

DOI:10.3969/j.issn.1673-4785.201104014

网络出版地址: <http://www.cnki.net/kcms/detail/23.1538.TP.20120316.1014.001.html>

求解旅行商问题的混合粒子群优化算法

沈继红¹, 王侃²

(1. 哈尔滨工程大学 理学院, 黑龙江 哈尔滨 150001; 2. 哈尔滨工程大学 自动化学院, 黑龙江 哈尔滨 150001)

摘要: 为高效解决旅行商问题, 结合光学寻优算法、混沌优化算法、粒子群优化算法, 提出了一种新的混合智能优化算法, 应用光学寻优算法的优点, 为粒子群中粒子找到了一组最优的初始值, 引入交换子、交换序列、混沌序列, 提出了适合旅行商问题的光学混沌粒子群算——并严格证明了新算法的稳定性、收敛性。数值实验仿真结果表明, 该算法收敛速度快、迭代次数少, 能快速找到令人满意的最优解, 为解决旅行商问题提供了新的思路。

关键词: 旅行商问题; 混沌优化算法; 费马原理; 粒子群算法; 光学寻优算法

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1673-4785(2012)02-0174-09

The light ray particle swarm optimization for solving the traveling salesman problem

SHEN Jihong¹, WANG Kan²

(1. College of Science, Harbin Engineering University, Harbin 150001, China; 2. College of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: A new hybrid intelligent optimization was given to solve the traveling salesman problem (TSP) by introducing the thought of an LRO algorithm, chaos optimization algorithm, and particle swarm optimization (PSO). A group of optimal initial values were found by using the features of LRO. Next, by employing the method of discrete chaotic particle swarm optimization and introducing the swap operator, swap sequence, and chaos sequence, an optical chaos PSO adaptive for the TSP problem was proposed. The stability and convergence of the optimization was proved decisively. The numerical simulation results show that this new optimization method has a good convergence rate and less iterative steps, thus allowing a satisfactory solution to be found rapidly. The method provides a new inspiration for solving the TSP problem.

Keywords: travel salesman problem; chaos optimization algorithm; Ferma's principle; particle swarm optimization; light ray optimization

优化问题可以自然分为2类:一类是连续变量的优化问题;另一类是离散变量的优化问题,即所谓的组合优化问题。旅行商问题(travel salesman problem, TSP)是组合优化问题中的一个著名NP难题, TSP因其典型性已经成为许多启发式搜索、优化算法的间接比较标准。同时TSP也是一个具有广泛的应用背景与重要理论价值的组合优化难题,对求解该问题高效的全局优化算法的研究,一直被科学界和工程界所高度重视。

TSP问题的求解方法归纳起来可以分为得到最

优解的精确算法和找到近似解的近似算法。完全枚举法、动态规划法和全局搜索算法属于精确算法。TSP问题精确算法的运行时间是指数级复杂度,难以适应大规模的实例,随着对TSP问题的认识加深,精确算法的研究越来越少。近年来受到自然界的启发,人们提出了各种各样的计算智能方法,如人工神经网络、遗传算法、蚁群优化算法、粒子群优化算法和人工免疫系统。智能优化算法为解决TSP问题提供了新的思路,它们被广泛地应用于各种NP难题的优化问题求解,虽然不能保证获取最优解,但在问题规模较大时也可以在可行时间内找到满意的解。

粒子群优化算法(particle swarm optimization, PSO)是一种群智能优化方法,它是由美国社会心理

收稿日期:2011-04-24。网络出版日期:2012-03-16。

基金项目:黑龙江省自然科学基金资助项目(F200931)。

通信作者:王侃。E-mail: wangkan198600@163.com。

学家 Kennedy 和电气工程师 R. Eberhart 在 1995 年提出的,它利用了生物群体中信息共享的思想,其概念简单、易于实现,同时又有深刻的智能背景,既适合科学研究,又适合工程应用. 因此,PSO 一经提出,就引起了众多学者的关注,得到了非常广泛的应用. 为解决组合优化问题, Kennedy 等^[1]首先提出了 PSO 算法的离散二进制版; Clerc^[2]提出了求解 TSP 问题的离散粒子群优化算法,对 TSP 问题的求解重新定义粒子的位置、速度和相关运算,但其性能与其他算法相比仍有不小的差距;高尚等^[3]在粒子群算法中加入遗传算法思想,构造了混合算法; Hendlass 等^[4]通过对离散 PSO 的每个粒子增加记忆功能,成功解决了一个小规模 TSP; 王康平等^[5]通过引入“交换子”和“交换序”的概念,给出了另一种解决 TSP 的 PSO 方法,为求解 TSP 问题提供了新的思路. 但在算法的收敛速度方面以及在城市规模较大的情况下,现有文献中的粒子群算法都存在着一定的缺陷,本文试图通过与其他智能优化算法的结合来解决这一问题. 光学寻优算法^[6]是 2007 年沈继红教授提出的模拟光自然属性的智能优化算法,利用在可行域中填充正方形介质,模拟光的折射以及反射现象,通过最基本光学定律找到最优值,算法迭代机理简单、收敛速度快,具有很强的并行计算能力. 2010 年,李焱等^[7]给出了基于正六边形网络的光学寻优算法,在高维迭代中具有良好的仿真效果,李加莲等^[8]给出了光学寻优算法的基本理论证明,并与其他算法进行了比较,完善了算法的理论体系. 本文通过正方形网络光学寻优算法的搜索机制形成初始粒子群,加入混沌优化的思想,并引入“交换子”概念,利用离散粒子群算法求解 TSP 问题,提出了一种新的解决 TSP 问题的光学混沌粒子群算法.

1 混合粒子群优化算法的构建

1.1 旅行商问题以及标准粒子群算法

TSP 的描述十分简单,即寻找一条最短的遍历 N 个城市的路径,其数学描述如下:

设有 N 个城市的集合 $C = \{c_1, c_2, \dots, c_N\}$, 每 2 个城市之间的距离为 $d(c_i, c_j) \in \mathbf{R}^+$, 其中 $c_i, c_j \in C$ ($1 \leq i, j \leq N$), 求使目标函数

$$T_d = \sum_{i=1}^{N-1} (c_{\Pi(i)}, c_{\Pi(i+1)}) + d(c_{\Pi(N)}, c_{\Pi(1)})$$

达到最小的城市序列 $(c_{\Pi(1)}, c_{\Pi(2)}, \dots, c_{\Pi(N)})$, 其中, $\Pi(1), \Pi(2), \dots, \Pi(N)$ 是 $1, 2, \dots, N$ 的全排列.

1998 年, Shi 等^[9]给出了标准的 PSO 算法的数学描述: 设搜索空间为 D 维空间, 粒子数为 n , 第 i

个粒子的位置用 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 表示; 第 i 个粒子的速度变化率用 $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 表示; 第 i 个粒子迄今为止搜索的得最好位置为 $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$, 记为 p_{best} , 整个粒子群迄今为止搜索到的最好位置为 $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$, 记为 g_{best} , 对于每一次迭代, 第 i 个粒子在第 D 维运动的表达式如下:

$$v_{id}(t+1) = wv_{id}(t) + c_2 \text{rand}() \cdot [p_{gd}(t) - x_{id}(t)] + c_1 \text{rand}() \cdot [p_{id}(t) - x_{id}(t)],$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1),$$

$$1 \leq i \leq n, 1 \leq d \leq D.$$

式中: c_1, c_2 为正常数, 称为加速因子; $\text{rand}()$ 为 $[0, 1]$ 之间的随机数; w 称为惯性因子. 第 d 维的位置和速度的变化范围为 $[-x_{d_{\max}}, x_{d_{\max}}]$ 和 $[-v_{d_{\max}}, v_{d_{\max}}]$, 如果在某一维中迭代的 x_{id}, v_{id} 超过了取值边界则按照边界取值.

1.2 光学寻优算法

光学寻优算法借鉴了费马定理, 利用光在传播过程中自动寻优的机制, 将光的折射与反射原理与最优化的寻优过程联系起来, 给出一种新的最优化搜索算法. 这种算法将坐标空间设想为填充了具有不同折射率的介质的空间, 如图 1 所示, 将搜索路径设想为光的传播路径, 通过光的折射原理, 使搜索方向趋向于目标函数值减小的方向, 通过光的反射原理, 改变搜索方向, 使得搜索在折射无法进行时继续下去.

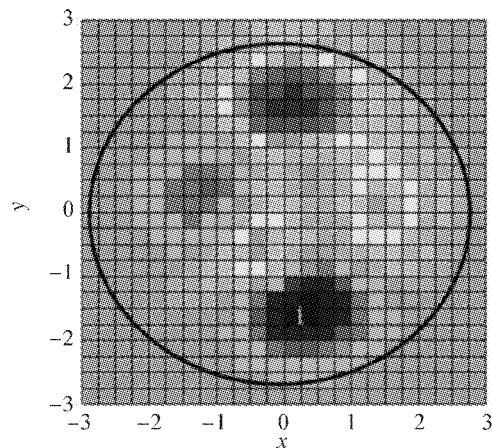


图1 在可行域中填充介质

Fig.1 Filling medium in feasible region

针对以下问题进行研究:

$$\min f(X), X \in E \subset \mathbf{R} \times \mathbf{R}.$$

式中: $f(X)$ 是正函数, 即 $\forall (x, y) \in M, f(x, y) > 0$. X 是可行解, M 是 $f(X)$ 的可行域, $\mathbf{R} \times \mathbf{R}$ 是二维实数空间. $P^{(i)}$ 为第 i 次的搜索方向, h, τ 为网格步长. 对于第 i 次迭代, 令搜索以 $P^{(i)}$ 方向在矩形分块 D_i 中搜索到点 $X^{(i)} = (x_i, y_i)$, 并在 $X^{(i)}$ 点改变搜索方向, 到达 $X^{(i+1)}$ 点, 方向的迭代关系满足:

1) 若 $\frac{v_{i+1}}{v_i} \sin \alpha_i \leq 1$, 发生折射, 如图 2(a) 所示.

$$\frac{\sin \alpha_i}{\sin \alpha_{i+1}} = \frac{v_i}{v_{i+1}}.$$

式中: α_i 为 D_i 中的入射角, α_{i+1} 为 D_{i+1} 中的折射角, v_i 为光在 D_i 中的传播速度, 可设定为 D_i 中 $X^{(i+1)} = (x_{i+1}, y_{i+1})$ 点的函数值. v_{i+1} 为光在 D_{i+1} 中的传播速度, 可设定为 D_{i+1} 中 $X^{(i+1)} = (x_{i+1}, y_{i+1})$ 点的函数值.

2) 若 $\frac{v_{i+1}}{v_i} \sin \alpha_i > 1$, 发生反射, 如图 2(b) 所示.

$$\alpha_i = \alpha_{i+1}.$$

式中: α_i 为 D_i 中的入射角, α_{i+1} 为 D_{i+1} 中的反射角. 图 3~4 是加入了多个临界面的水平以及竖直方向的搜索路径. 光学寻优算法能快速找到最优搜索路径.

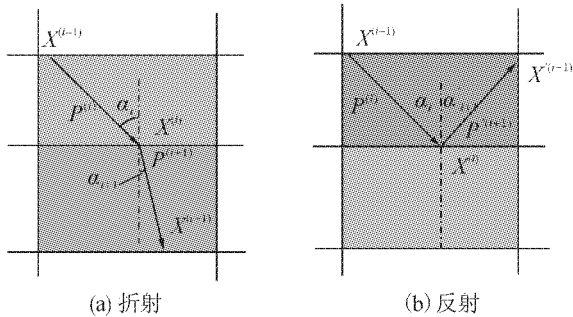


图 2 基于反射和折射搜索方向的更新

Fig. 2 Updating searching direction based on reflection and refraction

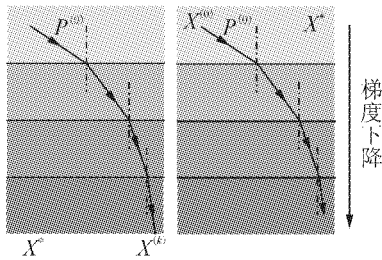


图 3 增加多个临界面光路的更新

Fig. 3 Updating light line after adding

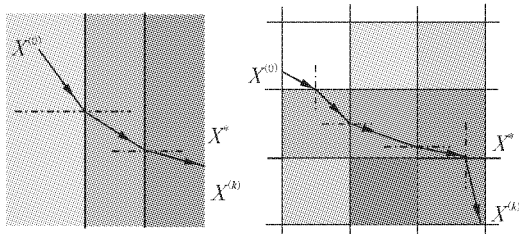


图 4 增加水平竖直临界面

Fig. 4 Adding horizontal and many critical surfaces vertical critical surfaces

1.3 TSP 问题中的光学寻优思想

光学寻优算法遵循费马原理, 费马原理表明, 光在介质中从一点向另一点传播时, 总是沿着时间最

少的路径, 光的传播在不同介质中速度不同, 运用光学寻优的思想可以很容易地得到粒子群的一组性能良好的初始粒子, 大大减少算法的迭代次数.

定义 1 当前城市密度. 从当前城市到其他未被遍历过的城市的最短路径.

定义 2 前沿城市密度. 去除已经遍历的城市, 剩余城市路径的平均值.

定义 3 TSP 问题中的反射. 从当前城市开始, 随机选择一条与未被遍历城市的路径.

定义 4 TSP 问题中的折射. 从当前城市开始, 选择与未被遍历城市之间最短的路径.

光学寻优初始化初始值的过程如下:

1) 随机选择一个城市作为出发点, 选择与当前城市之间的最短路径城市作为下一个遍历点.

2) 计算当前城市密度与前沿城市密度.

3) 如果当前城市密度小于前沿城市密度, 则发生折射, 选择与未被遍历城市之间最短路径城市作为下一个遍历点; 如果当前城市密度大于前沿城市密度, 随机选择一个未被遍历的城市作为下一个遍历城市.

4) 重复 3) 的过程直到所有的城市都被遍历.

5) 将每一个城市都作为起点, 依据光学寻优的思想形成 N 个城市序列, 作为混沌粒子群算法的初值.

2 光学混沌粒子群算法

2.1 TSP 问题中的混沌粒子群算法

2.1.1 TSP 问题中混沌粒子群算法的相关定义

定义 5 交换子. 2 个城市序列^[10] $X_i = [x_{i_1} x_{i_2} \cdots x_{i_m}]$ 与 $X_j = [x_{j_1} x_{j_2} \cdots x_{j_m}]$, 如果 2 个序列在相同的位置, 数值不相同, 即 $x_{i_a} \neq x_{j_a}$, 称 (x_{i_a}, x_{j_a}) 为城市序列的交换子, 即为 $V_{ij}(x_{i_a}, x_{j_a})$.

定义 6 交换序列. 由交换子组成的序列 $V = [V_1 V_2 \cdots V_n]$, 其中 n 为 2 个城市对应序列相同, 但数值不同的位置个数.

定义 7 粒子的位置. 粒子的位置是由城市序列 $X = [X_1 X_2 \cdots X_m]$ 表示, m 为城市的个数;

粒子的速度. 粒子的速度 $V = [V_{1_a} V_{2_b} \cdots V_{m_n}]$, 其中 V_{m_n} 表示交换子, 速度为交换序列.

2.1.2 交换子与交换序列的运算法则

1) 位置与交换子的加法.

位置与速度的加法形成新的城市序列: 设 $X = [X_1 X_2 \cdots X_m]$ 为城市序列, $V_{ij}(X_i, X_j)$ 为交换, 则

$$X = [X_1 X_2 \cdots X_i X_j X_m] + V_{ij}(X_i, X_j) = [X_1 X_2 \cdots X_j X_i X_m]. \quad (1)$$

$X = [X_1 X_2 \cdots X_j X_i X_m]$ 为新形成的城市序列.

例1:

$$X_1[4\ 3\ 6\ 2\ 1\ 5] + v(1,2) = X_2[3\ 4\ 6\ 2\ 1\ 5].$$

2)位置与位置的减法.

位置与位置的减法形成交换序列即生成新速度: $V_{ij} = X_i - X_j$, 其中 X_i, X_j 为城市序号. 先找到与第1个城市序列中第1个元素相同的第2个城市序列位置, 形成交换子 $v(1, i)$, 然后将此交换子作用在第1个序列上得到新的第1个序列, 再找到新的第1个城市序列与第2个城市序列数值相同的第1个位置, 形成交换子 $v(2, i)$, 依次进行下去, 得到2个城市序列的交换序列.

例2:

$$X_1[4\ 3\ 6\ 2\ 1\ 5] - X_2[5\ 6\ 2\ 1\ 4\ 3] = V(v(1,5) + v(2,6) + v(4,6) + v(5,6)).$$

3)交换子的数乘.

速度的数乘具有概率意义, 例如 $V_{ia} = c \cdot V_{ja}$, 其中 $c \in [0, 1]$ 是一个常数, 在计算 V_{ia} 时, 对 V_{ja} 中的每一维速度 V_{ja} 生成一个 $(0, 1)$ 的随机数.

$$V_{ia} = \begin{cases} V_{ja}, & \text{rand} < c; \\ 0, & \text{其他.} \end{cases} \quad (2)$$

4)混沌思想.

通常一类非常简单却又广泛应用的混沌系统是 Logistic 映, 其定义如下^[11]:

$$z_{k+1} = uz_k(1 - z_k), z_k \in (0, 1). \quad (3)$$

式中: z_k 为实值序列, u 为参数, 研究表明当 $3.571\ 448 \leq u \leq 4$ 时, 该混沌映射处于混沌状态, 把 $3.571\ 448 \leq u \leq 4$ 称为混沌区域. Logistic 混沌映射所生成的序列具有如下混沌特性: ①非周期的序列; ②该混沌序列不收敛; ③ z_k 可以遍历整个 $(0, 1)$ 区域.

本文取 $u = 4$, 设城市数量为 n , 按照顺序随机地生成 $(0, 1)$ 的 n 个随机数, 构成向量序列 $Z(k) = [z_1(k) \ z_2(k) \ \cdots \ z_n(k)]$, 将 z_1, z_2, \dots, z_n 按照从小到大排列 $1, 2, \dots, n$ 的下标号构成城市序列 $X_i = [X_{i1} \ X_{i2} \ \cdots \ X_{in}]$, 按照式(1)生成向量, 然后对 $Z(k+1)$ 中元素从小到大进行排列下标号 $1, 2, \dots, n$ 构成新的城市序列 $X_{(i+1)} = [X_{(i+1)1} \ X_{(i+1)2} \ \cdots \ X_{(i+1)n}]$. 混沌遍历的引入, 有效地增加了城市序列的多样性.

例3:

$$Z(k) =$$

$$[0.278\ 5\ 0.546\ 9\ 0.957\ 5\ 0.964\ 9\ 0.157\ 6\ 0.970\ 6].$$

生成的城市序列为 $X_i = [4\ 1\ 2\ 3\ 4\ 6]$, 运用式(3),

$$Z(k+1) =$$

$$[0.803\ 7\ 0.991\ 2\ 0.162\ 7\ 0.135\ 5\ 0.531\ 1\ 0.114\ 2].$$

生成新的城市序列为 $X_{i+1} = [6\ 4\ 3\ 5\ 1\ 2]$.

针对 TSP 问题, 结合混沌思想的粒子群的更新

公式变为:

$$V_i(k+1) = wV_i(k) + \vartheta_1(P_{best_i} - X_i(k)) + \vartheta_2(G_{best_i} - X_i(k)),$$

$$X(k+1) = f(x_i(k)) + V_i(k+1). \quad (4)$$

式中: ϑ_1, ϑ_2 为 $(0, 1)$ 的数, $f(x)$ 为 $x_i(k)$ 向量从小到大排列下标形成的向量函数, $X_i(k)$ 为当前城市序列, $X_i(k+1)$ 为新生成的城市序列, P_{best_i} 为当前个体最好序列, G_{best_i} 为全局最好序列. $P_{best_i} - X_i(k) = V_{p(k)}$, $G_{best_i} - X_i(k) = V_{G(k)}$ 表示为交换序列, $\vartheta_1(P_{best_i} - X_i(k))$ 表示当概率小于 ϑ_1 时发生交换操作, 当概率大于 ϑ_1 不进行交换操作, 城市序列保持不变.

2.2 光学混沌粒子群算法步骤

1)利用 1.3 中光学寻优的思想初始化粒子群, 从每一个城市出发, 得到 N 个初始值良好的城市序列; 随机生成混沌序列 $Z(1)$ 并运用函数 $f(x)$ 得到城市序列 $X_{f(1)}$.

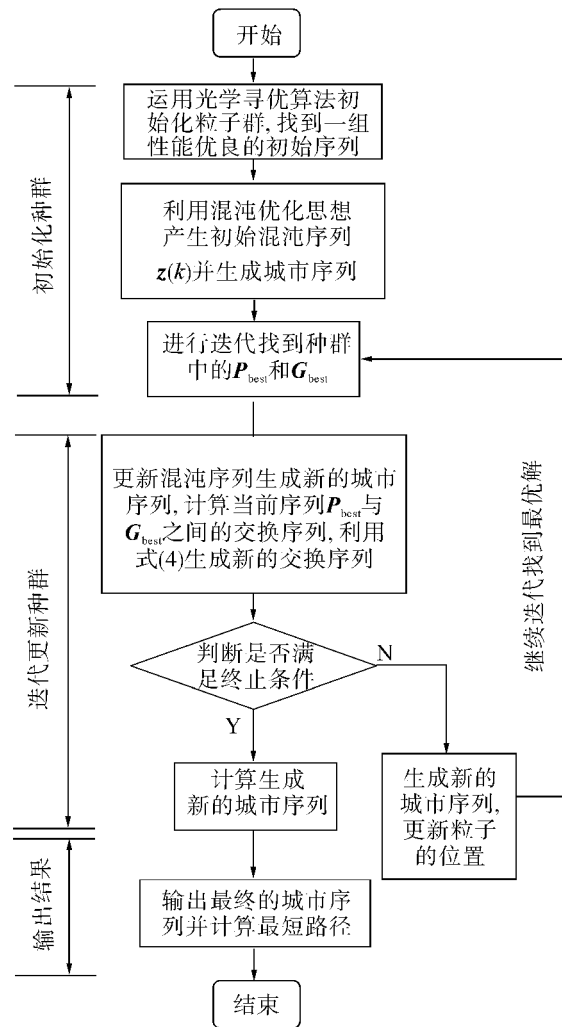


图5 混合粒子群算法的流程

Fig.5 A flow sheet of mixed particle swarm algorithm

2) 如果满足最优条件, 最短城市距离不再

变——或者达到最大迭代次数转到5),如不满足条件对初始种群进行更新,找到个体最好位置 P_{best_i} 以及全局最好位置 G_{best_i} .

3)根据式(4)更新城市序列;

①根据式(1)和(3)计算混沌生成序列 $Z(k)$ 并运用函数 $f(x)$ 得到新的城市序列 $X_{f(k)}$;

②运用例2的思想计算交换序列, $P_{best_i} - X_i(k) = V_{p(k)}, G_{best_i} - X_i(k) = V_{G(k)}$;

③根据式(2)计算 $\varphi_1(P_{best_i} - X_i(k)) + \varphi_2(G_{best_i} - X_i(k))$; φ_1, φ_2 为执行操作的控制概率.

④根据以上3个结果结合式(4)得到新的城市序列 $X_i(k+1)$.

4)迭代次数增加,更新粒子的位置转到2).

5)输出最优城市序列,并输出最短距离.

2.3 混合算法时间复杂度分析

算法的时间复杂度是对算法运行时间的度量,用来表示算法的计算效率的高低.算法的时间复杂度的大小在一定程度上反映了算法性能的优劣.忽略硬件及环境因素,假设每次执行时硬件条件和环境条件是完全一致的.设城市数量为 n ,运行迭代次数为 m ,则光学混沌粒子群算法的时间复杂度量级为 $O(m(2n^2 + 1))$.

以下来说明该算法的时间复杂度数量级为 $O(m(2n^2 + 1))$.根据混合算法的特点首先应用光学寻优算法初始化城市序列,并应用混沌方法生成新的城市序列,然后应用粒子群算法的核心思想不断地更新城市序列直到满足条件.

1)以一次迭代为例,城市数量为 n ,随机选取城市作为城市序列初始点,并应用光学寻优算法进行初始化,需要 $n(n-1)$ 次运算,所以时间复杂度为 $O(n^2 - n)$.

2)应用混沌思想更新城市序列,对 N 个序列都进行更新,时间复杂度为 $O(n)$.

3) n 个序列用来计算适应度函数的时间复杂度为 $O(n)$,在此基础上进一步确定个体极值以及群体极值,计算交换子序列从而对每一个城市序列进行更新,此步骤一共运行的时间复杂度为 $O(n^2)$.

4)在一次迭代完成之后判断是否达到终止条件,操作的时间复杂度为 $O(1)$.

通过以上的分析可以得出1)~4)的时间复杂度为 $O(2n^2 + 1)$,假设整体算法的迭代次数为 m ,则混合算法的整体运行时间为 $O(m(2n^2 + 1))$.因此整体算法的时间复杂度与城市的序列以及算法的运行代数有关,如何在保证精确度的前提下减少迭代次数也就成了控制算法运行时间的关键因素.

2.4 光学混沌粒子群算法收敛性分析

$$V_i(K+1) = wV_i(k) + \vartheta_1(P_{best_i} - X_i(k)) + \vartheta_2(G_{best_i} - X_i(k)),$$

$$X_i(K+1) = f(x_i(k)) + V_i(k+1).$$

根据混沌序列的更新以及例3可得 $f(x_i(k))$ 是线性映射,可以写成城市序列与交换子的线性组合的形式: $f(x_i(k)) = X(k) + \beta v(k)$, $\beta \in [0, 1]$,线性系统变成如下形式:

$$V_i(k+1) = wV_i(k) + c_1(P_{best_i} - X_i(k)) + c_2(G_{best_i} - X_i(k)),$$

$$X_i(k+1) = X_i(k) + \beta v_i(k) + V_i(k+1).$$

为了方便计算与分析,首先将空间简化为一维空间,将模型转化为式(5):

$$v_i(t+1) = wv_i(t) + \vartheta_1[P_{best_i} - x_i(t)] + \vartheta_2[G_{best_i} - x_i(t)],$$

$$x_i(t+1) = x_i(t) + \beta v_i(t) + v_i(t+1). \quad (5)$$

令 $\vartheta = \vartheta_1 + \vartheta_2$, 记 $P_{best_i} = p_i(t)$, $G_{best_i} = p_g(t)$, 并对模型进行简化可得表达式:

$$v_i(t+1) = wv_i(t) - \vartheta x_i(t) + \vartheta_1 p_i(t) + \vartheta_2 p_g(t),$$

$$x_i(t+1) = (w + \beta)v_i(t) + (1 - \vartheta)x_i(t) + \vartheta_1 p_i(t) + \vartheta_2 p_g(t).$$

整理记为式(6):

$$\begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} = A \begin{bmatrix} v_i(t) \\ x_i(t) \end{bmatrix} + B \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix}. \quad (6)$$

$$\text{式中: } A = \begin{bmatrix} w & -\vartheta \\ w + \beta & 1 - \vartheta \end{bmatrix}, B = \begin{bmatrix} \vartheta_1 & \vartheta_2 \\ \vartheta_1 & \vartheta_2 \end{bmatrix}.$$

通过标准粒子群算法的数学描述可以将系统变为离散线性系统,这里假定在 t 次迭代之后粒子找到最优位置时, p_{best}, g_{best} 将保持不变,式(6)中 $p_i(t), p_g(t)$ 不随着时间变化.

定理1 当 $w < 1 + \beta, \beta < \vartheta_1 + \vartheta_2 < 2w + 2 + \beta$ 时,线性定常离散系统(6)渐近稳定,并且系统收敛.

$$\lim_{t \rightarrow \infty} \begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\vartheta_1 p_i(t) + \vartheta_2 p_g(t)}{\vartheta_1 + \vartheta_2} \end{bmatrix} = \begin{bmatrix} 0 \\ p_{best} \end{bmatrix}.$$

证明 通过系统(6)以及模型(5)的表达式可将 $x_i(t)$ 消去得到如下的差分表达式:

$$v_i(t+2) + (\vartheta_1 + \vartheta_2 - w - 1)v_i(t+1) + (w - \beta)v_i(t) = 0. \quad (7)$$

对式(7)求特征方程可得

$$\lambda^2 + (\vartheta_1 + \vartheta_2 - w - 1)\lambda + w - \beta = 0.$$

为了对式(7)进行稳定性分析,用双曲线性变换将离散系统转化为线性系统,令 $\lambda = \frac{u+1}{u-1}$ 代入式(7)得:

$$\begin{aligned} \left(\frac{u+1}{u-1}\right)^2 + (\vartheta_1 + \vartheta_2 - w - 1)\frac{u+1}{u-1} + w - \beta &= 0, \\ (\vartheta_1 + \vartheta_2 - \beta)u^2 + (2 - 2w + 2\beta)u + \\ (2w + 2 - \vartheta_1 - \vartheta_2 - \beta) &= 0. \end{aligned} \quad (8)$$

根据劳斯-赫尔维茨判据^[12],很容易得到表达式:

$$\begin{cases} \vartheta_1 + \vartheta_2 - \beta > 0, \\ 1 - w + \beta > 0, \\ 2w + 2 - \vartheta_1 - \vartheta_2 - \beta > 0. \end{cases}$$

可得,当 $w < 1 + \beta, \beta < \vartheta_1 + \vartheta_2 < 2w + 2 + \beta$ 时,线性定常离散系统(6)渐进稳定.

当系统(6)稳定,

$$\begin{aligned} \begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} &= A \begin{bmatrix} v_i(t) \\ x_i(t) \end{bmatrix} + B \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix}, \\ \begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} &= A^{(t+1)} \begin{bmatrix} v_i(0) \\ x_i(0) \end{bmatrix} + \\ B(1 + A + A^2 + \dots + A^t) \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix} &= \\ A^{t+1} \begin{bmatrix} v_i(0) \\ x_i(0) \end{bmatrix} + B \left(\frac{1 - A^{(t+1)}}{1 - A} \right) \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix}. \end{aligned}$$

当参数满足条件(8), $|A| < 1$, 所以,

$$\begin{aligned} \lim_{t \rightarrow \infty} \begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} &= \lim_{t \rightarrow \infty} A^{(t+1)} \begin{bmatrix} v_i(0) \\ x_i(0) \end{bmatrix} + \\ \lim_{t \rightarrow \infty} B \left(\frac{1 - A^{(t+1)}}{1 - A} \right) \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix}, \\ \lim_{t \rightarrow \infty} \begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} &= \lim_{t \rightarrow \infty} \frac{B}{1 - A} \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix} = \\ (1 - A)^{-1} B \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix}, \\ (1 - A)^{-1} B &= \begin{bmatrix} \frac{1}{\beta} & -\frac{1}{\beta} \\ \frac{(w + \beta - 1)}{\beta\vartheta} & \frac{(1 - w)}{\beta\vartheta} \end{bmatrix}. \\ \begin{bmatrix} \vartheta_1 & \vartheta_2 \\ \vartheta_1 & \vartheta_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ \frac{\vartheta_1\beta}{\partial\beta} & \frac{\vartheta_2\beta}{\partial\beta} \end{bmatrix}, \\ \lim_{t \rightarrow \infty} \begin{bmatrix} v_i(t+1) \\ x_i(t+1) \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ \frac{\vartheta_1\beta}{\partial\beta} & \frac{\vartheta_2\beta}{\partial\beta} \end{bmatrix}. \\ \begin{bmatrix} p_i(t) \\ p_g(t) \end{bmatrix} &= \begin{bmatrix} 0 \\ \frac{\vartheta_1 p_i(t) + \vartheta_2 p_g(t)}{\vartheta_1 + \vartheta_2} \end{bmatrix}. \end{aligned}$$

有 $\lim_{t \rightarrow \infty} x_i(t) = \frac{\vartheta_1 p_i(t) + \vartheta_2 p_g(t)}{\vartheta_1 + \vartheta_2}, \lim_{t \rightarrow \infty} v_i(t) = 0$. 当迭代次数足够大必然 $p_{best_i} = G_{best_i}$, 所以 $\lim_{t \rightarrow \infty} x_i(t) =$

$G_{best_i}, \lim_{t \rightarrow \infty} v_i(t) = 0$. 无论参数 $\beta \in [0, 1]$ 如何选择, 在满足参数条件(9)的情况下, 本文提出的光学混沌算法收敛到全局最好位置, 即城市序列收敛到全局最好位置, 交换子序列收敛到空操作.

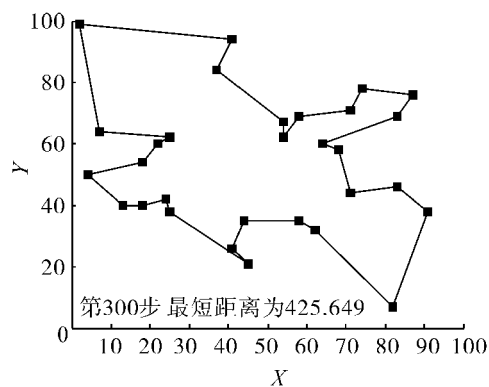
3 数值仿真

首先运用 30 个城市的标准 TSP 测试数据 Oliver30 对算法性能进行评估, 运用蚁群算法、遗传算法、模拟退火方法、禁忌搜索方法^[13]、混沌粒子群算法以及光学混沌粒子群算法, 在最大迭代次数 $M_t = 300$ 的限定下分别对测试城市进行计算, 根据定理 1 设定 $\vartheta_1 = 0.5, \vartheta_2 = 0.5, w$ 采用线性递减原则, t 为当前的迭代次数, $w = 0.6 - (t/M_t) * 0.5$, 混沌系数 $u = 4$. 每种算法运行 20 次, 对比如图 6 所示(X, Y 的散点坐标图, 取计算最好效果). 各种算法计算的最好结果、最差结果以及平均迭代次数如表 1 所示.

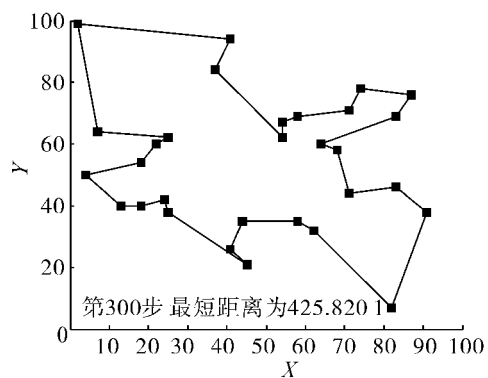
表 1 对于 Oliver30 的算法效果对比

智能优化算法	Oliver30		
	最优路径	最差路径	平均迭代次数
蚁群算法	425.649 1	430.820 1	88
遗传算法	439.359 5	472.474 6	213
模拟退火	429.380 3	435.718 2	175
禁忌搜索	492.202 5	497.079 2	195
粒子群优化算法	425.820 1	455.986 2	253
混沌粒子群	424.573 0	441.103 1	272
光学混沌粒子群	423.740 6	423.740 6	113

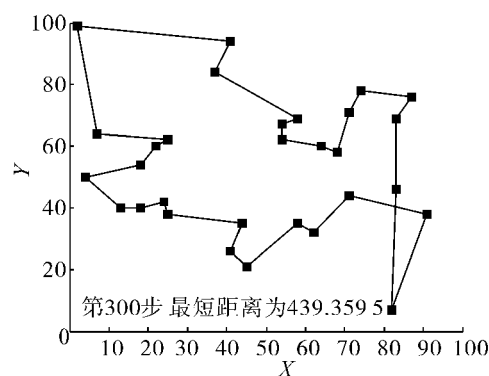
从表 1 结果中可以看出, 这 4 种经典智能算法精度有所差异, 在有限次迭代的情况下得到的最优解效果一般, 如果设置较高的迭代次数, 精度能达到满意的要求, 不过迭代次数较高, 往往容易收敛到局部最优. 本文构建的混沌粒子群算法以及加入光学原理的混沌粒子群算法具有良好的精度, 对比于基本的粒子群优化算法改进效果显著, 在最大迭代次数上限为 300 的情况下, 可以找到精确的结果, 其中光学混合粒子群算法性能更为突出, 每次运行都能找到最优值 423.740 6. 在迭代次数上, 蚁群算法具有较快的收敛速度, 但容易陷入局部最优点, 影响算法的精度. 同比与其他的 5 种算法, 光学混沌粒子群算法在收敛速度上有很大的优势. 从图 7 中可以看出光学混沌粒子群算法具有很强的收敛速度.



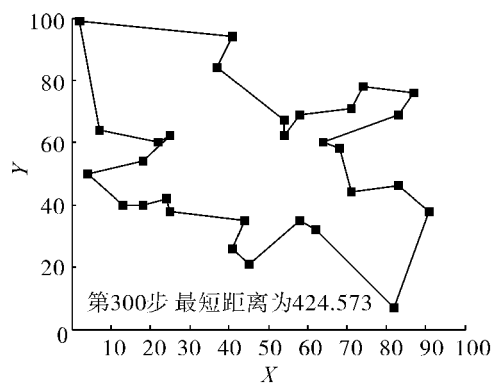
(a) 蚁群算法



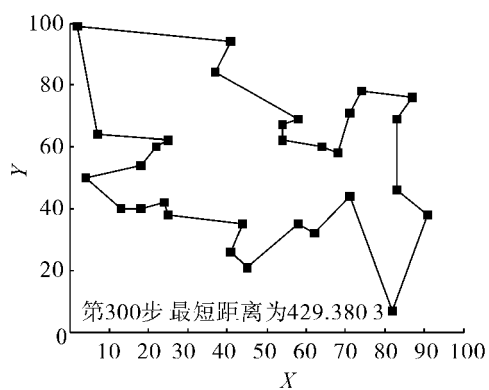
(e) 粒子群优化算法



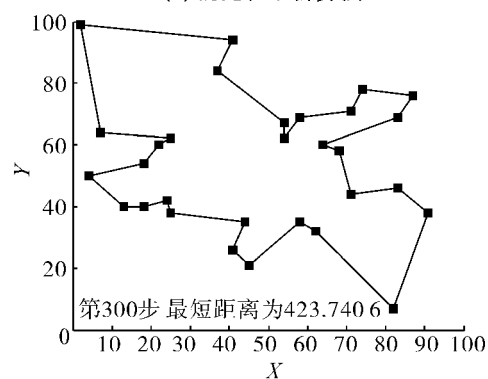
(b) 遗传算法 5



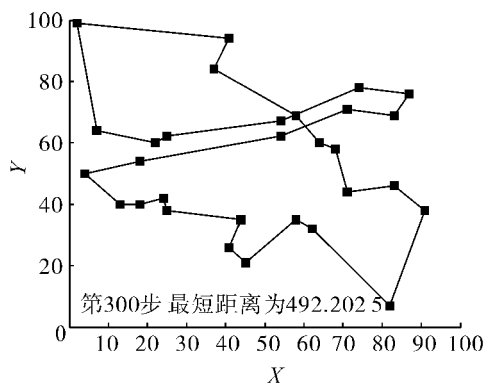
(f) 混沌粒子群算法



(c) 模拟退火算法



(g) 光学混合算法



(d) 禁忌搜索算法

图6 7种算法 Oliver30 效果对比

Fig.6 Seven contrast figures of Oliver30

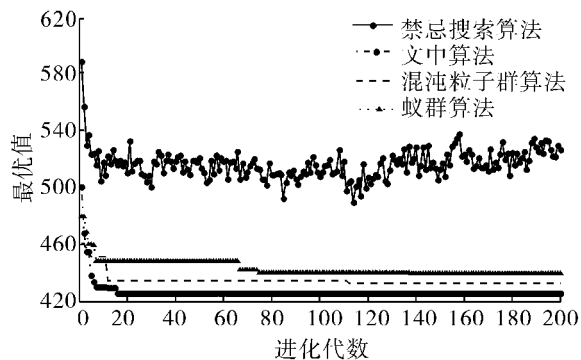


图7 Oliver30 4种算法进化曲线

Fig.7 Four evolutionary curves of Oliver30

本文提出的光学混沌粒子群算法的 GUI 界面如图 8 所示,得到的最优城市序列为(28,27,26,25,24,15,14,8,7,11,10,21,20,19,18,9,3,2,1,6,5,4,13,12,30,23,22,17,16,29),平均迭代 113 次迭代找到最优解,混沌粒子群算法得到的最好结果为 424.691 8,平均迭代次数为 272,远远大于光学混沌算法,无论是迭代时间、算法精度都要劣于加入光学寻优思想的粒子群算法.光学寻优算法初始化的本质就是要减少迭代次数,提高算法精度,原因就在于用光学寻优思想形成的初始解与最优序列之间部分区域序列是相似的,甚至是完全相同的,这样在运用粒子群算法迭代时就减少了迭代次数.混沌方法具备的全局遍历性在保证算法的精度前提下,加强了算法跳出局部最优优点的能力,增强了算法的适用性.

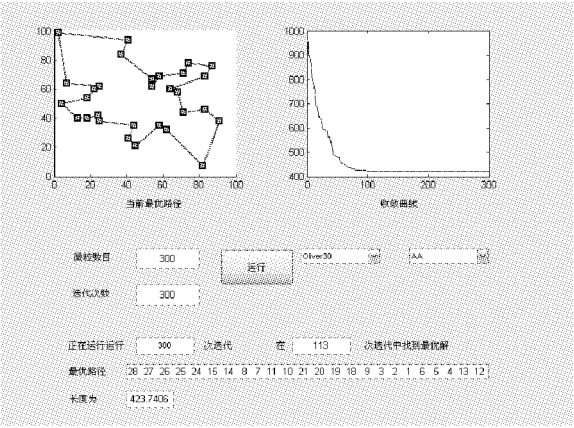


图 8 光学混合算法 GUI 效果图
Fig. 8 GUI interface

表 2 是对测试问题 eil51,参数设置与 Oliver30

实验相同,最大迭代次数 300,每种算法计算 20 次所得的数据.

表 2 对于 51 个城市问题 eil51 的算法效果对比
Table 2 Algorithm contrast effects of eil51

智能优化算法	eil51		
	最优路径	最差路径	平均迭代次数
蚁群算法	450.981 5	455.816 9	123
遗传算法	467.980 5	490.163 4	259
模拟退火	439.329 9	444.604 7	172
禁忌搜索	491.876 5	544.486 5	277
粒子群优化算法	456.533 9	487.740 8	264
混沌粒子群	434.622 2	439.006 5	289
光学混合粒子群	426.722 9	436.774 1	156

在 TSP 城市规模增大的情况下,光学混沌粒子群算法的优势得到明显体现,无论是算法精度还是算法的收敛速度都要好于其他算法.从以上结果可以看出本文提出的基于光学原理的混合粒子群算法相比于其他优化算法具有良好的效果,能成功解决中大型 TSP 路径优化问题并保持较高的精度,加入的光学寻优思想能大大节约迭代次数,保证算法在规定的迭代次数内找到最优解.

表 3 是对测试问题 CH130 的对比数据,CH130 是相对复杂的 130 个城市的 TSP 问题,最大迭代次数设定为 500,每种算法运行 20 次,误差计算是对比于 CH130 给出的精确解 6 110,应用最优路径计算得出,所有数据如表 3 所示.

表 3 对于 130 个城市问题 CH130 的算法效果对比
Table 3 Algorithm contrast effects of CH130

智能优化算法	CH130			
	最优路径	最差路径	平均迭代次数	误差/%
蚁群算法	6 417.987 5	6 430.966 5	151	5.02
遗传算法	7 118.257 2	7 322.387 8	432	16.51
模拟退火	6 350.443 1	6 471.039 4	266	3.92
禁忌搜索	10 516.179 5	12 258.249 5	354	72.11
粒子群优化算法	8 134.770 4	8 234.608 9	407	33.13
混沌粒子群	6 314.963 3	6 523.870 5	425	3.34
光学混合粒子群	6 210.422 0	6 256.519 5	395	1.64

从结果可以看出,对于较大规模的 TSP 问题光学混合算法效果显著,误差在所有同类算法中最低,

无论是从迭代时间搜索精度,还是误差大小同比于其他算法都有很大的优势.与标准粒子群算法以及

混沌粒子群算法相比,改进效果显著.图9为最优搜索形成的TSP效果图.

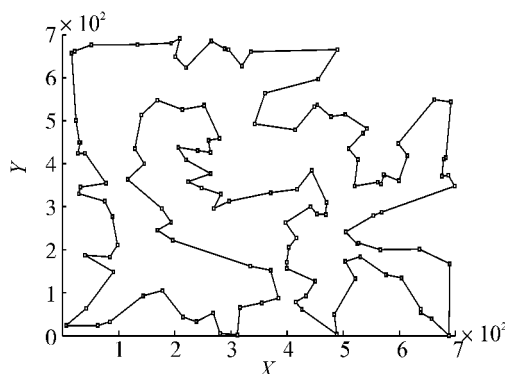


图9 应用光学混合算法求解 CH130 最短路径
6 210.422 0

Fig.9 The shortest path 6 210.422 0 of CH130 using the algorithm in this paper

4 结束语

本文提出了一种结合光学寻优算法、混沌思想的混合粒子群算法,通过光学寻优思想形成最优初值,利用加入混沌的粒子群算法成功解决了TSP问题,该算法迭代次数少、收敛速度快,对比于其他智能优化算法具有明显的优势,并用实验表明加入光学寻优思想的搜索方式大大减少了算法的迭代次数,并在一定程度上提高了算法的精度,为高效率解决大规模TSP问题提供了新的思路.

参考文献:

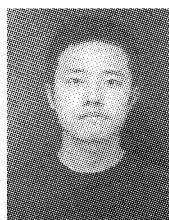
- [1] KENNEDY J, EBERHART R. A discrete binary version of the particle swarm algorithm[C]//Proceedings of the World Multiconference on Systemic, Cybernetics and Informatics. Piscataway, USA: IEEE Service Center, 1997: 4104-4109.
- [2] CLERC M. Discrete particle swarm optimization[C]//New Optimization Techniques in Engineering. Berlin: Springer-Verlag, 2004: 204-219.
- [3] 高尚, 韩斌, 吴小俊, 等. 求解旅行商问题的混合粒子群优化算法[J]. 控制与决策, 2004, 19(11): 1286-1289. GAO Shang, HAN Bin, WU Xiaojun, et al. Solving traveling salesman problem by hybrid particle swarm optimization algorithm[J]. Control and Decision, 2004, 19(11): 1286-1289.
- [4] HENDLASS T. Preserving diversity in particle swarm optimization[J]. Lecture Notes in Artificial Intelligence, 2003 (2718): 155-199.
- [5] XIE Shenli, TANG Min, DONG Jinxiang. An improved genetic algorithm for TSP problem[J]. Computer Engineering and Application, 2002, 38(8): 58-60.
- [6] SHEN Jihong, LI Yan. Light ray optimization and its parameter analysis[C]//Proceedings of the 2009 International Joint Conference on Computational Science and Optimization. Kunming, China, 2007: 918-922.
- [7] 沈继红, 李焱. 基于正六边形网格的光线寻优算法[C]//中国运筹学会第十届学术交流会论文集. 南京, 中国, 2010: 89-94. SHEN Jihong, LI Yan. Light ray optimization on hexagonal grid[C]//Proceedings of the 10th ORSC. Nanjing, China, 2010: 89-94.
- [8] SHEN Jihong, LI Jialian. The principle analysis of light ray optimization[C]//2010 Second International Conference on Computational Intelligence and Natural Computing. Wuhan, China, 2010: 154-157.
- [9] SHI Y, EBERHART R C. A modified particle swarm optimizer[C]//Proceedings of the Congress on Evolutionary Computation. Anchorage, USA, 1998: 69-73.
- [10] ZHANG Guoping, WANG Zhengou, YUAN Guolin. A chaotic search method for a class of combinatorial optimization problems[J]. Systems Engineering Theory & Practice, 2001, 21(5): 102-105.
- [11] 梁艳春, 吴春国. 群智能优化算法理论与应用[M]. 北京: 科学出版社, 2009: 17-21.
- [12] 郑大中. 线性系统理论[M]. 北京: 清华大学出版社, 2009: 213-251.
- [13] ANDRIES P E. 计算智能导论[M]. 北京: 清华大学出版社, 2010: 111-123.

作者简介:



沈继红,男,1966年生,教授,博士生导师,黑龙江省工业与应用数学学会副理事长,黑龙江省教学名师.主要研究方向为系统优化与建模.完成科研课题12项,获得省级科研和教学奖5项.

1996年获霍英东青年教师三等奖,主持的《数学建模》课程获得黑龙江省精品课程,曾获得美国数学建模竞赛一等奖8项,全国大学生数学建模竞赛一等奖1项,全国研究生数学建模竞赛一等奖1项.发表学术论文107篇.



王佩,男,1986年生,博士研究生,主要研究方向为智能优化算法以及复杂系统建模与仿真.