

李强, 刘晓峰. 一种基于IPGSA的多处理机调度算法[J]. 湖南科技大学学报(自然科学版), 2021, 36(1):99-105. doi:10.13582/j.cnki.1672-9102.2021.01.015

LI Q, LIU X F. A Job Scheduling Algorithm for Multiprocessor Based on IPGSA [J]. Journal of Hunan University of Science and Technology (Natural Science Edition), 2021, 36(1):99-105. doi:10.13582/j.cnki.1672-9102.2021.01.015

一种基于IPGSA的多处理机调度算法

李强^{1,2}, 刘晓峰^{3*}

(1. 太原理工大学 财经学院, 山西 太原 030024; 2. 山西省财政税务专科学校 信息学院, 山西 太原 030024;
3. 太原理工大学 大数据学院, 山西 太原 030024)

摘要: 多处理机作业调度问题是一个典型的非线性规划问题, 针对具有多条件限制的多处理机作业调度问题, 提出了一种基于改进的植物生长模拟算法(IPGSA)来搜索问题解的空间. 该方法首先将多条件限制和目标函数定义为该问题的数学模型, 然后使用改进的植物生长模拟算法搜索问题的解空间, 得出最佳的问题求解方案. 在实验中, 将该方法与粒子群算法、蚁群方法和HNN算法进行实验比较, 并给出具体算例测试并分析. 实验证明, 该方法具有比其他典型算法具有效率更高、更稳定的效果.

关键词: 植物生长模拟算法; 多处理机; 作业调度; 资源约束; 负载均衡

中图分类号: N94; TP391

文献标志码: A

文章编号: 1672-9102(2021)01-0099-07

A Job Scheduling Algorithm for Multiprocessor Based on IPGSA

LI Qiang^{1,2}, LIU Xiaofeng³

(1. College of Finance & Economics, Tai Yuan University of Technology, Taiyuan 030024, China;
2. College of Information, Shan Xi Finance & Taxation College, Taiyuan 030024, China;
3. College of Data Science, Tai Yuan University of Technology, Taiyuan 030024, China)

Abstract: Multiprocessor scheduling problem is a typical nonlinear programming problem, and an improved plant growth simulation algorithm (IPGSA) was proposed to solve the multiprocessor scheduling problem with multiple constraints. A mathematical model was defined by multiple constraints and objective function, and an improved plant growth simulation algorithm was used to search the problem space in order to obtain the best solution. In the experiment, the method was compared with particle swarm optimization, ant colony algorithm and HNN algorithm. Experimental results shown that the algorithm is more efficient and stable than the other typical algorithms.

Keywords: plant growth simulation algorithm; multiprocessor; job scheduling; resource constraints; load balancing

多处理机作业调度问题类似于Job-Shop问题, 属于NP难题. 伴随着近些年人工智能算法的进步, 蚁群算法、粒子群算法等在NP问题中都有了突出的表现, 为NP问题的解决提供了新的思路.

此外, 已经有大量的文献表明可以使用神经网络方法来解决车间作业调度的问题^[1-4]. 其中, 一个典

收稿日期: 2018-03-14

基金项目: 国家自然科学基金资助项目(61502330); 山西省高等学校科技创新项目资助(2020L0743); 山西省高等学校科技创新项目资助(20171119); 山西省软科学计划研究项目资助(2016041008-5); 校级课题(2020-Y27)

* 通信作者, E-mail: 122824940@qq.com

型的方法是使用 Hopfield 神经网络来实现作业的调度^[4],同时该方法也被成功地应用在多处理器作业调度中^[5].但是,使用该方法需要将问题定义为 hopfield 能量函数,虽然可以得到比较满意的结果,但是,数学描述复杂,运算量过大,在应用中受到限制.

使用植物生长模拟算法已经在车辆调度问题中表现出了其优越性^[6],并且在非线性整数规划问题中得到了证明该方法比蚁群算法、PSO 算法性能更高、更加稳定有效^[7-8].此外,该算法在物流体系选址、码头插船调度优化、GPU 调度优化、包装件配送和智能电网分析等领域中也得到了成功应用^[9-13].

多处理机作业调度问题实质是一个多条件限制的任务调度问题,往往受到时间、资源和环境各种条件的制约.本文首先通过问题的描述和分析,建立多条件限制的目标函数模型.然后,描述植物生长算法并采用一种可变生长长度的植物模型进行算法改进.最后,通过实验和算例来证明和分析该方法的有效性.

1 问题定义

多处理机作业调度问题可以定义为有 K 个处理机, M 个可以调度的作业,且需要满足若干约束条件.

定义 1 目标函数定义为满足所有作业在任意处理机上的总执行时间最短,可按式(1)表示为

$$f = \min(\max_T \sum_{i=1}^M \sum_{j=1}^K \sum_{t=1}^T y_{ijt});$$

$$y_{ijt} = \begin{cases} 1 \\ 0 \end{cases}.$$
(1)

式中: T 为最大周期数; y_{ijt} 记为作业 i 在 t 周期内运行于处理机 j 上的状态,若执行该值为 1,否则为 0.该公式表达了在最长时间周期下,各个处理和作业的所有组合中寻求最小的代价.

约束 1 每个处理机上任意时刻至多有一个作业在执行,其描述为

$$g_1 = \sum_{i=1}^K \sum_{j=1}^M \sum_{h=1}^T \sum_{\substack{l=1 \\ l \neq i}}^K y_{ijh} y_{ljh} = 0.$$
(2)

式(2)保证了任意一个节点在任意一个周期内不会同时执行 2 个作业.

约束 2 每个作业只能在一台机器上执行,可以表达为

$$g_2 = \sum_{i=1}^K \sum_{j=1}^M \sum_{h=1}^T \sum_{\substack{l=1 \\ l \neq i}}^K \sum_{m=1}^M y_{ijh} y_{ijm} = 0.$$
(3)

式(3)保证第 j 个机器节点不会出现在任意周期时间内上执行 2 个不同的作业.

约束 3 系统执行期间,不会有机器存在空闲周期,记为

$$g_3 = \sum_{j=1}^M \sum_{h=1}^T \sum_{i=1}^K (y_{ijh} - 1)^2 = 0.$$
(4)

式(4)中的平方项保证该值的非负性,整体保证约束为 0,即需要所有的平方项都为 0,也就需要每个机器任意时刻都为 1,即不会有空闲的机器.

约束 4 每个作业有一个完成的期限,表示为

$$g_4 = \sum_{i=1}^K \sum_{j=1}^M \sum_{h=1}^T y_{ijh} Q(w_{ijh}) = 0;$$

$$w_{ijh} = \frac{k}{p_i};$$

$$Q(w_{ijh}) = \begin{cases} 0, & \text{if } w_{ijh} \leq 1; \\ 1, & \text{if } w_{ijh} > 1. \end{cases}$$
(5)

式中: p_i 为每个作业都有个最长时间限制; K 为作业数; w_{ijh} 为作业完成比例; $Q(w_{ijh})$ 用来判断当前作业的完成情况,当完成比例小于等于 1 时,其值为 0,否则,该值为 1.约束条件 g_4 保证了不会有执行周期超长的作业出现,或强迫超长的作业停止执行.

约束 5 每个共享的资源在一个机器周期内只能为一个作业服务,可描述为

$$g_5 = \sum_{n=1}^F \sum_{h=1}^T \sum_{i=1}^K \sum_{j=1}^M \sum_{l=1}^K \sum_{m=1}^M (y_{ijh} S_{ijhn}) (y_{lmh} S_{lmhn}) = 0. \quad (6)$$

式中: S_{ijhn} 为资源 n 在第 h 个周期被运行在第 j 个处理机上的第 i 作业所占用的状态.若占用则该值为 1,否则为 0.

式(6)通过任意 2 个机器节点对相同的资源产生互斥来保证共享资源在任意周期内只能为一个作业占有.

2 算法描述

植物生长模拟算法,即 PGSA,是学术界在近些年仿生算法中的一个新宠,特别是在解决 NP 问题中,由于其表现比其他典型算法性能更高、更稳定而得到学者们的关注^[6].

本文将求解的多约束条件极值问题转化为植物的生长环境和动力,根据植物自然生长规律,通过算法迭代来模拟植物的自然生长过程求解问题.

2.1 算法描述

植物的生长过程可以简化:首先,由植物的种子开始发芽变成植物的根,再由植物的根部发育出植物的茎;然后,茎上再长出新芽;最后这些茎上再长出新的枝,枝上再反复长出新枝.这些能发育成新芽的位置被统称为“生长点”.

根据植物学知识,这些“生长点”并不一定都能长出新的枝,而是与生长点的生长激素的水平有关,其浓度越高越会发育成新枝.植物学基础理论证明,该生长激素的浓度与植物的生长条件有联系,而阳光是其中影响其发育的一种最主要的条件^[6].

这里使用随机变量 P_i 标记第 i 个生长点的生长激素水平,该值属于 0~1.对单个形态素的浓度可表示为

$$P_{Xi} = \frac{f(0) - f(S_{Xi})}{\sum_{i=1}^m (f(0) - f(S_{Xi})) + \sum_{j=1}^n (f(0) - f(S_{Yj}))}; \quad (7)$$

$$P_{Yi} = \frac{f(0) - f(S_{mi})}{\sum_{i=1}^m (f(0) - f(S_{Xi})) + \sum_{j=1}^n (f(0) - f(S_{Yj}))}; \quad (8)$$

$$\sum_{i=1}^m P_{Xi} + \sum_{j=1}^n P_{Yj} = 1. \quad (9)$$

式(7)和式(8)中: $f(*)$ 为目标函数; $f(0)$ 为原始生长点,即初始生长点.在迭代过程中,树干和树枝分别发育出新的生长点,并且均匀分布于树干中,该 2 个生长点集合分别是 $S_X = \{S_{X1}, S_{X2}, \dots, S_{Xm}\}$ 和 $S_Y = \{S_{Y1}, S_{Y2}, \dots, S_{Yn}\}$;该 2 个集合内的生长点($m+n$ 个)所对应的生长浓度概率集合分别为 $P_X = \{P_{X1}, P_{X2}, \dots, P_{Xm}\}$ 和 $P_Y = \{P_{Y1}, P_{Y2}, \dots, P_{Yn}\}$,并且该生长点的概率满足关系式(9),即每次迭代中所有生长点的浓度概率之和为 1.

式(7)中,构造了每个树干上生长点的生长概率.该式分母是所有生长点的目标函数与原始点目标函数 $f(0)$ 差的和,分子是原始点目标函数 $f(0)$ 与个别生长点的目标函数 $f(X)$ 之差.式(8)同理,构造了每个树枝上的生长点的生长概率.式(9)保持所有生长点的生长概率之和为 1.

每个生长点的生长激素浓度(生长概率)由植物的向光性条件所决定^[7],将该概率值简化为图 1 所描述的概率空间,也就是圆盘中的扇区大小代表不同概率值,所有扇区面积之和为 1,即所有生长点的概率值之和为 1.这里将植物的生长点的概率模拟为小球落入圆盘扇

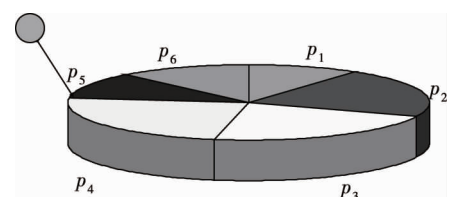


图 1 概率空间

区的过程,也就是盘上的小球落到哪个扇区获得相应的生长概率.每次循环中,获得概率最大的那个生长点长出单位新枝,并重新计算新的生长点集合,反复迭代此过程,直到找到答案或达到迭代上限,算法结束.

2.2 算法改进

该算法的实质是将组合优化问题转变成植物生长空间的搜索问题.以上算法每次迭代单位生长长度,并且每次仅有一个生长点发育成枝,收敛速度过慢.原算法的生长方式与实际植物生长情况差别较大,本研究中我们将在两方面改进:一是采用 θ 参数控制生长激素浓度的水平,即 θ 是一个阈值,当多个生长点的浓度超过该值将会同时发育;二是采用植物动力生长函数诱导植物生长,使植物的生长符合 Logistic 函数生长规律,如图 2 所示,其生长过程为逐渐加快生长至拐点,经拐点后开始生长缓慢,其最终生长形态如图 3 所示.

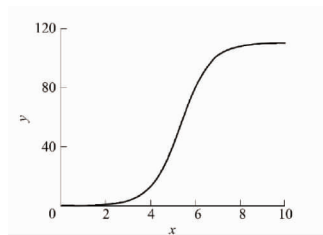


图 2 Logistic 函数

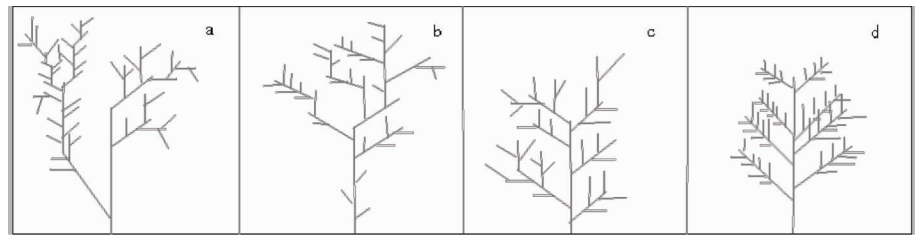


图 3 算法仿真效果

依据文献[8]和文献[14],植物的生长因素可以简化为由阳光积热模型所决定,这里采用阳光诱导的植物的 Logistic 函数来确定植物生长的过程.

$$L = \frac{\alpha}{1 + e^{\beta - \gamma \int_0^t \varphi(x) dx}}; \tag{10}$$

$$t_0 = \frac{\beta - \ln \frac{1}{2}}{\gamma} \alpha. \tag{11}$$

式中: α, β 和 γ 为生长系数,该 3 个参数可以控制生长的形态,其图形的拐点可由式(11)确定; $\varphi(x)$ 为生长作用函数.

$$\varphi(x) = 1\ 638\eta_t. \tag{12}$$

式中:1 638 为太阳辐射常数; η_t 为辐射利用系数,随时间变化,这里用来简化式(10).

$$V_t = \frac{\Delta L}{\Delta t} \approx \frac{\alpha \gamma \varphi(x) \exp[\beta - \gamma \int_0^t \varphi(x) dx]}{\{1 + \exp[\beta - \gamma \int_0^t \varphi(x) dx]\}^2} = \frac{1\ 638\alpha\gamma\eta_t \exp(\beta - \gamma 1\ 638\eta_t t)}{[1 + \exp(\beta - \gamma 1\ 638\eta_t)]^2}. \tag{13}$$

对式(10)求导,求解单位时间生长长度,可以得到式(13). V_t 是植物单位时间的生长长度,该变量是时间的函数,随时间而变化.

2.3 算法步骤

本文的多处理机调度程序步骤描述:

算法名称 Multiprocessor_Scheduling_IPGSA

输入:共享资源 $R_{ij} = \{r_{ij}\}$, 处理机的数量 K ,作业数 M ,单个作业的执行期限 p_i ,循环次数 N ,阈值 θ .

输出:路径 PATH.

- 1) 初始化,确定初始点 S_0 ,并将其添加到集合 S 中(初始点采用队列起始点作为 S_0).
- 2) 确定 $m+n$ 个新的生长点,并根据式(7)~式(9)计算每个生长点的生长概率.
- 3) 确定概率大于 θ 的生长点,开始同时按式(13)确定生长长度进行发育,并调整生长的方向,按 45° 调整,将新枝添加到树枝集合 S 中.

4) 判断新枝集合构成的路径是否最短,若最短将更新路径。(判断最短路径的方法即将新枝集合构成的路径与原有路径进行比较,如果新枝集合构成的路径比原有路径短,则更新路径)

5) 判断是否满足结束条件,若不满足则转入第 2 步.

6) 输出搜索的路径,程序结束.

该算法步骤如图 4 所示.

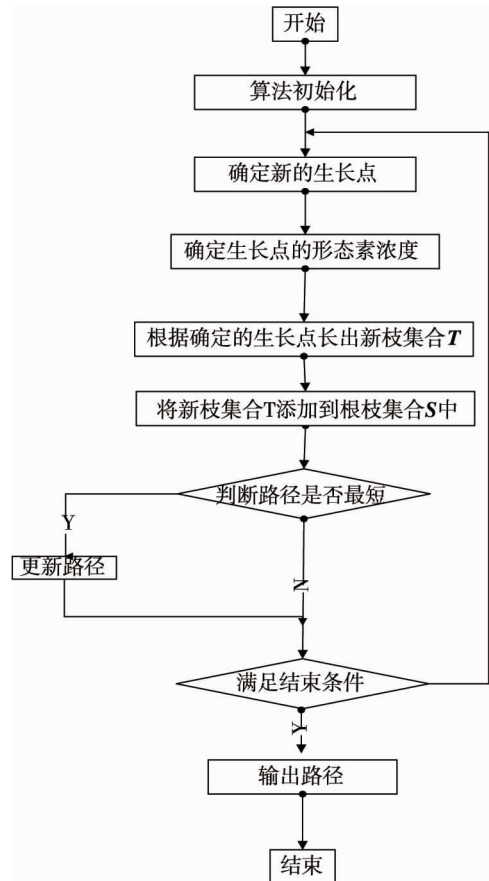


图 4 算法步骤

3 实验

3.1 实验条件

该实验中我们采用 3 台联想 i5 处理器进行实验,并用 VC++ 语言实现该算法.

实验中首先使用 7 个作业进行实验过程分析.然后,再采用 Pi Estimator 用例进行测试比较不同算法的性能.

本实验中采用大针茅植物模型^[15],即配置实验参数 $\alpha = 1.205$, $\beta = 2.608$ 和 $\gamma = 0.042$;并设置生长阈值 $\theta = 0.25$.

3.2 算例与分析

本文表 1 中提供的数据作为资源约束和执行周期.其中,资源约束矩阵 1 表示作业 J 占用资源 R .7 个作业分别被调度到 3 台处理器上执行,调度情况如图 5 所示.

3.3 结果分析

如表 2 所示,经过实验测试该推荐算法 IPGSA、粒子群算法、蚁群算法和 HNN 算法执行时间分别为 193,334,377 和 531 s.该算法比粒子群算法、蚁群算法和 HNN 算法的性能分别提高 42.2%,48.8% 和 63.7%,说明该算法比其他典型算法具有更高的执行效率.

表1 资源约束

作业	资源约束				执行周期 T
	R_1	R_2	R_3	R_4	
J_1	1	0	0	1	2
J_2	0	0	1	0	5
J_3	1	0	0	0	3
J_4	0	1	0	0	4
J_5	1	0	1	0	2
J_6	0	1	0	0	1
J_7	0	1	0	1	2

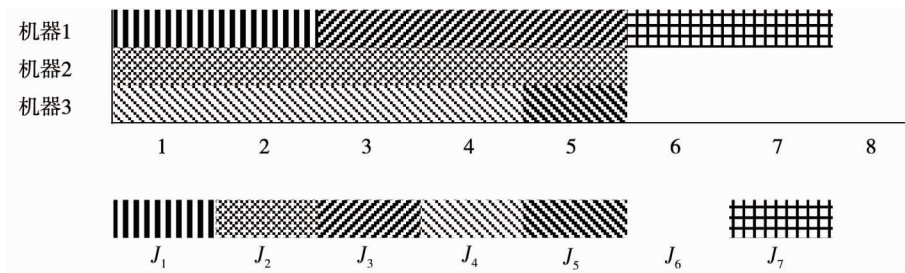


图5 作业调度过程

表2 算法运行时间比较

算法	IPGSA	粒子群算法	蚁群算法	HNN 算法
参数设置	$\alpha=1.205$	$\beta_1=0.625$	$\alpha=0.273$	$a_1=3.1$
	$\beta=2.608$	$\beta_2=0.03$	$\beta=0.318$	$a_2=3.3$
	$\gamma=0.042$		$\rho=0.117$	$a_3=1.3$
				$a_4=1.3$
				$a_5=5.3$
执行时间/s	193	334	377	531

从图6中,可以观测到3个处理器执行的情况,其中CPU的执行效率平均都在80%上下,内存占用率在82%左右,I/O利用率和网络利用率也均匀分布在37%~43%和31%~35%,3个节点的4个性能指标接近,说明该算法可以实现节点之间的负载均衡。

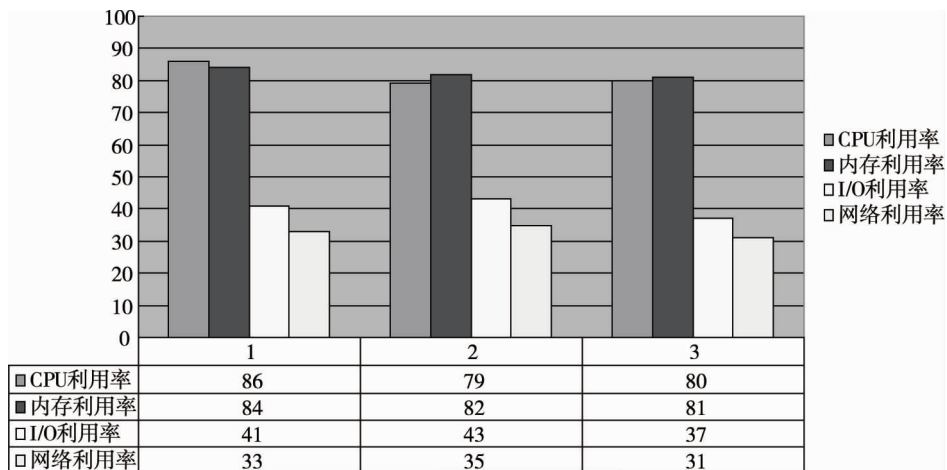


图6 节点性能

4个算法的主要性能的差异(包括,CPU、内存、I/O和网络4个方面的平均利用率)可以通过图7观测到.在HNN算法中,4个方面的主要指标都达到了85%,说明该种算法的计算量和调度量比较大,据分析是由于

HNN 算法本身的复杂度所导致.在蚁群算法中,I/O 和网络开销比较突出,说明该算法大量的工作花销在调度上,计算任务并不分布均匀.而粒子群算法正好相反,I/O 和网络开销较小,说明大量的计算花费在本地执行.IPGSA 的 4 个方面都平均分布在 75%左右,说明任务的分布比较均匀,系统的整体负载均衡.

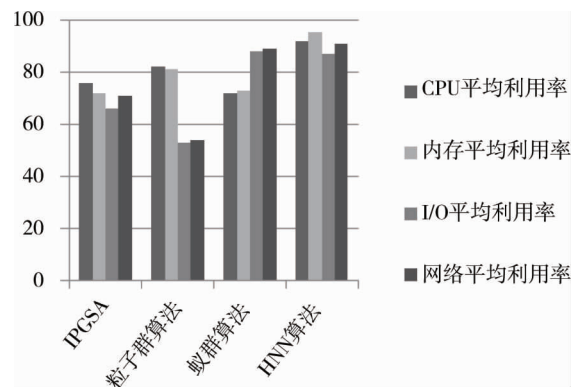


图 7 算法性能的比较

4 结论

该文针对多处理机作业调度问题,建立一种基于 Logistic 函数来诱导植物生长,提出一种更符合植物生长规律的植物生长模拟算法,改变了传统的单步生长的植物生长模拟算法不符合现实生长的弊端.该模型通过使用大茅针模型,配置实验参数 α , β 和 γ 进行试验.实验结果证明该方法在性能上分别比粒子群算法、蚁群算法和 HNN 算法提高 42.2%, 48.8% 和 63.7%,具有效率更高、可靠性更好的优势.

除了大茅针植物模型,还有冰草等不同植物生长模型.不同模型对植物的生长空间具有巨大的影响,因此不同植物模型以及起点的确定对算法的影响将是我们下一个研究的目标.

参考文献:

- [1] Huang Y M, Chen R M. Scheduling multiprocessor job with resource and timing constraints using neural networks [J]. IEEE Transactions on Systems Man and Cybernetics, 1999, 29(4): 490-502.
- [2] Zhang C S, Yan P F, Chang T. Solving job-shop scheduling problem with priority using neural network [C]//IEEE International Conference on Neural Networks Conference Proceedings, 1991, 14(2/3): 1361-1366.
- [3] Yang S, Wang D. Constraint satisfaction adaptive neural network and heuristic combined approaches for generalized job-shop scheduling [J]. IEEE Transactions on Neural Networks, 2011(2): 474-86.
- [4] 王万良, 吴启迪, 徐新黎. 基于 Hopfield 神经网络的作业车间生产调度方法[J]. 自动化学报, 2002(5): 838-844.
- [5] 王秀丽, 吴惕华. 一种求解多处理器作业调度的 Hopfield 神经网络方法[J]. 系统工程与电子技术, 2002(8): 13-16.
- [6] 曹庆奎, 刘新雨, 任向阳. 基于模拟植物生长算法的车辆调度问题[J]. 系统工程理论与实践, 2015, 35(6): 1449-1456.
- [7] 李彤, 王春峰, 王文波, 等. 求解整数规划的一种仿生类全局优化算法——模拟植物生长算法[J]. 系统工程理论与实践, 2005(1): 76-85.
- [8] 王莉, 秦勇, 徐杰, 等. 植物多向生长模拟算法[J]. 系统工程理论与实践, 2014, 34(4): 1018-1027.
- [9] 吴迪, 王诺, 宋南奇, 等. 边远群岛物流体系的选址-库存-路径优化[J]. 系统工程理论与实践, 2016, 36(12): 3175-3187.
- [10] 王诺, 沈铭棋, 刘忠波. 基于改进模拟植物生长算法的集装箱码头插船调度优化[J]. 运筹与管理, 2016, 25(6): 75-82.
- [11] 杨磊, 龚学余. 基于 GPU 的并行植物模拟生长算法[J]. 计算机工程与设计, 2016, 37(12): 3271-3279.
- [12] 樊贵香. 混合模拟植物生长算法在包装件配送中的应用[J]. 包装工程, 2016, 37(13): 43-49.
- [13] 余光正, 林涛, 陈汝斯, 等. 考虑智能配电网谐波谐振模式不确定性的抑制措施[J]. 电力自动化设备, 2016, 36(7): 60-66.
- [14] 李彤, 王众托. 大型城市地下物流网络优化布局的模拟植物生长算法[J]. 系统工程理论与实践, 2013, 33(4): 971-980.
- [15] 张彩琴, 杨持. 内蒙古典型草原生长季内不同植物生长动态的模拟[J]. 生态学报, 2007(9): 3618-3629.