第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

基于改进蚁群算法的无线传感器网络路由算法的研究

刘剑鸣 赵日记

(燕山大学电气工程学院 秦皇岛 066004)

摘 要 通过对蚁群算法、无线传感器网络及其路由算法的研究学习,根据单个传感器节点智力有限且需通过多个节点协同来完成复杂任务的特点,将其与具有群体智能特点的蚁群算法联系起来,进而提出了基于蚁群算法的无线传感器网络路由算法。在基本的蚁群算法的基础上增加蚂蚁的属性,并将能量、时延和带宽考虑进来,对蚁群算法进行优化,从而提出了基于改进蚁群算法的无线传感器网络路由算法。然后,对算法进行性能分析,发现改进后的算法在时延和网络寿命方面都有较大的提高。

关键词 无线传感器网络,路由算法,路由,蚁群算法

中图法分类号 TN911 文献标识码 A

Research of Wireless Sensor Network Routing Algorithm Based on Improved Ant Colony Algorithm

LIU Jian-ming ZHAO Ri-ji

(College of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China)

Abstract In this paper, through the study of the the ant colony algorithm, wireless sensor network and its routing algorithm, and according to the characteristics that a single sensor node's intelligence is limited, and multiple nodes are needed to accomplish complex tasks synergistically, we put forward the wireless sensor network routing algorithm based on ant colony algorithm with the ant colony algorithm of swarm intelligence characteristics. On the basis of the basic ant colony algorithm, the properties of the ant and the energy were inceased, and time delay and bandwidth were taken into account to optimize the ant colony algorithm. Wireless sensor network routing algorithm based on improved ant colony algorithm was prposed. Then, through the algorithm performance analysis, and found that the improved algorithm in time delay and network life has better improvement.

Keywords Wireless sensor network(WSN), Routing algorithm, Routing, Ant colony algorithm

1 引言

无线传感器网络(Wireless Sensor Network, WSN)由大量传感器节点构成,这些节点可以静止或移动,并且以自组织和多跳的方式构成[1]。其因高度交叉多学科,高度集成多方面知识,而在前沿热点研究领域引发高度关注。它把客观世界的物理信息和传输网络结合在一起,扩展了人们信息获取的能力,在下一代网络中将为人们提供最直接、最有效、最真实的信息[2]。

无线传感器网络的发展起源于越战时期使用的传统传感器系统——“热带树”传感器。“热带树”会将探测器感应出来的目标信息通过无线网络自动传输到服务器中心,从而对信息进行监控和指挥,它由飞机投放到森林中,进入泥土后其无线天线有如树枝而不易被敌人发现。该系统由震动和声响传感器组成,若有车队或人等经过,传感器就会探测出目标产生的震动和声响信息[2]。20 世纪 80 年代至 90 年代,传感器具备了感知能力、通信能力和计算能力,并成为 21 世纪最具影响的 21 项技术之一。21 世纪至今,传感器网络技术注重在网络传输自组织、节点设计低能耗方面的发展,被认为是继互联网之后的第二大网络,具有十分广阔的应用前景,能应用于军事国防、工农业控制等诸多领域[1]。

2 蚁群算法的研究

2.1 蚁群算法的由来

蚁群算法是意大利学者 Dorigo M 等受到自然界中真实蚂蚁觅食行为的启发于 1991 年提出的。到了 1996 年,DorigoM 等更加深入地研究了蚁群算法,这使全球各地的学者和研究机构的视线都聚集至此,促使这一领域的研究站在了国际学术的最前沿[3]。

2.1.1 蚁群的生物行为描述

蚂蚁属于群居物种,分为 4 种蚁型,分别是蚁后、雄蚁、工蚁和兵蚁。不同的蚂蚁有不同的分工,各司其职。地球上的蚂蚁会通过化学通信的方式进行交流,这是由于它自身会分泌一种化学刺激物。蚂蚁的结群协作能力要远比单独个体的能力高智商化,能完成很多大大超过其能力的复杂任务。蚂蚁的群体生物行为包括任务分配、聚类行为和觅食行为等 3

本文受 2012 年河北省自然基金项目(F2012203088)资助。

刘剑鸣(1961—),男,硕士,副教授,主要研究方向为物联网技术、自动化控制、无线传感器控制,E-mail:ppkkkk@126.com;赵日记(1989—),女,硕士生,主要研究方向为物联网技术、无线传感器网络。

种。其中,觅食行为是本文讨论的重点。下面分别介绍:

(1)任务分配。蚁群分工明确,每只蚂蚁各司其职,有的从事繁殖,有的寻找食物,也有的建筑巢穴。蚂蚁对任务的响应类似于现实中的生产调度、动态分配等问题。可以用优先级机制来解释它,优先级越高的任务对蚂蚁的刺激越大,且任务的优先级动态可变。假设某蚂蚁正在执行某任务,其在接到更高优先级的任务即受到一个大于其激活阈值的刺激时,就会停止当前任务而转去执行新任务;其新接到的任务优先级不高于当前任务即受到的刺激较小时,会保持原始状态。但随着时间的推移,不被执行的新任务等待的时间会不断增加,其紧迫程度就会越来越高,从而优先级也会越来越高,当大于当前任务的优先级时就会被执行,类似于进程调度中的动态优先权调度。当新任务被执行后,优先级就会降低,从而不会引起蚂蚁的注意[4]。

(2)聚类行为。蚂蚁的聚类行为有两种体现。其一,蚂蚁能够构造固定地点的墓地,并把分散在各处的蚂蚁尸体堆积起来,该行为被称作构造墓地;其二,蚂蚁会按照蚁卵的大小将其安放在不同的位置。

(3)觅食行为。在蚁巢的周围有食物源,这些食物源的位置分部是没有规律可循的,是随机的。但经过一段时间后,蚂蚁总是能找到一条从蚁巢(源节点)到食物(目的节点)的最短路径。

虽然单只蚂蚁的智能有限,但通过群体的协作就可完成任何蚂蚁个体都不能指挥完成的复杂任务,就像觅食行为一样,通过群体相互协作,蚂蚁就可以找到源和目的之间的最短路径。如图1(a)所示,A点为蚁巢,B点为食物,在A和B之间有一个障碍物。初始时所有蚂蚁从A出发到B会等概率地选择每条路径,如图1(b)所示。蚂蚁在前进过程中会释放信息素,并以信息素作为前进的向导。相同时间内,路径越短遗留的信息素越多,所以,随着时间的推移,选择较短路径的蚂蚁也会越来越多,如图1(c)所示。由此可见,大量蚂蚁组成的群体构成一个信息正反馈机制,某条路径上的蚂蚁越多,则信息素越多,蚂蚁选择该路径的概率也就更大,最终几乎所有蚂蚁都会沿着该最短路径前进,如图1(d)所示。

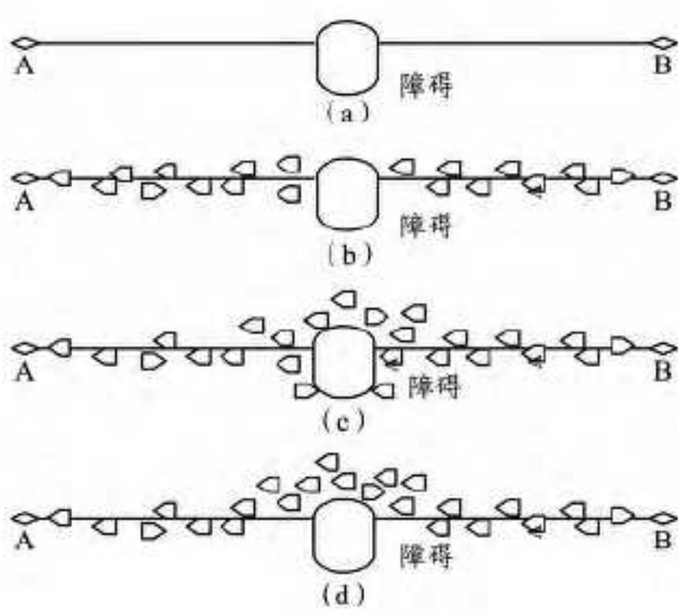

图1 现实中蚂蚁觅食过程

2.1.2 蚁群算法的思想起源

蚁群算法是通过对自然界真实蚁群的"觅食行为"的研究而提出的,所以与真实蚁群的很多观点相同,但也有区别。

对应于真实蚂蚁,蚁群算法中提出了人工蚂蚁。二者存在许多共同点,如下:

(1)交流机制相同。真实蚂蚁会在其所经的路径上释放信息素这种化学物质,人工蚂蚁则以数字信息的变化来模拟,并可以被后来的前向人工蚂蚁和后向蚂蚁所改写。蚁群通过路径上的信息素值或数字信息值来选择下一步的路径。由于蚂蚁释放的这种化学物质会逐渐挥发,因此与人工蚂蚁对应的数字信息也要定期人工挥发。

(2)目的相同。真实蚂蚁和人工蚂蚁都是要寻找一条从蚁巢(源点)到食物(目的地)的最短路径,且两者都必须按部就班,在相邻节点间一步一步地走,不可能跳跃。

(3)路径选择策略相同。真实蚂蚁和人工蚂蚁都只是通过当前可到路径上的信息值根据概率随机选择路径,并没有考虑到从源到目的整条路径,即只考虑了当前局部,没有考虑全局。

2.2 基本蚁群算法的原理

2.2.1 蚁群的搜索原理

真实蚂蚁运动时会释放信息素。对应于真实蚂蚁,人工蚂蚁也有数字信息,相当于人工蚂蚁的信息素,下文就简称信息素,而人工蚂蚁就简称蚂蚁。依靠信息素,蚂蚁可以找到路径。初始蚂蚁随机选择路径,并释放与路径长度成反比的信息素量。后来的蚂蚁走到这里时,就会根据信息素的多少按概率选择路径,信息素越多则被选中的概率越大。虽然单只蚂蚁的智能有限,但通过蚂蚁群体的合作最终就会找出最优路径。下面通过图2来详细说明蚁群的搜索原理。

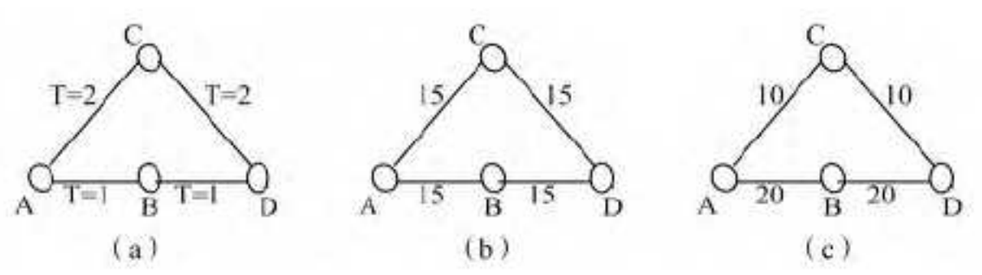

图2 蚁群的搜索原理

如图2所示,A点是源节点(蚁巢),D点是目的节点(食物)。蚂蚁有两条路径ACD和ABD,各点之间的时延如图2(a)所示。假设同一时间有30只蚂蚁从A出发前往D,同时有30只从D出发前往A,蚂蚁在所经路径释放的信息量为1。初始时每条路径的信息素相等,所以蚂蚁以等概率选择两条路径中的任意一条,如图2(b)所示。当$T=2$时路径ABD上的信息素量将是路径ACD的两倍,则此时选择路径AB、BD的将有20只蚂蚁,而选择AC、CD只有10只。随着时间的推移,路径ABD上的信息素会越来越大于路径ACD上的信息素,最终蚂蚁将会完全选择路径ABD。

2.2.2 基本蚁群算法的系统学特征

基本的蚁群算法(Basic Ant Colony Algorithm, BACA)本身可看作一个整体,是一个系统,满足整体大于部分和的条件。蚁群算法还是一种分布式算法,具有分布式计算的特点,从而也具有了强适应的能力。

这里通过图3来说明分布式特点带来的强适应能力。

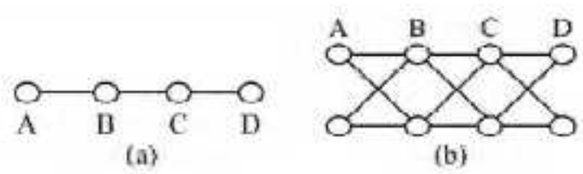

图3 分布式和非分布式对比图

图3(a)是非分布式的,其中ABC 3处任一处出现问题都

会影响全局，得不到想要的结果；而图3(b)是分布式的，ABC出错时不会影响全局，还可以通过其他节点来得到正确的结果。

蚁群算法是分布式的，在搜索从源到目的地的最优路径时，所有蚂蚁都在为这一目的努力着，且并不会因为某一只蚂蚁的缺陷而影响全局。

3 基于基本蚁群的无线传感器网络路由算法的设计与实现

3.1 WRAC算法设计目标

基于蚁群的无线传感器网络路由算法(Wireless Sensor Network Routing Algorithm Based on Basic Ant Colony, WRAC)的目标是设计一个能够解决无线传感器网络简单路由问题的算法，并使得算法的性能尽可能好，同时可以适应网络拓扑的变化。接下来的几节会详细介绍该算法是如何设计的。

3.2 WRAC算法的设计

3.2.1 问题描述

众多的传感器节点构成无线传感器网络，可用平面中的点和线来表示模型。若可直达，则连接两点，否则不连接，且每相同两点之间有且仅有一条边。给予每个节点和每条边相应的与实际情况相同或成比例的信息。该路由算法的目的是寻找一条从源节点到目的节点的较优路径。

3.2.2 算法的设计

具体问题具体分析，蚁群算法对不同类型的问题也往往会有不同之处。

(1)WRAC算法是为了寻找从源到目的的较优解，并不一定经过所有节点，而且一般情况下都不会走过所有的节点，所以初始化时WRAC是将m只蚂蚁放置在源节点处，而TSP是将其分别放在n个节点处。

(2)WRAC在TSP的全局更新的基础上增加了局部更新，扩大了蚂蚁的搜索范围，防止蚂蚁在前一次的最优路径的周围搜索。

(3)在WRAC中蚂蚁分为前向蚂蚁和返回蚂蚁。前向蚂蚁前进的同时记录路径信息并进行局部信息素更新，到达终点后就转变为返回蚂蚁并进行全局信息素更新。

(4)在WRAC中将考虑了带宽，若带宽不符合要求，即便信息素量较大、路径较短，也不予考虑，避免做无用功。

在WRAC算法中状态转移概率的计算规则及信息素的更新是需要分析考虑的重点。下面详细介绍状态转移概率的计算规则和信息素的更新。

(1)状态转移概率。蚂蚁根据路径上的信息量和路径的启发信息来计算转移概率p_{ij}^k(位于节点i的蚂蚁以概率p_{ij}^k选择j为下一跳)，使蚂蚁以较大的概率选择信息素较多且路径较短的节点为下一跳。概率公式如下：

$$p_{ij}^k=\begin{cases}u\\0\end{cases} \tag{1}$$

$u=\dfrac{pow(phe[i][j],alpha)*pow(yita[i][j],betra)}{\sum\limits_{s\in A_k}pow(phe[i][s],alpha)*pow(yita[i][s],betra)}$。$A_k$是蚂蚁$k$下一步所允许通信的节点的集合，若不是所允许的通信节点，则概率为0；$alpha$是信息启发因子，$phe[][]$指边上的信息素，若两节点间没有直接的通路，则信息素为0；$betra$是期望启发因子，$yita[][]$是启发函数，$yita[i][j]=100/d[i][j]$，$d[i][j]$为节点i到节点j的距离。

计算转移概率时，将$yita[][]$考虑进来可以扩大选择的范围，避免过早收敛，适当地延长网络的生存时间。用图4来说明其原因。

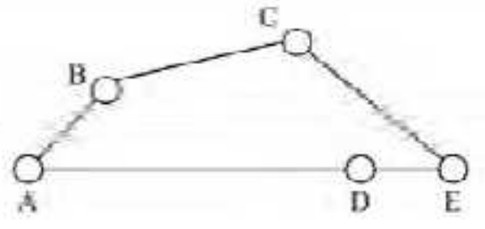

图4 启发函数作用说明辅助图

如图4所示，假设从A点出发前往E点，路径ADE的长度比ABCE的短，且路径越短，全局更新时信息素的增量越大。迭代n(未知数)次后，则AD上的信息素值大于AB上的信息素值。此时若蚂蚁从A点出发，根据信息素值的大小则会以较大的概率选择点D，而选B的概率很小，并且随着搜索的进行选择B的概率会越来越小，这样就很容易导致点D的能量一直消耗，最终成为瓶颈，使得能量消耗严重不平衡；若将$yita[][]$考虑进来，相当于考虑局部的路径长度，而此时AB的路径比AD短很多，则AB上的启发函数值比AD的要大，进而可适量地增大选择点B的概率，从而适量平衡能量的消耗，延长网络的生存时间。

(2)信息素的局部更新。前向蚂蚁在前进过程中对信息素进行局部更新，更新公式如下：

$$phe(i,j)=\max\{phe_{\min},\min\{phe(i,j)+\Delta phe(k,i,j),phe_{\max}\}\} \tag{2}$$

$phe_{\min}$和$phe_{\max}$分别表示信息素的最大值和最小值，$\Delta phe(k,i,j)$是蚂蚁k在本次搜索中在路径(i,j)上释放的信息素量。

(3)信息素的全局更新。后向蚂蚁根据蚂蚁自己记录的信息对信息素进行全局更新，更新公式如下：

$$phe(i,j)=\max\{phe_{\min},\min\{(1-rou)\,phe(i,j)+\Delta phe(k,i,j),phe_{\max}\}\} \tag{3}$$

这里的$\Delta phe(k,i,j)$与局部更新的不同。

$$\Delta phe(k,i,j)=\begin{cases}\dfrac{Q}{L_k^2}\\0\end{cases} \tag{4}$$

L_k是蚂蚁k从源节点到目的节点所经过的路径的长度，Q是常数，路径越短，信息素的增量越大。

3.3 WRAC算法的实现

在实现WRAC算法时，有关信息素的全局更新，可以采用两种方案。方案一，每当有一只蚂蚁到达目的节点后就进行一次全局更新；方案二，当所有蚂蚁都到达目的节点后才进行全局信息素的更新。下面会通过算法的具体实现步骤、程序流程图和伪代码3方面分别对两种方案进行实现和说明，且当出现一个节点的能量耗尽时算法就结束。

3.3.1 WRAC算法方案一的实现

具体实现步骤如下：

(1)初始化。蚂蚁走的路线置空，访问标志均初始化为未访问；将m只蚂蚁放到源节点上，并将源节点设置为已访问，路线的第一个节点为源节点；初始时，所有可直达的节点间的边上的信息素相同，并为常数；初始时，信息素增量为0。

(2)蚂蚁数 $k=0$。

(3)计算转移概率，根据其选择节点 i，将蚂蚁 k 移到 i 处，并将该节点标记为已访问，修改蚂蚁当前所在节点和前一节点的能量。

(4)若蚂蚁 k 已经到达目的节点，即 $end[k]==1$，则计算该蚂蚁的路径长度，进行信息素的全局更新，并初始化蚂蚁 k 的路径和标记信息等。

(5)若出现能量耗尽的节点，则转(7)执行；否则转(6)继续执行。

(6)$k=k+1$，若 $k<$蚂蚁数 m，则转(3)继续执行；否则转(2)继续执行。

(7)循环结束，输出结果。

实现流程如图5所示。

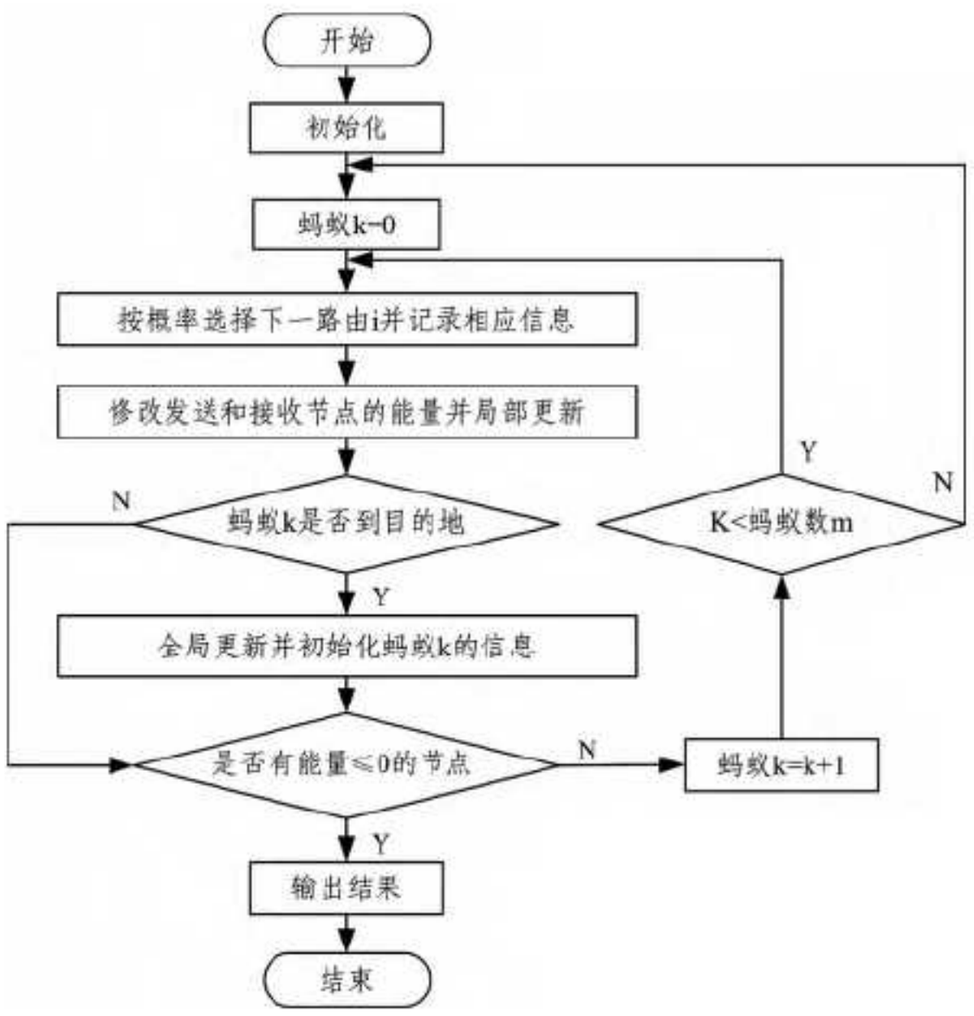

图5 WRAC 算法方案一实现流程

伪代码如下：

```
初始化；
将 m 只蚂蚁放在源点处；
while(没有能量用完的节点时)
{
for(int k=0;k<m;k++) //循环依次用每只蚂蚁搜索
{
  if(该蚂蚁是前向蚂蚁，即 end[k]==0，且其当前所在节点能量符合要求)
  {
    计算状态转移概率并根据其选择下一个节点；
  if(所选节点能量符合要求)
  {
    标记该蚂蚁走过的节点，并记录路径顺序；
  信息素局部更新，原节点能量更新；
  if(原节点能量大于 0)
{
    更新该蚂蚁的寿命，即加上该链路的时延；
    if(所选节点为目的节点)
      该蚂蚁转化为返回蚂蚁；
    else if(若该节点不是目的节点)
    {
      则蚂蚁寿命加上节点时延；
      蚂蚁前进到该节点，并更新该节点能量；
      若能量未达最低要求，则初始化该蚂蚁；
      若能量为 0，则标记循环结束标记；
}
}
}
else if(原节点能量等于 0)
{
  if(下一跳所选节点为目的节点)
      end[k]=1，即标记为已到达目的地；
  flag=1；
}
}
else if(还没到目的地但可直达的节点却都访问过或者可达却没被访问过的节点能量不为 0 但也不够 5)
{
  初始化该蚂蚁；
}
}
if(该蚂蚁为后向蚂蚁，及 end[k]==1)
{
      计算该蚂蚁的路径长度 solution[k]，并记录最短路径；
信息素全局更新；
初始化蚂蚁 k 的信息；
}
if(有能量耗尽的节点) break;
}
}
```

3.3.2 WRAC 算法方案二的实现

具体实现步骤如下：

(1)初始化。初始化循环次数 NC 为0，蚂蚁的访问标记信息为0，访问路径信息为 -1，每条路径的信息素为常数，根据路径长度初始化启发函数值(期望值)。

(2)人工输入源节点和目的节点，并将源节点标记为已访问。

(3)$SDAntnum=0$，即到达目的节点的蚂蚁数置为0。

(4)蚂蚁 $k=0$。

(5)计算状态转移概率，并根据该概率选择 j 为下一跳节点，修改当前节点能量。

(6)将蚂蚁移动到 j，将其标记为已访问，并修改该节点的能量。

(7)若出现能量耗尽的节点，则转(10)执行；否则转(8)。

(8)$k=k+1$，若 $k<$蚂蚁的数目 m，则转(5)执行；否则转(9)。

(9)若 $SDAntnum==m$，即所有的蚂蚁都到达目的节点，则转(10)执行；否则转(4)。

(10)计算到目的地的蚂蚁的路径长度并记录最优路径。

(11)若未出现能量耗尽的节点，则全局更新后转(3)执行；否则转(12)。

(12)循环结束，输出结果。

实现流程如图6所示。

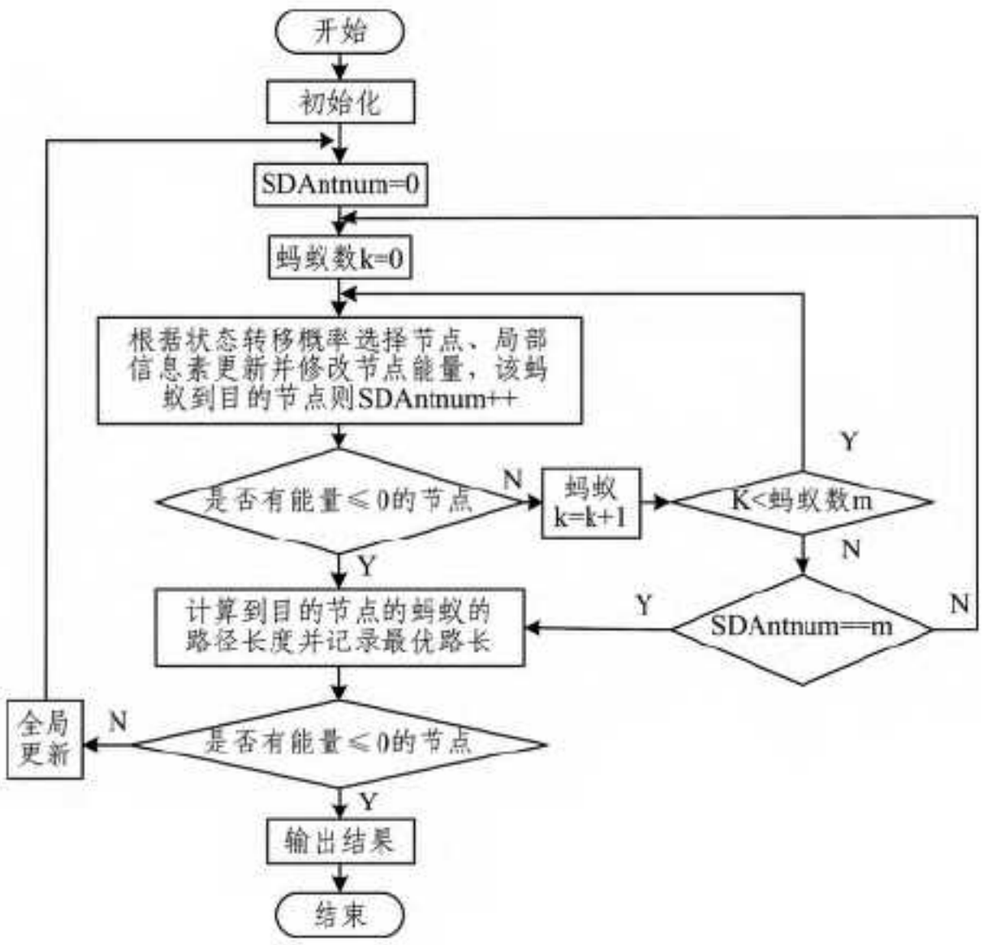

图 6　WRAC 算法方案二实现流程

伪代码如下：

```
初始化；
将 m 只蚂蚁放在源节点处；
while(没有能量耗尽的节点时)
{
  初始化每只蚂蚁的信息；
  while(SDAntnum<m,即有蚂蚁未到目的节点时)
{
  for(int k=0;k<m;k++) //循环依次用每只蚂蚁搜索
{
  if(该蚂蚁是前向蚂蚁,即 end[k]==0,且其当前所在的节点的能量符合要求)
{
  计算转移概率,并根据其选择下一跳节点；
  if(下一跳节点能量符合要求)
{
  记录该节点,并标记为已访问；
  局部信息素更新；
  源节点能量更新；
    if(原节点能量大于 0)
{
      更新该蚂蚁的寿命,即加上该链路的时延；
      if(所选节点为目的节点)
        {
          记录相应信息；
          该蚂蚁转化为返回蚂蚁；
        }
      else if(若该节点不是目的节点)
      {
        则蚂蚁寿命加上节点时延；
        蚂蚁前进到该节点,并更新该节点能量；
        若能量未达最低要求,则初始化该蚂蚁；
        若能量为 0,则标记循环结束标记；
      }
}
else if(原节点能量等于 0)
{
  if(下一跳节点为目的节点)
    end[k]=1,即标记该蚂蚁到达目的节点；
  flag=1,表示有能量耗尽的节点；
}
}
else if(还没到目的地但可直达的节点却都访问过或者可达却没被访问过的节点能量不为 0 但也不够 5)
  初始蚂蚁 k 的信息
}
if(有能量耗尽的节点)
{
  SDAntnum=m；
  break；
}
}
}
对到达目的地的每只蚂蚁,计算其路径长度；
记录最短路径信息；
信息素全局更新；
初始化所有蚂蚁的路径信息和标记信息；
}
```

结束语　本文阐述了无线传感器网络及其路由算法，研究了基本蚁群算法及其在无线传感器网络路由算法中的应用，并在此基础上对基本蚁群算法进行改进，提出了基于改进蚁群算法的无线传感器网络路由算法。本文的主要研究成果如下：

(1)研究了无线传感器网络的特点，并就其节点的特点提出了蚁群算法在其中的应用

(2)研究了蚁群算法的生物由来，并以 TSP 问题为例研究了蚁群算法，最后设计实现了基于 TSP 问题的蚁群算法。

(3)设计并实现了基于基本蚁群的无线传感器网络路由算法，并从时间空间复杂度、能耗、时延和解的情况等角度对其性能进行了分析，本算法整体性能较好。

(4)在 WRAC 的基础上，为了进一步降低时延，给蚂蚁增加了寿命属性，并在信息素全局更新中加入了时延的限制，使得时延越小的路径信息素增加的值尽量越大。能耗、时延和解的情况等方面的性能分析显示，系统的性能得到了适当的提高，时延减少了 7.34%，能耗稍降低，最优解的概率提高了 18.18%。

(5)在 WRAC 的基础上，为了延长网路的寿命，考虑到了能耗的平衡问题，从而在计算状态转移概率时将节点的剩余能量也考虑进来。提出了基于蚁群的能耗均衡无线传感器网络路由算法。改进后的算法适合用于对能耗均衡要求较高的网络。当然，视对能耗均衡要求强度的不同，可以适当改变 pro1 和 pro2 的比例来进行调整。

参 考 文 献

[1] 鲍荣，潘浩，董齐芬，等. 基于信息素扩散模型蚁群算法的无线传感器网络路由研究[J]. 传感技术学报，2011，24：1644-1647

[2] 王翥，王祁，等. 无线传感器网络中继节点布局算法的研究[J]. 物理学报，2012，61(12)

[3] 段海滨. 蚁群算法及其应用[M]. 北京：科学出版社，2007：13-89

[4] 刘乐柱，张季谦，许贵霞，等. 一个修改的混沌蚁群优化算法[J]. 物理学报，2013，62(17)：170501-1-6