

# 基于智能水滴算法置换流水线调度问题的研究

周季华 叶春明 盛晓华

(上海理工大学管理学院 上海 200090)

**摘要** 生产调度问题是制造系统中最基本、最重要和最困难的问题之一。提出了一种新颖的群智能优化算法即智能水滴算法求解置换流水线问题。智能水滴算法是群智能算法领域的最新研究成果,该算法模拟了自然界水系统通过和其周围环境的相互作用而形成河流水道过程。分析了智能水滴算法的基本原理和数学模型。应用 MATLAB7.0,对 Car1—Car6 以及 Rec01 和 Rec13 问题进行了仿真测试,并将智能水滴算法和微粒群算法相比较,仿真结果表明了智能水滴算法求解生产调度问题的可行性和有效性。

**关键词** 置换流水线问题,群智能优化,智能水滴算法,基本原理,数学模型

**中图分类号** TP183 **文献标识码** A

## Research on Permutation Flow-shop Scheduling Problem by Intelligent Water Drop Algorithm

ZHOU Ji-hua YE Chun-ming SHENG Xiao-hua

(Department of Management, University of Shanghai for Science and Technology, Shanghai 200090, China)

**Abstract** Production scheduling problem is the one of the most basic, important and difficult theoretical research in a manufacturing system. This paper proposed a novel group intelligent optimization algorithm named intelligent water drop algorithm for permutation flow-shop scheduling problems(PFSP). The intelligent water drop algorithm(IWD) is based on the processes that happen in the natural river systems and the actions and reactions that take place between water drops in the river and the changes that happen in the environment in which river is flowing. This paper analyzed the basic principle and mathematical model. Car1—Car6, Rec01 and Rec13 were tested by MATLAB. Compared to PSO, the results indicate that the intelligent water drop algorithm has better feasibility and validity for solving production scheduling problem.

**Keywords** Permutation flow-shop scheduling problem, Swarm intelligence optimization, Intelligent water drop algorithm, Basic principle, Mathematical model

## 1 引言

流水车间调度模型(FSP)是许多实际流水线生产调度问题的简化模型,也是一个典型的 NP-Hard 问题,它是目前研究得最广泛的一类调度问题<sup>[1]</sup>,对其研究具有重要的理论意义和实用价值。FSP 问题的求解方法可以分为精确算法和启发式算法。启发式算法已被证明是解决困难组合优化问题最有效的启发式方法之一<sup>[2]</sup>。用于求解 FSP 的启发式算法,如蚁群算法<sup>[3]</sup>、遗传算法<sup>[4]</sup>、神经网络<sup>[5]</sup>、萤火虫算法<sup>[6]</sup>、蝙蝠算法<sup>[7]</sup>等,已被应用于解决 Flow-shop 问题。

智能水滴算法是 Shah-Hosseini<sup>[8]</sup>在 2007 年首次提出的。智能水滴算法是受到自然启发的群智能优化算法领域的最新研究成果,该算法模拟了自然界中水系统通过和其周围环境的相互作用而形成河流水道过程。目前,该算法已成功解决了一些优化问题,例如旅行商问题(TSP)<sup>[8,9]</sup>、机器人路径规划问题<sup>[10,11]</sup>、 $n$ -queen puzzle<sup>[7]</sup>、Multidimensional

Knapsack Problem (MKP)<sup>[9,12]</sup>等。智能水滴算法是一种全新的仿生智能算法,用于求解 FSP 问题的研究目前还没有展开。本文设计一种基于升序排列(ROV)的操作进行编码,实现水滴从连续到工件排序的转换,进而设计出智能水滴算法求解置换流水线问题的思路,并将结果同标准粒子群算法进行比较,对其性能进行评估。

## 2 置换 Flow-shop 问题数学模型表示

Flow-shop 调度问题是许多实际流水线生产调度问题的简化模型,可以描述为  $n$  个工件在  $m$  台机器上加工,每个工件需要  $m$  道工序,每道工序要求不同的机器,而且工件在机器上的加工时间是给定的, $n$  个工件在  $m$  台机器上加工顺序相同。调度目标是确定  $n$  个工件在每台机器上的最优加工顺序,使完工时间达到最小。

令  $t_{i,j}$  表示工件  $i$  在机器  $j$  上的加工时间,工件  $c(j, k)$  表示工件  $j$  在机器  $k$  上的加工完毕时间,  $\pi = \{j_1, j_2, \dots, j_n\}$  表

到稿日期:2012-12-01 返修日期:2013-03-26 本文受国家自然科学基金项目(71271138),教育部人文社会科学规划基金项目(10YJA630187),上海市教育委员会科研创新项目(12ZS133),上海市大文科研究生培育计划项目资助。

周季华(1986—),女,硕士生,主要研究方向为生产调度;叶春明(1964—),男,教授,主要研究方向为工业工程、智能算法;盛晓华(1986—),男,硕士生,主要研究方向为生产调度。

示工件的排序,  $k=1, 2, \dots, m$ 。  $n$  个工件、 $m$  台机器的流水车间调度问题的完工时间可表示为:

$$c(j_1, 1) = t_{j_1, 1} \quad (1)$$

$$c(j_1, k) = c(j_1, k-1) + t_{j_1, k}, k=2, \dots, m \quad (2)$$

$$c(j_i, 1) = c(j_{i-1}, 1) + t_{j_i, 1}, i=2, \dots, n \quad (3)$$

$$c(j_i, k) = \max\{c(j_{i-1}, k), c(j_i, k-1)\} + t_{j_i, k}, i=2, \dots, n; k=2, \dots, m \quad (4)$$

最大完工时间为:

$$c_{\max}(\pi) = c(j_n, m) \quad (5)$$

其中, 式(5)即为最大完成时间。

### 3 智能水滴算法基本原理

#### 3.1 自然界中的水滴

在自然界中, 我们经常看到水滴在河流、湖泊和海洋中流动。由于水滴的流动, 它们改变自身所处的环境, 而且还将持续改变下去。此外, 环境本身也在影响着河流的走向。假设河流中的水是流动的, 从较高的地势流向较低的地势, 最终流向湖泊或者海洋。在人们的一般印象当中, 自然界几乎所有的河流流动的路径都充满了曲折。而河流当中的水滴也并不是依靠视觉才最终流到湖泊或者河流中的。如果我们当作河流中的一个水滴, 我们会受到地球引力作用。众所周知, 在物理学中, 重力的方向是朝向地心的, 因此假如没有障碍物阻挡, 水滴应该沿直线向地心方向流动, 这是水滴从源头到目的地的最短距离。然而, 由于各种障碍, 水滴流经的实际路径大都与理想路径远远不同, 这也是我们现在看到的河流样貌: 弯转、曲折。然而, 水滴总是试图改变实际路径, 使其更好地接近理想路径。随着时间的推移, 通过这种不懈的努力, 水滴能够改变前行的路径。

水滴有两个特点: 一是水滴有一定的速度, 二是水滴能够携带一定量的泥土, 这样水滴就能够将路径中的泥土从一个地方搬运到另一个地方。一般地, 泥土会从水流速度较高的地方搬移到水流速度较低的地方, 由于水流速度较高的区域被搬移的泥土更多, 河道就相应变得更深, 进而可以容纳体积更多的水, 这也就吸引更多水滴流经此处。水滴所携带的泥土会在水流速度变慢时卸载下来, 并沉积在河床上。河流系统中涉及的其他机制, 在这里就不再考虑了。

总之河流中的水滴是有一定速度的, 往往携带一定量的泥土。水滴从河床上携带一些泥土, 泥土从速度高的地方搬移到速度低的地方, 并沉积在河床上。从源头到目的地的路径当中可能存在多条分支, 水滴更倾向于选择较为易行的一条。

#### 3.2 智能水滴

前一小节当中讨论了自然界中水滴拥有的一些突出的性质。基于前文中阐述的这些性质, 可以构建出人工抽象的模型, 称之为智能水滴(IWD), 它具有以下两项重要的属性:

- 1) 水滴当前携带的泥土量, 表示为  $soil(IWD)$ ;
- 2) 水滴行进时具有的速度, 表示为  $velocity(IWD)$ 。

对于每个智能水滴而言, 这两项属性的值, 即  $soil(IWD)$  和  $velocity(IWD)$  在水滴流动的时候都会发生变化。环境取决于当前的问题。在某一环境下, 从起点到达终点的路径可能不止一条, 终点的位置可能是已知的也可能未知的。一类是终点已知的情形, 问题的解决需要按照某个标准找到从起

点到达终点的最佳路径。另一类问题, 终点的位置是事先不知道的, 目标就是要花费一定的代价或者其他问题当中定义的条件找到终点最优的位置。

我们假设智能水滴是按照有限离散的步数来运动的。水滴从当前位置移动到下一位置, 智能水滴速度的增量非线性反比于两位置之间路径上的泥土量。此外, 由于两连接位置路径当中移动了部分泥土, 智能水滴中携带的泥土量也相应增加。路径当中减少的泥土量和智能水滴中增加的泥土量是相等的。智能水滴移动的时间可以由物理学中的匀速直线运动模型计算出来。因此, 智能水滴移动时间正比于两位置之间的距离, 反比于水滴的移动速度。

智能水滴的另一项行为机制是在遇到多条可选路径时, 会更倾向于选择泥土量较少的路径而不是泥土量较多的路径。为实现这一路径选择的机制, 这里引入了均匀分布的路径选择概率。对于智能水滴可能选择的多个下一个位置, 每个位置以一定的概率被选择。下一路径的概率与路径当中的泥土量成反比, 即路径当中的泥土量越少, 被选择的可能性越大。

### 4 算法流程

(1) 参数初始化。水滴个数  $N_{IWD}$  由用户自己决定; 对于速度更新参数, 我们选用  $a_v = 0.1, b_v = 1, c_v = 1$ ; 对于泥土量更新参数, 设置  $a_s = 0.1, b_s = 1, c_s = 1$ ; 最大迭代次数为  $\max Gen$ 。

(2) 随机初始化所有水滴的位置。基于 ROV 规则编码, 计算水滴的适度值  $fit^{IWD}$  以及水滴的速度  $Vel^{IWD}$ 。水滴适度值表示所求问题的完工时间, 水滴速度与水滴的适度值呈反比例关系:

$$Vel^{IWD} = k / fit^{IWD} \quad (6)$$

式中,  $k$  是一个大于 0 的常数。

(3) 比较所有水滴的适度值大小, 记录最小值以及最小值对应的排序。数组  $iter\_fit$  记录每次迭代的最小完工时间,  $iter\_record$  记录每次迭代的工件的加工次序。

(4) 随机初始化路径中的泥土量  $soil(i, j)$ , 并计算水滴泥土量。水滴泥土量:

$$soil^{IWD} = fit^{IWD} / k \quad (7)$$

根据路径中的泥土量选择下一路径。水滴位于点  $i$  选择下一点  $j$  的可能性, 通过式(8)概率公式进行计算。

$$p_i^{IWD}(j) = \frac{soil(i, j)}{\sum soil(i, k)} \quad (8)$$

(5) 更新水滴速度。对每一水滴从点  $i$  移动到下一点  $j$ , 速度更新公式如下:

$$vel^{IWD}(t+1) = vel^{IWD}(t) + \frac{a_v}{v_b + c_v \cdot soil^2(i, j)} \quad (9)$$

式中,  $vel^{IWD}(t+1)$  是水滴更新后的速度,  $soil(i, j)$  是结点  $i$  到结点  $j$  之间路径上的泥土量。  $a_v, b_v$  和  $c_v$  是速度的更新参数, 确保速度在原始速度基础上以相同幅度增加。如果速度增加的幅度过快, 水滴会陷入局部最优解; 如果速度增加幅度过慢, 水滴则需要花费更多的时间来得到最优解, 为此, 选用  $a_v = 0.1, b_v = 1, c_v = 1$ 。其中,  $b_v$  可以确保方程的分母不为零。

(6) 计算水滴从路径中带走的泥土量 ( $\Delta soil(i, j)$ )。对一水滴, 计算它从路径  $(i, j)$  中带走的泥土量计算公式如下:

$$\Delta soil(IWD) = \frac{a_s}{b_s + c_s \cdot time^2(i, j)} \quad (10)$$

式中,  $time(i, j) = 1/vel^{IWD}(t)$  是水滴以速度  $vel^{IWD}$  从结点  $i$  移动到结点  $j$  所需要的时间。 $a_s, b_s, c_s$  是参数, 保证在原有泥土量基础上以相同幅度增加。若泥土量增加幅度过大, 水滴可能会陷入局部最优; 若泥土量增加幅度过小, 水滴会需要更多的时间得到最优。其中,  $b_s$  也确保方程的分母不为零。

(7) 更新路径中泥土量  $soil(i, j)$  以及水滴泥土量。对每一水滴  $soil^{IWD}$ , 更新它经过路径  $(i, j)$  的泥土量以及水滴本身所携带的泥土量, 公式如下:

$$soil(i, j) = (1 - \rho) \cdot soil(i, j) - \rho \cdot \Delta soil(i, j) \quad (11)$$

$$soil^{IWD} = soil^{IWD} + \Delta soil(i, j)$$

式中,  $soil^{IWD}$  表示水滴所携带的泥土量。参数  $\rho$  是一个小于 1 的正数, 这里  $\rho = 0.9$ 。

(8) 水滴变异以及水滴进化。

(9) 对每一水滴, 重复完成步骤(4)–(8), 直至达到最大迭代次数, 并输出  $iter-fit$  和  $iter-record$ 。

智能水滴算法求解 PFSP 的流程如图 1 所示。

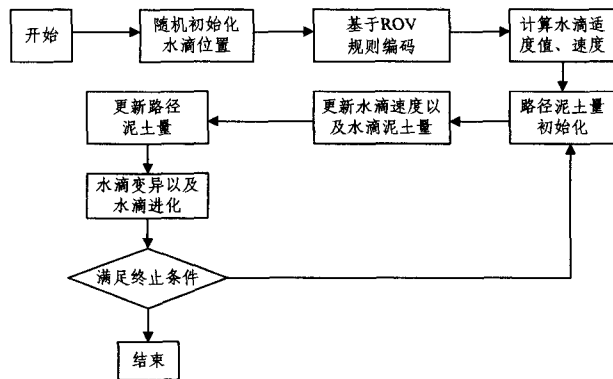


图 1 智能水滴算法求解 PFSP 流程图

## 5 仿真实验

本文采用仿真软件的测试平台为 WinXP 操作系统, 编译软件采用 MATLAB R2010a 版本。参数设置: 迭代次数  $\max Gen = 100, a_v = 0.1, b_v = 1, c_v = 1, a_s = 0.1, b_s = 1, c_s = 1, \rho = 0.9$ 。

### 5.1 函数仿真测试

测试函数:

$$(1) f_1(X) = \sum_{i=0}^{n-1} x_i^2, -5.12 \leq x_i \leq 5.12$$

$$(2) f_2(X) = \sum_{i=0}^{n-1} |x_i| + \prod_{i=0}^{n-1} |x_i|, -10 \leq x_i \leq 10$$

$$(3) f_{14} = \left( \frac{1}{500} + \sum_{j=0}^{24} (j+1 + \sum_{i=0}^j (x_i - \alpha_{ij})^6)^{-1} \right)^{-1}, -65.54 \leq x_i \leq 65.54$$

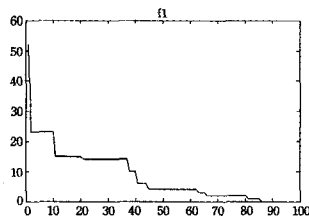


图 2 函数  $f_1$  的寻优收敛曲线

图 2 是应用智能水滴算法测试函数  $f_1$  得到的收敛曲线, 从图中可以看出该算法可以获得函数  $f_1$  最优解。对函数  $f_1, f_2, f_{14}$  独立运行 10 次, 得到的结果如表 1 所列。

表 1 IWD 算法测试函数结果

函数	最优解	运行次数	最优结果	平均结果	最差结果
$f_1$	0	10	$1.28e-17$	$6.44e-16$	$1.96e-15$
$f_2$	0	10	$2.33e-8$	$3.92e-8$	$4.17e-8$
$f_{14}$	0	10	$9.98e-1$	$9.98e-1$	$9.98e-1$

### 5.2 实例测试

本文采用经典 Car 类和 Rec 系列问题进行实验仿真测试, Car 系列、Rec 系列的测试数据分别是由 Carlier<sup>[13]</sup> 和 Reeves<sup>[14]</sup> 设计的。

图 3 分别表示 PSO、IWD 对 Car1 问题的寻优结果。两种算法都是种群个数为 50, 迭代次数为 50, 程序独立运行 10 次。图 3(a)、(b) 采用基本 PSO 和 FA 算法对 CAR1 问题进行仿真测试, 两种算法都独立运行了 10 次, 每次迭代 50 代, 从图中可以得到独立运行 10 次达到最优解 7038 的迭代次数。从平均来看, PSO 算法在 26 代左右可以达到最优解, IWD 算法在 9 代左右就可以达到最优解。图 3(c)、(d) 表明, 虽然都是独立运行 10 次, 两种算法都可以全部击中理论最优值(7038), 但从整体上可以看出 IWD 算法效率明显要高于基本 PSO 算法。

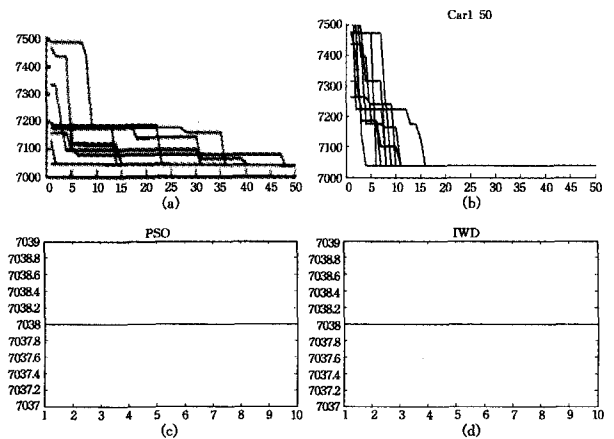


图 3 PSO 与 IWD 分别对 Car1 问题的寻优结果

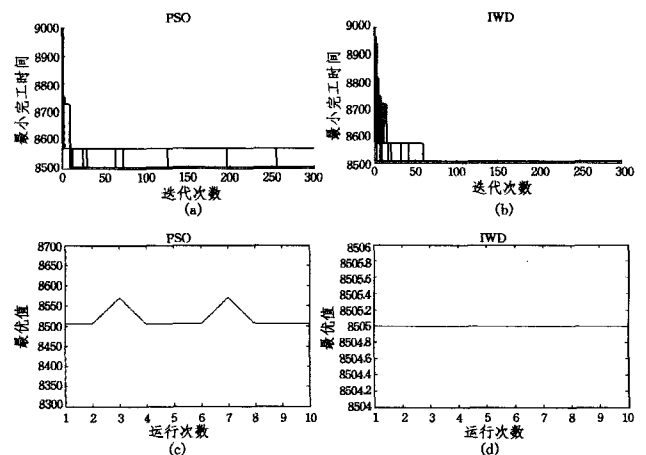


图 4 PSO 与 IWD 分别对 Car6 问题的寻优结果

图 4 表述了 PSO 算法与 IWD 算法对 Car6 问题的寻优结果, 这两种算法都是种群个数为 500, 迭代次数为 300, 程序独立运行 10 次。图 4(a)、(b) 采用基本 PSO 和 IWD 算法对 CAR6 问题进行仿真测试, 从图中可以得到 IWD 算法独立运行 10 次都能达到最优解 8505, 而 PSO 算法独立运行 10 次中有 8 次集中最优解 8505, 另外两次获得了局部最优解 8570。从平均来看, PSO 算法在 114 代左右可以达到最优解, IWD

算法在 22 代左右就可以达到最优解。结果表明,虽然都是独立运行 10 次,但 IWD 算法效率明显要高于基本 PSO 算法。

采用两种算法对每一个算例均独立运行 10 次进行测试,并对结果进行比较。

算法的参数设置如下:表 2 为标准粒子群算法(PSO)参数;表 3 为智能水滴算法(IWD)参数。

表 2 PSO 算法参数表

参数名	$\omega$	c1	c2	Num	MaxGen
参数值	1.0	2.0	2.0	30	100

表 3 IWD 算法参数表

参数名	$\rho$	$a_v$	$b_v$	$c_v$	$a_s$	$b_s$	$c_s$	N	MaxGen
参数值	0.9	0.1	1	1	0.1	1	1	30	100

表 4 中列出了两种算法对算例独立运行 10 次的仿真统计结果,其中,  $C^*$  表示算例问题的最优值, RE 表示应用算法计算的结果相对于  $C^*$  的误差百分比  $(C - C^*) / C^* \times 100\%$ , BRE 表示应用算法计算的最优结果对于  $C^*$  的误差百分比, ARE 表示应用算法计算的平均结果对于  $C^*$  的误差百分比, WRE 表示应用算法计算的最差结果对于  $C^*$  的误差百分比。

表 4 两种算法测试结果比较

问题	$C^*$	PSO			IWD		
		BRE	ARE	WRE	BRE	ARE	WRE
Car1	7038	0	1.98	9.19	0	0	0
Car2	7166	0	4.97	10.76	0	0	0
Car3	7312	0	4.38	7.76	0	1.32	2.30
Car4	8003	0	3.68	7.41	0	0	0
Car5	7720	0	2.91	6.5	0.16	0.71	1.32
Car6	8505	0	2.13	6.34	0.76	0.93	2.47
Rec01	1247	2.83	8.25	12.01	1.44	3.90	8.42
Rec13	1930	4.30	9.63	12.07	4.04	5.70	7.05

通过表 4,可以得到 8 个不同的实例,其中有 6 个(Car1—Car6)能取到已知的最优解,2 个实例(Rec01, Rec13)不能取到已知最优解,只能得到局部最优。通过与标准 PSO 算法对比可以看出,智能水滴算法到达最优解的迭代次数远远小于标准 PSO 算法,即使对于有些算例不能获得已知最优解,但误差还是比标准 PSO 算法小。由此,智能水滴算法可以很好地求解置换流水线调度问题,通过与标准 PSO 算法求解结果的对比,验证了智能水滴算法的有效性。

**结束语** 研究智能算法一直是生产调度领域的热点问

题,并取得了很多成果。本文通过对几个典型函数、经典的 Car1—Car6 问题以及 Rec01, Rec13 问题进行仿真测试,得出智能水滴算法在离散空间和连续空间优化的可行性与有效性,其具有很好的应用前景。目前,对于智能水滴算法应用到生产调度问题中的研究还很少,本文对其做了初步的研究,还存在很多问题,有待于做进一步的探讨。

## 参考文献

- [1] 叶春明,陈子皓,寇明顺.应用新型量子微粒群优化算法求解 PFSP 问题[J].技术与创新管理,2012,33(2):162-165
- [3] 黄华,肖菁,张军.改进并行蚁群算法求解置换流水线调度问题[J].计算机工程与设计,2010,31(3):582-585
- [4] 张松艳.基于遗传算法的大型 Flow-shop 生产调度[J].浙江科技学院学报,2010,22(2):102-106
- [5] 宋存利,时维国.基于启发式信息的蚁群算法在车间作业调度中的应用研究[J].科学技术与工程,2008,8(12):3359-3361
- [6] 刘长平,叶春明.一种新颖的仿生群智能优化算法:萤火虫算法[J].计算机应用研究,2011(9)
- [7] 盛晓华,叶春明.基于蝙蝠算法的 PFSP 调度干扰管理研究[J].计算机工程与应用,2012(8)
- [8] Shah-Hosseini H. Problem solving by intelligent water drops[C]//IEEE Congress on Evolutionary Computation,2007,CEC 2007. Sept. 2007:3226-3231
- [9] Shah-Hosseini H. The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm[J]. International Journal of Bio-Inspired Computation,2009,1(1/2):71-79
- [10] Duan H, Liu S, Lei X. Air robot path planning based on intelligent water drops optimization[C]//Institute of Electrical and Electronics Engineers. Hong Kong, China, 2008:1397-1401
- [11] Duan H, Liu S, Wu J. Novel intelligent water drops optimization approach to single uav smooth trajectory planning[J]. Aerospace Science and Technology,2009,13(8):442-449
- [12] Shah-Hosseini H. Optimization with the nature-inspired intelligent water drops algorithm[C]//Santos W P D, ed. Evolutionary Computation. Vienna: Tech, 2009:297-320
- [13] Carlier J. Ordonnancements a contraintes disjonctives [J]. R. A. I. R. O. Recherche Operationelle / Oper. Res., 1978,12:333-315
- [14] Reeves C R. A genetic algorithm for flowshop sequencing [J]. Comput. Oper. Res., 1995,22:5-13
- enclosing ball problem [J]. Computational Optimization and Applications, 2005,30:147-160
- [11] Pan S H, Li X S. An efficient algorithm for the smallest enclosing ball problem in high dimensions [J]. Applied Mathematics and Computation, 2006,172:49-61
- [12] Yildirim E A. Two algorithms for the minimum enclosing ball problem [J]. SIAM Journal on Optimization, 2008,19(3):1368-1391
- [13] 丛伟杰,刘红卫.求解最小闭包球问题的一种 SMO-型方法[J].西北大学学报:自然科学版,2010,40(6):965-969
- [14] Frank M, Wolfe P. An algorithm for quadratic programming [J]. Naval Research Logistics Quarterly, 1956,3(1/2):95-110
- [15] Guelat J, Marcotte P. Some comments on Wolfe's away steps [J]. Mathematical Programming, 1986,35(1):110-119
- [16] Chen P H, Fan R E, Lin C J. A study on SMO-type decomposition methods for support vector machines [J]. IEEE Transactions on Neural Networks, 2006,17:893-908

(上接第 236 页)

- [5] Badoiu M, Har-Peled S, Indyk P. Approximate clustering via core-sets [C]//Proceedings of the 34<sup>th</sup> Annual ACM Symposium on Theory of Computing. 2002:250-257
- [6] Badoiu M, Clarkson K L. Smaller core-sets for balls [C]//Proceedings of the 14<sup>th</sup> Annual ACM-SIAM Symposium on Discrete Algorithms. 2003:801-802
- [7] Badoiu M, Clarkson K L. Optimal core-sets for balls [J]. Computational Geometry: Theory and Applications, 2008,40(1):14-22
- [8] Kumar P, Mitchell J S B, Yildirim E A. Approximate minimum enclosing balls in high dimensions using core-sets [J]. The ACM Journal of Experimental Algorithmics, 2003,8(1)
- [9] Fischer K, Gartner B. The smallest enclosing ball of balls: Combinatorial structure and algorithms [J]. International Journal of Computational Geometry and Applications, 2004,14:341-378
- [10] Zhou G, Toh K C, Sun J. Efficient algorithms for the smallest