

两阶段动态多粒子群协作优化

刘 彬, 张仁津

(贵州师范大学 数学与计算机科学学院, 贵州 贵阳 550001)

摘 要: 为了解决粒子群优化(Particle Swarm Optimization, PSO)容易陷入到局部最优的问题,提出一种两阶段动态多粒子群协作优化算法. 算法中包含一个主粒子群和多个从粒子群,每个从粒子群都搜索部分问题域,主粒子群协调各从粒子群向最优解收敛并获得搜索到的最优解. 在第一阶段,在粒子少的问题域产生新的从粒子群,从而确保粒子比较好地覆盖问题域. 在第二阶段,删除同一子区域中位置重叠的从粒子群,减少搜索时间. 用五个测试函数与两层粒子群优化(Two-layer Particle Swarm Optimization, TLPSO)进行了比较,结果表明此算法能在高维多峰函数优化时获得更好的解.

关键词: 粒子群优化;局部最优解;全局最优解;协作;动态

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2011)10-0004-05

Two-stage Dynamic Cooperative Optimization of Multiple Particle Swarms

LIU Bin, ZHANG Ren-jin

(School of Mathematics and Computer Science, Guizhou Normal University, Guiyang 550001, China)

Abstract: To resolve the problem of trapping in a local optimum in particle swarm optimization, a two-stage dynamic cooperative optimization algorithm of multiple particle swarms was proposed. The algorithm contains a master particle swarm and multiple particle swarms. Every particle in the slave particle swarms searches a part of the problem domain. The master particle swarm coordinates every slave particle swarm converge to the optimum and obtain the optimum searched. In the first stage, the new slave particle swarms will be produced at the problem domain covered few particles, so that the particles can cover the problem domain very well. In the second stage, the slave particle swarms whose positions are overlapping will be deleted in the subregion for decreasing search time. Five benchmark functions are used to compare with two-layer particle swarm optimization. The result illustrate this algorithm can obtain better solutions in optimizing high dimensional multi-peak functions.

Key words: particle swarm optimization; local optimum; global optimum; cooperative; dynamic

1 引言

从鸟类飞翔、鱼群游动及人类的社会行为等生物群体的智能行为中受到启发, Kennedy 和 Eberhart 于 1995 年提出了粒子群优化. 虽然 PSO 提出的时间较短,但其发展非常迅速,已经获得了非常广泛的应用. 与其他基于群体的算法相比,如遗传算法、蚁群算法等,PSO 需要的群体较小,收敛速度

快. 但在高维多峰问题中,基本的 PSO 很容易陷入局部最优解. 为解决这个问题,研究人员通过各种各样的措施提高算法搜索最优解的能力,这些方法主要有如下三类方法. 第一类是对粒子实施变异操作,使粒子跳出局部解^[1]. 第二类方法是通过改变 PSO 算法的参数达到增强搜索最优解的目的^[2]. 第三类方法是把其他搜索算法与 PSO 混合在一起使用^[3-5],从而增强搜索最优解的能力.

收稿日期: 2011-04-20; **修回日期:** 2011-06-28

基金项目: 贵州省科学技术基金项目(黔科合 J 字[2010]2102 号)

文献[6]中提出一种分两阶段的 PSO,在第一阶段搜索可行解,第二阶段搜索最优解.文献[7]中提出 TLPSO,底层采用多个固定数目的粒子群,主要是为了让算法具有比单粒子群更大的搜索范围,顶层粒子群中的粒子来自于底层中的最优解.由于底层粒子群的个数是固定的,在复杂问题中还是存在着同单粒子群同样的问题,很难保证粒子覆盖到最优解所在的空间.当问题空间维数较高时,这种方法在获得的解并不令人满意.受文献[6-7]的启发,我们提出一种两段动态多粒子群协作优化(Two-stage Dynamic Cooperative Optimization of Multiple Particle Swarms, TDCOMPS),此算法的特点如下:

① 具有一个主粒子群和多个从粒子群,从粒子群的个数随搜索空间的复杂程度动态改变,确保了粒子群对解空间的覆盖,增强了算法对解空间的全局搜索能力;

② 在第一阶段,依据搜索空间中粒子群的分布情况,在粒子少的区域动态增加从粒子群,提高粒子的多样性,扩大搜索空间;

③ 在第二阶段,删除搜索空间中重叠的粒子群,提高搜索速度.

2 基本粒子群优化

粒子群优化将 d 维搜索空间中的每个解当作一个没有质量和体积的粒子,这个粒子在搜索空间中根据自身的经验及同伴的经验以一个 d 维的速度搜索全局最优解.假设粒子群体的规模为 N ,其中粒子 i ($i = 1 \sim N$) 在 t 时刻的坐标位置表示为 $x^i(t) = (x^{i1}, x^{i2}, \dots, x^{id})$,粒子的速度表示为 $v^i(t) = (v^{i1}, v^{i2}, \dots, v^{id})$,第 i 个粒子每搜索到一个位置后都通过适应度函数进行测评,第 i 个粒子迄今为止搜索到的最优位置表示为 $p^i = (p^{i1}, p^{i2}, \dots, p^{id})$,整个粒子群迄今为止搜索到的最优位置表示为 $g = (g^1, g^2, \dots, g^d)$,每个粒子的速度更新由式(1)计算出.

$$v^i(t+1) = \omega v^i(t) + c_1 r_1 (p^i - x^i(t)) + c_2 r_2 (g - x^i(t)) \quad (1)$$

式中, c_1, c_2 为正常数,用于调整群体最优和个体最优对个体的影响强度, r_1, r_2 为 $[0, 1]$ 内的随机数.参数 ω 为惯性因子,通常为了在搜索的早期提高搜索范围会用一个比较大的值,在后期为了提高局部搜索能力会取一个比较小的值.设 ω 依据迭代次数从 ω_s 线性递减至 ω_e . 设最大迭代次数为 $I_{\max}$,迭代的前代数为 I_c ,则 ω 的值可由式(2)求出.

$$\omega = \omega_s - (\omega_s - \omega_e) \frac{I_c}{I_{\max}} \quad (2)$$

粒子 i 在 $t+1$ 时刻的位置可由式(3)求出.

$$x^i(t+1) = x^i(t) + v^i(t+1) \quad (3)$$

粒子 i 求出新的解后需要与个体最优值 p^i 和群体最优值 g 进行比较,如果优于个体最优值 p^i 就需要更新 p^i ,如果优于个体最优值 g 就需要更新 g .可见,粒子就是依据自己当前的状况及最优的经验以及群体中最优的经验决定下一步的搜索方向.随着迭代的不断进行,粒子群逐渐汇集到最优位置.

3 两段动态多粒子群优化

TDCOMPS 主要解决的问题是确保扩大搜索空间,改善 PSO 早熟问题.在 TDCOMPS 中有两种粒子群,其结构如图 1 所示,包括一个主粒子群和 M 个从粒子群, M 的值在解的搜索过程中是变化的.假设每个从粒子群中有 N 个粒子,则某个粒子用 x^{ij} 表示 ($i = 1 \sim M, j = 1 \sim N$),此粒子的个体最优值表示为 p^{ij} ,从粒子群的最优值表示为 g^i .每个从粒子群的最优值都作为主粒子群中一个粒子的值,主粒子群中粒子的最优值为 q^i ,主粒子群的最优值为 g_{best} .通过主从粒子群的协作,所有粒子都朝最优解收敛.

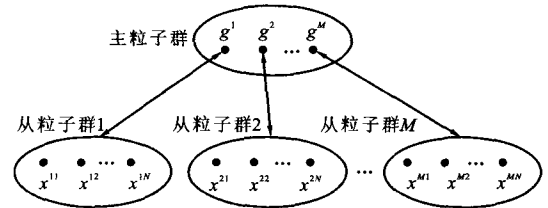


图 1 TDCOMPS 的结构

3.1 从粒子群增加策略

在 TDCOMPS 的第一阶段,为了扩大搜索空间,我们在粒子较少的空间增加新的粒子群.从基本 PSO 中可知,粒子群中的所有粒子都趋向于汇集到群体的最优值.因此在 TDCOMPS 中,我们可以这样设想,如果从粒子群中某个粒子 x^{ij} 与所有从粒子群的最优值的距离都大于 $D_{\min}$,则可以认为在此粒子的附近的粒子较少,可能有必要在此粒子附近增加一个新的粒子群.但如果仅以此条件就增加新的粒子群可能导致生成过多没有必要的粒子群.因为此粒子 x^{ij} 可能与主粒子群的最优值相差甚远,在其附近不会有最优解,故可以计算粒子 x^{ij} 与主粒子群最优值 r 的距离,如果小于 $D_{\max}$,就增加一个新的粒子群,否则就不增加.设粒子的维数为 d ,故粒子 $x^{ij} = (x_1^{ij}, x_2^{ij}, \dots, x_d^{ij})$,第 k 个从粒子群的最优值 g^k

$= (g_1^k, g_1^k, \dots, g_d^k)$, 它们之间距离按式(4) 计算.

$$\text{Dis}(x^j, g^k) = \sqrt{\sum_{l=1}^d (x_l^j - g_l^k)^2} \quad (4)$$

如果粒子 $x^j = (x_1^j, x_2^j, \dots, x_d^j)$ 需要生成新的从粒子群, 假设其分量 x_k^j 的取值范围为 $[x_k^{\min}, x_k^{\max}]$, 当前从粒子群的个数为 M , 则新从粒子群 $M+1$ 的粒子 $x^{(M+1)j} = (x_1^{(M+1)j}, x_2^{(M+1)j}, \dots, x_d^{(M+1)j})$, $x_k^{(M+1)j} (k \in [1, d])$ 通过式(5)、(6) 计算得出. $\text{rand}()$ 生成 $[0, 1]$ 之间的随机数, $\lceil \text{rand}() + 0.5 \rceil$ 是向下取整, 结果为 0 或 1.

$$z_k = x_k^j + (-1)^{\lceil \text{rand}() + 0.5 \rceil} \frac{x_k^{\max} - x_k^{\min}}{2(M+1)} \cdot \text{rand}() \quad (5)$$

$$x_k^{(M+1)j} = \begin{cases} x_k^{\min} & z_k < x_k^{\min} \\ z_k & x_k^{\min} \leq z_k < x_k^{\max} \\ x_k^{\max} & x_k^{\max} \leq z_k \end{cases} \quad (6)$$

3.2 从粒子群减少策略

在 TDCOMPS 的第一阶段已经对搜索空间进行了大范围的探索, 当第一阶段的迭代过程中不再出现新的从粒子群时即进入第二阶段, 在第二阶段主要是在已有粒子附近搜索. 在已有的从粒子群中不可避免存在着两个或两个以上的粒子群比较靠近, 这样会导致对相同区域进行了重复搜索, 故有必要减少这种重复的粒子群. 判断原则可以把两个从粒子群的最优值按式(4) 计算距离, 如果距离小于 D_{dec} 就把最优值小的从粒子群删除.

3.3 两段动态多粒子群优化过程

以下是 TDCOMPS 的详细过程:

(1) 初始化相关参数;

① 设置初始主粒子群中粒子数 $M = M_{\min}$, 从粒子群中粒子数 N , 常数 $c_1, c_2, \omega_s, \omega_e, I_{\max}$, 迭代次数 $I_c = 1$;

② 随机生成从粒子群中粒子的初始位置 $\{x^{i1}, x^{i2}, \dots, x^{iM}\}$, 粒子的位置表示为

$$x^j = (x_1^j, x_2^j, \dots, x_d^j) \quad i \in [1, M], j \in [1, N] \quad (7)$$

式中, $x_k^j (k \in [1, d])$ 按式(8) 随机生成.

$$x_k^j = x_k^{\min} + (x_k^{\max} - x_k^{\min}) \cdot \text{rand}() \quad (8)$$

③ 随机生成从粒子群中粒子的初始速度 $\{v^{i1}, v^{i2}, \dots, v^{iN}\}$, 粒子的速度表示为

$$v^j = (v_1^j, v_2^j, \dots, v_d^j) \quad i \in [1, M], j \in [1, N] \quad (9)$$

式中, $v_k^j (k \in [1, d])$ 按式(8) 随机生成.

$$v_k^j = \frac{x_k^{\max} - x_k^{\min}}{M} \cdot \text{rand}() \quad (10)$$

④ 初始从粒子群中粒子的最优值 $p^j = x^j$;

⑤ 求出从粒子群的最优值 g^i , 把 g^i 加入主粒子群中;

⑥ 求出主粒子群中的最优值 g_{best} , 随机生成主粒子群中粒子的初始速度 $u^i (i \in [1, M])$, 粒子的速度表示为

$$u^i = (u_1^i, u_2^i, \dots, u_d^i) \quad i \in [1, M] \quad (11)$$

式中, $u_k^i (k \in [1, d])$ 按式(12) 随机生成.

$$u_k^i = \frac{x_k^{\max} - x_k^{\min}}{M} \cdot \text{rand}() \quad (12)$$

(2) 设置第一阶段循环变量 $\text{cycle} = \text{false}$, 按式(13)、(14) 分别更新从粒子群中粒子的速度和位置, 其中 $\omega = \omega_s$;

$$v^j = \omega v^j + c_1 \text{rand}() (p^j - x^j) + c_2 \text{rand}() (g^i - x^j) \quad (13)$$

$$x^j = x^j + v^j \quad (14)$$

(3) 如 x^j 优于粒子最优值则更新粒子最优值 p^j , 如 x^j 优于所在群体的最优值则更新粒子群最优值 g^i ;

(4) 按式(4) 计算 x^j 与所有从粒子群最优值 p^j 的距离并按式(15) 求出最小距离 $d_{\min}$, 按式(4) 计算 x^j 与主粒子群最优值 g_{best} 的距离 l , 如果 $d_{\min} > D_{\min}$ 且 $l < D_{\max}$, 则 $\text{cycle} = \text{true}$, 按 2.1 节中的方法新增一个从粒子群, M 增加 1, 初始化新从粒子群中粒子的速度和最优值, 求出粒子群的最优值并添加到主粒子群中, 如果优于主粒子群的最优值 g_{best} 则更新 g_{best} ;

$$d_{\min} = \min_{k=1}^M (\text{Dis}(x^j, g^k)) \quad (15)$$

(5) 按式(16)、(17) 分别更新主粒子群的速度和位置, 其中 $\omega = \omega_s$, 如果粒子的新位置优于粒子最优值 q^i 则更新 q^i , 如果粒子的新位置优于主粒子群的最优值 g_{best} 则更新 g_{best} ;

$$u^i = \omega u^i + c_1 \text{rand}() (q^i - g^i) + c_2 \text{rand}() (g_{\text{best}} - g^i) \quad (16)$$

$$g^i = g^i + u^i \quad (17)$$

(6) 如果 $\text{cycle} = \text{true}$, 转到 ②;

(7) 初始最大迭代次数 G ;

(8) 按式(13)、(14) 分别更新从粒子群中粒子的速度和位置, 其中 ω 按式(2) 计算, 如 x^j 优于粒子最优值则更新粒子最优值 p^j , 如 x^j 优于所在群体的最优值则更新粒子群最优值 g^i ;

(9) 如果 $M > M_{\min}$, 按式(4) 计算两个从粒子

群最优值之间的距离,如果小于 D_{dec} 则删除最优值小的粒子群, M 减 1;

(10) 按式(16)、(17) 分别更新主粒子群的速度和位置,其中 ω 按式(2) 计算,如果粒子的新位置优于粒子最优值 q^i 则更新 q^i ,如果粒子的新位置优于主粒子群的最优值 g_{best} 则更新 g_{best} ;

(11) I_c 加 1,如果 $I_c < I_{\text{max}}$ 转到 ⑧;

(12) 返回 g_{best} .

4 测试

为了便于与文献[7] 的方法对比,本文选择相同的测试函数,因篇幅问题只选择了部分做测试,相应测试函数,变量的域及最优值如表 1 所示.

表 1 测试函数,变量的值域及全局最优值

测试函数	x_i 的值域	最优值
$f_1 = \sum_{i=1}^d [\sin(x_i) + \sin(\frac{2x_i}{3})]$	$[3, 13]$	1. 21598d(max)
$f_2 = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5. 12, 5. 12]$	0(min)
$f_3 = 418. 982\ 8d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	0(min)
$f_4 = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-5. 12, 5. 12]$	0(min)
$f_5 = \frac{1}{400} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0(min)

表 2 $d = 10$ 时 TDCOMPS 与 TLPSO 对测试函数进行 30 次独立测试所获平均结果的比较

测试函数	TDCOMPS			TLPSO		
	最好函数值	最差函数值	平均函数值	最好函数值	最差函数值	平均函数值
f_1	$2. 867\ 5 \times 10^{-11}$	$6. 732\ 8 \times 10^{-7}$	$4. 338\ 1 \times 10^{-8}$	$1. 334\ 8 \times 10^{-12}$	$4. 449\ 4 \times 10^{-9}$	$6. 564\ 6 \times 10^{-10}$
f_2	$7. 514\ 9 \times 10^{-13}$	$1. 376\ 2 \times 10^{-4}$	$3. 852\ 4 \times 10^{-6}$	$5. 826\ 2 \times 10^{-12}$	3. 981 5	0. 332 3
f_3	$1. 685\ 3 \times 10^{-2}$	20. 577 2	3. 221 5	$8. 726\ 5 \times 10^{-4}$	829. 067 5	426. 429 8
f_4	$8. 146\ 3 \times 10^{-6}$	$3. 257\ 6 \times 10^{-2}$	$2. 638\ 4 \times 10^{-3}$	0. 011 4	4. 234 8	2. 154 3
f_5	$8. 225\ 7 \times 10^{-8}$	$4. 247\ 5 \times 10^{-3}$	$5. 635\ 4 \times 10^{-5}$	0. 016 7	0. 338 4	0. 185 3

表 3 $d = 100$ 时 TDCOMPS 与 TLPSO 对测试函数进行 30 次独立测试所获平均结果的比较

测试函数	TDCOMPS			TLPSO		
	最好函数值	最差函数值	平均函数值	最好函数值	最差函数值	平均函数值
f_1	$6. 173\ 5 \times 10^{-10}$	$3. 776\ 3 \times 10^{-8}$	$1. 852\ 7 \times 10^{-9}$	$1. 323\ 4 \times 10^{-9}$	$7. 705\ 9 \times 10^{-7}$	$2. 905\ 4 \times 10^{-7}$
f_2	$3. 579\ 2 \times 10^{-11}$	$8. 264\ 3 \times 10^{-6}$	$1. 627\ 4 \times 10^{-8}$	$5. 361\ 5 \times 10^{-8}$	$6. 011\ 8 \times 10^{-4}$	$2. 157\ 6 \times 10^{-5}$
f_3	9. 857 6	107. 633 2	34. 146 7	236. 869 7	7 353. 819 1	2 828. 693 5
f_4	$1. 671\ 5 \times 10^{-3}$	$4. 158\ 3 \times 10^{-1}$	$7. 261\ 8 \times 10^{-2}$	91. 035	95. 011 7	93. 130 6
f_5	$5. 657\ 9 \times 10^{-9}$	$8. 051\ 6 \times 10^{-2}$	$4. 852\ 9 \times 10^{-4}$	$2. 584\ 4 \times 10^{-8}$	0. 024 57	$1. 716\ 1 \times 10^{-3}$

为了保证测试的公平性,我们在选择参数时尽量与它们保持相同. $c_1 = c_2 = 2, \omega_s = 0. 9, \omega_e = 0. 4, M_{\text{min}} = 20, N = 5, D_{\text{min}}, D_{\text{max}}, D_{\text{dec}}$ 分别按式(18) ~ (20) 取值.

$$D_{\text{min}} = \frac{\sum_{k=1}^d (x_k^{\text{max}} - x_k^{\text{min}})^2}{M_{\text{min}}} \quad (18)$$

$$D_{\text{max}} = \frac{\sum_{k=1}^d (x_k^{\text{max}} - x_k^{\text{min}})^2}{M_{\text{min}}/3} \quad (19)$$

$$D_{\text{dec}} = \frac{\sum_{k=1}^d (x_k^{\text{max}} - x_k^{\text{min}})^2}{20M_{\text{min}}} \quad (20)$$

我们测试了两组同文献[7] 类似的数据进行比较,分别是 $d = 10$ 和 100 的两组数据,分别如表 2、3 所示,其中 TLPSO 的测试数据来源于文献[7]. 从表中可以看出由我们所提出的方法对测试函数获得的结果要明显优于 TLPSO,获得了比较满意的结果. 但对于函数 f_3 和 f_4 ,当维数 d 比较高时,实际最优值仍有较大的差距,如果要获得更好的解可以通过调小 D_{min} 的数值,这样会在第一个阶段产生更多的从粒子群,扩大了搜索空间,但其缺点是会导致运行速度过慢.

5 结束语

文中提出了一种两阶段单主粒子群和多从粒子群协作的优化算法,在第一段中根据粒子疏密程度动态地增加从粒子群,从而起到扩大搜索范围的作用,在第二阶段时根据从粒子群的疏密程度动态减少从粒子群个数,起到减少运行时间的作用.通过与 TLPSO 比较,我们的方法在低维空间和高维空间都能获得较好的解,表明有比较好的灵活性,具有一定的优势.但由于在我们的方法增加了一个阶段,就是根据从粒子群中粒子的疏密程度增加从粒子群而扩大搜索范围,这样会导致搜索速度受到影响.在 TLPSO 中由于不能根据空间的复杂程度改变粒子群的分布,虽然采用了变异操作在一定程度上改变了粒子的分布,但在高维测试函数中粒子搜索的空间还是明显不够,导致最好的函数值也离实际最优值仍相差较远,由于 PSO 收敛速度快,所以即使增加迭代次数也难找出理想的解.我们的方法可以通过降低 $D_{\min}$ 的值改变这一问题,虽然会导致时间会长一些,但可以用于一些对于时间不是很敏感的应用中.

参考文献:

- [1] Yan Yanjun, Osadciw Lisa Ann. Density estimation using a new dimension adaptive particle swarm optimization algorithm [J]. Swarm Intelligence, 2009, 3(4): 275-301.
- [2] Yann Cooren, Maurice Clerc, Patrick Siarry. Perform-

ance evaluation of TRIBES, an adaptive particle swarm optimization algorithm [J]. Swarm Intelligence, 2009, 3(2): 149-178.

- [3] Chen MinRong, Li Xia, Zhang Xi, et al. A novel particle swarm optimizer hybridized with extremal optimization [J]. Applied Soft Computing, 2010, 11(2): 367-373.
- [4] Anghinolfi, Roberto Montemanni, Massimo Paolucci, et al. A hybrid particle swarm optimization approach for the sequential ordering problem [J]. Computers & Operations research, 2011, 38(7): 1076-1085.
- [5] Zhao S Z, Suganthan P N, Pan QuanKe, et al. Dynamic multi-swarm particle swarm optimizer with harmony search [J]. Expert Systems with Applications, 2011, 38(4): 3735-3742.
- [6] Cheng Chuntian, Liao Shenli, Wu Xinyu, et al. Two stage particle swarm optimization for long-term operation of hydroelectric power system [J]. The HongKong Institution of Engineers Transactions, 2009, 16(3): 8-14.
- [7] Chen Chichong. Two-layer particle swarm optimization for unconstrained optimization problems [J]. Applied Soft Computing, 2011, 11(1): 295-304.

作者简介:

刘彬男, (1973-), 硕士, 副教授. 研究方向为网络计算、计算机图形学.

张仁津男, (1963-), 硕士, 教授. 研究方向为软件方法学、组件技术、对象技术.

(上接第 3 页)

- [2] 郭大波. 彩色汽车图像车牌定位技术分析 [J]. 山西大学学报: 自然科学版, 2005, 28(1): 40-43.
- [3] 李炜, 黄心汉, 王敏, 等. 车牌自动识别系统中车牌分割的研究 [J]. 信号处理, 2000, 12(16): 41-44.
- [4] 刘晓薇. 基于聚类分析的车牌定位算法的研究 [J]. 通信技术, 2008, 12(41): 265-267.
- [5] Wang Cheng, Li Shaofa. An adaptive method of car plate image enhancement based on a simplified pulse coupled neural network [C] // Proc. of International Symposium on Computer Science and Computational Technology, China;

Shanghai, IEEE Computer Society, 2008.

- [6] 郭燮华. 数字图像处理原理与算法 [M]. 北京: 机械工业出版社, 2010.

作者简介:

葛二壮男, (1986-), 硕士研究生. 研究方向为图像处理、模式识别.

夏士雄 教授, 博士生导师.

闫秋艳 副教授.