OMNET++与NS2在无线传感器网络仿真中的比较研究*)

石为人 黄 河 鲜晓东 许 磊
(重庆大学自动化学院 重庆 400044)

摘 要 OMNET++是一个为大型网络提供开源的、基于组件的、模块化的开放网络仿真平台。针对无线传感器网络的仿真特点,将OMNET++与NS2一些主要性能指标进行比较;在相同仿真条件下,选取报文投递率、运行时间和内存消耗指标在OMNET++与NS2上分别进行定向扩散对比实验,证明OMNET++在无线传感器网络仿真中比NS2的仿真速度快,内存消耗少,是一个优秀的无线传感器网络仿真软件。

关键词 OMNET++,NS2,无线传感器网络,仿真工具

Comparison of OMNET++ and NS2 for WSN Simulation

SHI Wei-ren HUANG He XIAN Xiao-dong XU Lei
(College of Automation, Chongqing University, Chongqing 400044, China)

Abstract Wireless sensor networks have gained considerable attention in the past few years. We present a WSN simulator——OMNET++. Compared with some well-known simulator, OMNET++ has better performance than NS2. We demonstrated the use of the WSN simulation by implementing directed diffusion protocols, and performed performance comparisons (in the execution time and memory used) in simulating WSN in OMNET++ and NS2. The simulation study indicates the WSN in OMNET++ is much more scalable than NS2. It shows that OMNET++ is better than NS2 in large-scale WSN simulation.

Keywords OMNET++, NS2, WSN, Simulation tool

无线传感器网络[1-3](wireless sensor networks,简称WSN)是由部署在监测区域内的大量廉价微型传感器节点组成,通过无线通信方式形成的一个多跳自组织网络系统,其目的是协作地感知、采集和处理网络覆盖地理区域中感知对象的信息,并发布给观察者。它在环境监测、军事、医疗健康、家庭智能监控和其他商业领域有着广泛的应用前景,因此受到了学术界和工业界越来越广泛的关注。

随着无线传感器网络中各种网络方案日趋复杂,网络规模越来越大,掌握网络仿真技术是非常必要的。通过仿真,人们能够在一个可控的环境里研究无线传感器网络,观察由不可预测的干扰和噪声引起的节点间的相互作用,获取节点间的细节,来提高节点投放后的网络成功率,减少投放后的网络维护工作。

为了让研究人员在仿真中达到事半功倍的效果,要求仿真软件满足以下条件:

(1)无线传感器网络节点数量巨大,仿真时间很长,因此需要仿真软件具有方便快捷的追踪能力和调试能力,提高研究人员的工作效率和纠错能力。

(2)无线传感器网络节点大多采用分层结构,因此需要使用大量模块来构建模型,各个模块之间的关系通过分级确定,生成的模块可以重复使用,这样可以减少研究人员的工作量并且降低内存消耗。

(3)通常无线传感器网络需要与其他系统协同工作才能完成预期目标,因此需要自定义的标准组件和开放的数据接口来生成和处理能与其他软件交互的输入输出文件,实现将仿真模型嵌入到大型应用中的功能。这对内存管理、模块可复用能力等提出了更高的要求。

1 OMNET++介绍

OMNET++(Objective Modular Network Testbed in C++的缩写)是一个专门为大型网络提供开源的、基于组件的、模块化的开放网络仿真平台。OMNET++作为离散事件仿真器,具备强大完善的图形界面接口和可嵌入式仿真内核,可运行于多个操作系统平台,简便定义网络拓扑结构,具备方便快捷的编程、调试和跟踪支持等功能。

以下简单介绍OMNET++仿真软件的模型结构和内部构造。

1.1 模型结构

OMNET++[4,5]模型由简单模块和复合模块组成(图1)。简单模块是模块分级中的最小模块,它的主要任务是接收和发送信息。信息传输有门传输和直接传输两种方式,门(gate)发送信息(message)时,门之间通过连接(connection)连在一起,连接属性是可以修改的,包括传播延迟,数据传输速率和误码率。门传输就是通过模块之间的门和连接,按照一定的规则,将信息逐步传输到目的模块,而直接传输则是通过仿真内核直接传输信息到目的模块。

复合模块是由简单模块组合而成,复合模块与简单模块或者其他复合模块组合可生成更高一级的复合模块,这种模

*)基金项目:国家教育部博士点基金项目(20060611010),重庆市自然科学基金资助项目(CSTC 2006BB2191)。石为人 教授,博导,研究领域为智能系统、无线传感器网络及其应用、移动机器人控制;黄 河 硕士研究生,主要从事无线传感器网络方面的研究;鲜晓东 副教授,研究领域为无线传感器网络及其应用、移动机器人控制、电子信息;许 磊 博士研究生,主要从事无线传感器网络方面研究。

块分级没有限制。信息通过复合模块内部的各个简单模块相互协调、运算处理。在传感器网络中有些节点就是由许多简单模块组成的一个复合模块(如图 2(a)),layer0 模块是节点的物理层;Application 模块是节点的应用层;Coordinator 模块负责将外部采集的信息发送到相应的模块进行处理,Energy 模块负责能量的计算,Sensor 模块是传感器节点数据采集板。具体结构通过 NED 语言直接进行描述,也可以使用图形界面进行连接编辑(例如图 2(b)),但最终都将自动转化为拓扑描述语言 NED。

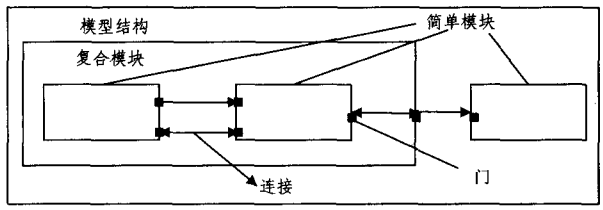
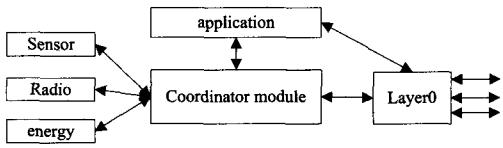
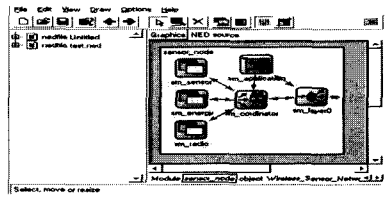


图 1 OMNET++模型结构



(a) 节点内部模块构成图例



(b) 图形编辑界面图例

图 2

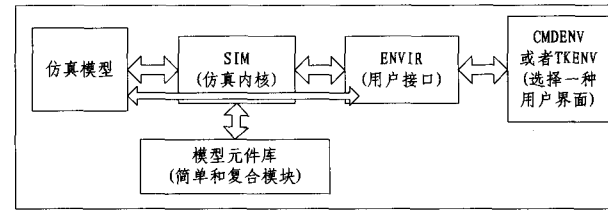
NED 拓扑描述语言是用来定义模型构造(即模块之间和模块内部的连接)的,它包含简单模块定义、复合模块定义和网络定义。

1.2 内部构造

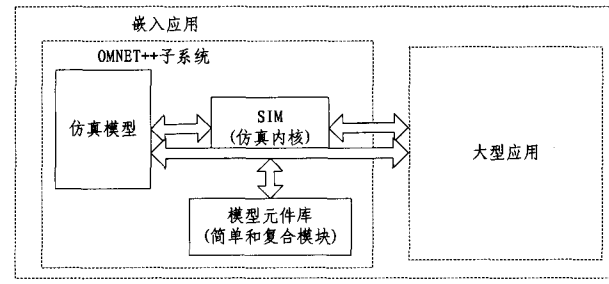
OMNET++运行时的内部构造[6]如图 3(a)所示。

SIM 为仿真内核,它是处理和运行仿真的核心。在 SIM 和用户接口(图 3(a))/大型应用(图 3(b))之间有一个通用接口,使用者可以通过替换用户接口,自定义仿真的运行环境。模型元件库包含了已经编译好的简单和复合模块。仿真模型方框包含的是一些常用的网络协议、应用以及通信模型,随着 OMNET++的广泛应用已经建立起许多可复用的模型和协议。

OMNET++提供了 TKENV 和 CMDENV 两种用户界面。TKENV 是 OMNET++的 GUI(图形用户界面)用户接口,它提供了 3 种工具:动画自动生成、模块输出窗口和对象监测器。动画自动生成工具能够将信息的传递和节点状态的变化等自动地在网络图中显示出来,加快仿真的处理速度。模块输出窗口可以为单独的模块/模块组打开各自的窗口,与传统的 printf()-style 调试比较起来模块输出窗口能够更加容易地观察、追踪程序的处理过程。对象监测器是一个与仿真对象相结合的 GUI 窗口,它能够用合适的方式显示出对象



(a) 内部构造图



(b)嵌入应用时的内部构造图

图 3

的状态或内容(例如对柱状图对象使用柱状图显示),也能对对象进行手动修改。在简单模块中不需要添加任何代码就能自动地监测所有仿真对象,方便调试者的同时提高了运行的速度和可靠性。以上这些功能满足了大型无线传感器网络仿真对调试能力和追踪能力的高要求。

CMDENV 是纯命令行界面,进行批处理仿真时非常有用。

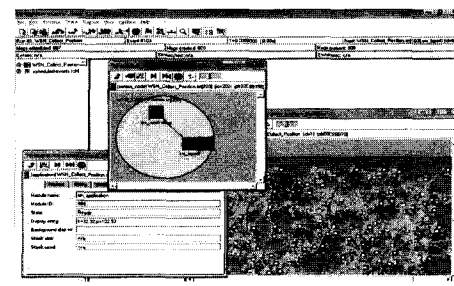


图 4 OMNET++中 TKENV 的用户界面截图

2 与 NS2 性能指标的比较

NS2[7,8]是一种针对网络技术的源代码公开的、免费的软件模拟平台,功能全面,能够应用于各种网络的仿真模拟。因此,大量研究人员使用它进行网络技术的开发,在非商业化的仿真软件中拥有大量的用户,得到学术界广泛认可。下面列出 OMNET++与 NS2 在无线传感器方面的重要性能指标的对比分析。

2.1 可编程能力

OMNET++与 NS2 在可编程能力方面功能都很强大,区别不大。

2.2 模型库和可用模型

NS2 拥有大量的协议模型,但这些模型大多是 TCP/IP 方面的模型。OMNET++不但支持 TCP/IP,SCSI 和 FDDI 等多种协议模型,并且随着用户数量的快速增加,模型库也迅速完善,完全能满足大型传感器网络仿真的需求。

由于 NS 协议模型过分单一,因此 OMNET++在模型

库和可用模型方面有很大优势。

2.3 拓扑结构和分级模型

NS2 使用 Tcl 语言描述网络拓扑结构，可以非常灵活地建立拓扑结构，但 Tcl 语言不能创建图形编辑器，使用起来很不直观和方便，对于初学者更是难以入门。另外，NS2 不能进行模型分级，这极大地限制了它在无线传感器网络仿真中的应用。OMNET++使用 NED 语言和图形编辑器定义拓扑，非常直观，也容易学习，还可以将拓扑参数化，而且 OMNET++允许任意数量的分级模型存在。

在建立拓扑结构和分级模型上，OMNET++比 NS2 更有优势。由于在传感器网络中还没有公认的最优拓扑、网络协议等，经常需要针对不同环境自定义拓扑和协议，因此 NS2 不太灵活的缺点严重影响了网络的仿真。

2.4 编程模型和仿真库提供的功能

现有编程模型有两种，一种是基于协同程序的编程模型；另一种是通过 FSMs 建立信息接收功能的编程模型，NS2 属于后者。OMNET++能够提供这两种编程模型，用户可根据需求随意选择。

仿真库中，相对于 OMNET++而言，NS2 仿真库提供的功能比较少。

2.5 对调试与追踪的支持

高效的调试与追踪能力对于传感器网络仿真是非常重要和必要的。OMNET++通过模块输出窗口、监测器和自动生成动画等 3 个工具来进行调试与追踪，仿真运行时对计算机要求不高，内存消耗小，速度很快。NS2 有自动生成动画效果的功能，但由于缺乏实时的图形环境支持，没有模块输出窗口和对象监测器两种功能。

在对无线传感器网络仿真的调试与追踪上，NS2 运行速度慢，内存消耗大。OMNET++有较好的表现，使用者能够观察实时图形变化，非常生动，而且使用方便。此外，OMNET++也有自己的命令行环境。

2.6 源程序开放性

提供开放的源代码不仅是嵌入和修改仿真引擎所必需的，还能对调试仿真模型提供很大的帮助。OMNET++和 NS2 都是完全开源的。不需要交纳昂贵的费用来获得其使用权，还能在应用中深入了解其内部构造。

综上所述，在无线传感器网络仿真中，NS2 的效果不如 OMNET++。因此，OMNET++更适用于无线传感器网络的仿真。

3 仿真实验结果与分析

本节将对 OMNET++和 NS2 进行仿真对比[9]，从实验数据结果中分析两种软件在无线传感器网络中的性能。

在 OMNET++和 NS2 平台上，对 simpleMAC 和 IEEE 802.11 MAC[10,11]协议下的定向扩散进行仿真对比。选取报文投递率、运行时间和内存消耗指标进行比较。报文投递率指网络中目的节点成功接收的报文数与网络中信源产生的报文数的比值，能够反映网络的数据处理能力，通过同一协议在使用不同仿真软件时的报文投递率对比能够反映仿真软件的仿真能力。运行时间和内存消耗是从时间和空间角度衡量仿真软件在进行仿真时的资源消耗量，反映仿真软件的仿真能力。实验分为 3 个方面，首先是在 simpleMAC 协议下，用 OMNET++与 NS2 分别进行定向扩散仿真[12]，对报文投递率进行对比；然后是在同样环境下，对比运行时间与内存消耗的变化情况；最后验证 IEEE 802.11 MAC 协议下，两种仿真软件在运行时间和内存消耗性能上的差别。通过 simpleMAC 和 IEEE 802.11 MAC 中的对比来验证 OMNET++在无线传感器网络仿真中的优势所在。

3.1 仿真环境设置

OMNET++仿真实验中首先使用 C++编写节点内部的各个简单模块的工作方式，实验使用的节点内部有 layer0 和 Application 两个模块分别对应物理层和应用层，layer0 负责接受和发送数据，Application 负责信息如何选择下跳等控制工作。在 Application 的 initialize ()中进行初始化设置，根据实验需要设置不同的请求节点生成几率，Application 的 handleMessage ()是信息处理部分，就是当信息到达一个节点的时候，该节点应该如何去处理和控制这个信息。然后使用 NED 语言(或使用图形编辑器)编写网络环境。最后，设置 omnetpp.ini 文件中相关的仿真基本信息，运行仿真并在 TKENV 获得仿真运行的实时数据。

NS2 仿真实验中先使用 NS2 自带的 cbrgen 工具来生成传输负载，它可以产生 TCP 数据流或者 CBR 数据流，本文采用 CBR 数据流。然后使用 NS2 自带的 setdest 工具来随机生成无线网络中节点的运动场景，这里的节点都是静止不动状态，即节点停留时间设置为大于仿真时间。最后编写 TCL 仿真代码，加载 CBR 数据流文件和场景文件对路由协议进行仿真，并使用 gawk 语言编写分析代码，对仿真完成后自动生成的 trace 文件进行分析，得到需要的数据。

实验中设置无线传感器网络仿真节点数量分别为 500，750，1000，1250，1500，1750 和 2000 个节点等 7 种情况，传感器区域的大小为 500m×500m，仿真时间为 300 秒，生成请求节点的数量设置为 10 和 100 两种情况。

3.2 simpleMAC 协议中定向扩散的报文投递率对比

simpleMAC 的定向扩散是将 N 个节点随机放置在一个矩形区域当中。设置部分节点生成 REQ 请求信息并发送到其兴趣区域申请数据传输，同时 REQ 属性表记录下第一次发送 REQ 信息时使用的路径。当一个节点接收到 REQ 信息时，它首先检查该信息属性表中是否包括自己的邻居节点。各个节点的属性表保存了邻居节点到目的节点的距离以及邻居节点的能量级。如果节点属性表中有邻居节点的信息时，就会检查这个信息最后的更新时间，如果该时间在设定的时间范围内，这个信息就可以用来确定下跳邻居。如果 REQ 请求信息不包含邻居节点信息或者更新时间超出设定的时间范围，节点就会发送一个 beacon 信标，周围的邻居节点都会收到该信标，这时邻居节点将会回发一个 beacon 反馈信标，用来更新邻居节点信息。然后通过 GEAR 路由协议[13]均衡距离与能量来计算得到下一个节点。REQ 信息到达感兴趣的区域后，会洪泛到区域中的每个节点，并且维持一个可见节点列表来防止链路进入死循环。兴趣区域中的某一个节点收到 REQ 信息就会发送回一个 REP 应答信息到发送 REQ 信息的源节点，REP 信息从节点那里获得 REQ 信息的反向路径信息并沿这个路径回传，当这个探测信息返回到源节点，源节点通过再次发送 REP 信息来增强路径。当再次发送的 REP 信息到达兴趣区域后，兴趣区域中的节点就可以按照 REQ 信息指定的速率向源节点发送数据信息。每隔一段固定时间，发送数据节点就将数据信息作个类似 REP 信息的标记，源节点收到这个带标记的数据后就会再次发送 REP 信息来重建路径，修复路径中出现的空洞。

实验采用 10 个请求生成节点，在设置完仿真环境后将节点随机地分布到传感器区域当中。图 4 是在相同拓扑和仿真环境下总节点数量改变时 OMNET++与 NS2 的报文投递率对比图。

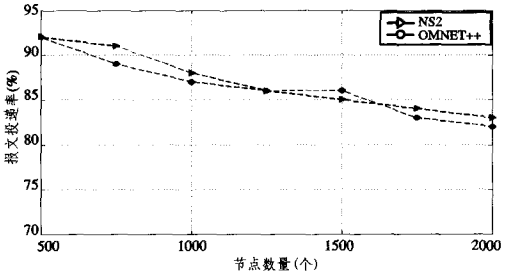


图 5　OMNET++与 NS2 在相同协议下的报文投递率对比

从图 5 中可以看到用 OMNET++与 NS2 仿真时，随着节点数量增加报文投递率降低。两种软件报文投递率非常接近，几乎没有差异。下面从仿真时间和内存消耗两方面进一步测试软件的仿真性能。

3.3　simpleMAC 中定向扩散的仿真时间与内存消耗对比

图 6 中显示的是发出请求节点数量分别为 10，100 时，采用两种仿真软件运行的时间数据。在整个网络节点比较少的时候 NS2 和 OMNET++的仿真运行时间是比较接近的。但是当总节点数量增加后，OMNET++处理通信和 REQ 生成的速度明显比 NS2 快，OMNET++运算 100 个请求节点的运行时间比 NS2 运算 10 个请求节点的运行时间都还要略少。OMNET++运行时间的增长速度也比 NS2 慢很多。当节点数为 1000 时，NS2 运行时间是 OMNET++运行时间的两倍左右，而 2000 个节点时，已经是 3.5 倍左右了。当节点数量在 2000 以上时，NS2 因内存耗尽无法进行更大数量节点的实验，因此实验选用 500～2000 节点作为仿真环境。

实验对比图中的数据表明 OMNET++在 simpleMAC 环境下运算速度明显比 NS2 更快，对计算机的性能要求更低。

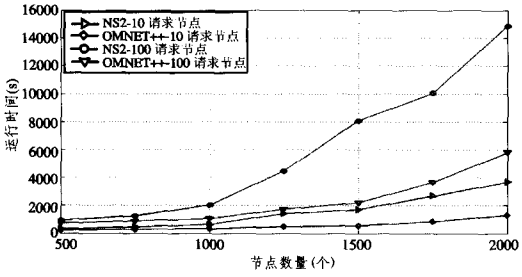


图 6　simpleMAC 下运行耗时对比图

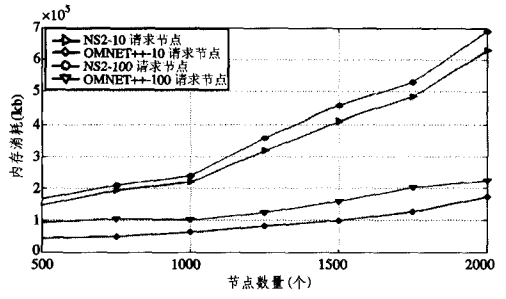


图 7　simpleMAC 下内存消耗对比图

在图 7 中显示的是生成请求节点数为 10 个和 100 个时的内存消耗情况。在节点数量为 500 时 NS2 的内存消耗是 OMNET++的 2 倍左右，节点数量在 1000 以下时 NS2 内存消耗的增长速度比 OMNET++的内存消耗要快一些，但还不是很明显，而节点数量超过 1000，特别是接近 2000 时，其内存消耗的增长速度开始大幅度增加，到 2000 节点时，NS2 的内存消耗量已经是 OMNET++消耗的 3 倍多了。

内存消耗方面也说明 OMNET++的仿真能力更强，而对计算机的性能要求却更低。

3.4　IEEE 802.11 MAC 中定向扩散的仿真时间与内存消耗对比

采用的请求节点同上。图 8 是仿真时间对比图。图形曲线与 simpleMAC 中得出的曲线接近，在节点数量不大时仿真运行时间差别不大，但随着节点数量增加 OMNET++比 NS2 所消耗的时间少很多，节点数量越大差距越明显，证明 OMNET++在 IEEE 802.11 MAC 环境下仍然有运算速度方面的优势。

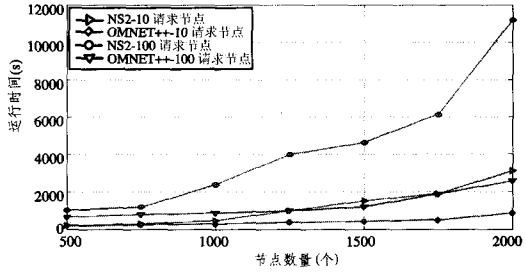


图 8　IEEE 802.11 MAC 下运行耗时对比图

图 9 是仿真结束时的内存消耗对比图。同样地 OMNET++比 NS2 内存消耗要少，随节点数量增加。NS2 内存消耗的增加速度也比 OMNET++快得多。同样反映出了 OMNET++在 IEEE 802.11 MAC 环境下内存消耗方面的优势。

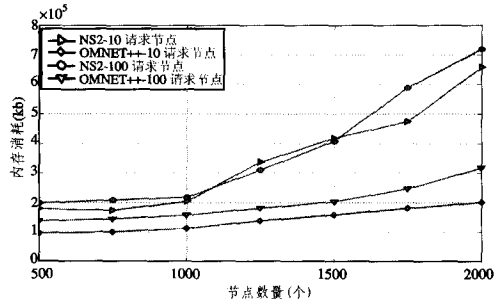


图 9　IEEE 802.11 MAC 下内存消耗对比图

本小节采用 IEEE 802.11b MAC 协议来比较 OMNET++和 NS2 在仿真时间与内存消耗上的差别。虽然实验数据显示的性能差距没有在 simpleMAC 环境下得出的大，但是 2000 节点时运行时间约 3 倍的差距和内存消耗约 2.5 倍的差距仍然足以得出 OMNET++在大型无线传感器网络中的仿真能力比 NS2 更好的结论。

结束语　从本文的研究中可以知道 OMNET++是一个优秀的分布式计算的可视化离散事件仿真软件。在仿真中的性能表现非常出色，其自身特点使得它完全能够满足无线传感器网络的需求。通过与 NS2 的比较，可以发现在无线传感器网络仿真中它比同为非商业化软件的 NS2 性能更好。

OMNET++良好的仿真性能，形象的图形化界面，方便

易懂的拓扑描述语言，以及易上手易操作等特点为相关研究人员和初学无线传感器仿真的研究人员提供了一款方便快捷的仿真工具。随着越来越多的人对 OMNET++的了解与深入研究，OMNET++的功能也必将更加丰富，更加适合在无线传感器网络仿真中的应用。

参 考 文 献

[1] 孙利民，李建中，陈渝，等. 无线传感器网络. 北京：清华大学出版社，2005

[2] Youssef, Mohamed, El-Sheimy, et al. Wireless Sensor Network: Research vs. Reality Design and Deployment Issues// Communication Networks and Services Research, CNSR '07, Fifth Annual Conference. 2007

[3] 任丰原，黄海宁，林闯. 无线传感器网络. 软件学报，2003，14(7)：1282-1291

[4] OMNET++. http://www.omnetpp.org/

[5] Varga A. OMNET + + — Discrete Event Simulation System Version 3.2 User Manual

[6] Varga A. The OMNET++ Discrete Event Simulation System. http://www.omnetpp.org, version 3.0

[7] NS-2. http://www.isi.edu/nsnam/ns/

[8] 徐雷鸣，庞博，赵耀. NS 与网络模拟[M]. 第 1 版. 北京：人民邮电出版社，2003

[9] Tran S P M, Yang T A. OCO: Optimized Communication & Organization for Target Tracking in Wireless Sensor Networks// IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, (SUTC'06). Vol 1 2006: 428-435

[10] IEEE Std. 802.11. Wireless LAN Media Access Control (MAC) and Physical Layer(PHY) Specifications, 1999

[11] 李春时，王光兴. 无线自组织网络中的 IEEE802.11 MAC 协议的研究. 计算机科学，2007，34(11)：26-28

[12] Intanagonwiwat C, Govindan R, Estrin D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks// Proc. ACM Intel Conf. Mobile Comp. and Networking. Aug. 2000

[13] 赵海霞. 无线传感器网络中的安全 GEAR 路由协议. 信息技术，2006(9)：44-48

(上接第 27 页)

(1)测试函数

测试函数从是否约束的角度可分为无约束的测试函数和有约束的测试函数，从 Pareto 前沿的特性可分为凸函数和非凸函数、连续函数和非连续函数、均匀分布函数和不均匀分布函数等。Deb 等人[12]通过仿真实验发现，利用不同的算法求解不同类型的测试函数，所获得的解的优劣性并不是一致的。

(2)遗传参数

遗传参数是另一个影响算法性能的重要因素。研究人员发现，对同一个算法，改变交叉概率、种群规模等遗传参数的设置，算法获得的解的收敛性和多样性会有很大不同。目前，人们在比较各种算法的性能时，都采用统一设置遗传参数的方式。显然，这种方法难以保证所有的算法都发挥出最佳性能。

此外，一些算法中涉及到外部非劣解集规模、共享参数、拥挤距离等概念，如何设置这些参数，还需要进一步研究。

结束语 由于基于 Pareto 最优概念的方法直接处理多个目标，而没有诸多限制，如 Pareto 前端非凸等，基于 Pareto 最优概念的多目标进化算法已成为多目标优化问题研究的主流方向。本文详细介绍了该领域的经典算法，重点阐述了各种算法在种群快速收敛并均匀分布于问题的非劣最优域上所采取的策略，并归纳了算法性能评估中需要进一步研究的几个问题。需要进一步指出的是，已有的研究成果仅仅局限于采用仿真的方式验证算法种群的收敛性和多样性，严格的理论证明还处于空白状态，这也是当前乃至将来一段时期内多目标进化算法研究的难点和热点。

参 考 文 献

[1] Chankong V, Haimes Y Y. Multiobjective decision making theory and methodology [M]. New York: North-Holland, 1983

[2] Hans A E. Multicriteria optimization for highly accurate systems [M]. New York: plenum press, 1988

[3] Srinivas N, Deb K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms [J]. Evolutionary Computation, 1995, 2 (3): 221-248

[4] Rosenberg R S. Simulation of Genetic Populations with Biochemical Properties [D]. Michigan: University of Michigan, 1967

[5] Schaffer J D. Some experiments in machine learning using vector evaluated genetic algorithms [D]. Tennessee: Vanderbilt University Electrical Engineering, 1984

[6] Richardson J T, Palmer M R, Liepins G, et al// Proc. Third Int. Conf. on Genetic Algorithms [C]. Morgan Kaufmann, 1989

[7] Fonseca C M, Fleming P J. Genetic Algorithm// Proceedings of the Fifth International Conference [C]. San Mateo, 1993

[8] Horn J, Nafpliotis N. Proceedings of the First. IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence [C]. Piscataway, NJ: IEEE Service Center, 1994

[9] Knowles J, Corne D. Proceedings of the 1999 Congress on Evolutionary Computation [C]. Piscataway, NJ: IEEE Press, 1999

[10] Zitzler E, Thiele L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271

[11] Zitzler E, Laumanns M, Thiele L. SPEA 2: Improving the Strength Pareto Evolutionary Algorithm [J]. 2001

[12] Deb K, Prata PA, Agarwal S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6: 182-197

[13] Shin S-Y, Lee I-H, Kim D, et al. Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing [J]. IEEE Transactions on Evolutionary Computation, 2005, 9 (2): 143-158

[14] Osman M S, Abo-Sinna M A, Mousa A A. IT-CEMOP: An iterative co-evolutionary algorithm for multiobjective optimization problem with nonlinear constraints [J]. Applied Mathematics and Computation, 2006, 183: 373-389

[15] Deb K, Goldberg D E. Proc. Third Int. Conf. on Genetic Algorithms [C]. Morgan Kaufmann, 1989

[16] Goldberg D E, Richardson J. Proc. Second Int. Conf. on Genetic Algorithms [C]. Lawrence Erlbaum, 1987

[17] Goldberg D E, Deb K. A Comparison of Selection Schemes used in Genetic Algorithms [M]. 1991

[18] Srinivas N. Multiobjective optimization using nondominated sorting in genetic algorithms [D]. Kanpur: Indian Institute of Technology, 1994

[19] Deb K. Genetic algorithms in multimodal function optimization [D]. Tuscaloosa: University of Alabama, 1989

[20] Mann J W, Smith G D. A Comparison of Heuristics for Telecommunications Traffic Routing [M]. 1996

[21] Laumanns M, Thiele L, Deb K, et al. Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization [J]. Evolutionary Computation, 2002, 10 (3)

[22] Deb K. Multiobjective Optimization Using Evolutionary Algorithms [M]. U. K: Chichester, 2001

[23] Zitzler E. Evolutionary algorithms for multiobjective optimization: Methods and applications [D]. Zurich: Swiss Federal Institute of Technology(ETH), 1999