

动态小生境微粒群优化技术在概念设计中的应用^{*}

刘希玉 王文平 刘弘 段会川

(山东师范大学管理学院 济南 250014)

摘要 本文提出了一种基于标准微粒群优化算法和动态小生境技术的动态小生境微粒群优化算法,该算法基于协同微粒和协同多群体模型。同时我们给出了算法在创新建筑概念中的应用。

关键词 微粒群优化,协同学习,共享小生境,学习算法,创新设计

Particle Swarm Optimization Based on Dynamic Niche Technology with Applications to Conceptual Design

LIU Xi-Yu WANG Wen-Ping LIU Hong DUAN Hui-Chua

(School of Management, Shandong Normal University, Jinan 250014)

Abstract Based on the standard particle swarm optimization (PSO) algorithm together with the widely used dynamic niche technology, this paper presents a new variation combined with the dynamic niche sharing technique on the basis of traditional PSO algorithm. We proposed a cooperative particle swarm optimization model with cooperative multi-population. Applications are given on creative conceptual architectural design.

Keywords Particle swarm optimization, Cooperative learning, Niche sharing, Learning algorithm, Creative design

1 引言

微粒群优化算法(Particle Swarm Optimization, PSO)是由 James Kennedy 和 Russell Everhart 在 1995 年提出来的^[4,5],是继蚁群算法之后的一种新的群体智能算法,已经成为进化计算的重要分支。微粒群算法自提出来之后,得到了众多相关领域学者的关注和研究^[1~3,6]。从进化计算的角度来看,微粒群算法其实是一种特殊的进化技术,是一种基于种群搜索算法,其种群的初始化为随机的,个体被称作微粒。在遗传算法中,种群的进化依赖于交叉、变异和选择等遗传算子,相对应地,在微粒群中,微粒在搜索空间中按照一定的速度和方向“飞行”,同时这个速度和方向自动调整。调整过程依赖于微粒自己的历史信息 and 种群中其它微粒的历史信息。

微粒群算法在提出后,已经得到了许多应用。2003 年,IEEE 召开了第一届群体智能国际学术会议,会议的 60 多篇论文中有 20 多篇是关于 PSO 的,这些论文已经部分发表在 IEEE Transaction on Evolutionary Computation, Vol 8 No 3, 范围包括算法、拓扑、参数、与其它进化计算技术的交叉、融合及应用等。其中,有若干文献探讨标准 PSO 的改进,如 Frans van den Bergh and Andries P. Engelbrecht^[11]提出了一种协同微粒群模型,CPSO。作者用这种 CPSO 模型协同优化解向量的各个分量。Konstantinos E. Parsopoulos and Michael N. Vrahatis^[12]提出了通过 deflection 和 stretching 技术来寻找目标函数的所有全局极小点等。

本文的另一个主题是动态小生境技术。小生境也是来源于生物界的一种模拟技术,它是自然界中环境的一个小环境,这个小环境可以用有限的同乡资源支持不同的生命类型。物种为具有相同生物特征的个体集合,它们可以在种群内部繁殖,但不能和种群之外的个体繁殖^[10]。在人工模拟系统中,

小生境一般认为是一个适应曲面上的峰,物种则为环绕峰的具有了类似特种的子群体。

小生境技术的一个重要方面是共享技术,共享主要有适应共享和隐适应共享^[10,7]。在适应共享中,适应值代表属于同一小生境的个体相互竞争的资源。适应共享需要建立鉴别个体属于同一小生境的机制,同时需要建立在它们之间共享适应度的准则。动态小生境技术由 B. L. Miller and Shaw M. J. 等在 1996 年提出(Miller, Shaw^[8])。其主要特点为动态确定形成的小生境,用动态确定的小生境来对个体进行分类,包括属于某个小生境,或不属于任何小生境。动态小生境技术已经广泛应用于遗传算法中。

本文的第三主题为创新设计。设计是一个复杂的过程,不管是建筑设计还是工程设计,所涉及到的参数都很多。在这其中,最复杂的是人的因素。一个设计是否成功,取决于它是否满足人们的功能、审美等需求,因此设计的优化从根本上是一个多目标优化过程。在确定任务之后,通过抽象化,拟定功能结构,寻求适当的作用原理及其组合等,确定出基本求解途径,得出求解方案,这一部分设计工作叫做概念设计。广义上的概念设计包含了从产品的需求分析到进行详细设计之前的设计过程。它包括功能设计、原理设计、形状设计、布局设计和初步的结构设计。概念设计的各个环节紧密相连,在设计过程中互相依赖,互为驱动。一般意义上的概念设计是指在产品的功能和原理基本确定的情况下产品外观造型的设计过程。

概念设计是设计过程中一个非常重要的阶段。不但产品的创新设计主要在概念设计阶段,而且根据有关的统计资料表明,产品工本费的 70%是在产品设计阶段决定的。同时,一旦概念设计被确定下来,产品设计的 60%~70%也就被确定下来。尤其需要提及的是,即使详细设计再好,也难以弥补

^{*} 山东省自然科学基金重大项目、国家自然科学基金项目、山东省中青年科学家奖励基金、国家自然科学基金、山东省教育厅科技计划项目资助课题、“泰山学者”建设工程专项经费资助。

概念设计阶段所出现的缺陷。在概念设计和回溯问题中通常会遇到组合爆炸问题。概念设计一般可以根据设计类型分为概念创新和概念搜索两种。概念搜索主要根据设计任务在已有的较成熟的各种方案中选择一种较好的组合。由于为了不致使系统出现“死角”，必须考虑到所有可能的组合。而回溯问题就是当产品设计层次过多，且某一设计方案不满足要求时，需要进行回溯以产生新的可行方案。产品设计往往是一个多阶段多子任务的链式序列，容易出现方案组合数量膨胀或者回溯节点过多的问题，这些都会产生组合爆炸。这一问题用传统的方法难以处理，而人工神经网络(ANN)得以广泛应用的主要原因就在于它的学习能力、多输入并行处理能力、非线性映射和容错能力，以及通过新的学习获得自适应性的能力，是处理大规模问题的一种强有力的工具。所以神经网络可以很好地解决概念设计中的组合爆炸问题。

在设计问题中，常见的优化方法有两种，即人工优化和机器优化。当这些方法与进化方法联系起来时，机器优化又称为 explorative。在这里，优化局限为一种穷举法，它仅提供问题可行解的若干方案，最终的优化过程仍由人来决定。这里面关键的问题是对设计的评价仍是一个复杂的问题，表现在量化上更是一个难以解决的问题。因此，所谓的机器优化只是一种穷举器(explorer)。

本文给出了一种新的基于动态小生境技术的微粒群优化技术。该技术具有协同的学习机制。我们给出了创新概念设计应用，并对试验进行了分析。

2 微粒群算法

微粒群算法是一种随机搜索算法，来源于模拟鸟群的捕食行为。在自然界中，一群鸟在随机搜索食物，鸟知道当前的位置离食物还有多远，那么找到食物的最优策略就是搜寻目前离食物最近的鸟的周围区域。

PSO 从这种模型中得到启示并用于解决优化问题。PSO 中，每个优化问题的解都是搜索空间中一只鸟。我们称之为“微粒”。所有的微粒都有一个由被优化的函数决定的适应值(fitness value)，每个微粒还有一个速度决定它们飞翔的方向和距离。PSO 初始化为一群随机粒子(随机解)。然后通过叠代找到最优解。在每一次叠代中，粒子通过跟踪两个“极值”来更新自己。第一个就是粒子本身所找到的最优解，另一个极值是整个种群目前找到的最优解。

2.1 基本微粒群算法

假设 PSO 在 n -维空间中进行搜索。设 $x_i = (x_i^1, x_i^2, \dots, x_i^n)$ 第 i 个微粒的当前位置，用 $v_i = (v_i^1, v_i^2, \dots, v_i^n)$ 表示其当前速度。设 $p_i = (p_i^1, p_i^2, \dots, p_i^n)$ 表示第 i 个微粒的个体最好位置， $f(x)$ 表示待优化的目标函数。个体最好位置 p_i 由下式确定：

$$p_i(t+1) = \begin{cases} p_i(t), & \text{if } f(x_i(t+1)) \geq f(p_i(t)) \\ x_i(t+1), & \text{if } f(x_i(t+1)) < f(p_i(t)) \end{cases} \quad (1)$$

现在我们假定群体中微粒的数量为 s ，所有微粒到目前为止发现的最佳位置为 $p_g(t)$ ，因此：

$$p_g(t) \in \{p_0(t), p_1(t), \dots, p_s(t)\} \\ = \min\{f(p_0(t)), f(p_1(t)), \dots, f(p_s(t))\} \quad (2)$$

则基本 PSO 的进化方程为：

$$\begin{cases} v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(p_i(t) - x_i(t)) \\ \quad + c_2 r_2(t)(p_g(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (3)$$

其中 c_1, c_2 表示加速系数， r_1 和 r_2 为取值于 $(0, 1)$ 区间的随机序列。

为保证微粒不会飞出搜索空间，速度 $v_i(t)$ 的各个分量要限制在某个区间 $[-v_{max}, v_{max}]$ 中。参数 ω 称为惯性因子，一般它在 0 和 1 之间线性变化。惯性因子的作用为改善收敛速度，加速系数控制在一次叠代中微粒的移动，一般来说，可以设为 2.0。

2.2 带有收缩因子的改进 PSO

标准 PSO 的一个变形是引入了一个成为收缩因子的参数。微粒群的运行方程为^[21]：

$$\begin{cases} v_i(t+1) = \chi(v_i(t) + c_1 r_1(t)(p_i(t) - x_i(t)) \\ \quad + c_2 r_2(t)(p_g(t) - x_i(t))) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (3)$$

这里 χ 为收缩因子，它代表了社会和认知因素，其值常常选择在 $[0, 1]$ 中的均匀分布。 χ 的值有如下公式

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}$$

其中 $\phi > 4, \kappa = 1, \phi = c_1 + c_2$ 。有关收缩因子的不同选择及其理论分析见文^[21]。

2.3 Lbest 模型

标准 PSO 使用的实际上是一种全局最优模型，Gbest 模型。为避免收敛的早熟，R. C. Eberhart, P. Simpson 和 R. Dobbins^[16] 提出了一种局部最优模型 Lbest 模型。与 Gbest 模型使用全局吸引子不同，Lbest 模型运用多重吸引子。首先将整个种群分解为若干个子群，在每一个子群中保留其局部最好位置或邻域最好位置。这样，PSO 进化方程为：

$$\begin{cases} N_i = \{p_{i-\sigma} \mid \sigma = 0, 1, \dots, 2l\} \\ p_i(t+1) \in \{N_i \mid f(p_i(t+1)) = \min_{x \in N_i} f(x)\} \\ v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(p_i(t) - x_i(t)) \\ \quad + c_2 r_2(t)(p_g(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (4)$$

注意子群 N_i 中的微粒与其在搜索空间中所处的位置无关，仅依赖于微粒的索引或编码。这样一方面避免了微粒间的聚类分析，节省了计算时间，另一方面能够加快更好解在整个群体间的扩散。

2.4 PSO 的其它改进

很多作者从其他方面对 PSO 进行了改进。根据本文的目的，我们列出其中的一部分。首先，1998 年 P. J. Angeline^[17] 提出了一个基于选择的 PSO 算法，作者根据联赛选择策略对微粒进行选择，选择的依据为适应值。过程为对微粒进行排序，然后将适应值低的一部分微粒用较好的微粒替换。具体过程如下：

(1) 选择一个个体，然后将它的适应值与其它个体进行比较，如果适应值高，将个体的分值增加 1。重复这个过程直到每一个个体都有一个分值。

(2) 根据分值对个体排序。

(3) 将分值高的一半个体复制，取代分值低的一半个体。

在另一篇文章^[18]中，P. J. Angeline 将标准 PSO 扩充，允许繁殖操作。根据预先确定的繁殖概率，选择一部分个体到繁殖池中，然后繁殖池中的个体进行交叉操作产生一部分后代，将后代取代父代，群体的总体数量保持不变。交叉操作由下式定义：

$$\begin{cases} x_a(t+1) = r_1 x_a(t) + (1 - r_1) x_b(t) \\ x_b(t+1) = r_1 x_b(t) + (1 - r_1) x_a(t) \end{cases} \quad (5)$$

$$v_a(t+1) = \frac{v_a(t) + v_b(t)}{\|v_a(t) + v_b(t)\|} \|v_a(t)\| \quad (6)$$

$$v_b(t+1) = \frac{v_a(t) + v_b(t)}{\|v_a(t) + v_b(t)\|} \|v_b(t)\| \quad (7)$$

这里参数 $r_1 : U(0,1)$ 。

2.5 小生境技术和 PSO

目前,将小生境技术与 PSO 结合的研究已经出现,例如 P. N. Suganthan^[19]。这种结合与 Lbest 模型类似,是它的一个变种,其邻域的定义基于空间位置而不是索引位置。在每一代,计算微粒之间的距离,其最大距离用 d_{max} 表示,对每一对微粒,用比值 $\|x_a - x_b\| / d_{max}$ 作为定义邻域的度量,其中 $\|x_a - x_b\|$ 表示微粒之间的距离。邻域结构是动态形成的,在初始时刻,邻域为微粒自身,然后邻域生长直至整个群体。

还有作者 J. Kennedy^[20] 讨论了邻域拓扑结构对 PSO 性能的影响,作者同时提出了几种基本的邻域结构:环状拓扑、轮状拓扑、邻域拓扑等。

3 一种改进的动态小生境模型

动态小生境技术最早由 B. L. Miller 和 Shaw M. J.^[8] 在 1996 年提出,其主要思想为动态确定小生境以及最优点,同时将所有点按照小生境以及非小生境进行分类。在动态小生境中,在一个小生境内的适应值共享为其原始适应值除以小生境的大小。或者,个体不属于任何小生境。应用已经证明,动态小生境技术比标准的适应值共享具有更好的寻优效果。

在本文中,我们引入了一个动态模型,即动态协同进化模型。此模型融合了动态小生境共享技术与进化算法的优点。事实上,动态小生境模型具有以下几个方面的特点。首先,整个群体可以动态地分成多个子群体。因为小生境核本身就是动态的,所以这一动态过程可以自动实现。其次,动态小生境模型中的小生境核包括了问题的当前最好解。第三,小生境核结构是动态的,这取决于所有子群体的进化过程,反过来它也影响小生境进化进程。

动态协同小生境模型可以用来自动地给小生境定位。由于通常情况下最优解并不是均匀分布的,因而这一动态模型可以通过消除那些没有最优解的区域来有效地减少计算时间。通过这种技术,我们可以以一个可以接受的计算度找到所有可能的最优解。

子群体之间的协同方式有两种:基于小生境核的共享协同和完全协同。一般来说,第一种协同方式简单直接。但是,完全协同也有它的优点,因为我们可以共享好的遗传基因和小生境半径、权等参数信息,这些信息对其它小生境也是有效的。因此通过这种方式这样可以加速整个子群体的进化进程。

图 1 表明了动态小生境模型中的协同关系,其中小生境核是所有子群体的中心的集合,它包含了共同进化模型中的最终最优解。在进化进程中,这一结构中的个体就是当前最优解。小生境核的初始化可以从显型空间中随机选择。

群体由小生境核动态地分为许多子群体。这一动态过程按照一定的策略按照某种度量进行。例如,适应值、小生境间的距离都可能是决定这一动态方式的依据。在进化中每一个子群体将产生一小生境并增加到小生境核结构中。

对于完全协同模型,一种普通的方式是迁移。在这种方式中,通过动态共享小生境技术所有子群体的进化是相互独立的。当新的小生境产生时,某一子群体中最好小生境的一

部分被选出并以一定的概率移植到其他子群体中。在这一模型中,子群体将交换好的遗传基因。其他的方式还有互相迁移、单向迁移、渗透等。

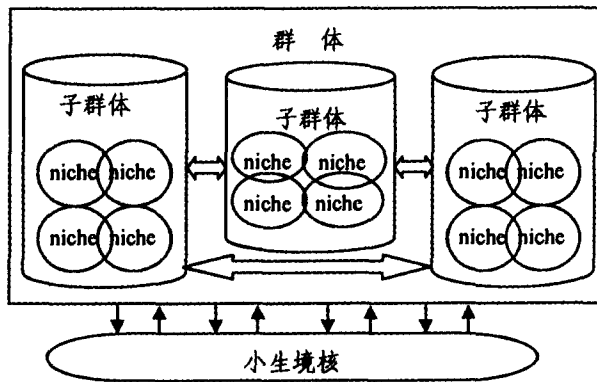


图 1 动态小生境模型

4 协同学习

按照协同的观点,我们可以将遗传算法看作具有协同机制的学习机^[9~11]。Clearwater 等人^[9]对协同作了如下定义:协同是指在解决问题时通过交流信息而互相影响的代理的集合。显然,通过把遗传算法的群体成员作为代理,并通过信息交换,我们就能把遗传算法看作是一个协同系统^[11]。

根据这一观点,Frans van den Bergh 和 Andries P. Engelbrecht^[11]提出了协同微粒群优化算法,即 CPSOs。在这一算法中,我们不再用一个群去寻找最优向量,而是把这个向量分解使得这些微粒群成为单维的最优向量。因此,我们提出了一种并行协同学习模型。然而,我们并不是沿着分解的思路进行,而是使用基于代理的并行模型。

4.1 PCLM 模式

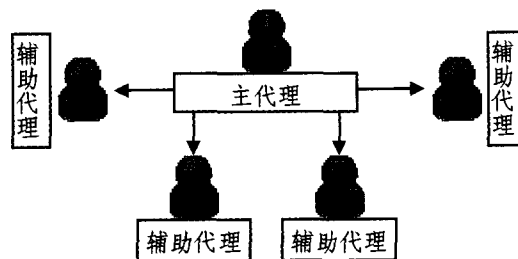


图 2 主从模型

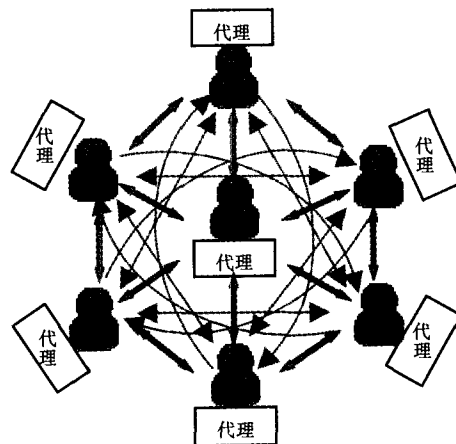


图 3 岛屿模型

我们把整个群体分成几个以并行模式进化的子群体，每个子群体由一个代理操纵。这些代理在一般的并行体系结构中扮演了 CPU 的角色并形成了两种模型：主从模型和岛屿模型。

在主从模型中，定义一个特殊的代理作为主代理。每个子群体由一个辅代理负责。主代理监督并分配整个群体的任务和工作。主代理从整个群体中选择好的个体并将其种子移植到所有子群体中。子群体的代理们接受这些种子并进行选择和舍弃，移植操作可以一对多或者一对一。

在第二种模型中，每一个代理被分配到一个子群体中，来监督它所在子群体的进化过程，子群体在一定的时间间隔内互相交换好的个体，迁移操作可以一对多或者一对一。

4.2 迁移

迁移主要用来在子群体间交换好的个体。对于岛屿模型有两种迁移模式，一对一和一对多。

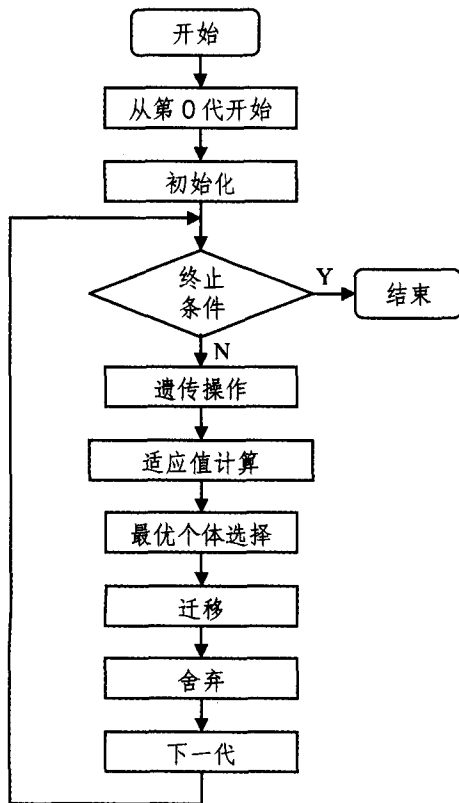


图4 一对多迁移

在一对多模式中，每一个子群体都和其它所有的子群体共享每一代中所产生的最好个体。只要它接收到一个好的个体，就会根据适应值把最差的个体舍弃。一对多模式将加速群体收敛并减少群体多样性。

第二种是一对一模式。多样性是这一模式中涉及到的主要因素。每一子群体定一发送比率和发送概率来发送最优个体。发送速率决定了发送个体的数量发送概率决定了这一操作发生的频率。被发送的个体在选择时可以是随机的也可以是根据适应值设定的优先级来定。而这些个体的接收方可以根据就近原则或者环形拓扑布局来定。

5 基于动态小生境的微粒群优化

在这一节我们提出了基于动态小生境技术的协同微粒群优化模型。通常情况下适应值函数是非线性的而且往往不只

有一个峰值。在这里我们仅仅假设适应值函数或者说目标函数有有限个峰值且这些峰值之间有最小距离。我们将会定义另外一个改进的适应值函数。简单地说，最初的适应值函数可以称作原始适应值函数。以后我们将对这一模型做详细的描述。

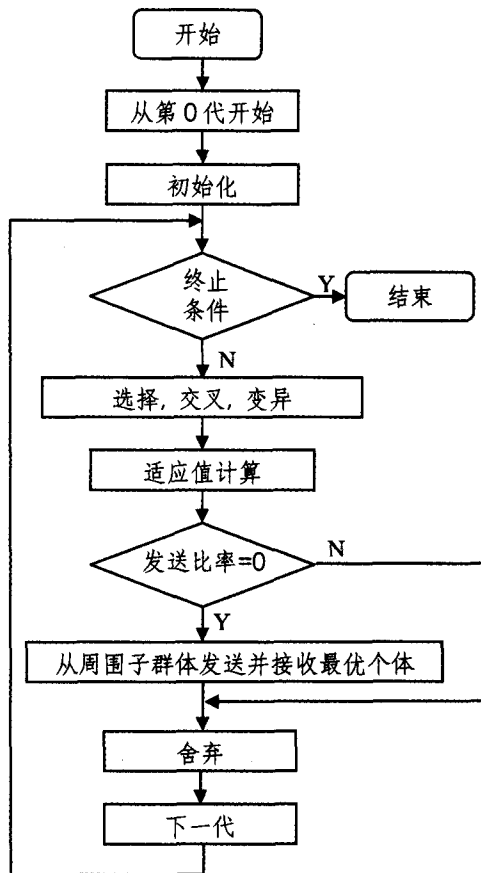


图5 一对一迁移

动态小生境的关键在于确定形成小生境的峰值并把所有的个体分类到相应的小生境或者无峰种类中。现在我们在原始适应值函数的基础上定义微粒的适应值。设群体 P 中，有以 i 为索引的微粒 ξ_i 。群体 P 被分到以 j 为索引的小生境中。用 n_j 表示小生境 j 的群体大小并设原始适应值为 $f(\xi_i)$ 。假设 ξ_i, ξ_j 为两个微粒，定义它们之间的距离为：

$$d(\xi_i, \xi_j) = \xi_i, \xi_j \text{ 之间的海明距离} \quad (8)$$

我们注意到公式(8)适用于二进制模式的群体。如果微粒是以向量形式表示的，我们可以用欧几里得距离表示 $d(\xi_i, \xi_j)$ ：

$$d(\xi_i, \xi_j) = \xi_i, \xi_j \text{ 的欧几里得距离} \quad (9)$$

对于一个小生境来说，它的半径就是它的微粒中两两之间距离的最大值。我们用下面得函数定义个体 x_i 的小生境共享范围：

$$m_i = \sum_{j=1}^N \text{share}(\sigma_0^{-1} d(\xi_i, \xi_j)) \quad (10)$$

σ_0 是小生境半径，函数 $\text{share}(t)$ 在 $[0, 1]$ 之间递减，如下所示：

$$\text{share}(t) = \begin{cases} 1, & t \leq 0; \\ 1 - \phi(t), & t \in [0, 1] \\ 0, & t \geq 1 \end{cases} \quad (11)$$

$\phi(t)$ 在 $[0, 1]$ 之间递增，例如下面的函数：

$$\phi(t) = t^\alpha \text{ 且 } \alpha > 0 \quad (12)$$

我们考虑到

$$dm(\xi_i) = \begin{cases} \frac{m_i}{n_j}, & \text{if } \xi_i \text{ is within niche } j, \\ 1, & \text{otherwise} \end{cases} \quad (13)$$

然后我们如下定义微粒 i 的动态适应值

$$df(\xi_i) = \frac{f(\xi_i)}{dm(\xi_i)} \quad (14)$$

这是一个可变化的适应值。如果一个微粒适应性比较强,它的适应值将减小,否则,将保持不变。因此,我们可以用动态小生境技术将所有可能的微粒定位。因为小生境的进化进程是动态的,所以这一过程也是动态的。

为了使群体能够在没有小生境信息的先验知识下进化,我们动态地将群体分成几个子群体。首先,群体先分成一个子群体,即它本身,然后进化开始。当某一子群体发现最优解时,我们就把它放在子群体中包含最优解的小生境核中,并调整子群体的数量和小生境信息。

下一步我们根据小生境核和子群体的新设置重新开始进化进程。如果一个群体在小生境核的最优解的动态小生境范围内,那么我们根据式(14)重新计算它的适应值,并将进化进程继续下去,直到发现新的最优解加入到核中,或达到预先设定的终止条件后停止。

我们用 Ω 表示小生境核结构,为了得到确切的可能解决方案,小生境要取一个最小值,这样我们定义 σ_0 为小生境半径的最小值。那么对于解决方案 ζ ,我们可以用它与小生境核内其它解决方案的距离计算出动态小生境的半径

$$2\sigma(\zeta) = \max\{2\sigma_0, \rho(\zeta, \Omega \setminus \{\zeta\})\} \quad (15)$$

设 N_i 为子群体,局部最优微粒核全局最优微粒如下所述:

$$\begin{cases} p_i(t+1) \in \{N_i \mid f(p_i(t+1)) = \min_{x \in N_i} f(x)\} \\ p_g(t) \in \{p_i(t) \mid f(p_g(t)) = \min_i f(p_i(t))\} \end{cases} \quad (16)$$

为了提高全局寻优效果,我们把随机产生的一些微粒加入到每一子群体中。对于某一子群体,我们按照适应值将其所有微粒分类,选择一定数量的适应性差的微粒 β ,并将其用在寻优空间中得到的微粒替换掉。为了避免局部收敛, β 应是比较小的整数。如果 n_i 为子群体 N_i 的微粒数,那么

$$\begin{cases} \text{sort } N_i \text{ according to fitness value} \\ x_j(t+1) = \text{rand}(x_j(t)), \text{ for } n_i - \beta < j \leq n_i \end{cases} \quad (17)$$

对于其他微粒,进化方程为

$$\begin{cases} v_i(t+1) = \chi[\omega v_i(t) + c_1 r_1(t)(p_i(t) - x_i(t)) \\ \quad + c_2 r_2(t)(p_g(t) - x_i(t))] \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (18)$$

这里 χ 是收缩因子,通过下面公式得到:

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (19)$$

其中 $\phi > 4, \kappa = 1$, 且 $\phi = c_1 + c_2$ 。

通过动态小生境技术,每一小生境只能找到一种解,这样可以避免在不合理的狭小范围内找到最优解。小生境的最小值可通过值 σ_0 调整。然而,这在很大程度上取决于问题的编码情况。与基因型距离相比,这个更容易通过先验知识来估计。图 6 是协同进化 PSO 模型。

6 Eco-conscious 建筑设计

基于协同思想的建筑群设计目前已成为一种趋势,尤其在中国大陆。具有地方特色的建筑设计最根本的内容是先进技术的使用。先进技术在概念设计中的应用和地方的文化习

俗是产生具有民间特色设计的重要因素。这样不仅使其在国内具有卓越的竞争质量,而且更容易打入国际市场^[7,22]。

为了协调国内影响力和国际环境因素间的平衡,我们将注意力放在在进化建筑形态上。我们所关心的既不是在北纬地区增大室内受光面积也不是在热带地区减小室内受光面积(在拥有较大建筑面积的情况下)。考虑到建筑设计的广泛应用性,可能会有很多复杂性的约束。但是为了简化应用模型,我们只考虑室内受光面积和实用面积。

6.1 微粒及其寻优空间

寻优空间是由所有的个体和微粒组成的,我们把个体定义为如图 7 所示的结构。在这一结构中,个体由三部分组成,即:区域内建筑物的总数量、描述建筑分布的规划设计、建筑物相关信息。每一建筑单元包括位置,高度,类型和其它尺度。

为了简单起见,我们用一整型变量 ζ 表示建筑物的类型,其二维空间用两个半径 R, r 表示。因此对于某一个体,微粒可用下面公式表述:

$$\xi = (\chi, B_1, \dots, B_n) \quad (20)$$

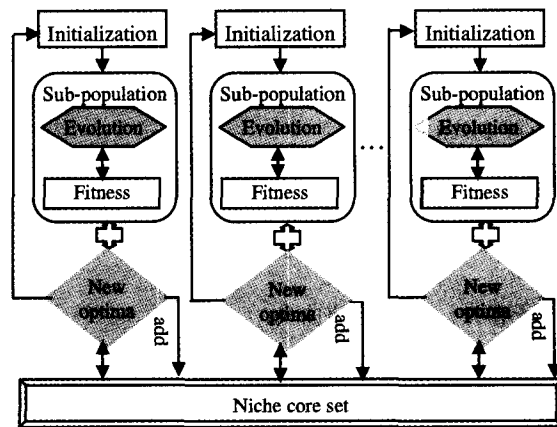


图 6 协同进化 PSO 模型

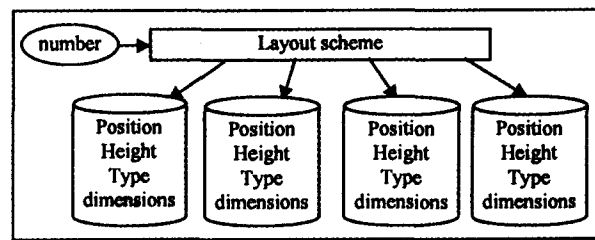


图 7 Phenotypes

在式(20)中, n 代表建筑物的数量, χ 是表示设计方案的整型变量, B_1, \dots, B_n 是建筑物的参数,具有以下形式:

$$B_k = (x^k, y^k, z^k, h^k, \zeta^k, R^k, r^k) \quad (21)$$

这样,我们直接用式(20)和(21)形成微粒而不用遗传算法的模式。为了限制参数在一定的范围内,我们假定式(21)中所有参数在区间 $[-M, M]$ 中,其中 M 是预先设定的值,式(20)中的整数 n 具有最大值。

式(21)中的参数和第 5 部分讲述的那样随着微粒的变化而变化,对于整型变量 χ 取其最近的值。只有一个变量记录微粒长度的变化,但我们可用下面的方法克服这一困难。一方面可以舍弃不必需的数据,另一方面就是若数据不充分,用随机数据填充,这样就可以保证数据结构的完整性。因此,微粒的长度在进化过程中会动态地调整。

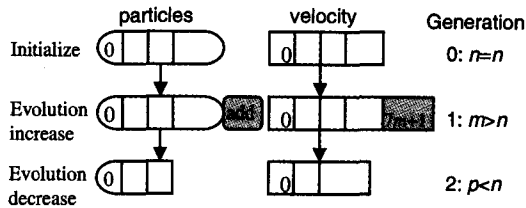


图8 速度和维

6.2 测试与评价

设微粒 ξ , 建筑物的面积由它的大小、形状、高度决定, 我们定义面积为:

$$A(\xi) = \sum_{k=1}^n \alpha(\xi^k) R^k r^k h^k \quad (22)$$

这里我们用 $\alpha(\xi^k)$ 代表一常量, 依赖于建筑物的面积与形状之间的关系。我们再考虑阳光这一约束条件, 定义 t 为一天的标准时间, 即在白天有阳光时, t 在 0 到 π 之间变化。可设 $vsun(t)$ 为阳光的方向函数。另外, $v_0 \in (0, 1)$ 体现了中午光线的最大角度。这样可用下面的等式描述向量 $vs(t)$:

$$vs(t) = (v(t)\cos(\pi-t), v(t)\sin(\pi-t), -v_0\sin(\pi-t))$$

$$v(t) = \sqrt{1-v_0^2\sin^2(\pi-t)}$$

阳光的强度在不同的时间按下面的公式变化:

$$\rho(t) = \rho_0 \sin(\pi-t)$$

现在如果 ξ_0 是某个个体里高度为 h_0 的建筑物, 中心位置是 $(x_0, y_0, 0)$, 那么在时间 t 越过建筑物的最高点的光线等于日光参数 u :

$$P = (x_0, y_0, 0) + vs(t)u$$

影子在 $x-y$ 平面形成曲线:

$$\begin{cases} x = x_0 + \frac{h_0}{v_0} v(t) \text{ctg}(\pi-t), \\ y = y_0 + \frac{h_0}{v_0} v(t) \end{cases}$$

假设我们处于北半球, y 轴方向指向北极点, 用 Θ_0 表示影子区域, 并且 χ_{Θ_0} 是 Θ_0 的特征函数。如果一建筑物有图(5.2)所示的影子, 而正好有高度为 h_1 , 中心位置为 $(x_1, y_1, 0)$, 最大半径为 R^1 的建筑物 ξ 坐落在此影子区域内。现在计算它所接收的总的光线, 为了方便起见, 我们将它的原点移到第一座建筑物的位置, 这样它的原点为 $(x_1 - x_0, y_1 - y_0, 0)$ 。假设 $x_0 = y_0 = 0$, 我们可用 D_1 定义 ξ 的面积:

$$D_1 = (x_1 - 0.5R^1, x_1 + 0.5R^1) \times (y_1, y_1 + h_1)$$

因此它被其它建筑物的影子遮住的面积为

$$R_s(1, 0) = c_0 \iint_{D_1} \chi_{\Theta_0} \rho(t) dx dy \quad (23)$$

其中 c_0 是常量。因此一个个体所受到的影子为

$$R_s(\xi) = c_0 \sum_{\substack{i, j=1 \\ i \neq j}}^n \iint_{D(i)} \chi_{\Theta(j)} \rho(t) dx dy \quad (24)$$

现在可以看出算法应解决的问题为

$$\begin{cases} \text{find individual } \xi \text{ such that} \\ \max A(\xi), \\ \min R_s(\xi) \end{cases} \quad (25)$$

我们选择两个函数 $\Gamma(u), \Delta(u, v)$ 和两个变量 γ, δ 。 $\Gamma(u)$ 和 $\Delta(u, v)$ 随着 u 和 v 而逐渐递增, 然后适应值函数可以如下定义:

定义1 对于特定微粒, 用下面的能量等式定义适应值函数

$$f(\xi) = \gamma\Gamma(1+A(\xi)) + \delta\Delta(A(\xi), R_s(\xi)) \quad (26)$$

下面我们直接定义建筑物间距离限制。设 $D(i, j)$ 是建筑物 i, j 间的距离。定义 i 和其它建筑物间的最小距离:

$$D(i) = \min_j D(i, j)$$

这样一个个体总的距离限制可以描述为

$$D(\xi) = \sum_{i=1}^n D(i)$$

因此我们有了一个适应值或能量函数的变量定义:

定义2 对于特定微粒, 定义适应值函数

$$f(\xi) = \gamma\Gamma(1+A(\xi)) + \delta\Delta(A(\xi), R_s(\xi)) + \epsilon\Lambda(D(\xi)) \quad (27)$$

其中 $\Lambda(u)$ 是递增函数, ϵ 是正常数。

7 实验数据

在这一部分中, 我们给出一些实验数据。其中适应值函数如图9中所示进化进程中的适应值曲线。

这些实验数据中我们可以看出, 我们描述的这一模型可以很好地解决问题。还有一点我们发现适应值最优化方法相对于传统遗传算法优点不是很多, 但它更直接更简单。

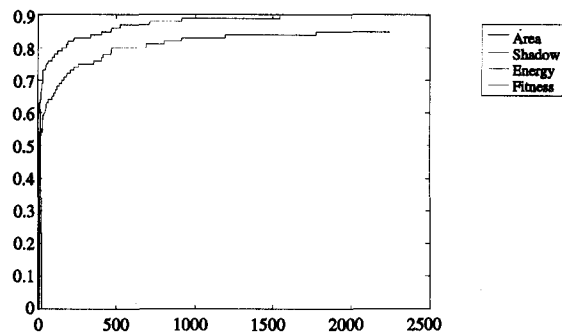


图9 适应值曲线

结论 本文中, 基于标准 PSO 算法和广泛应用的动态小生境技术, 我们提出了基于传统 PSO 算法的动态小生境共享技术。同时使用了协同微粒群优化算法模型, 最后我们给出了算法在建筑创新概念中的应用。虽然 PSO 在建筑创新设计方面仍存在很多问题, 但它却为我们提供了一种具有一定发展潜力的新方法。

致谢 本文受“泰山学者”建设工程专项经费资助。本文同时受到香港理工大学合作研究基金 G. YY. 34)、山东省自然科学基金重大项目(Z2004G02)、山东省中青年科学家奖励基金(03BS003)、山东省教育厅计划项目、国家自然科学基金的大力支持, 在此表示感谢。

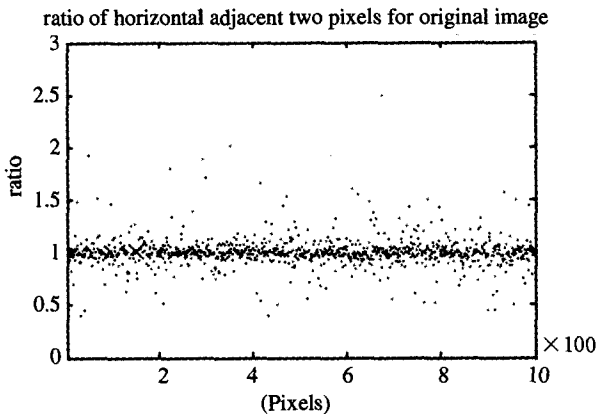
参考文献

- 1 Bonabeau E, Dorigo M, Theraulaz G. Swarm Intelligence. From Natural to Artificial Systems. New York: Oxford University Press, 1999
- 2 Wu Bin, Zheng Yi, Fu Weipeng, et al. A customer behavior analysis algorithm based on swarm intelligence. Chinese Journal of Computers, 2003, 26(8):913~918
- 3 Chen Ling, Shen Jie, Qin Ling, et al. An adaptive ant colony algorithm based on equilibrium of distribution. Journal of Software, 2003, 14(8):1380~1387
- 4 Kennedy J, Eberhart R C. Particle Swarm Optimization. In: Proc. IEEE Int Conference on Neural Networks, Piscataway, NJ: IEEE Service Center, 1995. 1942~1948

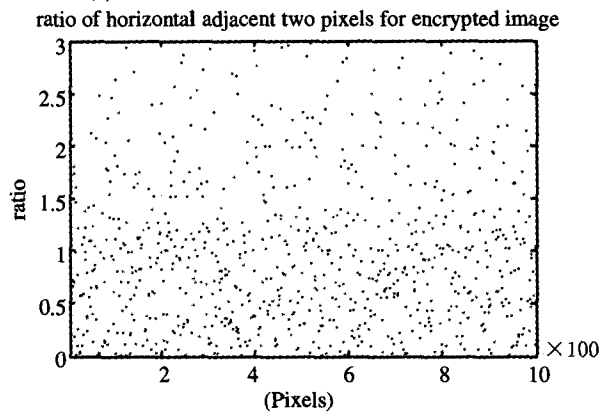
(下转第 209 页)

坏像素间的这种相关性,从而增强算法的抗统计分析能力。可以借助概率论的相关系数来衡量相邻像素的相关性^[7],相关系数定义如下:

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)} \sqrt{D(y)}} \quad (13)$$



(a) 未加密图像的水平相邻像素的灰度值之比



(b) 加密图像的水平相邻像素的灰度值之比

图6 图像的水平相邻像素的相关性

其中: x, y 是相邻像素的灰度值。在数值计算过程中, $E(\cdot)$, $D(\cdot)$ 和 $\text{cov}(\cdot)$ 计算如下:

$$E(x) = \frac{1}{N} \sum_{i=1}^N x_i \quad (14)$$

$$D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2 \quad (15)$$

$$\text{cov}(x, y) = E[(x - E(x))(y - E(y))] = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y)) \quad (16)$$

仿真实验演示了从明文图像和加密图像中随机选取的1000个水平相邻像素对的灰度值之比(图6)。在图6(a)中,大多数水平相邻像素的灰度值之比接近于1,表明相邻像素的相关性比较高。而在图6(b)中,大多数水平相邻像素的灰度值之比比较分散,表明图像经加密后相邻像素的相关性较低。

结论 本文提出了一种新的基于随机混沌序列的图像加密算法,理论分析和仿真实验表明,本文的算法具有良好的密钥敏感性和很大的密钥空间,同时具有较好的抗统计攻击、差分攻击和选择明文攻击能力。不过,由于在加密的过程中,每个密钥的生成要经过多次迭代,因此本文的算法在加密的速度上有待进一步提高。

参考文献

- 1 Wang K, Pei W J, Zou L H, et al. On the security of 3D Cat map based symmetric image encryption scheme. *Physics Letters A*, 2005, 343: 432~439
- 2 Li S J, Mou X Q, Cai Y L, et al. On the security of a chaotic encryption scheme; problems with computerized chaos in finite computing precision. *Computer Physics Communications*, 2003, 153: 52~58
- 3 Alvarez G, Montoya F, Romera M, Pastor G. Cryptanalyzing an improved security modulated chaotic encryption scheme using ciphertext absolute value. *Chaos, Solitons and Fractals*, 2005, 23: 1749~1756
- 4 Alvarez G, Montoya F, Romera M, Pastor G. Cryptanalysis of dynamic look-up table based chaotic cryptosystems. *Physics Letters A*, 2004, 326: 211~218
- 5 Alvarez G, Montoya F, Romera M, Pastor G. Cryptanalysis of an ergodic chaotic cipher. *Physics Letters A*, 2003, 311: 172~179
- 6 Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. *Int J Bifurcat Chaos*, 1998, 8(6): 1259~1284
- 7 Chen G R, Mao Y B, Chui C K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons and Fractals*, 2004, 21: 749~761
- 8 Kennedy J, Everhart R C. A discrete binary version of the particle swarm algorithm. In: *Proceedings 1997 Conference on Systems, Man and Cybernetics*. Piscataway, NJ: IEEE Service Center, 1997. 4104~4109
- 9 Zeng Jianchao, Jie Jing, Cui Zhihua. *Particle Swarm Optimization Algorithm*. Beijing: Science Press, 2004
- 10 Liu Xiyu, Tang Mingxi, Frazer J H. An eco-conscious housing design model based on co-evolution. *Advances In: Engineering Software*, 2005, 36: 115~125
- 11 Miller BL, Shaw M J. Genetic algorithms with dynamic niche sharing for multimodal function optimization. In: *Proc of the IEEE International Conference on Evolutionary Computation*. Piscataway, NJ: IEEE, 1996. 786~791
- 12 Clearwater S H, Hogg T, Huberman B A. Cooperative problem solving. In: *Computation: The Micro and Macro View*, Singapore, World Scientific, 1992. 33~70
- 13 Cioppa A D, De Stefano C, Marcelli A. On the role of population size and niche radius in fitness sharing. *IEEE Trans. Evolutionary Computation*, 2004, 8(6): 580~592
- 14 van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization. *IEEE Trans Evolutionary Computation*, 2004, 8(3): 225~239
- 15 Parsopoulos K E, Vrahatis M N. On the computation of all global minimizers through particle swarm optimization. *IEEE Trans Evolutionary Computation*, 2004, 8(3): 211~224
- 16 Garcia-Pedrajas N, Hervás-Martínez C, Ortiz-Boyer D. Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE Trans Evolutionary Computation*, 2004, 9(3): 271~302
- 17 Cioppa A D, De Stefano C, Angelo Marcelli. On the role of population size and niche radius in fitness sharing. *IEEE Trans. Evolutionary Computation*, 2004, 8(6): 580~592
- 18 Liu Xiyu, Tang Mingxi, Frazer J H. An eco-conscious housing design model based on co-evolution. *Advances in Engineering Software*, 2005, 36: 115~125
- 19 Eberhart R C, Simpson P, Dobbins R. *Computational Intelligence PC Tools*. Academic, 1996, 6: 212~226
- 20 Angeline P J. Using selection to improve particle swarm optimization. In: *Proc IEEE World Congress on computational intelligence*, ICEC-98, Anchorage, Alaska, 1998. 84~89
- 21 Angeline P J. Evolutionary optimization versus particle swarm optimization; philosophy and performance differences. In: Porto W, Saravanan N, Waagen D E, et al. Eds. *Evolutionary Programming VII, 7th International Conference*, EP98. San Diego, CA, USA, 1998. 601~610
- 22 Suganthan P. Particle swarm optimiser with neighbourhood operator. In: Angeline, P J, Michalewicz Z, Schoenauer M, *Proceedings of the Congress of Evolutionary Computation*, Vol 3. IEEE Press, 1999. 1958~1962
- 23 Kennedy J. Small worlds and mega-minds; effects of neighborhood topology on particle swarm performance. In: *Proc Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Service Center, 1999. 1931~1938
- 24 Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Computation*, 2002, 6: 58~73