基于 Qt 的软 PLC 梯形图编辑软件的设计与实现①

陈　雪[1,2]，刘荫忠[2]，徐恩松[1,2]

[1](中国科学院研究生院，北京 100049)
[2](中国科学院沈阳计算技术研究所，沈阳 110168)

摘　要：本文重点是软 PLC 梯形图编辑软件的设计与实现。采用跨平台能力很强的 Qt 作为开发工具，能够将系统应用于多种操作平台，使系统具有更强的开放性；在对梯形图整体结构进行深入分析的基础上，利用面向对象的方法，对系统进行了类层次结构的设计，提高了软件开发的执行效率；采用双层双向链表存储梯形图，使得动态编辑操作更加方便、灵活；通过顺序扫描梯形图程序，将源语言转换为指令表程序，此方法使得转换过程相对简单。最后对软件进行测试，验证了设计方法的正确性与可用性。

关键词：Qt；梯形图；指令表；类层次；存储结构；

Design and Implementation of SoftPLC Ladder Diagram Editing Software Based on Qt

CHEN Xue[1,2], LIU Yin-Zhong[2], XU En-Song[1,2]

[1](Graduate University, Chinese Academy of Sciences, Beijing 100049, China)
[2](Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: This paper mainly introduces the design and performance of SoftPLC Ladder Diagram editing software. Adoppting Qt as the developing tool which applied in multi-operating system, so the system is more opening. Based on in-depth analysis of the ladder diagram's overall structure, the systerm's class hierarchy is designed using object-oriented method, which enhances the efficiency of the software. Applying the storage structure of double layer and double linked list makes the dynamic editing operations more flexible and convenient. The source language converted to instruction list through sequential scanning the ladder diagram makes the process simpler. At last, the experimental results in testing the software show that the design method is accurate and feasible.

Key words: Qt; ladder diagram; instruction list; class hierarchy; storage structure

1　引言

可编程逻辑控制器 PLC，是利用来自输入输出装置的信号及存储的程序，来控制机械或程序的动作[1,2]。随着计算机技术的快速发展和广泛应用，传统 PLC 逐渐表现出其弊端，这严重制约了传统 PLC 的发展，随着计算机技术的快速发展及工业自动化领域 IEC61131-3 国际标准[3]的推出和实施，用软件实现传统 PLC 功能的软 PLC 技术应运而生，并得到广泛发展。

梯形图被称为 PLC 的第一编程语言，凭借其直观易学的优点，成为了最广泛的编程语言。梯形图采用图形语言，沿用了继电器的触点、线圈、串并联等术语和图形符号[4]，在计算机上和控制技术上称为“面向生产过程的语言”。

目前，已开发出很多软 PLC 梯形图编辑软件，大部分编辑软件在 Windows 操作系统下，采用 MFC 开发工具进行开发设计。在此平台下开发的软件只能应用于单一的 Windows 操作系统。针对此问题，本文采用跨平台能力很强的 Qt 作为开发工具，从而使开发软件能够应用于大部分操作平台。

将编辑软件的设计分为两个模块，编辑模块和转换模块。编辑模块主要实现梯形图的绘制、编辑、程

① 基金项目:“高档数控机床与基础制造装备”国家科技重大专项(2011ZX04016-071)
收稿时间:2011-04-03;收到修改稿时间:2011-04-28

序的存储和加载等功能；转换模块主要完成梯形图程序的语法逻辑检查、梯形图转换指令表等工作。

2 编辑模块

Qt 的 QMainWindow 类提供了一个应用程序主窗口，包括菜单栏、工具栏、中心部件和状态栏等。在本系统的设计过程中，主框架继承自 QMainWindow，这为以后的设计工作带来了很多方便之处。

中间的编辑视图区不仅要绘制梯形图程序，还需要将转换后的指令表程序显现出来。因此，在主窗口的中央区域采用了多文档界面（Multiple Document Interface, MDI）应用程序，或者称为 MDI 应用程序。在本编程软件的设计过程中，将 QMdiArea 类作为中央窗口部件，通过让每一个窗口都作为这个 QMdiArea 的子窗口部件，实现多文档界面设计。

2.1 类层次结构设计

2.1.1 系统类的设计

梯形图是由一个个像素组成的位图，计算机没有分析梯形图程序，以及判断每个元素间逻辑关系的能力，因此，需要找到一种计算机可以“读懂”程序的描述方法。软件设计过程中，采用面向对象的设计方法。根据梯形图中对象的性质和功能将其抽象归并为四大类，分别为梯形图类 LadLadder、梯级类 LadRung、梯级行类 LadLine 和图元类 LadComponent，如图 1 所示。

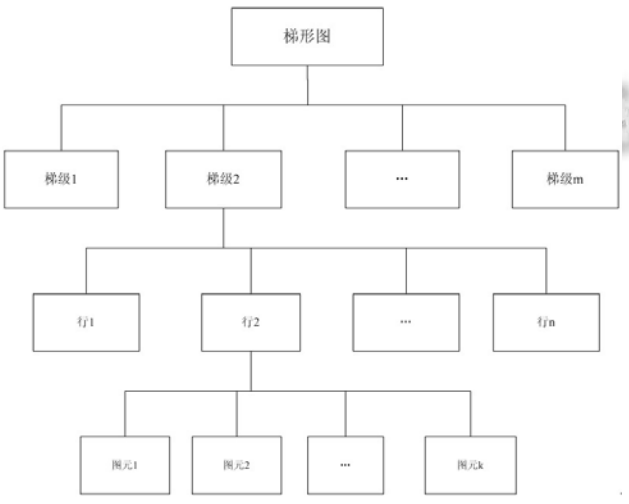

图 1 梯形图结构图

2.1.2 图元对象的设计

在设计过程中，为了减少代码的书写量，提高执行效率，为每一个图元建立一个类，根据图元类生成实例对象，将这些对象组合起来就构成了一个梯形图程序。

根据梯形图图元的复杂程度不同，将梯形图图元分为两类：基本图元和功能指令，并相应抽象出两个基类 GraphElement 和 GraphFB，所有图元都继承自这两个基类。下面分别对图元基类和图元派生类进行设计。

(1) 图元基类的设计

图元基类分成两部分：一类为基本图元基类，另一类为功能指令基类。下面分别介绍对这两个基类的设计方法。

①基本图元基类

基本图元结构简单，是在设计顺序程序时最常用的指令，它们进行执行一个位运算的操作。包括常开触点、常闭触点、（取反）输出线圈等，从这些图元中抽象出一个基类 GraphElement，定义了其共有的属性和操作接口。基本图元基类定义如下：

```
class GraphElement{
public:
GraphElement(QPixmap *pix1);
QString address;        //对应的输入地址
QPoint point;           //所画图元的起始坐标
QPixmap *pix2;          //用作绘图设备
virtual ~GraphElement();
protected:
virtual void draw();};  //画图函数
```

可见，GraphElement 类抽象出了此类图元的共有属性，包括输入地址和坐标等信息，共有方法为画图函数。

②功能指令基类

在绘制程序时，有些功能很难用执行位运算的基本图元来实现，此时用相对复杂的功能指令编程会更方便。功能指令包括二进制加（减）、逻辑与、计数器、计时器等。功能指令基类 GraphFB 定义如下为：

```
class GraphFB{
public:
GraphFB(QPixmap *pix1);
int FBtype;             //用来标识不同的功能指令
QPoint point;           //绘制图元的起点坐标
QPixmap *pix2;          //用作绘图设备
```

```
virtual ~GraphFB();
protected:
virtual void draw();};
```

(2) 图元派生类的设计

图元基类作为父类，封装了此类图元共有的属性和方法，根据每个图元的不同特性，从基类派生出各个图元子类，从而封装了每种图元对象所特有的属性，描述图元除了各种属性之外还有一些成员函数。有些方法是所有图元共有的，但具体的实现方法不同，在基类中将这样的函数声明为虚函数，在各个图元派生类中进行重载。例如在屏幕上绘图函数 void draw()，由于每个图元的图符显示不同，因此需要将此函数定义为虚函数，在每个派生类中进行重载。此方法充分体现了面向对象语言中接口和实现分离的思想。

2.2 梯形图的存储结构

梯形图的编辑过程是一个动态存储的过程，本文采用链表的数据结构存储梯形图来表述这种动态过程。梯形图由一个或若干个相互独立的梯级组成，每个梯级由一行或若干行组成。每一行又由很多图元组合而成。因此，梯形图梯级的存储结构选择采用两个层次的双向链表结构。梯形图梯级的存储结构如图 2 所示，下面分别对图元和行头结点设计数据结构。

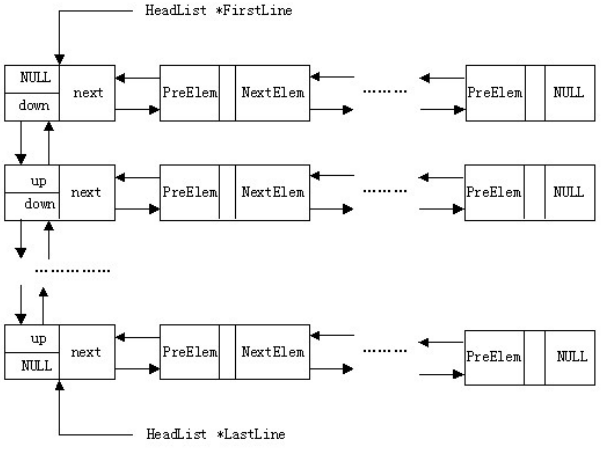

图 2　梯形图梯级存储结构图

(1) 图元的数据结构

为了方便梯形图编程，降低存储不同图元的复杂度，在设计过程中，采用统一的数据结构存储梯形图图元。图元的数据结构定义如下：

```
typedef struct Element
{
QPoint point;          //图元所在的坐标
QString address;       //图元对应的地址
int GraphType;         //图元的类型
int FunInsType;        //标识具体的功能指令
……
Element *PreElem; //指向前一个图形组件的指针
Element *NextElem;//指向后一个图形组件的指针
} Element , *LineNode;
```

定义的图元数据结构清晰地反映图元的信息，包括图元的类型，图元对应的地址，输入域信息，以及前向图元指针和后项图元指针，利用这两个指针能够将一行中的所有图元都连接起来。

(2) 行头结点的数据结构

行头结点的数据结构定义如下：

```
typedef struct HeadNode
{
QPoint point;
HeadNode *up;
HeadNode *down;
Element *next;
}HeadNode, *HeadList;
```

行头结点仅仅是为了方便连接行与行而定义的抽象结点，并不是实际存在的图元。数据结构包括行所在的当前位置，指向其上和其下的行头结点指针，以及其后的第一个图元指针。从而能够将行与行，行头结点与此行第一个图元的逻辑关系清晰准确的反映出来。

2.3 绘制与编辑功能

该软件为用户提供一个具有绘制与编辑功能的梯形图编程环境。设计过程中，使用一个行头结点 HeadList 的全局对象 FastLadder 作为梯形图存储结构的索引，通过全局对象上、下、左、右扫描可以得到一个完整的梯形图存储结构。

Qt 中的二维图形引擎是基于 QPainter 类的。要想在绘图设备上绘图，只需创建一个 QPainter，再将指针传到该设备上[5]，而在 Qt 的默认情况下要使用 QPainter 画图必须在 paintEvent()函数里完成。paintEvent()是系统自带的函数，可以在类中重载以响应 paint 事件。系统封装的 update()函数可以自动调用 paintEvent()函数。因此，在实现梯形图绘制或编辑功

能时，需要在其对应的函数体中调用 update()函数，update()函数会自动调用 paintEvent()函数，从而能够完成将梯形图显示在绘制区域上的操作。

2.4 文件的存储和加载

在梯形图编程操作的过程中，可能会再次使用已编辑好的梯形图程序，若重新绘制，不仅会给用户带来很多麻烦，而且会浪费大量时间。因此为了操作方便，需要使用文件将梯形图程序储存并加载。在本系统设计过程中，创建了一种以“.lad”为后缀名的的文件，专门用来存储梯形图程序。存储文件时，需要将该文件所需的信息全部保存起来。加载文件时，把已储存在文件里的程序读出来，再根据程序的数据重建相对应的图元对象，并识别图元对象之间的逻辑关系，从而能够将存储的程序重新绘制到编辑区域中。

梯形图语言是一个图形化编程语言，不能作为文本文件处理，于是采用将梯形图程序存储为二进制文件的形式。QDataStream 类提供了将二进制文件串行化的功能，实现了 C++的基本数据类型串行化[6]。

在存储过程中，从梯形图存储结构中的全局对象行头结点 FastLadder 开始，对每个行头结点信息以及其连接的图元信息序列化。从而能够将梯形图程序完整且准确的存储起来。当进行文件加载操作时，通过定义一个对象用来从文件中读出数据，利用新定义的数据结构存储读出的数据信息，从而能够恢复梯形图链表。

3 转换模块

3.1 梯形图语法逻辑检查

在进行梯形图的绘制和编辑过程中，不可避免会遇到一些违背编程规则的情况。例如，短路、断路等语法逻辑错误。因此，需要对梯形图进行全面检查，确定准确无误后，才能将梯形图程序转换为对应的指令表程序。

短路是指两个垂直连接线之间只有水平连接线，而没有任何其他图元出现。在检查时，只需顺次扫描梯形图程序，先找到两并联垂线，再判断两者之间是否有直接相连的情况。可以通过记录两并联垂线间的图元个数得出结论，若图元个数为零则表明出现短路；断路是指某一个干路或支路上缺少图元，可以通过扫描梯形图程序并记录下扫描到的图元，扫描结束后若发现还有未扫描到的图元就判定为断路。

除此两种情况以外，系统还可能出现其它语法逻辑错误，此处不再一一列举。

3.2 梯形图转换指令表

指令表称为语句表或布尔助记符，是一种用助记符表示用户逻辑程序的文本化语言。与梯形图相比，指令表更接近于机器语言，更易于被转换为软 PLC 运行系统所识别的目标代码。IEC61131-3 国际标准规定的其它四种编程语言，都可以转换为指令表语言。因此，若能实现对指令表语言的编译，即可实现对其它编程语言的扩展编译。

3.2.1 转换流程

梯形图向语句表的转换方法有很多种，例如，葛芬等[7]将梯形图映射为 AOV 图，并由其建立二叉树来表示指令间的逻辑关系，通过遍历二叉树实现 PLC 梯形图与指令表的转换。

每个梯形图程序由若干个梯级组成，本文采用以梯级为单位，按照从上到下，从左到右的顺序进行，与以上提出的转化方法比较相对简单。对于没有并联支路的梯级，只要根据梯形图图元在梯级中的位置和图元类型，即可将梯形图图元信息转换成对应的指令表语言。对于包含并联支路的梯级，转换过程相对复杂。梯形图程序转换过程如下所述。

(1)扫描行头结点，找到此行的第一个垂线（即左母线），判断左母线之后的垂线是否为 NULL。若为 NULL，转向 2），否则转向 3）。

(2)若为 NULL，说明此行为串联模块，根据图符或者线圈的逻辑关系，按照从左到右的顺序进行转化。判断下一行头结点是否为 NULL，若不为 NULL，转向 1），若为 NULL，转向 4）。

(3)说明此梯级有并联模块，确定当前的并联模块，判断此模块与前一个并联模块之间是否有图元相连（第一个并联模块除外），若有图元相连，先转换模块间相连的图元，再转换并联模块。判断此模块后面是否还有垂线，若有垂线继续操作 3）。若没有垂线，判断下一行头结点是否为 NULL，若不为 NULL，转向 1），若为 NULL，转向 4）。

(4)梯形图转换结束。

按照此方法，对每个梯形图图元的转化既不会重复也不会遗漏，可以实现将每个梯形图梯级准确的转化为对应的指令表程序段。梯形图转换流程图如图 3 所示。

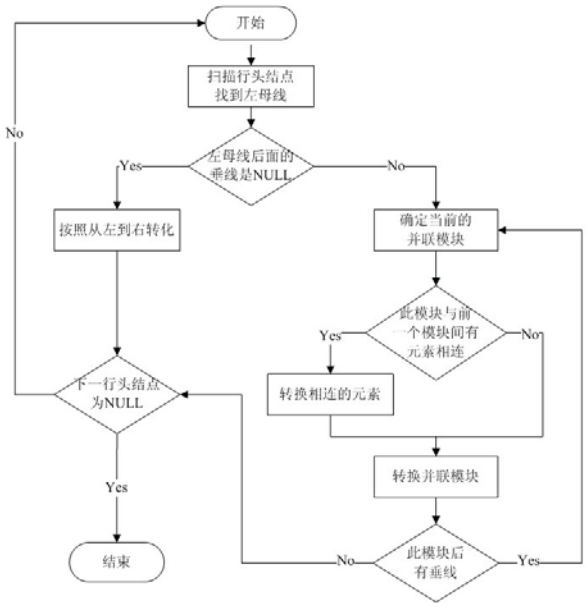

图 3　梯形图转换流程图

3.2.2 转换后的指令表形式

由于一个梯形图程序可能包含一个或多个梯级，因此转换后的指令可能包含多个程序段，每个程序段的构成形式如图 4 所示，即程序段以关键字 PROGRAM 开始，后面为程序段名、变量定义段、指令列表，最后以 END_PROGRAM 结束。

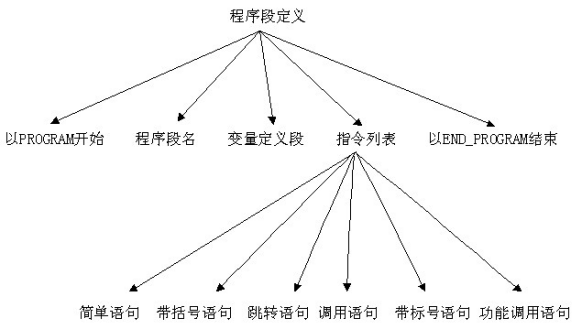

图 4　指令表程序段的构成形式

4　实验测试

编辑一个用于实现进给轴控制功能的简单梯形图程序，如图 5 所示。

以图 5 所示的梯形图程序为例，对其进行修改以破坏程序的正确性，修改后的程序在第三行中出现断路错误，第七行出现短路错误。对其进行语法逻辑检查，检查结果如图 6 和图 7 所示。

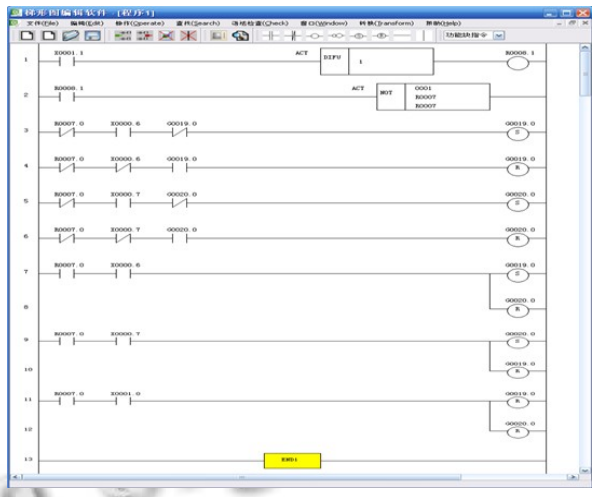

图 5　指梯形图编辑程序实例

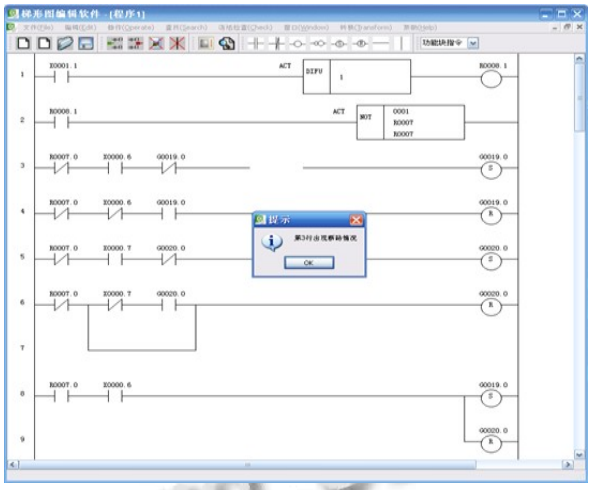

图 6　语法逻辑检查结果(1)

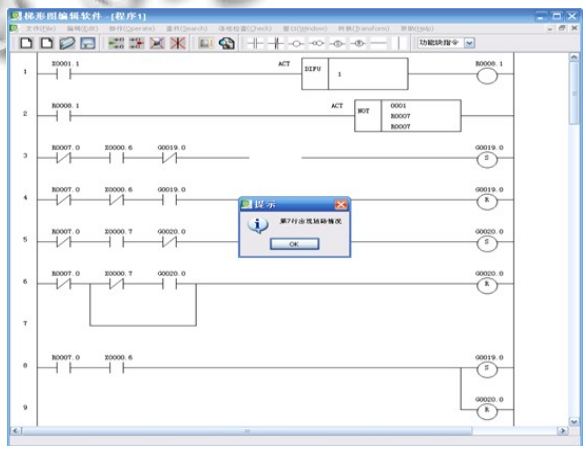

图 7　语法逻辑检查结果(2)

下面对梯形图转换成指令表的功能进行测试，仍以图 5 的程序为例，其转换后的指令表如图 8 所示。

通过实验测试，验证了本梯形图编辑软件设计的正确性与可用性。

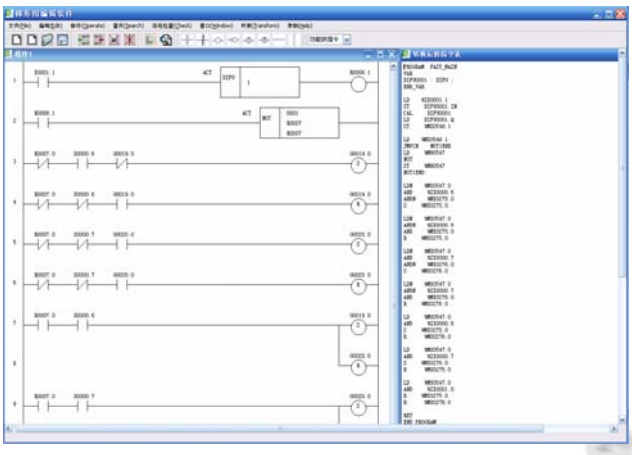

图 8 梯形图程序（图 5）及转换后的指令表程序

5 结语

基于 Qt 开发工具，设计了一款直观、方便、开放、高效的梯形图编辑软件，具有编辑和转换功能。经过测试，该软件的绘制、编辑操作方便、灵活，并能对梯形图程序进行语法逻辑检查，将其正确地转换为指令表程序。

参考文献

1 王爱玲,张吉堂,吴雁.现代数控原理及控制系统.北京:国防工业出版社,2002.

2 刘晓玲,董平.数控技术发展新动向.装备制造技术,2008,(4):116–119.

3 John KH, Tiegelkamp M. IEC61131-3. Programming Industrial Automation Systems. Germany: Springer-Verlag Company, 2001.

4 蒲志新,熊永超,熊晓红.PLC 梯形图语言编辑功能的软件实现.机械,2003,30(3):54–55.

5 Blanchette J, Summerfield M. C++ GUI Programming with Qt4. New York: Prentice Hall, 2008:138.

6 蔡志明,卢传富,李立夏等.精通 Qt4 编程.北京:电子工业出版社,2009.219–222.

7 葛芬,吴宁.基于 AOV 图及二叉树的梯形图及指令表互换算法.南京航空航天大学学报,2006,38(6):754–758.

（上接第 59 页）

Proceedings of the IEEE International Green Computing Conference (IGCC).Chicago:IEEE,2010:357–364.

11 Wang Y, Wang X. Power Optimization with Performance Assurance for Multi-tier Applications in Virtualized Data Centers. 39th International Conference on Parallel Processing Workshops.San Diego: IEEE,2010:1–8.

12 Srikantaiah S, Kansal A. Zhao F. Energy Aware Consolidation for Cloud Computing. HotPower'08, Dec 2008.

13 Dhiman G, Marchetti G, Rosing T. vGreen: a System for Energy Efficient Computing in Virtualized Environments. Proceedings of International Symposium on Low Power Electronics and Design(ISLPED), 2009.

14 Li B, et al. EnaCloud: An Energy-Saving Application Live Placement Approach for Cloud Computing Environments. Proc. IEEE Int. Conf. on Cloud Computing (CLOUD'09), 2009,17–24.

15 Cardosa M, Korupolu M, Singh A, Shares and utilities based power consolidation in virtualized server environments. Proc. Of IFIP/IEEE Integrated Network Management 2009, 2009.

16 Goiri I, et al. Energy-aware Scheduling in Virtualized Datacenters. Proceedings of the 12th IEEE International Conference on Cluster Computing (Cluster 2010), Heraklion, Crete, Greece, September 20–24, 2010.

17 Liu L, Wang H, Liu X, Jin X, He WB, Wang QB, Chen Y. GreenCloud: a new architecture for green data center, Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session, Barcelona, Spain: ACM, 2009,29–38.

18 Dorigo M, Stutzle T. 张军,胡晓敏,罗旭耀等译.蚁群优化.北京:清华大学出版社,2006.