

一种自适应权重的并行 PSO 快速装箱算法

廖 星 袁景凌 陈旻骋

(武汉理工大学计算机科学与技术学院 武汉 430070)

摘 要 随着智能制造时代的到来,生产线后期产品的智能装箱已成为工业生产的重要环节,如何更快速地得到装箱结果对于提高生产效率尤为重要。以快速装箱为目标,文中提出了一种适用于工业生产线的智能化装箱算法。该算法采用自适应权重法改进了粒子群优化算法,相较于标准粒子群优化及遗传等传统启发式算法有更快的收敛速度;并采用 GPU 加速,实现了高性能的并行计算,大幅加快了计算速度。实验表明,所提算法同样能得到很好的空间利用率,同时其收敛速度也显著优于传统算法。

关键词 智能装箱,粒子群优化,自适应权重,并行计算,GPU 加速

中图分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.03.036

Parallel PSO Container Packing Algorithm with Adaptive Weight

LIAO Xing YUAN Jing-ling CHEN Min-cheng

(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract With the arrival of intelligent manufacturing, the intelligent packing of product or container in the late production line has become an important part of industrial production, and how to get the packing results faster is also important for improving the production efficiency. Mainly aiming at the rapid packing, this paper proposed an intelligent packing algorithm for industrial production line. The algorithm uses the adaptive weight method to improve the particle swarm optimization algorithm, which has a faster convergence rate than the traditional heuristic algorithm, such as standard particle swarm optimization algorithm and genetic algorithm. The calculation speed is greatly accelerated by achieving high performance parallel computing with GPU acceleration. Experiments show that the algorithm proposed in this paper can also get very high space utilization rate, and its convergence speed is faster than the traditional algorithm.

Keywords Intelligent packing, PSO, Adaptive weight, Parallel computing, GPU acceleration

1 引言

装箱问题(Container Packing Problem)是计算机学科领域中的经典问题之一,也是一个 NP 难题,通常很难获得求解该问题的有效算法。随着工业 4.0 和智能制造时代的到来,在工业生产、物流运输等领域,加快生产线的装箱速度、降低生产成本、提高生产效率,在工业实践和科学研究中愈发重要。

目前,国内外已有较多相关研究,它们大多采用遗传算法等启发式算法来获得近似解。Ngoi 等人^[1]和 Bischoff 等人^[2]人提出了一种启发式算法,并根据该算法产生了一些装箱实例^[2-3]。Gehring 等人^[4]基于塔的概念,提出了一种遗传算法。Bortfeldt 等人^[5]进一步扩展了块的概念,提出了一种禁忌搜索和混合遗传算法。Moura 等人^[6]提出了一种基于

“剩余空间”的贪心随机自适应启发式算法 GRASP。Bischoff^[7]在层的基础上,提出了一种基于分支定界法的启发式算法。张德富等人^[8]基于块装载的方法,提出了一种多层启发式搜索算法。刘胜等人^[9]通过优条和优层将问题降维,提出了一种启发式正交二叉树搜索算法。以上算法均有很好的空间利用率,但是遗传算法等启发式算法的收敛速度较慢,无法满足工业生产的快速化要求。

粒子群优化(Particle Swarm Optimization, PSO)算法是一种基于群体智能的全局随机搜索算法^[10],是研究人员受人工生命研究的启发,通过模拟鸟群觅食过程中迁徙和群聚的行为而提出的^[11]。PSO 算法由于优化效果明显,收敛速度快,因此近年来得到了广泛研究。但是标准 PSO 算法在进化后期容易缺失粒子多样性,陷入局部最优解,从而出现早熟收敛^[12]。

收稿日期:2017-06-19 返修日期:2017-08-31 本文受国家自然科学基金(61303029),教育部留学回国启动基金([2012]1707),湖北省自然科学基金(2014CFB836)资助。

廖 星(1994—),男,硕士生,主要研究方向为智能计算、机器学习,E-mail:crazyurus@whut.edu.cn;袁景凌(1975—),女,教授,博士生导师,CCF 会员,主要研究方向为绿色计算、机器学习、数据挖掘,E-mail:yuanjingling@126.com(通信作者);陈旻骋(1990—),男,博士生,主要研究方向为机器学习、数据挖掘,E-mail:wester589@gmail.com。

本文针对遗传算法收敛速度慢以及标准 PSO 算法易早熟问题,提出了一种快速装箱算法。主要工作如下:

1)改进标准 PSO 算法。通过引入自适应权重,可以避免 PSO 算法早熟,从而跳出局部最优。

2)实现并行化 PSO 算法。对改进后的 PSO 算法进行并行计算分析,并利用 GPU 加速了 PSO 算法的计算。

与现有研究相比,改进后的算法同样能获得很好的空间利用率,并且优于传统算法。

2 问题描述

装箱问题是指将箱子按照一定顺序装入容器中,目的是使容器的空间利用率达到最高。由于装箱问题被应用于不同的领域,因此实际上被分成了多种不同的类型,存有不同的优化目标和装载约束。张德富等人总结了装箱问题的类型,并给出了相应的分类^[3],如一维、二维、三维装箱问题及方向性、稳定性、完全切割等约束条件下的装箱问题。本文研究具有约束的三维装箱问题,其形式化定义描述如下。

给定一个容器 C 和一系列待装载箱子的集合,容器和待装箱的形状均为长方体。容器 C 的长为 l_c , 宽为 w_c , 高为 h_c , 体积 $v = l_c w_c h_c$; 每个箱子 b_i 的长为 l_i , 宽为 w_i , 高为 h_i , 每个箱子的体积为 $v_i = l_i w_i h_i$ 。则有:

$$C = (l_c, w_c, h_c)$$

$$B = \{b_1, b_2, \dots, b_n\}$$

$$b_i = (l_i, w_i, h_i)$$

设 F 是 B 的一个子集,定义 F 中所有箱子的体积之和为:

$$V_F = \sum_{b_i \in F} v_i$$

定义子集 F 对应的空间利用率为 V_F/V 。问题的目标是从 B 中选择一个子集 F ,使得空间利用率达到最大,并且满足以下条件:

- 1)在容器 C 中,任何箱子 $b_i \in F$ 均有一个装填位置;
- 2)在容器中放置的任何两个箱子都不重叠;
- 3)所有 F 中的箱子全部包含在 C 中。

此外,根据工业生产的实际需要,本文还考虑了如下约束条件:

C1)方向性约束(Orientation Constraint),即待装箱的装载是具有方向的,并不是所有边都可以垂直放置。该约束定义了箱子的长、宽、高是否可以垂直放置,并且定义了至少有一边可以垂直放置。

C2)稳定性约束(Support Constraint),即箱子的装载必须稳定,每个被装载的箱子必须有容器底部或者其他已经装载的箱子作为支撑。

3 快速装箱算法

3.1 标准 PSO 算法

假设一个 D 维的搜索空间和一个包含 N 个粒子的粒子群在其约束条件下搜寻全局最优解,每个粒子包含 3 个 D 维向量的信息,分别为:速度向量 $V_i = \{V_i^1, V_i^2, \dots, V_i^D\}$,位置向量 $X_i = \{X_i^1, X_i^2, \dots, X_i^D\}$ 和自身最优位置向量 $P_i = \{P_i^1, P_i^2, \dots, P_i^D\}$ 。在连续的搜索空间中,一个位置向量的每个维度

对应于决策变量的一个值,换言之,每个粒子的位置对应于问题的一个潜在解,该粒子的适应度可以通过把 P_i 中的各个值代入到目标函数中进行计算来获得,适应度越大说明粒子的位置越好。速度向量代表了粒子在各个维度上经过的距离,自身的最优位置向量是各个粒子各自到达过的最佳位置。种群更应考虑全局最优位置,即种群中使适应度值最优的那个粒子的最优位置,表示为 $P_g = \{P_g^1, P_g^2, \dots, P_g^D\}$ 。一个粒子的状态由其位置和速度两个因素表征,其更新规则分别由以下两个公式表示:

$$V_k^i = \omega V_k^{i-1} + c_1 r_1 (P_i^{k-1} - X_i^{k-1}) + c_2 r_2 (P_g^{k-1} - X_i^{k-1})$$

$$X_k^i = X_i^{k-1} + V_k^i$$

其中, k 表示迭代的次数; i 为种群中的第 i 个粒子; ω 是惯性权重,表示粒子受自身影响的程度,它能够调整粒子的飞行速度,从而使粒子趋于收敛; c_1 和 c_2 为学习因子,分别表示粒子受个体经验和种群经验的影响程度; r_1 和 r_2 是 $(0, 1)$ 范围内的随机数; P_i^k 表示第 i 个粒子的个体最优位置; P_g^k 表示粒子种群的全局最优位置; X_i^k 表示第 i 个粒子在第 k 代的位置。

3.2 自适应权重的 PSO 算法

与其他进化算法相比,PSO 算法同样存在早熟现象,容易陷入局部最优解。早熟现象即指种群在找到全局最优点之前已经停滞不动,这些早熟收敛点可能是局部极小值,也可能是局部极小值邻域内的一个点。为了使 PSO 算法逃离局部极小点,避免早熟,本文将自适应权重、压缩因子以及变学习因子引入到 PSO 算法中。通过引入以上参数,可以有效地改善早熟现象。自适应权重的 PSO 算法的流程图如图 1 所示。

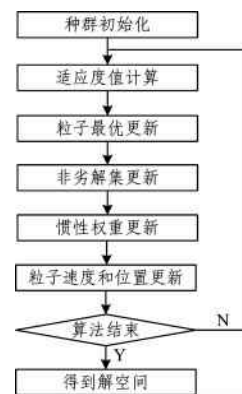


图 1 自适应权重的 PSO 算法流程

Fig. 1 Flow of adaptive weighting PSO algorithm

其中,惯性权重是 PSO 算法的核心参数,对算法的收敛性能有重要影响^[14]。通过在 PSO 算法的每一轮迭代中实时监测粒子种群的运动状态,并根据种群的运动状态来动态调整各粒子的惯性权重,实现自适应的权重,可以消除算法迭代过程中粒子惯性分量的不良影响,平衡全局搜索和局部搜索能力。惯性权重的更新公式为:

$$\omega = \omega_{\max} - \frac{t(\omega_{\max} - \omega_{\min})}{t_{\max}}$$

如果仍未跳出局部最优,可采用随机权重来达到自适应:

$$\begin{cases} \omega = \mu + \sigma * N(0, 1) \\ \mu = \mu_{\min} + (\mu_{\max} - \mu_{\min}) * \text{rand}(0, 1) \end{cases}$$

引入压缩因子可以更好地优化学习因子,减小惯性的影响。粒子的更新公式变为:

$$V_i^k = \phi[\omega V_i^{k-1} + c_1 r_1 (P_i^{k-1} - X_i^{k-1}) + c_2 r_2 (P_g^k - X_i^{k-1})]$$

其中:

$$\phi = \frac{2}{2 - C - \sqrt{C^2 - 4C}}$$

$$C = c_1 + c_2$$

引入变学习因子同样可以减小惯性的影响。学习因子的更新公式如下:

$$c_1 = c_2 = c_{\max} - \frac{k(c_{\max} - c_{\min})}{k_{\max}}$$

其中, k 是收敛因子。

3.3 并行 PSO 算法

PSO 算法在计算过程中的并行性主要体现在以下 3 个方面^[15]:

- 1) 计算粒子个体位置与评价个体的最佳位置并更新是并行的;
- 2) 种群的最佳位置的评价和更新是并行的;
- 3) 产生下一代种群的过程是并行的。

由此可知,PSO 算法易于并行计算,适合通过 GPU 实现加速。

根据 PSO 算法的并行特点,本文采用粒子个体和线程对应的策略来实现 GPU 中的并行化计算。具体做法是:在 GPU 中创建与粒子相同数量的线程,并为每个粒子分配独立的计算空间。每个线程执行一个粒子的寻优计算,利用 GPU 计算的并行性,将多个粒子的计算时间缩短成一个粒子的计算时间,从而提升计算速度。图 2 是基于 GPU 的粒子位置更新并行计算模型。

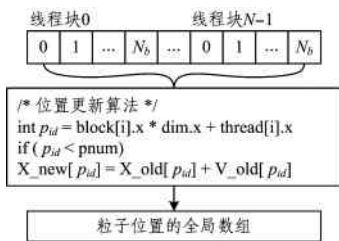


图 2 基于线程的并行执行过程

Fig. 2 Thread-based parallel execution process

其中, $N_b + 1$ 代表每个线程块内的线程数加 1。CUDA 中的一个 wrap 包含 32 个线程,每个线程长 32 bit,因此 GPU 的计算单位 half-warp 刚好是 $32 \text{ bit} * 32 / 2 = 64 \text{ Byte}$; p_{id} 代表当前线程正在计算的粒子; P_{num} 代表粒子总数; N 代表线程块总数。

3.4 装箱算法

本文采用了三空间分割法进行装箱。

开始装箱之前,首先按照约束条件整理要装载的箱子,包括箱子的装载顺序、是否符合方向性约束等。装载开始时,首先将整个容器作为当前的装载空间,并将第一个箱子放置在容器靠近原点的左下角位置,使得容器被分成 3 个空间:上空间、右空间和前空间,如图 3 所示。之所以这样分割空间,一方面是为了使剩余空间尽可能大,有利于提高空间的利用率;

另一方面是确保小箱子在大箱子上面,满足稳定性约束。在放入第二个箱子时就将这 3 个空间作为当前的装载空间,每个装载空间放入箱子后又被分割成上、右、前 3 个空间。不断重复上述过程,将箱子依次装入,直到箱子全部装入容器或容器没有空间。

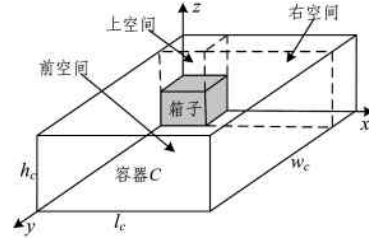


图 3 三空间分割示意图

Fig. 3 Division diagram of three space

自适应权重的并行 PSO 装箱算法如下。

算法 1 自适应权重的并行 PSO 装箱算法

```

AdaptiveWeightPSO(B, C)
V = l_c * w_c * h_c
// 随机生成 m * n 个装箱序列
F = random(B, C, m, n)
Do
// 并行计算每个装箱序列的空间利用率
Parallel. For(i = 1: m)
    Parallel. For(j = 1: n)
        v[j] = F[i][j].l * F[i][j].w * F[i][j].h
        x[i] = x[i] + v[j]
// 选出种群内的最优装箱序列
pg = Parallel. best(x)
// 更新粒子的速度和位置
Parallel. For(i = 1: m)
// 惯性权重的更新
w = (w_max - w_min) * i / m
// 学习因子更新
c_1 = c_2 = c_max - (c_max - c_min) * i / m
Parallel. For(j = 1: n)
    v[j] = w * v[j] + c_1 * rand(1, 1) * (p[j] - x[j]) + c_2 * rand
        (1, 1) * (pg[i] - x[j])
    x[j] = x[j] + v[j]
    p[j] = Parallel. best(x[j], x[j-1])
F = update(pg, p) // 更新装箱序列
While(fitness(F) || m) // 适应度满足要求或达到最大迭代次数
Return F
    
```

4 实验设计与结果分析

实验在 Windows10 环境下使用配置为 IntelCorei5 6500, RAM 16GB, NVIDIA GeForce GTX 1080 Ti 的计算机运行;将 MATLAB R2014a 作为开发环境,使用了 Parallel Computing Toolbox on CUDA GPUs 工具箱。

4.1 有效性实验

为了验证所提算法的有效性,将其与标准 PSO 算法、遗传算法进行了实验对比,以上算法均满足装箱问题的约束条件。实验数据集采用 OR-Library 的 BR1-BR10 的 1000 个

三维装箱实例。这些实例共有 10 种类型,每种类型有 100 个实例。每个类别的箱数相同,箱子的多样性由弱到强,因而能够很好地测试在不同多样性下算法解决装箱问题的性能。表 1 列出 3 种算法在 10 组实验中对 100 个实例的装箱结果的平均空间利用率。

表 1 3 种算法的空间利用率对比

Table 1 Comparison of spatial utilization of three algorithms

	自适应权重的 并行 PSO 算法	标准 PSO 算法	遗传算法
BR1	0.940	0.930	0.935
BR2	0.950	0.915	0.940
BR3	0.965	0.960	0.980
BR4	0.955	0.920	0.950
BR5	0.920	0.905	0.945
BR6	0.975	0.950	0.970
BR7	0.930	0.920	0.930
BR8	0.925	0.900	0.915
BR9	0.970	0.960	0.985
BR10	0.950	0.935	0.940

本文算法的解空间分布如图 4 所示。算法搜索到的解构成了 Pareto 面,证明该算法具有很好的收敛效果。

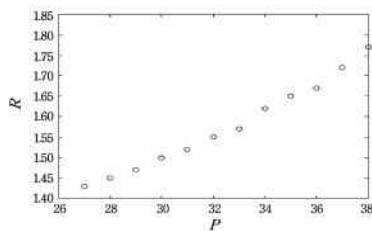


图 4 自适应权重的并行 PSO 算法解空间的分布

Fig. 4 Distribution of solution space of parallel PSO algorithm with adaptive weight

实验表明,本文所提算法是有效的,且具有很好的空间利用率。

4.2 快速性实验

为了验证所提算法具有更快的计算速度,同样将其与标准 PSO 算法、遗传算法进行了实验对比。实验数据集与 4.1 节一致,最终得到的结果如表 2 所列。

表 2 3 种算法的空间运行时间对比

Table 2 Comparison of space operation time of three algorithms

	自适应权重的 并行 PSO 算法	标准 PSO 算法	遗传算法
BR1	0.381989	3.168105	4.870194
BR2	0.374306	3.916100	4.915479
BR3	0.342067	3.202101	4.993638
BR4	0.379902	3.225172	5.232352
BR5	0.435398	3.568836	4.779449
BR6	0.360818	3.245126	4.977630
BR7	0.425196	3.388588	5.149225
BR8	0.447407	3.254509	5.400923
BR9	0.341328	3.191366	5.082133
BR10	0.340437	3.225567	4.830586

实验证明,本文提出的自适应权重的并行 PSO 算法要快于标准 PSO 算法和遗传算法,可以大幅缩短装箱问题的计算时间。

结束语 装箱问题经典且实用,空间利用率不断接近极限最优,要使其得到提高越来越困难。本文结合工业生产的实际情况,提出了自适应权重的并行 PSO 快速装箱算法,通过引入自适应权重法加快 PSO 算法的收敛速度并避免早熟,同时将算法并行化引入 GPU 加速运算中,以从快速装箱的角度出发来解决工业生产问题。本文所提算法具有较好的空间利用率,同时可大幅加快解决问题的速度,提高工业生产效率。

参考文献

- [1] NGOI B K A, TAY M L, CHUA E S. Applying spatial representation techniques to the container packing problem[J]. International Journal of Production Research, 1994, 32(1): 111-123.
- [2] BISCHOFF E E, RATCLIFF B S W. Issues in the development of approaches to container loading[J]. Omega, 1995, 23(4): 377-390.
- [3] DAVIES A P, BISCHOFF E E. Weight distribution considerations in container loading[J]. European Journal of Operational Research, 1999, 114(3): 509-527.
- [4] GEHRING H, BORTFELDT A. A genetic algorithm for solving the container loading problem[J]. International Transactions in Operational Research, 1997, 4(5/6): 401-418.
- [5] BORTFELDT A, GEHRING H. A hybrid genetic algorithm for the container loading problem[J]. European Journal of Operational Research, 2001, 131(1): 143-161.
- [6] MOURA A, OLIVEIRA J F. A GRASP approach to the container loading problem [J]. IEEE Intelligent Systems, 2005, 20(4): 50-57.
- [7] BISCHOFF E E. Three-dimensional packing of items with limited load bearing strength[J]. European Journal of Operational Research, 2006, 168(3): 952-966.
- [8] ZHANG D F, PENG Y, ZHANG L L. A Multi-Layer Heuristic Search Algorithm for Three Dimensional Container Loading Problem[J]. Chinese Journal of Computers, 2012, 35(12): 2553-2561. (in Chinese)
张德富, 彭煜, 张丽丽. 求解三维装箱问题的多层启发式搜索算法[J]. 计算机学报, 2012, 35(12): 2553-2561.
- [9] LIU S, ZHU F H, LV Y S, et al. A Heuristic Orthogonal Binary Tree Search Algorithm for Three Dimensional Container Loading Problem[J]. Chinese Journal of Computers, 2015, 38(8): 1530-1543. (in Chinese)
刘胜, 朱风华, 吕宜生, 等. 求解三维装箱问题的启发式正交二叉树搜索算法[J]. 计算机学报, 2015, 38(8): 1530-1543.
- [10] WEI Z, WU L, GE F Z, et al. Hybrid PSO Algorithm Based on Memetic Framework[J]. Pattern Recognition and Artificial Intelligence, 2012, 25(2): 213-219. (in Chinese)
魏臻, 吴雷, 葛方振, 等. 基于 Memetic 框架的混合粒子群算法[J]. 模式识别与人工智能, 2012, 25(2): 213-219.
- [11] REN P. Research and Applications of Particle Swarm Optimization[D]. Shenyang: Northeastern University, 2006. (in Chinese)
任萍. 粒子群优化算法研究与应用[D]. 沈阳: 东北大学, 2006.

步提高基于 PCANet 的车牌检测性能。

参 考 文 献

- [1] LE T S, TRAN V N, HAMAMOTO K. A robust and flexible license plate detection method[C]// Proceedings of International Conference on Advanced Technologies for Communications, Hanoi, Vietnam, 2015: 326-331.
- [2] DU S, IBRAHIM M, SHEHATA M, et al. Automatic license plate recognition (ALPR): a state-of-the-art review[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2013, 23(2): 311-325.
- [3] ZHENG D, ZHAO Y, WANG J. An efficient method of license plate location[J]. Pattern Recognition Letters, 2005, 26(15): 2431-2438.
- [4] FARADJI F, REZAIE A H, ZIARATBAN M. A morphological-based license plate location[C]// Proceedings of International Conference on Image Processing, San Antonio, Texas, USA, 2007: 57-60.
- [5] DUN J, ZHANG S, YE X, et al. Chinese license plate localization in multi-lane with complex background based on concomitant colors[J]. IEEE Intelligent Transportation Systems Magazine, 2015, 7(3): 51-61.
- [6] ZHOU W, LI H, LU Y, et al. Principal visual word discovery for automatic license plate detection [J]. IEEE Transactions on Image Processing, 2012, 21(9): 4269-4279.
- [7] WAFY M, MADBOULY A M M. Efficient method for vehicle license plate identification based on learning a morphological feature[J]. Intelligent Transport Systems, 2016, 10(6): 389-395.
- [8] FAN C M, INFORMATION S O, UNIVERSITY W. Introduction and analyzing of the license plate location technology[J]. Information Technology, 2013, 37(11): 167-168.
- [9] FAN R E, CHANG K W, HSIEH C J, et al. Liblinear: A Library for Large Linear Classification[J]. Journal of Machine Learning Research, 2008, 9(9): 1871-1874.
- [10] WANG R, SANG N, WANG R, et al. Novel license plate detection method for complex scenes[C]// Proceedings of International Conference on Image & Graphics, Qingdao, Shandong, China, 2013: 318-322.
- [11] CHEUNG B, SABLE C. Hybrid evolution of convolutional networks[C]// Proceedings of International Conference on Machine Learning and Applications Workshops, Honolulu, Hawaii, USA, 2011: 293-297.
- [12] HE K, ZHANG X, REN S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9): 1904-1916.
- [13] LIU X, GAO J, HE X, et al. Representation learning using multi-task deep neural networks for semantic classification and information retrieval[C]// Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, 2015: 912-921.
- [14] ZHAO Z H, YANG S P, ZENG-QIANG M A. License plate character recognition based on convolutional neural network lenet-5[J]. Journal of System Simulation, 2010, 22(3): 638-641.
- [15] CHAN T H, JIA K, GAO S, et al. PCANet: a simple deep learning baseline for image classification? [J]. IEEE Transactions on Image Processing, 2014, 24(12): 5017-5032.
- [16] SUN C, VALLOTTON P. Fast linear feature detection using multiple directional non-maximum suppression[C]// Proceedings of International Conference on Pattern Recognition, Hong Kong, China, 2009: 147-157.
- [17] LIU C, WECHSLER H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition[J]. IEEE Transactions on Image Processing, 2002, 11(4): 467-476.
- [18] CUN Y L, HUANG F J, BOTTOU L. Learning methods for generic object recognition with invariance to pose and lighting[C]// Proceedings of IEEE Computer Society Conference on Computer Vision & Pattern Recognition, USA, 2004: 97-104.
- (上接第 234 页)
- [12] MENG F, HUANG T A, XIE Z C. New Simplified Particle Swarm Optimization and Its Application to Container Loading Problem [J]. Science Technology and Engineering, 2013, 13(31): 9214-9218. (in Chinese)
孟菲, 黄太安, 解志斌. 一种简化粒子群算法及在三维装箱问题中的应用[J]. 科学技术与工程, 2013, 13(31): 9214-9218.
- [13] ZHANG D F, PENG Y, ZHU W X, et al. A Hybrid Simulated Annealing Algorithm for the Three-Dimensional Packing Problem[J]. Chinese Journal of Computers, 2009, 32(11): 2147-2156. (in Chinese)
张德富, 彭煜, 朱文兴, 等. 求解三维装箱问题的混合模拟退火算法[J]. 计算机学报, 2009, 32(11): 2147-2156.
- [14] AO Y C, SHI Y B, ZHANG W, et al. Improved Particle Swarm Optimization with Adaptive Inertia Weight[J]. Journal of University of Electronic Science and Technology of China, 2014, 43(6): 874-880. (in Chinese)
敖永才, 师奕兵, 张伟, 等. 自适应惯性权重的改进粒子群算法[J]. 电子科技大学学报, 2014, 43(6): 874-880.
- [15] CAI Y, LI G Y, WANG H. Research and implementation of parallel particle swarm optimization based on CUDA[J]. Application Research of Computers, 2013, 30(8): 2415-2418. (in Chinese)
蔡勇, 李光耀, 王琥. 基于 CUDA 的并行粒子群优化算法的设计与实现[J]. 计算机应用研究, 2013, 30(8): 2415-2418.