[自控·检测]

DOI:10.3969/j.issn.1005-2895.2019.04.012

基于混合蚁群粒子群算法的立体库堆垛机路径优化

陈 晨,茅 健

(上海工程技术大学 机械与汽车工程学院,上海 201620)

摘 要:针对自动化立体仓库中的堆垛机路径优化问题,课题组通过分析立体仓库中堆垛机的工作特点与运行情况,提出了基于混合蚁群粒子群算法的路径优化方法,在传统的蚁群算法中结合粒子群算法思想,使算法同时具备蚁群算法的正反馈与粒子群的多样性。通过对实例进行MATLAB仿真分析表明:混合算法路径优化速度较快,且比以往的路径更短。研究使堆垛机的运行效率得到提高。

关 键 词:智能物流;堆垛机;蚁群算法;粒子群算法;路径优化

中图分类号:TH692.3;TP273 **文献标志码**:A **文章编号**:1005-2895(2019)04-0063-05

Path Optimization of Stereoscopic Warehouse Stacker Based on Mixed Ant Colony Particle Swarm Algorithm

CHEN Chen, MAO Jian

(School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: In order to solve the problem of crane path optimization in automated warehouse, research project's group first analyzed the working characteristics of the crane in warehouse and the situation of the path, a path optimization method based on hybrid ant colony particle swarm optimization was proposed. In the traditional ant colony algorithm, particle swarm algorithm thought was combined to make the algorithm have a combination of positive feedback of ant colony algorithm and particle swarm diversity. Through the simulation, the algorithm path optimization speed becomes faster, and the path is shorter than ever before. This research improves the operation efficiency of stacker.

Keywords: intelligent logistics; stacker; ant colony algorithms; particle swarm algorithm; path optimization

近年来,工业4.0战略规划促使整个工业生产体系进入全新的水平[1]。智能物流作为工业4.0的核心组成部分,是连接客户、供应链和制造业的重要环节,是未来智能工厂的重中之重。堆垛机是智能物流中重要设备之一,其路径的好坏决定了仓库的运作效率,因此对堆垛机的路径规划也成为了研究热点,国内外学者已经做出了很多贡献[2]。李巧玲[3]通过改进蚁群算法的路径选择策略和信息素更新策略,优化了复杂制造系统的物流配送路径;卫三军等[4]通过引入模拟退火算法对蚁群算法的参数进行控制,获得较高效率的运行路径;庞龙等[5]通过利用遗传算法的全局搜索能力,克服了单一蚁群算法易陷入局部最优解的缺点,对堆垛机路径优化提出了改进;李擎等[6]提出了一种全局异步和精英策略相结合的信息素更新方式,提高了实际运行效率;夏辉等[7]根据微粒群的思想,优化了蚁群算法中的参数,从而优化了堆垛机路径;Enagls C等[8]基于两元素优化法对路径解进行重组,以寻求更优解;Blum C等[9]对遗传算法与蚁群算法进行融合,通过更新全局最优解来相互指导,提高了系统的运行效率。

课题组根据立体库的实际情况,基于混合蚁群粒子群算法,研究了堆垛机运行情况,建立了堆垛机路径

收稿日期:2018-10-07;**修回日期**:2019-02-15

基金项目:上海工程技术大学研究生科研创新项目(E3-0903-18-01008)。

第一作者简介:陈晨(1995),男,安徽枞阳人,硕士研究生,主要研究方向为智能制造。E-mail:2976738906@qq.com

优化模型并进行仿真。结果表明混合蚁群粒子群算法较单一蚁群算法,可以提高堆垛机的工作效率。

1 问题概述与建模

堆垛机未工作时,与货架左下角货位相距 1 个库位的距离。用 V_x 表示堆垛机的左右运动速度,用 V_y 表示其上下运动速度。在实际生产中,一般是批次取货。批次取货的作业原理:堆垛机从原点运行到第 1 个指定货位取货回原点,取出货物再沿原路返回原货格,将托盘送回原货格,放回托盘后不返回原点,再继续到下一个指定货格取货回原点,如此反复直到取完所有的货物为止[10]。现以 3 条指令为例,说明堆垛机批次作业流程,其简化执行路线如图 1 所示。

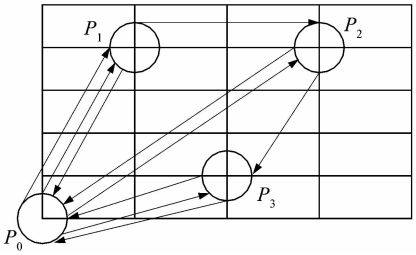

图 1 3 条指令的复合方式作业示意图

Figure 1 Composite operation schematic diagram of 3 orders

当堆垛机执行 1,2,3,…,n 条指令时,时间

$$T = \sum_{i=1}^{n}(T_{0,i} + T_{i,0}) + (\sum_{i=1}^{n} T_{i-1,i} + T_{n,0})。 \quad (1)$$

式中:T 为执行 n 条指令堆垛机运行的时间;$T_{0,i}$ 为从起始点 P_0 到 P_i(任务终点)所用的时间;$T_{i,0}$ 为堆垛机从点 P_i(任务终点)到起始点 P_0 所用的时间;$T_{i-1,i}$ 为从点 P_{i-1}(现在所处货位)到 P_i(下一指令货位)所用时间;$T_{n,0}$ 为堆垛机完成此批次任务后回到原点 P_0 所用的时间。

不难发现,当一批出入库任务指定后,不管任务执行的先后顺序如何,$\sum_{i=1}^{n}(T_{0,i} + T_{i,0})$ 的值总是确定的,堆垛机的运行时间不受其影响,此部分运行情况如图 2 所示,记为第Ⅰ部分运行情况;$(\sum_{i=1}^{n} T_{i-1,i} + T_{n,0})$ 为一个从原点出发途经各指定库位的回路,堆垛机采用不同的作业顺序,路径长度是不同的,执行的时间也不相同,此部分运行情况如图 3 所示,记为第Ⅱ部分运行情况。第Ⅱ部分堆垛机所走的路径就是一个从原点出发途经各指定库位再回到原点的回路,这是一个典型的旅行商(Travelling Salesman Problem)问题,即 TSP 问题[11]。

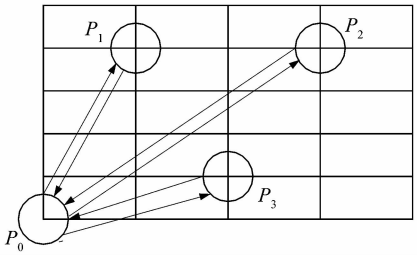

图 2 3 条指令第Ⅰ部分运行情况示意图

Figure 2 Part Ⅰ schematic diagram of 3 orders

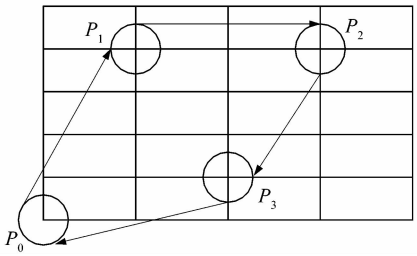

图 3 3 条指令第Ⅱ部分运行情况示意图

Figure 3 Part Ⅱ schematic diagram of 3 orders

2 混合蚁群粒子群算法

2.1 蚁群算法

蚁群算法(Ant Colony Optimization, ACO)是由意大利学者 Dorigo M 提出的一种模仿蚂蚁觅食的新兴群智能算法。该方法利用蚂蚁在路上留下的信息素含量来判断路径的优化程度,具有正反馈与分布式计算的特点。

将蚁群算法引入 TSP 问题的思路:设整个蚁群中蚂蚁的数量为 m,需要遍历 n 个地点,地点 i 与 j 之间的距离为 $d_{ij}(i,j=1,2,\cdots,n)$,t 时刻地点 i 与 j 连接路径上的信息素浓度为 $\tau_{ij}(t)$。各个地点之间连接路径上的信息素浓度在算法初始时是相同的,即 $\tau_{ij}(0)=\tau_0$。

蚂蚁 $k(k=1,2,\cdots,m)$ 根据各个地点间连接路径上的信息素浓度决定其下一个到达的库位,设 P_{ij}^k 表示 t 时刻蚂蚁 k 从库位 i 转移到库位 j 的概率,且有

$$P_{ij}^k = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s\in A_k}[\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & s \in A_k; \\ 0, & s \notin A_k。 \end{cases} \quad (2)$$

式中:$\eta_{ij}(t)$ 为启发函数,$\eta_{ij}(t)=1/d_{ij}$,表示蚂蚁从地点 i 到达地点 j 的概率;$A_k(k=1,2,\cdots,m)$ 为蚂蚁 k 待访问地点的集合,开始时,A_k 中有 $(n-1)$ 个元素,当 A_k 中的元素减少为空集时,即表示遍历全部地点;α 为信息素重要程度因子,其值越大,信息素的浓度在转移中起到的作用越大;β 为启发函数重要程度因子,其值与

启发函数在转移中的作用成正比,即相比距离长的地点,蚂蚁更倾向于转移到距离短的地点。

需要注意的是,各个地点间连接路径上的信息素随着时间流逝而挥发,信息素的挥发程度用参数 $\rho(0<\rho<1)$ 表示。因此,需要实时更新连接路径上的信息素浓度。

$$\left.\begin{aligned}\tau_{ij}(t+1) &= (1-\rho)\tau_{ij}(t)+\Delta\tau_{ij};\\ \Delta\tau_{ij} &= \sum_{k=1}^{n}\Delta\tau_{ij}^{k}。\end{aligned}\right\} \quad (3)$$

式中,$\Delta\tau_{ij}^{k}$ 表示第 k 只蚂蚁在地点 i 与 j 连接路径上释放的信息素浓度;$\Delta\tau_{ij}$ 表示全部蚂蚁在2点之间产生的信息素浓度总量。

2.2　粒子群算法

粒子群算法(Particle Swarm Optimization,PSO)是一种基于鸟群智能觅食的随机优化技术,也叫做鸟群算法,该算法具有群体智能、内在并行性、迭代格式简单和可快速收敛到最优解等优点。粒子群算法的优势在于可以较为简便地求出多个粒子共存或合作时的最优解。其优化思想在于对于每个粒子都有自己的速度与位置,粒子们在自己速度影响的同时也追随当前最优的粒子,从而迭代得到最优值。其中第 i 个粒子表示为一个 D 维的向量 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$,代表第 i 个粒子在 D 维空间中的位置,亦代表问题的一个潜在解。定义单个粒子在飞行过程中所获取的最优解为个体最优解(p_{Best}),在整个粒子群所获得的最优解记作全局最优解(g_{Best}),用 N 维速度 $V_i=(v_{i1},v_{i2},\cdots,v_{iN})$ 与位置 $P_i=(p_{i1},p_{i2},\cdots,p_{iN})$ 表示粒子状态,在算法中不断地通过自身速度与位置进行状态更新,产生新一代群体。

$$V_{id}^{k+1}=wV_{id}^{k}+c_1r_1(P_{id}^{k}-X_{id}^{k})+c_2r_2(P_{gd}^{k}-X_{id}^{k}); \quad (4)$$

$$P_{id\mathrm{new}}^{k}=P_{id}^{k}+V_{id}^{k}。 \quad (5)$$

式中:w 为惯性权重;$d=1,2,\cdots,D$;$i=1,2,\cdots,n$;k 为当前迭代次数;公式(4)中 V_{id}^{k+1} 为粒子速度更新值;公式(5)中 $P_{id\mathrm{newk}}$ 表示粒子新的位置;c_1 和 c_2 为加速度因子;r_1 和 r_2 是分布于[0,1]区间的随机数。

粒子群算法的具体实现流程:①粒子群初始化,将粒子群中各个参数设置好;②在适应度函数的基础上,将每一个粒子适应度值表示出来;③将粒子的当前适应度值与历史最优适应度值比较,更新历史最优值;④将当前适应度与种群历史最优位置适应度值比较,更新历史最优值;⑤运用式(4)与式(5)进行计算;⑥若获得最优值,则输出最优值结果,否则跳转②继续进行迭代。

2.3　混合蚁群粒子群算法

蚁群算法的优点在于采用正反馈机制,使得搜索过程不断收敛,最终逼近最优解;缺点在于传统的蚁群算法,收敛速度慢,易于陷于局部最优。粒子群的优点在于其全局搜索能力强,参数的更新比较容易;其缺点在于算法容易早熟收敛,局部寻优能力较差。

在衡量了2个算法的优缺点之后,混合算法首先由蚁群算法寻找初始解,然后结合粒子群算法,让蚂蚁在信息素与最优蚂蚁2者的作用下进行路径寻优。算法的主要实现步骤如下:

1)初始化蚁群算法与粒子群算法参数,让蚁群随机分布在求解空间内;

2)由蚁群算法寻找初始解集,并记录较优路径数据;

3)记录蚂蚁的个体最优值 p_{best} 与全局最优值 g_{best};

4)在蚂蚁具有粒子性质的情况下,根据新种群中粒子适应度值更新个体最优值 p_{best} 与全局最优值 g_{best};

5)若迭代次数超过设定值则输出最优值,否则进入步骤4)继续进行运算。

在MATLAB模拟中设置蚁群算法中迭代次数为200,信息重要程度因子为1.0,启发函数重要程度因子为5.0,信息素蒸发系数为0.1,信息素增加强度系数为1.0。在粒子群算法中惯性权重取0.5,加速度因子 c_1 为1.5,c_2 为2.0。

混合算法流程如图4所示。

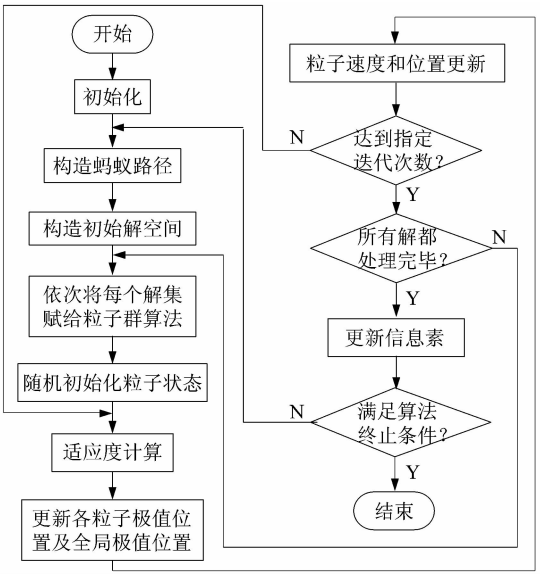

图4　混合算法流程

Figure 4　Flowchart of hybrid algorithm

3 实例分析

为了验证方法是否可行,选取某沿海公司仓库为例。货格规格为 500 mm × 500 mm,每个货架分为 10 层,20 列。堆垛机的速度为 V_x = 60 mm/s, V_y = 20 mm/s。货格编号如图 5 所示。

某批出入库任务如表 1 所示。

199	197	195	193	191	189	187	185	183	181	182	184	186	188	190	192	194	196	198	200
179	177	175	173	171	169	167	165	163	161	162	164	166	168	170	172	174	176	178	180
159	157	155	153	151	149	147	145	143	141	142	144	146	148	150	152	154	156	158	160
139	137	135	133	131	129	127	125	123	121	122	124	126	128	130	132	134	136	138	140
119	117	115	113	111	109	107	105	103	101	102	104	106	108	110	112	114	116	118	120
99	97	95	93	91	89	87	85	83	81	82	84	86	88	90	92	94	96	98	100
79	77	75	73	71	69	67	65	63	61	62	64	66	68	70	72	74	76	78	80
56	54	52	50	48	46	44	42	40	39	41	43	45	47	49	51	53	55	57	60
10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36	38	58
1	2	3	4	5	6	7	8	9	19	21	23	25	27	29	31	33	35	37	59

图 5 货格编号

Figure 5 Storage location number

表 1 任务表

Table 1 Table of task

序号	货格编号	序号	货格编号	序号	货格编号
1	4	11	65	21	135
2	19	12	76	22	124
3	33	13	97	23	149
4	10	14	93	24	143
5	26	15	82	25	156
6	56	16	92	26	164
7	48	17	109	27	197
8	39	18	105	28	185
9	47	19	112	29	190
10	67	20	139	30	138

分别用混合算法与蚁群算法去优化堆垛机路径,优化的结果如图 6 和图 7 所示。2 种算法在送货任务中得到了同样的优化路径,且均为最短路径。可以清晰的看到混合算法能得到一个较优的解,实际上由混合算法所得到的路径长度为 64.246 m,而单一蚁群算法所得到的路径长度为 70.337 m。

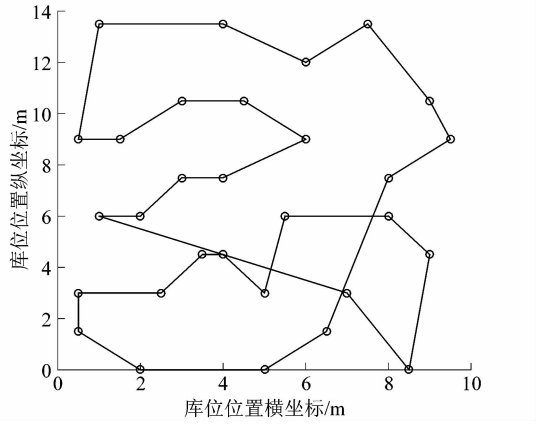

图 6 蚁群算法优化后路径

Figure 6 Optimized path by ant colony algorithm

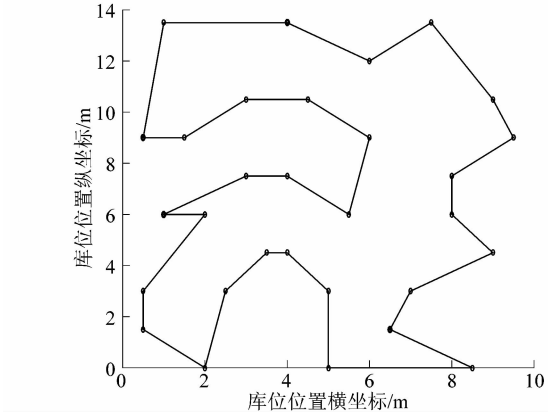

图 7 蚁群粒子群混合算法优化后路径

Figure 7 Optimized path by mixed ant colony particle swarm algorithm

图 8 为 2 种算法的收敛速度比较,可以清晰地看出,混合算法在收敛速度上优于单一蚁群算法。在实际应用中,可以使得堆垛机快速完成任务,从而提高仓储效率。

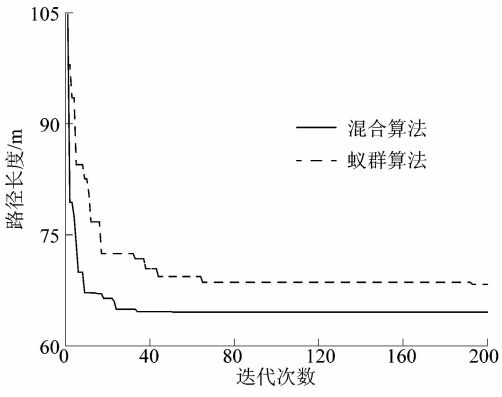

图 8 2 种算法收敛情况比较

Figure 8 Comparison of convergence of two algorithms

(下转第 72 页)