

# 支持创新设计的 协同进化设计系统

刘 弘  
山东师范大学

关键词：创新设计 协同设计 进化计算

## 引言

协同设计是一种群体协作的工作过程。现代产品的复杂性使得单个设计人员已不能胜任全部设计任务。参加设计的人员具有不同的背景和知识水平，只有他们之间的密切合作才能够缩短生产周期，提高产品质量，加速产品的开发进程。协作是保证产品设计成功和具有市场竞争力的关键。目前，以交互式图形系统和分析计算为主的计算机辅助设计系统，主要是为设计人员提供设计工具，仅能支持单个设计者的独立设计和人机交互，不能为启发设计人员的创造性思维提供支持，也不支持协同设计。

本文介绍如何采用进化计算方法及计算机技术，设计并实现一种支持创新设计的多Agent协同设计系统，其目标是结合多Agent技术与进化计算技术，开发支持协同设计，并为设计人员的创造性思维提供支持的设计环境。

## 支持创新设计的进化计算

生物进化过程（从简单到复杂，从低级到高级）本身是一个自然的、并行发生的、稳健的优化过程，这一过程在于使生命体达到适应环境的最佳结构与效果，通过“优胜劣汰”来达到进化的目的。

进化计算是一类模拟生物进化过程与机制求解问题的自组织、自适应人工智能技术。进化计算

的应用研究涉及很多传统方法难以解决的问题。如优化问题，分为函数优化和组合优化。函数优化是进化计算的经典应用领域，包括非线性优化、多模型优化和多目标优化等；组合优化问题有推销员问题，作业调度问题，物流调度问题，树搜索问题，背包问题，装箱问题等<sup>[2]</sup>。

在工业设计领域中，进化计算最初是用于电路设计，主要用于线路板布局等问题，后来发展为用进化计算方法的自动进化机制进行生成的设计，以及创新设计<sup>[3]</sup>。

基于遗传算法的进化模型是研究人工生命的重要理论基础。虽然人工生命的研究尚处于初级阶段，但遗传算法已在其进化模型、学习模型、行为模型和自组织模型等方面得到了应用。著名的基于进化计算的人工生命模型有：蚁群模型<sup>[4]</sup>、微粒群模型<sup>[5]</sup>、免疫系统模型和协同进化模型<sup>[6]</sup>。

由于设计（特别是有创意的设计）过程对人的智能有着强烈依赖性，如何将计算智能应用于该过程还是一个新的研究课题。

牛津大学公共科学院院长理查德·道金斯（Richard Dawkins）教授致力于生物进化理论的研究，他在1986年出版的《盲人钟表匠（The Blind Watchmaker）》一书中，展示了首个计算机进化设计程序Blinder Watchmaker，第一次将计算机界的遗传算法应用于生物形态的进化模拟实验。英国剑桥大学教授约翰·弗雷泽（John Frazer）对计算机辅助建筑设计的发展起了很大的导向作用。他的著作《进化架构（An Evolu-

tionary Architecture. Architectural Association)》<sup>[8]</sup>成为计算机辅助进化建筑设计的经典读物。澳大利亚悉尼大学设计计算与认知研究中心的约翰·席律(John Gero)教授和他的同事一起,进行了用遗传算法生成建筑平面图的研究,可以满足多个模糊的限制和目标管理。他们还展示了进化如何通过对着名的建筑风格学习,生成新的建筑物<sup>[9]</sup>。意大利米兰工艺学院的塞莱斯蒂诺(Celestino Soddu)教授利用计算机模拟技术生成艺术品,如台灯、雕刻等<sup>[10]</sup>。伦敦大学的彼得·本特利(Peter Bentley)教授<sup>[11]</sup>撰写的《计算机进化设计(Evolutionary Design by Computers)》<sup>[3]</sup>首次将世界范围内的计算机辅助进化设计学者的重要研究成果加以分类总结,较为系统的构建了计算机辅助进化设计的研究框架,为计算机辅助进化设计研究的发展奠定了基础。毕业于麻州理工学院的卡尔·西姆斯(Karl Sims)在1991年发表的《人工进化计算机图形学(Artificial Evolution for Computer Graphics)》一文标志着计算机辅助进化设计进入绘画艺术领域,并从此揭开了进化艺术这一新兴的艺术领域。3年后,他发表了《虚拟生物的进化(Evolving Virtual Creatures)》<sup>[12]</sup>,使计算机辅助进化设计进入人工生命模拟领域,对进化设计发展起到了推动作用。

在国内,潘云鹤将人工智能引入计算机图形学和CAD技术,提出综合推理和形象思维模型,并研制成功了轻纺花型、图案创作等多个新颖实用的智能CAD/CAM系统,而且在虚拟现实和计算机辅助产品创新技术等领域也取得优秀的成果,产生了显著经济效益和社会效益。潘云鹤、孙守迁等人在创新原理、创新过程的描述模型和计算模型方面做了很多探索性的工作。

浙江大学的冯培恩教授及其团队对基于仿生物学产品概念设计方法学进行了深入的研究,通过对生物基因工程与产品原理方案设计的系统类比,引出产品原理方案设计中的信息遗传基础“产品基因”的概念,揭示了产品基因在功能设计与原理方案设计之间的桥梁作用,建立了基于产品基因遗传和重组的概念设计框架。

天津大学王太勇教授及其团队提出了一种将创新设计理论与功能分析方法相结合的功能原理求解过程模型,将创新设计理论中的物场分析、效应知识库以及冲突解决原理等嵌入到总功能和分功能原理求解过程中,并将该模型用于颜料灌装机的创新设计。

南京理工大学的李东波教授及其团队进行了面向产品创新设计的功能模块划分方法研究。根据客户需求与功能单元之间的相关程度构造模糊相似矩阵,然后以功能模块内的聚合度与功能模块间的分离度作为筛选条件,把功能单元之间流的关系作为评判划分是否合理的依据,在此基础上,应用遗传算法进行产品模块划分。

上海交通大学马利庄教授及其团队开发了一个概念创新设计原型系统CIDS,提出了一种改进的计算相似度的算法,并使用该系统进行了手机产品概念设计。

浙江大学机械与能源工程学院的谭建荣教授及其团队,进行了集成环境下创新产品进化设计技术的研究,分析了创新产品设计概念和设计过程的发展规律,提出了基于语义单元的创新产品进化设计理论和方法,深入地研究了产品运动进化设计、装配进化设计、结构进化设计和重用进化设计中的关键技术。

山东师范大学通过与香港理工大学7年多的合作研究,已在支持创新设计的多Agent协同设计环境方面做出了有意义的尝试。开发的原型系统已经结合遗传算法及机器学习技术,生成了各种不同形状的手机、台灯及艺术品造型<sup>[13-15]</sup>。

## 支持创新设计的协同进化系统

### 多Agent协同设计系统

多Agent协同设计系统采用开放式结构,提供一个框架,将软件Agent、设计工具和设计人员组成一个整体。同一设计组的Agent和设计人员通过局域网,不同组的Agent之间通过互联网及移动设备进行通信,交换设计数据和知识(图1)。

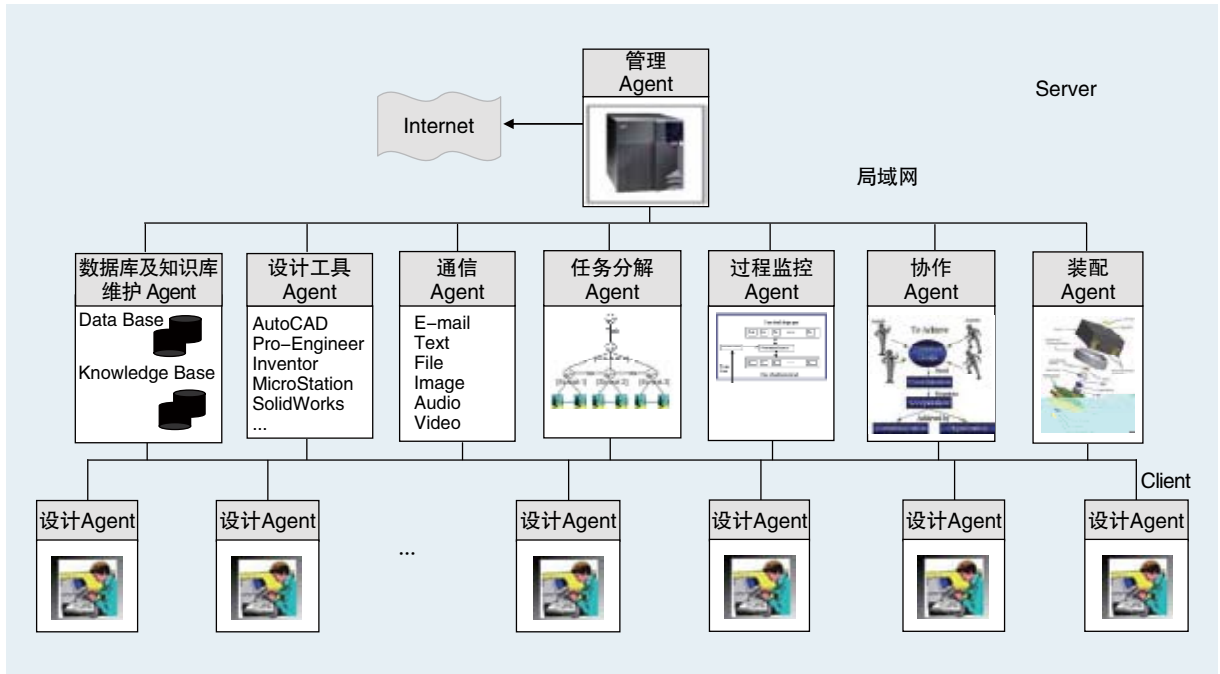


图1 多Agent系统设计系统结构

在该环境中，复杂的设计由多个Agent协作完成，每个Agent有自己独立的知识和设计决策方案，能理解设计状态表示，并能协助人类设计专家完成设计。Agent的策略依赖于确定的算法，如遗传算法、分类算法等。Agent的行为被系统事件或其它Agent所驱动。管理Agent在一组工具Agent（如共享

知识库维护Agent、公用数据库维护Agent、过程监控Agent、通讯Agent、任务分解Agent等）的协助下，组织并协调设计Agent完成设计任务。

管理Agent的知识库中存放着各Agent的名字、地址、功能及在以往设计中的绩效等历史纪录，这些信息有助于管理Agent在新的设计过程中选择合适的Agent。除了知识库，管理Agent的存储缓冲器中还存放着各设计子任务的执行状态和各个Agent的工作状态。

当有新设计任务时，任务分解Agent将整个设计任务分解为子任务集合，并用产品设计树表示分解结果。任务分解Agent的知识库中有很多产品设计树模板，根据产品种类选择合适的模板并推荐给设计工程师，由设

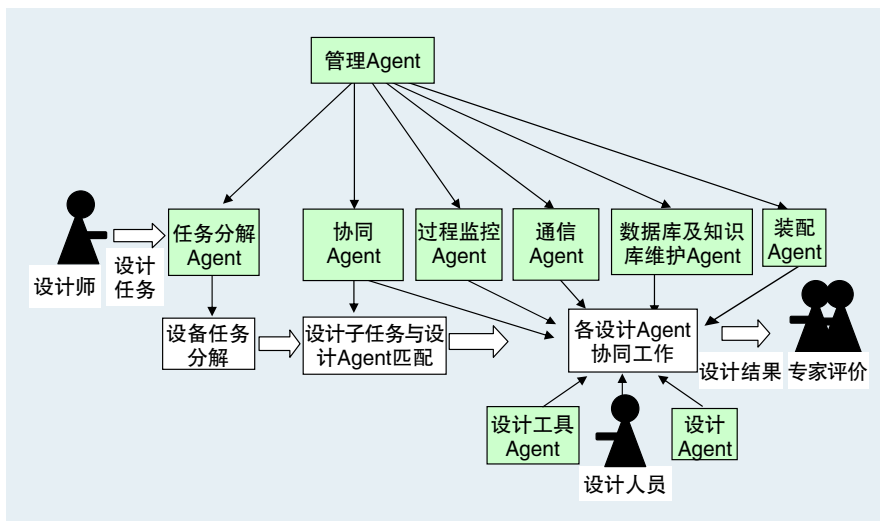


图2 多Agent协同设计过程

计工程师做初步决策并把结果发到公告板上<sup>[16]</sup>。

经过一个动态的分配过程,设计任务分配给相应的设计Agent,各Agent根据自己的任务协助设计人员工作,并在设计子任务完成后,将设计结果提交给装配Agent。装配Agent产生各部件的组合布局,然后由设计组工程师作出最后的决策。在部件组合过程中,装配Agent进行组装限制检查,并对不符合限制条件的部件,向设计Agent发送要求修改的信息。

## 基于遗传算法构件设计

基于自然选择和进化原理的遗传算法,能够自然地模拟进化过程,这一点是其它方法所无法比拟的。但是,将遗传算法应用于设计领域,必须考虑到设计的特点。因此,我们从以下几个方面对传统的遗传算法进行了改进:

1. 遗传算法中常用的固定长度编码方式不够灵活,因此采用树结构编码方法直接表示实体。这种方法有两个优点:一是容易描述具有分层结构的设计对象,二是容易修改。

2. 根据创新设计的特点和编码方法的不同,定义了树结构表示遗传算法的交叉、变异操作。

3. 遗传算法的适应度函数受许多因素影响,采用让设计人员给出适应度值的方法。这些值将作为知识保存在设计Agent的知识库中,可以在出现类似情形时重用。

4. 遗传算法用于两个设计阶段:部件设计阶段及部件组装阶段。第一个阶段用基于数结构的遗传算法产生部分构件;第二个阶段将遗传操作作用于产品树。产品树的基本元素是部件,装配Agent进行装配限制检查,遗传算法产生部件及其特征的新组合。

## 构件的集成

### 构件的产生

构件库中的构件通过4种方式产生。

**自动生成** 设计Agent执行基于数学表示二叉树结构的遗传算法,生成二维草图,然后对草图进行旋转或者扫掠以形成实体。在生成过程中,设计人

员与设计Agent进行交互,给出适应度值,以使进化过程沿着设计人员的引导进行。

**已有构件修改** 设计人员对生成的构件修改,以符合设计人员的习惯及实际设计需求。

**设计人员独立设计** 使用系统提供的实体生成及实体修改工具,手工绘制图形,生成构件,或者使用设计人员习惯使用的设计工具。

**识别、重构、修改或进化** 通过图形识别器对原有的以图片形式存放的构件进行识别重构,并在现有构件的基础上进行手工修改或者遗传操作以生成新构件。

### 构件的检索与浏览

检索的目的是根据需求从构件库中找到合适的构件进行组装。构件检索包括构件需求和目标构件库两个方面,即需求方的问题空间和构件库的解空间。问题空间由实际问题细化到设计师所理解的论域问题空间,再到查询空间,即从用户需求到库检索系统能够识别的一个确定过程。为了缩短查询时间,提高检索效率和准确性,采用自动检索和手动检索两种方式,自动检索采用模拟退火算法实现,手动检索的过程是由设计师给出检索条件,Agent计算各个构件和设计师给出的检索条件之间的匹配度,并向设计师显示满足要求的构件。设计师对构件原型实例库进行检索,检索结果有三种情况:

1. 如果找到与检索条件完全匹配的构件,则直接将其取出来,应用于后续设计。

2. 如果只搜索到与检索条件部分匹配的构件,则将构件按照其匹配值的大小顺序排列出来,设计师通过预览选择较合适的实例,并将其调入造型设计台进行修改完善,直至满足设计要求。

3. 如果搜索不到与检索条件匹配的构件,则设计师从头开始设计,通过完全独立的创作设计出全新的构件。

对于第一种情况,不存在新构件产生的问题,故而不用更新构件库。对于后面两种情况,由于对构件实例进行了修改或者创建出了全新的构件,所以需要由设计师进行评价,然后对构件库实施更新操作。

### 构件的组合布局

**定义1** 特征  $F_i$  是一个三元组  $(F_iID, t_i, v_i)$ , 其中  $F_iID$  是特征标示符,  $t_i$  是特征类型,  $v_i$  是特征值。这里的特征值是广义的, 可以是数字、字符串、数组、函数、表达式或文件等。

**定义2** 部件树定义为  $CT=(FD, FR)$ , 其中  $FD$  是特征节点集,  $FR$  是特征节点之间的限制和关系集。

**定义3** 产品树定义为  $PT=(CD, CR)$ , 其中  $CD$  是部件节点集,  $CR$  是节点之间的关系和限制集。

由上述定义, 我们可以得到一种产品的分层结构(图3)。

构件组合布局通过将遗传操作作用于产品树产生, 对父产品树之间的特征、部件执行交叉、变异等操作后, 可以产生新的子布局。经人机交互选择, 令人满意的布局可以进入下一代进化过程。

### 协同设计过程

设计工程师将设计任务分解后, 每个设计Agent根据自己的任务协助设计人员完成相应的构件设计。

**步骤1** 初始化种群。种群通过随机在操作数和操作符集中选择形成的数学表达式来生成。采用堆栈及数据结构中相应的算法检查产生的数学表达式是否是一个有效的数学表达式及表达式中的括号是否平衡, 然后, 把生成的表达式作为字符串, 根据运算顺序, 用解析算法构造数学表示二叉树。如果新产生的树无法用计算机显示生成的形状, 则该种子被淘汰。

**步骤2** 通过与设计人员交互, 得到种群中个体

的适应度值。设计人员可以选择他们感兴趣的二维草图, 查看其三维图像, 然后再修改适应度值。

**步骤3** 根据群体中个体的适应度值, 确定新的种群。

**步骤4** 对种群执行交叉和变异操作。

**步骤5** 如果设计人员不选择退出, 转步骤2。

该选择、交叉及变异运算过程一直进行到被设计人员中止。生成的构件被分类并保存在构件库中。

装配Agent执行基于产品树的遗传算法产生各构件的组合布局, 然后由设计组工程师作最后的决策。在构件组合过程中, 装配Agent进行组装限制检查, 并针对不符合限制条件的构件, 通过通讯Agent向设计Agent发送要求修改的信息。每个设计阶段结束, 设计结果将被评估, 由设计专家决策, 是否进行进一步的修改, 重复进行再设计过程, 直至满足要求。

## 多Agent协同设计系统的通信机制

### 系统协同通信过程

系统采用客户端/服务器 (Client/Server, C/S) 网络结构。传统的C/S结构设计系统一般都采用服务器端集中保存数据, 各个客户端保存数据副本。这种方式要求服务器功能复杂, 除了必备的消息转发及广播功能外, 还要求兼有维护共享数据库的功能。这些功能增大了网络负载, 降低了系统的响应性。多Agent协同设计系统网络结构与传统的C/S网络结构不尽相同, 服务器只需完成对客户端发来

信息的转发功能, 记录并保存整个设计过程中的消息日志; 系统的各个设计端(客户端)采用对等的网络结构, 任意设计人员发往服务器的合法消息都会被正确地发往其目的客户端, 整个系统的协同同步功能通过“消息驱动操作”来实现的。在网络的环境下通过对注册消

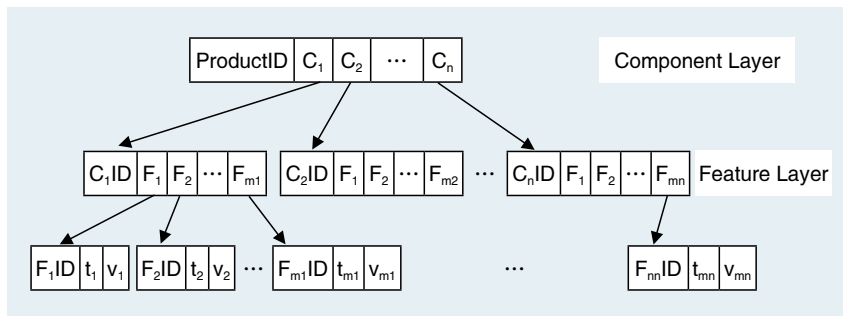


图3 产品树结构

息的传递以及客户端对消息的识别，实现设计操作在各个设计端的再现，从而保证了数据的一致性。

系统采用令牌环的方式避免冲突的产生。终端客户首先要获取令牌，然后其设计操作才会向服务器发出消息，从而得以服务器的转发。在没有获得令牌时，客户可以在本地进行设计操作，但服务器对这些设计操作所发送的消息“视而不见”。

### 系统协同通信过程

1. 客户通过发送“CreateSession”和“JoinSession”初始化创建了一个设计任务。该任务包含了一个共享的视图，视图的内容可以是一个存储于远端服务器的用户自定义的HSF文件。

2. 另一用户连接到网络通信服务器。获取当前任务列表，选定任务，加入到特定的任务，并发送“JoinSession”到服务器，由此本地创建共享视图。

3. 客户端与共享视图进行交互，如：平移、缩放、旋转，或者具体的创建实体的操作。在操作执行过程中或者执行完毕时，发送标识所做操作的全局唯一的消息，通知服务器和异地客户。

4. 服务器接受消息，对并发操作进行冲突检测，对冲突操作进行消解。广播消息，消息驱动异地客户实现本地操作。

### 基于消息驱动操作的协同通信设计

消息驱动操作是指每一项协同设计操作对应一个已经注册且全局唯一的消息，由于系统的各个设计端之间都是对等的结构，因此消息的发出端与消

息的响应端有相同的操作集合，当响应端接收到消息后，响应端会自动根据消息与操作的映射关系调用本地的操作，实现协同设计的一致性。

### 消息驱动操作的几个基本术语

**体操作函数** 协同设计环境下的基本设计操作，如创建各类实体、删除、修改等等。

**回显函数** 本地操作在异地实现时需调用的函数，是消息驱动的最终调用对象。

**消息结构** 发送的消息满足如下的结构：

消息名 ID号	segment名	参数1	…参数n
------------	----------	-----	------

**Segment名** 内部存储结构，每一项操作都要在某一segment中执行，因此，在进行操作之前以及操作之后要分别进行打开segment和关闭segment的操作。

**操作映射类** 该类中完成对一个回显函数的调用。

**消息注册** 调用注册函数将全局唯一的消息同一个操作映射类相关联。

### 消息驱动操作的几个关键步骤

1. 针对系统中的每一项实体操作函数，声明并实现一个该操作的“回显函数”。例如：如果定义一个在三维空间中创建球体的操作，需要声明并实现一个以网络传递的数据为参数的回显函数。

2. 定义操所映射类，该类中调用步骤1中声明的回显函数。

3. 将2中定义的类同一个全局唯一的消息关联，并注册消息到服务器。

## CCF推出“招聘求职”发布平台

为会员及时、方便地发布招聘或求职信息，CCF网站推出招聘求职在线发布平台。会员可以在登录状态下，实名发布招聘或求职信息。

发布招聘信息的会员需填写招聘单位及招聘岗位等相关信息，发布求职信息的会员需按照网站提供的求职模板填写个人简历，并可上传照片。招聘和求职信息均可以在个人信息管理平台进行修改。详情请点击：[www.ccf.org.cn](http://www.ccf.org.cn)的“招聘求职”板块。

(雨)

4. 在实体操作函数中添加发送消息的代码, 消息按照定义2的结构填写。

通过上述步骤, 可建立本地实体操作函数到一个全局唯一的消息、消息同操作映射类、消息映射类同回显函数的映射集合。在设计过程中, 当前设计端进行设计操作, 调用实体操作函数, 发送标识该操作的全局消息到服务器端, 经服务器确认该消息的合法性(是否已经注册), 然后将该消息广播发送到与消息的发出者属于同一个会话(Session)中的所有终端用户; 消息接受端接收到消息后, 根据第2、3步建立的消息-操作映射调用操作, 调用回显函数, 完成一次消息驱动操作的过程, 从而实现数据和操作的一致性。

## 总结与展望

设计是工业生产中最能体现人的智能并决定产品性能和成本的重要手段。创造性设计是一个极依赖于人的智慧的活动, 任何计算技术都不能取代人在设计中的作用。然而, 与传统的设计工具相比, 设计环境中进化计算技术的应用能拓宽设计

人员的想象空间, 为设计人员的创造性思考提供帮助。本文根据协同设计的特点, 提出了多Agent协同设计系统框架模型, 并介绍了在该系统中如何采用遗传算法及多Agent技术支持创新设计。尽管还有许多问题没有很好地解决, 但是我们相信, 协同设计环境中的进化技术研究最终会为计算机在设计领域中的应用, 开辟一条切实可行的路径。■



刘弘

CCF高级会员。山东师范大学教授。主要研究方向为CSCW、协同进化计算、多Agent系统。  
hongliu@sdu.edu.cn

## 参考文献

- [1] 潘云鹤, 庄越挺. 创新-产品设计的灵魂. 计算机世界报, 2004, 35期
- [2] 高隽. 智能信息处理方法导论. 机械工业出版社, 2004
- [3] Bentley, P. J. Evolutionary Design by Computers, Morgan Kaufmann Pub, 1999
- [4] 段海滨. 蚁群算法原理及其应用. 北京: 科学出版社, 2005
- [5] Y. L. Gao, Z. H. Ren. Adaptive Particle Swarm Optimization Algorithm With Genetic Mutation Operation. Third International Conference on Natural Computation (ICNC 2007), 2007: 211~215
- [6] 焦李成, 刘静, 钟伟才. 协同进化计算与多智能体系统. 北京: 科学出版社, 2006
- [7] Richard Dawkins. The Ancestor's Tale: A Pilgrimage to the Dawn of Life. UK and Commonwealth Publisher: Weidenfeld and Nicholson, 2004
- [8] John Frazer. An Evolutionary Architecture. Architectural Association Publications, London, 1995
- [9] John Frazer et al. Generative and Evolutionary Techniques for Building Envelope Design. In Proceedings of 5th International Conference GA2002, Milan, December, 2002. Politecnico di Milano University, Italy
- [10] Gero, J.S. and Peng, W. Assisting Interactions in a Dynamic Design Process: A New Role for an Adaptive Design Tool, in K Hampson, K Brown and P Scuderi (eds), Clients Driving Construction Innovation: Mapping the Terrain, CRC-CI, Brisbane, 2005: 201~210
- [11] Soddu, C. Generative Art in Visionary Variations, Art+Math=X conference, University of Colorado, Boulder, 2005
- [12] Bentley, P. J. Investigations into Graceful Degradation of Evolutionary Developmental Software. In Journal of Natural Computing, Springer, London. 2005, v4, 417~437
- [13] H. Liu, M. X. Tang, J. Frazer. Supporting evolution in a multi-agent cooperative design environment. International Journal of Advances in Engineering Software, 2002, 33(6): 319~328
- [14] H. Liu, M. X. Tang, J. Frazer. Supporting creative design in a visual evolutionary computing environment. International Journal of Advances in Engineering Software, 2004, 35(5): 261~271
- [15] Hong Liu. Generative 3D Images in a visual evolutionary computing system. Computer Science and Information Systems. 2010, 7(1): 111~125
- [16] 刘弘, 林宗楷. 一种支持动态任务分配的协同设计方法. 软件学报, 2001, 12(12): 1830~1836