

文章编号: 1001-0920(2006)09-0969-05

半导体生产线工序参数的逻辑时序微粒群优化策略

康琦, 汪镭, 吴启迪

(同济大学电子与信息工程学院, 上海 200092)

摘要: 提出一种具有逻辑时序特征的微粒群优化算法, 并将其应用于半导体封装生产线的工序参数优化中。对实际的半导体封装生产线进行建模, 并针对生产线中机器对不同产品的加工时间最优分配这一典型的工序参数优化问题, 提出了以单位时间利润值和机器均衡度为评价指标的逻辑时序微粒群优化策略。最后进行了计算机仿真和结果分析。

关键词: 半导体封装生产线; 工序参数优化; 微粒群优化算法

中图分类号: TP18 **文献标识码:** A

Logical Time-sequenced Particle Swarm Optimization Strategy of Procedure Parameters in Semiconductor Production Line

KANG Qi, WANG Lei, WU Qi-di

(College of Electronics and Information Engineering, Tongji University, Shanghai 200092, China Correspondent: KANG Qi, Email: kangqi_kz@hotmail.com)

Abstract: A kind of particle swarm optimization method with the characteristic of logical time-sequenced is proposed and applied to procedure parameters optimization of semiconductor assembly production line. The actual semiconductor assembly production line is modeled. In allusion to such a kind of typical procedure parameters optimization as process time distribution to different products on each machine, the logical time-sequenced particle swarm optimization strategy with the evaluate index of average profit in unit time and equilibrium degree of machine is put forward and simulated.

Key words: Semiconductor assembly production line; Procedure parameters optimization; Particle swarm optimization method

1 引言

半导体制造是当今最复杂的制造过程之一, 具有高度不确定性、高度复杂性和多目标等特点, 其生产线的优化已成为半导体制造业最为关注和亟需解决的难题。目前, 一些智能优化方法, 如遗传算法、多智能体技术等已成功应用于半导体生产线的优化问题求解^[1-3]。

微粒群算法是又一类新型的启发式全局优化技术, 它具有典型的群体智能特征。该算法由 Kennedy 等^[4]于 1995 年首次提出, 其基本思想来源于对鸟群

简化社会模型的研究和行为模拟。由于具有概念简单、实现方便和收敛速度快等优点, 微粒群算法迅速得到了国际演化计算研究领域的认可^[5], 并有效应用于一些大规模复杂问题的求解, 如电力系统优化、工程设计优化和交通规划等^[6-8]。

本文以半导体封装生产线为研究对象, 从宏观层面上对生产线的工序参数优化问题——机器对不同产品加工时间的分配方案优化进行研究, 提出一种具有逻辑时序特征的微粒群优化策略, 并以单位时间利润值和机器加工均衡度作为评价指标进行了

收稿日期: 2005-07-11; 修回日期: 2005-09-13

基金项目: 国家自然科学基金重点项目(70531020); 国家自然科学基金项目(70271035, 60104004); 国家发改委 CN G I 计划基金子项目(CN G I-04-15-5A-2); 国家 973 基金子项目(2002CB 312202)。

作者简介: 康琦(1980—), 男, 河南商丘人, 博士生, 从事群体智能、生产调度等研究; 吴启迪(1947—), 女, 浙江永嘉人, 教授, 博士生导师, 从事控制理论与应用、自动化系统工程等研究。

计算机仿真

2 基本微粒群算法

微粒群算法是一种基于种群的进化计算方法。初始种群由随机产生的解组成,种群中的个体(称为微粒)在搜索空间中以一定的速度飞行,该速度可根据其自身以及同伴的飞行经验进行动态调整。

目前已开发了多种微粒群模型,最基本的模型为:设微粒群规模为 N ,微粒 $i(i=1\sim N)$ 在 D 维空间中的坐标位置可表示为 $x_i=(x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD})$,微粒 i 的速度定义为每次迭代微粒移动的距离,表示为 $v_i=(v_{i1}, v_{i2}, \dots, v_{id}, \dots, v_{iD})$ 。微粒 i 在第 $d(d=1\sim D)$ 维子空间中的飞行速度 v_{id} 和微粒的位置更新调整如下:

$$v_{id} = \omega v_{id} + c_1 \text{rand}_1(\bullet) (p_{id} - x_{id}) + c_2 \text{rand}_2(\bullet) (p_{gd} - x_{id}), \quad (1)$$

$$x_{id} = x_{id} + v_{id}. \quad (2)$$

其中: ω 为惯性权重; c_1 和 c_2 为加速常数; $\text{rand}_1(\bullet)$ 和 $\text{rand}_2(\bullet)$ 分别按照均匀分布在 $[0, 1]$ 之间产生一个随机数; p_{id} 是当前微粒的历史最佳位置记录,而 p_{gd} 是群体最佳位置记录;两者与当前微粒的位置之差分别应用于改变当前微粒向其历史最佳位置和群体最佳位置运动的增量分量,并分别运用 $\text{rand}_1(\bullet)$ 和 $\text{rand}_2(\bullet)$ 进行一定程度的随机化。

首先,在寻优空间中随机产生一个微粒群,并为每个微粒的飞行速度随机赋予初值。各微粒的 p_{best} 设为其初始位置, g_{best} 则为 p_{best} 中的最优值。算法进入迭代寻优后的每个周期中,首先评价各微粒的适应度,即将微粒所处位置代入目标函数计算得到;将适应度值进行比较之后,更新并记录各微粒的历史最优位置 p_{best} 和群体最优位置 g_{best} ;然后对各微粒进行操作,即根据式(1)和(2)更新并记录各微粒的速度和位置。对所有微粒实施上述操作之后,算法进入下一个周期,用更新后的微粒位置来计算微粒的适应度,重复以上步骤,直到找到满意的解或者达到设定的寻优周期为止。

在整个寻优过程中,各微粒受其自身和群体的历史运动状态的影响,通过不断调整其飞行速度和位置,使微粒群从整体上逐渐逼近寻优目标。

3 逻辑时序微粒群优化描述

在一些优化问题中,经常会包含若干具有一定约束关系的子问题,在运用微粒群算法进行求解的时候,子问题之间的约束增加了运算的复杂度,使优化的效率受到限制。为了解决上述问题,可将这些相互耦合的约束关系进行解耦处理,使其转化为求解一个具有逻辑时序特征的子问题序列。本文所提出

的逻辑时序微粒群优化算法就是基于这样一种思路。

在基本微粒群算法的基础上,首先对优化问题按照一定的原则进行子问题逻辑时序关系排序处理,根据子问题的数目和各子问题的规模确定微粒群的寻优空间,然后将空间划分为一定的子空间,并与待解子问题序列建立对应关系。在寻优开始后,先利用微粒群的相应子空间来求解逻辑时序关系排列靠前的子问题,并将输出作为下一优化子问题的约束条件。依此类推,按照相应的逻辑时序关系逐个对各子问题进行相应微粒子空间上的优化求解。逻辑时序微粒群算法的具体步骤如下:

Step 1: 对问题进行子问题划分,确定子问题数目,并对所有子问题进行逻辑时序关系排序;

Step 2: 确定微粒群的寻优空间维数,并按照子问题的数目进行子空间进行划分,分别与待解子问题建立对应关系;

Step 3: 对算法的基本参数进行初始化设置,并对微粒群按照逻辑时序关系依次确定初始可行子空间,并进行位置和速度的随机初始化(排序靠后的可行子空间约束受已确定的可行子空间的影响);

Step 4: 采用微粒群约束优化求解模式(微粒在相应可行子空间中的速度和位置更新等式仍采用式(1)和(2))对子问题进行优化迭代,输出优化结果,并按照排定的逻辑时序关系,逐个确定下一代周期中各微粒群子空间的可行约束条件;

Step 5: 判断是否已经对所有子问题进行了优化求解。如果是,输出优化结果,算法结束;否则,返回 Step 4 继续寻优。

从整体上来看,算法在每一个迭代周期中对所有子问题的优化求解过程是同时完成的,不存在实际的次序问题。由于对算法的具体实施依赖于具体问题的描述,在对半导体封装生产线建模之后,本文将具体描述算法的实施过程。

4 半导体封装生产线建模

半导体封装是半导体生产后道工序中的主要工序,指将通过测试的晶圆按照产品型号和功能需求加工得到独立芯片的过程^[9]。典型的封装工艺流程为:划片—装片—键合—塑封—去飞边—电镀—打印—切筋和成型—外观检查—包装。

为了简化问题的描述,可以将半导体封装工艺流程简化为5个主要工序:划片(Saw) 装片(DA) 键合(WB) 塑封(Molding) 打印(Marking)。

本文对实际的生产线进行一定的简化,并以其中3种代表产品为例,建立基本符合实际的生产线模型。文中所采用的数据均取自实际的生产数据。定

义3 种代表性产品分别为A ,B 和C, 其产品型号和产品生产任务如表 1 所示

表 1 产品定义及其生产任务

产品	产品型号	加工要求/(千件/d)
A	SD IP52	50
B	TSSOP48	25
C	QFP80	17.5

定义如下加工中心, 并进行机器配置:

- 1) Saw: 进行划片工艺的加工 配置两台机器 Saw -1 和Saw -2
- 2) DA: 进行装片以及装片检查和装片胶固化等 配置 5 台机器DA -1~ DA -5
- 3) WB: 进行键合、内部检查和内部QC 等 配置 20 台机器, 并根据加工能力的不同分为 5 组:WB-1 ~ WB-5, 各组具有相同的加工能力
- 4) Molding: 对独立芯片进行封装, 并完成切筋和成形 去飞边以及电镀等工艺 配置 3 台机器M d-1~ M d-3
- 5) M arking: 打印型号和产品号码 配置 8 台机器, 根据加工能力的不同分成 4 组: M k-1 (机器 1, 2), M k-2 (机器3~ 5), M k-3 (机器6, 7) 和M k-4 (机器 8).

以上各加工中心中的不同机器对不同的产品具有不同的加工能力, 各机器的加工能力如表 2 所示 表中数据为单台机器的加工能力, 由于与实际值存在 ± 10% 的偏差, 机器的加工能力取实际加工能力的平均值 对于WB 和M arking, 本文以机器组为单位进行建模, 将整个组看作是一台加工能力更大的机器, 其加工能力通过单台加工能力与组中机器数目的乘积来计算

表 2 各加工中心中机器的加工能力 千件/d

				千件/d					
	A	B	C		A	B	C		
DA	1	30	32	10	Saw	1	60	70	40
	2	30	32	10		2	90	80	60
	3	30	32	10	Molding	1	60	×	×
	4	30	32	10		2	×	60	×
	5	30	32	10		3	×	×	60
WB	1	5	6	3	M arking	1	30	40	20
	2	3	×	10		2	18	25	12
	3	×	5	2.8		3	25	30	18
	4	4	5.5	×		4	40	50	30
	5	5.5	×	3.2					

注: × 表示产品不能在该机器上加工

假设产品A ,B 和C 同时进入生产线, 按照给定加工工序进行加工 在上述生产线模型的基础上, 以

单位时间(1d)内各加工中心中各台机器对不同产品的加工时间分配作为所考察的工序参数, 运用逻辑时序微粒群算法对该参数进行优化配置(忽略产品的加工次序与优先级问题), 以实现生产线单位加工时间的收益(利润)最大化

5 逻辑时序微粒群优化策略与仿真

5.1 逻辑时序微粒群优化策略

将前面提出的逻辑时序微粒群算法用于半导体封装生产线模型中各加工中心的工序参数优化, 得到加工中心中机器对不同产品的加工时间最优配置方案, 且优化过程是按照工序的次序对各加工中心进行的 整体的优化思路为:

- 1) 对第一道工序所在加工中心进行逻辑时序微粒群优化, 得到 3 种产品加工时间的最优配置方案, 并得到经过第一道工序后的优化产量值;
- 2) 将得到的3 种产品的产值作为下一道工序所在加工中心的投产, 运用逻辑时序微粒群算法进行优化;
- 3) 对下一道工序实施逻辑时序微粒群优化策略 直到产品加工完成, 输出整个生产线的工序参数优化结果

根据以上思路, 设计用于各加工中心的逻辑时序微粒群优化策略: 将加工中心中每台机器上对 3 种产品的加工时间分配定义为微粒的位置坐标, 将产品加工数量和机器的加工能力等对机器加工时间分配的限制条件作为微粒的可行空间, 微粒在相应的可行空间中以单位时间内的利润值和加工中心机器的均衡度作为优化评价指标进行寻优运动, 得到最优的加工时间分配 由于机器的加工能力不同, 微粒的初始化和可行空间确定采用一定的时序策略, 即先根据总加工能力对各机器进行排序, 然后按照机器次序进行初始化, 并确定可行空间

将表 1 中的各产品加工任务作为投产输入 设计整个半导体封装生产线的优化过程的总体模型, 如图 1 所示(机器的排序通过计算得到).

Saw 为该模型的第一个加工中心, 其投产相当于整个生产线模型的总输入, 其他加工中心的投产为以上加工中心的输出, 除此之外, 各加工中心的优化过程类似 因此以 Saw 为例设计逻辑时序微粒群优化步骤如下:

Step 1: 微粒群体规模为 M . 根据待求解的问题来确定微粒群的寻优空间, 例如, 微粒的位置 X 和速度 V 的坐标维数为 $D = \text{机器数} \times \text{产品数}$, 这里, $D = 2 \times 3 = 6$ 由于考察的是机器加工时间分配问题, 因此, 微粒位置和速度的各维值均表示加工时间 (h), 并用 t_{ij} 表示机器 i 对产品 j 的加工时间, 例如,

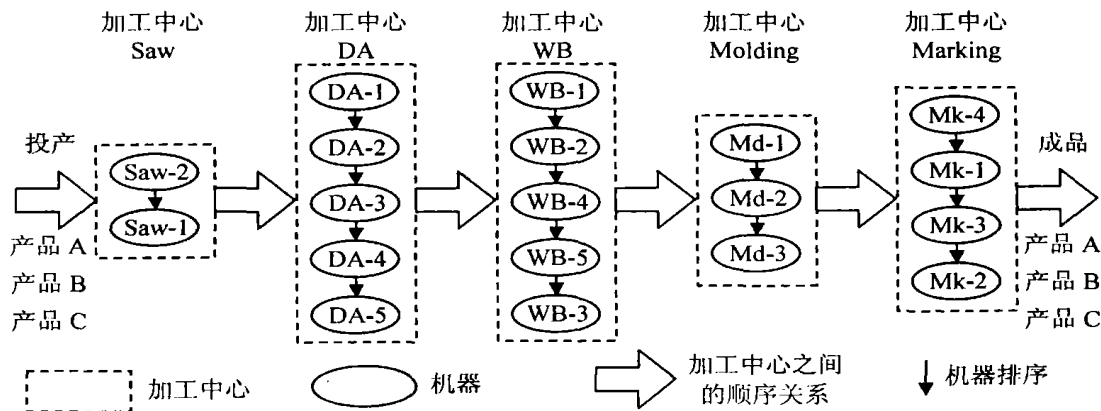


图1 半导体封装优化过程总体模型

微粒位置 $X = (x_1, x_2, \dots, x_6)$ 可以表示为 $(t_{1A}, t_{1B}, t_{1C}, t_{2A}, t_{2B}, t_{2C})$, 然后根据子问题数目(待优化的机器数目)得到划分后的子空间分别为: $((t_{1A}, t_{1B}, t_{1C}), (t_{2A}, t_{2B}, t_{2C}))$.

Step 2: 确定可行空间, 即确定每台机器上各产品可能的加工时间范围

由于机器对不同产品的加工能力存在一定差异, 因此在确定可行空间时, 可通过一定的次序对机器进行考察, 方法是在该工序内将所有机器按其总加工能力从大到小的顺序进行排队. 根据计算得到 Saw 的机器顺序为 Saw-2 Saw-1, 并确定对应的子空间次序为 $(t_{2A}, t_{2B}, t_{2C}) (t_{1A}, t_{1B}, t_{1C})$.

可行空间(微粒可行约束条件)可按如下方法进行确定: 对机器 Saw-2, 微粒位置的子空间坐标 (t_{2A}, t_{2B}, t_{2C}) 在整个空间受到的约束定义为

$$\begin{cases} t_{2A} & T_{2A}, \\ t_{2B} & T_{2B}, \\ t_{2C} & T_{2C}, \\ t_{2A} + t_{2B} + t_{2C} & 24 \end{cases} \quad (3)$$

其中 T_{2A}, T_{2B} 和 T_{2C} 表示在一定的投产条件下, 产品 A, B 和 C 在机器 Saw-2 上的最大加工时间, 通过分别将 3 种产品的投产数转化为加工时间来计算. 若 $T_{2A} > 24$, 则 $T_{2A} = 24$; 对 T_{2B} 和 T_{2C} 有同样的设置. 该约束条件表示在机器 Saw-2 上 3 种产品可能的最大加工时间, 且加工时间之和必须在一日之内, 即不超过 24h.

微粒位置的子空间坐标 (t_{2A}, t_{2B}, t_{2C}) 的约束条件影响 (t_{2A}, t_{2B}, t_{2C}) , 即机器 Saw-1 对各产品的最大加工时间还取决于机器 Saw-2 对各产品加工时间的实际分配值, 即 (t_{2A}, t_{2B}, t_{2C}) 的实际值 (t_{1A}, t_{1B}, t_{1C}) 的约束条件为

$$\begin{cases} t_{1A} & T_{1A}, \\ t_{1B} & T_{1B}, \\ t_{1C} & T_{1C}, \\ t_{1A} + t_{1B} + t_{1C} & 24 \end{cases} \quad (4)$$

其中 T_{1A}, T_{1B} 和 T_{1C} 表示在一定的投产条件下, 产品 A, B 和 C 在机器 Saw-1 上的最大加工时间. 它们的值与前一步在机器 Saw-1 上各产品的加工时间有关, 为 3 种产品的投产数对应的加工时间与机器 Saw-2 上各产品的实际加工时间的差值, 即 $T_{1A} = T_{2A} - t_{2A}, T_{1B} = T_{2B} - t_{2B}$ 和 $T_{1C} = T_{2C} - t_{2C}$. 该约束条件表示在机器 Saw-1 上 3 种产品的加工时间不能超过机器 Saw-2 加工之后剩余的可能的最大加工时间, 且加工时间之和不超过 24h.

Step 3: 微粒群初始化. 在可行空间内对各微粒的位置和速度进行随机赋值, 并对算法的基本参数进行初始化设置.

Step 4: 微粒群寻优. 进入微粒群优化过程, 经过一定次数的迭代之后, 微粒群找到一个最优值——产品 A, B 和 C 在机器上的加工时间最优分配, 满足单位时间的利润值最高且满足各机器的均衡加工要求(机器均衡度: 用加工中心各机器总加工时间两两之间的均方差平均值来衡量).

Step 5: 检查结束条件(设置最大迭代次数为 500 次). 若满足结束条件, 寻优结束并输出优化结果; 否则, 继续寻优过程.

这里的机器顺序问题是一种假想的逻辑时序关系, 在宏观上通过一定的次序来进行初步的机器规划, 实际的寻优过程不存在时间先后顺序. 由于各加工中心具有相似性, 优化过程是一致的, 不同之处是投产数有区别, 即前一道工序的实际产量作为下一道工序的投产数, 通常在机器加工能力较为有限的情况下, 后面工序的投产数将逐渐减少.

5.2 计算机仿真和结果分析

在半导体封装生产线模型和实际数据的基础上,按照上述方法和步骤对半导体封装优化过程进行计算机实现和仿真。基本参数为: $M = 60$, $v_{\max} = x_{\max}$, $k_1 = 0.4$, $k_2 = 0.9$ 。通过仿真,得到各台机器对3种产品的加工时间最优分配,并得到了生产线模型中各机器的占用率和各加工中心机器均衡度。仿真结果如表3和表4所示。

表3 各台机器对3种产品的加工时间最优分配结果

工序	产 品			机器占用率/%	
	A/h	B/h	C/h		
Saw	1	7.91	3.87	5.05	70.125
	2	8.06	4.09	3.63	65.75
DA	1	7.23	5.92	7.59	86.42
	2	7.18	6.97	5.50	81.875
	3	6.23	3.48	10.62	84.71
	4	7.55	1.51	10.30	80.67
WB	5	8.94	0.92	7.97	74.29
	1	13.96	10.04	0	100
	2	14.03	x	9.97	100
	3	x	6.01	17.99	100
	4	23.97	0.03	x	100
Molding	5	23.52	x	0.47	99.96
	1	20.00	x	x	83.33
	2	x	6.03	x	25.125
Marking	3	x	x	7.00	29.17
	1	11.72	2.17	4.90	78.29
	2	8.56	1.94	3.97	60.29
	3	9.98	1.40	4.25	65.125
	4	11.10	3.69	6.59	89.08

表4 各加工中心机器平均占用率和机器均衡度

评价指标	加工中心				
	Saw	DA	WB	Molding	Marking
各机器的平均占用率/%	67.94	81.59	99.99	45.875	58.56
各加工中心的机器均衡度	1.05	1.358	0	9.313	3.982

根据仿真结果,还得到整个生产线的机器平均占用率为77.59%,平均机器均衡度为3.406。

从仿真结果可以看出,逻辑时序微粒群算法有效地解决了半导体封装生产线中各加工中心机器对产品加工时间的配置问题,克服了以往由人工进行分配的难题,并实现了这一复杂生产线工序参数的智能优化。从总体上看,机器平均占用率较为合理,

批产品的日利润值比较理想。实验所得结果得到了厂方的认可。

前面仿真实验中的时间域为1d,如果对该时间域按比例进行一定程度的缩小后,可得到一个较短时间内的分配方案,即转化为一种连续的机器调度问题。因此,本文所讨论的工序参数优化问题对以后微粒群算法在调度优化问题中的深入研究提供了新的思路。

6 结 论

本文将所提逻辑时序微粒群优化算法应用于半导体封装生产线的工序参数——机器对不同产品的加工时间分配的优化求解中。文中对问题进行了一定的简化假设,并且所做的假设均在与相应生产部门的协商下进行,所建模型基本符合生产实际,采集的数据真实有效,所得结果得到了认可。

本文主要讨论的生产线工序参数优化问题,属于生产线调度优化的一个子集,因此对此问题的解决为以后更深入地研究调度优化问题提供了新的思路。

参考文献(References)

- [1] Shen Y X, Leachman R C. Stochastic Wafer Fabrication Scheduling [J]. *IEEE Trans on Semiconductor Manufacturing*, 2003, 16(1): 2-14
- [2] 吕文彦, 党延忠. 基于B-T规则与遗传算法的可重入生产系统调度[J]. *系统仿真学报*, 2005, 17(4): 993-996
(Lv W Y, Dang Y Z. Scheduling Re-entrant Lines Based on GA and Integrated Rules [J]. *J of System Simulation*, 2005, 17(4): 993-996)
- [3] 吴继伟, 萧蕴诗. 多智能体技术在半导体生产线调度中的应用[J]. *计算机集成制造系统*, 2003, 9(8): 641-644
(Wu J W, Xiao Y S. Multi-agent Technology in Scheduling of Semiconductor Production Line [J]. *Computer Integrated Manufacturing Systems*, 2003, 9(8): 641-644)
- [4] Kennedy J, Eberhart R C. Particle Swarm Optimization [A]. *Proc of IEEE Int Conf on Neural Networks* [C]. Perth: IEEE Piscataway, 1995: 1942-1948
- [5] Eberhart R C, Shi Y. Particle Swarm Optimization: Developments, Applications and Resources [A]. *Proc of Congress on Evolutionary Computation* [C]. Seoul: IEEE, 2001: 81-86
- [6] Esmi A, Lambert Torres G, Zamboni de Souza A C. A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization [J]. *IEEE Trans on Power Systems*, 2005, 20(2): 859-866

(下转第978页)

好的聚类效果;

2) 该算法对于不同形状的数据集具有较好的聚类适应性;

3) 该算法对数据集中的噪音数据具有较好的抗干扰能力;

4) 该算法对于文本等非结构化的数据聚类并不适合^[6,7].

5.4 与DBSCAN和SUDBC算法的性能比较

DBSCAN算法是一种高效率的聚类算法,是典型的基于密度的聚类算法;SUDBC算法也是一种基于密度单元的聚类算法.图4为3种聚类算法对不同规模数据集进行聚类的性能比较.从图4中可以看出,SECDU算法具有最快的聚类速度,与其他两种算法相比,具有两个数量级的速度优势.可见对于大规模的数据集,SECDU算法的速度优势非常明显.

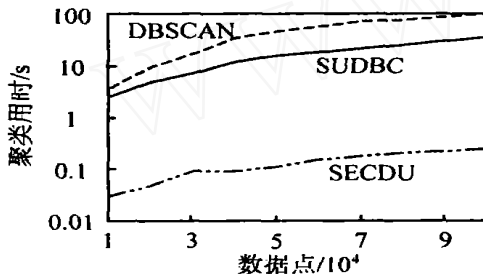


图4 3种聚类算法性能比较

6 结 语

本文介绍了一种基于密度单元的聚类算法SECDU.该算法聚类效果好、速度快,对具有不同分布特性的数据集有较好的适应性,适合聚类大规模的数据集.

聚类参数的设置问题在聚类分析领域中始终是一个难点问题,在SECDU算法中也是如此.未来的工作主要集中在如何更加合理地设置聚类参数,从而进一步提高聚类结果的质量.

参考文献(References)

- [1] Macqueen J. K-means: Some Methods for Classification and Analysis of Multivariate Observations [A]. *The 5th Berkeley Symp on Mathematical Statistics and Probability* [C]. Berkeley, 1976: 56-68.
- [2] Markus M, Breunig, Hans-Peter Kriegel, et al. Data Bubbles: Quality Preserving Performance Boosting for Hierarchical Clustering [A]. *ACM SIGMOD* [C]. Santa Barbara, 2001: 99-112.
- [3] Samer Nassar, Jorg Sander, Corrine Cheng. Incremental and Effective Data Summarization for Dynamic Hierarchical Clustering [A]. *ACM SIGMOD* [C]. Paris, 2004: 13-18.
- [4] Guha S, Rastogi R, Shim K. CURE: An Efficient Clustering Algorithm for Large Databases [A]. *ACM Special Interest Group on Management of Data* [C]. Washington, 1998: 73-84.
- [5] Zhang T, Ramakrishnan R, Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases [A]. *ACM SIGMOD Int Conf on Management of Data* [C]. Montreal, 1996: 103-114.
- [6] Ankerst M, Breunig M, Kriegel H, et al. OPTICS: Ordering Points to Identify the Clustering Structure [A]. *ACM Special Interest Group on Management of Data* [C]. Philadelphia, 1999: 49-60.
- [7] Sander J. Density-based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications [J]. *Data Mining and Knowledge Discovery*, 1998, 2(2): 169-194.
- [8] Ester M, Kriegel H, Sander J. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [A]. *Knowledge Discovery and Data Mining* [C]. Portland, 1996: 226-231.
- [9] 王明善, 沈恒慈. *概率论与数理统计* [M]. 北京: 高等教育出版社, 1999.
(Wang M S, Shen H C. *Probability and Statistics* [M]. Beijing: Higher Education Press, 1999.)

(上接第973页)

- [7] Gaing Z L. A Particle Swarm Optimization Approach for Optimum Design of PD Controller in AVR System [J]. *IEEE Trans on Energy Conversion*, 2004, 19(2): 384-391.
- [8] 李宁, 邹彤, 孙德宝. 带时间窗车辆路径问题的粒子群算法 [J]. *系统工程理论与实践*, 2004, 24(4): 130-135.
(Li N, Zou T, Sun D B. Particle Swarm Optimization

- for Vehicle Routing Problem with Time Windows [J]. *Systems Engineering — Theory and Practice*, 2004, 24(4): 130-135.)
- [9] Daniel Q, Heinrich K. Production Planning in Semiconductor Assembly [A]. *Proc of the 4th Aegean Int Conf on Analysis of Manufacturing Systems* [C]. Samos Island, 2003: 181-189.