表 2 所示。

表 2　DPVF-PSO 实验参数

实验参数	取值
区域边长/m	50
节点个数/个	35
通信半径/m	5
感知半径/m	7
离散粒度	1
栅栏行数、列数	5、5
粒子规模/个	20
惯性权重最大、最小值	0.9、0.4
最大迭代次数/次	500
$\omega_1:\omega_2:\omega_3$	0.6∶0.2∶0.2
速度最大、最小值/m·s^{-1}	2、−2
位置最大、最小值/m	50、0

由图 11 可见，DPVF-PSO 算法在性能上明显优于传统 PSO 算法和其他增强版 PSO 算法。传统 PSO 算法虽然表现稳定，但最终覆盖率仅约为 80%，说明其局部搜索能力较弱，容易陷入局部最优。加入动态参数调整的 PSO 算法和加入局部搜索的 PSO 算法在初期有一定提升，但最终覆盖率都未超过 85%。加入虚拟力的 PSO 算法在初期能快速将覆盖率提升至 80% 左右，但后期优化不足，最终覆盖率也未超过 85%。加入栅栏覆盖和 D*算法的 PSO 算法收敛速度较慢，适应性较差，覆盖率提升有限。

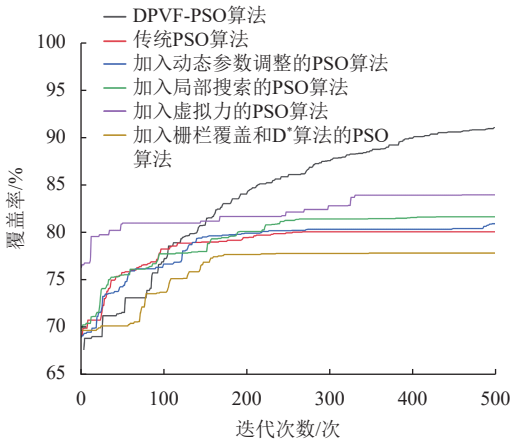


图 11　DPVF-PSO 算法仿真对比图

相比之下，DPVF-PSO 从迭代初期就表现出稳定的寻优性能，覆盖率在迭代过程中持续提升，在 150 次迭代左右时，覆盖率就已超过其他所有算法，在 200 次迭代后覆盖率接近 85%，500 次迭代后超过 90%。相比其他对比算法易陷入局部最优的缺陷，该算法展现出更强的持续寻优能力与全局覆盖性能，体现了其在复杂环境中的鲁棒性和适应性。

3.3　DPVF-PSO 算法与其他优化算法对比（无障碍物）

为了验证改进的粒子群算法在无线传感器网络覆盖优化中的性能，本文将 DPVF-PSO 算法与混沌交叉人工蜂群算法（Chaos ABC）[10]、改进布谷鸟搜索算法（IIC-CS）[25]、改进猎人猎物优化算法（IPHO）[11]和多策略混合改进哈里斯鹰算法（HHO）[12] 进行对比实验。在50 m×50 m的监测区域内，分别随机部署 20、30、40、50 个传感器节点，感知半径为 5 m，迭代次数为 500 次，粒子数量为 20 个。

表 3 展示了在无障碍物环境下 5 种算法在覆盖率、冗余度和能量消耗 3 项性能指标下的实验结果，数据来自不同算法在设定实验场景下的对比实验，实验参数详见表 2。DPVF-PSO 在 3 项指标中均取得最优表现，不仅在覆盖率上明显高于其他算

法，还有效控制了冗余节点数量，并保持最低的能量消耗，体现出其在多目标优化中的综合优势。

表 3　5 种算法的性能指标数据

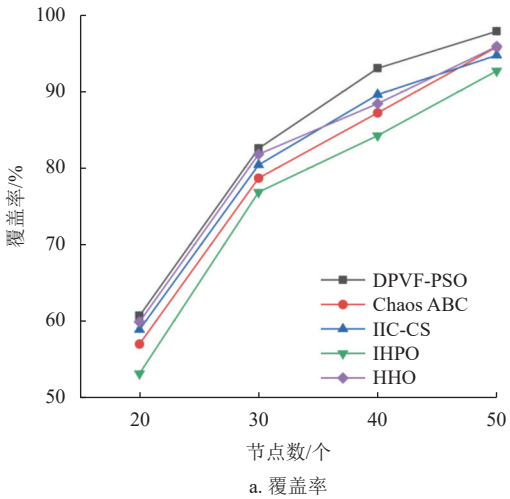
算法	$n=20$			$n=30$			$n=40$			$n=50$		
	C_r/%	C_e/%	E_a/J	C_r/%	C_e/%	E_a/J	C_r/%	C_e/%	E_a/J	C_r/%	C_e/%	E_a/J
DPVF-PSO	60.688	96.636	1 598.903	82.526	87.607	2 254.348	92.987	74.034	2 894.627	97.809	62.298	3 626.723
ChaosABC	56.988	90.745	1 926.771	78.657	83.500	2 592.651	87.167	69.400	3 418.725	95.764	60.996	4 204.725
IIC-CS	58.875	93.750	2 073.822	80.374	85.322	2 715.824	89.543	71.292	3 562.332	94.674	60.301	4 422.615
IHPO	53.136	84.611	2 132.234	76.824	81.554	2 883.115	84.198	67.036	3 596.826	92.644	59.008	4 512.635
HHO	59.876	95.343	1 826.825	81.764	86.798	2 572.672	88.366	70.355	3 291.651	95.826	61.035	4 029.634

图 12a～图 12c 分别为表 3 中不同节点数在覆盖率（C_r）、冗余度（C_e）和能量消耗（E_a）下的可视化，展示了 5 种算法在无障碍物环境中的覆盖性能对比。

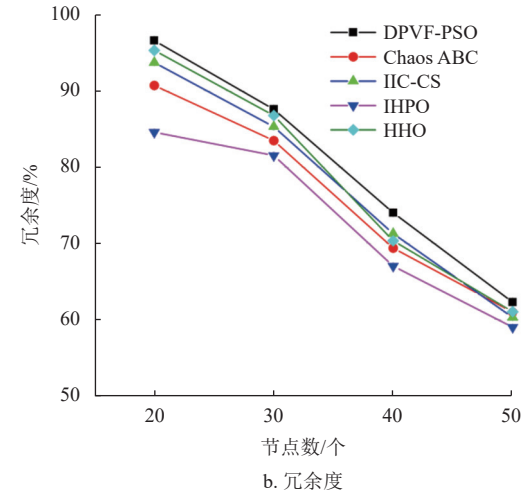
图 12a 显示，与 Chaos ABC、IIC-CS、IHPO 和 HHO 算法相比，DPVF-PSO 算法在少量节点（20 个）时即能实现较高的覆盖率（约 61%），随着节点数的增加，DPVF-PSO 算法的覆盖率迅速提高，在 50 个节点时达到近 97%。其优势在于虚拟力模型有效填补了监控盲区，并通过动态参数调控保持了冗余度与覆盖率的平衡。

图 12b 表明，所有算法的冗余度随着节点数从 20 增加到 50 而逐渐降低，这表明随着节点数量增加，覆盖范围提升的同时，冗余度逐步减少，资源利用更加高效。虽然 DPVF-PSO 算法的冗余度始终较高，但由于引入了虚拟力模型、动态参数调控和最短路径规划，其冗余度的下降速度最快，从 20 个节点到 50 个节点下降了 34.338%。其他算法的冗余度下降分别为 29.749%、33.448%、25.602% 和 34.308%。

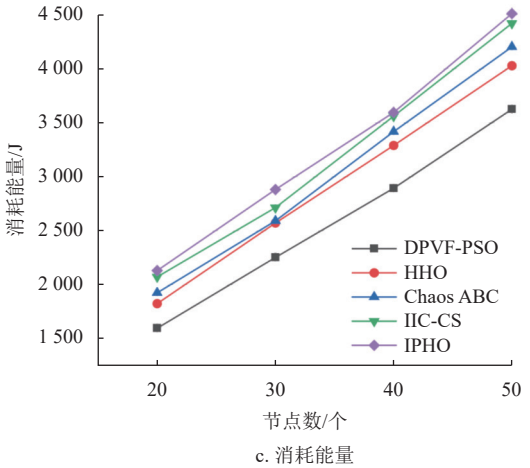
从图 12c 可以看出，DPVF-PSO 算法通过虚拟力机制、动态参数调控和路径优化，在不同节点数量下始终保持最低的能耗。与其他算法相比，DPVF-PSO 算法在扩展性、能效管理和障碍物适应性方面表现优越，特别适合大规模节点网络。



a. 覆盖率



b. 冗余度



c. 消耗能量

图 12　5 种算法在无障碍区域的覆盖性能指标比较

图 13 展示了在 50 m×50 m 范围内随机部署 35 个传感器节点，在其他参数与表 2 保持一致的条件下，采用不同优化算法后的节点部署和覆盖情况。可以明显看出，DPVF-PSO 算法的节点部署最为合理，覆盖面积最广。与 Chaos ABC、IIC-CS、IPHO 和 HHO 算法相比，DPVF-PSO 算法的覆盖率分别提高了 7.827%、8.764%、15.883% 和 5.365%。

在无障碍环境下，DPVF-PSO 算法与 Chaos ABC 算法相比，能够在节点数量较少时就实现较高的覆盖率，且冗余度下降速度更快，从而在提高覆盖率的同时减少冗余节点。与 IIC-CS 算法相

比，DPVF-PSO 算法通过虚拟力模型和动态参数调控的结合，优化了节点部署，并避免了局部最优问题，在覆盖率和能效方面展现了更高的效率。与 IHPO 算法相比，DPVF-PSO 算法不仅在覆盖率提升上表现优异，而且其动态参数调控和虚拟力机制使得其在无障碍环境中能够更加高效地优化节点部署和路径规划。与 HHO 算法相比，DPVF-PSO 算法在减少节点冗余的同时，始终保持较低的能耗，并且在扩展性和能效管理方面表现出更优的效果。

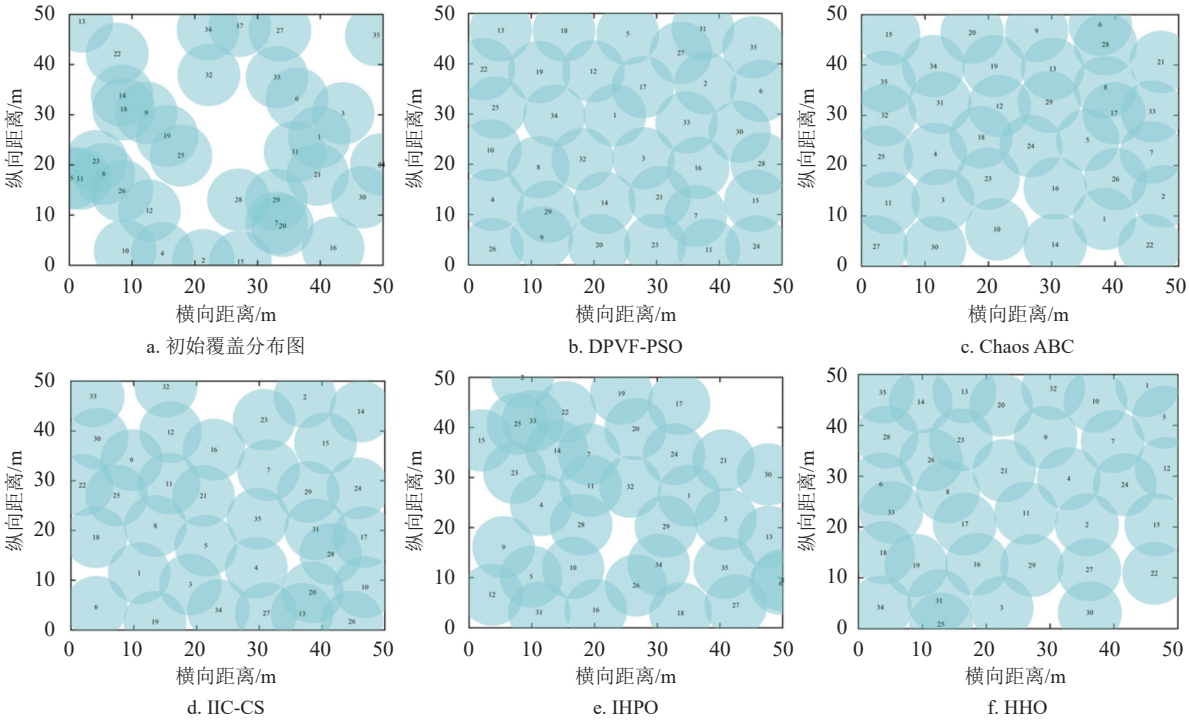


图 13　5 种算法的初始分布和覆盖优化图（无障碍物）

3.4　DPVF-PSO 算法与其他优化算法对比（有障碍物）

在实际应用中，无线传感器网络的覆盖通常会受到各种障碍物的影响。为更好地评估不同优化算法在有障碍物环境下的性能，本节仅考虑两类常见几何形状的障碍物：正方形和等腰直角三角形。前文已介绍了障碍物对 WSN 覆盖的影响、障碍物的一般类型和数学模型。在此基础上，本节对比 DPVF-PSO 算法与其他优化算法在有障碍物条件下的覆盖优化性能。

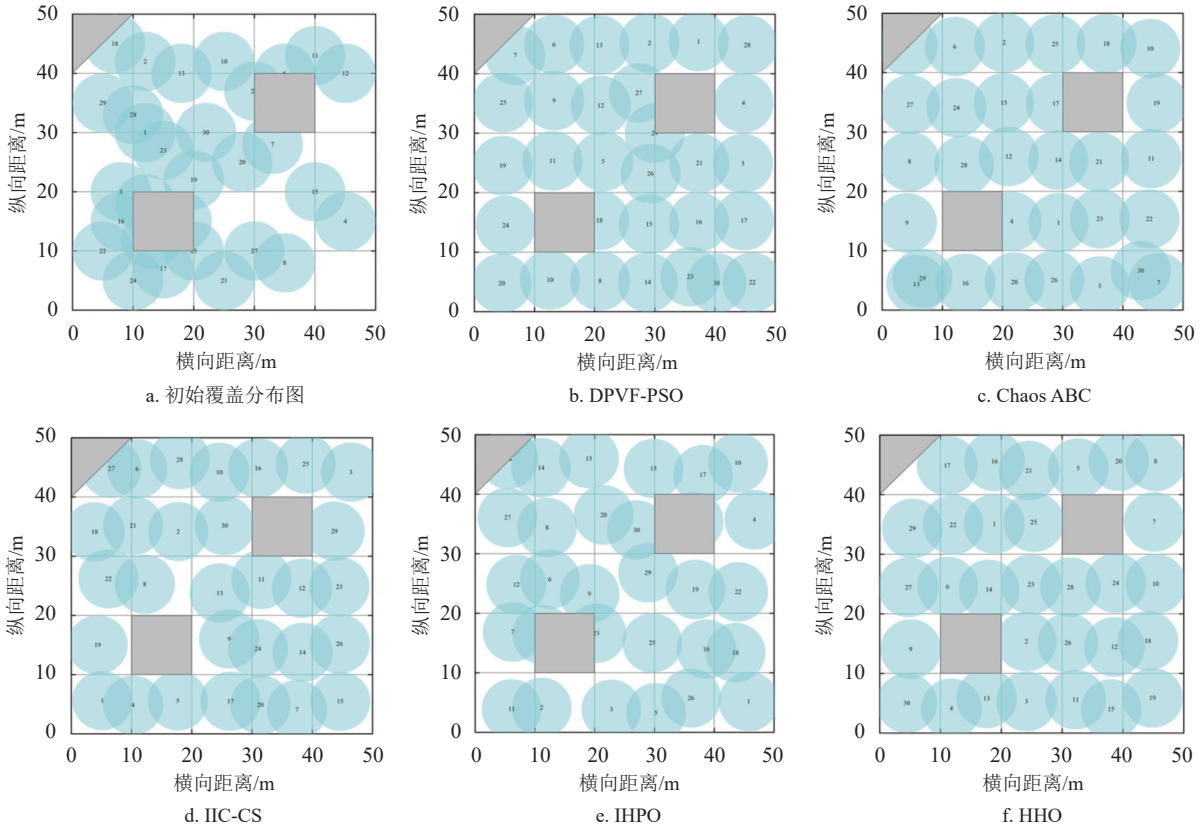
实验中，检测区域为50 m×50 m，感知半径为 5 m，迭代次数为 500 次，节点数为 30，粒子数量为 20。设置两个10 m×10 m的正方形障碍物位于点（15, 15）、（35, 35）的正中心，以及 1 个边长为 10 m的等腰直角三角形位于左上角。

对 5 种不同算法在上述设定实验场景下进行对比实验，其余基础参数与表 2 保持一致。图 14 展示了 5 种算法在该环境下的优化节点位置，帮助评估各算法在处理障碍物时的性能差异。其中，浅蓝色圆形为传感器节点的感知区域，深灰色区域为障碍物，黑色方格线为规划的网格坐标参考线。

DPVF-PSO 算法通过对栅栏区域的划分，为节点提供了明确的目标区域，从而使得节点分布更加均匀，并有效减少了不必要的计算负担。虚拟力机制引导节点避开障碍物，确保节点不会部署在不可感知的区域。融合 D*算法后，DPVF-PSO 算法能够在障碍物周围规划最短路径，优化节点的移动轨迹，减少节点的移动距离和能耗。具体而言，D*算法在 DPVF-PSO 算法中扮演了路径规划的核心角色。当节点移动时，D*算法根据当前环境的变化动态更新路径，避免了重新计算整个路径的高昂代价。因此，D*算法使得节点能够在变化的环境中实时调整路径，从而有效避免障碍物的干扰，确保节点走最短路径。在有障碍物的环境中，DPVF-PSO 算法的覆盖率比 Chaos ABC、IIC-CS、IHPO 和 HHO 算法分别高出 3.623%、5.762%、10.643% 和 4.385%。

在有障碍物的环境中，DPVF-PSO 算法通过虚拟力机制和 D*路径规划展现了显著的优势。与 Chaos ABC 算法相比，DPVF-PSO 算法在处理障碍物时

能够更有效地避开障碍并优化路径，从而提高覆盖率和能效。与 IIC-CS 算法相比，DPVF-PSO 算法的路径规划更加精确，能够有效避免障碍物带来的干扰，同时提高节点的部署效率。与 IHPO 算法相比，DPVF-PSO 算法通过动态参数调控和虚拟力引导，使得节点的分布更加均匀，冗余度更低，覆盖效果更好。与 HHO 算法相比，DPVF-PSO 算法通过引入 D*算法，能够更灵活地调整路径，避免了因障碍物造成的冗余移动，从而显著降低能耗，提升整体性能。



a. 初始覆盖分布图　b. DPVF-PSO　c. Chaos ABC

d. IIC-CS　e. IHPO　f. HHO

图 14　5 种算法的初始分布和覆盖优化图（有障碍物）

4　结束语

本文提出的 DPVF-PSO 算法在提升无线传感器网络的覆盖率和能量效率方面表现出显著优势，尤其在障碍物环境中展现了较强的适应性和鲁棒性。通过动态调整惯性权重和学习因子，结合虚拟力模型与 D*算法最短路径规划，DPVF-PSO 算法不仅优化了区域栅栏覆盖，还有效减少了节点冗余，并在路径规划中考虑了障碍物的影响，从而提升了网络的整体性能。理论上，本文通过将虚拟力模型与 D*算法结合，提出了一种新的动态优化机制，该机制能够在不同节点数量和环境复杂度下实现高效的区域栅栏覆盖，尤其是在处理障碍物时，D*算法的应用有效提高了节点部署效率并减少了路径计算中的能量消耗。

DPVF-PSO 算法的核心贡献在于其创新性地解决了现有算法面临的网络覆盖盲区和过高能量消耗的问题。通过虚拟力引导节点进行局部优化，并结合全局搜索策略，DPVF-PSO 算法能够在不同网络环境下高效地完成节点优化任务，从而提供更优的资源配置。此外，DPVF-PSO 不仅提升了网络的覆盖率，还有效降低了冗余节点，保持了较低的能量消耗，展示了其在实际应用中对低功耗需求的适应性。

实验结果表明，DPVF-PSO 算法在不同节点数量和环境复杂度下均能有效提升覆盖率。与其他算法相比，DPVF-PSO 算法在节点数量较少时便能达到较高的覆盖率，且随着节点数量的增加，其覆盖率快速提升，最终在 50 个节点时实现近 97% 的覆盖水平。

尽管在处理超大规模网络时 DPVF-PSO 算法存在一定的计算复杂度，但优异的性能和适应能力使其在实际应用中具有较高潜力。未来研究可以进一步优化算法的动态参数调整机制，以提升计算效率，并尝试融合其他智能优化算法，在超大规模网络场景中进一步验证其扩展性和智能性，为智能交通、环境监测等应用提供更加高效的解决方案。

参 考 文 献

[1] BINH N T M, BINH H T T, LUONG H V D, et al. An efficient exact method with polynomial time-complexity to achieve k-strong barrier coverage in heterogeneous wireless multimedia sensor networks[J]. Journal of Network and Computer Applications, 2024, 231: 103985.

[2] SÁMANO-ORTEGA V, ARZATE-RIVAS O, MARTÍNEZ-NOLASCO J, et al. Multipurpose modular wireless sensor for remote monitoring and IoT applications[J]. Sensors, 2024, 24(4): 1277.

[3] WU P, YU L, YI X M, et al. Research on WSN reliable ranging and positioning algorithm for forest environment [J]. Scientific Reports, 2024, 14: 5417.

[4] ZHU L F, WANG D. Robust multi-factor authentication for WSNs with dynamic password recovery[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 8398-8413.

[5] BHARGAVI K V N A, VARMA G P S, HEMALATHA I, et al. An enhanced particle swarm optimization-based node deployment and coverage in sensor networks[J]. Sensors, 2024, 24(19): 6238.

[6] DE ASSIS P A Q, RODRIGUES L R, GALVÃO R K H, et al. A genetic algorithm method for improving suboptimal sensor arrangements in coverage and connectivity problems[J]. Applied Soft Computing, 2024, 166: 112047.

[7] 余修武, 晋诗琪. 融合 VFA 和 ISSA 的多目标优化 WSN 覆盖算法[J]. 西安电子科技大学学报, 2024, 51(6): 60-72.
YU X W, JIN S Q. Multi-objective optimized WSN coverage algorithm integrating VFA and ISSA[J]. Journal of Xidian University, 2024, 51(6): 60-72.

[8] LI Z Q, ZHANG Z H, WU B C, et al. An adaptive coverage strategy for WSNs with dynamic energy decay based on multiobjective dingo optimization algorithm[J]. IEEE Sensors Journal, 2024, 24(9): 15133-15144.

[9] 贾润亮, 张海玉. 改进群体智能算法的无线传感器网络覆盖优化[J]. 西南大学学报（自然科学版）, 2024, 46(1): 155-166.
JIA R L, ZHANG H Y. Improved population intelligence algorithm for wireless sensor network coverage optimization[J]. Journal of Southwest University (Natural Science Edition), 2024, 46(1): 155-166.

[10] 谢珊, 马琳娟, 苏鑫, 等. 无线传感器网络覆盖优化的混沌交叉人工蜂群算法[J]. 南京理工大学学报, 2024, 48(3): 360-366.
XIE S, MA L J, SU X, et al. Chaotic crossover artificial bee colony algorithm for coverage optimization of wireless sensor networks[J]. Journal of Nanjing University of Science and Technology, 2024, 48(3): 360-366.

[11] 杨乐, 张达敏, 何庆, 等. 改进猎人猎物优化算法在 WSN 覆盖中的应用[J]. 计算机应用, 2024, 44(8): 2506-2513.
YANG L, ZHANG D M, HE Q, et al. Application of improved hunter-prey optimization algorithm in WSN coverage[J]. Journal of Computer Applications, 2024, 44(8): 2506-2513.

[12] 张士荣, 赵俊杰, 谈发明. 基于多策略混合改进 HHO 算法的 WSN 节点覆盖优化[J]. 计算机工程与设计, 2024, 45(2): 328-338.
ZHANG S R, ZHAO J J, TAN F M. Wireless sensor network node coverage optimization based on multi-strategy hybrid improved Harris hawk algorithm[J]. Computer Engineering and Design, 2024, 45(2): 328-338.

[13] LIU X B, WANG Y, GAO H, et al. A coverage-aware task allocation method for UAV-assisted mobile crowd sensing[J]. IEEE Transactions on Vehicular Technology, 2024, 73(7): 10642-10654.

[14] WANG Z D, HUANG L L, YANG S X, et al. Multi-strategy enhanced grey wolf algorithm for obstacle-aware WSNs coverage optimization[J]. Ad Hoc Networks, 2024, 152: 103308.

[15] MA Y Y, OTA K, DONG M X. Multi-RIS deployment location optimization for coverage enhancement in outdoor wireless communication networks[J]. IEEE Transactions on Vehicular Technology, 2024, 73(10): 15332-15344.

[16] 范兴刚, 王超, 杨静静, 等. 一种基于选择框的有向 *K*-栅栏构建算法[J]. 计算机学报, 2016, 39(5): 946-960.
FAN X G, WANG C, YANG J J, et al. A strong *K*-barrier construction scheme based on selecting box for directional sensor networks[J]. Chinese Journal of Computers, 2016, 39(5): 946-960.

[17] 党小超, 李月霞, 郝占军, 等. 一种基于改进蚁群算法的三维 *K*-栅栏覆盖算法[J]. 计算机工程, 2020, 46(2): 221-229.
DANG X C, LI Y X, HAO Z J, et al. A 3D *K*-barrier coverage algorithm based on improved ant colony algorithm[J]. Computer Engineering, 2020, 46(2): 221-229.

[18] AFIZUDEEN S, PAVITHRA R. Grundy number-based optimal sensor placement in 3D wireless sensor network[J]. IEEE Access, 2024, 12: 148502-148515.

[19] 孙泽宇, 伍卫国, 曹仰杰, 等. 无线传感器网络中能量均衡参数可控覆盖算法[J]. 西安交通大学学报, 2016, 50(8): 77-83.
SUN Z Y, WU W G, CAO Y J, et al. EBPCC: Energy balance parameters-controlled covering algorithm for wireless sensor networks[J]. Journal of Xi'an Jiaotong University, 2016, 50(8): 77-83.

[20] HU H B, YANG Y. Piecewise combinatorial particle swarm optimization[J]. Applied Soft Computing, 2024, 166: 112060.

[21] WEN Q, ZHAO X Q, CUI Y P, et al. Coverage enhancement algorithm for WSNs based on vampire bat and improved virtual force[J]. IEEE Sensors Journal, 2022, 22(8): 8245-8256.

[22] SUN J G, WANG W S, LI S Z, et al. Scheduling optimization for UAV communication coverage using virtual force-based PSO model[J]. Digital Communications and Networks, 2024, 10(4): 1103-1112.

[23] ZHANG W, WANG N X, WU W H. A hybrid path planning algorithm considering AUV dynamic constraints based on improved A* algorithm and APF algorithm[J]. Ocean Engineering, 2023, 285: 115333.

[24] YANG Y L, LUO X Y, LI W, et al. AAPF*: A safer autonomous vehicle path planning algorithm based on the improved A* algorithm and APF algorithm[J]. Cluster Computing, 2024, 27(8): 11393-11406.

[25] TIAN Y S, ZHANG D, ZHANG H B, et al. An improved cuckoo search algorithm for global optimization[J]. Cluster Computing, 2024, 27(6): 8595-8619.

责任编辑　张　莉