

基于机器学习的本体概念相似性研究^{*}

欧 灵^{1,2} 张玉芳¹ 吴中福¹ 钟 将¹

(重庆大学计算机学院 重庆 400044)¹ (西南大学计算机科学系 重庆 400715)²

摘 要 现有的知识系统使用的是集中式的、一致性的、可扩充的 Ontology 库,不同本体间的语义匹配是语义网发展面临的最富挑战性的问题之一。本文针对领域中存在不同的 Ontology 的问题,讨论了一种基于多策略机器学习的 Ontology 匹配方法,重点分析了本体概念的相似度计算,并提出了一种相似度测量算法。

关键词 本体,语义网,匹配,相似

Research on Ontology Concept Similarity Based on Machine Learning

OU Ling^{1,2} ZHANG Yu-Fang¹ WU Zhong-Fu¹ ZHONG Jiang¹

(College of Computer, Chongqing University, Chongqing 400030)¹

(Dept. of Computer Science, Southwest University, Chongqing 400715)²

Abstract Current knowledge system is using a centralized, persistent, scalable ontology repository. Mapping between semantic heterogeneous ontologies among semantic Web is becoming an ever more important issue. In this paper, an ontology mapping system is described, which applies multiple classification methods to learn the mapping between ontologies, and then the similarity measuring method between different concepts which belongs to different ontology respectively is analyzed. As a result, the algorithm of similarity measure is brought forward.

Keywords Ontology, Semantic Web, Mapping, Similarity

1 引言

1998 年, Berners. Lee 提出了 Semantic Web 的概念,语义网正成为下一代互联网的研究热点。语义 Web 是现有 Web 的扩展,语义 Web 提供了一个通用的框架,使得跨越不同企业的应用程序和软件能够共享和重用数据。语义 Web 所描述的信息被赋予了定义良好的语义,这大大方便了计算机与计算机以及计算机和人的协同。

有了大量富含语义信息的网页,就好像有了一个巨大的全球互联的数据库。若机器能理解网页内容和网络服务,必将实现更大范围和更深层次的网络资源共享。同时软件在语义信息的帮助下智能将大大提高。

为真正发挥语义 Web 的力量,全球语义共享是目标。为此,系统间语义的互联是亟待解决的问题,Ontology 是语义网体系结构中的描述语义的核心层,从某种程度上看,语义共享就是 Ontology 间概念匹配和语义匹配的问题,只有实现语义共享,语义网才会得到广泛的应用。

2 Ontology 的匹配问题示例

1991 年 Neches 等人在人工智能界最早提出 Ontology 的概念^[2]。1993 年, Gruber 给 Ontology 的定义是“Ontology 是概念模型的明确的规范说明”^[6]。Studer 等人总结的 Ontology 的四层含义是:概念模型(conceptualization)、明确(explicit)、形式化(formal)和共享(share)^[7]。

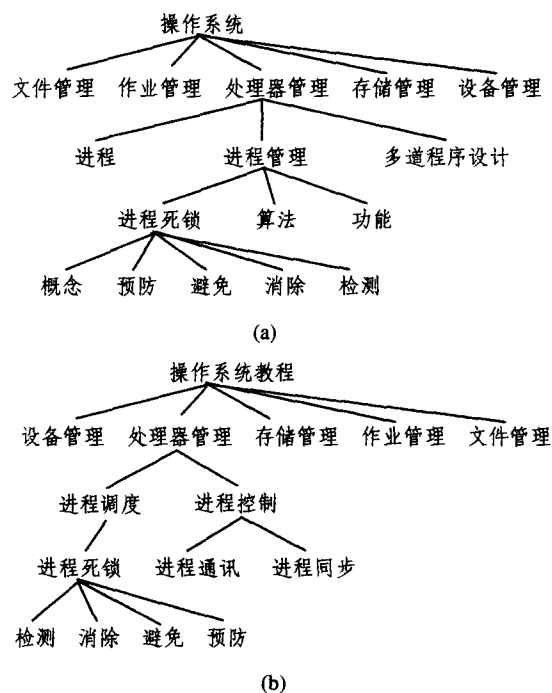


图 1 操作系统课程概念本体分类树示例

同一领域存在着大量不同的语义异构的本体,它们分别支撑各自系统的运行。W3C 为规范和标准化本体开发语言作了大量的工作,RDF(Resource Description Framework)是描述元数据以及元数据与元数据之间关系的有力工具,它为

^{*} 项目资助: CNGI 远程教学公用通信平台系统(101048620060012)2005 年度国家发改委科学研究计划项目资助。欧 灵 副教授, 博士研究生, 主要研究方向为语义网、网络计算、网络安全、计算机远程教育。张玉芳 副教授, 主要研究方向为计算机远程教育、网络计算。吴中福 教授, 博士生导师, 主要研究方向为计算机网络与通讯、网络安全、网络计算。钟 将 讲师, 主要研究方向为网络安全、免疫计算。

在应用程序间交换机器可理解的网络数据提供了互操作性, OWL(Web Ontology Language)更为 Ontology 的自动推理提供了标准统一的本体描述语言。但本体间的语义匹配仍然是最富挑战性的问题之一(W3C)。

Ontology 通过一组概念、属性和关系对一个领域进行规范化的描述,概念是领域中感兴趣的实体的抽象,我们可以用一棵分类树来模拟描述一个 Ontology^[3],例如,图 1 是网络教育中两套不同的操作系统课程教学系统的 Ontology 概念分类树。

分类树中的节点是概念,每个概念都用同一组实例来支撑说明。比如:进程死锁“检测”的具体示例、课件(其属性可用元数据描述标准 Dublin Core 描述)等就是“检测”这一概念的数据实例;当然这些实例也可以作为“检测”这一概念的祖先概念的实例,如“进程”的实例。

每个概念也可以用一组属性来描述它,如: Dublin Core 属性组(Title:资源的名字; Author or Creator:资源内容的创建者或组织者; Subject and Keywords:资源的主题及关键字; Description:资源内容的原文描述)。

同时概念之间、概念和实例之间的关系可以用一组关系来描述,如:用 is-a, is-part-of, name-of, attribute-of, instance-of 等用来说明这些关系。

考察这样的情形: a 学校使用的是图 1-a 所示 Ontology 解释的教学系统, b 学校使用的是图 1-b 所示 Ontology 解释的教学系统,如果 a 校的一个学生正在网上学习操作系统这门课程的“进程死锁”部分,他想学习更多的有关死锁知识,他可以通过某种办法方便准确地找到 b 校教学系统的相关部分进行学习,这就是 Ontology 匹配的问题。

给定两个相同领域的 Ontology, Ontology 的匹配问题就是去发现它们之间的语义匹配关系,最简单的匹配是 1 对 1 的匹配,如: a 图的“进程死锁”对 b 图的“死锁”;还有 1 对多,多对 1 的匹配,如: a 图的“处理机管理”对应 b 图的“进程调度”和“进程控制”;除此之外也存在多对多的匹配,如 a 图的“进程”和“进程管理”对应 b 图的“进程控制”和“进程调度”。

对于 Ontology 匹配的研究存在各种不同的手段和工具,其中基于机器学习的研究途径一般有几个关键步骤,机器学习、相似度评估、规则指导下的匹配标记以及相容性检测等,其中,概念相似度评估是极其重要的一步^[4]。为简化问题的复杂性,本文所讨论的相似度评估主要针对 1 对 1 的匹配问题。

相似度评估包括结构相似性、目标特征相似性、个体相似性。相似性计算可以根据属性值的相似度计算绝对值距离、欧氏距离或麦考斯基距离来实现。但是,这样的相似度计算在反映概念的语义上存在局限。

本文进行的相似度评估是在分析实例相似的基础上,根据相似实例数来计算概念相似度。

3 定义和术语

本文中涉及的定义和术语都是针对领域本体定义的。

领域本体(Domain Ontology):领域本体是在一个特定的领域中,提供对该领域特定的概念、术语、词汇的定义,并提供对概念之间关系的精确描述。实例是概念的具体化。同时领域本体还包括对领域中的主要理论和基本原理的准确描述。目前很多领域都开发有领域本体,如:医学概念本体,计算机科学领域知识的本体等等^[8]。通常领域 Ontology 可以用一

个概念分类树来简化表示^[5],如图 1 所示。

定义 1 T 是一个领域 Ontology 分类树,用一个 5 元组描述:

$$T=(N,R,H,S,X)$$

其中:

N 表示概念的集合,是树中的节点;

R 是一个关系集合,是树中的边,如:用 is-a, is-part-of, name-of, attribute-of, instance-of 等用来说明概念间的关系;

H 描述树中概念间以及概念与实例间的层次分类关系;

S 是实例的集合;

X 是描述概念节点及实例的元数据属性集合,如: Dublin Core。

为简单化概念树的复杂结构和概念与概念间的复杂关系,作如下合理的假设,相关的问题留到匹配标记时进行考虑。

假设 1:概念树是一棵有根数,并且任意子树也是有根树。同一个实例 s 不能同时分配到不具父子关系的两个概念之下。

假设 2: s 是概念 C 的实例,若: $C < C'$ (表示 C 是 C' 的子孙),那么 s 也是概念 C' 的实例。

定义 2 概念 C 是实例 s 的最近概念节点:如果实例 s 被分类到节点 C 下,节点 C 是 s 最近的概念节点必须满足条件:

s 是 C, C' 的实例, $C \in N$, 对于 $\forall C' \in N$ 有:

$$C < C'$$

定义 3 映射 M : 对于领域中的两个领域本体 Ontology 概念树 $T=(N,R,H,S,X), T'=(N',R',H',S',X')$,

$$M: N \times N' \rightarrow \text{ref}$$

其中, ref 是一个可以接受的解释语义关系的名词集合。

定义 4 两个领域 Ontology 中的概念节点 n, n' 间存在如下四种关系:

$$1, n < n' \Rightarrow \text{若 } \forall s \in H(n \downarrow) \text{ 那么 } s \in H'(n' \downarrow)$$

$$2, n > n' \Rightarrow \text{若 } \forall s \in H'(n' \downarrow) \text{ 那么 } s \in H(n \downarrow)$$

$$3, n \perp n' \Rightarrow H(n \downarrow) \cap H'(n' \downarrow) = \emptyset$$

$$4, n \equiv n' \Rightarrow H(n \downarrow) \subseteq H'(n' \downarrow) \text{ 并且 } H(n \downarrow) \supseteq H'(n' \downarrow)$$

式 1 表示 n 是 n' 的子概念;

式 2 表示 n 是 n' 的父概念或祖先概念;

式 3 表示 n 和 n' 是不同概念;

式 4 表示 n 和 n' 是相同概念。

其中: $H(n \downarrow)$ 表示 Ontology 概念树 T 中概念 n 下所有实例的集合。

考虑到两个系统的 Ontology 实例完全相同是不现实的, 本文采用实例特征相同来类似地定义相似关系, 以便寻找可能的映射匹配。

我们可以把解决语义匹配问题的过程看作是 Ontology 匹配(或者映射)的过程, 典型的映射过程涉及到分析、比较 Ontology 来判断概念之间的对应关系, 对于同一领域的两个 Ontology, 它们的元素在很大程度上是存在重叠的, 因此, 我们可以在一个 Ontology 中为另一个 Ontology 的一个概念结点寻找一个最相似的一个结点。

为找到分类树 T_a 中概念 A 在另一个分类树 T_b 中的匹配位置, 我们需要找到 T_b 中三个关键的概念节点:

1、求 T_b 中最相似的候选概念节点 B ;

2、求 T_b 最接近 A 的父节点 C ;

3、求 T_b 最接近 A 的子节点 D 。

定义5 s 是两个领域本体中的任意实例, $P(A, B)$ 表示实例 s 的特征既满足 T_a 中的概念 A 所表示的概念特征, 又满足 T_b 中的概念 B 所表示的概念特征的可能性。对于任意实例 s 以及给定的两个概念 A, B 有:

$$P(A, B) + P(A, \bar{B}) + P(\bar{A}, B) + P(\bar{A}, \bar{B}) = 1$$

定义6 T_a 中的概念 A 与 T_b 中的概念 B 的相似度可用 Jaccard 系数定义:

$$\begin{aligned} \text{Sim}(A, B) &= P(A \cap B) / P(A \cup B) \\ &= P(A, B) / [P(A, B) + P(A, \bar{B}) + P(\bar{A}, B)] \end{aligned}$$

$\text{Sim}(A, B)$ 取值在 $(0, 1)$ 之间, 0 表示 A, B 是不同的概念, 为 1 表示 A, B 是相似性最大的概念对。

定义7 若 $\text{Sim}(A, B) = \text{Max}\{\text{Sim}(A, x), \forall x \in N_b\}$, 那么, B 是 A 在 T_b 中最相似的候选节点,

其中, N_b 是 T_b 中概念节点的集合。

定义8 若 T_a 中的概念 A 下的任意一个实例 s 的特征都满足作为 T_b 中的 C 概念实例所需的特征, 那么称概念 A 是概念 C 的子概念, 称 C 是 A 的父概念或祖先概念。

定义9 记 $P(A|C)$ 是 T_a 中的概念 A 作为 T_b 中的概念 C 的子概念的可能性。

若 $P(A|C) = 1$, 即概念 A 是概念 C 的子概念;

若存在概念 A 下的一个实例 s 的特征不满足作为 C 概念实例所需的特征, 那么, $P(A|C) \neq 1$, 即概念 A 不是概念 C 的子概念。

定义10 如果 $P(A|C) = 1$, 计算 T_b 中的概念 C 下有多少实例满足 T_a 中的概念 A 的特征。记为:

$$W(A|C) = \text{Num}(S_a^{A,C}) / \text{Num}(S_c)$$

其中: $\text{Num}(S_a^{A,C})$ 表示 T_b 中的概念 C 下有多少实例满足 T_a 中的概念 A 的特征, $\text{Num}(S_c)$ 表示 T_b 中的概念 C 下的实例数。 $W(A|B)$ 取值在 $0, 1$ 之间。它直观意义是描述节点 A 与节点 C 在树中的亲近关系。 $W(A|B)$ 值越大越近。

定义11 T_b 中的概念 C 是 T_a 中概念 A 的最接近的父亲。

$$W(A/C) = \max\{W(A|x), \forall x \in N_c, \text{且 } P(A|x) = 1\}$$

直观地, 若概念节点 C 就是我们要寻找的概念 A 的匹配节点, 这里不作处理, 留到匹配标记时处理。

类似地, 可以定义 T_b 中最接近 A 的子节点 D 。这里我们不再作表述。

4 基于机器学习的 Ontology 匹配

给定两个 Ontology, 其本体分类树分别为 T_a 和 T_b , 假设我们要寻找 T_a 到 T_b 的映射, 即为 T_a 中的一个结点 A 在 T_b 中找到与其最相似的一个结点 B , T_a 定义为源 Ontology (源本体), T_b 定义为目标 Ontology (目标本体), 匹配系统把源 Ontology 及其数据实例看作是学习分类器的样本库, 把训练得到的分类器 L 用于对目标 Ontology 实例的分类, 目标 Ontology 中的每一个概念结点可以看作是已分类的类别, 而且每一个结点类别下都有已经标记的数据实例, 这些实例将被再次分类, 如图 2, 3 所示。当系统完成学习和分类后, 交换源 Ontology 和目标 Ontology, 再次学习和分类所有的数据实例。根据分类结果, 系统使用下面的相似性度量算法来计算每一对概念结点的相似性, 为最终实现 Ontology 之间概念的匹配关系奠定基础。

为了达到更好的学习分类效果, 可采用多策略的学习分类方法, 由于不同的分类器对不同的信息分类效果不同, 有

的使用数据实例内部的信息, 有的使用数据实例之间的语义联系信息, 这样可以获得比单一分类器更好的结果^[3-5]。

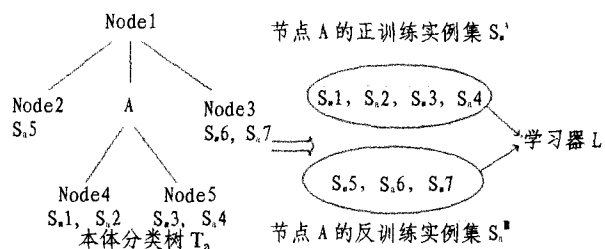


图2 训练学习器 L 的过程示意

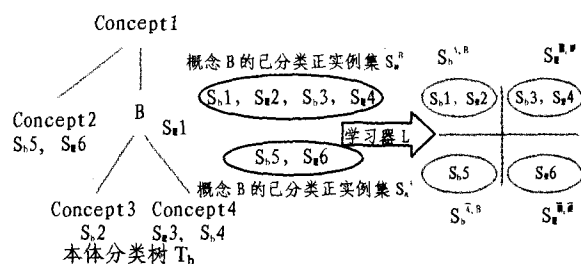


图3 分类过程示意

本体分类树中的概念节点匹配是最终语义匹配的基础, 概念匹配问题的解决将极大地帮助其他问题的解决。关系、取值、公理等问题是下一步匹配标记和相容性检测应考虑的问题。这里我们只关心它的语义内容, 而不关心它的句法特征。

5 相似度计算算法

相似度计算的目的是通过已知的本体信息、领域知识, 特别是数据实例, 为一棵分类树中的每个节点, 去寻找另一棵分类树中的最相似节点。

相似度计算算法 $\text{ConceptSim}(T_a, T_b, A, B)$ 如下:

1. 输入 $T_a = (\text{Node}, R_a, H_a, S_a, D)$ 和 $T_b = (\text{Concept}, R_b, H_b, S_b, D)$;
2. 根据 T_a 分类树将其实例分为: $S_a^A, S_a^{\bar{A}}$;
3. 用 A 的正反实例 $S_a^A, S_a^{\bar{A}}$ 训练一个学习器;
4. 根据 T_b 分类树将其实例分为: $S_b^B, S_b^{\bar{B}}$;
5. 应用学习器 L 分别去区分 $S_b^B, S_b^{\bar{B}}$, 得到区分结果 $S_b^{A,B}, S_b^{\bar{A},B}, S_b^{A,\bar{B}}, S_b^{\bar{A},\bar{B}}$;
6. 将 T_a 和 T_b 交换训练和分类角色, 可以得到分类的结果 $S_a^{A,B}, S_a^{\bar{A},B}, S_a^{A,\bar{B}}, S_a^{\bar{A},\bar{B}}$ 。

7. 计算实例落入可能情况 $P(A, B)$:

$$P(A, B) = \frac{[\text{Num}(S_a^{A,B}) + \text{Num}(S_b^{A,\bar{B}})]}{[\text{Num}(S_a) + \text{Num}(S_b)]}$$

(其中: 函数 Num 表示集合实例的数量。)

8. 类似地, 计算: $P(\bar{A}, B), P(A, \bar{B}), P(\bar{A}, \bar{B})$;

9. 计算相似度

$$\text{Sim}(A, B) = P(A, B) / [P(A, B) + P(A, \bar{B}) + P(\bar{A}, B)]$$

说明:

1. 算法可以计算任意一对概念的相似度 $\text{Sim}(A, B)$ 。
2. 假设 T_a, T_b 中概念节点的数量为: n, m ; 那么, 计算 T_a, T_b 中所有节点对只需重复上述算法 $4n * m$ 次。算法的时间复杂度的取决于所采用的机器学习算法。
3. 若计算了所有节点对的相似度, 即获得了 T_a, T_b 中所

有节点对的相似情况矩阵。

4、采用类似的思想:我们可以设计算法,寻找一棵分类树 T_a 中的概念节点 A 在另一个分类树 T_b 中的最接近的父节点以及最接近的子节点。

6 相关的工作

1、训练实例、分类实例中包含大量的不同类型的信息,而目前的机器学习算法只在一些比较特殊的方面敏感。因此选择什么样的学习算法,如何充分利用各种学习算法的优点,组合多策略的学习算法是研究者们非常关心的问题^[4]。新加坡南洋理工大学的 Myo Myo Naing 等人在这方面作了卓有成效的工作。

已有的文献表明:用 Bayes 规则的 unigram 模型,一阶谓词逻辑文本分类器等文本学习器,名字学习器是必要的选择,但多策略模型和相似性分析模型也将对匹配产生重大影响。

2、匹配研究是目前在语义网研究中的重要方向,匹配研究中采用得更多的是基于 Agent 协商的体系结构,机器学习可以看作是 Agent 之间协商的一种手段。这样的研究还处于起步阶段,目前遇到的最大问题是:完成语义匹配必须受到本体规则、关系、领域知识以及本体一致性检测的约束,虽然有文献在作这方面的工作,但现在还没有文献表明这一工作有重大进展。满足本体描述逻辑公理的匹配似乎极其困难。

结论 现有的知识系统使用的是集中式的、一致性的、可扩充的 Ontology 库,语义异构的领域本体匹配是语义网发展面临的最富挑战性的问题之一,虽然完全程度上的本体匹配难于实现,但是概念匹配的研究还是为语义互联提供了可能。本文在一定程度上解决了概念匹配的问题,为最终的语义匹配提供了参考信息,下一步的工作就是希望在本体规则、

关系、领域知识的约束下,如何通过关系标记预测手段,实现半自动的语义匹配。

参考文献

- 1 Berners-Lee T, Hendler J, Lassila O. The Semantic Web, Scientific American, 2001, 284(5): 34~43
- 2 Neches R, Fikes R E, Gruber T R, et al. Enabling Technology for Knowledge Sharing. AIMagazine, 1991, 12(3): 36~56
- 3 Bouquet P, et al. Contextualizing ontologies, Web Semantics: Science, Services and Agents on the World Wide Web 1, 2004. 325~343
- 4 Zhang D, Lee W S. Learning to integrate Web taxonomies, Web Semantics: Science, Services and Agents on the World Wide Web 2, 2004. 131~151
- 5 Bouquet P, et al. Peer-to-peer semantic coordination, Web Semantics: Science, Services and Agents on the World Wide Web 2, 2004. 81~97
- 6 Gruber T R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 1993, 5: 199~220
- 7 Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods. Data and Knowledge Engineering, 1998, 25(122): 161~197
- 8 Jurisica I, Mylopoulos J, Yu E. Ontologies for Knowledge Management: An Information Systems Perspective, Knowledge and Information Systems, Springer-Verlag London Ltd., Jul. 2004, 6(4): 380~401
- 9 Noy F N. What Do We Need for Ontology Integration on the Semantic Web, Position Statement, In: Proc. of the Workshop on Semantic Integration, the 2nd International Semantic Web Conference, Sanibel Island, Florida, USA, Oct. 2003

(上接第 168 页)

3.3 维数增加时新算法的性能

为了比较优化函数不同维数下两个算法的性能,本文选取 Rastrigrin 为基准函数,由 10 维增加到 60 维。对每维独立实验 50 次,每次运行 1000 代,计算出两个算法在不同维数下的 MBF、SD 以及 SR,具体如表 3 所示。

表 3 显示,随着被优化函数维数的增加,新算法在收敛精度、稳定性和全局搜索性能方面仍然优于原算法。因此,不论迭代次数增加或维数增加,新算法的优越性仍然保持。

结论 作为群体智能的代表性方法之一,粒子群优化算法给大量非线性、不可微和多峰复杂优化问题的求解提供了一种新的思路和解决方法,但也存在容易陷入局部最优、易发散等缺点。本文借鉴生物界中雁群的飞行特征,给出了一种改进的 PSO 算法。新算法一方面将全局极值变换为排序后其前面那个较优粒子的个体极值,这样所有粒子不止向一个最优解的方向飞去,避免了粒子趋向于同一化,保持了粒子的多样性;另一方面新算法使粒子利用其他粒子的更多有用信息,将个体极值替换为各个粒子的个体极值与当前适应值的加权平均,个体的激励变大,加强了粒子之间的合作与竞争。实验结果表明,新算法具有较高的收敛精度、较快的收敛速度、良好的全局搜索能力,在探测和开发能力之间达到适当的平衡。

纵观 PSO 算法的研究现状,还有许多方向值得进一步深入研究。首先,PSO 算法的数学理论基础比较薄弱,缺乏深刻且具有普遍意义的理论分析,因此对数学基础的研究非常

重要,包括算法的收敛性、收敛速度估计和避免陷入局部最优等的证明;其次,进行与其他优化算法或典型群体的比较,将其它算法或典型群体的优点与 PSO 相结合以提高算法性能,将是一个非常潜力的研究方向;PSO 算法虽然已得到广泛的应用,但是进一步拓展 PSO 算法的新的应用领域仍是一项非常有意义的工作。

参考文献

- 1 Kennedy J, Eberhart R C. Particle swarm optimization [A]. In: Proceedings of the IEEE International Conference on Neural Networks [C], 1995. 1942~1948
- 2 Eberhart R C, Shi Y H. Particle Swarm Optimization: Developments, Applications and Resources [A]. In: Proc. Congress on Evolutionary Computation 2001 [C], Piscataway, NJ: IEEE Press, 2001. 81~86
- 3 曾建潮, 介婧, 崔志华. 微粒群算法. 北京: 科学出版社, 2004
- 4 Shi Y H, Eberhart R C. A modified Particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation. Anchorage, Alaska, May 1998. 69~73
- 5 Shi Y H, Eberhart R C. Parameter selection in Particle swarm optimization. Annual Conference on Evolutionary Programming. San Diego, March 1998. 591~600
- 6 Shi Y H, Eberhart R C. Empirical study of particle swarm optimization [A]. In: Proceeding of Congress on Evolutionary Computation [C], Piscataway, NJ: IEEE Service Center, 1999. 1945~1949
- 7 Beekman M, Rantnieks FLW. Long-range foraging by the Honey-bee, Apis Mellifera L. Functional Ecology, 2000, (14): 490~496
- 8 Wilson E O. Sociobiology: The New Synthesis [M]. Cambridge: Belknap Press, 1975
- 9 Ratnaweera A, Halgamuge S. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evolutionary Computation, 2004, 8(3): 240~255