

随机扩散算法求解二次背包问题

刘 勇^{1,2}, 马 良¹

(1. 上海理工大学 管理学院, 上海 200093; 2. 盐城工学院 基础教学部, 江苏 盐城 224051)

摘要: 针对二次背包问题, 提出一种新的基于群体智能的随机扩散算法. 算法采用一对一的通信机制; 利用部分函数估计评价候选解; 利用量子机制构造个体; 采用 1-OPT 和异或操作提高搜索性能. 通过数值实验并与微粒群算法、蚁群算法作比较, 结果表明算法具有较好的优化性能.

关键词: 一对一通信; 部分函数估计; 二次背包问题

中图分类号: TP181 **文献标识码:** A

Stochastic diffusion search algorithm for quadratic knapsack problem

LIU Yong^{1,2}, MA Liang¹

(1. School of Management, University of Shanghai for Science and Technology, Shanghai 200093, China;
2. Department of Fundamental Studies, Yancheng Institute of Technology, Yancheng Jiangsu 224051, China)

Abstract: To solve the quadratic knapsack problem, we propose a stochastic diffusion search algorithm which is a novel algorithm based on swarm intelligence. This algorithm adopts one-to-one communication mechanism. The candidate solutions are estimated by the partial function evaluation. Individuals are produced by quantum computation. 1-OPT and XOR operations are employed to improve the search ability. Comparison of the experiment results with those obtained from the particle swarm optimization and the ant colony optimization shows that the proposed algorithm is more effective.

Key words: one-to-one communication; partial function evaluation; quadratic knapsack problem

1 引言(Introduction)

背包问题是运筹学中的一个经典的优化问题, 有着极广泛的应用背景和经济价值, 如资金预算、项目分配、投资组合、原料切割、货物装箱等问题^[1]. 背包问题可简单描述为: 给定一个有重量限制的背包, 有一些物品要放入背包中, 每个物品都有预先设定的重量和价值, 旅行者要在这些物品中挑选部分物品放入背包中, 在不超过背包最大重量的前提下使得背包里装有的物品价值达到最大值. 从计算复杂性理论的角度来看, 背包问题是一个经典的 NP-问题. 通常所说的背包问题的目标函数都是线性的, 但大量实际问题中决策变量之间往往都存在着非线性关系, 研究非线性背包问题有着重要的实际意义. 二次背包问题就是一类典型的非线性背包问题, 是一种比求解简单背包问题更困难的组合优化难题, 甚至没有多项式算法的求解方法. 二次背包问题有着极其重要的应用价值^[2], 如设施布局、水资源分配、机器编译器设计、数字集成电路设计以及图论中的一些问题最终都转化为二次背包问题的模型进行求解.

基于二次背包问题的重要性和普遍性, 人们一直

在研究对它的有效解法. 目前的求解方法主要是经典方法, 如分支定界法、Lagrangian 松弛法等, 但 these 方法都无法精确求解. 二次背包问题、二次分配问题和度约束最小树问题等都是著名的优化难题, 计算困难是这些问题的固有性质, 目前还无有效方法求解, 但二次背包问题在实际问题中有着广泛而重要的应用, 探寻有效并快速的方法颇为重要.

群体智能算法是一种模拟自然界生物群体行为的现代启发式优化算法, 如蚁群算法、微粒群算法等, 已成功用于很多优化问题中, 并且在求解一些复杂和困难的优化问题上表现出良好的性能, 近些年来已成为一个研究热点^[3]. 以蚁群算法^[4]为例, 该算法是一种模拟蚁群利用信息素进行间接通信的优化方法, 已在旅行商问题^[5]、车辆路径问题^[6]、工件排序^[7]等方面得到广泛的应用. 但蚁群算法、微粒群算法等群体智能算法目前主要用于求解线性背包问题, 对二次背包问题研究较少. 文献[8]研究发现一种特殊的蚁群, 这种蚁群直接通过一对一的通信来寻找最短路径, 无需通过信息素进行交流. SDS 算法(stochastic diffusion search algorithm)是一种新型的基于模拟个体之间一对一的通信方式, 同时

利用部分函数估计进行优化的群体智能算法^[9,10]. SDS算法最早为求解模式识别中的刺激等效问题提出^[11], 现已成功用于文本搜索^[12]、特征追踪^[13]、机器人自定位^[14]、无线网络位置的安排^[15,16]等方面. 目前SDS算法主要用于模式识别领域, 但还未用于组合优化问题的研究. 本文利用SDS算法求解二次背包问题. 首先介绍了二次背包问题的数学模型, 然后分析了SDS算法的基本原理, 构造出一种用于求解二次背包问题的SDS算法, 最后进行大量仿真实验. 根据实验结果和与其他算法的比较, 验证了新算法的有效性.

2 二次背包问题的模型(Model of quadratic knapsack problem)

二次背包问题是线性背包问题的一种推广形式, 属于典型的NP难题, 无论是理论研究还是实际应用, 二次背包问题在优化领域中都有着极其重要的地位^[2]. 数学模型具体如下:

$$\begin{aligned} \max f(x) &= \sum_{i=1}^n p_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n p_{ij} x_i x_j, \\ \text{s.t.} \quad &\begin{cases} \sum_{i=1}^n w_i x_i - c \leq 0, \\ x_i \in \{0, 1\}, i = 1, 2, \dots, n. \end{cases} \end{aligned} \quad (1)$$

其中: p_i 和 p_{ij} 为价值系数, w_i 为权重系数, c 为背包容量.

3 基本SDS算法(Simple stochastic diffusion search algorithm)

群体智能算法成功用于求解复杂优化问题, 是基于模拟一组相互之间能够间接或直接进行通信交流的生物群体, 个体行为虽然简单但个体之间可以通过交互作用涌现出群体智能行为^[17], 群体能够完成单个个体所不能完成的任务, 如蚂蚁觅食、蜜蜂筑巢等, 体现出系统整体大于部分之和的整体突现性或涌现性原理. 现有的群体智能算法都要利用整个目标函数评价候选解的质量, 计算量大, 影响算法的求解速度. SDS算法最显著的特点是每次搜索时, 随机选择一个部分函数估计候选解的质量. 这种方法可以减少计算量, 同时当算法收敛时, 重新计算原目标函数仍能保证达到最优^[9,18].

SDS算法是受一种直接进行通信的蚁群启发的方法^[16], 和经典的基于信息素间接通信的蚁群算法不同. 在SDS算法中, 个体之间通过一对一的直接交流, 建立一种正反馈机制, 使得较好的解在群体中快速传播^[9], 引导群体向最优解方向进化; 同时算法也暗藏着负反馈机制, 即通过对失败个体和成功个体的进一步操作, 增强生成新解的随机性. 随机性的影响一方面在于接受在某种程度上退化的解, 另一方面又能保证搜索区域保持足够的大. 正反馈缩小搜

索空间, 引导群体向优化方向演化, 负反馈保持一定的寻优空间, 避免算法早熟收敛. 通过正反馈和负反馈的结合, 实现群体的自组织性, 能自我创造与更新. 另外, 算法可以避免蚁群算法中有关信息素等参数设置的问题, 使得算法更易于实现.

以最小化问题为例, 给出算法的优化过程分析. 假设目标函数 f 可以表示成 n 个部分函数 f_i 之和, 其中每个部分函数 f_i 的取值为0或1^[9,10], 即

$$\min_{\forall s \in S} f = \min_{\forall s \in S} \sum_{i=1}^n f_i, \quad f_i: S \rightarrow \{0, 1\}, \quad (2)$$

其中: s 表示一个假设解, S 表示解空间.

算法在优化过程中主要分为测试和扩散两个阶段. 在测试阶段, 每个个体对目标函数都有一个假设解 s , 随机选择一个部分函数 $f_i (i = 1, 2, \dots, n)$ 对该解进行评估, 若 $f_i(s) = 0$, 则定义该个体为成功个体; 若 $f_i(s) = 1$, 则定义该个体为失败个体. 在扩散阶段, 成功个体和失败个体直接进行交流, 建立一种正反馈机制, 使得较优解能在群体中迅速传播; 同时结合具体问题的特征, SDS算法对失败个体和成功个体采取进一步操作, 保证算法既可以对整个可行空间进行探索又能对局部区域进行精细搜索. 上述过程不断重复, 直到算法找到最优解或满意解.

大量实际应用^[9~16]表明算法有较好的优化性能, 同时很多文献对算法展开了理论研究. 文献[19]对算法全局优化性能和收敛性作了分析, 文献[20]对算法的时间复杂度进行了研究, 这些为算法的进一步的应用研究奠定了一定的理论基础.

4 求解二次背包问题的SDS算法(Stochastic diffusion search algorithm for quadratic knapsack problem)

为将SDS算法用于求解二次背包问题, 先将模型(1)进行等价变形, 令

$$g_1 = \sum_{i=1}^n p_i x_i, \quad g_2 = \sum_{i=1}^{n-1} \sum_{j=i+1}^n p_{ij} x_i x_j, \quad (3)$$

$$\begin{aligned} \max f(x) &= g_1 + g_2 = \sum_{i=1}^n p_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n p_{ij} x_i x_j, \\ \text{s.t.} \quad &\begin{cases} \sum_{i=1}^n w_i x_i - c \leq 0, \\ x_i \in \{0, 1\}, i = 1, 2, \dots, n. \end{cases} \end{aligned} \quad (4)$$

4.1 个体构造(Individual construction)

不同的个体构造方法会影响算法的性能. 本文基于量子计算中有关量子比特、叠加态等理论, 将个体定义为量子个体, 增强种群的多样性, 提高算法的优化能力^[21]. 个体视为由量子比特构成的串:

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \cdots & \beta_n \end{array} \right], \quad (5)$$

其中: n 表示个体的长度, $[\alpha_i, \beta_i]^T$ 表示第 i 个量子比特的概率幅($i = 1, 2, \dots, n$). 一个长度是 n 的个体可以表达 2^n 个状态的叠加. 叠加态特性能使单个个体表示更多的状态, 潜在地增加群体的多样性. 本文利用量子比特的概率幅 $|\beta_i|^2$ 生成算法所需的二进制解, 具体方法为: 生成一个 $[0, 1]$ 上的随机数 r , 若 $r < |\beta_i|^2$, 则相应的二进制位取1, 否则取0. 不断重复上述过程直到可行解产生.

4.2 部分函数(Partial function)

在SDS算法中, 部分函数的设计是算法的一个关键. 很多优化问题的目标函数都可以分解成多个函数之和的形式, SDS算法正是利用这些部分函数对候选解做评价. 由模型(2)可知, 算法要求部分函数 f_i 只能取0或1两个值, 如果不满足要求, 可以通过变换将函数值转化为0或1两个取值^[10]. 以二次背包问题为例, 其部分函数 g_i 的取值是不小于零的整数, 需重新定义部分函数.

令 f_i 和 g_i 对应,

$$t_i = \frac{g_i - g_i^{\min}}{g_i^{\max} - g_i^{\min}}, \quad (6)$$

如果 $t_i > u$ (u 为 $[0, 1]$ 之间是随机数), 则与 g_i 对应 f_i 的值取1; 否则与 g_i 对应 f_i 的值取0. 其中 g_i^{\max} 和 g_i^{\min} 表示 g_i 的最大值和最小值. 算法利用 t_i 实现对解的部分函数估计($i = 1, 2$).

4.3 成功个体和失败个体(Winner and loser)

因基本SDS算法是对最小化问题定义成功个体和失败个体, 而二次背包问题为最大化问题, 在定义个体类型时有所不同. 部分函数值越大, 个体被定义为成功个体的可能性越大. 具体方法如下: 个体得到候选解后, 随机选择一个部分函数 g_i ($i = 1, 2$); 若该个体的候选解使 g_i 对应的 f_i 的值取1, 则相应的个体为成功个体; 否则相应的个体为失败个体.

4.4 1-OPT和异或操作(Operations of 1-OPT and XOR)

根据二次背包问题的解只能取0或1的特点, 对失败个体和成功个体分别采用1-OPT和异或操作, 提高寻优性能. 失败个体在与成功个体进行交流后, 对采用的候选解进行1-OPT操作. 1-OPT操作是指在当前候选解的邻域内进行搜索, 邻域定义为和当前解的海明距离为1的解的集合. 在每次搜索中, 计算并保留最好的目标函数值. 成功个体采用异或操作, 随机选择一个个体, 若被选择的个体为成功个体, 则利用两者的信息进行进一步的搜索. 具体方法如下: 设 v_1 表示成功个体, v_2 表示被选中的成功个体, 令 $v'_1 = v_1 \text{ xor } (e - v_2)$, 通过比较 v_1 和 v'_1 的优劣更新成功个体的候选解. 其中: xor表示异或运算, e 表示长度为 n 的全1向量.

4.5 算法流程(Algorithm process)

求解二次背包问题的SDS算法的主要步骤可描述如下:

Step 1 初始化群体, 设置种群规模和迭代次数;

Step 2 测试阶段: 随机选择一个部分函数对每个个体进行测试, 根据测试结果将个体分为成功个体或失败个体;

Step 3 扩散阶段: 每个失败的个体随机选择一个个体进行交流, 若被选择的个体为成功个体, 则复制该成功个体的假设解并进行1-OPT操作, 否则随机选择一个可行解作为候选解; 每个成功个体随机选择一个个体, 若被选择的个体为成功个体, 则进行异或操作, 否则随机选择一个候选解;

Step 4 若当前迭代次数小于最大迭代次数, 则转Step 2, 否则转Step 5;

Step 5 输出当前最优结果.

5 数值实验(Numerical experiments)

为验证算法的有效性, 对下列问题进行测试. 本文算法采用MATLAB7.1语言编程实现, 运行在操作系统为WindowsXP、CPU为Pentium 3.4 GHz、内存为1G的计算机平台上. 本文需要设置的参数有3个, 其中概率幅 α_i 在 $[0, 1]$ 之间随机生成, $\beta_i = \sqrt{1 - \alpha_i^2}$; 群体数量为20, 两次实验的迭代次数分别为20和80. 本文将SDS算法与采用1-OPT操作的微粒群算法^[22](PSO)和蚁群算法^[1](ACO)这两种典型的群体智能算法的求解性能进行比较, 其中群体规模和迭代次数与SDS算法相同, 其它参数设置见相应文献. 每次实验独立运行20次, 记录最优解、最差解、平均解、获优次数和总运行时间等指标, 计算结果如表1所示.

问题 1

$$\begin{aligned} \max f(x) &= 2x_1 + 5x_2 + 2x_3 + 4x_4 + \\ & 8x_1x_2 + 6x_1x_3 + 10x_1x_4 + \\ & 2x_2x_3 + 6x_2x_4 + 4x_3x_4, \\ \text{s.t. } & 8x_1 + 6x_2 + 5x_3 + 3x_4 \leq 16, \\ & x_i \in \{0, 1\}, i = 1, 2, 3, 4. \end{aligned}$$

问题 2

$$\begin{aligned} \max f(x) &= \\ & 91x_1 + 78x_2 + 22x_3 + 4x_4 + \\ & 48x_5 + 85x_6 + 46x_7 + 81x_8 + \\ & 3x_9 + 26x_{10} + 55x_1x_2 + 23x_1x_3 + \\ & 35x_1x_4 + 44x_1x_5 + 5x_1x_6 + 91x_1x_7 + \\ & 95x_1x_8 + 26x_1x_9 + 40x_1x_{10} + 92x_2x_3 + \\ & 11x_2x_4 + 20x_2x_5 + 43x_2x_6 + 71x_2x_7 + \\ & 83x_2x_8 + 27x_2x_9 + 65x_2x_{10} + 7x_3x_4 + \\ & 57x_3x_5 + 33x_3x_6 + 38x_3x_7 + 57x_3x_8 + \end{aligned}$$

$$\begin{aligned}
& 63x_3x_9 + 82x_3x_{10} + 100x_4x_5 + 87x_4x_6 + \\
& 91x_4x_7 + 83x_4x_8 + 44x_4x_9 + 48x_4x_{10} + \\
& 69x_5x_6 + 57x_5x_7 + 79x_5x_8 + 89x_5x_9 + \\
& 21x_5x_{10} + 9x_6x_7 + 40x_6x_8 + 22x_6x_9 + \\
& 26x_6x_{10} + 50x_7x_8 + 6x_7x_9 + 7x_7x_{10} + \\
& 71x_8x_9 + 52x_8x_{10} + 17x_9x_{10}, \\
\text{s.t. } & 34x_1 + 33x_2 + 12x_3 + 3x_4 + 43x_5 + 26x_6 + \\
& 10x_7 + 2x_8 + 48x_9 + 39x_{10} \leq 145, \\
& x_i \in \{0, 1\}, i = 1, \dots, 10.
\end{aligned}$$

表1 3种方法的计算结果
Table 1 Results of 3 algorithms

算例	结果	算法		
		PSO	ACO	SDS
问题1	最优值	28	28	28
	平均值	28	28	28
	最差值	28	28	28
	获优次数	20/20	20/20	20/20
	总运行时间/s	3.8	32.6	1.4
问题2	最优值	1609	1609	1609
	平均值	1578.7	1578.8	1586.1
	最差值	1444	1524	1506
	获优次数	12/20	12/20	15/20
	总运行时间/s	32.9	153.4	6.9

由表1可知, SDS算法最显著的特点就是在保证优化性能的前提下所需的计算时间很少. 在第1个测试问题中, PSO和ACO这两种算法的最优值、最差值、平均值和获优次数与SDS算法相同, 但SDS算法的运行时间较少. 在第2个测试问题中, 除最差值劣于ACO算法外, 无论是最优值、平均值还是获优次数, SDS算法较其他两种算法均取得了更优的结果, 特别是算法的计算速度较快.

SDS, PSO和ACO算法都是基于群体寻优的智能算法, 但几种方法的优化原理不同. 在SDS算法中, 个体之间直接进行一对一的交流, 有利于较好的解在群体中快速传播; 同时个体只需对候选解进行部分函数估计, 计算量小, 搜索速度快. PSO源于对鸟群觅食行为的研究, 算法通过个体间的协作实现群体的最优. 在ACO算法中, 蚂蚁之间通过信息素进行间接通信, 路径越短, 信息素浓度越高, 会吸引更多的蚂蚁. PSO和ACO这两种算法的一个共同点是在优化过程中都需对整个目标函数进行考虑, 计算量较大, 相对而言计算速度较慢.

二次背包问题是一类很难求解的组合优化问题, 对其相关的数学性质研究较少. SDS算法借助于部分函数估计的方法, 将目标函数分解成两个部分, 降

低了求解难度, 提高了搜索效率. SDS算法没有任何复杂的优化机制, 所需参数较少, 易于实现, 并且具有较强的优化性能, 为二次背包问题提供了一个新的有竞争力的求解方法.

6 结论(Conclusion)

本文提出了求解二次背包问题的SDS算法, 分析了算法的原理, 给出了算法的求解步骤, 并用于二次背包问题的求解. 仿真实验结果表明, 该算法稳定性好, 优化精度高, 不易陷入局部极值, 是一种能有效求解二次背包问题的方法. 将算法用于二次分配问题是进一步的研究方向.

致谢 感谢英国伦敦大学Bishop教授提供的有关stochastic diffusion search algorithm论文资料.

参考文献(References):

- [1] 马良, 朱刚, 宁爱兵. 蚁群优化算法[M]. 北京: 科学出版社, 2008. (MA Liang, ZHU Gang, NING Aibing. *Ant Colony Optimization Algorithm*[M]. Beijing: Science Press, 2008.)
- [2] 周宇. 蚁群算法求解MKP问题的设计与实现[D]. 苏州: 苏州大学, 2007. (ZHOU Yu. *The design and implementation of ant colony algorithm for MKP*[D]. Suzhou: Soochow University, 2007.)
- [3] ZHOU Y R. Runtime analysis of ant colony optimization algorithm for TSP instances[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 1083 – 1092.
- [4] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents[J]. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 1996, 26(1): 1 – 13.
- [5] DORIGO M, GAMBARDILLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 53 – 66.
- [6] BULLNHEIMER B, HARTL R F, STRAUSS C. An improved ant system algorithm for the vehicle routing problem[J]. *Annals of Operations Research*, 1999, 89(0): 319 – 328.
- [7] BLUM C, SAMPELS M. An ant colony optimization algorithm for shop scheduling problems[J]. *Journal of Mathematical Modelling and Algorithms*, 2004, 3(3): 285 – 308.
- [8] MOGLICH M, MASCHWITZ U, HOLLOBLER B. Tandem calling: a new kind of signal in ant communication[J]. *Science*, 1974, 186(4168): 1046 – 1047.
- [9] MEYER K D, NASUTO, S J, BISHOP M. Stochastic diffusion search: partial function evaluation in swarm intelligence dynamic optimization[M] // Abraham A, Grosan C, Ramos V. *Studies in Computational Intelligence*. Berlin: Springer-Verlag, 2006, 31: 185 – 207.
- [10] MEYER K D. *Foundations of stochastic diffusion search*[D]. Reading: University of Reading of England, 2003.
- [11] BISHOP M. *Anarchic techniques for pattern classification*[D]. Reading: University of Reading of England, 1989.
- [12] BISHOP M. Stochastic searching network[C] // *Proceedings of the 1st IEEE Conference on Artificial Neural Networks*. London: IEEE, 1989: 1 – 4.
- [13] GRECH-CINI E. *Locating facial features*[D]. Reading: University of Reading of England, 1995.
- [14] BEATTIE P D, BISHOP M. Self-localization in the 'senario' autonomous wheelchair[J]. *Journal of Intelligent and Robotic Systems*, 1998, 22(3/4): 255 – 267.

- [15] HURLEY S, WHITAKER R M. An agent based approach to site selection for wireless networks[C] // *Proceedings of the 2002 ACM symposium on Applied Computing*. Madrid: Association for Computing Machinery Press, 2002: 574 – 577.
- [16] 王丽芳, 曾建潮. 随机扩散搜索法综述[J]. 模式识别与人工智能, 2008, 3(3): 351 – 356.
(WANG Lifang, ZENG Jianchao. A survey of stochastic diffusion search[J]. *Pattern Recognition and Artificial Intelligence*, 2008, 3(3): 351 – 356.)
- [17] BONABEAU E, DORIGO M, THERAULAZ G. Inspiration for optimization from social insect behavior[J]. *Nature*, 2000, 406(6791): 39 – 42.
- [18] JIN Y C. A comprehensive survey of fitness approximation in evolutionary computation[J]. *Soft Computing*, 2005, 9(1): 3 – 12.
- [19] NASUTO S J, BISHOP M. Convergence analysis of stochastic diffusion search[J]. *Journal of Parallel Algorithms and Applications*, 1999, 14(2): 89 – 107.
- [20] NASUTO S J, BISHOP M, LAURIA S. Time complexity analysis of stochastic diffusion search[C] // *Proceedings of the International ICSC/IFAC Symposium on Neural Computation*. Vienna: ICSC Academic Press, 1998: 260 – 266.
- [21] HAN K H, KIM J H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(6): 580 – 593.
- [22] KENNEDY J, EBERHART R C. A discrete binary version of the particle swarm algorithm[C] // *Proceedings of the 1997 Conference on Systems, Man, Cybernetics*. New York: IEEE, 1997: 4104 – 4108.

作者简介:

- 刘 勇 (1982—), 男, 博士研究生, 目前研究方向为智能优化、系统工程, E-mail: liuyong.seu@163.com;
- 马 良 (1964—), 男, 教授, 博士生导师, 目前研究方向为智能优化、系统工程, E-mail: maliang@usst.edu.cn.

(上接第 1139 页)

- [14] MATA M, ARMINGOL J M, FERNANEZ J, et al. Object learning and detection using evolutionary deformable models for mobile robot navigation[J]. *Robotica*, 2008, 26(1): 99 – 107.
- [15] ABE S. Introduction of support vector machines for pattern classification VI current topics[J]. *Systems, Control and Information*, 2009, 53(5): 205 – 210.
- [16] HAO P Y, CHIANG J H. Fuzzy regression analysis by support vector learning approach[J]. *IEEE Transactions on Fuzzy Systems*, 2008, 16(2): 428 – 441.
- [17] CORTES C, VAPNIK V. Support-vector networks[J]. *Machine Learning*, 1995, 20(3): 273 – 297.
- [18] PARK J H. Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots[J]. *Fuzzy Sets and Systems*, 2003, 134(1): 189 – 203.
- [19] SMITS G F, JORDAAN E M. Improved SVM regression using mixtures of kernels[C] // *Proceeding of the 2002 International Joint Conference on Neural Networks*. New York: IEEE, 2002: 2785 – 2790.

作者简介:

- 王丽杨 (1980—), 女, 博士研究生, 讲师, 研究领域为两足机器人、智能控制, E-mail: ddd0wwl@sohu.com;
- 刘 治 (1977—), 男, 博士, 教授, 研究领域为智能控制、机器人系统与技术, E-mail: lz@gdut.edu.cn;
- 赵之光 (1988—), 男, 硕士研究生, 研究领域为两足机器人、神经网络, E-mail: 546045545@qq.com;
- 章 云 (1963—), 男, 教授, 博士生导师, 研究领域为智能控制、自动化装备, E-mail: yz@gdut.edu.cn.