

改进型细菌觅食算法及多货叉仓库调度应用

杨文强, 邓丽, 牛群, 费敏锐

(上海大学 a. 机电工程与自动化学院, b. 上海市电站自动化技术重点实验室, 上海 200072)

摘要: 针对多货叉仓库调度优化问题, 提出一种改进型细菌觅食算法. 首先, 分阶段对趋化步长进行自适应调节, 引导搜索沿最优方向进行; 其次, 提出基于个体种群多样性贡献率的启发式迁移策略, 降低进入局部最优的机率; 再次, 采用不可行解部分保留策略以增加求出最优解的机会; 最后, 对该算法的收敛性进行证明, 并结合工业现场调度问题对其性能进行验证. 算例结果表明, 所提出的算法对多货叉仓库调度优化问题在解的质量及收敛速度上都取得了较好效果.

关键词: 多货叉仓库调度; 细菌觅食算法; 步长自适应调节; 种群多样性; 不可行解部分保留策略

中图分类号: TP278

文献标志码: A

Improved bacterial foraging algorithm and application of warehousing scheduling with multi-shuttles

YANG Wen-qiang, DENG Li, NIU Qun, FEI Min-rui

(a. School of Mechatronics Engineering and Automation, b. Shanghai Key Laboratory of Power Station Automation Technology, Shanghai University, Shanghai 200072, China. Correspondent: DENG Li, E-mail: dengli@shu.edu.cn)

Abstract: For the automated warehousing scheduling problem with multi-shuttles, an improved bacterial foraging algorithm is proposed. Firstly, the chemotactic stepsize is adaptively adjusted to guide search toward the optimum solution. Then, the heuristic elimination and dispersal strategy is devised based on the level of individual contributing rate to the diversity of population, which reduces the probability of falling into local optimum. The strategy of preserving some of infeasible solutions is adopted to increase the chances of obtaining the optimal solution. Finally, the convergence of the proposed algorithm is proved and the performance is tested through simulation combined with the industrial real world case. Results show that the proposed algorithm achieves better performance in terms of the solution quality and the convergence efficiency for the automated warehousing scheduling problem with multi-shuttles.

Keywords: warehousing scheduling with multi-shuttles; bacterial foraging algorithm; adaptive stepsize; diversity of the population; strategy of preserving some of infeasible solutions

0 引言

立体仓库调度是组合优化问题, 也是一种 NP 难问题. 国内外学者在这方面已开展了一些研究工作. 文献 [1] 为提高仓库作业效率, 对堆垛机的停靠点策略进行了研究, 并提出策略评估模型; 文献 [2] 提出带水平和垂直拣货台的拣选机制, 以提高拣选效率; 文献 [3] 用分支定界算法求解, 求解所用的时间与求解问题的规模成指数关系, 因而通用性有一定的局限性; 文献 [4] 以提高拣选效率为目标, 构建了基于专用存储策略货位分配模型, 并用模拟退火算法对其进行求

解; 文献 [5] 提出一种深度方向上的分类存储策略, 并与随机存储策略在调度性能上进行了比较; 文献 [6] 基于禁忌搜索算法对特定仓库调度优化问题开展了研究, 但求解效率有待改进; 文献 [7] 用蚁群算法求解仓库调度优化问题, 但解的精度仍有较大的提升空间. 以上文献关注较多的是单货叉仓库调度优化问题, 而从仓库调度系统高效性的角度, 调度系统的设计者应考虑多货叉仓库的调度问题. 与单货叉仓库相比, 多货叉仓库的优点在于: 在一个作业周期内, 可存取多个货物, 能有效缩短空载时间, 提高作业效率. 因而,

收稿日期: 2013-10-17; 修回日期: 2014-01-28.

基金项目: 国家自然科学基金项目(61074032, 61273040); 上海市青年科技启明星计划项目(12QA1401100); 上海市教委创新基金项目(12YZ020); 上海市科委重大基础研究项目(14JC1402200).

作者简介: 杨文强(1984—), 男, 博士生, 从事生产计划与调度优化、智能控制的研究; 费敏锐(1961—), 男, 教授, 博士生导师, 从事网络先进控制理论与系统实现等研究.

在当前节能减排的背景下,对其开展调度优化研究意义深远。

组合优化问题的求解复杂度随问题规模的增大而呈指数级增长,会产生组合爆炸,因而采用基于数学规划的精确算法很难在有限的时间内求出其最优解或次优解.智能优化算法相对于精确算法具有一定的智能性,能够引导种群向好的方向进化,因而在求解效率和质量上有很大的提高,但也存在早熟、收敛速度慢、参数敏感等问题。

Passino^[8]于2002年提出了细菌觅食优化算法(BFO),引起了不同领域研究者的关注,并得到了广泛应用^[9-14].然而,目前将BFO算法应用于仓库调度优化问题的研究还较少见.该算法固定的趋化步长降低了收敛效率,种群的单一性限制了解空间的搜索范围,使其易于陷入局部最优.针对这些不足,本文对BFO算法的趋化步长进行改进,使其能够根据寻优过程中的全局及局部信息进行自适应调节,以提高收敛速度;同时,采用种群多样性保持策略,扩大对解空间的搜索范围;然后,采用不可行解部分保留策略以增加求出最优解的机会;最后,通过仿真实验验证了本文算法的可行性。

1 问题描述及模型构建

1.1 问题描述

在某次出入库作业中,仓库调度问题具有如下特征:

- 1) 入库任务有 m 个,它们在仓库中的位置分别表示为 $p_{i1}, p_{i2}, \dots, p_{im}$;
- 2) 出库任务有 n 个,它们在仓库中的位置分别表示为 $p_{o1}, p_{o2}, \dots, p_{on}$.

考虑到现场的实际情况,在大部分情况下,某次出入库作业中的入库作业数量与出库作业数量是不相等的,即 $m \neq n$.同时,堆垛机有 N 个货叉,对于双命令周期的堆垛机在一个作业周期中可完成 $2N$ 个出入库任务.因此,对于有 m 个入库任务和 n 个出库任务的作业,组成 $m+n$ 个货位点,即 $p_1, p_2, \dots, p_m, p_{m+1}, \dots, p_{m+n}$,则需要堆垛机执行 $\max\{[m/N], [n/N]\}$ 次出入库任务.这里,堆垛机执行一次出入库任务是指堆垛机在出入库缓冲区以最大载货量装载要入库的货物,在入库的过程中,同时以最大载货量拣选要出库的货物并返回到出入库缓冲区.同时, $\max\{[m/N], [n/N]\}$ 中的 $[m/N]$ 、 $[n/N]$ 分别表示堆垛机在最大载货量的情况下,完成所有入库任务、出库任务需要堆垛执行的次数.堆垛机在执行一次出入库任务的过程中,既可以执行入库任务,又可以执行出库任务.因此,通过 \max 函数来取 $[m/N]$ 和

$[n/N]$ 两者中的最大值.仓库调度优化需解决的问题是从堆垛机所有可能的路径中,挑选出能完成所有出入库任务并且满足约束的最优 $\max\{[m/N], [n/N]\}$ 条路径。

在如图1所示的多巷道仓库系统中,为了方便研究问题,忽略堆垛机的启动和制动时间.设堆垛机的水平运动平均速度和垂直运动平均速度分别为 v_x 、 v_y ,并且水平和垂直方向的运动相互独立;仓库每个货位的长、宽、高分别记为 L 、 W 、 H ,巷道的宽度为 D_l ,每排货架的列数为 C ,并把货位坐标定义为 $p_i(x_i, y_i, z_i)$,3个坐标分量依次表示所在货架的列号、层号及巷道号,出入库缓冲区指定为 $p_0(0, 0, 0)$.

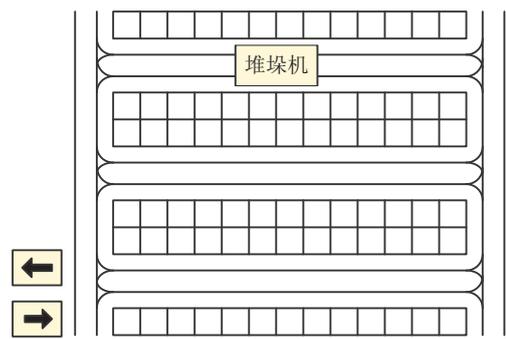


图1 多巷道仓库示意图

定义1 如果堆垛机在执行任务过程中经过路径 r ,则 $s_r = 1$;否则, $s_r = 0$.如果货位 p_i 属于子路径 r ,则 $g_{ir} = 1$;否则, $g_{ir} = 0$.

定义2 如果堆垛机在执行任务过程中连续经过货位 $p_i(x_i, y_i, z_i)$ 、 $p_j(x_j, y_j, z_j)$,则 $e_{ij} = 1$;否则, $e_{ij} = 0$.对于巷道两端均可进出的仓库系统,堆垛机所用时间 t_{ij} 可表示为

$$t_{ij} = \begin{cases} \max\{A_1, B\}, & z_i = z_j; \\ \max\{\min\{A_2, A_3\}, B\}, & z_i \neq z_j. \end{cases} \quad (1)$$

其中

$$A_1 = (W \times |x_i - x_j|)/v_x,$$

$$A_2 = (W \times (x_i + x_j) + D_l \times |z_i - z_j|)/v_x,$$

$$A_3 = (W \times ((C - x_i) + (C - x_j)) + D_l \times |z_i - z_j|)/v_x,$$

$$B = (H \times |y_i - y_j|)/v_y.$$

1.2 模型构建

该仓库调度问题的优化目标为堆垛机完成所有出入库任务所用的时间最小,其数学模型包括目标函数和约束,定义如下:

$$\min f(e) = \sum_{r=1}^{\max\{[m/N], [n/N]\}} \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} t_{ij} e_{ij} g_{ir} g_{jr}. \quad (2)$$

$$\text{s.t.} \quad \sum_{r=1}^{\max\{[m/N], [n/N]\}} s_r = \max\{[m/N], [n/N]\}; \quad (3)$$

$$\sum_{p_m \in r_s} \text{Is_Storage}(p_m) + \text{Empty}(r_s) \geq \sum_{p_n \in r_s} \text{Is_Retrieval}(p_n), \forall r_s \subseteq r; \quad (4)$$

$$\sum_{i=1}^{m+n} \sum_{r=1}^{\max\{[m/N], [n/N]\}} g_{ir} s_r = m + n; \quad (5)$$

$$\sum_{r=1}^{\max\{[m/N], [n/N]\}} g_{ir} = 1, \quad \forall i \in \{1, 2, \dots, m+n\}; \quad (6)$$

$$\sum_{i=1}^{m+n} g_{ir} \leq 2N, \quad \forall r \in \{1, 2, \dots, \max\{[m/N], [n/N]\}\}; \quad (7)$$

$$\sum_{j=1}^{m+n} e_{0j} = [m/N]; \quad (8)$$

$$\sum_{i=1}^{m+n} e_{i0} = [n/N]; \quad (9)$$

$$p_u \in p_{0j}, \forall r(p_i, p_j, \dots, p_u). \quad (10)$$

其中: $\text{Is_Storage}(\cdot)$ 、 $\text{Is_Retrieval}(\cdot)$ 分别为入、出库任务判别函数, 对子路径中的出入库任务属性进行判别. 如果当前判断的任务为入库任务, 则 $\text{Is_Storage}(\cdot)$ 的返回值为 1; 如果当前判断的任务为出库任务, 则 $\text{Is_Retrieval}(\cdot)$ 的返回值为 1. $\text{Empty}(\cdot)$ 为堆垛机经过当前路径后, 空闲货叉的数量.

在上述模型中, 式 (2) 是需要优化的目标; 式 (3) ~ (10) 为约束条件. 其中: 式 (3) 为完成所有任务需要堆垛机执行的次数; 式 (4) 为保证要出库的货物有空闲货叉可用; 式 (5) 为堆垛机需完成的任务量; 式 (6) 为一个任务只在子路径中出现一次; 式 (7) 为堆垛机的负载约束; 式 (8) 为堆垛机的入库起点约束; 式 (9) 为堆垛机的出库终点约束; 式 (10) 为每条路径中入库作业优先完成约束.

2 改进型细菌觅食算法 (IBFO)

鉴于可行解部分寻优信息的缺失、固定步长及进化过程中种群的单一性对细菌觅食算法性能的影响, 本文从以下几个方面对该算法进行改进.

2.1 部分不可行细菌保留策略

定义 3 对于仓库调度问题, 细菌个体 i 的位置可表示为 $\theta(i) = (\theta_1, \theta_2, \dots, \theta_n)$, 其中 θ_i ($i = 1, 2, \dots, n$) 表示一个待存储或检索的货位.

不可行解可能携带一些可行解无法获取的有用信息, 比如距离全局最优解较近, 能够快速搜索到全局最优解. 因而适当地保留一些适应度较高、违反约

束较少的不可行解, 对提高收敛速度会起到一定的作用, 定义 α ($0 < \alpha < 1$) 为不可行解在整个种群的保留比例. 具体策略为: 首先按违反约束的程度进行等级划分, 违反一个约束条件的不可行解为第 1 等级, 违反两个约束条件的不可行解为第 2 等级, 以此类推, 并将他们按等级升序进行排列; 然后按式 (2) 的倒数分别在每个等级中对不可行解进行适应度计算, 并在每个等级中按适应度降序对不可行解进行排序; 最后在经过排序后的不可行解集中保留比例为 α 较优的不可行解. 如果当前不可行解的数量小于所需的数量, 则从当前可行解中随机产生不可行解, 直到达到所要求的数量; 反之亦然.

2.2 自适应步长设计

定义 4 对于离散仓库调度问题, 为了方便研究问题, 采用整数任务号编码策略. 那么在不同位置的两个细菌个体 $\theta(i)$ 和 $\theta(j)$ ($\theta(i) \neq \theta(j)$), 细菌个体 i 要到达细菌个体 j 所在的位置 (记为 $\theta(i) \rightarrow \theta(j)$), 可以通过两两交换自身的基因位 θ_p 和 θ_q 来实现. 这种交换行为定义为 $s(u_p, u_q)$, 交换的次数定义为两个细菌个体之间的距离, 即

$$D(\theta(i), \theta(j)) = \text{card}(\{s(u_p, u_q) | \theta(i) \rightarrow \theta(j), u_p, u_q \in \theta(i)\}). \quad (11)$$

为了在每个种群优化过程中能兼顾到全局及局部寻优, 特把当前代全局最优及局部最优解引入到步长调节中, 以引导该算法在解空间内进行彻底的搜索. 同时考虑在搜索前期增大趋化步长有利于全局寻优, 在搜索后期减小趋化步长有利于局部寻优的因素, 在步长设计中考虑了趋化次数, 以使步长随着趋化过程进行自适应调整. 设 N_{cc} 为截止目前趋化已执行的次数, θ_{gbest} 为当前代种群的最优位置, θ_{lbest}^i 为第 i 个细菌个体的最优位置, θ^i 为第 i 个细菌个体的当前位置. 分阶段对步长 $C(i)$ 进行自适应调整, 有

$$C(i) = \begin{cases} c_1, \text{ tumbling} \ \&\& \ J(\theta_{\text{gbest}}) > J(\theta^i); \\ 1, \text{ tumbling} \ \&\& \ J(\theta_{\text{gbest}}) \leq J(\theta^i); \\ c_2, \text{ swimming} \ \&\& \ J(\theta_{\text{lbest}}^i) > J(\theta^i); \\ 1, \text{ swimming} \ \&\& \ J(\theta_{\text{lbest}}^i) \leq J(\theta^i). \end{cases} \quad (12)$$

其中

$$c_1 = D(\theta^i, \theta_{\text{gbest}}) / \sqrt{N_{cc}}, \\ c_2 = D(\theta^i, \theta_{\text{lbest}}^i) / \sqrt{N_{cc}}.$$

2.3 考虑种群分布特性的启发式迁移概率参数设计

种群多样性对最优解的求取起到非常关键的作用, 如果种群分布集中, 则在进化搜索过程中容易陷入局部最优, 使搜索速度减慢, 降低收敛效率. 因此, 基于种群的分布特性调整种群分布对提高细菌觅食

算法的优化性能具有很重要的作用.

定义 5 种群多样性贡献率表征了个体对种群多样性的贡献程度. 对于由 $\theta(1), \theta(2), \dots, \theta(n)$ 这 n 个细菌个体组成的种群 Ω , 可能存在多个可行的最优个体, 从中随机选择一个作为最优个体 $\theta(j)$, 则个体 $\theta(i)$ 对种群多样性的贡献率为

$$\eta_i = \frac{D(\theta(i), \theta(j))}{\sum_{\theta(k) \in \Omega \& k \neq j} D(\theta(k), \theta(j))}. \quad (13)$$

从式 (13) 可以看出, 贡献率越小的个体越接近种群中的最优个体. 因此, 为保持种群的多样性, 防止搜索进入局部最优, 其他非最优个体的迁移概率应与贡献率近似成反比. 为了更好地引导群体向最优方向进化, 对当前代最优个体 $\theta(j)$ 采用不迁移策略, 即迁移概率为 0, 与最优个体 $\theta(j)$ 适应度相同的其他最优个体迁移到一个随机位置. 其他非最优细菌个体的迁移概率为

$$p_{ed}^i = 1 - (\eta_i / \eta^{\max}). \quad (14)$$

2.4 改进后的算法流程

为更好地阐述 IBFO, 特对一些符号做出说明: N_{iter} 为进化代数, N_c 为趋化次数, N_s 为游动次数, N_{re} 为繁殖次数, N_{ed} 为迁移次数, P_{ed} 为迁移概率, j, k, l, n 分别为趋化次数、繁殖次数、迁移次数、进化代数计数器.

定义 6 按照本文定义的细菌位置更新规则, 细菌 $\theta(i)$ 要到达细菌 $\theta(j)$ 所在的位置, 可通过集合 $\{s(u_i, u_j) | \theta(i) \rightarrow \theta(j), u_p, u_q \in \theta(i)\}$ 中元素的组合来实现, 本文将每种组合定义为趋化方向.

为方便理解, 下面以细菌个体 $\theta(1) = (1, 6, 3, 2, 4, 5)$ 到达细菌个体 $\theta(2) = (3, 4, 1, 5, 6, 2)$ 所在的位置为示例, 具体如图 2 所示.

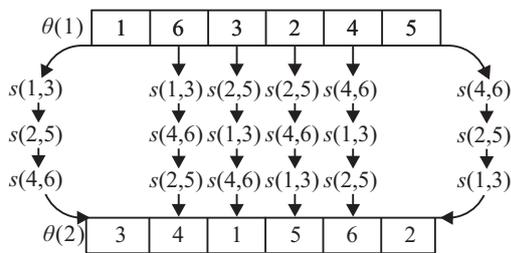


图 2 细菌趋化方向示意图

从图 2 可以看出, 细菌个体 $\theta(1)$ 可通过 6 个趋化方向到达细菌个体 $\theta(2)$ 所在的位置.

改进后的算法具体流程如下.

Step 1: 初始化细菌种群个体位置, 初始化参数 N_{iter} 、 N_c 、 N_s 、 N_{re} 、 N_{ed} .

Step 2: 进化代数, $n = n + 1$.

Step 3: 迁移行为, $l = l + 1$.

Step 4: 繁殖行为, $k = k + 1$.

Step 5: 趋化行为, $j = j + 1$.

对种群内所有细菌而言, 趋化操作包括两个行为: 翻转、游动. 第 i 个细菌进行到第 j 次趋化、第 k 次繁殖、第 l 次迁移时, 其位置为 $\theta^i(j, k, l)$. 计算每个细菌的适应度 $J(\theta^i(j, k, l))$, 从中选择适应度最好的细菌作为全局 θ_{gbest} , θ_{lbest}^i 是第 i 个细菌趋化过程中局部最优位置, 对于初次趋化操作, 随机选择一个趋化方向 $\varphi(i)$, 并用下式更新当前位置.

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) \otimes \varphi(i) \oplus C(i). \quad (15)$$

其中: 趋化步长 $C(i)$ 根据式 (12) 进行选择, 符号 \otimes 表示方向的选择, \oplus 表示从当前位置按方向 $\varphi(i)$ 移动 $C(i)$ 个单位步长.

由式 (15), 如果当前细菌 $\theta^i(j, k, l)$ 的适应度大于 θ_{lbest}^i , 则更新 θ_{lbest}^i , 并沿着当前方向 $\varphi(i)$ 进行游动, 当其达到最大游动次数 N_s 时, 随机选择一个方向 $\varphi(i)$, 同时根据式 (12) 自适应选择步长 $C(i)$ 进行翻转, 如果适应度没有得到改善, 则继续翻转, 直到得到改善时转为游动.

Step 6: 部分不可行解保留. 不可行解可能携带一些可行解无法获取的有用信息, 如距离全局最优解较近, 能够快速搜索到全局最优解. 为此, 采用本文提出的部分不可行解保留策略以提高算法的收敛速度.

Step 7: 如果 $j \leq N_c$, 则返回 Step 5 继续趋化操作.

Step 8: 种群繁殖. 达到临界趋化次数的细菌将完成一次生命周期, 并根据健康度评价进行优胜劣汰, 对于目前给定的序号为 1 到 F 的 F 个细菌个体, 定义第 i 个细菌的健康度为

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(\theta^i(j, k, l)), \quad i \in (1, 2, \dots, F). \quad (16)$$

式 (16) 表示种群中每个细菌在趋化过程完成后适应度的累加和. 对 F 个细菌的健康度指标按降序进行排列, 只保留健康度较好的一半细菌, 并对这些细菌进行一分为二的分裂复制, 以保持种群大小不变, 经过复制产生的这些新生细菌具有与父代相同的位置.

Step 9: 如果 $k \leq N_{re}$, 则返回 Step 4 继续繁殖操作.

Step 10: 迁移操作. 为了增强细菌的全局寻优能力, 保持种群的多样性, 按照本文设计的细菌个体迁移规则进行概率迁移, 种群内的所有细菌将会按照自身指定的概率迁移到解空间中的一个随机位置.

Step 11: 如果 $l \leq N_{ed}$, 则返回 Step 3 继续迁移操作.

Step 12: 如果 $n \leq N_{iter}$, 则返回 Step 2 继续进化;

否则输出优化结果.

3 收敛性分析

定义 7 设 $\{X_k, k = 0, 1, \dots\}$ 是一离散序列, X_k ($k = 0, 1, \dots$) 的有限状态空间为 $S = \{\gamma_1, \gamma_2, \dots, \gamma_N\}$, 若对于任意的 $k \geq 0$ 及 $\gamma_i, \gamma_j, \gamma_0, \gamma_1, \dots, \gamma_{k-1} \in S$, 都有下式成立:

$$P(X_{k+1} = \gamma_j | X_0 = \gamma_0, X_1 = \gamma_1, \dots, X_k = \gamma_i) = P(X_{k+1} = \gamma_j | X_k = \gamma_i), \quad (17)$$

则称 $\{X_k, k = 0, 1, \dots\}$ 为有限 Markov 链.

定义 8 条件概率 $P(X_{k+1} = \gamma_j | X_k = \gamma_i)$ 称为 Markov 链 $\{X_k, k = 0, 1, \dots\}$ 在时刻 k 处于状态 γ_i 的条件下, 在时刻 $k+1$ 转移到状态 γ_j 的转移概率, 记为 $p_{ij}(k)$. 如果 $p_{ij}(k)$ 与时间 k 无关, 则称 $\{X_k, k = 0, 1, \dots\}$ 为齐次有限 Markov 链.

定理 1 IBFO 的种群状态序列是齐次有限 Markov 链.

证明 假设细菌的位置称为细菌的状态, 并设细菌所有状态构成的空间 $S = \{\psi_1, \psi_2, \dots, \psi_N\}$ 是个有限集. 同时, 细菌从当前状态 ψ_i 转移到下一个状态 ψ_j 要经过趋化、繁殖或迁移的过程, 从中可看出, 其转移概率 $p_{ij}(k)$ 只与 ψ_i 有关, 与时间 k 无关, 因而 IBFO 的种群状态序列是齐次有限 Markov 链. \square

定义 9 假设 $J_k = \max\{J(\theta^i) | i \in (1, 2, \dots, F)\}$ 表示在第 k 代可行解中适应度最好的细菌群体, 如果满足下式:

$$\lim_{k \rightarrow \infty} P\{J_k \cap J^* \neq \emptyset\} = 1, \quad (18)$$

则本文提出的 IBFO 是全局收敛的, 其中 J^* 为所求问题的全局最优解集.

定理 2 本文提出的 IBFO 具有全局收敛性.

证明 明显可知

$$P\{J_k \cap J^* \neq \emptyset\} = 1 - P\{J_k \cap J^* = \emptyset\}, \quad (19)$$

本文算法采用可行最优细菌个体不迁移策略, 因此式 (19) 可等价表示为

$$\begin{aligned} P\{J_k \cap J^* \neq \emptyset\} &= \\ &1 - P\{J_1 \cap J^* = \emptyset, J_2 \cap J^* = \emptyset, \dots, \\ &J_k \cap J^* = \emptyset\} = \\ &1 - P\{J_1 \cap J^* = \emptyset\} \times \\ &P\{J_2 \cap J^* = \emptyset | P_{J_1 \cap J^* = \emptyset}\} \times \dots \times \\ &P\{J_k \cap J^* = \emptyset | P_{J_{k-1} \cap J^* = \emptyset}\}. \end{aligned} \quad (20)$$

由定理 1 可知, 种群进化过程中的状态变迁会出现不同时刻状态相同的情况, 状态转移概率也会出现相同的情况. 令

$$\begin{aligned} P\{J_M \cap J^* = \emptyset | J_{M-1} \cap J^* = \emptyset\} &= \\ \text{avg}\{P\{J_2 \cap J^* = \emptyset | J_1 \cap J^* = \emptyset\}, \end{aligned}$$

$$\begin{aligned} P\{J_3 \cap J^* = \emptyset | J_2 \cap J^* = \emptyset\}, \dots, \\ P\{J_k \cap J^* = \emptyset | J_{k-1} \cap J^* = \emptyset\}, \end{aligned} \quad (21)$$

其中函数 $\text{avg}\{\cdot\}$ 表示均值. 由式 (20) 和 (21) 可得

$$\begin{aligned} P\{J_k \cap J^* \neq \emptyset\} &\geq \\ &1 - (P\{J_M \cap J^* = \emptyset | J_{M-1} \cap J^* = \emptyset\})^{k-1}. \end{aligned} \quad (22)$$

因为,

$$0 < P\{J_M \cap J^* = \emptyset | J_{M-1} \cap J^* = \emptyset\} < 1,$$

所以,

$$\begin{aligned} \lim_{k \rightarrow \infty} P\{J_k \cap J^* \neq \emptyset\} &\geq \\ &1 - \lim_{k \rightarrow \infty} (P\{J_M \cap J^* = \emptyset | J_{M-1} \cap J^* = \emptyset\})^{k-1} = \\ &1 - 0 = 1. \end{aligned} \quad (23)$$

同时

$$0 \leq \lim_{k \rightarrow \infty} P\{J_k \cap J^* \neq \emptyset\} \leq 1, \quad (24)$$

由式 (23) 和 (24) 可得

$$\lim_{k \rightarrow \infty} P\{J_k \cap J^* \neq \emptyset\} = 1. \quad (25)$$

因此, 本文提出的 IBFO 具有全局收敛性. \square

4 算例验证与分析

为验证 IBFO 的性能, 随机产生 60 个出入库任务进行测试, 并与标准细菌觅食优化算法 (BFO)、标准遗传算法 (GA) 进行比较. 实验在 Windows XP 系统平台, VC++6.0 开发环境下进行. 进化代数为 600, IBFO 的趋化次数为 25, 游动次数为 4, 繁殖次数为 4, 迁移次数为 3, 迁移概率根据式 (14) 进行自适应调整. BFO 的迁移概率为 0.15, 其他参数与 IBFO 相同. GA 的交叉概率及变异概率分别为 0.9、0.07. 自动化立体仓库系统的参数为: $L = 30 \text{ cm}$, $W = 50 \text{ cm}$, $H = 40 \text{ cm}$, $D_1 = 180 \text{ cm}$, $C = 75$, $N = 6$, $v_x = 1 \text{ m/s}$, $v_y = 0.5 \text{ m/s}$.

不可行解比例 α 的大小可能会因种群规模 N_p 的不同而对最优解产生不同的影响, 为了更直观地看出 α 的大小对算法性能的影响程度, 下面对不同 α 及不同种群规模进行了测试, 结果如图 3 所示. 图 3 中不同的颜色表示取得的不同目标值, 具体每种颜色表示的数值大小由图 3 中右端带刻度的颜色标尺来指示.

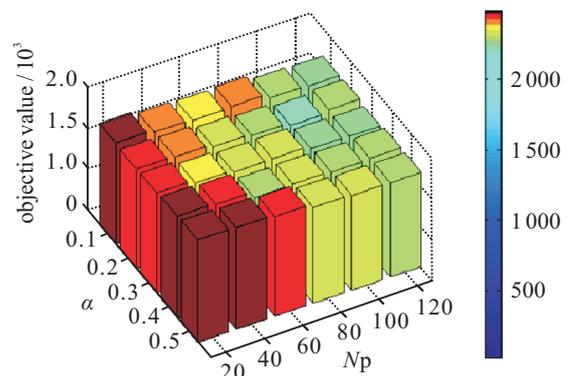


图 3 α 及 N_p 对算法性能的影响

从图3可以看出,对于规模固定的种群,随着 α 的增大,虽然最优解出现波动的情况,但其质量总体呈现先变好后变差的趋势.由此可以得出:如果保留较少的不可行解,则不能充分利用不可行解中的有用信息,从而影响求解的质量;反之,则会造成可行解比例较少,破坏解空间的合理结构,不能有效引导算法向可行最优解的方向搜索,以致不能搜索到最优或次优解.因此,保留适当数量的不可行解有助于改善算法的性能.在 α 不变的情况下,随着种群规模的增大,虽然最优解出现波动,但其质量总体呈现变好的趋势,并且在增加到一定程度时,种群规模的增加对最优解质量的提升越来越小.这主要是因为对于给定的问题,在初始种群规模较小时,随着种群规模的增加,种群的多样性不断增加.因此,种群覆盖的解空间不断变大,求解的质量也不断改善.但是,当种群规模增大到一定程度时,种群基本上可以覆盖到整个解空间.在这种情况下,再增加同等规模的种群对求解质量的改善将不再显著,同时会增加计算时间.由图3,在综合考虑求解效率及求解质量的情况下,本文 α 及 N_p 的取值分别为0.2、100.图4为各算法对本文测试算例的最优解随进化代数的变化趋势.

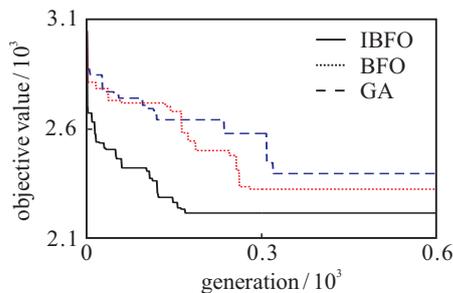


图4 最优解随进化代数的变化趋势

为了对种群多样性进行定量分析,对种群空间稀疏度 ρ 做了定义.具体如下:对于有 $\theta(1)$ 、 $\theta(2)$ 、 \dots 、 $\theta(n)$ 这 n 个细菌个体组成的种群 Ω ,最优个体记为 $\theta(j)$,则种群空间稀疏度 ρ 可表示为

$$\rho = \frac{\sum_{\theta(k) \in \Omega \& k \neq j} D(\theta(k), \theta(j))}{n-1}. \quad (26)$$

图5给出了各算法种群多样性随进化代数的变化情况.

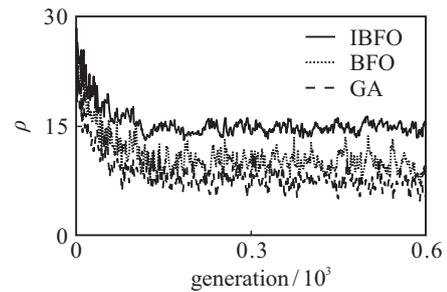


图5 种群多样性随进化代数的变化趋势

从图4、图5可以看出,IBFO无论在求解质量、收敛速度及种群多样性上都显示出较大的优势.首先GA单一的变异机制较容易陷入局部最优,而BFO通过翻转及迁移操作在一定程度上减少了进入局部最优的机率,因此其相对于GA在种群多样性上显示出一定的优势,但GA采用的固定变异概率和BFO采用的固定迁移概率降低了种群的多样性.而本文提出的IBFO启发式迁移策略,使IBFO算法的种群多样性得到较好的保持,有效减少了IBFO进入局部最优的概率;部分不可行解保留策略的实施,使可搜索的解空间得到进一步扩大,为其求出最优或次优解创造了条件;再则IBFO通过采用趋化步长自适应调整策略,能兼顾全局及局部寻优,使其能有效地进行搜索前期的全局寻优及搜索后期的局部寻优,与改进前的BFO相比,寻优效率得到了很大程度的提高.

通过以上分析可以得出:本文提出的启发式迁移策略、部分不可行解保留策略及趋化步长自适应调整策略在使可搜索的解空间得到扩大的同时,根据进化阶段可以有效地引导IBFO从多个方向向最优解靠近,不仅提高了求解质量,也提高了收敛速度.关于求解质量及收敛速度还可以从表1得到验证.为了具有普适性,以国际标准TSPLIB^[15]实例库为基准,对每种算法各运行30次,表1给出了30次的最优解optimal、最优解的平均值opt_avg及收敛时间的平均值con_tm作为比较依据.

表1 算法之间的优化性能比较

实例	TSPLIB最优值	GA			BFO			IBFO		
		optimal	opt_avg	con_tm	optimal	opt_avg	con_tm	optimal	opt_avg	con_tm
Bays29	2020	2 113	2 157	3.50	2 020	2 035	3.13	2 020	2 020	2.04
Oliver30	423.74	454.60	470.27	3.19	423.74	450.85	3.55	423.74	423.74	1.87
Dantzig42	699	769	836	5.24	731	787	3.49	699	699	2.83
Att48	33 522	36 874	37 506	5.18	34 908	35 128	5.07	33 522	33 522	2.46
Eil76	538	627	751	7.15	594	682	6.48	538	550	3.13
Eil101	640.21	779.16	1 184.89	9.21	719.95	863.08	10.92	640.21	695.29	5.73
Gr120	6 942	8 746	9 597	14.16	7 983	8 596	13.38	7 095	7 184	8.01
Ch130	6 110.90	7 422.38	7 930.94	13.75	7 082.35	7 641.40	15.11	6 238.25	6 391.01	8.62

从表1可以看出:在求解质量上,对于规模较小的TSP问题,IBFO能搜索到TSPLIB的最优值,对于规模较大的TSP问题,IBFO虽然不能搜索到TSPLIB的最优值,但与其非常接近,而BFO在这方面劣于IBFO;在收敛速度上,对于所有实例,IBFO都能以较快的速度收敛于最优解或次优解;从算法整体性能上来讲,BFO相对GA显示出了一定的优势,但IBFO的寻优性能明显好于BFO和GA,特别是对于大规模的情况,IBFO的优势更为突出,这进一步表明了IBFO有较强的全局开拓能力及局部探索能力。

5 结 论

为了优化多货叉自动化立体仓库调度,本文提出了一种改进的细菌觅食算法.通过实例仿真可以看出,种群多样性启发式策略及部分不可行解保留策略的实施扩大了解空间的寻优范围,自适应步长的应用提高了寻优效率.特别对于大规模TSP问题,本文提出的改进型细菌觅食算法相对标准细菌觅食算法和遗传算法都表现出良好的性能.该算法对于求解智能制造、物流、航空等相关领域的组合优化问题也有一定的借鉴意义。

参考文献(References)

- [1] Regattieri A, Santarelli G, Manzini R, et al. The impact of dwell point policy in an automated storage/retrieval system[J]. *Int J of Production Research*, 2013, 51(14): 4336-4348.
- [2] Hu Y H, Huang S Y, Chen C, et al. Travel time analysis of a new automated storage and retrieval system[J]. *Computers and Operations Research*, 2005, 32(6): 1515-1544.
- [3] Belenguer J M, Benavent E, Prins C, et al. A branch and cut method for the capacitated location-routing problem[J]. *Computers and Operations Research*, 2011, (38): 931-941.
- [4] Atmaca E, Ozturk A. Defining order picking policy: A storage assignment model and a simulated annealing solution in AS/RS systems[J]. *Applied Mathematical Modelling*, 2013, 37(7): 5069-5079.
- [5] Cardin O, Castagna P, Sari Z, et al. Performance evaluation of in-deep class storage for flow-rack AS/RS[J]. *Int J of Production Research*, 2012, 50(23): 6775-6791.
- [6] 杨文强, 邓丽, 费敏锐, 等. 基于改进禁忌搜索的多目标自动化仓库调度[J]. *计算机集成制造系统*, 2013, 19(8): 2097-2104.
(Yang W Q, Deng L, Fei M R, et al. Multi-objective automated warehousing scheduling based on improved tabu search[J]. *Computer Integrated Manufacturing Systems*, 2013, 19(8): 2097-2104.)
- [7] 李梅娟, 陈雪波, 王莉. 多巷道固定货架拣选作业优化问题的研究[J]. *控制与决策*, 2008, 23(12): 1338-1342.
(Li M J, Chen X B, Wang L. Research on order picking optimization problem for multiple aisles fixed storage racks[J]. *Control and Decision*, 2008, 23(12): 1338-1342.)
- [8] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control[J]. *IEEE Control Systems*, 2002, 22(3): 52-67.
- [9] Panigrahi B K, Pandi V R, Sharma R, et al. Multiobjective bacteria foraging algorithm for electrical load dispatch problem[J]. *Energy Conversion and Management*, 2011, 52(2): 1334-1342.
- [10] Dos Santos Coelho L, Da Costa Silveira C, Sierakowski C A, et al. Improved bacterial foraging strategy applied to team workshop benchmark problem[J]. *IEEE Trans on Magnetics*, 2010, 46(8): 2903-2906.
- [11] Singh S, Ghose T, Goswami S K. Optimal feeder routing based on the bacterial foraging technique[J]. *IEEE Trans on Power Delivery*, 2012, 27(1): 70-78.
- [12] Bakwad K M, Pattnaik S S, Sohi B S, et al. Parallel bacterial foraging optimization for video compression[J]. *Int J of Recent Trends in Engineering*, 2009, 1(1): 118-122.
- [13] Bhushan B, Singh M. Adaptive control of nonlinear systems using bacterial foraging algorithm[J]. *Int J of Computer and Electrical Engineering*, 2011, 3(3): 335-342.
- [14] 马苗, 梁建慧, 郭敏. 基于细菌觅食算法的SAR图像阈值分割[J]. *西安电子科技大学学报*, 2011, 38(6): 152-178.
(Ma M, Liang J H, Guo M. SAR image thresholding segmentation based on the bacteria foraging algorithm[J]. *J of Xidian University*, 2011, 38(6): 152-178.)
- [15] TSP95[DB/OL]. <http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/>.

(责任编辑: 齐 霖)