项目批准号	61004089
申请代码	F030120
归口管理部门	
收件日期	

国家自然科学基金
资助项目进展报告

资助类别： 青年科学基金项目

亚类说明：

附注说明：

项目名称： 定性定量结合的并行智能优化仿真方法的研究与应用

负 责 人： 李妮 电话： 010-82338375

电子邮件 ： lini@buaa.edu.cn

依托单位： 北京航空航天大学

联 系 人： 杜润秋 电话： 82338500

资助金额： 20.0000(万元) 累计拨款： 20.0000 (万元)

执行年限： 2011.01-2013.12

填表日期：2011 年 12 月 10 日

国家自然科学基金委员会制（2004 年 11 月）

关于填报《国家自然科学基金资助项目进展报告》的说明

一. 项目负责人每年须填报《国家自然科学基金资助项目进展报告》(简称《进展报告》)，以此作为自然科学基金资助项目跟踪、管理的主要依据。

二. 项目负责人应认真阅读自然科学基金项目管理和财务管理有关规定、办法（查阅 http://www.nsfc.gov.cn），在年度工作的基础上，实事求是地撰写《进展报告》。

三. 项目依托单位认真审核，于每年 1 月 15 日前将本单位受资助项目的《进展报告》统一报送国家自然科学基金委员会归口管理部门。

四. 《进展报告》由报告正文和附件两部分组成，报告正文请参照 “《进展报告》报告正文撰写提纲”撰写，并可根据需要增设栏目，要求层次分明，内容准确。项目执行过程中的进展或研究成果、计划调整情况等，须在报告中如实反映。

五. 国家自然科学基金委员会归口管理部门负责审核项目年度《进展报告》、跟踪项目进展与研究成果、核准项目负责人的次年度研究计划和调整要求，确定项目继续资助的情况。对不按要求填报《进展报告》，或项目执行不力，或内容、人员等调整不当而影响项目顺利进展的，视其情节轻重要求负责人和依托单位及时纠正，或给予缓拨资助经费、中止或撤消项目等处理。

六. 《经费执行情况报表》由重大项目的课题和重点项目填报，重大项目每年度填报随进展报告一同报送，重点项目在进行中期检查的年度填报。其他项目无需填报《经费执行情况报表》，只需在《进展报告》中对经费使用情况和下一年度经费安排做出必要的说明。

注：国家自然科学基金强调科学道德和良好的学风，反对弄虚作假和浮躁作风，要求工作认真、填报材料实事求是。部分探索性研究内容，虽经过努力，也可能没获得理想结果或甚至失败，特别是面上项目。如有这种情况，也请在报告中实事求是地反映出来，说明工作状况和发展态势，供国家自然科学基金委员会和专家参考。

报 告 正 文 撰 写 提 纲

1. **年度计划要点和调整情况**。简要说明是否按计划进行，哪些研究内容根据国内外研究发展状况及项目进展情况做了必要的调整和变动，哪些研究内容未按计划进行，原因何在。

2. **研究工作主要进展和阶段性成果**。(本部分是进展报告的重要部分,请认真撰写)。请分层次叙述所开展的研究工作、取得的进展或碰到的问题等，给出必要的数据、图表。根据实际情况提供国内外有关研究动态的对比分析及必要的参考文献。本部分亦包括国内外合作与学术交流、研究生培养情况等。

3. **下一年度工作计划**，包括国内外合作与交流计划。如要求对原研究内容和主要成员作重要调整，需明确要求调整的内容，并说明理由、必要性以及对项目实施的影响。(注：为保证基金项目顺利进行，研究人员要求稳定，一般不作变更。如确需变更，须按基金项目管理办法规定的要求提出申请，经自然科学基金委归口管理部门核准后方可变更。)

4. **当年经费使用情况与下一年度经费预算**。给出必要的经费使用情况的说明，逐项列出固定资产超过 5 万元的设备的名称、使用情况等有关说明。

5. **存在的问题、建议及其他需要说明的情况**。说明项目执行中的问题和建议。对部分探索性强的研究，有可能未获得理想结果或甚至失败，请如实地反映，说明原因、工作状况、发展态势和建议等，供基金委管理人员或同行专家参考。

6. **附件：**给出标注基金资助的已发表和已有录用通知的论文目录、其他成果清单和必要的证明材料复印件等。发表论文按常规文献引用方式列出。

报 告 正 文

1．年度计划要点和调整情况

1.1 2011 年计划要点：

① 国内外研究现状进一步调研分析；

② 在定性仿真方法基础上研究基于知识库的定性推理方法；

③ 研究定性与定量结合的方法在仿真优化过程中的集成应用，建立优化问题的定性定量分析方法；

④ 发表论文 1～2 篇；

⑤ 完成年度总结报告。

1.2 研究计划调整情况

按研究计划正常进行，对计划未做调整。

2. 研究工作主要进展和阶段成果

2.1 研究工作进展

2.1.1 基于 QSIM 的定性仿真方法研究

定性推理（定性仿真）是使用定性信息对系统结构、行为和功能进行描述，并研究它们之间的关系和因果性推出定性解释，以模仿人类定性常识推理的一种跨领域推理方法。定性仿真相对传统的定量仿真而言，力求非数字化，克服定量仿真的弱点，用非数字手段处理输入、建模、行为分析和输出等仿真环节，通过定性模型推导系统的定性行为描述。相对而言，定性仿真能处理更多形式的信息，能初步模仿人的思维，具有一定的推理能力和学习能力。通过对定性仿真进行研究，探索定性与定量信息结合的方法以及在仿真优化过程中进行集成应用的可能性。

2.1.1.1 定性仿真方法的研究

随着定性仿真理论的发展和不断完善，形成了多种定性仿真理论，其中 Kuipers 的 Qsim 算法由于其坚实的理论基础和广泛的应用领域而得到流行，Kuipers 所在的定性推理研究中心在因果推理，基于时序的定性仿真，定性仿真中引入高阶导数约束，混合系统定性仿真，半定性仿真等方面有着广泛深入的理论研究；K.D.Forbus 所在的工作组着重于工程问题的解决，在定性模型合成和分解，定性空间推理，定性时序推理，常识推理等相关方面有着深入的研究。

定性仿真在故障诊断，社会经济领域，系统辨识，工程和工业过程，医疗诊断，教育系统等都有成功应用。Ceyda Güngör Şen 等采用模糊理论将定性目标和定量目标结合起来对企业软件的选择作出决策。S. S. Mustapha 介绍了 QPT 定性仿真方法在化学教学中的应用，通过对化学反应中的特性，行为等推理建模，构建了化学教学中的定性仿真实验系统和定性分析实验系统，帮助学生掌握化学原理，提高学生化学实验中定性分析的能力。C.Schults 将定性仿真应用在 GIS 系统空间推理中，采用模糊逻辑描述空间位置关系，构建了模糊定性查询方法。K.D.Forbus 介绍了定性空间推理在人工智能游戏中的应用，提高了游戏在路径搜索方面的效率。C.A. Erignac 提出了交互式半定性仿真（ISQS，Interactive Semi-quantitative Simulation），并将其引入虚拟环境。ISQS 系统允许用户相互交换实体结构信息，可以通过模型的内部信息获取实体的行为，而且用户在使用过程中可以修改实体信息，打破了用户只能预先编辑实体信息，一旦编辑完成不能修改的局限性。Bratko.I 将定性推理应用在心电图的识别上，根据定性模型来产生心脏工作状况，根据规则归纳系统产生诊断规则库，他给出了心电图诠释系统-KARDIO，澳大利亚的 Telectronics 公司已将此系统的部分成果应用于他们的心脏病诊治系统 Intelligent Pacemaker 中。

国内对于定性推理和定性仿真的研究起步较晚，仅限于少数院校的少数研究者，主要还处在跟踪研究和方法改进阶段。中国科技大学白方周教授所在的课题组紧跟国际最新定性仿真研究动态，在医疗诊断，并行定性仿真，定性仿真模型建立，定性仿真算法等方面取得了不错的成果。陈宗海等人将灰色理论与定性仿真理论结合起来提出系统整体仿真结果的灰色定性仿真算法，结合 Reiter 规则并基于第一原理的故障诊断理论，在系统的故障诊断中应用了灰色定性仿真理论，黄元亮等针对 Kuipers 的摆动变量辨识和高阶导数约束理论进行剖析，提出一种改进方法，引入了定性推理与系统常识相结合的思想，使 Kuipers 的 Qsim 算法得到扩展。朱六璋等介绍了复杂系统的定性建模和定性控制研究进展情况，讨论了主要的定性建模研究方法及定性模型特性，描述了定性控制系统及其结构，定性控制方法分类与分析以及定性控制系统分析等。朱六璋等给出了一类在经济和商业中常见的离散动态系统的定性仿真方法，运用定性差分方程形式定性建模，结合 Qsim 算法中约束传播定量信息进行定性定量集成仿真，适合于延时和分段单调的离散系统。

2.1.1.2 基于 QSIM 的定性仿真研究

定性仿真是定性推理的一种主要方法，Kuipers 提出的基于定性微分方程(QDE)

的 QSIM 算法是定性推理研究领域最为成熟的理论方法之一。QSIM 算法从一个定性约束集和一个初始状态出发，按照一定的推理规则，得到系统中每一个变量的定性变化方向和变化幅度，预测系统未来所有可能的行为。

QSIM 算法是一种面向约束的方法。在 QSIM 算法中，系统被描述成一个参数集以及这些参数间的约束集。QSIM 算法从物理系统的结构来描述和推导系统的行为，而系统的结构是由一个代表物理参数的符号集合，以及描述这些物理参数间相互关系的一个约束方程集合组成的。

其中，算术约束有：

(1)ADD(f, g, h)：f(t)+g(t)=h(t)

(2)MULT(f, g, h)：f(t)*g(t)=h(t)

(3)MINUS(f, g)：f(t) =-g(t)

(4)DERIV(f, g)：f’(t) =g(t)

定性函数约束有：

(1)M+(f, g)：f(t)=H(g(t))，若 H’(x)>0, M+(f, g)为真

(2)M-(f, g)： f(t)=H(g(t))，若 H’(x)<0, M-(f, g)为真

QSIM 中涉及的概念如下：

- 可推理函数 f:[a,b] →R，满足条件 1) f 在[a,b]上连续；2) f 在[a,b]连续可微；3) f 在任何有界区间里含有限个极值点。
- 路标值，f 的行为或状态上重要点的集合，f 在这些点上的行为或状态将发生变化。设路标值集合为：

  $$l_0 < l_1 < \cdots l_k$$

- 可区分时间点，f 函数值发生重要变化的时间点。设可区分时间点集合为：

  $$a = t_0 < t_1 < \cdots < t_n = b$$

- f 在时刻 t 的定性状态 Qs(f, t)表示为

  $$QS(f,t) = < qval, qdir >$$

  其中

  $$qval = \begin{cases} l_j & if \quad f(t) = l_j \\ (l_j, l_{j+1}) & if \quad f(t) \in (l_j, l_{j+1}) \end{cases}$$

$$qdir=\begin{cases} inc & if \quad f'(t)>0 \\ std & if \quad f'(t)=0 \\ dec & if \quad f'(t)<0 \end{cases}$$

- f 在可区分时间点区间的定性状态为 $Qs(f,t_j,t_{j+1})$
- 定性行为描述，由包括可区分时间点上的定性状态和可区分时间点区间的定性状态序列表示：

  $\{Qs(f,t_0),Qs(f,t_0,t_1),\cdots Qs(f,t_{n-1},t_n),Qs(f,t_n)\}$
- 定性状态转移，函数 f 从可区分时间点到可区分时间区间的定性状态转换，称之为 P 转换 $Qs(f,t_i)\Rightarrow Qs(f,t_i,t_{i+1})$，函数 f 从可区分时间区间到可区分时间点的定性状态转换,称之为 I 转换 $Qs(f,t_i,t_{i+1})\Rightarrow Qs(f,t_{i+1})$。其中转换表如表 1：

表 1 状态转换表

P 转移	$Qs(f,t_j)\Rightarrow Qs(f,t_j,t_{j+1})$		I 转移	$Qs(f,t_j,t_{j+1})\Rightarrow Qs(f,t_{j+1})$	
P_1	$<L_j,std>$	$<L_j,std>$	I_1	$<L_j,std>$	$<L_j,std>$
P_2	$<L_j,std>$	$<(L_j,L_{j+1}),std>$	I_2	$<(L_j,L_{j+1}),inc>$	$<L_{j+1},std>$
P_3	$<L_j,std>$	$<(L_{j-1},L_j),dec>$	I_3	$<(L_j,L_{j+1}),inc>$	$<L_{j+1},inc>$
P_4	$<L_j,inc>$	$<(L_j,L_{j+1}),inc>$	I_4	$<(L_j,L_{j+1}),inc>$	$<(L_j,L_{j+1}),inc>$
P_5	$<(L_j,L_{j+1}),inc>$	$<(L_j,L_{j+1}),inc>$	I_5	$<(L_j,L_{j+1}),dec>$	$<L_j,std>$
P_6	$<L_j,dec>$	$<(L_{j-1},L_j),dec>$	I_6	$<(L_j,L_{j+1}),dec>$	$<L_j,dec>$
P_7	$<(L_j,L_{j+1}),dec>$	$<(L_j,L_{j+1}),dec>$	I_7	$<(L_j,L_{j+1}),dec>$	$<(L_j,L_{j+1}),dec>$
			I_8	$<(L_j,L_{j+1}),inc>$	$<L^*,std>$
			I_9	$<(L_j,L_{j+1}),dec>$	$<L^*,std>$

2.1.1.3 对开源 QSIM 定性推理库 CQ 的集成

CQ 是基于 C 语言开发的对 Kuipers 动态定性仿真算法 QSIM 的程序实现。CQ 算法开源库按照 QSIM 定性仿真算法机理，实现 ADD、MULT 和 DERIV 算术约束和 M+、M- 定性函数约束。根据 P 和 I 转换规则，实现定性状态推进机制，即由当前定性状态产生其后继状态的一个不断推进的过程。本课题在 CQ 开源代码库基础上，进行集成封装，提供一个灵活的定性仿真环境，对 Kuipers 的定性仿真方法进行应用，为后续下一步定性定量结合的方法研究打下基础。

2.1.1.3.1 定性模型的输入

由于CQ只是对QSIM的定性约束和状态转换机制等进行了函数实现，没有一套系统的问题解决流程，人机交互性差。这里对CQ进行集成封装，使被解决问题以规范的格式进行描述，利用CQ的定性推理功能，推算出结果。首先，对定性问题的描述进行划分，分为定性系统变量的路标值空间、系统的常值变量、系统变量的约束关系、新增路标值标志和系统的初始状态等，并把此划分方式以规范的格式写入到问题定义文件中，格式如下：

```
quantity_spaces        //定性系统变量的路标值空间
    *
end quantity_spaces
independent            //系统的常值变量
    *
end independent
constraints            //系统变量的约束关系
    *
end constraints
print_names            //新增路标值标志
    *
end print_names
make_initial_state     //系统的初始状态
    *
end make_initial_state
```

并且采用表 2 的接口完成对定性问题描述信息的读取。

表 2　信息读取接口定义

接口	功能
xform	读取模型文件
doqspace	读取定性系统变量的路标值空间
doindependent	读取常值变量
doconstraints	读取约束关系
doprintnames	读取新增路标值标志
doinitialstate	读取初始状态

其次，对CQ中的推理机制进行集成，对CQ中算术约束（ADD、DERIV、MULT、M+和M-）、状态转换接口和过滤函数接口进行调用。其接口功能如表3：

表 3 CQ 集成接口

接口	功能
mplus	M+
mminus	M-
add	ADD
deriv	DERIV
mult	MULT
mkinitstate	初始化
genpvals	状态转换接口
cfilter	过滤函数接口

2.1.1.3.2 算法流程

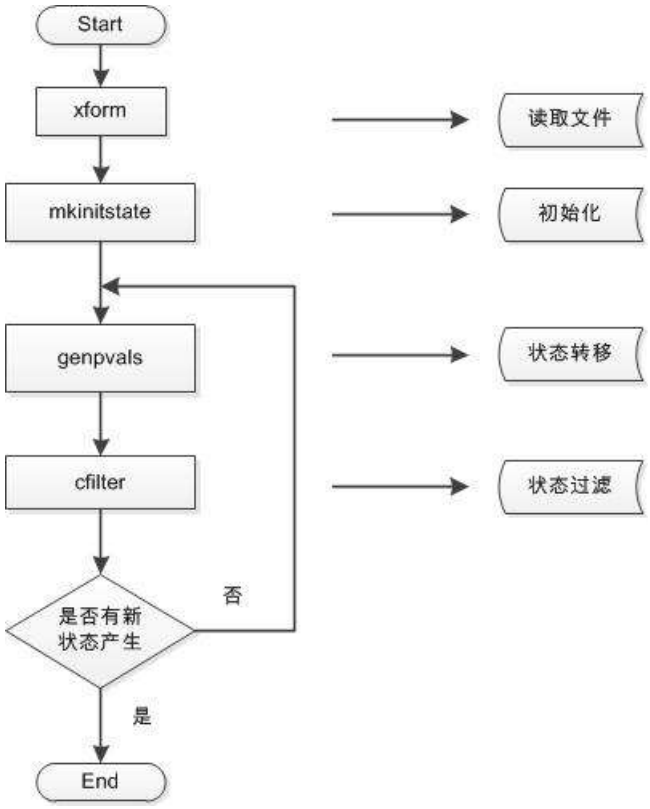


图 1 集成 CQ 的定性推理算法流程

定性推理算法的主要步骤如下：

1、 调用信息读取接口接口，从问题定义文件中读取初始化状态；

2、 根据初始化文件，调用 mkinitstate 函数对各个参数状态进行初始化；

3、调用 genpvals 函数，对每个参数按转换表进行所有可能的转移；

4、调用 cfilter 函数进行状态过滤：（1）对每一个约束，产生状态转换的二元或三元组集合，依约束关系，过滤掉与约束不一致的元组；（2）对有公共变元的约束进行组对，对于组对的元组作一致性过滤，即具有相同函数的两个元组，对同一个函数的转换必须一致；（3）从过滤后剩余下的元组中加以组合，生成所有可能的全局解释，每个解释生成一个新状态作为当前状态的后继状态，如果不存在相应的全局解释，则作不一致标记， 否则标记它们为当前状态的后继状态；（5）对新产生的定性状态作全局过滤，剩下的状态送入活动表中。全局滤波排除下列状态：

1) 无变化情形：新状态与它的直接前驱一致；

2) 循环情形：新状态与某个前辈状态相同；

3) 发散情形：某参数值为，这时当前时间点，必为结束点。

5、如果经过步骤 3 和步骤 4 后，有新的状态产生，则继续状态转换和过滤；如果没有新的状态产生，则程序结束。

在算法运行过程中采用以下几种过滤技术：

(1) 约束一致性过滤

根据函数间的约束关系将各个函数的独立转换组合为相应的元组，由限定这些元组的约束方程来检验。包括定性值的一致性和变化方向的一致性。

(2) 配对一致性检验

主要采用 Waltz 算法：“依次访问每个约束，查看所有与它相邻的约束，由它们所联系着的元组组成元组对，如果一个元组赋予共享函数的转换和它相邻的一个约束的所有元组中不存在，则删除这个元组。”

(3) 全局解释

全局解释的目的是把经过过滤剩下的函数转换赋给相应的函数，以产生系统新的状态。

2.1.1.3 定性仿真应用实例

根据上述的集成方法，开发了集成CQ的定性推理仿真软件工具ISML，能够根据问题定义文件的描述，通过不断的状态转换实现定性推理，并且给出推理结果。

基于此软件建立水池系统的应用实例，对集成了CQ的定性推理方法进行验证。水池系统如图2所示，水池开始是空的，水龙头中的水以一定的流量流到水池，同时水池的排水系统是开着的，利用集成了CQ的定性推理算法推测系统可能的行为。

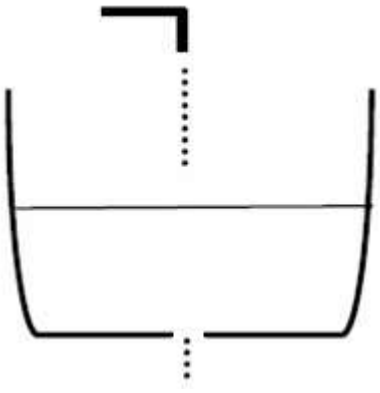


图2 水池系统

考虑系统的水压、水量、水位、进出水量、净水量的关系，建立系统定性微分方程，进而描述整个系统。

以上述定义的输入进行描述：

```
quantity_spaces        //定性系统变量的路标值空间
      amount      (0   full)          //水量
      level       (0   top   inf)     //水位
      pressure    (0   inf)           //水压
      outflow     (0   inf)           //出水量
      inflow      (0   if    inf)     //进水量
      netflow     (minf  0   inf)     //净水量
end quantity_spaces
independent            //系统的某些变量是常值
        inflow
end independent
constraints            //系统变量的约束关系
      (M+   amount   level)    (0 0) (full top)      //水位随水量增加而增加
      (M+   level    pressure)  (0 0) (inf inf)      //水压随水位增加而增加
      (M+   pressure outflow)   (0 0) (inf inf)      //出水量随水压增加而增加
      (A+ netflow outflow inflow)   (0 0 0)          //inflow=netflow+outflow
      (d/dt   amount   netflow)                      //d(amount)/dt=netflow
end constraints
print_names            //新增路标值的标志
      amount      A
      level       L
      pressure    P
      outflow     OF
      inflow      IF
      netflow     NF
end print_names
make_initial_state     //系统的初始状态
      inflow      (if   std)
      amount      (0   nil)
end make_initial_state
```

通过解析输入信息，将参数带入定性推理算法的运行流程中，得到推理结果。在ISML 中的运行结果截图如图 3 所示：

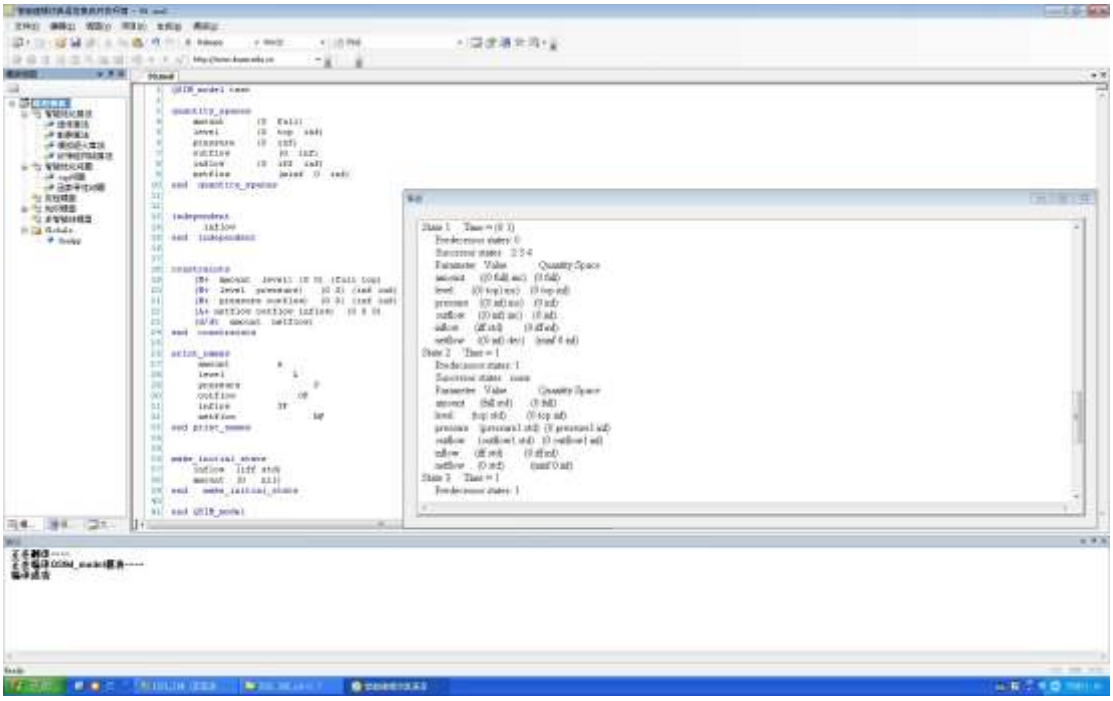


图 3 定性模型实例截图

最后推理出的稳定状态包含三种，即水池满水从水池上沿溢出、水池中水位不变进水量等于出水量、水池为空进水量等于出水量，三种状态均满足实际情况。具体推理结果如图 4：

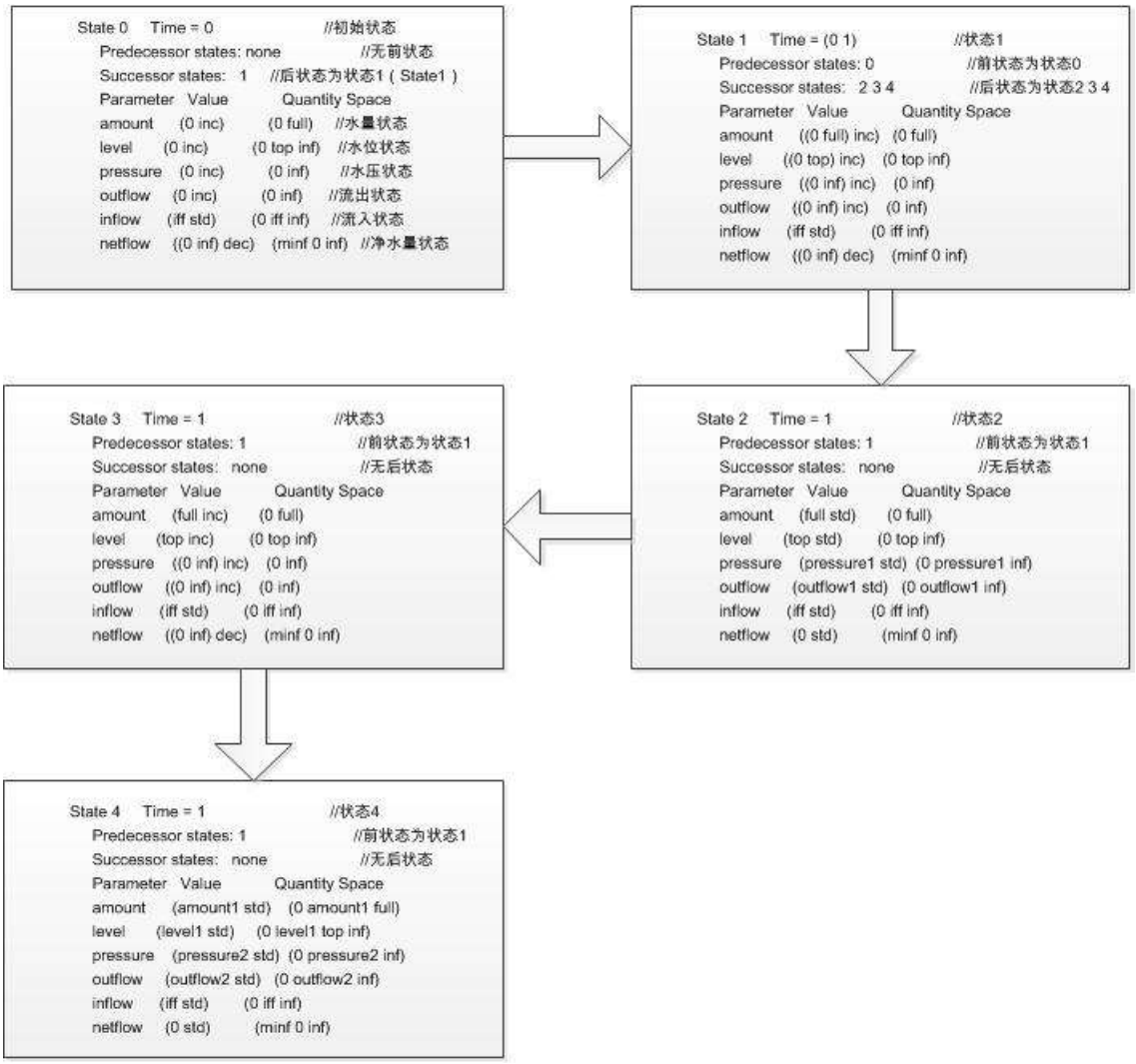


图 4 定性推理结果图

2.1.1.4 基于 QSIM 的定性仿真分析

由此可见，基于 QSIM 的定性仿真是建立在系统结构能够用定性微分方程描述的基础上，以描述系统定性结构的定性微分方程和系统的一个初始状态为输入，通过仿真得到输出结果。但是，如果描述系统的定性微分方程十分复杂，或者难以用常规定性微分方程描述，则利用此方法就有些困难。

尤其在复杂产品的设计过程中，涉及到多个学科，物理规律的分析和耦合关系复杂。若采用基于 QSIM 的定性仿真来指导定量优化过程，在定性微分方程的建立上存在较大的困难，不利于定性定量信息的结合。因此在后面的研究过程中进一步考虑应用知识库和优化算法结合的方式为定量优化过程提供指导。

2.1.2 基于 COMMONKADS 及 CLIPS 的知识库构建

2.1.2.1 CommonKADS 原理及模型套件

CommonKADS 是一种广泛应用的知识工程方法学，实现了知识工程从理论到应用的转变，强调知识的动态属性。它明确提出了关键决策性知识是知识获取的对象，把知识看作是具有良好结构的函数模型，可以根据通用的种类、模式和结构来进行明确的分析。在工程方法上，CommonKADS 提出了知识获取的工程技术路线和一套组织与任务分析工具，从识别组织的问题入手，研究其解决方案及其可行性。从任务相关的知识中获取关键决策知识，为组织应对各种需求的变化提供了实用有效的知识系统需求分析模型，从而帮助组织确认知识资产在组织流程中的价值，帮助组织快速、正确的导入知识系统，以减少人力、时间和资源的浪费。

CommonKADS 结构化方法的提出主要基于以下四种基本思想：

- 知识工程不是“从专家的头脑中挖掘”的某种东西，而是由构造人类知识不同方面的模型组成；
- 知识级原则：在知识模型中，首先要集中考虑知识的概念结构，而把编程细节留在以后考虑；
- 知识具有稳定的内部结构，由可区别的特定的知识类型和角色进行分析；
- 知识项目必须根据从经验中所学，以可控的“螺旋”方式来管理。

根据四种基本思想设计的 CommonKADS 模型套件结构如图 5 所示，其中的精髓为知识模型部分。

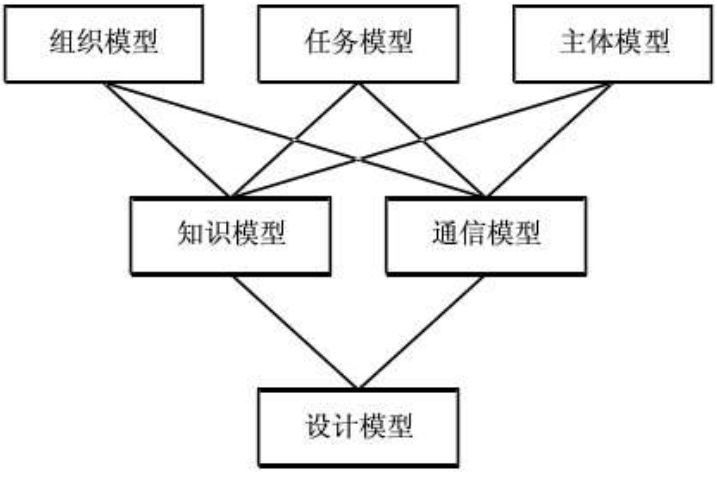


图 5 CommonKADS 模型套件

CommonKADS 含有一个预先定义好的模型集合，每一个模型侧重于一个限定的方面，综合起来就提供了一个全面的视图。其中组织模型，任务模型和主体模型分析了组织环境和相应知识系统中的关键的成功因素。知识模型和通信模型为问题求解功

能以及知识系统要处理和提交的数据生成概念上的描述。设计模型把它转换成一个技术规范说明，作为软件系统实现的基础。

2.1.2.2 知识模型组件

在 2.1.2.1 节中提到知识模型是详细解释执行任务时用到的知识的类型和结构。不同的知识组件在解决问题时所担当的角色是不一样，知识模型描述了这些角色的独立的实现，而它采取的方式是人可以理解的。这使得知识模型在系统开发过程中成为了一种重要的交流工具。

知识是一切智能行为的基础，也是人工智能的重要研究内容。要使计算机具有智能，就必须使它具有知识。知识模型本身是一个帮助我们阐明知识-密集型信息-处理任务结构的工具，一个应用的知识模型可提供应用的所需要的数据和知识结构的规范说明。知识模型中并不包含任何实现术语，实现的部分属于设计模型。

知识模型的目的是详细解释执行任务时用到的知识的类型和结构。它包括三部分，每一部分包含一组相关的知识结构，而每一部分被称为知识范畴。一个知识模型由三个知识范畴构成。第一个范畴称为领域知识。这个范畴详细说明特定领域知识应用中需要的信息类型。一个领域知识的描述在某种程度上类似于软件工程中的“数据模型”或“对象模型”。第二个范畴包含推理知识。推理知识描述了使用领域知识的基本推理步骤-相当于推理机构件。第三个范畴是任务知识。任务知识描述一个应用所要达到的目标是什么，如何分解任务和推理来实现这些目标。知识系统内的知识结构分布在这三个范畴中，其关系如图 6 所示。

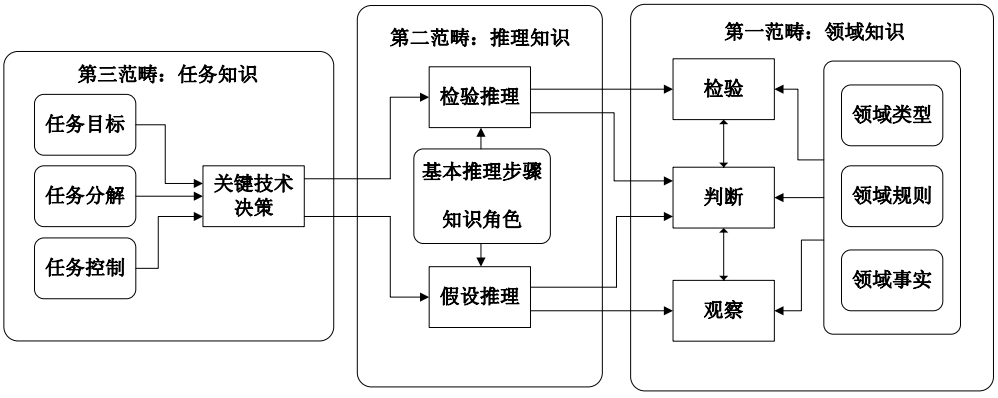


图 6 知识模型结构

（1）领域知识

描述一个应用领域中主要的静态信息和知识对象，由两种成分组成：领域模式和知识库。

领域模式是通过一些类型定义对特定领域知识和信息进行的刚要性描述模式描述了应用领域的静态信息/知识结构。从软件工程的角度看，领域模式类似于数据模型或对象模型。领域模式中主要的建模结构是概念（CONCEPT），概念用来描述应用领域中发生的具有类似特征的对象或实例，类似于其他方法中的类的概念。概念的特征使用属性（ATTRIBUTE）来表示，每个属性具有相应的值（VALUE）以及值类型（VALUE TYPE），属性值使概念实例具有的一条原子信息。

知识库包含某个领域模式中详细说明类型实例。在知识库中包含了领域事实和领域规则两部分。领域事实是对领域模式中概念的具体化，领域规则是对领域事实属性间关系的具体描述，通过产生式规则进行表述。

（2）推理知识

推理知识是描述如何将领域知识用于执行一个推理过程，相当于执行一系列的推理函数。它处于领域知识和任务知识之间。

（3）任务知识

任务知识是描述知识的目标以及为实现目标所采取策略的知识范畴。人们对任务的表达方式多种多样，有可能不同的表达方式表达的就是相同的含义。也有可能相似的表达方式表达的却是不同的含义。

2.1.2.3 CLIP 推理引擎集成方法

推理技术是知识应用的核心，而推理技术最重要的核心是匹配算法。当前，主要的模式匹配算法有线性算法 (LinearAlgorithm)、协商算法 (TreatAlgorithm)、跳跃算法(Leaps Algorithm)和 Rete 算法 (ReteAzgorithm)。这些算法都可以单一或者混合使用作为规则引擎技术的推理基础和规则系统的算法基础。其中最为规则推理系统所广泛接受和使用的 RETE 算法是在 1978 年就由 Dr.CharleSL.Forgy 于卡耐基梅隆大学提出，并于 1982 年推出此算法的第一个实现版本。

2.1.2.3.1 Rete 算法基本思想

Rete 算法的基本思想是保存以往匹配过程中产生的所有信息，只将新增加的事实与规则进行匹配。虽然这样增加了处理的空间复杂度，但是考虑到如果反复地对规则和所有事实进行匹配，每一个执行周期中的匹配运算时间复杂度都将是 O(RF^P)(R 指规则库中的规则数量，F 为当前工作记忆中的事实数量，P 是规则包含的平均模式数量)，且其中的众多匹配结果与前一个执行周期中得出的结论是重复的，以空间代价来换取系统的执行速度是值得的，这样非常有效地提高了系统的执行效率。

具体来说，Rete 算法的基本思想基于两个重要的依据，它们是时间冗余(TemporalRedundaney)和结构相似(StruCturalSimilarity)。

时间冗余指的是存在这样的事实：在规则系统的运行时态(Runtime)，规则库是确定且固定的，可以认为此时的规则是不变的，即没有增加也没有删减，规则包含的模式同样没有增加和没有删减并且不做修改。唯一变化的是事实，在运行周期之间，工作记忆中的事实不断地被添加或者移除，但是这种变化事实的量相对于整个工作一记忆所占的比率是很小的，所以只将新增加的事实与全部的或者相应的规则进行匹配。

结构相似指的是规则库中的许多规则都含有相同的模式，即同一个模式经出现在多个规则的 LHS（末项）部分。因此，记录下已经与事实进行过匹配的模式也可减少各个运行周期中匹配运算的次数。

基于以上两个依据，Rete 算法可以使匹配运算的时间复杂度降低到近似 0(RFP)，其中 R 指规则库中的规则数量，F 为当前工作记忆中的事实数量，P 规则包含的平均模式数量，与 O(RF^P)相比大大减小了匹配运算的时间复杂度，从而提高系统效率。

2.1.2.3.2 知识模型推理机调用

CLIPS（C Language Integrated Production System，C 语言集成产生式系统）是美国航空航天局约翰逊空间中心于 1984 年用 C 语言设计、开发的一种用于编写基于规则的通用专家系统开发工具。它是人工智能与专家系统领域最成熟，应用最为广泛的逻辑程序设计语言。CLIPS 提供一种易于理解、类似于自然语言的语法，通过定义事实、规则和目标，其推理机能够基于有限的事实和规则对信息进行逻辑推理。目前通过与数据库技术相结合，CLIPS 被成功地用于不同领域智能系统和知识库的构建。

CLIPS 是基于 Rete 算法的前向推理语言，其推理机的推理是基于正向推理的控制策略，是在事实的基础上通过设定的规则对事实进行模式匹配，从而实现推理。

由于 CLIPS 是采用标准 C 语言编写的专家系统开发工具，提供了与 FORTRAN、ADA 和 C 等高级程序设计语言的接口，提供了数百个函数供 C 语言调用，而且是源代码开放的专家系统开发工具,所以十分适合嵌入某种 C 语言开发工具进行二次开发。在实际操作中，可以采用的方法也很多如：源代码嵌入、DLL 库方式、包装类、控件等。

CLIPS 的接口函数主要分为环境设置函数、调试函数、模板相关函数、事实相关函数、规则相关函数等，调用 CLIPS 的推理机涉及到的接口函数主要包括载入模板函数，载入事实函数，搜索策略设置函数、存储当前事实函数、运行推理机函数等。图

7 所示是用 VC++调用 CLIPS 的流程。

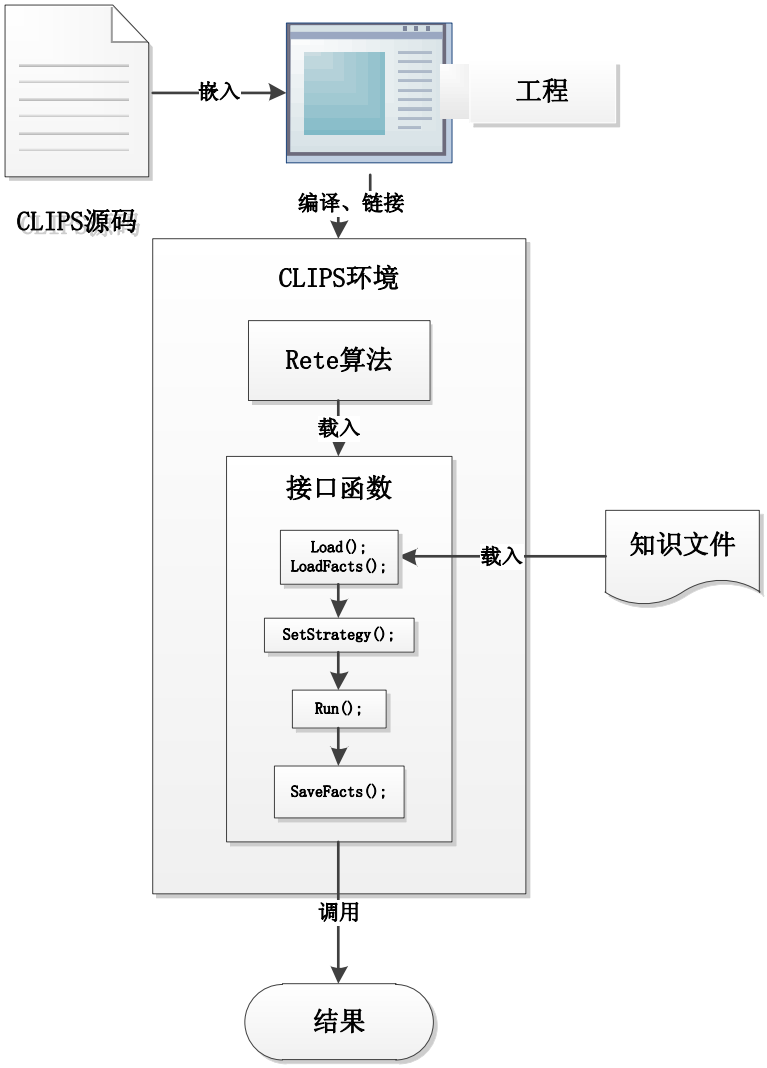


图 7 C 语言调用 CLIPS 的流程

使用 CLIPS 的推理机进行推理，只有在所有规则中再也找不到可以匹配的事实和规则时才会停止推理，不同于传统的 C 语言是按程序的流程进行匹配。

C 语言与 CLIPS 语言都可以进行知识的匹配推理，但两者有很大的不同。首先 CLIPS 的推理是面向知识的，能很好的表述智能模型，而 C 语言对知识的描述比较复杂，并且不具有良好的可读性；由于 CLIPS 采用了 Rete 匹配算法作为推理核心，当面对实际应用中的复杂模型，尤其当领域事实的更新比较迅速时，能快速的进行匹配，而 C 语言只有在结构相对简单的模型中才能体现出优势；CLIPS 的程序运行结构并不是顺序的，而是基于当前领域事实确定运行结构，推理的结束以当前而没有可以进行匹配的事实规则为准，而 C 语言是顺序结构运行的，确定推理结束相对比较复杂。

C 与 CLIPS 语言的关键区别是程序的执行流程。CLIPS 的程序执行具有随机性，

例如当前的知识模型中存在规则 1-规则 5 共 5 条规则，若当前领域事实与规则 5 相匹配，则 CLIPS 执行规则 5，若在规则 5 被触发时生成了一条新的事实，而此条事实和规则 1 相匹配，CLIPS 会自动对新生成的规则进行匹配，直到无可匹配为止。而 C 语言则会一直按程序流程执行，不断的将所有事实和所有规则进行匹配，直到测试完所有事实都不能匹配任何规则时才停止。因而基于 CLIPS 的推理效率高，并且更符合实际情况。

将 CLIPS 与基于产生式规则的典型语言 Amzi Prolog 进行比较，可以看出 CLIPS 与 Prolog 的主要区别为 CLIPS 为前向链推理，由当前事实推理得到结论，而 prolog 为后向链推理，从结论进行推理得到支持此结论的事实；前向链推理主要应用于控制与决策，而后项链推理主要用于诊断系统。使用两种语言分别实现了 15 条规则与 6 条事实之间的匹配推理，在推理中一共用到了其中的三条规则，并通过 C 语言进行调用，测试得到的结果如表 4 所示：

表 4 解决知识模型问题性能比较表

	运行时间	结果精度	代码效率	易用程度	应用范围
Amzi Prolog	0.031 秒	高	61 行	较好	较广
CLIPS	0.016 秒	高	62 行	好	广

从两者比较结果可以看出：CLIPS 的运行时间比 Prolog 短，代码行数相同但代码量更少，用户使用更为简单，可以更好的解决包括面向对象、数学计算的问题，应用范围更广。

2.1.2.4 基于知识库推理的应用实例

在复杂产品的设计过程中，涉及到多个学科，物理规律的分析和耦合关系复杂。若采用一般的定性推理方法，其复杂程度难以建立精确的定性微分方程对其表示。若采用基于 CommonKADS 的知识工程方法对复杂产品设计过程进行知识表示，同时集成知识模型推理机，对设计过程进行推理，大大提高了设计过程的有效性。以指挥控制系统为例，建立一个大约 20 条规则的中等级别知识库，验证此方法的可行性。

指挥控制系统（以下简称指控系统）是指挥自动化系统的核心，是战斗力“倍增器”的载体，目前应用最广泛的军队指控自动化系统为 C3I 系统，C3I(Command，Control，Communication，Intelligence)意指“指挥、控制、通信与情报”。根据基于 CommonKADS 知识工程方法学对 C3I 规则集中目标选择部分进行建模，按照知识模型组件进行知识的表示，主要进行了领域知识的描述。将其分为领域模式与知识库两

部分。

为了在不同问题中进行应用验证，同上节中提到的集成了 CQ 的基于 QSIM 的定性推理方法，建立描述问题的知识模型输入文件，并定义其格式，如下：

```
knowledge_model                    //模型名称
   domain_knowledge          //领域知识
     domain_schema                  //领域模版
          schema_fact                //事实模版
            concept               //模版名
                 < type slotname =    default value >
            end concept
end schema_fact
end domain_schema
knowledge_base                //知识库
       domain_fact              //领域事实
         //根据事实模版书写的事实
concept              //模版名
                 <    slotname =    value    >
            end concept
       end domain_fact
       domain_rule              //领域规则
         rule                        //规则名称
[
                  //变量声明
]
            if                        //规则前件
then                 //规则后件
          end rule
end domain_rule
end knowledge_base
end domain_knowledge
end knowledge_model
```

针对本实例，其输入文件的描述如下：

✧ 领域模式

根据 C3I 目标选择规则集中知识的描述，提取除了领域模式中的 4 种概念，分别为长机、僚机、目标和战场状态概念，每个概念包含有不同的属性，以长机为例，其属性表示如下：

```
长机概念中的属性：
长机名称  name 长机名称信息；
存活情况 live  表示长机存活信息，为字符型，可选值{yes，no};
当前状态 state  描述长机当前状态，字符型，可选值{RTB，LLTR，NULL，Etc.}；
```

最大任务时间 maxTasktime 浮点型;
当前任务时间 currentTasktime 浮点型;
允许发射的最大弹药量 maxfiredAmmo 整型;
已发射弹药量 firedAmmoCount 整型;
持续在轨时间 resumeOrbitTime 浮点型;
导引目标 leadtoPosition 字符型;
最大探测范围 maxDetectArea 浮点型
到达拦截点信息 arriveHeadoffPoint 字符型;
射程最大 wq 的发射距离 maxEmissionLength 浮点型;
最大地面距离 maxGroundLength 浮点型;
交战目标 FightTarget 字符型;
飞机与拦截点之间距离 distanceToHeadoffPoint 浮点型;
雷达探测范围 radarDetectRange 浮点型;
长机综合性能指标 performancePara 浮点型;
目标当前速度值 vectorScale 浮点型;
长机当前速度矢量 vector 浮点型，内容为 x 轴矢量，y 轴矢量，z 轴矢量;
长机当前位置 positon 浮点型，内容为 x 轴坐标，y 轴坐标，z 轴坐标;
机体轴线矢量 planeAxes 浮点型，内容为 x 轴矢量，y 轴矢量，z 轴矢量;

✧ 知识库

在知识库中包含了对 C3I 目标选择规则集中具体知识的存储，其又分为领域事实和领域规则两部分。

（1）领域事实包含了当前战场的基本态势，是对领域模式中概念的具体化，其内容是由指控中心传送的，选取领域模式中的长机概念的事实表述如下所示。

```
concept leadplane
        name = leadplane1
        live =yes
        state = NULL
        maxTasktime=400.0
        currentTasktime=300.0
        ……
end concept
```

领域事实中概念的属性名称与属性值类型应符合领域模式的定义，对属性的赋值是以“属性名 ‘=’ 属性值”的方式表示的，如果某个属性的值与领域模式中定义的属性的默认值相同，可以在领域事实中不再进行赋值。

（2）领域规则是对概念表达式之间关系的描述，是对专家知识的一种反映，根据 C3I 规则文档提取出目标选择与导航导引部分的规则。

在目标选择规则集中包括的规则有：对我方编队飞机当前状态进行状态检查，主要检查了长机的存活情况，长机所处状态，长机发射弹药数的检查以及任务时间的检

查。对检查到的不同情况分别采用不同的处理方式；对目标状态进行选择，包括执行指控中心发出的命令，检查目标禁止时间是否过期，检查目标当前所处区域，检查武器交战状态等。根据当前态势进行目标选择，得到我方协同攻击分配方案。

在导航导引规则集中包括的规则有：记录进入导引过程时刻，检查持续在轨时间，确定长机及僚机导引方向，评估目标列表，检测目标与长机距离，飞机攻击战术选择，检查目标所在区域等。

由于该实例比较复杂，代码量较大，并且规则结构较为相似，因而只对其中一条典型规则进行说明：

```
rule checkrtb [leadplane.state , battlefield.commondToSystem/whetherStopReason/currPhase]
        if leadplane.state==RTB and battlefield.whetherStopReason == no and
battlefield.currPhase == FTSP and battlefield.commondToSystem == NULL
        then battlefield.whetherStopReason = yes battlefield.commondToSystem =
exitCurrentPhase
end rule
```

此条规则的名字是 checkrtb，后面的“[]”内的为此条规则用到的一些属性，如长机状态、战场命令等，该规则实现了当长机返回基地时系统停止推理并且传出退出当前阶段的指令。

在此输入文件的基础上，在后台调用集成了 CLIPS 的推理机，进行推理，最后得出结果。

通过此方法对某实际系统进行建模仿真，某时刻当前战场存在 1 架长机，1 架僚机，3 架敌机，我机与敌机之间相向飞行，敌机的速度是 350m/s，其武器攻击范围是 70km-90km，雷达探测距离范围是 140km-160km，我机的参数为对应敌机参数的 1.5 倍。本例运行截图如图 8 所示，输出窗口显示相应结果。

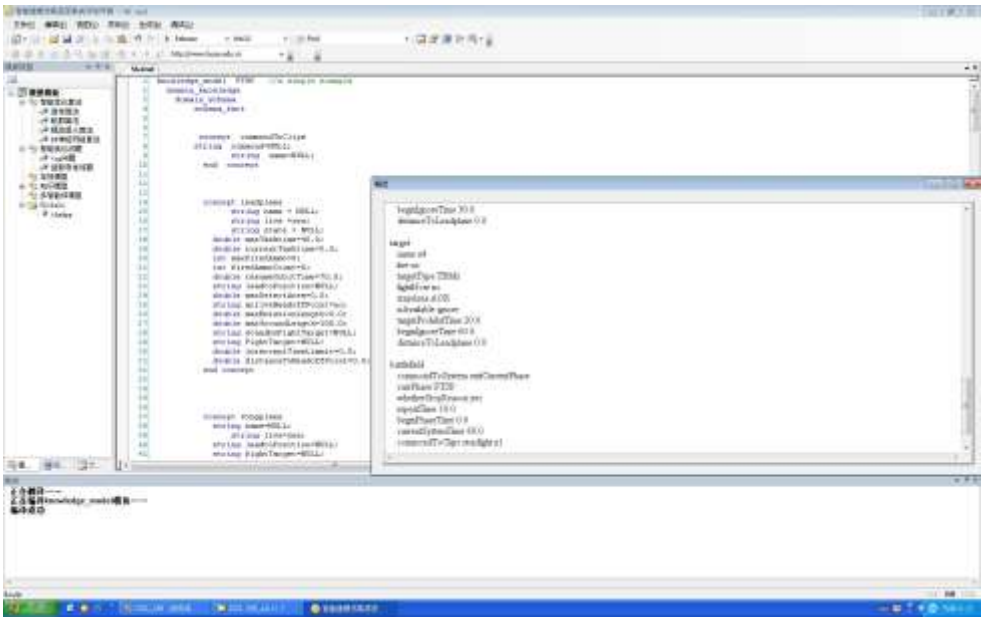


图 8 C3I 知识模型实例截图

在上述战场态势的初始条件下我机与敌机的位置及目标选择结果如图 9 所示。

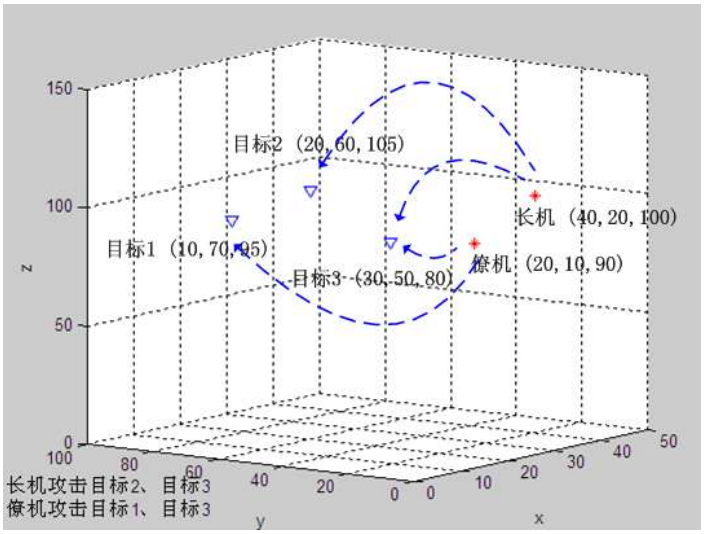


图 9 目标选择结果示意图

2.1.2.5 基于知识库推理的方法分析

针对在复杂产品的设计过程中，涉及到多个学科，物理规律的分析和耦合关系复杂，其复杂程度难以建立精确的定性微分方程对其表示的问题。在分析复杂产品设计过程中的特点的基础上，提出采用基于 CommonKADS 的知识工程方法对复杂产品设计过程进行知识表示，同时集成知识模型推理机，对设计过程进行推理，并通过指挥控制系统的目标分配实例，验证了基于知识库推理方法的可行性和有效性。

然而，知识库不仅可以通过推理机制在复杂产品设计过程中得到应用，当前，智能优化算法正被广泛应用于复杂产品设计的参数优化过程中，如果以已有知识为算法

参数的选择提供指导，对模型最优解应满足的规则给出判断，将已有的知识与智能优化算法结合也许可以有效提高算法收敛速度，辅助复杂产品设计。因此，本课题开展了基于知识库指导的蚁群算法的研究。

2.1.3 基于知识库的改进蚁群优化算法研究

在实际工程应用中，对于待解决的问题，用户根据以往的成功案例、经验或人工估算可以对问题的可能解、搜索方向会有大致的判断，所以将智能优化算法与知识工程相结合，即将针对不同类型问题的相关案例、经验、规则存入知识库之中，可以指导智能优化算法针对当前问题产生初始状态、改进优化搜索策略。

具体而言，基于知识库的算法改进体现在如下三个方面：

1、基于知识库对智能优化算法初始及运行阶段的状态、参数选取给出建议；

2、借鉴知识库找出类似实例指导当前初始状态的产生；

3、根据知识库中的定性约束、定量关系等专家经验指导搜索策略（搜索方向、禁忌）。

根据这三条基本方法，结合不同智能优化算法及待求解问题的特点，基于知识库给出具体改进策略。

基于知识库改进智能优化算法，将会有效提高算法各项优化指标。可以更好的解决设计、决策、调度等问题。下面以蚁群算法为例，说明基于知识库的改进蚁群算法模型、应用流程并通过其解决旅行商问题来验证效果。

2.1.3.1 基于知识库的动态蚁群算法模型

分析基本蚁群算法思想可知，算法认为蚂蚁对待解决问题没有了解，所以在初始化阶段，蚂蚁被随机放置在各城市，各城市间的信息素浓度都是相等的，在蚁群算法寻优的过程中，其主要参数都不能随寻优状态的变化而动态改变，算法本身也缺乏跳出局部极值的能力。

现有对蚁群算法模型或参数的改进多是针对算法的某一个或几个参数采取特定的方法，如信息素强度调整方式、蚂蚁放置方式等，这样的改进对某类特定问题会取得较好的效果，而解决其他类型问题时往往就无能为力了。

针对基本蚁群算法的两个主要缺点，提出基于知识库的动态蚁群算法（Dynamic Ant Colony Algorithm based on knowledge base，DACA），其特点是建立蚁群算法知识库，将专家经验、历史数据、算法参数等内容有序存储在知识库之中，DACA 在知识库的指导下对算法参数赋初值，并根据寻优状态动态改变参数，使蚁群算法较快收敛，

当算法停滞时引入扰动，尽量避免算法陷入局部极值。

DACA 包括了三个部分，即知识部分、接口部分和应用部分，DACA 结构如图 10 所示。

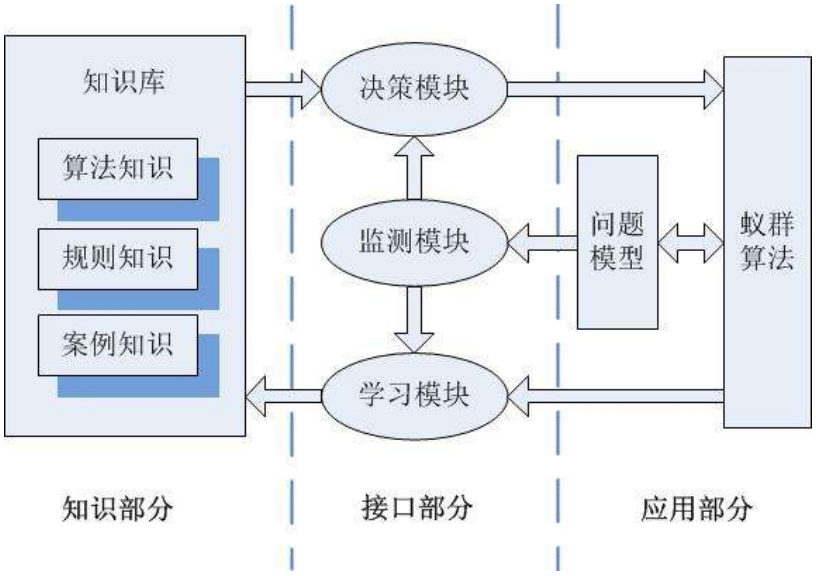


图 10　DACA 结构图

知识部分存储历史知识和当前寻优过程提取的知识，包括算法知识、规则知识和案例知识。算法知识是针对蚁群算法不同参数的可选范围，如蚁群算法的信息素更新策略主要包括 Ant-Cycle、Ant-Quantity、Ant-Density、Max-Min 等方式。规则知识是针对不同实际问题选取算法参数的方法及算法运行中的一些规则，如针对不同类型问题蚁群算法选择一种或几种信息素更新策略。案例知识是针对典型问题的已有成功案例，包括问题的最优解和最优解对应的算法参数。

接口部分是应用部分和知识部分的接口，包括决策模块、监测模块和学习模块。监测模块实时监测问题模型的状态，若模型状态趋于稳定，并且在不同扰动下未获得更优解，则认为获得了问题的最优解，并向学习模块发送信号。学习模块分析问题模型和蚁群算法当前参数状态，经知识提取后存入知识库之中，若模型状态发生变化或在一定周期内未能继续收敛，则向决策模块发送信号。决策模块根据知识库中的相关知识和问题模型的当前状态改变蚁群算法的相关参数。

应用部分类似于传统的算法应用，包括了蚁群算法和问题模型两部分，问题模型受到实时监控，若得到模型的较优解，则当前模型状态和算法参数被记录，否则将根据知识库以较大概率改变算法参数。

知识部分和应用部分通过接口部分进行信息传递，各部分的具体内容及应用通过

下文的实例进行详述。

基于蚁群算法的已有知识，参照图 10 建立算法知识库。下面具体说明 DACA 知识库中的算法知识和规则知识。

分析基本蚁群算法相关参数及针对蚁群算法的研究成果，总结算法知识，DACA 可表示为如下的 12 元组。

DACA={α，β，ρ，Q，m，τ_0，τ_c，η，L，P，γ，f}

集合 DACA 中的元素分别为：

α-信息启发式因子；　　β-期望启发式因子；

ρ-信息素挥发因子；　　Q-信息素强度；

m-蚂蚁数目；　　τ_0-初始信息素浓度；

τ_c-信息素更新策略；　　η-启发函数；

L-蚂蚁放置方式；　　P-选择下一节点方式；

γ-随机扰动方式；　　f-评价函数。

其中部分元素组成如下：

Q ={Q_1, Q_2}为信息素强度的集合。Q_1 为固定值，Q_2 为时变值。

τ_c ={τ_{c1}, τ_{c2}, τ_{c3}, τ_{c4}}为信息素更新策略的集合。τ_{c1} 为 Ant-Cycle 方法，τ_{c2} 为 Ant-Quantity 方法，τ_{c3} 为 Ant-Density 方法，τ_{c4} 为 Max-Min 方法。

L={L_1, L_2, L_3, L_4}为蚂蚁放置方式的集合。L_1 为随机放置，L_2 为均匀放置，L_3 为中央城市优先放置，L_4 为偏远城市优先放置。

P={P_1, P_2, P_3}为蚂蚁选择下一节点方式的集合。P_1 为赌轮选择法，P_2 为 Q 学习选择法，P_3 为蚂蚁感觉阈值法。

以蚂蚁选择下一节点方式 P 为例，说明算法的规则知识。在图 11 中 k 为算法迭代序号，T_p 为参数 P 对应的参数判断周期，C_{Pi} 为集合 P 中因子 P_i 对应的选择概率。算法初始化阶段对 C_{Pi} 赋初值，算法每经过 T_p 轮迭代，判断解的均值情况，若优于上一轮均值则增大当前 P_i 对应的选择概率 C_{Pi}，否则降低 C_{Pi}，然后基于 C_{Pi} 按照赌轮法选择 P_i 并进入下一轮迭代。算法所有的动态参数都按照类似流程基于规则知识对参数进行调整。

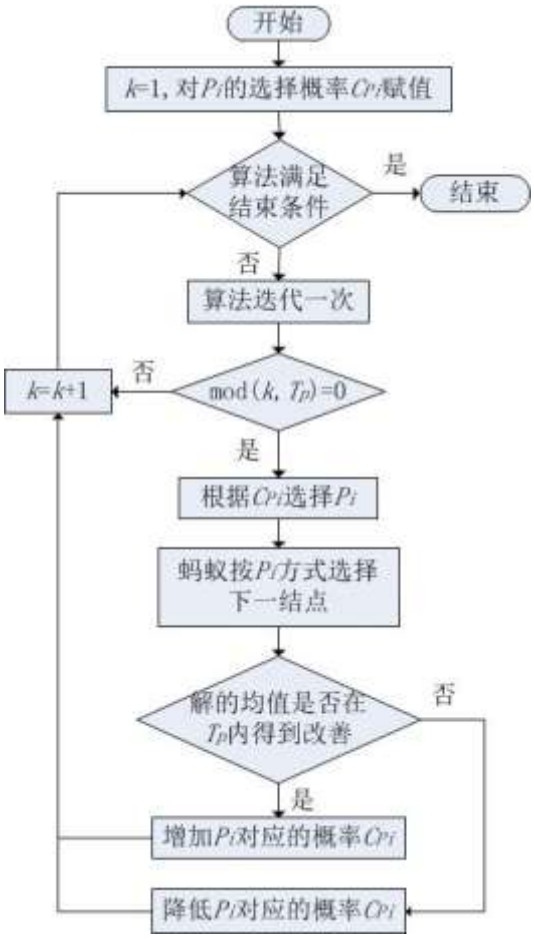


图 11 P 操作基于规则知识的调整流程

2.1.3.2 算法应用流程及分析

下面以旅行商问题为例，说明 DACA 的应用流程。参考 DACA 结构图，针对旅行商问题特征，将相关的定性定量知识存入知识库之中，在 DACA 运行的三个阶段，基于知识库调整算法参数。

在算法初始化阶段，针对动态问题模型的特性，分析知识库内已有知识，首先通过比较函数寻找与当前问题模型相似的案例知识，根据被选案例与当前问题模型的符合程度确定加权系数，参考相似案例的加权对当前算法参数进行赋值。同时，根据专家经验，即针对当前问题的已有经验或人对问题观察后而得到的一些指导性意见，对信息素初始浓度进行赋值，例如在解决 CHN144 问题时，观察城市分布状况，易知最优路径应连续通过相对集中的东北地区的十余个城市，所以将这十余个城市之间赋予较高的初始信息素浓度。初始化阶段的针对性参数赋值使算法的初始解接近了最优值，避免了一些盲目的搜索，从而在一定程度上加快了算法收敛。

在算法运行开始后，在规则知识的指导下，DACA 动态的调整算法参数，如图 11

所示，根据平均解的比较结果，调整算法知识中不同算子对应的概率，基于赌轮法选择算子，以较大概率选择合适算子，进而加快了算法收敛；同时也对算法带来一定扰动，增大了算法的搜索区域，避免算法陷入局部极值。在旅行商问题中，若有路径交叉则非最优解，据此规则知识，在算法搜索过程中进行判断，若某只蚂蚁选择路径有交叉，则相应降低交叉部分的信息素值，使蚂蚁在下一轮搜索中降低路径交叉的可能性，从而提高搜索效率。

在算法运行后期，算法参数趋于稳定，为避免算法陷入局部极值，当最优解连续几个参数周期停滞时，根据规则知识引入扰动算子，尽量使算法跳出局部极值，以搜索全局最优解。

当算法满足终止条件并寻得当前问题的较优解时，可通过学习模块提取经验、规则并存入知识库之中，便于以后对类似问题给出更好的指导。

DACA 输入为基于知识库选定的算法 12 组初始参数和问题模型初始解，输出最优解及对应参数，算法包括如下 5 个关键步骤。

Step 1：初始化阶段

根据已有案例知识及专家经验对算法参数，即 DACA 对应 12 元组中的元素进行赋值，DACA 中 Q、τ_c、L、P 取其对应集合中的某一因子，DACA 中的其他元素取固定值，其中 τ_0 参考案例知识进行赋值。

假设 $x \in \{Q, \tau_c, L, P\}$，对参数 x 的判断周期 T_x 及 x 对应集合中各因子的选择概率 C_{xi} 进行初始化。设定算法迭代序号 k=1 及算法终止条件。

Step 2：算法一次迭代

根据 DACA 当前参数进行一次迭代并更新信息素，若某只蚂蚁选择路径存在交叉，则相应降低交叉部分的信息素值。迭代结束后记录解的均值及最优解。

Step 3：调整算法参数

基于规则知识参照图 11 所示流程调整参数 x。若 $\mathrm{mod}(k, T_x)=0$，则根据解的均值是否在 T_x 周期内得到改善来调整 x_i 对应的概率 C_{xi}，并基于赌轮法重新在参数 x 集合中选择因子。

Step 4：适时引入扰动

若最优解连续几个参数周期停滞，则基于规则知识改变部分算法参数，例如局部信息素浓度过高时则降低 Q，或调整 C_{xi}。

Step 5：判断算法是否结束

若满足算法终止条件，则输出最优解并根据当前寻得的最优解提取知识，否则 $k=k+1$ 并转至 Step 2。

算法通过 ρ、Q、τ_0、τ_c 等参数对信息素进行更新，通过 α、β、L、P 等参数放置蚂蚁并选择路径，DACA 根据已有知识及当前状态，动态的改变算法参数 Q、τ_c、L、P。其中参数 Q 取时变值，其较大时算法易陷入局部极值，反之则收敛速度较慢，参数 τ_c、L、P 的不同因子针对问题的规模、特性和阶段各有优缺点。DACA 能够以较大概率选择合适参数使算法较快收敛到最优值，避免了基本蚁群算法前期需要大量时间调整参数，算法后期因参数固定而导致收敛速度慢、易陷入局部极值等问题。

假设 TSP 问题中蚂蚁数为 m，城市数为 n，经 N 次迭代后得到最优解，则可知传统蚁群算法各阶段时间复杂度：初始化阶段为 $O(n^2)$，迭代一次为 $O(m \cdot n^2)$，信息素更新并判断算法是否结束为 $O(m \cdot n)$，故总的时间复杂度为 $O(n^2)+N_1 \cdot O(m \cdot n^2)+O(m \cdot n) \approx N_1 \cdot O(m \cdot n^2)$。DACA 中参数选择及调整阶段的时间复杂度为 $O(m \cdot n)$，同理其可知复杂度近似为 $N_2 \cdot O(m \cdot n^2)$，由前文分析可知 $N_2 << N_1$，所以 DACA 的时间复杂度更小。

知识库中知识的丰富程度对寻优结果有较大影响，当知识库中包含有效的算法、规则和案例知识时，可以大幅提高算法性能。

2.1.3.3 实验结果及分析

为全面衡量算法性能的优劣，这里引入评价算法离线性能的三个基本指标，即优化性能指标、时间性能指标和鲁棒性指标。

用“相对误差 E_e”来表示优化性能指标，如式 2.1 所示，该指标用于衡量算法对问题的最佳优化度，体现了算法是否能避免陷入局部最优，其值越小意味着算法的优化性能越好。其中“理论最优解”是指问题的最优值（或已知的最佳优化值），而“最优解”是当前算法经过多次运行所得到的最佳优化值。

$$\text{相对误差} = \frac{\text{平均值} - \text{理论最优解}}{\text{理论最优解}} \times 100\% \quad (2.1)$$

用“搜索率 E_t”来表示时间性能指标，如式 2.2 所示，该指标用以衡量算法对问题解的收敛快慢程度即效率，在设定的最长收敛时间固定的情况下，E_t 值越小说明算法收敛速度越快。

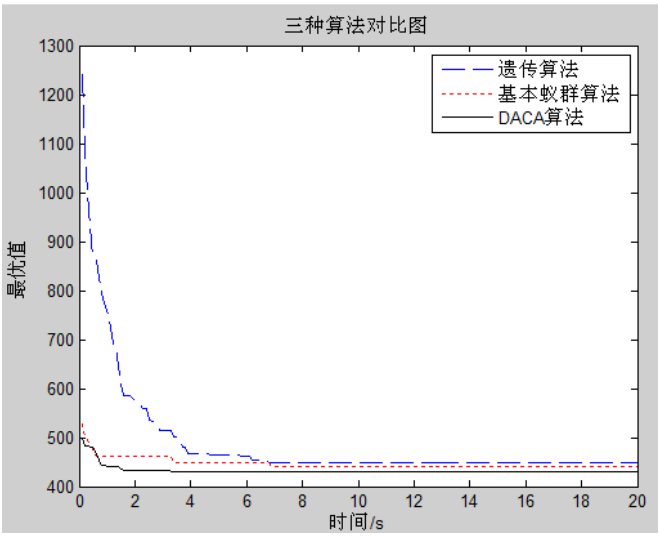
$$\text{搜索率} = \frac{\text{平均收敛时间}}{\text{最长收敛时间}} \times 100\% \quad (2.2)$$

用“波动率 E_f”来表示鲁棒性指标，如式 2.3 所示，该指标用以衡量算法在随机初值下求得最优解的波动程度，值越小说明算法的鲁棒性越高。

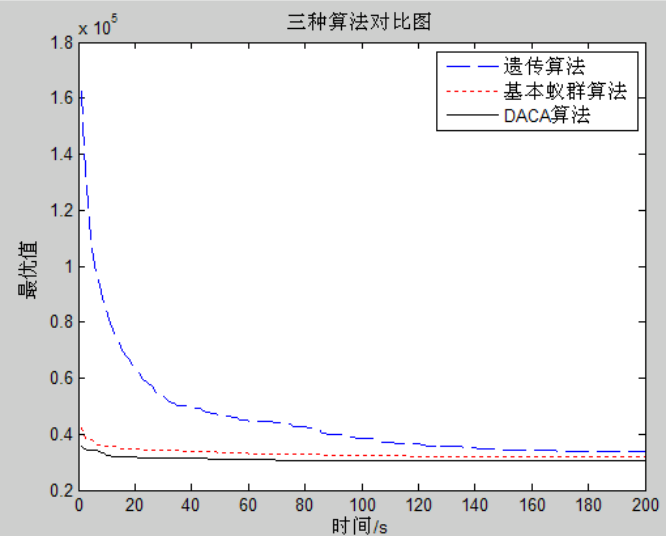
$$\text{波动率}=\frac{\text{最差解}-\text{最优解}}{\text{理论最优解}}\times 100\% \qquad (2.3)$$

分别用遗传算法、基本蚁群算法和 DACA 解决 TSPLIB 中的 Eil51 和 CHN144 实例。其中遗传算法主要参数设置为：交叉概率 0.9、变异概率 0.13。参考相关文献可知，蚁群算法中参数的最优设置为：$m = [0.6n, 0.9n]$，$\alpha \in [1.0, 2.0]$，$\beta \in [4.0, 6.0]$。在问题规模（城市个数）n 固定情况下，取蚂蚁个数 m=0.7n，信息素重要程度参数 α=1，启发式因子重要程度参数 β=5。DACA 中的参数 Q、τ_c、L、P 则基于知识库动态变化。为便于比较，在解决同一实例时，三种算法设定相同搜索时间。运行平台为 Windows XP 系统，Matlab R2008a，CPU 为 Intel Core Duo E4600，内存为 2G。

各独立进行 10 次实验，计算过程为浮点运算，图 12 为某次实验中三种算法解决 Eil51 和 CHN144 的收敛曲线对比图。三种算法解决两个实例的实验数据及三项性能指标列于表 5，6 中。将相关文献的实验数据亦列于表中（均统一为浮点型格式，其中“—”表示原文未提供相关数据及无法算得数据）。



(a) 解决 Eil51 实例



(b) 解决 CHN144 实例

图 12 三种算法解决 TSP 的收敛特性

表 5 Eil51 实例的对比实验结果

算法	平均收敛时间/s	平均值	最优解	最差解	E_e	E_t	E_f
遗传算法	8.32	453.58	442.70	471.48	5.76%	48.94%	6.71%
基本蚁群算法	6.54	442.98	437.47	459.06	3.29%	38.47%	5.03%
最大最小蚁群算法	—	432.27	428.87	—	0.79%	—	—
自适应蚁群算法	—	431.46	428.87	—	0.60%	—	—
DACA 算法	2.19	430.52	428.87	436.61	0.38%	12.88%	1.80%

表 6 CHN144 实例的对比实验结果

算法	平均收敛时间/s	平均值/km	最优解/km	最差解/km	E_e	E_t	E_f
遗传算法	167.4	33245.6	32451.3	34576.8	9.53%	76.09%	7.00%
基本蚁群算法	119.2	32115.9	31985.6	33186.1	5.80%	54.19%	3.96%
遗传蚁群算法	—	30612.0	30354.0	31153.0	0.85%	—	2.63%
DACA 算法	71.5	30510.5	30353.9	30914.6	0.52%	32.50%	1.94%

从图中曲线和表中数据可知，DACA 得到了两个实例的已知最优解（Eil51 为 428.87175，CHN144 为 30353.860997），且三项性能指标均好于其他算法，说明 DACA 的收敛速度快、误差小、波动小，即 DACA 能够迅速平稳的收敛到较优值。

2.1.3.4 算法总结

针对基本蚁群算法易陷入局部极值，收敛速度慢等问题，在分析蚁群算法特点的基础上提出了基于知识库的动态蚁群算法。最后通过对比实验验证了 DACA 在优化性能、时间性能和鲁棒性三个方面的优势。

由于智能优化算法在结构上存在相似性，都要经历产生初始解、根据优化搜索策略产生新解、根据适应度函数选择较优解、多次迭代并逐渐收敛等过程，只是在不同算法中实现方式有所区别。已有知识为算法参数的选择提供了指导，对模型最优解应满足的规则给出了判断，将这些知识与智能优化算法结合可以有效提高算法收敛速度、降低误差、减少波动。因而基于知识库指导蚁群算法寻优的思想可以扩展到其他智能优化算法之中，用基于知识库的动态智能优化算法解决不同领域的实际问题。

2.1.4 基于自学习的群智能算法研究

以 2.1.2 和 2.1.3 两节中提到的基于知识库的推理方法和基于知识库的改进蚁群优化算法都需要人为的建立知识库为前提。若考虑在复杂产品参数优化过程中，基于已有的样本数据特点，并利用 BP 神经网络的学习能力，指导优化过程，那么优化算法的优化效率和应用范围就会变得更加广泛。因此，本文对基于自学习的群智能算法开展了研究。

2.1.4.1 粒子群算法研究

粒子群算法其实也是一种演化计算技术，该算法将鸟群运动模型中的栖息地类比于所求问题空间中可能解的位置，通过个体间的信息传递，导引整个群体向可能解的方向移动，在求解过程中逐步增加发现较好解的可能性。群体中的鸟被抽象为没有质量和体积的“粒子”，通过这些“粒子”间的相互协作和信息共享，使其运动速度受到自身和群体的历史运动状态信息的影响。以自身和群体的历史最优位置对粒子当前的运动方向和运动速度加以影响，较好地协调粒子本身和群体之间的关系，以利于群体在复杂的解空间中进行寻优操作。

粒子群优化算法的数学描述如下：

假设在一个 D 维的目标搜索空间中，有 N 个代表潜在问题解的粒子组成的一个种群$\mathrm{S}=\{\mathrm{X}_1,\mathrm{X}_2,\cdots,X_n\}$，其中 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$。将 Xi 代入一个与求解问题相关的目标函数可以计算出相应的适应值。用 $P_i=(p_{i1},p_{i2},\cdots,p_{iD})$ 记录第 i 个粒子自身搜索到的最优点（所谓的最好，是指计算得到的适应值最优，即 pbest）。而在这个种群中，至

少有一个粒子是全局最优的，将其编号记为 g，则$P_g=(p_{g1},p_{g2},\cdots,p_{gD})$就是种群搜索到的最优值（即 gbest）。而每个粒子还有一个速度的变量，可以用$V_i=(v_{i1},v_{i2},\cdots,v_{iD})$表示第 i 个粒子的速度。PSO 算法一般采用下面的公式对粒子进行操作：

$$v_{id}^{t+1}=w^t v_{id}^t+c_1 r_1^t(p_{id}^t-x_{id}^t)+c_2 r_2^t(p_{gd}^t-x_{id}^t) \tag{2.4}$$

$$x_{id}^{t+1}=x_{id}^t+v_{id}^t \tag{2.5}$$

在公式中$d\in(1,2,\ldots,D)$，$i\in(1,2,\ldots N)$，N 为种群中粒子个数，上标 t 显示的是迭代次数，w 为惯性权重，r1、r2 为取值在[0,1]之间的随机数，c1、c2 为正常数，为了控制 V_i 和 X_i 的值在合理的区域内，需要指定 Vmax 和 Xmax 来限制。

粒子群算法概念简单、容易实现、搜索范围大，和其他优化算法相比，它的优点突出。已成功应用于函数优化、神经网络等不同的领域。作为一种随机优化算法，也具有局部收敛，进化后期收敛速度慢，精度较差等缺点。近些年来，对改善 PSO 的性能提出了一些解决方案，并取得一定的效果，基本可归纳成三类：第一类是通过研究算法的参数来改善粒子群的性能；第二类是 PSO 与其他进化方法结合的研究；其中在粒子群进化后期对最优个体进行变异，使其离开局部最优的变异粒子群方法，取得了较好的效果；第三类是研究协作和竞争方法的粒子群。这些方法都在一定的程度上对算法的性能有所改善。

2.1.4.2 神经网络算法研究

误差反向传播网络（Back Propagation，简称 BP 网络）是典型的前馈网络，它是人工神经网络中最为重要的网络之一，也是迄今为止应用最广泛的神经网络算法，实践证明，三层结构 BP 网络即具有很强的映射能力，三层 BP 网络拓扑结构图如图 13 所示。

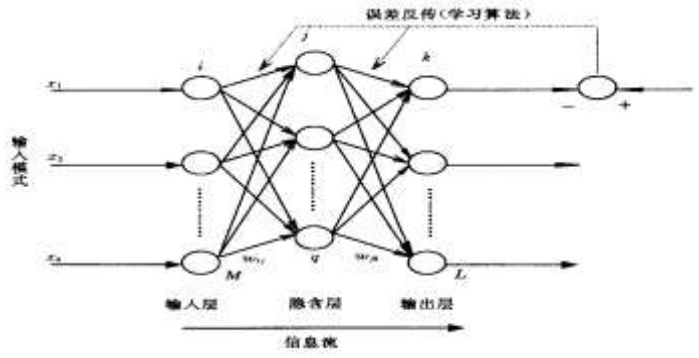


图 13 BP 神经网络结构

BP 算法是一种基于优化技术的算法，其基本思想是最小二乘算法。它采用梯度搜索技术，以期使网络的实际输出值与期望输出值的误差均方值为最小。B-P 算法的学习目的是对网络的连接权值进行调整，使得调整后的网络对任一输入都能得到所期望的输出。BP 算法学习过程可以描述如下：

（1）工作信号正向传播。输入信号从输入层经隐层传向输出层，在输出端产生输出信号，这就是工作信号的正向传播。在信号的向前传递过程中网络的权值是固定不变的，每一层神经元的状态只影响下一层神经元的状态，如果输出层不能得到期望的输出，则转入误差信号反向传播。

（2）误差信号反向传播。网络的实际输出与期望输出之间差值即为误差信号，误差信号由输出端开始逐层向前传播，这是误差信号的反向传播。在误差信号反向传播过程中，网络的权值由误差反馈进行调节，通过权值的不断修正使网络的实际输出更接近期望输出。

（3）反复迭代，直到满足要求或达到最大次数。

2.1.4.3 基于自学习的群智能算法

2.1.4.3.1 算法介绍

本文提出了一种新型的基于自学习的群智能算法（PBPO，Partical Back Propagation Neural Network Optimization），采用了粒子群算法中的一些基本概念。算法的基本思想是存在一个群体，其中包含了多个粒子，每个粒子的位置都是空间中的一个可行解，通过对粒子位置的移动寻找最优解。在移动过程中粒子之间能进行信息的交互，每个粒子有对自身历史知识的记忆，使用神经网络根据粒子的现有知识移动粒子位置，使粒子不断的向解空间中的最优位置移动。神经网络在调整粒子位置时使用的知识主要有群体中所有粒子的历史最好位置，每个粒子在移动中的自身历史最优位置。

人工神经网络具有学习能力，只要输入一系列样本并加以训练，就能获得训练数据库中所具有的控制关系。输入和输出之间的映射关系是非线性的，无需研究各因素之间的作用机理和关系。BP 神经网络(Back - Propagation Neural Network) 模型在诸多的人工神经网络模型中，应用最为广泛，可用于函数逼近、模式识别、分类等。BP 神经网络模型的基本思想是通过多层前馈计算输出，采用误差反向传播修正学习，属于多层前馈反向传播网络。

将 BP 神经网络用于种群中粒子的位置移动，其中神经网络的输入为粒子的当前位置，输出为粒子的调整后位置，而神经网络的输出误差为粒子当前位置与种群中所

有粒子的历史最好位置的误差。PBPO 使用的 BP 神经网络模块的基本结构如图 14：

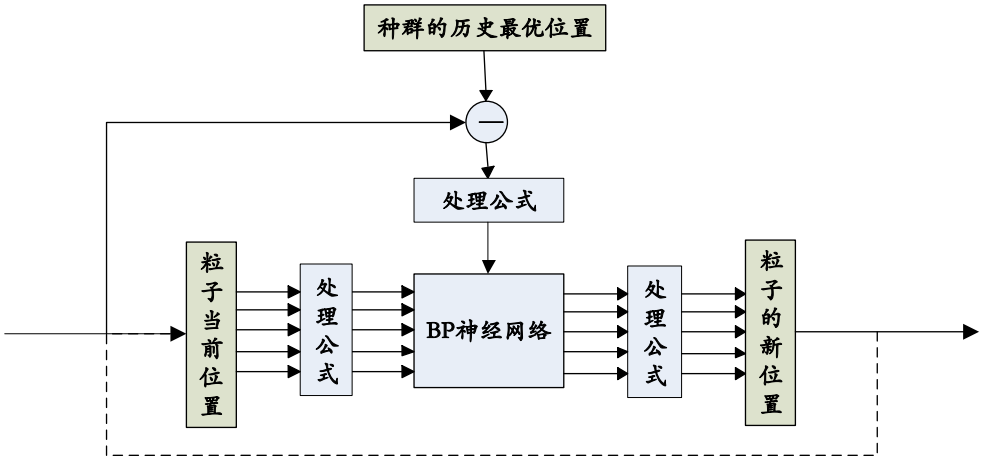


图 14 BP 神经网络模块

图中的 BP 神经网络模块总共有三层，分别为输入层、输出层、隐含层。输入层与输出层神经元数目为解空间的维数，隐含层的神经元数目为输入层神经元数目加 1，神经元的激活函数采用 Sigmoid 函数 $f(x)=\dfrac{1-e^{-x}}{1+e^{-x}}$，其值域为（-1，1）。由于神经网络的值域为（-1，1）之间，所以输入要经过归一化处理，将粒子的当前位置根据归一化公式（2.6）调整到（-a，a）之间，将处理后的粒子位置作为神经网络的输入，神经网络的输出经过归一化公式（2.7）处理后作为粒子的新位置。公式(2.3)(2.4)中的 a 为归一化因子，其取值范围在 0 到 1 之间，随着 a 的取值不断增大，函数寻优过程中的收敛速度会越来越大，但随着收敛速度的加大，函数的收敛效果会越来越差。归一化因子主要实现了搜索空间的移动。

$$\hat{x}=\frac{x-x_{\min}}{x_{\max}-x_{\min}}\times 2a-a, 0<a\le 1 \tag{2.6}$$

$$y=(\hat{y}+a)/(2a)\times(x_{\max}-x_{\min})+x_{\min}, 0<a\le 1 \tag{2.7}$$

为了使搜索具有更好的性能，将 $x_{\min}$， $x_{\max}$ 通过极值调整公式（2.8）进行处理，公式（2.8）中的 c 为极值调整因子，当 c 取值过大或过小时都会导致算法的收敛速度慢或收敛效果差，一般取值在 1 或 2 之间效果比较好，通过对极值调整因子的调整实现了粒子搜索空间的移动以及对当出现 $x_{\max}=x_{\min}$ 情况时的处理。

$$\begin{cases} x_{\min}=x_{\min}/c \\ x_{\max}=x_{\max}\times c \end{cases}, 1<c<2 \tag{2.8}$$

在 PBPO 算法中，每一个粒子对应一个 BP 神经网络，粒子的位置在神经网络模块中不断迭代，直到粒子的新位置优于此粒子个体最优位置时或达到神经网络模块的最大迭代次数时才停止迭代，从而完成了对粒子位置的一次修改。神经网络的权值在种群的迭代中不断调整，使得它越来越熟悉所属粒子的特性，从而能更好的进行寻优。

神经网络模块的基本思想是通过对权值的训练使粒子的位置不断向全局最优靠近，但只要调整到优于粒子本身的历史最优位置或到达最大迭代次数时就停止迭代，又保持了一定的随机性，从而使粒子的位置不断的向最优解靠拢。算法的流程图 15 所示：

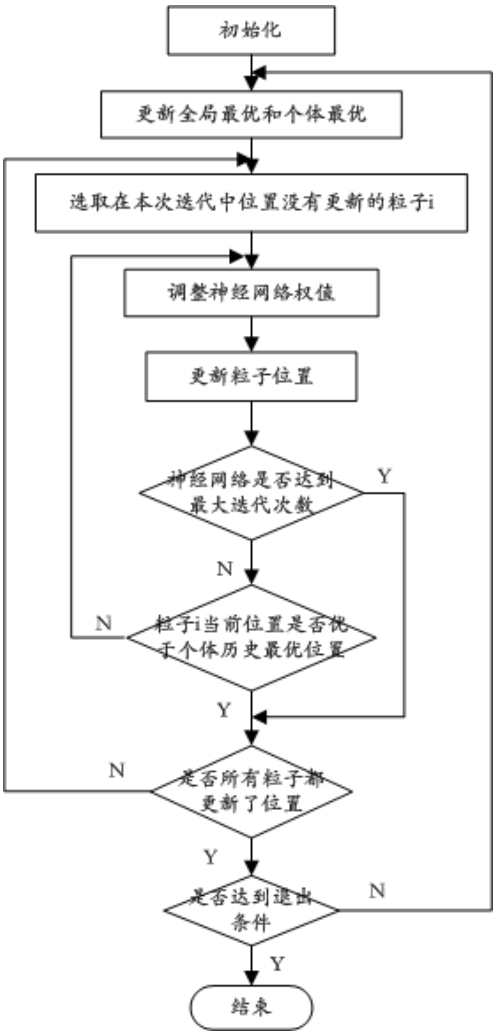


图 15 算法的流程图

从图中可以看出，整个算法分为两部分迭代，第一部分是神经网络模块的迭代，第二部分是种群的迭代，所以 PBPO 算法的迭代次数为两部分迭代次数的乘积。但由于在神经网络模块中每个粒子的迭代次数都不一样，因此使用神经网络的最大迭代次数作为神经网络部分的迭代次数。

PBPO 的算法步骤如下：

（1）初始化种群，设定最大迭代次数T_{PSO}，设定粒子数目 m，对每一个粒子 i，在解空间内随机初始化粒子位置x_i，初始化一个神经网络BP_i，神经网络的权值随机初始化，设定神经网络的参数和最大迭代次数T_{BP}。

（2）计算每个粒子的位置的适应度，根据适应度更新粒子的个体历史最优位置与种群的历史最优位置，种群的迭代次数加 1。

（3）对种群中的每一个粒子$i, i = 1 \cdots m$，进行步骤（4）-（6）。

（4）对粒子 i，计算其当前位置与种群历史最优位置的偏差，将偏差值作为此粒子对应的神经网络BP_i的输出误差，神经网络根据输出误差调整权值。

（5）将粒子 i 的当前位置作为 BP 神经网络的输入，神经网络的输出作为粒子 i 的新位置。神经网络模块迭代次数加 1，判断神经网络模块是否达到最大迭代次数T_{BP}，若是，则转步骤（7）。

（6）判断粒子 i 的新位置是否优于粒子 i 的历史最优位置或种群的历史最优位置，若否，则转步骤（4）。

（7）判断是否达到结束条件或到达最大迭代次数T_{PSO}，若否，则转步骤（2）。

2.1.4.3.2 改进算法测试

（1）与群智能算法的纵向比较测试

为了证明算法的有效性，将 PBPO 算法与群智能算法中的粒子群类算法相比较。选用了 3 个经典测试函数进行测试，并与标准 PSO 算法以及 CHOPSO 和 HAEPSO 算法进行比较。本次测试所用 3 个测试函数分别为：

Rosenbrock

$$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2] \quad x_i \in (-100,100) \tag{2.9}$$

Griewank

$$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1 \quad x_i \in (-600,600) \tag{2.10}$$

Rastrigrin

$$f(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i^2)+10] \quad x_i\in(-10,10) \tag{2.11}$$

在对每个函数的寻优测试中，每次测试算法共迭代 1000 次，总共测试 20 次，函数的解空间为 30 维。共使用了 6 个参数来衡量算法的好坏，分别是：

最小值：在所有 20 次测试中得到的最优适应值。

最大值：在 20 次测试中得到的最差的适应值。

平均值：20 次测试得到的适应值的平均。

中间值：20 次测试中的中间适应值，有一半的适应值高于此值。

迭代数：达到各个函数的给定标准所需要的平均迭代次数。

成功率：在 20 次测试中成功运行的次数在所有测试中所占的百分比。成功运行指在 1000 次迭代中达到了各个函数的给定标准（见表 7），即函数的最适应值小于等于表 7 中的数据。

表 7 函数的给定标准

Rosenbrock	Griewank	Rastrigin
100	0.01	100

在测试中粒子的种群规模为 50，各个粒子的初始位置在定义域内随机初始化，BP 神经网络的学习速率取 0.7。动量因子取 0.3，神经网络的最大迭代次数为 10 次，归一化公式（2.6）(2.7)中的 a 取 0.3，极值调整公式（2.8）中的 c 取 1.1。在测试中的最大迭代次数为 1000 次，由于 PBPO 的神经网络最大迭代次数为 10 次，所以在测试中 PBPO 设置的最大迭代次数为 100 次。测试结果如表 8 所示，图 16 为 PBPO 算法的适应度随着迭代次数收敛曲线。

表 8 测试结果

函数名	参数	PSO	HAEPSO	CHOPSO	PBPO
ROSENBROCK	最小值	180	28.7	27	3.04E-4
	最大值	19675	28.7	34	3.08E-1
	平均值	4942	428.7	28	6.25E-2
	迭代次数	1000	-	44	75
	成功率	0%	-	100%	100%
GRIEWANK	最小值	0.01	1.62E-9	0	0

	最大值	0.07	6.06E-8	3.00E-15	0
	平均值	0.02	2.07E-8	3.00E-16	0
	迭代次数	935	-	45	107
	成功率	90%	-	100%	100%
RASTRIGIN	最小值	22.62	4.43E-6	0	0
	最大值	60	1.4E-4	0	0
	平均值	54.05	5.1E-5	0	0
	迭代次数	800	-	27	56
	成功率	50%	-	100%	100%

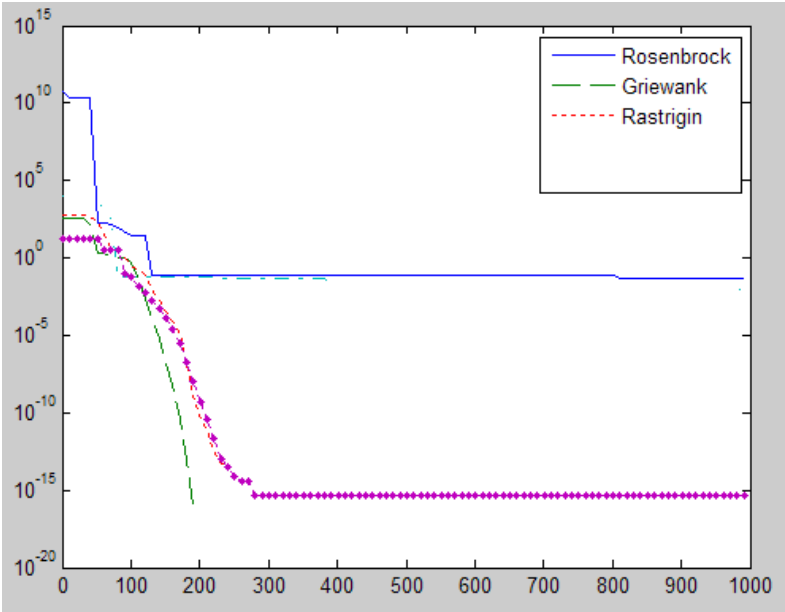


图 16 函数收敛图

从实验结果看出，PBPO 在 3 个函数的寻优中没有一次失败，说明了算法具有很好的收敛能力。PBPO 算法在 1000 次迭代后的平均值没有大于 1 的情况出现，这是因为 PBPO 算法使用 BP 神经网络对粒子进行移动，因此减弱了寻优效果与被寻优函数本身特性之间的关系，从而对于一些由于自身结构而不容易进行寻优的函数如 Rosenbrock 函数，表现出了良好的寻优性能。

（2）算法横向比较测试

为测试 PBPO 的综合寻优性能，将 PBPO 算法与遗传算法进行比较。选择 J.D. Schaffer 提出的函数作为基准函数，如下：

$$f(x,y)=\frac{\sin^2(\sqrt{x^2+y^2})-0.5}{\left[1+0.001\left(x^2+y^2\right)\right]^2}-0.5 \quad x,y\in[-100,100] \tag{2.12}$$

该函数理论上的最优状态和最优值为 $\min(f(x^*)) = f(0,0) = -1$。此函数在距离全局最优点大约 3.14 范围内存在无穷多个局部极小值将其包围，并且函数强烈振荡。将 PBPO 算法与传统遗传算法，以及完全多样化成长策略的遗传算法进行横向比较，总共进行 10 次测试，PBPO 的神经网络最大迭代次数设为 10 次，粒子的数目为 100，归一化公式（2.6）（2.7）中的 a 取 1，极值调整公式（2.8）中的 c 设为 1.5。测试结果如下，表 9 中的测试结果为在 10 次测试中的最差结果。

表 9 算法时间测试结果

	指标	完全的多样化成长策略的遗传算法	PBPO	传统遗传算法
最差结果	运行时间（s）	14.1	1.23	1.5
	迭代次数	333	220	391
	最优解	x=-4.768e-5 y=4.768e-5	x=-6.43e-9 y=-6.43e-9	x=2.022 y=-2.041
	目标函数值	-1.000000	-1.000000	-0.990284
	正确率	100	100	30
	理想目标值	-1.000000	-1.000000	-1.000000

从表中可以看出，与传统遗传算法相比，PBPO 的寻优效果非常好，虽然加入了神经网络模块，但耗费的时间并不长，而得到了比较好的结果，而与完全的多样化成长策略的遗传算法相比，无论是在收敛时间还是收敛效果都有明显的提高，从而证明了 PBPO 的良好的寻优能力。

（3）算法时间复杂性测试

使用 GRIEWANK 函数对算法在不同解空间维数下的运行时间进行测试，算法的设置同上，测试的时间为达到表 7 中的给定标准所需时间。测试结果如图 17 所示。

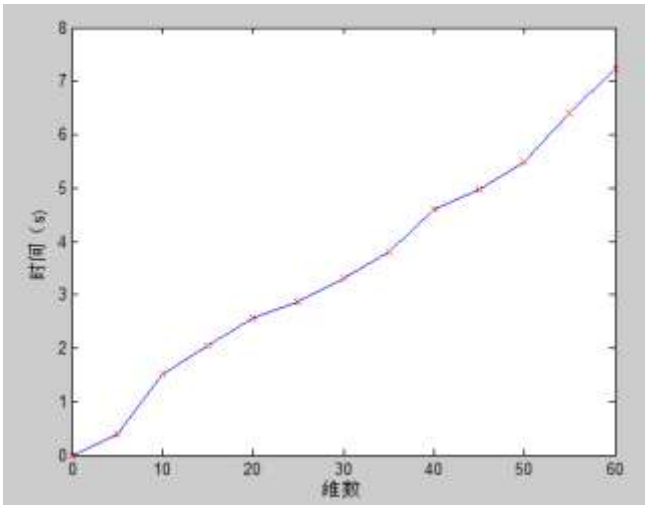


图 17 维度时间变化曲线

从图中可以看出，随着解空间维度的增加，算法的寻优时间基本上是线性变化的，说明算法并不会随着解空间维数的变化而剧烈的变化，算法的计算时间为线性阶的。

（4）测试结论

本论文提出的 PBPO 算法吸收了群智能的基本思想，采用了粒子群算法的基本概念，提出了使用 BP 神经网络移动粒子位置，实现了群智能方法和梯度下降方法的结合，是一种新型的寻优方法。从算法的测试结果可以看出，PBPO 对函数的寻优与传统的算法相比有了很大的提升，寻优的效果与被寻优函数的结构关系不大，在收敛精度和寻优成功率方面表现出优异的性能。

2.1.5 定性定量结合方法的进一步研究思路

通过以上几节可以得出，QSIM 算法是在系统能用定性微分方程(QDE) 表示的基础上进行定性推理，而对于某些特殊的复杂产品设计，定性微分方程难以完全描述系统的特性，基于 QSIM 算法的定性推理体现出了一定的局限性。另外，在复杂产品设计过程中，对于待解决的问题，由于以往的成功案例或经验缺乏，基于经验或人工估算的知识库难以构建，但在实际问题中，定性的知识又是解决问题不可缺少的前提。如何从大量的实验数据中提取出定性知识，将这种用自然语言表述的定性知识转化为定量的数学模型是解决复杂产品设计问题的关键。

云是用语言值表示的某个定性概念与其定量表示之间的不确定性转换模型，构成定性和定量间的映射。云用期望、熵、超熵三个数字特征来整体表征一个概念，它不仅体现了模糊推理的模糊性，同时考虑了随机性。

云模型是云具体实现的方法，由定性概念到定量表示的过程，也就是由云的数字

特征产生云滴的具体实现，称为正态云发生器；由定量表示到定性概念的过程，也就是由云滴群得到云的数字特征的具体实现，称为逆向云发生器。常用的云模型如下图18 所示。

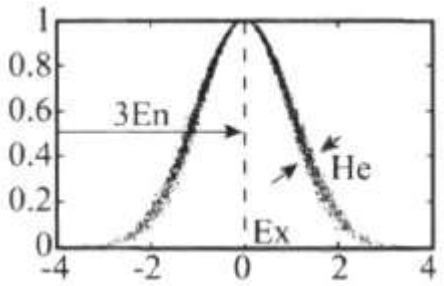


图18　正态云

正态云发生器是从定性概念到定量的映射，它根据云的数字特征（E_x， E_n， H_e）产生云滴，其中E_x为期望，是最能够代表定性概念的点；熵E_n定性概念的不确定性度量，由概念的随机性和模糊性共同决定；超熵H_e是熵的不确定性度量，即熵的熵。由熵的随机性和模糊性共同决定。如图 19 所示。逆向云发生器是实现从定量值到定性概念的转换模型。它可以将一定数量的精确数据转换为以数字特征（E_x， E_n， H_e）表示的定性概念，如图 20 所示。

图 19 正态云发生器

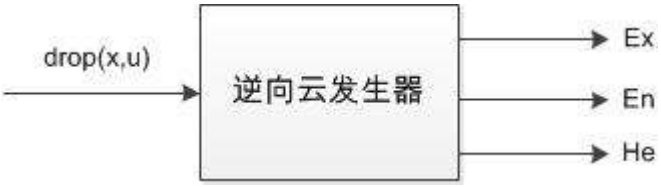
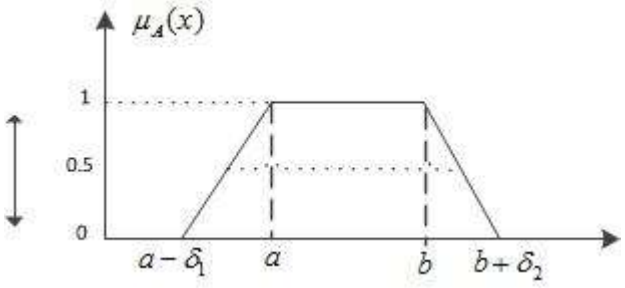


图 20 逆向云发生器

基于模糊数学实现定性数学约束到定量数值的转换，使用常见的模糊四元组$[a, b, \delta_1, \delta_2]$来表示模糊数，如图 21 所示，其中，$\mu_A(x) \in [0,1]$为截集水平。$\mu_A(x)$越小，表示对参数的接受范围越大。采用这种描述构成的量空间可以在集合与值之间构成一种联系。如实数 3.5 转换成模糊数[3.5,3.5,1,1]，这样就将一个具体的数值转换成了一个凸函数。模糊数学的运算法则包括加法、减法、乘法、除法和取负等。



$$\mu_A(x)=\begin{cases}0 & x<a-\delta_1\\ \delta_1^{-1}(x-a+\delta_1) & a-\delta_1\le x<a\\ 1 & a\le x<b\\ \delta_2^{-1}(b+\delta_2-x) & b\le x<b+\delta_2\\ 0 & x\ge b+\delta_2\end{cases}$$

图 21 模糊四元组

经验就是知识，这种知识带有不确定，但是比数学表达更真实、更具有普遍性，由于定性概念和定量数据之间普遍存在着不确定性，尤其是随机性和模糊性。李德毅院士提出的云模型是用自然语言值表示的定性概念与其定量数据表示之间的不确定性转换模型，反映了客观世界中事物或人类知识中的模糊性和随机性，并把二者完全集成在一起，构成了定性和定量间的相互映射。同时，当定性知识是设计变量之间定性数学约束时，可以使用模糊数学实现定性数学约束到定量数值的转换。因此如何利用云模型和模糊数学实现定性与定量之间的转换，是下一阶段研究的重点。

2.2 发表论文及申请专利

1．自项目获批起至 2011 年间发表的论文

[1]董丽丽，龚光红，李妮，孙勇，基于云模型的自适应并行模拟退火遗传算法，北京航空航天大学学报，2011, 37(9): 1132-1136. （EI 检索号 20114514503987）

[2] Li Ni, Gong Guanghong, Sun Yong, Li Liang, Study on Meta-synthetic Technology of SBA. 2010 International Conference on Future Information Technology, 2010, 220-223.

[3] Deng Yingcan, LI Ni, Particle Back Propagation neural network Optimization Algorithm for CMTA, 2010 International Conference on Future Information Technology, 2010, 425-428.

[4] Li Ni，Tang Liyong. RESEARCH ON MILITARY SCENARIO EDITOR AND PLAN VIEW DISPLAY BASED ON SIMULATION MODELS[A]. 2012 International Conference on Communication and Electronics Information.(已录用)

2．自项目获批起至 2011 年间提交并获批的专利

[1] 李妮, 高栋栋, 龚光红, 韩亮. 基于 TBB 的多核并行蚁群设计方法, 受理（申请号：2009100771198；公开号：CN101464965）

[2] 李妮, 邓英灿, 龚光红, 马耀飞. 一种空战决策的粒子群优化方法, 受理（申请号：2010102287582）

[3] 李妮, 董丽丽, 龚光红. 基于线程构造模块的多核并行模拟退火方法, 受理（申请号：2009100874796）

2.3 研究生培养情况

2011 年已完成与本自然基金项目有关的博士及硕士学位论文答辩：

1）孙勇，博士学位论文题目：“智能建模仿真语言关键技术研究”（2011 年 5 月通过答辩）。

2）宫庆义，硕士学位论文题目：“虚拟采办综合技术研讨厅关键技术研究”（2011 年 12 月通过答辩）。

3. 下一年度工作计划

3.1 研究工作

按项目计划任务书的要求完成以下研究工作：

1）研究定性定量信息的转换方法，实现定性推理仿真和并行智能优化定量仿真之间的反复优化迭代过程；

2）研究混合优化算法研究和基于 TBB 平台的并行实现，完成软件编程；

3.2 发表论文及提交专利

发表论文 1～2 篇， 提交专利申请 1～2 项。

3.3 完成年度总结报告

4. 当年经费使用情况与下一年度经费预算

4.1 当年经费使用情况

2011 年资助经费到款 20 万元，使用情况如下：

科研业务费 3.3 万元；实验材料费 0.2 万元； 仪器设备费 0.8 万元；

国际交流 0.9 万元； 劳务费 1.44 万元；管理费 0.3 万元。

2. 下一年度经费预算：

本项目经费已经全部到位，2012 年度经费预算如下：

科研业务费 3.5 万元；实验材料费 0.3 万元； 仪器设备费 1.0 万元；
国际交流 0.6 万元； 劳务费 1.0 万元；管理费 0.3 万元。

5．附件

标注资金资助的已发表和已有录用通知的论文：

项目负责人签字及部门审核意见表

项目负责人承诺：

我所承担的项目（编号：61004089 名称：定性定量结合的并行智能优化仿真方法的研究与应用）《进展报告内容》实事求是，数据详实。我在下一步研究中将认真工作并及时报告重大情况变动等。

项目负责人（签章）：

日期：

项目依托单位科研管理部门审查和对今后工作的意见：

经办人（签章）：　　单位公章：

日期：

科学处审核意见：

1．计划完成情况（在□内打✓，例☑）

□ 按原计划完成任务

□ 基本按原计划完成任务

□ 未完成原计划任务

2. 经费拨款情况（在□内打✓）：

□ 按计划拨款

□ 暂缓拨款

□ 中止拨款

3. 其它意见：

负责人（签章）：

日期：

科学部核准意见（对重点、重大、国家杰出青年科学基金等及拨款情况发生变化的项目）：

负责人（签章）：