

一种基于动态跳数距离的有洞无线传感器网络定位方法

张松涛^{1,2} 蒋洪波² 刘文字²

(佛山职业技术学院电子信息系 佛山 528000)¹ (华中科技大学电子与信息工程系 武汉 430074)²

摘 要 在无线传感器网络中,与距离无关的定位技术一直是一项挑战性的工作。尤其是在有洞的各向异性网络中,多跳节点之间的距离估算更是一个难点。针对有洞的无线传感器网络,提出一种新的距离无关定位方法,该方法可以较好地估算未知节点到参考节点之间的距离。其主要思想是,先估算各信标节点对之间的平均单跳距离,然后选择平均单跳距离较大并且最短路径通过未知节点的信标节点对作为参考节点来估算未知节点的位置。新算法能够较好地滤除距离估算误差较大的信标节点作为参考节点。实验表明,新算法比以前的算法定位更准确。

关键词 无线传感器网络,定位,动态跳数距离

中图分类号 TP393 **文献标识码** A

Localization Algorithm Based on Dynamic Hop Distances in Wireless Sensor Networks with Holes

ZHANG Song-tao^{1,2} JIANG Hong-bo² LIU Wen-yu²

(Department of Electronics and Information, Foshan Polytechnic, Foshan 528000, China)¹

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)²

Abstract Range-free positioning technology in wireless sensor networks is a challenge. It is difficult to estimate the distances between sensor nodes which are more than one hop away in anisotropic sensor networks with holes. We introduced a range-free localization approach in anisotropic wireless sensor networks with holes, which can estimate the ranges accurately between unknown node and reference nodes. First, the algorithm estimates different average distances of one hop along the shortest path between different anchor pairs. Then some unknown node picks up the anchor pairs, whose average distance of one hop along the shortest path is larger than some threshold and the shortest path passes corresponding unknown node, as reference nodes to estimate its position. The new algorithm can filter out the anchor nodes with higher errors of estimation distances as reference nodes. Simulation results show that the localization accuracy of new algorithm is higher than previous algorithms.

Keywords Wireless sensor networks, Localization, Dynamic hop distances

1 引言

对于很多基于位置的应用而言,无线传感器网络是一种有潜在价值的工具^[1-3]。节点定位可以为这些应用提供重要的支撑。在一个传感器网络中,通常会有一小部分节点可以通过 GPS 接收装置或人为安装在指定位置的方式获得自身位置信息。这部分节点我们叫信标节点(Anchor nodes)。对于更多的其它节点,一般通过随机布撒的方式进行布局。这些节点没有自身的位置信息,我们称它们为普通节点或未知节点(Unknown nodes)。无线传感器网络节点定位的目标就是要借助少数已知位置的信标节点以及节点之间的距离测量信息或连通信息来确定普通节点的可能位置。

由于无线传感器网络定位的重要性,已提出一些很好的定位方法^[4-11]。根据是否使用节点之间的距离测量信息,可以将无线传感器网络定位方法分为基于距离的定位方法和距离无关的定位方法。距离无关定位方法相对基于距离的定位方法而言,定位误差较大。但由于它的经济适应性,距离无关

定位方法受到了越来越多人的关注。

在无线传感器网络,尤其是在有洞的各向异性网络定位中,多跳以外节点之间的距离估算一直是一个技术难点。在密集分布的各向同性均匀网络中,人们已经总结出,可以用网络平均单跳距离乘以两节点之间的最短路径跳数作为两节点之间的距离估计^[12]。因为在这种情况下,两节点之间的最短路径近似为一条直线,网络中各节点之间基于跳数的距离估计是比较可靠的。可是在各向异性网络,比如有洞的网络中,两节点之间的最短路径由于可能要绕过洞的边缘而完全偏离直线,基于网络平均单跳距离与节点之间最短路径跳数乘积的距离估计会远远大于两节点之间的实际距离(也就是直线距离)。距离估计不可靠,会造成节点定位位置严重偏离其真实位置。所以,在有洞的无线传感器网络中,距离无关的定位技术是一项具有挑战性的工作。

本文提出一种距离无关定位方法,以较好地解决有洞的各向异性网络中的节点定位问题。该算法能够较好地克服各向异性网络对节点之间距离估算的影响,选择有可靠距离估

到稿日期:2010-02-09 返修日期:2010-04-22 本文受国家自然科学基金(60803115,60873127),国家 863 计划(2007AA01Z223)资助。

张松涛 博士生,主要研究方向为无线传感器网络等,E-mail:stzhang2000@gmail.com;蒋洪波 博士,副教授,主要研究方向为计算机网络、无线传感器网络等;刘文字 博士,教授,博士生导师,主要研究方向为计算机视觉、无线通信等。

算的信标节点作为参考节点来估算未知节点的位置,从而提高节点定位准确性。如何选取有可靠估算距离的信标节点是本算法的关键。

2 相关研究工作

上节已述及,基于距离定位方法能够得到节点间比较准确的距离估计,因此节点平均定位结果相对比较准确。但它们需要额外的测距硬件支持,因此投资成本也相对较高。相反,距离无关定位方法只需要节点之间的连通信息,无需测距硬件支持。距离无关方法虽然定位结果通常不如基于距离定位方法,但对于很多应用已经足够。正是距离无关定位方法的这种经济适用性,使得该方法越来越受到人们的青睐。

由 Niculescu 和 Nath 提出的 APS 系统^[12](Ad hoc Positioning System),是较早提出的比较经典的距离无关定位方法。APS 系统最先采用距离向量交换思想,使网络中的各节点获得到各信标节点以跳数为单位的距离(distance vector hop, DV-hop)。然后信标节点利用自身坐标和到其它信标节点的跳数距离,估算网络平均单跳距离,并将该值在网络中传播。普通节点利用收到的网络平均单跳距离值和到信标节点的跳数值,估算到各信标节点的距离;进而运行基本的三角测量算法估算自身位置。DV-hop 方法在密集分布的各向同性网络中能够获得不错的定位结果,在有洞的网络中由于到信标节点的距离估计偏差较大,造成节点定位准确性急剧下降。Shang Y^[13]等提出一种对 DV-hop 的改进算法(为便于讨论,我们称此算法为 Nearest-4 算法)。该算法利用到未知节点最近的 4 个信标节点作为参考节点来对未知节点进行定位估算。两节点之间跳数距离近,受网络各向异性影响较小,这是可以理解的。但是,有些普通节点近距离内的可用信标节点数量可能很少,不得不选用较远距离的信标节点;另外,选择最近 4 个信标节点作为参考节点的机制也会排斥一些距离较远但有较好距离估计的信标节点,从而降低了节点定位的准确性。Wang 和 Xiao^[14]提出一种基于上界约束的多圆交迭定位方法。该算法以到信标节点的估算距离为上界,以各参考信标节点为圆心,以未知节点到各参考信标节点的估算距离为半径的各圆的交迭部分作为未知节点的可能估算位置。Li M 和 Liu Y H^[15]提出一种针对有洞网络的传感器网络定位方法 REP(Rendered Path)。该算法通过路径着色和增加虚拟圆的方法,可以较好地估算最短路径偏离直线时两节点之间的距离,从而实现较准确的节点定位。不过该算法设计思路比较复杂,要进行比较复杂的几何计算不太容易。

3 Hophole 定位方法

本文提出的算法是基于跳数(hop counts)距离的,而且它能较好地解决有洞(hole)的各向异性的网络节点定位问题。为便于讨论,我们把本文的算法叫 Hophole 定位方法。Hophole 算法的关键是,针对不同的普通节点,利用不同信标节点对(两信标节点组成的一对信标节点,称信标节点对)动态平均单跳距离滤除距离估算不准确的信标节点,保留距离估算比较准确的信标节点作为参考节点对普通节点进行坐标位置估算,以提高定位准确性。

这里我们假设:1) 网络的各向异性主要是由网络中的空洞造成的,即除了有洞的区域外,网络的其它部分节点大致是

均匀分布的;2) 所有节点的通信半径大致是相同的,网络中有一部分节点是信标节点,有自身位置信息;3) 我们不需要节点之间的距离测量,只需感知邻居节点的存在,也就是我们的定位方法是距离无关的。

3.1 Hophole 算法产生的动机

在 DV-hop^[12]方法中,作者先利用信标节点的已知位置 and 不同信标节点之间的最短路径跳数,估算网络的平均单跳距离,并将该值在网络中传播。然后普通节点利用网络平均单跳距离和到信标节点的最短路径跳数估算到相应信标节点的距离;最后运行三角测量算法来估算普通节点坐标位置。这种算法能较好地工作的前提条件是:在密集分布的均匀网络中,两节点之间的最短路径通常比较接近一条直线,用最短路径距离可以较好地估计两节点之间的测量距离。可是,现实的无线传感器网络中,经常会有局部的空洞出现,比如由于较大障碍物的存在而形成的空洞。在有洞的网络中,两点之间的最短路径可能由于要绕过洞的边缘而明显偏离直线。此时,两点之间的最短路径距离会远远大于两点之间的测量距离,也就是直线距离。也就是说,用网络平均单跳距离与两节点之间最短路径跳数的乘积来估算的两节点之间的距离,会远远大于两节点之间的测量距离。

我们的想法就是,尽可能利用到普通节点的最短路径接近于直线的信标节点作为参考节点,从而用基于跳数的最短路径距离可以较准确地估计未知节点到参考节点的测量距离,提高定位准确性。通过观察可以发现,在用两信标节点之间的最短路径跳数和信标节点的坐标来估算两信标节点之间的平均单跳距离时,如果得到的平均单跳距离越大,说明该两信标节点之间的最短路径越有可能接近一条直线。正是基于以上思路,产生了下面的 Hophole 算法。

3.2 Hophole 算法具体实现

首先,估算并保存不同信标节点对之间的平均单跳距离矩阵 H 。在一个连通的网络中,各信标节点利用广播的方式可以获得各信标节点到网络中其它各节点(包括信标节点和普通节点)之间的最短路径跳数。各信标节点利用自身位置和其它信标节点位置可以计算到其它各信标节点之间的距离。因此,用下式

$$h_{ij} = \frac{dis_{ij}}{hc_{ij}} \quad (1)$$

可以估算信标节点 i 和 j 之间的最短路径上的平均单跳距离 h_{ij} ,我们简称为信标节点 i 和 j 之间的平均单跳距离。式(1)中, dis_{ij} 和 hc_{ij} 分别为信标节点 i 与 j 之间的距离(distance)和最短路径跳数(hop counts)。显然,不同信标节点对之间的平均单跳距离通常是不一样的。所有信标节点对之间的平均单跳距离就组成了平均单跳距离矩阵 H 。

我们用图 1 来分析路径曲直性与路径上的平均单跳距离之间的关系。图 1 中 $A1, A2$ 和 $A3$ 为 3 个信标节点,由它们围成的下面的近似扇形为传感器网络中由于障碍物形成的洞。从图 1 我们可以看到, $A1$ 到 $A3$ 的最短路径由于要绕过洞的边缘而完全偏离直线,而 $A1$ 到 $A2$ 的最短路径则比较接近一条直线。 $A1$ 到 $A2$ 的距离 dis_{12} 与 $A1$ 到 $A3$ 的距离 dis_{13} 基本上是相等的。可是, $A1$ 到 $A3$ 的最短路径由于要绕过洞的边缘,其跳数比 $A1$ 到 $A2$ 的最短路径增加了 4 跳。所以,按式(1)估算的路径 $A1$ 到 $A2$ 上的平均单跳距离会比路径

A1 到 A3 上的平均单跳距离大很多。也就是说,在一个节点通信半径基本一致的网络中,平均单跳距离比较大的信标节点对之间的最短路径会比较接近直线,而平均单跳距离比较小的信标节点对之间的最短路径会明显偏离直线。

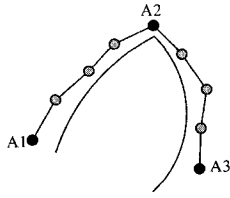


图1 路径曲直性与平均单跳距离关系示意图

按照上面的分析,我们对有较大的平均单跳距离的信标节点对比较感兴趣。所以,在对某普通节点进行坐标位置估算时,如果某信标节点对的最短路径同时满足下面两个条件,我们会选择该信标节点对为该普通节点的参考节点:(1) 某信标节点对的最短路径通过该普通节点;(2) 该信标节点对之间的最短路径上的平均单跳距离大于某一阈值(比如整个网络平均单跳距离值)。我们简单分析一下这两个条件的意义。条件(2),信标节点对之间的最短路径上的平均单跳距离大于某一阈值,表明该信标节点对之间的最短路径接近一条直线。若整个最短路径接近一条直线,那么路径的任意一个端点到路径上的任意一点形成的较短路径(也就是整个路径的一个子集)同样接近一条直线。条件(1),信标节点对的最短路径通过该普通节点,表明该普通节点在该接近于直线的最短路径上。该普通节点到其中任意一端的信标节点的最短路径作为整个路径的一个子集,也是接近直线的。

按上述方法选择的信标节点到待求普通节点的最短路径是接近直线的。然后我们用信标节点到普通节点的最短路径跳数与动态平均单跳距离的乘积来估算信标节点到普通节点的距离,可以得到比较准确的距离估计。所谓动态平均单跳距离是指:不同信标节点到不同普通节点之间采用不同信标节点对之间的平均单跳距离,而不用全网平均单跳距离。

这里,我们进一步说明普通节点如何选择参考信标节点和相应的平均单跳距离。通过类似距离向量交换的方法,各信标节点可以获得到其它信标节点的最短路径。各普通节点也可以记录哪些信标节点对之间的最短路径通过了本节点。在这个过程中,同时产生了全网平均单跳距离和各信标节点对之间的平均单跳距离矩阵 H 。

设最短路径通过某普通节点的信标节点对集合为

$$AS_i = \{(a_j, a_k)\} \quad (2)$$

式中, i 为普通节点标识, j, k 为信标节点标识, a_j, a_k 表示信标节点。然后,对 AS_i 中的元素按不同信标节点对之间的平均单跳距离 h_{jk} 大小进行排序,按一定的阈值滤除掉平均单跳距离较小的信标节点对,得到平均单跳距离较大的信标节点对子集

$$AST_i = \{(a_j, a_k) | (a_j, a_k) \in AS_i, h_{jk} \geq \text{some threshold}\} \quad (3)$$

我们将在子集 AST_i 中按平均单跳距离大小依次选择参考信标节点和相应的平均单跳距离 h_{jk} , 形成如下参考信标节点和平均单跳距离对集合

$$AH_i = \{(a_j, h_{jk}) | (a_j, a_k) \in AST_i, \text{ or } (a_k, a_j) \in AST_i\} \quad (4)$$

也就是说,普通节点 i 将在集合 AH_i 中依次选择参考信标节点 a_j 和相应的平均单跳距离 h_{jk} 。

在按信标节点对之间的最短路径通过普通节点以及该最短路径上的平均单跳距离大于某一阈值的思路选择参考信标节点时,有些普通节点很可能无法找到足够多的参考信标节点来运行基本的三角测量定位算法。此时,我们将参照文献[13]中的思路,继续选择离该普通节点跳数距离最小的若干信标节点作为参考节点。对以跳数距离最小产生的参考信标节点,我们将用网络平均单跳距离和相应信标节点到普通节点之间最短路径跳数的乘积来估算参考信标节点到待求普通节点的距离。

最后总结一下 Hophole 定位算法伪码,如图 2 所示。

输入:随机分布的有洞的传感器网络,含 $m+n$ 个节点

其中有, m 个信标节点, n 个普通节点

输出: n 个普通节点的估算位置

- 1 各信标节点广播自身位置,各节点记录各信标节点位置及到各信标节点最短路径跳数。
- 2 各信标节点计算到其它各信标节点的平均单跳距离及网络平均单跳距离。
- 3 对所有普通节点 $s_i; i=1$ to n
- 4 次产生集合 AS_i, AST_i , 和 AH_i 。
- 5 在 AH_i 中依次选择参考信标节点 a_j 和相应的平均单跳距离 h_{jk} , 用 h_{jk} 与相应最短路径跳数的乘积估算 s_i 到 a_j 的距离。
- 6 如果参考信标节点数量不足:
- 7 按到 s_i 跳数距离最小的原则,继续选择参考信标节点 a_j 。此时,用网络平均单跳距离与相应最短路径跳数的乘积估算 s_i 到 a_j 的距离。
- 8 如果参考信标节点数量足够,比如大于或等于 3:
- 9 运行基本三角测量定位算法估算 s_i 的坐标。
- 10 否则:
- 11 用 $(0, 0)$ 作为估算坐标,表示定位失败。
- 12 输出各普通节点坐标。

图2 Hophole 定位算法

图 2 中,1-2 行进行网络初始化,得到一些基本参数,包括信标节点位置、各节点到各信标节点的最短路径跳数、平均单跳距离及网络平均单跳距离。4-11 行循环执行,用于估算各普通节点位置坐标。

4 性能评估

本文搭建实验平台,实现了 Hophole 定位算法,并做了大量的仿真实验。我们将 Hophole 算法的相关性能,包括平均定位误差、定位误差累积分布等与之前基于平均单跳距离的算法,包括选择所有信标节点作为参考节点(DV-hop 方法)或选择跳数距离最小的 4 个信标节点作为参考节点(Nearest-4 方法)等方法,进行了对比分析。为了更好地与 Nearest-4 方法进行对比,我们的 Hophole 方法也最多只选择 4 个最可靠的信标节点作为参考节点进行节点定位。

4.1 仿真环境设置

为了测试在有洞的网络中 Hophole 定位算法的性能,我们考虑在一个 $100\text{m} \times 100\text{m}$ 的正方形区域中有一个约 $42.5\text{m} \times 42.5\text{m}$ 的小正方形。小正方形的边分别与大正方形的两条对角线平行,两个正方形的中心重合。传感器节点在属于大正方形而不属于小正方形的区域内随机布撒。也就是说,小正

方形形成了网络的一个空洞。传感器节点通信半径 R 为 10m, 在多次实验中保持不变。传感器节点数量从 400 个逐渐增加到 1000 个。信标节点比例分别假定为 10% 和 20%, 做 2 组实验。图 3 是符合上述区域设置的 400 个节点时的一个含洞的传感器网络节点分布图, 红色实心点为信标节点。

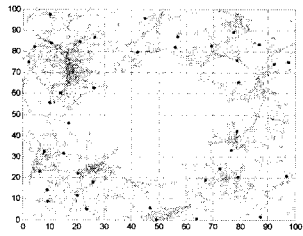


图 3 一个含洞网络示例, 400 个节点

定位误差定义为节点定位位置与节点真实位置的距离。为便于分析, 定位误差按传感器节点通信半径进行了归一化。如本实验节点通信半径为 10m, 若定位误差为 3.6m, 则按通信半径归一化的定位误差为 36%。所有实验结果为 100 次实验的平均值。

4.2 仿真结果分析

图 4 是在 10% 信标节点比例下平均定位误差随着网络节点数的变化情况。3 个算法定位误差总的变化趋势都是随着节点数的增加而减小。DV-hop 方法定位结果最差, 因为它选择所有信标节点作为参考节点对普通节点进行定位, 有很多的信标节点到待求普通节点的最短路径由于要绕过洞的边缘而严重偏离直线, 造成到这些信标节点的基于平均单跳距离的距离估计出现较大偏差, 所以 DV-hop 算法在有洞网络中的定位误差很大。Nearest-4 方法选择最近的 4 个信标节点作为参考节点, 受网络各向异性影响相对较小, 所以定位结果有了很大的改善。Hophole 算法尽量选择到待求普通节点最短路径接近直线的信标节点作为参考节点, 并且用动态平均单跳距离估算到参考信标节点的距离, 以提供更加准确的距离估计, 从而提高定位准确性。从图 4 可以看出, 在不同节点总数的情况下, Hophole 算法的定位误差比 Nearest-4 算法低 5~10 个百分点; 在节点总数接近 1000 时, 定位误差比 Nearest-4 算法降低了 10.5 个百分点以上。

图 5 是在 10% 信标节点比例、节点总数为 600 时 3 个算法的累积误差分布情况。我们以 Hophole 算法曲线为例加以说明, 定位误差为 50% 对应的定位节点比例约为 70%, 表示定位误差小于等于 50% 的节点约占普通节点总数的 70%。也就是说, 在不超过某一定位误差的相同范围内, 累积定位节点比例越高, 说明定位算法准确性越好。从图 5 可以看出, 累积误差分布反映的定位准确性与图 4 平均定位误差反映的定位准确性是一致的: 在有洞的各向异性网络中, 3 个算法中 DV-hop 算法定位准确性最差, 而且与其它两个算法差距很大, Hophole 算法最好。在相同的较小误差范围内 (30%~90%), DV-hop 算法定位节点比例比其它两算法低 20%~30%, Hophole 算法定位节点比例比 Nearest-4 算法高 2%~8%。

与图 4 和图 5 类似, 图 6 和图 7 分别是平均定位误差随节点数的变化情况图和累积误差分布图; 与图 4 和图 5 中 10% 信标节点比例不同的是, 图 6 和图 7 中信标节点比例为 20%。

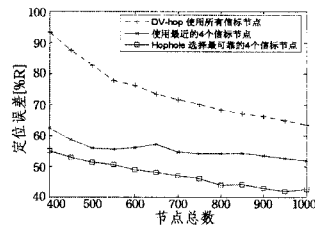


图 4 定位误差随节点总数变化情况 (10% 信标节点)

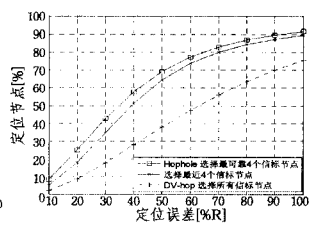


图 5 累积误差分布 (10% 信标节点, 600 总节点)

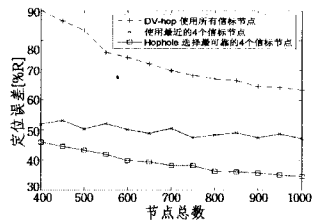


图 6 定位误差随节点总数变化情况 (20% 信标节点)

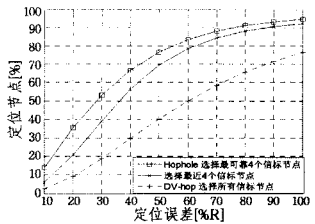


图 7 累积误差分布 (20% 信标节点, 600 总节点)

对比图 6 和图 4, 在 20% 信标节点比例时, 其平均定位误差可以得到类似于 10% 信标节点比例时的结论: 在有洞的各向异性网络中, Hophole 算法和 Nearest-4 算法定位准确性要明显高于 DV-hop 算法, 它 Hophole 算法又要明显好于 Nearest-4 算法。同样地, 对比图 6 和图 4, 随着节点总数的增加, Hophole 算法定位误差比 Nearest-4 算法下降更快。在不同节点总数的情况下, Hophole 算法的定位误差比 Nearest-4 算法低 6~13 个百分点; 而 10% 信标节点比例时 Hophole 算法定位误差比 Nearest-4 算法低 5~10 百分点。

从图 7 可以看出, 在 20% 的信标节点比例和 600 个总节点情况下, 在相同的较小误差范围内 (20%~80%), DV-hop 算法累积定位节点比其它两个算法明显要低 20%~35%; 而 Hophole 算法累积定位节点比 Nearest-4 算法要高出 3%~15%。

图 4 到图 7, 从平均定位误差到累积误差分布, 变化网络相关参数, 包括网络节点总数、信标节点比例等, 在有洞的各向异性传感器网络中对 Hophole 等算法进行了多方面的分析。实验表明, Hophole 算法定位性能较好。

结束语 本文利用不同信标节点对之间平均单跳距离的差异, 提出一种距离无关定位算法, 在有洞的各向异性网络中能够较好地完成节点定位。其主要思想是选择最短路径上平均单跳距离比较大的信标节点对作为待求普通节点的参考节点, 可以获得比较准确的距离估算, 从而得到比较可靠的定位结果。

参考文献

- [1] 孙利民, 李建中, 陈渝, 等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005
- [2] Want R, Schilit B. Guest editors' introduction: Expanding the horizons of location-aware computing[J]. Computer, 2001, 34(8): 31-34
- [3] Culler D, Estrin D, Srivastava M. Guest editors' introduction: Overview of sensor networks[J]. Computer, 2004, 37(8): 41-49
- [4] Savvides A, Han C C, Srivastava M B. Dynamic fine-grained localization in ad-hoc networks of sensors[C] // Proceedings of

ACM MobiCom, Rome, Italy, July 2001

- [5] Savarese C, Rabaey J, Langendoen K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks[C]// Proceedings of the USENIX technical annual conference, Monterey, CA, USA, June 2002
- [6] He T, Huang C D, Blum B M, et al. Range-free localization schemes for large scale sensor networks[C]// Proceedings of ACM MobiCom, San Diego, CA, USA, Sep. 2003
- [7] Hu L, Evans D. Localization for mobile sensor networks[C]// Proceedings of ACM MobiCom, Philadelphia, PA, USA, Sep 26-Oct 1, 2004
- [8] Rudafshani M, Datta S. Localization in wireless sensor networks [C]// Proceedings of ACM/IEEE IPSN, Cambridge, MA, USA, Apr. 2007
- [9] Lederer S, Wang Y, Gao J. Connectivity-based localization of large scale sensor networks with complex shape[C]// Proceedings of IEEE INFOCOM, Phoenix, AZ, USA, Apr. 2008
- [10] Kung H T, Lin C, Lin T, et al. Localization with snap-inducing shaped residuals(SISR); coping with errors in measurement[C]// Proceedings of ACM MobiCom, Beijing, China, Sep. 2009
- [11] Zhong Z, He T. Achieving range-free localization beyond connectivity [C] // Proceedings of ACM SenSys, Berkeley CA, USA, Nov. 2009
- [12] Niculescu D, Nath B. Ad-hoc positioning system[C]// Proceedings of IEEE Globecom, San Antonio, TX, USA, Nov. 2001
- [13] Shang Y, Shi H, Ahmed A. Performance study of localization methods for ad-hoc sensor networks[C]// Proceedings of IEEE MASS, Fort Lauderdale, FL, USA, Oct. 2004
- [14] Wang C, Xiao L. Locating sensors in concave areas [C]// Proceedings of IEEE INFOCOM, Barcelona, Catalunya, Spain, Apr. 2006
- [15] Li M, Liu Y H. Rendered path, range-free localization in anisotropic sensor networks with holes [C] // Proceedings of ACM MobiCom, Montreal, Quebec, Canada, Sep. 2007

(上接第 19 页)

统的主要原因。

虽然基于学习的规划系统的研究已经取得了一定的成果,但如何在降低学习代价的同时,提高学习的质量,尽可能减少学习对规划系统整体性能的负面影响,使得基于学习的规划系统能通过有效学习在性能表现上能可靠地、令人信服地超越非学习的规划系统,改变目前最佳的非学习的规划系统的表现优于所有基于学习的规划系统的尴尬局面,仍然是尚待解决的难题。

参 考 文 献

- [1] Fox M, Thiébaux S. Advances in Automated Plan Generation [J]. Artificial Intelligence, 2009, 173(5/6): 501-788
- [2] 国际智能规划大赛网址 [EB/OL]. <http://ipc.informatik.uni-freiburg.de/>, 2008
- [3] 智能规划器的部分列表 [EB/OL]. <http://www.iai.ed.ac.uk/links/planning.html>, <http://planning.cis.strath.ac.uk/planning/index.php?page=planners>, <http://www.csc.ncsu.edu/faculty/stamant/planning-resources.html>
- [4] Bylander T. The computational complexity of propositional STRIPS planning [J]. Artificial Intelligence, 1994, 69(1/2): 165-204
- [5] Bacchus F, Kabanza F. Using temporal logics to express search control knowledge for planning [J]. Artificial Intelligence Journal, 2000, 16: 123-191
- [6] Nau D, Cao U, Lotem A, et al. Shop: Simple hierarchical ordered planner [C] // Proceedings of the International Joint Conference on Artificial Intelligence, 1999: 968-973
- [7] Fikes R E, Nilsson N J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving [J]. Artif. Intell., 1971, 2(3/4): 189-208
- [8] Minton S, Carbonell J, Knoblock C A, et al. Explanation-based learning: A problem solving perspective [J]. Artificial Intelligence Journal, 1989, 40: 63-118
- [9] Estlin T A, Mooney R. Multi-strategy learning of search control for partial-order planning [C] // Proceedings of 13th National Conference on Artificial Intelligence, 1996
- [10] Huang Yi-Cheng, Selman B, Kautz H. Learning declarative control rules for constraint-based planning [C] // Proceedings of Seventeenth International Conference on Machine Learning, 2000: 415-422
- [11] Aler R, Borrajo D, Isasi P. Using genetic programming to learn and improve control knowledge [J]. Artificial Intelligence Journal, 2002, 141(1/2): 29-56
- [12] Khardon R. Learning action strategies for planning domains [J]. Artificial Intelligence Journal, 1999, 113(1/2): 125-148
- [13] Martin M, Geffner H. Learning generalized policies in planning domains using concept languages [C] // Proceedings of Seventh International Conference on Principles of Knowledge Representation and Reasoning, 2000
- [14] Yoon Sungwook, Fern A, Givan R. Learning Control Knowledge for Forward Search Planning [C] // JMLR, 2008, 9: 683-718
- [15] Dzeroski S, Raedt L D, Driessens K. Relational reinforcement learning [J]. Machine Learning Journal, 2001, 43: 7-52
- [16] Fern A, Yoon Sungwook, Givan R. Approximate policy iteration with a policy language bias: Solving relational markov decision processes [J]. Journal of Artificial Intelligence Research, 2006, 25: 85-118
- [17] Botea A, Enzenberger M, Muller M, et al. Macro-FF: Improving AI planning with automatically learned macro-operators [J]. Journal of Artificial Intelligence Research, 2005, 24: 581-621
- [18] Coles A I, Smith A J. Marvin: A heuristic search planner with online macro-action learning [J]. Journal of Artificial Intelligence Research, 2007, 28: 119-156
- [19] Gerevini A, Saetti A, Vallati M. An Automatically Configurable Portfolio-based Planner with Macro-actions, PbP [C] // Proceedings of 19th International Conference on Automated Planning and Scheduling, 2009
- [20] Gerevini A, Schubert L K. Discovering state constraints in DISCOPLAN; some new results [C] // Proceedings of National Conference on Artificial Intelligence, AAAI Press/The MIT Press, 2000: 761-767
- [21] Fox M, Long D. The automatic inference of state invariants in TIM [J]. Journal of Artificial Intelligence Research, 1998, 9: 367-421
- [22] Rivest R L. Learning decision lists [J]. Machine Learning, 1987, 2(3): 229-246