

基于改进粒子群优化算法的虚拟机迁移选择策略研究

武兴宇 孙 磊 胡翠云 孙瑞辰
(解放军信息工程大学 郑州 450001)

摘 要 粒子群优化算法由于实现容易、精度高、收敛快,在解决多目标优化问题时呈现出较强的优越性。在定义匹配距离的基础上,引入粒子群优化算法思想制定虚拟机迁移选择策略,并对粒子群优化算法做出改进,引入规避列表思想,将剩余性能不满足虚拟机性能需求的服务器加入到规避列表中,以避免多个满足非劣最优解的虚拟机迁移到一台服务器,导致资源占用率超过结点资源上限。通过在 CloudSim 平台上与基本粒子群优化算法进行的仿真对比实验证明了本算法具有更快的收敛速度和选择速度。

关键词 粒子群优化算法,虚拟机迁移,选择策略,规避列表

中图法分类号 TP301 文献标识码 A

Research of Live Migration of Virtual Machines Selection Strategy Optimization Problems Based on Modified Particle Swarm Optimization

WU Xing-yu SUN Lei HU Cui-yun SUN Rui-chen
(PLA Information Engineering University, Zhengzhou 450001, China)

Abstract Due to easy implementation, high accuracy, fast convergence, particle swarm optimization algorithm is considered to have advantages in solving the problem of multi-objective optimization. Virtual machine migration selection policies are defined based on the definition of matching distance and the theory of particle swarm optimization algorithm. Moreover, the particle swarm optimization algorithm is improved by introducing the ideas of avoid list. In this way, the servers, which do not have enough remaining performance to meet the demand of the virtual machine, are added to the avoid list. Therefore, it can avoid multiple virtual machines that satisfy pareto optimal solutions to migrate to the same server, causing the resource usage rate to exceed the maximum resources limit of the node. Simulation experiment was done based on the CloudSim platform, and compared with basic particle swarm optimization algorithm, our algorithm was proved to have faster speed of convergence and choice.

Keywords Particle swarm optimization, Live migration of virtual machines, Selection policy, Avoid list

1 引言

近年来,在云计算高速发展下,虚拟化技术已经取得了长足的进展,为云计算的迅速发展提供了坚实的基础。互联网数据中心(Internet Data Center, IDC)在大规模使用虚拟化技术后,虚拟机的数量和负载会随用户的需求经常发生变化,从而对虚拟机集群的资源调度提出了挑战。虚拟机迁移技术^[1]可以把包括操作系统在内的整个运行环境从一台物理服务器迁移到另一台物理服务器上,从而有效管理数据中心的资源,方便物理服务器的维护和节能。

虚拟机在进行迁移时,要选择合适的虚拟机,以减少虚拟机迁移带来的系统开销。如何寻找最优的将虚拟机迁移到目标节点的方案,从而使每个节点中虚拟机的使用资源之和不超过节点服务器的资源上限,并且使用的服务器数量最少的问题,实际可以理解为一个多目标优化问题(Multi-objective

Optimization Problem, MOP)。大多数科学和工程问题都是多目标优化问题,往往不存在唯一的最优解,而是一组最优解集,如何快速获取这些最优解集是研究的重点。

文献^[2]提出了一个经典的多目标优化问题,即既要减少物理机和网络设备的数目,以降低 IDC 的能耗,同时还要保证应用的性能,避免网络堵塞。在解决多目标问题时,比较经典的有基于经典的第二代多目标优化算法 NSGA-II^[3]中的方法,以及基于 NSGA-II 算法提出的虚拟机放置多目标优化遗传算法^[4],它们利用遗传算法群体自我进化的特点,在保证服务器多级目标的前提下,有效减少了所使用服务器数量和虚拟机迁移次数,但没有分析算法复杂度。

粒子群优化算法^[5](Particle Swarm Optimization)由于具有计算简单、鲁棒性强、收敛快、依赖的经验参数少等优点,相比目前较为流行的蚁群算法等,在求解较为复杂的问题时显示出较好的优越性,在 MOP 问题中逐渐成为研究热点。

本文受国家 863 计划基金项目(2008AA01Z404),国防预研基金项目(910A26010306JB5201)资助。

武兴宇(1990—),男,硕士生,主要研究方向为虚拟化技术、信息安全, E-mail: xingyuwul23@sina.com; 孙 磊(1973—),男,博士,副研究员,主要研究方向为云计算基础设施可信增强、可信虚拟化技术; 胡翠云(1985—),女,博士,主要研究方向为云计算、可信虚拟化技术; 孙瑞辰(1991—),男,硕士生,主要研究方向为云计算、可信虚拟化技术。

文献[6]提出了一种基于全局最优位置自适应选取的多目标粒子群优化算法,其引入拥挤距离机制,具有较好的收敛性和分布性。文献[7]提出了一种改进的多目标粒子群优化算法,其通过增加跳数改变机制,改变粒子搜索的方式,使非劣最优解更加收敛于 Pareto 前沿,有较好的收敛性。

文献[8]将虚拟机放置问题分解成装箱的组合优化问题 and 多目标优化问题两部分。首先采用遗传算法处理虚拟机放置到结点的组合优化问题,然后结合模糊逻辑优化多个目标,包括总体资源浪费、能源消耗和热量消散代价最小化。

本文针对虚拟机迁移的选择策略问题,提出一种基于改进的粒子群算法(Modified Particle Swarm Optimization, MP-SO)的虚拟机迁移选择策略,即通过定义匹配距离,对多种性能量化处理,并将服务器剩余性能不满足待迁移虚拟机的服务器加入到为该待迁移虚拟机服务的粒子群的规避列表(Avoid List)中,使后续的粒子群搜索范围缩小,收敛速度更快。

2 粒子群优化算法

假设在一个 D 维的目标搜索空间中,有 n 个粒子组成一个群落,其中第 i 个粒子表示为一个 D 维的向量,

$$X_i = (x_1, x_2, \dots, x_i), i = 1, 2, \dots, n$$

第 i 个粒子的“飞行”速度也是一个 D 维的向量,记为

$$V_i = (v_1, v_2, \dots, v_i), i = 1, 2, \dots, n$$

第 i 个粒子迄今为止搜索到的最优位置称为个体极值,记为

$$p_{best} = (p_1, p_2, \dots, p_i), i = 1, 2, \dots, n$$

整个粒子群迄今为止搜索到的最优位置为全局极值,记为

$$g_{best} = (g_1, g_2, \dots, g_i), i = 1, 2, \dots, n$$

在找到这两个最优值时,粒子根据如下的式(1)和式(2)来更新自己的速度和位置。

$$v_i^{k+1} = \omega \times v_i^k + c_1 \times rand_1 \times (p_{best} - x_i^k) + c_2 \times rand_2 \times (g_{best} - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

其中, k 为迭代次数, c_1 和 c_2 为学习因子,也称加速常数(acceleration constant), r_1 和 r_2 为 $[0, 1]$ 范围内的均匀随机数。

3 基于改进粒子群算法的虚拟机迁移选择策略

· 虚拟机迁移选择策略

虚拟机和服务器性能特征主要包括 CPU、内存、网络带宽等。当使用 k 个性能特征描述虚拟机性能特征时,可为每个虚拟机建立性能需求向量 $(VM_{n1}, \dots, VM_{nk})$,为每个服务器建立性能向量 $(Ser_{m1}, \dots, Ser_{mk})$ 。若 $(Ser_{mk} - VM_{nk})$ 为负,说明服务器无法满足虚拟机的性能需求;若为正值,则需要衡量虚拟机和各服务器之间的适应程度(Fitness)。

由于不同性能的量纲不同,服务器剩余性能和虚拟机需求性能的数量级也存在差异,因此,将服务器剩余性能和虚拟机需求性能进行去量纲归一化处理[9]。

设 $\{Ser_{1p}, \dots, Ser_{mp}\}$ 为服务器 Ser_m 同类剩余性能 p 的集合, $\{VM_{1p}, \dots, VM_{np}\}$ 为虚拟机 VM_n 相对应同类需求性能 p 的集合。服务器性能归一化算法为:

$$S_{mp} = \frac{Ser_{mp} - Ser_p^{\min}}{Ser_p^{\max} - Ser_p^{\min}}$$

其中, $Ser_p^{\max}$ 为服务器同类剩余性能 p 的集合 $\{Ser_{1p}, \dots, Ser_{mp}\}$ 的最大值, $Ser_p^{\min}$ 为最小值。虚拟机需求性能归一化算法为

$$V_{np} = \frac{VM_{np} - VM_p^{\min}}{VM_p^{\max} - VM_p^{\min}}$$

为了定量分析虚拟机和服务器的适应度,本文提出匹配距离 $Mad_{VM_n Ser_m}$ (以下简称为 Mad_{nm}) 概念。匹配距离主要通过待迁移虚拟机需求性能和服务器剩余性能的欧氏距离得出:

$$Fit_m = Mad_m = \sqrt{\sum_{p=1}^k (S_{np} - V_{mp})^2} \quad (3)$$

匹配距离越小,说明服务器剩余性能越适应待迁移虚拟机的性能需求。在本文算法中,匹配距离的大小反映了适应度值的大小,适应度值越小,说明服务器越适应待迁移虚拟机。

· 服务器规避列表

下面给出服务器规避列表的定义:

定义 1 若虚拟机迁移到服务器后,服务器的剩余性能不满足某些虚拟机的性能需求,则将服务器加入到为该待迁移虚拟机服务的粒子群规避列表中。

为了使每个节点中虚拟机的使用资源之和不超过节点服务器的资源上限并且使用的服务器数量最少,当服务器迁移了一台虚拟机后,剩余性能 Ser_{np} 如果不满足其他虚拟机的性能需求 VM_{mp} ,则加入到属于该待迁移虚拟机的粒子群规避列表中。

· MPSO 算法流程

下面给出 MPSO 算法流程:

- ① 初始化粒子群,包括群体规模 N 、每个粒子的位置 x_i 和速度 V_i 。
- ② 初始化规避列表,规避列表清空。
- ③ 根据目标函数和式(3)计算每个粒子的适应度值 $f(i)$ 。

For $i=1$ to N

Fitness 1[i] = $f_1(X[i])$;

...

Fitness n[i] = $f_n(X[i])$;

Next i

- ④ 依据目标函数,对每个粒子,求得个体极值 $pBest$ 。

For $i=1$ to N

$pBest[1, i] \leftarrow f_1(x)$;

...

$pBest[n, i] \leftarrow f_n(x)$;

Next i

- ⑤ 在目标函数下,对 $f_1(x), \dots, f_n(x)$ 求得全局极值 $gBest$ 。

For $i=1$ to N

$gBest[1, i] \leftarrow f_1(x)$;

...

$gBest[n, i] \leftarrow f_n(x)$;

Next i

- ⑥ 根据全局极值更新规避列表,若服务器剩余性能不满足虚拟机的需求性能,则将服务器加入到该待迁移虚拟机粒子群规避列表中。

- ⑦ 根据式(1)、式(2)更新粒子的速度 v_i 和位置 x_i 。

- ⑧ 如果满足结束条件(虚拟机都迁移至最合适的服务器或到达最大循环次数),退出,否则返回②。

算法流程如图 1 所示。

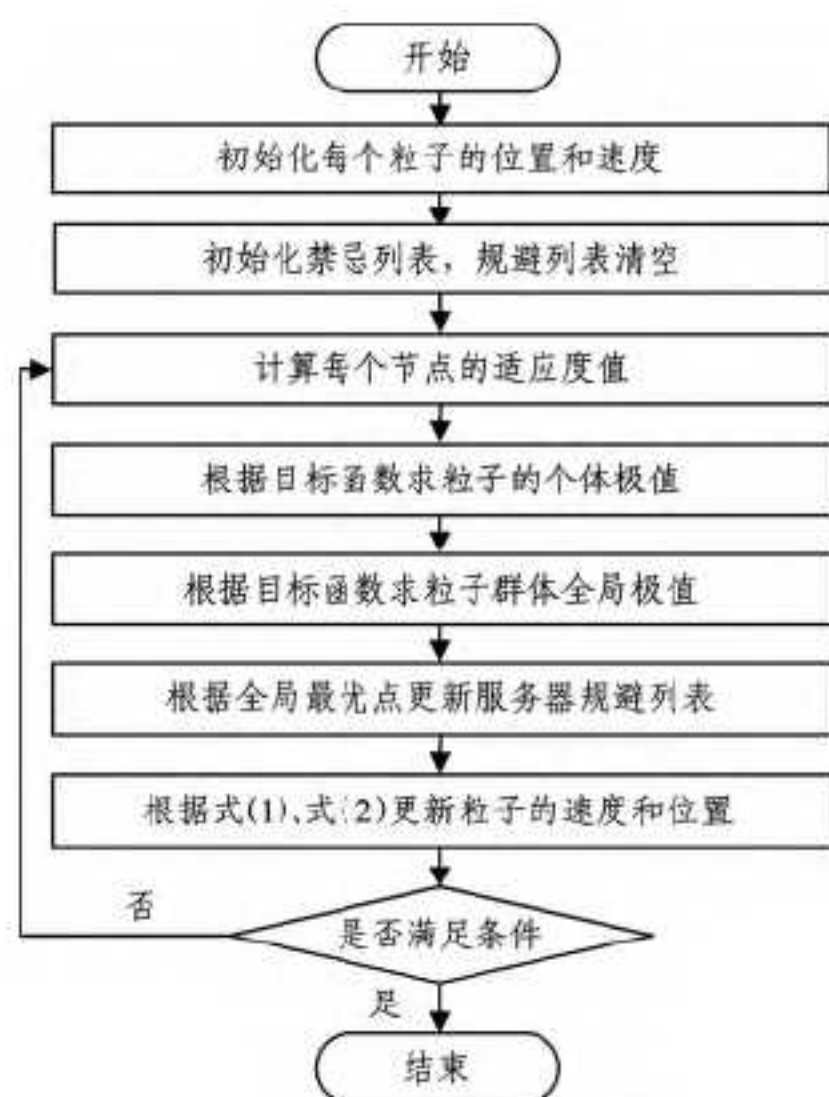


图1 算法流程

4 仿真实验与分析

4.1 参数设置

本文采用澳大利亚墨尔本大学的网络实验室和 Gridbus 项目提出的云仿真平台 CloudSim^[10] 作为实验仿真工具。

需要对 Vm、Host、Datacenter 类进行修改, 添加继承类的属性与方法。

通过重新编写 Datacenter 类, 建立一个包含 25 个服务器 (Host) 的 Datacenter。对 Host 类进行扩展, 加入服务器可用性能量化处理, 在方法 allocateHostForVm(Vm vm) 中计算匹配向量, 选取目标服务器 (Host)。

使用 Ant 对扩展的 CloudSim 重新编译, 首先将命令行锁定到 CloudSim 根路径 build.xml 所在的目录下, 随后在命令行下输入命令 D:\cloudsim-3.0.2>ant, 对 CloudSim 源文件进行批量编译, 生成的文件将根据 build.xml 指定路径存储到相应的位置。最后在命令行中输入 D:\cloudsim-3.0.2>ant makejar 命令打包生成 cloudsim-3.0.2-new.jar 文件。

选取 CPU、内存和网络带宽 3 个性能指标作为匹配距离衡量参数, 如表 1 所列; 并提交 100 个虚拟机 (Vm) 迁移任务请求, 如表 2 所列。

表1 服务器列表

Num	CPU/个	CPU/MIPS	内存/Gb	带宽/Mb/s
0	2	2000	4	1000
1	2	3000	4	1500
2	2	2000	4	2000
3	4	3000	6	1500
4	4	3000	6	2500
⋮	⋮	⋮	⋮	⋮
25	6	8000	8	5000

表2 提交虚拟机列表

Num	CPU/个	CPU/MIPS	内存/Gb	带宽/Mb/s
0	1	300	1	100
1	1	350	1	200
2	1	250	0.5	300
3	1	300	1	250
4	1	350	0.5	350
5	2	400	1.5	400
6	2	350	1	500
7	2	25	1.5	450
8	2	300	0.5	300
⋮	⋮	⋮	⋮	⋮
100	4	800	3	600

算法所选取的测试函数为 Deb 在文献^[11]中提出的, 这些函数在其它文献中经常被用于最优值算法的求解, 具有很强的代表性, 对这些函数的最优值求解结果能够反映出算法的改进效果。测试函数如下:

$$\text{Min } f_1(x_1, x_2) = x_1$$

$$\text{Min } f_2(x_1, x_2) = \frac{g(x_2)}{x_1}$$

$$g(x_2) = 2.0 - \exp\left\{-\left[\frac{x_2 - 0.2}{0.004}\right]^2\right\} - 0.8 \exp\left\{-\left[\frac{x_2 - 0.6}{0.004}\right]^2\right\}$$

对文中算法参数的设置如下: 学习因子 $c_1 = c_2 = 1.49$, $\omega = 0.7$, 每个函数进行 100 次迭代, 运行 200 次, 对最优值取平均数。

4.2 算法比较分析

为了验证算法的收敛性, 将本文提出的 MPSO 与基本 PSO 进行比较, 在每秒提交相同任务请求并发个数的前提下, 对算法的平均收敛时间进行比较, 如图 2 所示。可以看出, 本文的算法在收敛速度方面优于基本粒子群优化算法, 能够更快地获得搜索结果。

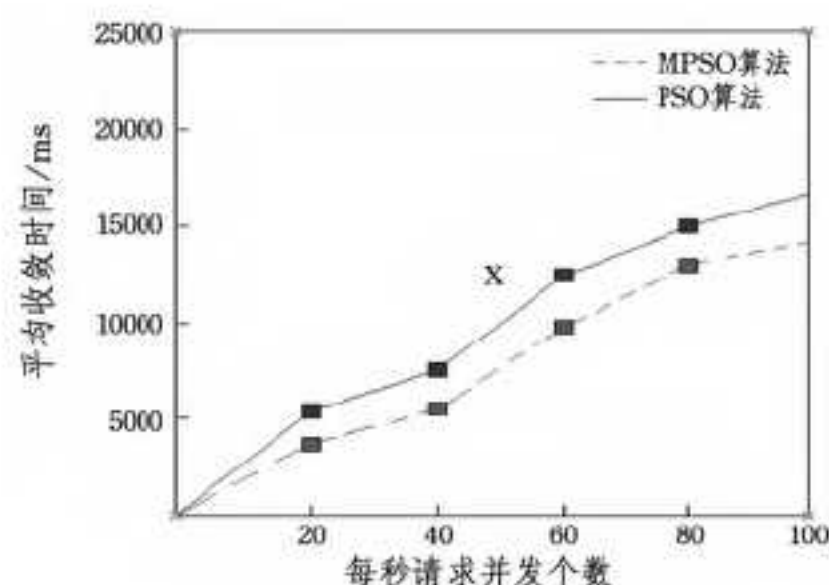


图2 MPSO 和 PSO 收敛速度情况比较

由测试结果可以看出, 随着每秒虚拟机迁移请求个数的增加, 两种算法的收敛时间都逐渐增长, 但本文提出的 MPSO 算法收敛时间一直少于基本粒子群算法, 大约减少了 15% 到 21%。

为了对比本文选择策略的选择速度, 设计如下场景: 选取 5 个时刻 ($t_1, \dots, t_5$), 时间间隔为 30 分钟, t_1 时刻发出请求虚拟机个数为 10, t_2 时刻为 17 个, t_3 时刻为 15 个, t_4 时刻为 20 个, t_5 时刻为 16 个。对比本文算法和基本粒子群优化算法在各时刻从虚拟机迁移任务请求发出到选择服务器的时间, 结果如图 3 所示。

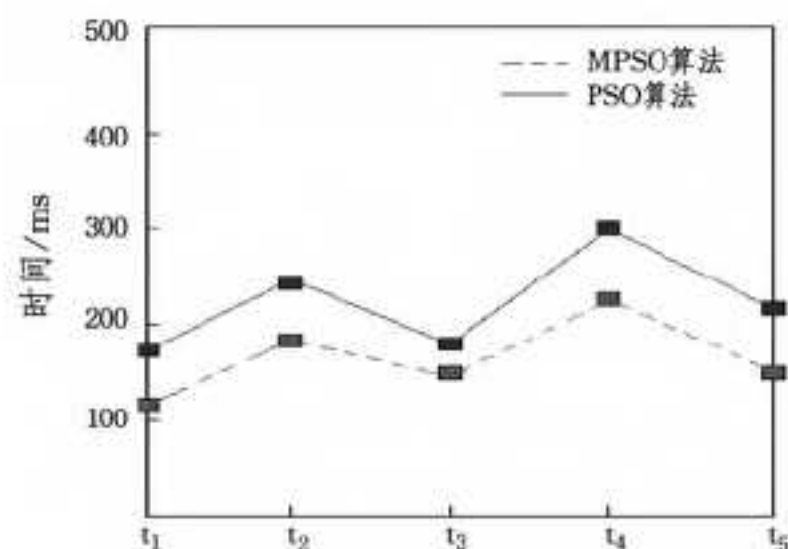


图3 各时刻选择速度比较情况

由测试结果可以看出, 在处理相同虚拟机迁移任务请求个数时, 基于 MPSO 算法的选择策略在选择速度上比基于粒子群优化算法的选择策略大约降低了 20% 到 35%。

结束语 本文提出一种基于粒子群优化算法的虚拟机迁移选择策略。通过定义匹配距离定量服务器和虚拟机的适应

度,加入规避列表,避免了虚拟机所占用的资源超过服务器资源上限,并达到了组合优化的目标。实验证明了该算法的有效性,同时算法的收敛速度得到了提高。

下一步工作将主要围绕算法复杂度的改进和解决高维多目标优化问题展开。

参 考 文 献

- [1] Nelson M, Lim Beng-Hong, Hutchins G. Fast Transparent Migration for Virtual Machines [C] // Proceedings of USENIX ATC. 2005
- [2] Deb K. Multi-Objective Optimization [M]. Springer: Multi-Objective Optimization Using Evolutionary Algorithms, 2001: 13-46
- [3] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197
- [4] 李强, 郝沁汾, 肖利民, 等. 云计算中虚拟机放置的自适应管理与多目标优化[J]. 计算机学报, 2011, 34(12): 2253-2264
- [5] Kennedy J, Eberhart R. Particle swarm optimization[C] // Proceedings of IEEE International Conference on Neural Networks.

(上接第 19 页)

表 4 检测误差

	d_0	d_G	d_I
本文算法	255.5626	258.759	255.5865
Hough 变换	255.2288	253.3537	255.6493
δ_1	0.13%	2.12%	0.02%
δ_2	0%	1.25%	0.01%

实验表明,文中的方法在保持与标准 Hough 变换相同精度的前提下,大大提高了直线检测效率。加入噪声后,计算机需要处理的特征点数增加,所以检测时间增加。由于脉冲噪声是随机分布的,只有部分噪声点分布在边缘附近,这些噪声可以通过迭代拟合被删除,因此脉冲噪声对该算法的检测精度影响较小。高斯噪声属于加性噪声,即噪声值直接加到原图上,使边界模糊。因此在高斯噪声的影响下,该算法的检测精度略有下降。

结束语 1) 基于“多对一”的映射原理,结合热熔法制得预浸料图像的特点,提出了一种基于直线拟合的预浸料边缘直线检测方法。

2) 所提算法利用“局部-整体”的思想,相比 Hough 变换大大减少了计算量,提高了检测速度。

3) 直线拟合时引入权重系数,通过迭代计算,有效抑制预浸料边缘因纤维损伤外露而引起的伪边缘点对检测精度的影响,提高了检测的鲁棒性。

参 考 文 献

- [1] 文立伟, 张建宝, 肖军, 等. 大型筒段构件自动铺带成型技术[J]. 航空学报, 2009, 30(12): 2456-2461
- [2] Gutowski T G. Advance composite manufacturing [M]. New York: John Wiley & Sons Press, 1997: 32-83

volume 4, 1995

- [6] 黄敏, 江渝, 毛安, 等. 基于全局最优位置自适应选取与局部搜索的多目标粒子群优化算法[J]. 计算机应用, 2014, 34(4): 1074-1079
- [7] 冯金芝, 陈兴, 郑松林. 一种改进的多目标粒子群优化算法及应用[J]. 计算机应用研究, 2014, 31(3): 1001-1005
- [8] Xu J, Fortes J. Multi-objective virtual machine placement in virtualized data center environments [C] // Proceedings of 2010 IEEE/ACM International Conference on Green Computing and Communications (GreenCom'2010). Hangzhou, 2010: 179-188
- [9] 程理民, 吴江, 张玉林. 运筹学模型与方法教程[M]. 北京: 清华大学出版社, 2002
- [10] Calheiros R N, Ranjan R, De Rose C A F, et al. Cloudsim: A novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services [R]. GRIDS-TR-2009-1. Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia 2009
- [11] Deb K. Multi-objective genetic algorithms: Problem difficulties and construction of test problem[J]. Evolutionary computing, 1999, 7: 205-230

- [3] 李俊斐, 王显峰, 肖军. 开孔曲面自动铺放轨迹规划算法研究[J]. 航空学报, 2013, 34(7): 1716-1723
- [4] 李俊斐, 王显峰, 肖军. 网格化曲面的固定角度铺丝轨迹规划算法[J]. 计算机辅助设计与图形学学报, 2013, 25(9): 1410-1415
- [5] 杜善义. 先进复合材料与航空航天[J]. 复合材料学报, 2007, 24(1): 1-12
- [6] 张旭坡. 自动铺丝预浸纱制备技术研究[D]. 南京: 南京航空航天大学, 2007
- [7] 李成功, 傅恒志, 于翹. 航空航天材料[M]. 北京: 国防工业出版社, 2002: 132-133
- [8] Hogg M W. Method for laying composite tape; U. S. Patent 7, 842, 145[P]. 2010-11-30
- [9] 丁幼春, 陈红. 基于 HT 的多条直线的检测特点及其算法改进[J]. 华中农业大学学报, 2008, 27(6): 802-806
- [10] 张江鑫, 沈小兰, 王辉, 等. 快速随机 Hough 变换多直线检测算法[J]. 浙江工业大学学报, 2013, 41(3): 346-350
- [11] 宋冬冬, 薛锴. 基于直线拟合的高温板材边缘检测方法[J]. 工具技术, 2010, 44(7): 37-39
- [12] Kyrki V, Kälviäinen H. Combination of local and global line extraction[J]. Real-Time Imaging, 2000, 6(2): 79-91
- [13] Galamhos C, Matas J, Kittler J. Progressive probabilistic Hough transform for line detection[C] // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. New York, NY, USA: IEEE, 1999: 1-10
- [14] 王燕清, 辛柯俊, 陈德运, 等. 基于启发式概率 Hough 变换的道路边缘检测方法[J]. 计算机科学, 2013, 9(40): 279-287
- [15] Huang Qiang, Li Ming, Ji Zhen. Precisely Locating Multi-QR Code Based on Straight Line Fitting[C] // 2010 Chinese Conference on Pattern Recognition (CCPR). New York, NY, USA: IEEE, 2010: 1-5