

文章编号: 1001-0920(2012)01-0156-05

基于改进粒子群优化的无线传感器网络定位算法

黄艳¹, 臧传治², 于海斌²

(1. 苏州市职业大学 电子信息工程系, 江苏 苏州 215104; 2. 中国科学院 沈阳自动化研究所, 沈阳 110016)

摘要: 无线电干涉定位系统获取的干涉距离是4个传感器节点间距离的线性组合值. 针对以两个节点间距离作为输入的传统定位算法无法直接利用上述干涉距离进行定位的问题, 提出一种基于改进粒子群优化的定位方法. 利用干涉距离的实验数据, 分析比较了遗传算法和改进粒子群优化在无线传感器网络节点定位问题中的性能. 结果表明, 基于改进粒子群优化的定位方法的平均耗费时间远远小于基于遗传算法的定位方法, 具有更高的优化效率.

关键词: 无线传感器网络; 干涉定位; 粒子群优化; 遗传算法

中图分类号: TP393

文献标识码: A

Localization method based on modified particle swarm optimization for wireless sensor networks

HUANG Yan¹, ZANG Chuan-zhi², YU Hai-bin²

(1. Department of Electronic Information Engineering, Suzhou Vocational University, Suzhou 215104, China; 2. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. Correspondent: HUANG Yan, E-mail: hyan@jssvc.edu.cn)

Abstract: The interferometric range which the radio interferometric positioning system obtains is a linear combination of distances among four sensor nodes. For the problem that the traditional localization methods using range estimates between a pair of nodes as input can not be applied directly in localization, a localization method based on modified particle swarm optimization(PSO) is proposed. Experimental data of the interferometric range are used to compare the performance of two localization methods based on genetic algorithm(GA) and modified PSO for wireless sensor networks. The results show that the localization method based on modified PSO takes less time and achieves higher optimizing efficiency.

Key words: wireless sensor networks; interferometric positioning; particle swarm optimization; genetic algorithm

1 引言

在无线传感器网络(WSN)的各种应用中, 监测到事件之后人们关注的一个重要问题即是该事件发生的位置. 因此, 位置信息是传感器节点采集数据中不可缺少的部分, 没有位置信息的监测通常毫无意义. 确定事件发生的位置或采集数据的传感器节点位置是WSN最基本的功能之一, 对WSN应用的有效性起着关键的作用^[1-2].

近年来, 越来越多的应用对定位精度不断提出更高的要求, 从低精度区域定位, 到米级, 甚至厘米级、毫米级的高精度无线空间三维定位. 如何提高节点定位精度已成为当前WSN定位技术的研究热点^[3-7]. Maroti等人^[8-9]以新的实现形式将传统的无线

电干涉原理应用到无线传感器平台上, 组建了无线电干涉定位系统(RIPS). 实验表明, 在多径效应很小的理想环境下, 该系统的定位精度可达到厘米级, 从而证明该原理很有可能实现高精度的定位目标. 目前, 该方法距离实用化还有一段距离, 存在许多关键问题需要解决^[10].

RIPS中获得的干涉距离是4个传感器节点间距离的线性组合值 d_{ABCD} , 即两个发送节点A和B以及两个接收节点C和D间距离的线性组合值, 计算方法如图1所示, 其中 d_{ij} 为两个节点间的距离, 而且是未知的.

现有的定位算法都是以两个节点间的距离 d_{ij} 作为输入, 无法直接利用上述干涉距离 d_{ABCD} 计算节点

收稿日期: 2010-09-16; 修回日期: 2011-07-07.

基金项目: 国家自然科学基金项目(60725312); 江苏省自然科学基金面上项目(BK2011318); 苏州市职业大学创新团队基金项目(3100125); 苏州市职业大学校青年基金项目(2010SZDQ01).

作者简介: 黄艳(1981-), 女, 讲师, 博士, 从事无线传感器网络、信息处理的研究; 于海斌(1964-), 男, 研究员, 博士生导师, 从事无线通信、分布控制系统等研究.

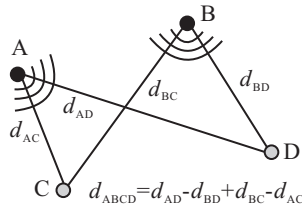


图1 干涉距离

位置. Maroti 等人采用基于遗传算法的最优化方法搜索节点坐标的近似值, 但是遗传算法的复杂度高且收敛速度较慢. 对此, 本文采用计算简单、收敛速度快的粒子群优化方法搜索节点坐标的近似值, 并利用干涉距离的实验数据分析比较两种最优化方法在 RIPS 节点定位问题中的性能.

2 基于遗传算法的 RIPS 节点定位

2.1 遗传算法

1975 年 Michigan 大学的 Holland 教授^[11]提出了遗传算法(GA), 这是一种基于自然选择和基因遗传学原理的高度并行性全局搜索算法. GA 由代表问题所在解集的一个种群开始, 该种群由经过基因编码的一定数目的个体组成. 初始种群产生之后, 按照适者生存和优胜劣汰的原理, 逐代演化产生出越来越好的近似解. 在每一代, 根据问题域中个体的适应度大小挑选个体, 并借助自然遗传学的遗传算子进行组合交叉和变异, 产生出代表新的解集的种群. 如同自然进化一样, 后代种群比前代种群更加适应环境. 该过程循环执行直到满足结束条件. 末代种群中的最优个体经过解码便可作为问题的近似最优解.

2.2 基于遗传算法的定位方法

给定 n 个待定位节点和一组干涉距离测量值 M , 待定位节点 i 的三维坐标为 (x_i, y_i, z_i) . 其中: $0 \leq x_i \leq x_{\max}, 0 \leq y_i \leq y_{\max}, 0 \leq z_i \leq z_{\max}$. $x_{\max}, y_{\max}, z_{\max}$ 为节点坐标允许的最大取值, 由定位算法的搜索空间决定. 基于无线电干涉测距的定位问题可描述为如下的最优化问题:

$$\min e(s) = \frac{1}{m} \sqrt{\sum_{ABCD \in M} [d_{ABCD} - d_{ABCD}(s)]^2}. \quad (1)$$

其中: s 为 GA 中的一个个体, 即解空间的一个解, 亦即所有待定位节点的坐标集合 $\{(x_i, y_i, z_i)\}_{i=1, \dots, n}$; m 为 M 中干涉距离的数量; d_{ABCD} 为测量得到的干涉距离, $d_{ABCD}(s)$ 为根据 s 计算得到的干涉距离.

基于 GA 的定位方法采用式 (1) 的误差函数作为适应度函数. 为了获得充分的定位约束条件, 需要对不同的节点组合进行多次测量. 基于 GA 的定位方法通过搜索寻找使测量得到的干涉距离和计算得到的干涉距离误差最小的节点坐标近似值. 基于遗传算法的定位方法具体步骤如下:

Step 1: 生成初始种群, 群体规模大小为 populationSize.

Step 2: 在初始种群中随机选择得到被选种群, 群体规模大小为 subpopulationSize.

Step 3: 用式 (1) 评估被选种群中的所有个体.

Step 4: 根据误差大小对被选种群按升序排列.

Step 5: 淘汰掉被选种群中最差的 20% 个体, 从最好的 20% 个体中随机选择两个个体作为父代, 利用交叉算子和变异算子生成新的个体, 直到获得新的被选种群.

Step 6: 检验是否符合结束条件, 如果当前的迭代次数达到了预先设定的最大次数或者达到最小错误要求, 则输出最优解; 否则, 转 Step 2.

上述方法中采用了如下的遗传算子生成新的个体:

1) 交叉算子. 对于每一个待定位节点, 随机选择一个父代个体中的坐标近似值作为新个体 s 中的坐标.

2) 变异算子. ① 用高斯随机数, 以一定的概率更新每一个待定位节点的坐标; ② 随机更新一个待定位节点的坐标; ③ 用高斯随机数更新所有待定位节点的坐标.

3 基于改进粒子群优化的 RIPS 节点定位

3.1 粒子群优化算法

Kennedy^[12]于 1995 年提出了粒子群优化算法, 这是一种新的基于群体智能理论的演化计算方法, 是对鸟群觅食过程中的迁徙和聚集的模拟, 也可以说是对社会心理学的一种模拟.

PSO 算法首先初始化一群随机粒子, 每个粒子具有位置和速度两个特征, 还有一个由被优化函数决定的适应值, 适应值的大小决定了粒子的优劣. 在每一次迭代中, 粒子通过跟踪两个“极值”来更新自己: 一个是粒子本身所找到的最好解, 即个体极值; 另一个是整个粒子群找到的最好解, 即全局极值. 假设第 k 代群体中第 i 个粒子的位置为 $X_i^k = [x_{i1}^k, x_{i2}^k, \dots, x_{iD}^k]^T$, 速度为 $V_i^k = [v_{i1}^k, v_{i2}^k, \dots, v_{iD}^k]^T$, 则更新方程为

$$v_{id}^{k+1} = \omega \cdot v_{id}^k + c_1 \cdot \text{rand}_1 \cdot (\text{pbest}_{id} - x_{id}^k) + c_2 \cdot \text{rand}_2 \cdot (\text{gbest}_d - x_{id}^k), \quad (2)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}. \quad (3)$$

其中: ω 为惯性权重; v_{id}^k 为粒子 i 在第 k 次迭代中第 d ($d \in [1, D]$) 维的速度, 被限制在 $[v_{\min}, v_{\max}]$ 之内; c_1 和 c_2 为学习因子或加速系数; rand_1 和 rand_2 为 $[0, 1]$ 之间的随机数; x_{id}^k 为粒子 i 在第 k 次迭代中第 d 维的

位置, 被限制在 $[x_{\min}, x_{\max}]$ 之内; pbest_{id} 为粒子 i 的个体极值第 d 维的位置; gbest_d 为全局极值第 d 维的位置. 两个“极值”在迭代过程中不断更新, 最后输出的全局极值即是算法得到的最优解.

上述是全局版 PSO 算法, 虽然收敛快, 但有时会陷入局部最优. 局部版 PSO 通过保持多个吸引子来避免早熟, 用其中一部分作为粒子的邻居. 所有邻居中的最好解即为局部极值, 用 lbest 表示其位置. 粒子通过跟踪 pbest 和 lbest 来更新自己. 此外, 压缩因子有助于确保 PSO 收敛, 其速度更新方程为

$$v_{id}^{k+1} = \chi(v_{id}^k + c_1 \cdot \text{rand}_1 \cdot (\text{pbest}_{id} - x_{id}^k) + c_2 \cdot \text{rand}_2 \cdot (\text{gbest}_d - x_{id}^k)). \quad (4)$$

其中: $\chi = 2 / |2 - \phi - (\phi^2 - 4\phi)^{1/2}|$ 为压缩因子; $\phi = c_1 + c_2$, 且 $\phi > 4$.

3.2 基于改进粒子群优化的定位方法

尽管 PSO 和 GA 都是基于群体和适应值, 但 PSO 的优势在于结构相对简单并且没有许多参数需要调整. 粒子还有一个重要的特点, 就是具有记忆, 所有粒子都共享迄今为止问题最好的解. 而在 GA 中, 一旦种群改变, 问题原来的信息将被破坏. 与 GA 相比, 在大多数情况下, 所有的粒子可能更快地收敛于最优解. 因此本文提出采用基于 PSO 的最优化方法搜索节点坐标的近似值.

通过实验可以发现, 直接采用上述 PSO 算法解决 RIPS 节点定位问题时, 搜索得到的节点坐标近似值误差非常大. 这是因为, PSO 无论是早熟收敛还是全局收敛, 粒子群中的粒子都会出现“聚集现象”, 所有粒子或者聚集在某一特定位置, 或者聚集在某几个特定位置. 要想克服早熟收敛, 就必须提高 PSO 跳出局部最优解的能力. 从上节对 PSO 的分析可见, PSO 的信息共享机制不同于 GA. 在 GA 中, 染色体互相共享信息, 整个种群比较均匀地向最优区域移动, 遗传操作中的变异算子可以使 GA 具有局部的随机搜索能力并且保持群体的多样性. 而 PSO 中, 只有 gbest 或 lbest 给其他粒子传递信息, 这是单向的信息流动, 整个搜索更新过程即是跟随当前最优解的过程. 由于所有的粒子都向最优解的方向飞去, 粒子会趋向同一化, 使得后期收敛速度明显变慢. 当收敛到一定精度时, 无法继续优化, 所能达到的精度比 GA 低.

针对 RIPS 节点定位问题的特点, 本文提出一种改进粒子群优化算法. 为了维持种群的活性, 这里借鉴 GA 中变异的思想, 在每次迭代时对利用式 (3) 和 (4) 更新过的粒子的位置进行如下扰动:

$$\text{for } d = 1, \dots, D \\ \text{if rand.nextBoolean()}$$

$$x_{id}^k = x_{id}^k + \Delta \cdot \text{rand.nextGaussian}() \\ \text{end} \\ \text{end}$$

其中 $\Delta = e(s)$ 或者 $\Delta = \text{rand.nextDouble}()$, 两种情况的概率相等. $\Delta = e(s)$ 可以保证当误差函数值较大时粒子获得较大的扰动; 随着搜索的进行, 误差函数值会越来越小, 粒子获得的扰动也变小, 从而保证算法收敛. 通过对粒子的位置进行微扰, 可以防止粒子趋向同一化, 更充分地搜索整个空间, 即增强了粒子群的全局搜索能力. 当然, 改进 PSO 算法的运算量要比基本 PSO 算法略有增加.

与基于 GA 的定位方法类似, 粒子的位置为解空间的一个解, 对应于 2.2 节中的 s , 粒子的适应值也由式 (1) 决定. $x_{d\max}$ 的取值与基于 GA 的定位方法相同, $x_{d\min} = 0$, $v_{d\max} = x_{d\max}/2$, $v_{d\min} = -x_{d\min}/2$, $D = 3n$. 基于改进 PSO 的定位方法具体步骤如下:

Step 1: 生成初始种群, 群体规模大小为 pSize . 其中, 所有粒子的初始速度均设为 0, 计算所有粒子的适应值; 每个粒子的 pbest 设为其当前位置, pb 为其对应的适应值; gbest 设为种群中具有最小适应值的粒子的位置, gb 为其对应的适应值.

Step 2: 在初始种群中随机选择并得到被选种群, 群体规模大小为 spSize ; 找到被选种群具有最小适应值的粒子, 位置为 lbest , 其对应的适应值为 lb .

Step 3: 被选种群中除 lbest 以外的粒子均按式 (3) 和 (4) 用 pbest 和 lbest 更新速度和位置.

Step 4: 对更新后的粒子位置进行微扰.

Step 5: 计算被选种群中所有粒子的适应值. 如果小于该粒子当前的 pb , 则将 pbest 设为其当前位置, 并更新 pb .

Step 6: 如果被选种群中具有最小适应值的粒子的 pb 小于当前整个种群的 gb , 则将 gbest 设为具有最小适应值的粒子的位置并更新 gb .

Step 7: 检验是否符合结束条件. 如果当前的迭代次数达到了预先设定的最大次数或达到最小错误要求, 则输出最优解; 否则, 转 Step 2.

4 定位算法性能分析

4.1 实验设置

Maroti 等人^[8]采用美国 Crossbow 公司生产的 Mica2 系列微型传感器节点构建了 RIPS 原型系统, 并且进行了实验. RIPS 的软件包括两部分: 在微型传感器节点上运行的 TinyOS 应用程序和在基站上运行的 java 应用程序. 本文利用 Maroti 等人在足球场测量得到的干涉距离^[13]验证并比较两种定位方法.

图 2 为定位实验设置. 其中: 节点 7788, 2562, 7551 和 1124 为锚节点; 待定位节点的数量为 $N = 12$; 干涉距离的数量为 $m = 7258$. 对于基于 GA 的定位方法, $x_{\max} = 100 \text{ m}$, $y_{\max} = 100 \text{ m}$, $z_{\max} = 0 \text{ m}$; 对于基于改进 PSO 的定位方法, $c_1 = c_2 = 3$.

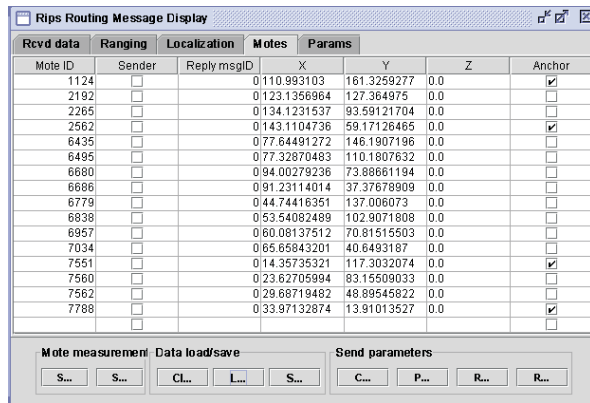


图 2 定位实验设置

4.2 两种定位方法的性能分析

采用的计算机配置为: CPU - Intel Pentium 3 GHz, 内存 1 GB. 为了有效地比较两种优化算法应用于 RIPS 系统时的搜索性能, 定义以下评价指标: 1) 使平均误差 $\text{avg}E$ 小于给定阈值所需要的平均迭代次数 $\text{avg}K$; 2) 使平均误差 $\text{avg}E$ 小于给定阈值所需要的平均耗费时间 $\text{avg}T$ (单位为 ms).

平均误差 $\text{avg}E$ 的定义如下:

$$\text{avg}E = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}, \quad (5)$$

其中 (x_i, y_i, z_i) 和 $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ 分别为待定位节点 i 的真实位置和估计位置.

GA 和 PSO 的优化质量和效率与算法控制参数的设定有关, 需要通过大量的仿真实验来比较各参数所对应的算法性能. 在不同的仿真设置下, 两种定位方法各运行 15 次, 以计算评价指标的平均值. 令 $\text{populationSize} = \text{pSize} = \text{pS}$, $\text{subpopulationSize} = \text{spSize} = \text{spS}$. 表 1 ~ 表 3 分别给出了 $\text{pS} = 20 \& \text{spS} = 10$, $\text{pS} = 100 \& \text{spS} = 10$ 以及 $\text{pS} = 100 \& \text{spS} = 50$ 时, 两种定位方法的性能比较.

从表 1 ~ 表 3 可以看出:

1) 两种定位方法都能在有限的进化代数内找到节点坐标的近似值, 其中基于改进 PSO 的定位方法在

表 1 $\text{pS} = 20 \& \text{spS} = 10$ 时的性能比较

	$\text{avg}E < 10 \text{ m}$		$\text{avg}E < 5 \text{ m}$		$\text{avg}E < 1 \text{ m}$	
	$\text{avg}K$	$\text{avg}T$	$\text{avg}K$	$\text{avg}T$	$\text{avg}K$	$\text{avg}T$
GA	2687	77 949	3 858	111 730	5 591	162 302
MPSO	772	36 198	977	45 720	1 804	85 041
MPSO/GA/%	29	46	25	41	32	52

表 2 $\text{pS} = 100 \& \text{spS} = 10$ 时的性能比较

	$\text{avg}E < 10 \text{ m}$		$\text{avg}E < 5 \text{ m}$		$\text{avg}E < 1 \text{ m}$	
	$\text{avg}K$	$\text{avg}T$	$\text{avg}K$	$\text{avg}T$	$\text{avg}K$	$\text{avg}T$
GA	4 587	140 940	6 984	214 800	10 380	320 330
MPSO	805	43 636	1 213	66 970	2 571	146 380
MPSO/GA/%	18	31	17	31	25	46

表 3 $\text{pS} = 100 \& \text{spS} = 50$ 时的性能比较

	$\text{avg}E < 10 \text{ m}$		$\text{avg}E < 5 \text{ m}$		$\text{avg}E < 1 \text{ m}$	
	$\text{avg}K$	$\text{avg}T$	$\text{avg}K$	$\text{avg}T$	$\text{avg}K$	$\text{avg}T$
GA	1 391	74 317	1 987	107 090	2 987	165 390
MPSO	236	43 669	314	58 565	570	108 170
MPSO/GA /%	17	59	16	55	19	65

较少的进化代数内, 其解群便向最优解的方向收敛, 平均耗费时间远远小于基于 GA 的定位方法. PSO 的优势在于计算简单、易于实现, 且没有很多参数需要调整. 与 PSO 相比, GA 需要确定较多的参数.

2) 在种群大小不变的情况下 (表 2 和表 3), 随着被选种群的增加, 两种定位方法的平均耗费时间都有所减少. 这是因为每一次迭代中参与搜索的个体相应增加了.

3) 在被选种群大小不变的情况下 (表 1 和表 2), 随着种群的增加, 两种定位方法的平均耗费时间都明显增加. 通常种群大小对算法最优解的搜索能力具有决定性的作用, 但这是以时间的增加为代价. 因此应针对具体的应用要求选择最佳的种群大小.

5 结 论

RIPS 中的干涉距离是两个发送节点和两个接收节点间距离的线性组合值. 传统的以两个节点间的距离作为输入的定位算法无法直接利用上述干涉距离计算节点的位置. 基于 GA 的定位方法虽然可以通过搜索求解节点坐标的近似值, 但是计算复杂度高、收敛速度较慢. 本文提出一种基于改进粒子群优化的定位方法, 借鉴 GA 中变异的思想, 在每次迭代时对更新过的粒子位置进行微扰以维持种群的活性. 利用干涉距离的实验数据, 分析比较了两种最优化方法在节点定位问题中的性能, 得出如下结论: 1) 基于 GA 的定位方法和基于改进 PSO 的定位方法, 都能在有限的进化代数内找到节点坐标的近似值, 可见, PSO 算法可以应用于 RIPS 系统的定位问题; 2) 基于改进 PSO 的定位方法的平均耗费时间远远小于基于 GA 的定位方法, 具有更高的优化效率; 3) 适当增加每一次迭代中参与搜索的个体数量, 可以减少两种定位方法的平均耗费时间, 从而提高了优化效率.

参考文献(References)

[1] Liu Y H, Yang Z, Wang X P, et al. Location, localization, and localizability[J]. J of Computer Science and Technology, 2010, 25(2): 274-297.

- [2] Amitangshu Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges[J]. J of Network Protocols and Algorithms, 2010, 2(1): 45-74.
- [3] Li F F, Luo F, Wang J X, et al. An effective self-adapting localization algorithm in wireless sensor networks[J]. J of Applied Mechanics and Materials, 2011, 58-60: 1013-1017.
- [4] Jian L R, Zheng Y, Liu Y H. Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization[C]. Proc of IEEE INFOCOM 2010. San Diego: IEEE Press, 2010: 1-9.
- [5] Kung H T, Lin C K, Lin T H, et al. Localization with snap-inducing shaped residuals: Coping with errors in measurement[C]. Proc of MobiCom 2009. Beijing: ACM Press, 2009: 333-344.
- [6] Li Z, Trappe W, Zhang Y, et al. Robust statistical methods for securing wireless localization in sensor networks[C]. Proc of IPSN 2005. Los Angeles: IEEE Press, 2005: 91-98.
- [7] Zhang Q X, Di Q L, Xu G Y, et al. A RSSI based localization algorithm for multiple mobile robots[C]. Proc of CMCE'10. Changchun: IEEE Press, 2010: 190-193.
- [8] Maroti M, Kusy B, Balogh G, et al. Radio interferometric geolocation[C]. Proc of SenSys'05. San Diego: ACM Press, 2005: 1-12.
- [9] Kusy B, Ledeczi A, Maroti M, et al. Node-density independent localization[C]. Proc of IPSN'06. Nashville: ACM Press, 2006: 441-448.
- [10] 黄艳, 臧传治, 于海斌. 传感器网络中无线电干涉定位系统的多径误差分析[J]. 控制与决策, 2009, 24(2): 231-235.
(Huang Y, Zang C Z, Yu H B. Multipath error analysis of radio interferometric positioning system in wireless sensor networks[J]. Control and Decision, 2009, 24(2): 231-235.)
- [11] Vasconcelos J A, Ramirez J A, Takahashi R H C, et al. Improvements in genetic algorithms[J]. IEEE Trans on Magnetics, 2001, 37(5): 3414-3417.
- [12] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [13] <http://tinyos.cvs.sourceforge.net/tinyos/tinyos-1.x/contrib/vu/tools/java/isis/snest/localization/trips/footballFieldData.zip?view=log>.

(上接第155页)

- [5] Shi Y H, Eberhart R C. A modified particle swarm optimizer[C]. Proc of the IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 1998: 69-73.
- [6] Shi Y H, Eberhart R C. Fuzzy adaptive particle swarm optimization[C]. Proc of the IEEE Conf on Evolutionary Computation. Piscataway: IEEE, 2001: 101-106.
- [7] 张顶学, 关治洪, 刘新芝. 一种动态改变惯性权重的自适应粒子群算法[J]. 控制与决策, 2008, 23(11): 1253-1257.
(Zhang D X, Guan Z H, Liu X Z. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight[J]. Control and Decision, 2008, 23(11): 1253-1257.)
- [8] Zhan Zhi-hui, Zhang Jun, Li Yun, et al. Adaptive particle swarm optimization[J]. IEEE Trans on Systems, Man and Cybernetics, Part B: Cybernetics, 2009, 39(6): 1362-1380.
- [9] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.
- [10] Xin Bin, Chen Jie, Peng Zhi-hong, et al. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization[J]. Science China Information Sciences, 2010, 53(5): 1869-1919.
- [11] Liu Bo, Wang Ling, Jin Yi-hui, et al. Improved particle swarm optimization combined with chaos[J]. Chaos, Solutions and Fractals, 2005, 25(5): 1261-1271.
- [12] Wang Xi-huai, Li Jun-jun. Hybrid particle swarm optimization with simulated annealing[C]. Proc of Int Conf on Machine Learning and Cybernetics. Shanghai: IEEE, 2004: 2402-2405.
- [13] 张顶学, 廖锐全. 一种基于种群速度的自适应粒子群算法[J]. 控制与决策, 2009, 24(8): 1257-1260.
(Zhang D X, Liao R Q. Adaptive particle swarm optimization algorithm based on population velocity[J]. Control and Decision, 2009, 24(8): 1257-1260.)
- [14] Trelea I C. The particle swarm optimization algorithm: Convergence analysis and parameter selection[J]. Information Processing Letters, 2003, 85(6): 317-325.