

# 认知演化算法

王 磊 王维平 杨 峰 朱一凡

(国防科技大学信息系统与管理学院系统工程系 长沙 410073)

**摘 要** 受人类创造性思维过程的启发,借鉴认知心理学和创新计算的研究成果,提出了一种模拟人类基于创造性思维的问题求解过程和行为的智能算法——认知演化算法。该算法以知识为核心,将问题求解看成一个基于知识的创造性思维过程,对发散思维、收敛思维、记忆、执行、学习和价值体系 6 个关键模块进行了建模,充分发挥了知识演化和基于知识的创造性思维技巧在问题求解中的作用。通过数值实验分析了 CEA 各参数对算法性能的影响,并以一个扩展的路径优化问题将 CEA 与经典的智能算法进行了比较。实验结果表明,针对知识密集型优化问题,该算法能够以较少的目标评价次数得到问题的较优解。

**关键词** 认知演化算法,创造性思维,问题求解,创新计算,认知心理学

**中图分类号** TP18 **文献标识码** A

## Cognition Evolutionary Algorithm

WANG Lei WANG Wei-ping YANG Feng ZHU Yi-fan

(Institute of Systems Engineering, National University of Defense Technology, Changsha 410073, China)

**Abstract** Inspired by the process of human creative thinking, cognition evolutionary algorithm, a novel intelligent algorithm based on cognition science and computational creativity, was proposed, which simulates the creative thinking based problem solving process and behaviors. The algorithm is comprised of six components which are divergent thinking, convergent thinking, memory, execution, learning and value measures. Taking the problem-solving as a knowledge based creative thinking process, knowledge evolution and knowledge based creative thinking skills play important roles in the algorithm. The impact of the parameters of cognition evolutionary algorithm on the performance of the algorithm was analyzed through an extended path optimization problem. The results show that the novel algorithm can reduce object score evaluation times compared with other classic intelligent algorithms for solving knowledge intensive optimization problems.

**Keywords** Cognition evolutionary algorithm, Creative thinking, Problem solving, Computational creativity, Cognitive psychology

智能算法是模拟自然界生物体系和人类智能现象发展而来的软计算方法,它通过计算机再现自然界和人类的某些智能行为,来解决现实中难以用传统方法求解的复杂问题。常见的智能算法包括遗传算法(GA)、蚁群算法(ACO)、分布估计算法(EDAs)、人工免疫算法(AIS)等。这些算法在求解现实问题时,往往收敛速度过慢<sup>[1-3]</sup>。如果优化的目标函数计算是计算密集型的(比如复杂的军事仿真),那么过多的迭代次数将会耗费大量的目标评价时间,造成优化效率过低。同时现有的智能算法大多是受自然进化法则(如 GA)、动物群体智慧(如 ACO)以及生命系统(如 AIS)机理启发而提出的,并未很好地利用问题背景以及问题求解过程中所产生的知识<sup>[4]</sup>。也有学者研究了将智能算法与知识相结合来求解问题,但是知识并不存在于这些算法对应的原始系统框架之内(比如基因的遗传变异不存在知识的因素,蚂蚁的路径选择也

并不需要复杂知识的控制),所以这种结合存在困难。从算法实现上看,这个特点是这类智能算法的优势所在,它们按“自然法则计算”<sup>[5]</sup>,满足“生成-评价-生成”的基本模式,并不需要过多的启发信息,算法结构简单、适用面广;另一方面,这个特点也限制了智能算法自然地与知识相结合,充分发挥知识的作用。

与知识紧密联系的是以认知系统为基础的人类自然智能,包括人类的学习、推理等认知行为都是以知识为基础的。人类的思维过程是问题求解的根本过程。图琴在提出元系统跃迁理论时指出“发生在人脑中的试错方法的进行比现实生活快很多倍”<sup>[6]</sup>。这说明对人类思维机制进行模拟有望得到比基于自然选择机理的智能算法更高的求解效率。创造性思维是解决复杂问题的重要途径。著名科学家钱学森在提出求解复杂问题的系统集成方法时就指出,创造性思维是智慧的

到稿日期:2009-10-26 返修日期:2010-01-20

王 磊(1982—),男,博士生,主要研究方向为系统论证与仿真评估、智能算法,E-mail: laye\_wong@sina.com;王维平(1962—),男,教授,主要研究方向为体系工程、组合仿真等;杨 峰(1976—),男,副教授,主要研究方向为系统论证与仿真评估、数字化论证工程等;朱一凡(1962—),男,教授,主要研究方向为装备体系工程等。

源泉<sup>[7]</sup>。近来的研究发现,通过计算机对创造性思维进行建模、模拟,可望使计算机达到和产生具有人类水平的创新能力<sup>[8]</sup>,这类研究称为创新计算(computational creativity)。

因此本文借鉴了认知心理学和创新计算的相关研究成果,将创造性问题求解的认知过程和行为进行算法化,提出了一种新的可用于复杂问题求解的智能算法——认知演化算法(Cognition Evolutionary Algorithm,CEA)。

## 1 认知演化算法原理

### 1.1 知识-解互映射平行演化机制

任何思维成果的形成都是由其心理因素驱动的,行为层面的研究应该与心理层面的研究相结合,即解的演化过程受其心理目标演化过程的支配。问题求解过程中采用的不是直接从旧解到新解的演化机制。在演化形成新解的过程中,存在一个从旧解到知识空间的映射过程,新解的产生是一个由知识空间到解空间的映射过程。本文称这种演化机制为“知识-解”互映射平行演化机制(Knowledge to Solution Bidirectional Mapping Parallel Evolution mechanism, KSBMPE),如图1所示。

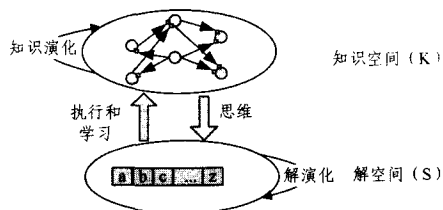


图1 “知识-解”互映射平行演化机制

假设知识空间为 $K$ ,解空间为 $S$ 。

执行和学习映射指通过对解的学习形成知识,可表示为 $ELM: S \rightarrow K$ 。ELM实际上是由两个映射合成:首先是执行映射将解作用于外部环境,生成数据样本,其次是学习映射将数据样本映射为知识。

思维映射指通过基于知识的思维过程产生新的解,可表示为 $TM: K \rightarrow S$ 。

平行演化是指在CEA的执行过程中,知识空间和解空间处于同时演化的状态,并且知识空间演化和解空间演化是相互关联的。这种关联性是由ELM和TM造成的,通过ELM从旧解中产生新知识,新知识的补充使得知识空间发生演化;通过TM在当前知识空间形态的基础上产生更合理优化的解,使得解空间发生演化。

互映射指从解空间到知识空间的执行和学习映射以及从知识空间到解空间的思维映射。

### 1.2 认知演化算法的基本框架

创造性问题求解可看作由记忆、学习、创造性思维以及执行4部分组成,创造性思维又具体包含了两类思维技巧——发散思维和收敛思维。

CEA可以用如下的六元组进行描述: $CEA = \langle L, M, V, D, C, E \rangle$ 。其中 $L$ 为知识学习模块, $M$ 为记忆模块, $V$ 为价值体系模块, $D$ 为发散思维模块, $C$ 为收敛思维模块, $E$ 为执行模块。图2给出了CEA的基本框架,并标明了各模块之间的关系。

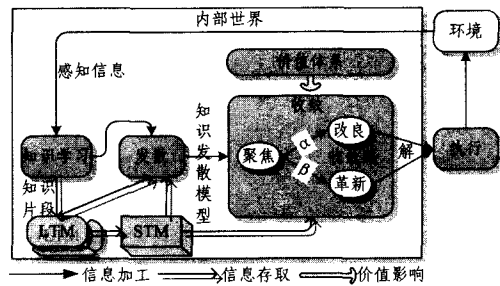


图2 CEA基本框架

模块的作用分别是:

- (1)知识学习模块将从环境中感知得到的信息通过学习形成知识;
- (2)记忆模块将通过学习得到的知识进行组织和管理;
- (3)价值体系模块是表征“思考者”价值取向的标准,用于对整个解或部分解进行价值评估,价值体系将会对各思维模块造成影响;
- (4)发散思维模块是将已有的知识进行组合,形成结构化的知识发散模型,可以看作是一个人脑内部知识的自组织过程,所形成的知识发散模型又可看成是一个包含了若干可行解的集合;
- (5)收敛思维模块是在发散思维得到的知识发散模型基础上,按照一定的规则从中选取一个由若干可行解组成的收敛集,并创造性地进行改良或革新,得到新的解;
- (6)执行模块将通过思维产生的新解作用于环境。

## 2 认知演化算法实现步骤

### 2.1 解和目标

**定义1(解和局部解)** 解是对目标系统的输入,在CEA中表示为长度为 $l$ 的字符串。字符串中的每一分量称为局部解。解的全体称为解空间,用 $S$ 表示。 $s = \{l_i | l_i \in LS, i = 1, 2, \dots, l\} \in S$ 。其中 $l_i$ 为局部解, $LS$ 为局部解空间。

解是对系统的输入,以此控制系统,使得系统从当前状态转移到期望的目标状态。解就是系统状态转移过程中采用的手段集合,每个局部解 $l_i$ 都是一个促使状态发生转移的手段。局部解空间就是这样一些手段的集合。每一种手段都可能使系统发生状态上的转移 $s_i \rightarrow s_{i+1}$ 。

**定义2(目标和子目标)** 目标是指系统期望达到的状态,也是一个解所能产生的效益。目标可表示为 $o_i = \{st_i | i = 1, 2, \dots, m\}$ ,其中 $st_i$ 为子目标,每个子目标 $st_i$ 具备以下3个属性:标识 $ID_i$ 、值 $v_i$ 、完成度 $\phi_i$ 。 $O_i \in \Omega_i$ , $\Omega_i$ 为子目标空间,包含了在达成目标过程中可能的子目标。 $v_i$ 可能是连续型或者离散型。

在CEA中不妨将局部解与子目标之间看作是一一对应的关系,即每一个局部解都唯一存在一个与之对应的子目标。所以必定存在一个从局部解到子目标之间的映射。

**定义3(全局映射和局部映射)** 全局映射将一个目标映射为一个解,局部映射将一个或多个子目标映射为一个局部解。

CEA假设系统在朝目标状态转移的过程中要依次达成若干子目标,这些子目标按照达成序列排列,从而形成了一条问题求解的路径。解是达成目标的手段,每个局部解都会对应一个或多个子目标。CEA在求解之前,除了确定解的编码

以外,还需要确定子目标空间。

## 2.2 知识学习模块和执行模块

知识学习模块的作用是确定子目标的完成度以及建立各子目标之间的定量关系,这种关系的表达就是知识。由于CEA中包含了大量对知识的操作,因此知识是CEA的核心。

执行模块和知识学习模块是CEA中两个直接与外部环境进行交互的模块,也是认知系统能够持续演化的动力之源。

### 2.2.1 知识表示

结合问题求解的需求,下面将对知识片段的概念进行定义。

**定义4(知识片段)** 知识片段是指由子目标按照顺序连接而成的知识结构,可表示为  $KF = \{(st_i, st_j, r_{ij}) \mid st_i, st_j \in \Omega, r_{ij} \in Re\}$ ,其中  $Re$  是这些子目标达成效果之间影响关系的集合。

根据知识类型的不同, $st_i$  和  $r_{ij}$  可用不同的方法表示。比如用贝叶斯网络的形式表示,那么  $st_i$  就是贝叶斯网络节点,而  $r_{ij}$  是各节点所维护的条件概率表。知识模型的选取一方面要与问题相关,同时必须能够反映知识片的特征和所包含的信息。

### 2.2.2 解到知识的映射方法——执行和知识学习

执行模块将解作用于环境,环境将产生一个响应(用数据集表示),因此执行模块可表示为从解到数据集的映射过程:  $execute: S \rightarrow DS$ ,其中  $S$  为解空间, $DS$  为数据集。

知识学习是从数据集到知识片段的映射过程,可表示为  $learn: DS \rightarrow KF$ ,其中  $KF$  为知识片段。在CEA中并未规定具体采取何种学习算法,在解决实际问题时应该根据问题的特点灵活选择。知识学习方法与具体采用何种知识模型密切相关。

## 2.3 记忆模块

记忆模块包含长时记忆(LTM)和短时记忆(STM)两类记忆。其中,LTM用于存储学习得到的知识片段,STM用于存储思维的过程数据。记忆模块可表示为  $M = \langle LTM, STM \rangle$ ,其中LTM为长时记忆,STM为短时记忆。

LTM是知识片段的容器,通过学习得到的知识片段存在于LTM中。LTM可以表示为  $LTM = \langle KF \rangle$ 。其中  $KF$  为知识片段集合,  $KF = \{kf_i \mid i = 1, 2, \dots, n\}$ ,  $kf_i$  为独立的知识片段,  $kf_i = \langle D_{kf_i}, t_{kf_i} \rangle$ ,  $D_{kf_i} \in R^+$  为  $kf_i$  所对应解的目标达成程度,  $t_{kf_i} \in Z^+$  为知识片段在LTM中存在的时间。

人类的记忆容量据推测应该在  $10^{12}$  字节左右,但是现实所使用的记忆容量比这要小得多。研究证明,人类的记忆并不是真正地遗忘,而是不能被提取。记忆提取模型将在2.4.1节中研究。

STM是在思维过程中临时开辟的记忆空间,供思维活动的开展。STM可以表示为  $STM = \langle C_{STM} \rangle$ ,其中  $C_{STM}$  为STM容量,它控制了CEA中发散思维的思维发散度,而思维发散度决定了在相同的知识条件下所能产生的解的数量。

## 2.4 发散思维模块

发散思维是将知识片段按需组装的过程。在没有问题触发的情况下,知识片段以独立的形式存于LTM中。一旦有问题触发,则知识片段将会按照问题的特征进行组合<sup>[9]</sup>。发散思维将不同知识片段中的子目标进行重新组合,从而产生新的目标达成路径。

发散思维就是对记忆模块中存储的某些知识片段进行组合,形成知识组装树(Knowledge Assembly Tree, KAT),并从中找到当前知识条件下所有的可行路径的过程。发散思维可表示为如下映射:  $divergent: LTM \rightarrow KAT$ ,其中LTM为长时记忆,KAT为知识组装树。 $divergent$  由两个步骤组成:记忆提取和知识组装。

### 2.4.1 记忆提取

将知识片段从LTM中提取出来是发散思维的基础。本文假设,知识片段的提取由刺激强度  $D_{kf_i}$  以及知识片段存储时间  $t_{kf_i}$  控制。

$C_{LTM}$  称为记忆提取能力,它决定了从LTM中能提取的最大知识片段的数量;  $D_{kf_i}$  说明了知识片段所对应的解所能达到的目标程度,刺激强度越大,就越不容易被遗忘;  $t_{kf_i}$  说明知识片段在LTM中存在的时间,  $t_{kf_i}$  越大则越容易被遗忘。

廖海林<sup>[10]</sup>根据艾宾浩斯曲线建立了一个考虑了遗忘速度和时间的遗忘模型,建立了如下的记忆提取模型:

$$GK(kf_i) = D_{kf_i} e^{-\mu kf_i} \quad (1)$$

式中,  $kf_i$  为LTM中第  $i$  条知识片段,  $GK(kf_i)$  为  $kf_i$  的提取强度。提取强度越大,则在发散思维时能够被提取的可能性越高。在发散思维时,会提取LTM中  $GK(kf_i)$  值最大的  $C_{LTM}$  条知识片段。

### 2.4.2 知识组装

在CEA中采用知识组装树模型对知识片段进行组装。

**定义5(知识组装树)** 知识组装树是将知识片段进行按需组合而成的数据结构,可以表示为如下的五元组:  $KAT = \langle o, M, C, S, C_{STM} \rangle$ 。其中  $o$  为目标节点,是能够表征问题特征的参数;  $M$  为普通节点集合;  $C$  为节点之间的有向边集合;  $S$  为每个节点所维护的关系参数,  $C_{STM}$  为STM容量,这里对应于KAT中所包含的最大路径数。

下面提出了一个能够将LTM中的知识片段进行组合,生成知识组装树的算法。

#### 算法1 知识组装树构造算法

1. 初始化KAT,加入目标节点  $O$ ,节点计数器  $m = 1$ ;
2. for each  $kf_i \in KF$ 
  - for each  $N_j \in kf_i$ 
    - 取出  $N_j$  的父节点  $N_p$ ,将  $(N_p, N_j)$  插入知识映射表  $map\_kp$ ;
3.  $N_i = O$
4. for each  $(N_p, N_i) \in map\_kp$ 
  - 将  $N_p$  作为下级节点插入到节点  $N_i$  下,
  - 如果  $N_p$  为源节点,  $m++$ ,并将  $N_p$  到  $O$  的路径存入路径集合  $Path$ ;
  - 如果  $m \geq C_{STM}$ ,转步骤5,否则  $N_i = N_p$ ;
5. return KAT

图3给出了从知识片段到知识组装树的过程。

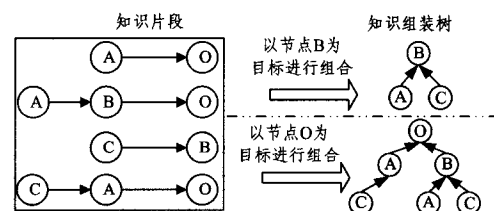


图3 通过知识片段组装构造知识组装树

考察图3中左侧的4条知识片段。首先根据算法1中的

步骤 2 拆分这些知识片段,可得如下的子目标对: $A \rightarrow O, A \rightarrow B, B \rightarrow O, C \rightarrow B, C \rightarrow A$ ,然后根据步骤 3 和步骤 4 把这些拆分出来的子目标对分别按目标  $B$  和  $O$  进行组合,可得图 3 右的两棵知识组装树。

下面给出与 KAT 相关的几个定义:

目标节点指 KAT 中描述问题求解目标所对应的节点。KAT 中有且仅有一个目标节点。KAT 中的目标节点表明了发散思维的目的性。图 3 中两个 KAT 的目标节点分别为  $B$  和  $O$ 。普通节点是指 KAT 中除目标节点外的所有其他节点。从问题求解的角度看,普通节点代表与问题求解相关的子目标。源节点指在至少一条知识片段中不含有父节点的节点。图 3 中  $A$  和  $C$  为源节点。

路径指从 KAT 中任意一个源节点到目标节点所经历的有序通路,可表示为  $path = (v_1, v_2, \dots, v_m)$ ,其中  $v_i \in M \cup O$ 。一条路径代表了问题求解的一条可能的“道路”。

KAT 在本质上就是在已有的知识条件和认知水平(由  $LTM, C_{LTM}$  和  $C_{STM}$  决定)下列举所有可能的解决方案。同时 CEA 中需要保证可以找到某个映射将一条路径转化为解。这种假设在一般情况下都是可以保证的。 $C_{STM}$  反映了人类工作记忆的大小,对应于 STM 的容量,具备更大容量的 STM 可以提高思维发散度。

路径可以看作是一类特殊的知识片段,但是知识片段与 KAT 中的路径并不存在一一对应的关系,即参与 KAT 构建的所有知识片段均能在 KAT 中找到其对应的路径,但并不是 KAT 中的每条路径都能找到其对应的知识片段。该性质说明发散思维在本质上已经具备“组合创新”的能力。考察图 3 所示的知识片段组装过程,可以发现路径  $C \rightarrow B \rightarrow O$  在知识片段中并不存在,它是由知识片段  $C \rightarrow B$  和  $B \rightarrow O$  组合而产生的,即  $C \rightarrow B$  和  $B \rightarrow O$  组合产生了创新的知识  $C \rightarrow B \rightarrow O$ 。

## 2.5 价值体系

潜在的价值是指导人们进行决策和行动的关键因素。问题求解是带有目的性的,目的性本身意味着系统的内在价值。CEA 是面向问题求解的,因此其内部行为都需要知道如何将可能会更好,如何将可能会变糟,而不是一个完全随机的过程。而这个判断过程需要一个价值体系进行指导。

具体地,CEA 中价值体系主要有以下两个作用:(1)对解的满意度进行估计;(2)对子目标进行评估,辅助对问题进行分解。

### 2.5.1 价值定义

在 CEA 的价值体系内定义了 5 类价值,每一类价值都对一条路径及路径中的节点。

$$(1) \text{ 目标价值 } V_a(T) = est(T)$$

目标价值  $V_a(T)$  是问题求解追求的最终目标,用于度量路径  $a$  满足需求的程度。其中  $est(\cdot)$  是目标估计函数,其估计准确度与知识的精度和估计方法相关。目标价值主要起着对路径满意度进行估计的作用。

$$(2) \text{ 子目标价值 } SV_a(T, ST_i) = V_a(T) - V_a(T|st_i)$$

其中  $V_a(T|st_i)$  表示当路径  $a$  中的子目标  $st_i$  未能达到时  $a$  的目标价值。子目标价值用于度量子目标的达成对最终目标达成的贡献。

$$(3) \text{ 潜在价值 } PV_a(T, st_i) = dV_a(T)/dV_a(st_i)$$

潜在价值为子目标达成程度对目标达成程度的影响程度,潜在价值表明子目标达成对目标价值实现的提升程度。

$$(4) \text{ 期望投入 } EI_a(st_i) = \frac{v_a - V_a(T|st_i)}{PV_a(T, st_i)}$$

期望投入表明,为了达成目标子目标的达成度需要提升的程度。 $v_a \in R^+$  为目标的期望价值,是由决策者指定的, $V_a(T|st_i)$  表示当子目标  $st_i$  达到时所能产生的目标价值。

$$(5) \text{ 附加投入 } AI_a(st_i) = EI_a(st_i) - V_a(st_i)$$

附加投入表示子目标期望投入与实际投入的差,表明了为达到满意的目标达成度,子目标达成度还需提高的程度。

这 5 类价值均可通过路径中的关系  $r_{ij}$  进行计算,其中子目标价值、潜在价值、期望投入和附加投入这 4 个价值具有评估局部解的作用。需要指出的是,这里虽然给出了 5 类价值的定义,但是具体每类价值的计算方式与知识的表示形式紧密相关,所以在针对具体问题时,还要将这些价值的计算方式进行实例化。

### 2.5.2 KAT 基本操作

下面在所定义的价值体系基础上,对发散思维所形成的 KAT 的基本操作进行定义,收敛思维将在这些基本操作的基础上进行定义。

$$(1) \text{ 最大价值路径操作 } \max\_value\_path(): path_{\max\_value} = \operatorname{argmax}_{path_i \in KAT} V_{path_i}(T);$$

$$(2) \text{ 最小价值目标操作 } \min\_value\_target(path): st_{\min\_value} = \operatorname{argmin}_{st_i \in path} SV_{path}(T, st_i);$$

$$(3) \text{ 最大潜在价值目标操作 } \max\_pv\_target(path): st_{\max\_pv} = \operatorname{argmax}_{st_i \in path} PV_{path}(T, st_i);$$

$$(4) \text{ 最小附加投入目标操作 } \min\_ai\_target(path): st_{\min\_ai} = \operatorname{argmin}_{st_i \in path} AI_{path}(T, st_i);$$

$$(5) \text{ 替换操作 } replace\_target(path_{old}, st_{new}, st_{old}): path_{new} = path_{old} - st_{old} + st_{new}, \text{ 替换操作的含义是,用新子目标 } st_{new} \text{ 替换路径 } path_{old} \text{ 中的旧子目标 } st_{old}, \text{ 形成新路径 } path_{new}。$$

## 2.6 收敛思维模块

收敛思维将发散思维所形成的知识发散模型,通过聚焦、改良或革新操作,形成新解。

收敛思维可表示为如下的映射:  $convergent: KAT \rightarrow S$ 。  $convergent$  由 3 个操作组成,包括聚焦操作  $focus$ 、改良操作  $adapt$  和革新操作  $innovate$ 。

### 2.6.1 聚焦操作

在利用已有的知识片段进行知识组合以后,将会产生若干条路径。人类在思考问题时不可能同时对多个对象进行考察,而是在思维进行到一定阶段后采用一个“聚焦”过程,经过聚焦以后常常会留下有限的几个对象,这就是思维收敛的过程。

收敛规则  $F$  可描述为二元组:  $F = \langle f(a_i), n_{am} \rangle$ ,其中  $f(a_i)$  为聚焦函数,  $a_i \in A$ ,  $n_{am}$  为收敛集维度,即聚焦对象数量。且  $f(a_i): A \rightarrow [0, 1]$ 。聚焦函数将 KAT 中所有路径映射为  $p \in [0, 1]$ ,  $p$  反映了每条路径可能被选择聚焦的概率,  $p$  越大则路径被选择聚焦的可能性越大。KAT 中价值较高的路径将会有更高的被选择概率,因此  $f(a_i) = V_{a_i}(T)$ 。

### 2.6.2 改良操作和革新操作

#### (1) 基本操作

在 CEA 的一次迭代过程中由于只能在改良型创新和革新型创新两种创新样式中按照概率二选其一,通过对收敛集  $A'$  中的路径进行变换,从而产生新解。变换是由改良和革新两个操作实现的。

改良和革新操作可以表示为  $AI(path, creative\_style, v_e, a, b)$ 。其中  $path$  为聚焦操作产生的聚焦路径;  $creative\_style \in \{0, 1\}$  为创新风格,  $creative\_style = 0$  时表明采用静态创新风格参数;  $creative\_style = 1$  时表明采用动态创新风格参数,  $v_e \in R^+$  为目标期望值;  $a \in [0, 1]$  为改良操作系数, 表明了在一次迭代中选择进行改良操作的概率。  $\beta = 1 - \alpha$  为革新操作系数, 表明了在一次迭代中选择进行革新的概率。

英国心理学家 M. J. Kirton 在其所提出的“适应-变革理论”(adaption-innovation theory) 中区分了两种创新样式: 改良型(adapters)和革新型(innovators)<sup>[11]</sup>。他认为创新个体并非绝对的改良者和革新者, 而是一个混合体, 通常处于完全改良者和完全创新者之间的一种创新风格<sup>[12, 13]</sup>, 这表明改良操作和革新操作的选择并不是绝对的。  $\alpha, \beta$  两个参数可以反映不同人的创新风格, 通过调整这些参数可以使算法具备不同的“创新风格”, 以适应不同的问题特点。所以  $\alpha$  和  $\beta$  统称为创新风格参数。

改良操作可表示为  $adapt(ST, G, \Psi, \Theta)$ , 其中  $ST$  是改良的子目标;  $G$  为实施改良操作期望达到的目标,  $G = AI_{path}(T, st_{min\_ai})$ ,  $\Psi$  为实施改良操作时所处环境,  $\Psi$  可理解为聚焦路径中除  $ST$  外其他子目标的完成程度和之间的影响关系;  $\Theta$  为资源约束,  $\Theta$  可理解为在保证其他子目标资源需求的条件下剩余的可用资源。

在 CEA 中并没有规定改良操作的具体行为, 而是将改良操作当作一种当前层次问题求解与下一层次问题求解之间的接口, 规定了在何种环境和资源约束下应该达到的子目标。而操作具体行为的选择则是根据子问题求解的特点而灵活选择的。

革新操作可以表示为  $innovate(\Omega)$ , 其中  $\Omega$  为子目标空间, 规定了革新的范围, 即通过革新得到的新子目标  $st_{new} \in \Omega$ 。

改良操作和革新操作均分两步完成。第一步, 根据一定的定位准则, 定位于路径中的某一个普通节点, 称所定位的节点为操作对象, 定位准则与价值体系相关; 第二步: (1) 改良操作将对操作对象进行改进, 提高操作对象的完成度; (2) 革新操作将从子目标空间中随机地选择一个对象作为节点替换路径中的操作对象。

### (2) 动态创新风格参数

不难理解, 针对不同的问题或者问题求解阶段, 所需要的创新风格是不同的。一个足够聪明的人, 会根据环境的变化而选择不同的创新操作。因此, CEA 中的创新风格参数可能是动态变化的。没有具体的研究成果表明认知风格如何根据环境的变化而改变。本文假设: 当解决方案所能达到的价值与期望目标价值相差较大时, 将倾向于选择革新型创新, 反之倾向于改良型创新。这个假设建立在对问题求解上, 当状态离目标状态较近时, 通常选择保守的方式; 当差距较大时, 往往选择激进革新的方式。

如果用  $\Delta v$  表示解决方案所能达到的价值与期望目标价值的差, 那么  $\Delta v = v_e - v$ 。其中  $v_e$  为期望目标价值,  $v$  为当前解决方案所能达到的价值。

动态创新风格参数定义为  $\alpha^* = 1 - (v_e - v) / v_e$ ,  $\beta^* = 1 - \alpha^*$ 。其中,  $\alpha^*$  和  $\beta^*$  为动态创新风格参数。可见随着  $v$  的提高, 实施改良型创新的概率逐步提高, 实施革新型创新的概率逐步降低。

### 2.6.3 收敛思维算法

### 算法 2 收敛思维算法

1. 进行聚焦操作  $path = focus(KAT)$ , 得到聚焦路径  $path$ ;
2. 生成  $(0, 1)$  间的均匀分布随机数  $r$ , 如果  $r < \alpha$ , 进行  $min\_ai\_target(path)$  操作, 得到  $st_{min\_ai}$ , 转 2; 如果  $r \geq \alpha$ , 转 4;
3. 进行改良操作  $path_{new} = adapt(st_{min\_ai}, G, \Psi, \Theta)$ , 转 6;
4. 进行  $min\_value\_target(path_{max\_value})$  操作, 得到  $st_{min\_v}$ ;
5. 通过  $innovate(\Omega)$  产生  $st_{new}$ , 通过  $replace\_target(path_{max\_value}, st_{new}, st_{min\_v})$  产生  $path_{new}$ ;
6. 将  $path_{new}$  映射  $S_{new}$ , 输出  $S_{new}$ , 结束。

算法 2 中  $min\_ai\_target(path)$  和  $min\_value\_target(path)$  操作均为 KAT 的基本操作, 分别为改良操作和革新操作的对象定位规则。

### 2.7 CEA 算法总体流程

根据以上对各模块的研究, 下面给出 CEA 的总体流程。

#### 算法 3 认知演化算法

1. 初始化参数: 记忆提取能力  $C_{LTM}$ 、短时记忆容量  $C_{STM}$ 、创新风格  $creative\_style$ 、创新风格参数  $\alpha$ 、收敛集维度  $n_{con}$ 、最大迭代次数  $iteration\_num$ ; 如果  $creative\_style = 1$ , 初始化期望值  $v_e$ ;
2. 确定解的编码方式和子目标空间  $\Omega$ ;
3. 生成初始解集合  $S_{init}$ , 且当前集合  $S(0) = S_{init}, t = 0$ ;
4. 如果  $t < iteration\_num$ , 分别执行  $S(t)$  中的每个解  $data\_set_i = execute(s_i)$ , 得到  $n_{con}$  个数据样本, 否则转 10;
5. 通过学习  $kf_i = learn(data\_set_i)$ , 将  $n_{con}$  个数据样本映射为知识片段;
6. 将步骤 5 中学习得到的知识片段存入 LTM;
7. 对长时记忆  $LTM(t)$  中的知识片段进行发散思维  $KAT(t) = divergent(LTM(t))$ , 得到知识组装树  $KAT(t)$ ;
8. 对  $KAT(t)$  中的所有路径进行收敛思维  $S_{con}(t) = convergent(KAT(t))$ , 得到收敛解集合  $S_{con}(t)$ ;
9. 对  $S_{con}(t)$  中的每个解, 进行改进和联想操作  $S(t+1) = IA(S_{con}(t), creative\_style, v_e, \alpha, \beta)$ , 得到新解集合  $S(t+1), t = t + 1$ , 返回 4);
10. 返回  $S(t)$  中的最优解  $s$ 。

### 3 实验分析

本文使用扩展的路径优化问题对 CEA 进行测试。扩展的路径优化问题可以描述为: 给定图  $G = (V, A)$ , 其中  $V = \{v_i\}$  为节点集, 每个节点  $v_i$  包含裕度  $e_i, e_i \in \{1, 2\}, A = \{a_{ij}\}$  为各顶点相互连接组成的边集, 各节点间的连接距离为  $d_{ij}$ ; 要求找到一条最长的路径, 路径的起始点为  $v_S$ , 终点为  $v_E, v_S, v_E \in V$ ; 一条路径中不包含相同的节点, 且路径的起始点和终点分别为  $v_S$  和  $v_E$ 。路径中两个节点之间的距离  $d_{ij}$  通过下式进行计算:

$$d_{ij} = \begin{cases} d_{ij}^k e_i d_{ij}^k, & d_{ij}^k \neq 0 \\ d_{ij}^k, & d_{ij}^k = 0 \end{cases} \quad (2)$$

式中,  $d_{ij}^k = 0$  表示  $v_i = v_S$ 。

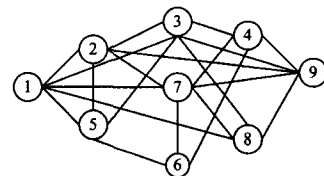


图 4 CEA 验证实例图模型

如图 4 中  $1(1) \rightarrow 2(2) \rightarrow 3(1) \rightarrow 4(1) \rightarrow 9(1)$  为一条路径, 括号中的数字表示每个节点所对应的裕度, 则  $d_{23} = d_{12}^1 e_2 d_{23}^1$ 。该优化问题可以形式化描述为:

$$\max_{p_i, e_i} \sum_{i,j} d_{ij} \quad (3)$$

$$\sum e_i \leq m$$

同时定义了两个指标——思维定势度和信息量,供分析时使用。

思维定势度  $SE(t)$  定义为:

$$SE(t) = \sum_i \frac{k_i(t)}{t} \ln \frac{k_i(t)}{t} \quad (4)$$

思维定势度是指通过收敛思维得到的解的聚集程度。其中  $t$  为演化代数,  $SE(t)$  为第  $t$  代时的思维定势度 ( $SE(t) < 0$ ),  $k_i(t)$  为第  $i$  个曾经聚焦的解截止第  $t$  代时被聚焦的总次数。

信息量  $IQ(t)$  定义为:

$$IQ(t) = \frac{n_f(t)}{2n^2} \quad (5)$$

式中,  $IQ(t)$  为第  $t$  代时的信息量 ( $0 \leq IQ(t) \leq 1$ ),  $n_f$  为当前长时记忆中不同的节点对数量,  $n$  为图的节点数。

### 3.1 算法参数实验分析

算法参数实验将对  $C_{LTM}$ ,  $C_{STM}$  以及创新风格参数对算法性能的影响进行分析。

图 5 和图 6 分别给出了  $C_{LTM}$  和  $C_{STM}$  对算法性能的影响。

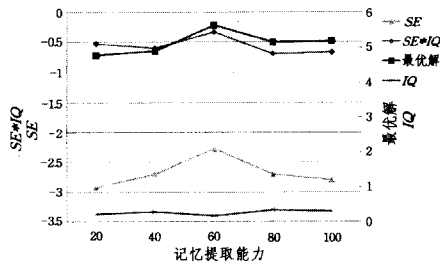


图 5  $C_{LTM}$  与思维定势度、信息量和最优解的关系

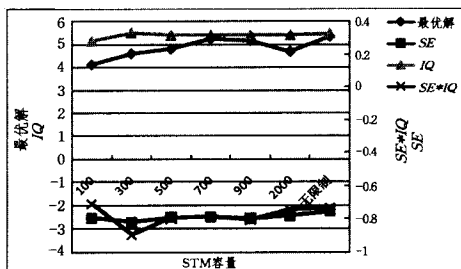


图 6  $C_{STM}$  与思维定势度、信息量和最优解的关系

从图 5 可知,  $C_{LTM}$  对优化效果会产生很大的影响。随着  $C_{LTM}$  不断提高, 问题求解的效果会经历一个先提升 ( $C_{LTM} = 80$  时效果最好)、后逐渐降低的过程。用通俗的话来讲就是记性好不一定能干。一方面随着  $C_{LTM}$  的增加求解效果总趋势是不断提高(最优解的目标值越来越大), 另一方面思维定势度随着  $C_{LTM}$  的增加, 呈现先增后减的趋势。思维定势度的增加说明了在收敛思维时容易将“焦点”集中在某几个相对数量较少的解上; 而信息量随着  $C_{LTM}$  的增加, 呈现先增后减再增的趋势, 信息量的增加说明了对问题求解所需信息了解得更全面深入。从图中可以看出一个规律: 最优解的目标值与思维定势度和信息量的乘积(表示为  $SE * IQ$ )呈反比关系, 故可以认为是  $SE * IQ$  决定了问题求解的效果。

从图 6 可知, 当  $C_{STM}$  从 700 提高到 2000, 最优解的目标值呈下降趋势。在 CEA 中这种现象可以解释为: 随着可利用的信息增多, 思维的发散度将会增大, 而当发散度过大时将会

影响求解效果。

图 7 给出了创新风格参数对算法性能的影响。从实验结果可以发现, 当使用动态创新风格参数, 且选取的  $v_c$  接近最优值时, CEA 可以取得很好的优化效果。而当  $v_c$  值偏离最优值较大时, 则可能使算法的性能大大下降。所以在使用 CEA 进行问题求解时, 如果对目标最优值的大概范围有一定的了解, 则最好使用动态创新风格参数; 如果对目标值的范围没有一定的认识, 则应该使用静态认知风格参数。采用动态创新风格参数时,  $SE * IQ$  值低于所有使用静态创新风格参数时的情况。

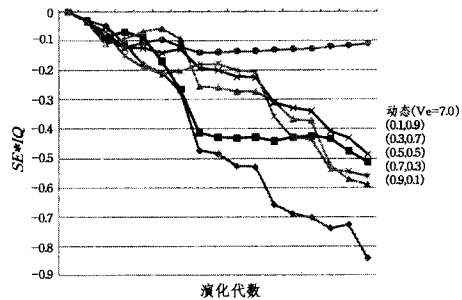


图 7 创新风格参数与  $SE * IQ$  关系图

在静态创新风格参数的情况下, 随着  $\beta$  取值的不断增大, 其获得较优解的能力也不断提高。其原因在于, 联想操作比改进操作可以获得更多的信息量, 这一结论可以从平均信息量列看出。通过提高  $\beta$  可以提高算法的全局寻优能力, 所以一般可将  $\beta$  值设为 0.6 到 0.9, 以期得到更好的全局寻优能力。

### 3.2 对比实验分析

通过实验比较 CEA 与 GA, EDAs 和 ACO 的问题求解效率, 每种算法的参数设置如下:

CEA:  $iteration\_num = 100, \alpha = 0.7, \beta = 0.3, C_{LTM} = 60, n_{con} = 1, C_{STM} = 10000$ ; GA: 种群 = 100, 迭代次数 = 50, 交叉概率 = 0.9, 变异概率 = 0.1, 运行次数 = 50; EDAs: 种群 = 300, 迭代次数 = 50, 选择概率 = 0.2; ACO: 信息启发式因子 = 0.8, 期望启发式因子 = 0.6, 信息挥发系数 = 0.5。

对每种算法运行 50 次, 求其均值, 实验结果如表 1 所列。

表 1 CEA 与 GA, EDAs, ACO 对比实验结果

算法	平均距离	平均时间(s)	平均评价量
CEA	6.1586	9.274	200
GA	5.42	8.135	2500
EDAs	6.2725	0.437	4530
ACO	5.9330	5.068	3000

从表 1 中第一列可以发现, CEA 与 EDAs 具备相当的优化效果, GA 的优化效果最差。从第二列平均花费时间来看, CEA 每次花费的时间要远远高于其他 3 种算法, 这是因为 CEA 的每次演化都需要进行复杂的学习和思维操作。其中发散思维模块的运行随着长时记忆中的知识片段的逐渐增多, 会消耗更多的运行时间, 这也说明 CEA 具有更高的知识利用率。从第三列平均个体评价量来看, CEA 所进行的评价量仅占其他算法的 4.4%~8%, 这说明 CEA 在用于对目标评价计算密集的优化问题时, 可以极大地减少所需的目标评价时间。

通过对以上实验的分析, 可以得到以下两个方面的结论:

(1) CEA 在求解复杂优化问题时, 可以比当前常用的

GA, EDAs 和 ACO 取得更高的效率。CEA 的优势来自于可以将一次目标评价所产生的知识更好地进行利用,通过复杂的思维过程代替了评价操作,从而提高了优化效率。

(2)CEA 的演化过程是可以解释的,因为 CEA 是对人类认知行为的直接模拟,符合人类的认知习惯,这是其他智能算法所不具备的。

但是 CEA 在求解中也存在以下的局限性。首先,需要目标函数评价是计算密集型操作才能体现 CEA 的优势,否则复杂的思维操作将抵消由于减少目标评价次数所带来的收益;其次,必须能够从目标评价活动中发现知识,否则反而增加了学习等操作的开销;最后,需要合理地设置 CEA 的参数。

**结束语** 创造性思维是人类社会进步的动力。本文借鉴认知心理学的研究成果,通过对人类认知系统和创造性思维过程进行建模,提出了一种以知识为基础(学习和记忆),综合两类思维技巧(发散和收敛),以价值为导向(价值体系)的新的复杂问题求解算法——认知演化算法。通过实验证实了本算法可以大大减少目标评价次数,提高优化效率。

本文的研究成果已成功应用于基于仿真的作战方案生成的研究中,可以有效地减少计算密集型仿真的运行次数,提高仿真优化效率。下一步研究需要对结论进行严格的理论推导,同时后续研究可将以非逻辑思维为主的创造性思维以及群体创新思维纳入算法框架中。

### 参考文献

[1] Goldberg D E. Genetic algorithms in search, optimization and machine learning[M]. Inc. Boston, MA, USA; Addison-Wesley Longman Publishing Co., 1989

[2] Rudolph G. Convergence analysis of canonical genetic algorithms [J]. IEEE Transactions on Neural Networks, 1994, 5(1):96-101

[3] 马永杰, 马义德, 蒋兆远, 等. 一种快速遗传算法及其收敛性[J]. 系统工程与电子技术, 2009, 31(3):714-718

[4] Bonissone P P, Subbu R, Eklund N, et al. Evolutionary algorithms + domain knowledge = real-world evolutionary computation [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3):256-280

[5] 董占球, 刘岩. 一类智能计算思想: 按自然法则计算[J]. 计算机科学, 1996, 23(03):14-16

[6] Turchin V F, Frenzt B. The phenomenon of science[M]. New York: Columbia University Press, 1977

[7] 钱学森. 创建系统学[M]. 太原: 山西科技出版社, 2001

[8] Boden M A. The creative mind: myths and mechanisms[M]. 1991

[9] Gabora L. Cognitive mechanisms underlying the creative process [C]//Proceedings of the 4th Conference on Creativity & Cognition. Loughborough, UK; ACM, 2002: 126-133

[10] 廖海林. 关于记忆与遗忘定量研究的两个数学模型[J]. 江西广播电视大学学报, 2003, 20(4):41-42

[11] Kirton M J. Adaptors and innovators: A description and measure [J]. Journal of Applied Psychology, 1976, 61(5):622-629

[12] Higgins L F. A comparison of scales for assessing personal creativity in IS [C]//Proceedings of the 29th Hawaii International Conference on System Sciences. Hawaii, US; IEEE, 1996: 13-19

[13] Higgins L F, Couger J D. Comparison of KAI and ISP instruments for determining style of creativity of IS professionals [C]//Proceedings of the 28th Hawaii International Conference on System Sciences. Hawaii, US; IEEE, 1995: 566-570

(上接第 186 页)

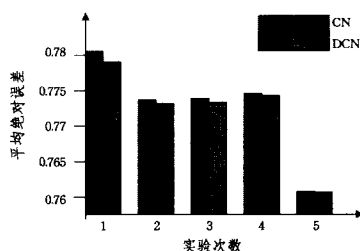


图 1 CN 和 DCN 算法的 MAE 值在 5 次实验上的结果比较

在时间复杂度方面,我们可以在查找共同评分项目的同时,计算它们的评分差,因此本方法与 CN 方法有相同的时间复杂度。

**结束语** 本文在 CN 算法的基础上考虑用户的评分信息,得到改进的 CN 算法——DCN 方法。并在 movielens 数据集上,对 CN 方法和本文提出的改进算法进行了实验对比分析。实验结果表明,本文算法 DCN 在一定程度上提高了评分预测准确度。未来的研究工作将进一步在其他网络数据集上进行试验,检测距离因素对推荐效果的影响。同时研究距离因素是否与最近邻居数有关系,以及随着最近邻居数目的改变,它对相似度的影响情况有何变化趋势。

### 参考文献

[1] Adomavicius G, Tuzhilin A. Expert-driven validation of rule-based user models in personalization applications[J]. Data Mining and Knowledge Discovery, 2001, 5(1/2):33-58

[2] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Transaction on Knowledge and Data Engineering, 2005, 17(6):734-749

[3] 梅田望夫. 网络巨变元年——你必须参加的大未来[M]. 先觉: 先觉出版社, 2006

[4] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [C]//The 14th Conference on Uncertainty in Artificial Intelligence, 1998: 43-52

[5] Liu Jie, Shang M S, Chen D B. Personal recommendation based on weighted bipartite networks [C]//The 6th International Conference on Fuzzy Systems and Knowledge Discovery, 2009, 8: 134-137

[6] Resnick P, Iacovou N, Suchak M, et al. GroupLens: An open architecture for collaborative filtering of Netnews [C]//Proceeding of ACM 1994 Conference on Computer Supported Cooperative Work, 1994: 175-186

[7] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithm [C] // The 10th Interling World Wide Web Conference, 2001: 285-295

[8] Sørensen T. A method of establishing groups of equal amplitude in a plant based on similarity of species content and its applications to analysis of vegetation on Danish commons [J]. Biologiske Skrifter, 1948, 5:1-34

[9] Leicht E A, Holme P, Newman M E J. Vertex similarity in networks [J]. Physical Review E73, 2006: 026120

[10] Shang M S, Jin C H, Zhou T, et al. Collaborative filtering based on multi-channel diffusion [J]. Physica A, 2009, 388(23):4867-4871