

蔡继萌,王卫兵,曲家灏,等.基于遗传和蚁群交互算法的穴盘苗稀植移栽路径优化[J].华中农业大学学报,2025,44(4):248-258.  
DOI:10.13300/j.cnki.hnlkxb.2025.04.024

## 基于遗传和蚁群交互算法的穴盘苗稀植移栽路径优化

蔡继萌,王卫兵,曲家灏,郭小龙,李国栋,吴潇雨

石河子大学机械电气工程学院,石河子 832000

**摘要** 针对穴盘苗移栽到低密度穴盘路径规划效率低下问题,基于遗传算法和蚁群算法提出蚁群-遗传(ant colony-genetic optimization algorithm,ACGO)和遗传-蚁群(genetic-ant colony optimization algorithm,GACO)交互算法进行稀植移栽路径优化。通过仿真试验,使用固定顺序法和其他5种算法计算从72-32、72-50、128-50、128-32孔穴盘的移栽路径长度,对比分析不同算法在优化路径长度和计算时间上的差异,并通过相对标准差评估算法的稳定性。结果显示,在72孔到32孔穴盘移栽中,对比固定顺序法,GACO算法的平均路径长度缩短59.3%,平均计算时间为5.15 s,相对标准差约为1.5%;ACGO算法的平均路径长度缩短19.2%,平均计算时间为13.50 s,相对标准差约为1%。进一步研究显示,ACGO算法在200孔移栽至72孔和105孔场景的优化效果弱于贪婪算法,而GACO算法在不同孔数组合和缺苗数下展现出更高的普适性和稳定性。研究表明,ACGO和GACO 2种交互算法均可提升原算法的性能,但GACO算法在处理复杂稀植移栽路径规划问题时表现更为优越。

**关键词** 穴盘苗;稀植移栽;路径优化;交互算法;遗传-蚁群算法

**中图分类号** TP18    **文献标识码** A    **文章编号** 1000-2421(2025)04-0248-11

目前,苗木种植方式逐渐从传统的分散土地耕作向集中规模化的穴盘育苗模式转变<sup>[1]</sup>。温室环境下的穴盘育苗技术已广泛应用于果蔬种植,尤其是在番茄等经济作物的生产中,通过高密度培育提高幼苗存活率和生长一致性。然而,当幼苗生长至一定阶段后,需要从高密度穴盘移栽到低密度穴盘,以确保苗木获得更优的生长条件<sup>[2]</sup>。由于移栽穴盘中的待移栽苗木分布不均,目标穴盘中空穴的位置也不确定,使得合理规划移栽路径成为提高作业效率和降低移栽成本的关键问题。

稀植移栽的过程类似于经典组合优化的旅行商问题(traveling salesman problem,TSP),即需要寻找1条最短或成本最低的路径来访问一组点。在稀植移栽中,这些点是待移植的钵苗和待稀植的空穴;在TSP中,这些点是需要访问的城市。贪婪算法(greedy algorithm,GrA)计算简单、运行速度快,是解决此问题最经典有效的算法,但容易陷入局部最优,未必能得到全局最优解,可作为候选算法进行稀植移栽路径规划对比测试<sup>[3]</sup>。遗传算法(genetic algo-

rithm, GA)和蚁群算法(ant colony optimization, ACO)都是解决TSP问题的有效工具,GA算法通过编码、生成初始种群、适应度评估、选择、交叉、变异以及迭代优化等步骤实现稀植移栽路径的优化<sup>[4-5]</sup>,ACO算法则通过分布初始信息素、信息素蒸发以及迭代求解等机制优化稀植移栽的路径<sup>[6]</sup>,它们通过不同的机制寻找最优或近似最优解,为温室穴盘苗间稀植移栽路径规划提供了有效的解决方案。GA算法以其强大的全局搜索能力被广泛应用,但可能遭遇过早收敛的问题,并在维持种群多样性方面遇到挑战。ACO算法在找到高质量局部优化解方面表现出色,但在处理规模较大的路径规划问题时,其效率有待提高<sup>[7-9]</sup>。通过ACO算法优化GA算法后的蚁群-遗传算法(ant colony-genetic optimization algorithm,ACGO)可以增强搜索过程的多样性、加速算法收敛、提升解决方案的品质、增加参数调整的灵活性,并提高对环境变化的适应能力<sup>[10]</sup>。遗传-蚁群算法(genetic-ant colony optimization algorithm,GACO)是利用GA算法的全局搜索能力优化ACO算法,进

收稿日期:2024-06-30

基金项目:国家自然科学基金项目(61763042)

蔡继萌,E-mail:cjmimmortal@foxmail.com

通信作者:王卫兵,E-mail:wwbshz@163.com

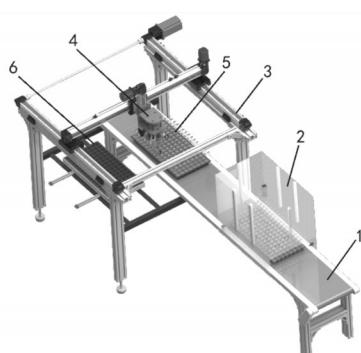
一步提高搜索效率、加快收敛速度、增强算法稳定性、优化参数配置,提升了解决方案的整体质量<sup>[11]</sup>。针对稀植移栽,童俊华等<sup>[12]</sup>提出一种基于贪心算法的温室钵苗稀植移栽路径优化策略,验证了贪心算法提高移栽机稀植移栽效率的有效性。Tong等<sup>[13]</sup>进一步提出了一种贪婪遗传算法来优化路径,并在不同密度穴盘和不同缺苗数条件下验证了交互算法在稀植移栽路径规划中的综合性能。

使用常规的固定顺序法(fixed order method, FOM)进行稀植移栽时,由于移栽穴盘缺苗位置的不确定性和目标穴盘的多选择性,选定移栽的路径往往不合理,从而增加管理成本并降低生产效益,因此,通过优化路径规划减少末端执行器的无效移动成为提高稀植移栽作业效率的关键措施。研究表明,无论穴盘的密度如何,所采用的优化算法需要处理规划路径以最小化移植总距离的核心问题是一致的,密集穴盘和低密度穴盘表现出相似的变化规律<sup>[14-16]</sup>。本研究选取从72孔到32孔穴盘的移植试验作为研究对象,提出ACGO和GACO算法来优化稀植移栽路径,比较不同优化算法在该特定场景下的表现,探索它们在实际应用中的有效性和普适性,旨在缩短稀植移栽路径距离以提高末端执行器的作业效率。

## 1 材料与方法

### 1.1 稀植移栽机结构

穴盘苗稀植自动移栽机如图1所示,穴盘统一标准尺寸为250 mm×500 mm<sup>[17]</sup>,常规移栽穴盘规格一般为72、128孔,目标穴盘为50、32孔。目前,FOM在穴盘间移栽的应用仍然相对广泛<sup>[18]</sup>,如图2所示,



1. 传送装置 Transport device; 2. 视觉检测系统 Visual detection system; 3. 移栽系统 Transplanting system; 4. 末端执行器 End effector; 5. 移栽穴盘 Supply plug tray; 6. 目标穴盘 Target plug tray.

图1 温室钵苗稀植移栽机

Fig.1 Integrated automated transplanter

模拟了末端执行器从72孔移栽穴盘到32孔目标穴盘4种方法的移栽路径。

### 1.2 蚁群-遗传算法稀植移栽路径规划方案

在蚁群-遗传算法的稀植移栽路径规划方案的框架中,GA算法负责执行全局搜索并生成初步的解决方案种群,ACO算法负责对这些解决方案进行局部优化和细节调整。GA算法生成1个初始种群后,用染色体编码表示机械臂从移栽穴盘的穴孔到目的穴盘的空穴的移动序列,计算每个染色体的适应度,通过选择、交叉和变异等操作进行全局搜索和生成遗传多样性,此时ACO算法对高适应度的解决方案进行局部优化和精细调整,此迭代过程持续进行,直到满足终止条件,输出最优化的移栽路径结果。

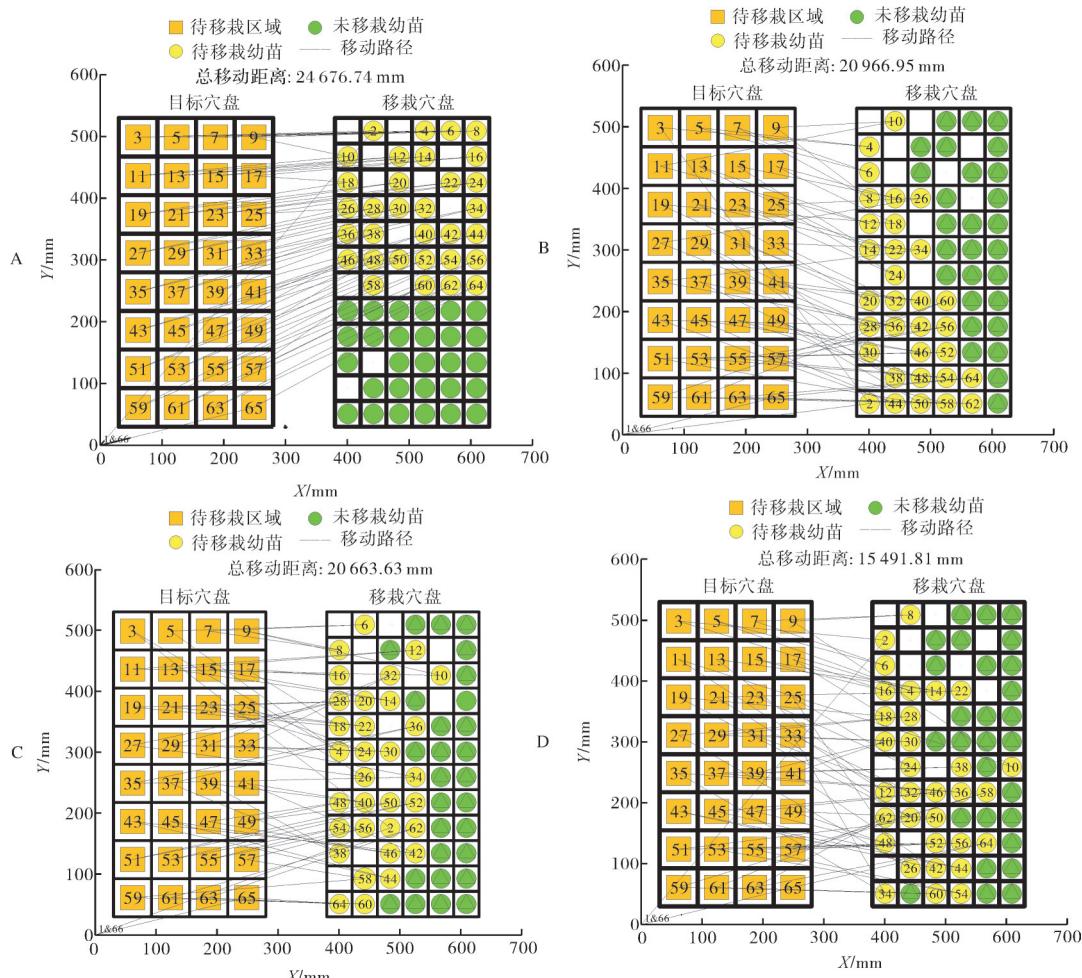
1) 初始种群生成。初始种群数量设置为200,随机生成200个不同的穴孔访问序列作为候选解。染色体使用整数编码,初始染色体通过随机过程生成,以保证种群的多样性。采用双穴盘交替编码方式,按照随机生成的方式,从移栽穴盘至目的穴盘交替的方式进行标记,例如,随机移动序列[3, 1, 4, 2, 5, 7, 6, 8, 9, 10],第1个数字“3”表示移栽穴盘的标号3穴孔的苗木被移植到目的穴盘的标号1穴孔,以此类推。初始种群进行1次基本的适应度评估,确定每个染色体的起始表现,为后续操作提供初步的数据支持。

2) 适应度评估与遗传操作。适应度函数包括2个部分:计算路径长度的函数和将路径长度转换为适应度值的函数。稀植移栽路径规划的目标之一是路径长度最小化,适应度越高的个体越有可能被选中进行繁衍,因此直接使用路径长度的负值将其转换为一个最大化问题。适应度值 $F(x)$ 的表达式:

$$F(x) = -D = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (1)$$

式(1)中, $(x_i, y_i)$ 和 $(x_{i+1}, y_{i+1})$ 分别是路径序列中相邻两点的坐标; $n$ 是路径中点的总数。采用轮盘赌方法基于个体的适应度随机选择下一代,以确保适应度较高的个体有更大的机会被选中。外循环遍历每个个体位置,为每个位置选择1个新个体。内循环通过与随机数( $r_p$ )与累积概率( $P_x$ )的比较来选择个体。一旦 $P_x(j)$ 大于或等于 $r_p$ ,就选择该 $j$ 位置的个体复制到新种群中。

使用混合交叉技术进行交叉操作,交叉概率( $P_c$ )设定为0.8,父代染色体组通过将选择后的种群矩阵( $XSel$ )一分为二,分别赋值给2个父代矩阵。计算1个特定的交叉因子 $\beta$ 系数用于控制生成的子代离其



A: 固定顺序穴盘稀移植模型 Fixed order plug sparse planting model; B: 贪婪算法优化稀移植路径模型 Greedy algorithm optimizes sparse planting transplanting path model; C: 遗传算法优化稀移植路径模型 Genetic algorithm to optimize sparse transplanting path model; D: 蚁群算法优化稀移植路径模型 Ant colony algorithm to optimize sparse transplanting path model.

图2 4种稀植方案路径长度对比

Fig.2 Comparison of path length of four sparse planting schemes

父代的距离,以在探索与利用之间实现平衡。 $\beta$ 表达式如式(2):

$$\beta = \begin{cases} 2\mu^{\frac{1}{\eta_c+1}} & \mu \leq 0.5 \\ \left(\frac{1}{2(1-\mu)}\right)^{\frac{1}{\eta_c+1}} & \mu > 0.5 \end{cases} \quad (2)$$

式(2)中,  $\mu$ 是 $[0, 1]$ 之间的随机数;  $\eta_c$ 是分布指数,控制交叉后后代的多样性。

使用计算得到的 $\beta$ ,结合父代基因值,按照混合交叉的公式生成子代,表达式为:

$$\delta = \begin{cases} \left(2\mu + (1-2\mu)\left(1 - \frac{x - x_{\min}}{x_{\max} - x_{\min}}\right)^{\frac{1}{\eta_c+1}}\right) & \mu \leq 0.5 \\ 1 - \left(2(1-\mu) + 2(\mu-0.5)\left(1 - \frac{x_{\max} - x}{x_{\max} - x_{\min}}\right)^{\frac{1}{\eta_c+1}}\right) & \mu > 0.5 \end{cases} \quad (4)$$

式(3)中,  $C$ 是新生成的子代个体,  $P_1, P_2$ 是选中进行交叉的2个父代个体。

同时,变异操作采用单点变异和多点变异2种方法,对于种群中的每个基因,随机生成1个 $0 \sim 1$ 之间的数,如果该数小于变异概率( $P_m$ ),则该基因位置被选为变异。使用  $\mu = \text{rand}(N, D)$  生成变异方向和幅度的控制参数,如果  $\mu$  小于 0.5, 变异增加基因值;如果大于 0.5, 变异减少基因值,  $\delta$  新值表达式如式(4):

$$C = 0.5[(1+\beta)P_1 + (1-\beta)P_2] \quad (3)$$

式(4)中,  $x$  是被变异基因的当前值;  $x_{\min}$  和  $x_{\max}$  分别是该基因的最小值和最大值;  $\mu$  是  $[0, 1]$  之间的随机数;  $\eta_m$  是分布指数, 用于控制变异步长的大小, 从而影响解的多样性。

使用  $\min$  和  $\max$  函数强制执行边界, 确保所有变异后的值都在允许的范围内, 避免因超出参数范围而导致的潜在问题。

3) 蚁群算法的集成。GA 算法建立了一个较为稳定且多样化的初始种群后, 将 ACO 算法集成到 GA 算法作为一种局部优化策略。在 GA 算法中为了保持解的质量, 迭代过程设定为 200 次, 设定 10% 的精英选择比例, 这些个体被直接保留至下一代, 不再经过任何操作。ACO 算法集成的具体操作是, 每隔 50 代, 选取种群中 10% 的个体通过 ACO 算法进行局部搜索优化, 寻找更优路径。对每个选定的路径, 初始信息素, 根据 ACO 算法规则进行迭代, 不断更新信息素, 直到达到一定的迭代次数或解的质量不再显著提升。根据最终的信息素浓度, 选择最优路径后, 将通过 ACO 优化后的路径更新回 GA 的种群中, 这些新路径替换原有的个体或作为新的个体加入种群。这种方法既可以刷新种群中的部分基因, 又可以引入新的优质基因, 增强种群的整体性能。

4) 迭代进化与结果输出。迭代进化是 GA 算法的核心机制。当该算法完成 200 代迭代后, 进程将停止。当迭代过程结束时, 算法从最终的种群中筛选出适应度最高的个体, 将其确定为最优解。ACGO 算法规划的从 72 孔穴盘到 32 孔穴盘的路径, 并且在移植穴盘中有 12 个空穴的情况下, 结果如图 3 所示。

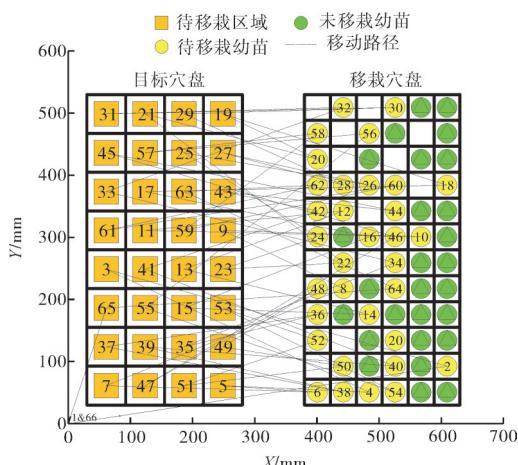


图 3 蚁群-遗传算法优化稀移植路径模型

Fig.3 ACGO algorithm to optimize sparse planting transplanting path model

### 1.3 遗传-蚁群算法稀移植路径规划方案

在遗传-蚁群算法的稀移植路径规划方案的框架中, GA 算法用于优化 ACO 算法的关键参数, 而 ACO 算法负责具体的路径搜索。首先将 ACO 算法中影响性能的参数编码为染色体的一部分, 然后进行初始化种群、适应度评估和其他遗传操作, 这个过程重复进行, 直至适应度不再显著改进。最终, 选择适应度最高的染色体作为最优参数组合, 用于配置 ACO 算法, 实现在稀移植路径规划问题上的最佳性能。

1) 遗传算法优化蚁群算法。遗传算法用于设定蚁群算法中信息素蒸发率(evapRate)、信息素重要度( $\alpha$ )和启发式因子( $\beta$ )重要度的取值范围。信息素蒸发率设定为 0.1~0.9。低蒸发率加速信息素积累, 强化了算法对已发现优良路径的开发; 较高的蒸发率则加强了探索新路径的动力, 防止算法陷入局部最优。信息素重要度设定为 1~5。低信息素重要度使得蚁群在路径选择时更依赖于启发式信息; 较高的值加强了信息素浓度, 有助于算法加速收敛至最佳路径。启发式因子重要度设定为 1~5。此参数的增加强化了启发式信息在路径决策中的作用。

通过遗传算法的操作不断进化种群, 寻找表现最优的参数组合。在每一代中, 以蚁群算法在当前参数下解决特定优化问题的效果为标准评估个体的适应度。通过多代迭代, 从最终种群中选择适应度最高的个体, 确定为最优参数集, 以此配置运行蚁群算法, 达到优化目标。

2) 初始化信息素和距离矩阵。在算法开始时, 根据通过遗传算法优化得到的参数设置 ACO 的运行环境, 为了在搜索开始阶段不对任何路径产生偏见, 所有路径上的信息素浓度通常被初始化为 0.1。在稀移植路径优化的场景中, 节点代表钵苗抓取位置和投放位置, 因此, 距离矩阵记录移植穴盘中每个可能的抓取位置到目标穴盘中每个可能的投放位置之间的距离。距离  $d_{i,j}$  采用欧氏距离(Euclidean distance)来计算, 表达式如式(5):

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

式(5)中,  $(x_i - x_j)$  和  $(y_i - y_j)$  分别是位置  $i$  和位置  $j$  的坐标。

3) 路径构建。路径构建是整个算法核心的一部分, 涉及到蚂蚁在解空间中根据信息素和启发式信息进行迭代搜索, 最终构建出从起点到终点的 1 条完

整路径。起始每只蚂蚁被放置在起点上,起点是随机选择确定的。每只蚂蚁在其当前位置选择下1个位置时,基于信息素浓度和启发式信息进行转移概率计算,表达式如式(6):

$$P_{i,j} = \frac{(T_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum_{k \in \text{allowed}} (T_{i,k})^\alpha (\eta_{i,k})^\beta} \quad (6)$$

式(6)中,  $P_{i,j}$  是从节点  $i$  到  $j$  的转移概率;  $T_{i,j}$  是节点  $i$  到  $j$  的信息素浓度;  $\eta_{i,j}$  是启发式信息;  $\alpha$  和  $\beta$  是控制信息素和启发式信息重要性的参数;  $\text{allowed}$  是当前蚂蚁还未访问的节点集。

根据计算概率,使用轮盘赌选择法决定蚂蚁的下1个移动,每只蚂蚁会被记录其访问过的路径,包括移动顺序和经过的节点。记录每只蚂蚁到达终点构建完整路径的信息,用于后续的信息素更新。

4)信息素更新。信息素更新包含2个主要过程:信息素蒸发和信息素强化。信息素蒸发是一个减少路径上信息素强度的过程,模拟自然界中信息素随时间而消散的现象。这一机制鼓励蚁群探索新的路径,而不是持续沿用旧路径。在每次迭代结束后,对所有路径上的信息素量进行更新,表达式为:

$$\tau_{i,j+1} = (1 - \rho) \cdot \tau_{i,j} \quad (7)$$

式(7)中,  $\rho$  (其中,  $0 < \rho < 1$ ) 是蒸发率参数  $\tau_{i,j}$  表示从节点  $i$  到  $j$  的路径上的信息素浓度。

信息素增强是基于蚂蚁找到的路径质量来增加路径上的信息素量,以鼓励其他蚂蚁走同样的路。优质的路径会获得更多的信息素,从而吸引更多的蚂蚁。在每只蚂蚁完成其路径后,计算每条路径的信息素增加量,表达式为:

$$\Delta\tau_{i,j}^k = \frac{Q}{L_k} \quad (8)$$

式(8)中,  $Q$  为常数;  $L_k$  是蚂蚁  $k$  的路径长度。遍历每只蚂蚁走过的路径,将相应的信息素增加量加到对应的信息素矩阵元素上,信息素更新表达式如式(9):

$$\tau_{i,j+1} = \tau_{i,j} + \sum_{k=1}^m \Delta\tau_{i,j}^k \quad (9)$$

式(9)中,  $m$  是本次迭代中总的蚂蚁数量。在所有蚂蚁完成其路径后,对信息素矩阵集体进行1次更新,确保信息素的变化反映了所有蚂蚁的搜索成果。通过这两个操作,ACO算法维持了对理想路径的强化和对不理想路径的弱化,保证了算法的探索性和开发性之间的平衡,该机制使得算法能够逐渐逼近最优解,同时避免固定在局部最优解上。

5)迭代更新与结果输出。重复迭代次数设置为

100,每次迭代都会基于现有的信息素矩阵进一步优化路径选择。在多次迭代后,选择具有最低总成本或最短距离的路径作为最优路径。最后,返回这条最优路径的细节,包括路径序列和总距离。在移栽穴盘中有12个空穴的情况下,GACO算法规划的从72孔穴盘到32孔穴盘的路径结果如图4所示。

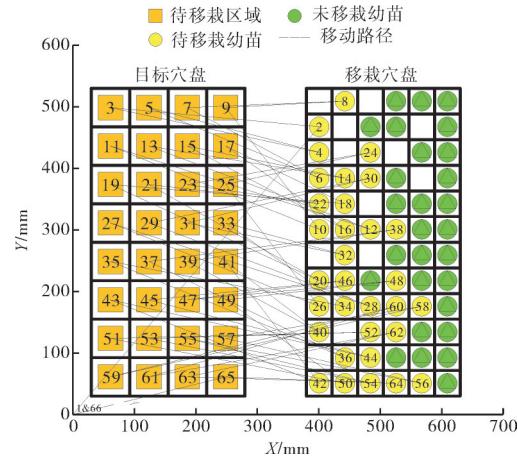


图4 遗传-蚁群算法优化稀植移栽路径模型

Fig.4 GACO algorithm to optimize sparse planting transplanting path model

## 2 结果与分析

为验证本研究提出的2种算法在穴盘苗稀植移栽应用场景中的适用性,选取FOM、GrA、GA、ACO等算法与ACGO、GACO算法进行对比。一般移栽穴盘的缺苗平均值在12.5%<sup>[19]</sup>,因此,选取缺苗数为9时,从72孔穴盘稀植移栽到32孔穴盘,随机分布的10组样本试验数据进行路径长度比较,结果如表1所示。5种算法路径优化效果如表2所示。

由表1和表2可知,与传统的FOM方法相比,GrA、GA、ACO、ACGO和GACO均在优化路径长度方面有所改进。与FOM算法相比,GrA算法优化幅度为15.7%,在所有算法中优化效果最小。ACGO算法规划的平均路径长度为19 786.77 mm,路径优化幅度较GA算法的13.9%有所改进,路径优化幅度为19.2%,较GA算法提升5.3百分点。GACO算法在所有算法中性能最佳,平均路径长度大幅度缩短到9 970.53 mm,平均路径长度较FOM规划的路径长度减少了14 532.91 mm,平均路径优化幅度高达59.3%,较引入GA算法之前的ACO算法的路径优化幅度提升20.4百分点。在稀植移栽路径规划中,ACO算法展示了更佳的性能,其平均路径优化幅度为38.9%,远超GA算法的13.9%,经过优化后的GACO更是显示了远超其他算法的卓越性能。

表1 从72穴盘到32穴盘且缺苗数为9的稀植移栽路径规划长度

Table 1 Path planning length of sparse transplanting from 72 to 32-hole tray with 9 missing seedlings mm

样本编号 Number of samples	FOM	GrA	GA	ACO	ACGO	GACO
1	24 620.81	20 265.41	21 335.35	14 179.29	19 767.79	10 036.32
2	24 007.16	20 719.45	21 425.60	14 955.15	19 782.62	10 038.16
3	24 101.78	20 696.76	20 689.10	14 705.37	21 176.19	9 931.28
4	24 908.42	20 299.42	21 041.66	14 850.16	19 421.30	9 967.72
5	24 664.44	20 460.48	21 291.95	15 580.36	19 266.87	10 021.84
6	24 923.67	20 949.76	21 562.62	14 883.95	19 079.85	9 759.95
7	24 527.68	20 651.56	19 853.47	14 680.08	20 569.92	9 955.87
8	24 429.26	20 999.00	21 296.15	15 367.89	19 329.19	10 210.01
9	24 552.25	20 498.11	21 259.42	14 809.64	19 921.57	9 716.17
10	24 298.92	20 647.84	21 265.96	15 753.31	19 552.39	10 068.02

表2 稀植移栽路径规划长度分析

Table 2 Analysis on path planning length of sparse planting transplanting

路径规划算法 Path planning algorithm	平均路径长度/mm Average path length	平均缩短路径/mm Average shortened path	平均优化幅度/% Average optimization range	改进后优化提升幅度/% Improvement rate of optimized
FOM	24 503.44	/	/	/
GrA	20 647.84	3 855.60	15.7	/
GA	21 102.13	3 401.31	13.9	/
ACO	14 976.52	9 526.92	38.9	/
ACGO	19 786.77	4 716.67	19.2	5.3
GACO	9 970.53	14 532.91	59.3	20.4

为了验证所提出算法在目标穴盘上移植幼苗总数少于可用穴位情况下的有效性,选取缺苗数从44到56,间隔为4的情况进行验证。对每个缺苗数,进行10次试验取平均值以确保结果的稳定性。图5展示了各种缺苗数下算法的路径长度对比结果。

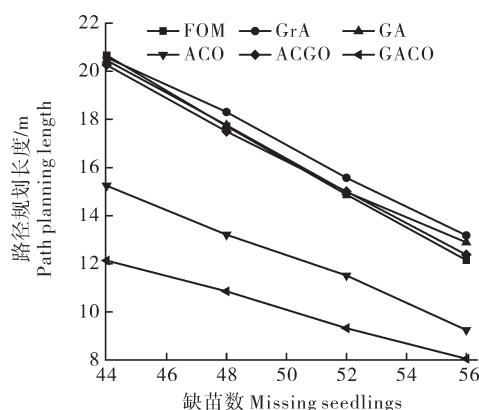


图5 幼苗数少于穴孔数路径长度对比

Fig.5 Comparison of path length of seedling number less than hole number

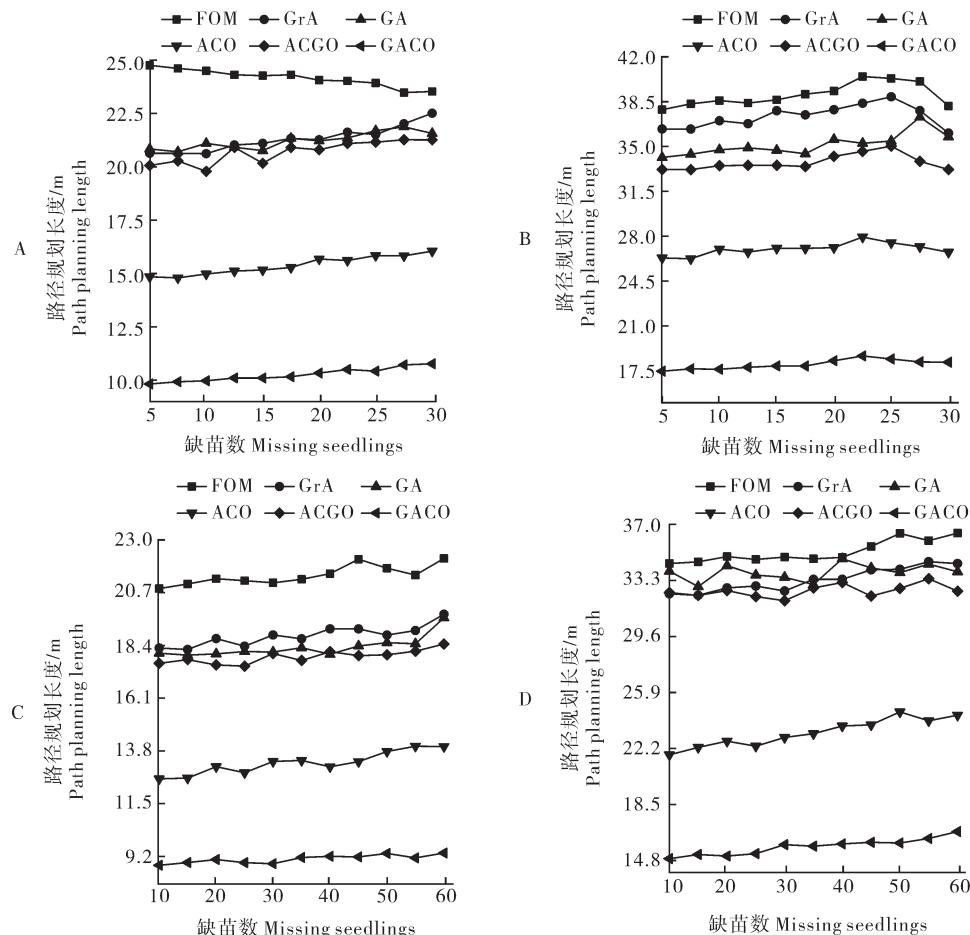
由图5可知,当幼苗数量小于目标盘孔数时,GrA、GA和ACGO算法的优化路径性能并不优于固定顺序的方法。然而,随着幼苗数量的增加,这3种算

法开始显现出其优化效果。ACO和GACO算法在整个测试范围内始终展示出较好的路径优化性能。

为进一步验证本研究提出的算法是否适用于穴盘苗稀植移栽的应用场景,对规划128-50、128-32、72-50、72-32孔穴盘稀植移栽路径长度进行比较。72-50、72-32孔移栽穴盘的苗数从5至25,以2为间隔等差数列的形式递增,128-50、128-32孔移栽穴盘的苗数从8至48,以4为间隔等差数列的形式递增,每种缺苗数取10组样本取平均值,5种算法平均优化路径长度如图6所示,GrA、GA、ACO、ACGO和GACO算法平均运行时间为0.12、1.47、3.25、13.50、5.15 s。FOM方法的计算时间最短,由于其采用固定顺序法,因此无需额外计算路径优化。GrA方法计算时间为0.12 s,表现出快速求解能力。相比之下,GA算法的计算时间为1.47 s,ACO算法计算时间为3.25 s,而ACGO算法计算时间较长,为13.50 s,这主要是由于其优化过程中涉及更复杂的计算。GACO算法的计算时间为5.15 s,尽管高于GA和ACO,但在路径优化质量上展现出更优的性能。综合考虑计算效率和优化路径质量,GACO算法在本研究的应用场景中表现最佳。

本研究针对稀植移栽路径规划问题,通过比较4种算法在不同规格的移栽场景下的性能,发现GrA算法和GA算法相比固定顺序法均能有效缩短路径长度。然而,ACGO算法的性能明显优于GrA和GA算法,表明其在路径优化上具有更加显著的效果。尽管如此,ACGO算法规划的路径长度依然长于ACO算法,显示出ACO算法在这一应用中的

优越性。最为突出的是GACO优化算法,其规划能力远超过ACO算法,进一步优化了路径长度。这一结果表明,基于蚁群算法结合遗传算法的GACO策略在处理稀植移栽路径规划问题时表现出色。该策略能够有效克服局部最优问题,并提升搜索效率,通过综合2种算法的优点,达到更优的路径规划结果。



A:72-32孔各算法规划路径长度变化对比 Comparison of the path length changes of each algorithm planning in 72-32 holes; B:72-50孔各算法规划路径长度变化对比 Comparison of the path length changes of each algorithm planning in 72-50 holes; C:128-32孔各算法规划路径长度变化对比 Comparison of the path length changes of each algorithm planning in 128-32 holes; D:128-50孔各算法规划路径长度变化对比 Comparison of the path length changes of each algorithm planning in 128-50 holes.

图6 不同规格穴盘移栽路径优化长度对比

Fig.6 Comparison of optimal length of plug transplanting path with different specifications

FOM在移栽钵苗时采用固定的逻辑,不涉及计算时间的考量。ACGO算法的平均计算时间为13.50 s,原因是引入ACO算法优化所需的大量计算资源。而GACO算法的平均计算时间为5.15 s,比单独的ACO算法增加了58.46%,这反映出整合2种优化技术的优势时,对计算资源和时间的需求也相应地增加。尽管如此,GACO算法的计算时间在穴盘苗稀植移栽问题中仍属于可接受的时间成本范

围内。

优化算法相对标准差(RSD)如图7所示,通过RSD对4种优化算法进行稳定性评价。图7清晰展现了算法性能在多次运行中的一致性。GA算法表现出一定的变异性,其RSD的中值在2.5%左右,箱形图的上界较高,表明在不同试验中解的波动较大。虽没有显著的离群点,但数据的扩散程度较广,说明算法在多次运行中的一致性较差。这些特征反映了

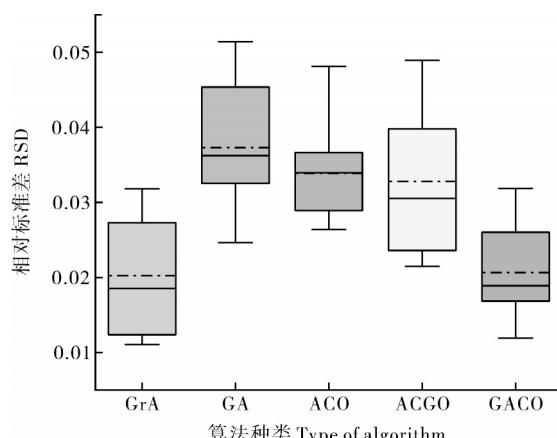


图 7 优化算法相对标准差

Fig.7 Optimize the relative standard deviation of the algorithm

GA 算法在稳定性方面存在的局限,尤其是在处理需要高一致性的问题时可能面临的挑战。GrA 算法表现出中等水平的变异性, RSD 的中位数大约在 2.5%~3.0%。该算法的 RSD 分布较为紧凑,表明其解的波动在一定范围内,虽然没有达到 GACO 算法的稳定性,但也比 GA 算法显示出更好的一致性。ACO 算法展现了较高的稳定性, RSD 中值约为 1.5%, 变异性相对较小, 这进一步证实了算法在多次运行中维持较一致性能的能力。ACGO 算法表现出了与 ACO 相媲美的高稳定性, 其 RSD 中值及分布范围均紧凑, 反映出该算法在多次运行中解的一致性。

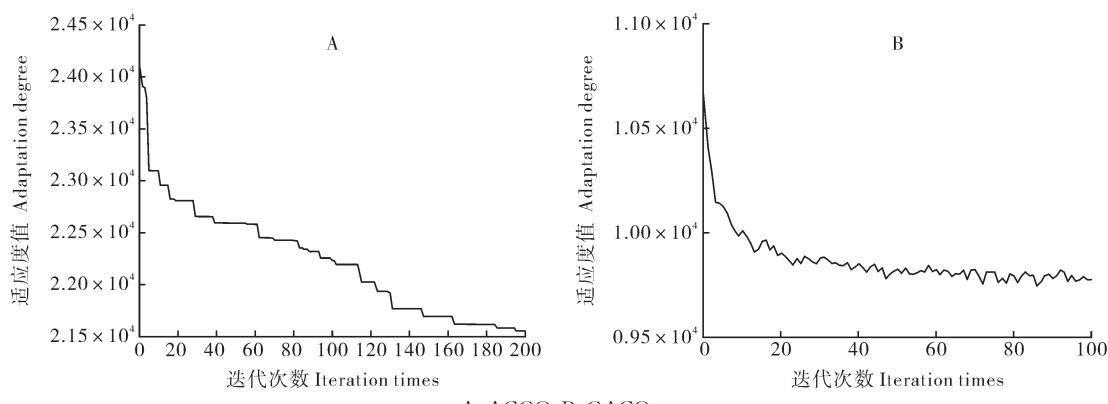


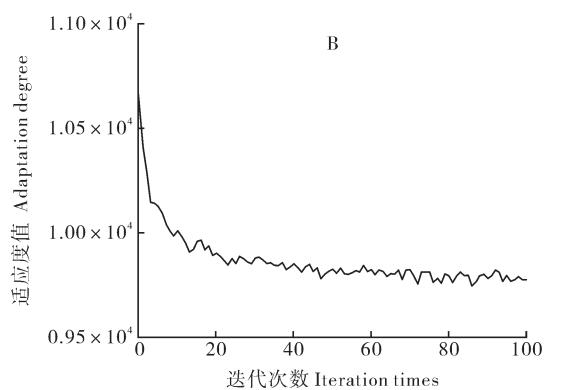
图 8 交互算法迭代优化过程中的性能趋势

Fig.8 Performance trends in the iterative optimization process of optimization algorithms

为进一步增加算法的普适性,增加 200 孔到 105 孔、128 孔和 72 孔的算例比较。通过增加这些大规模和不同孔数组合的实验场景,能够更全面地验证所提出的交互优化算法在处理不同规模、复杂度和稀植需求下的有效性,路径优化结果如图 9 所示。3 种稀移植裁场景中 200-128 孔的路径长度规划成本最

较高,稳定性优良。GACO 算法展现了所有算法中最优的表现,其 RSD 紧凑,中值约为 1%,显著低于其他算法。这表明 GACO 算法具有卓越的稳定性和高度的可靠性,使其在需要高稳定性的应用中非常适用。GACO 算法的卓越表现突显了遗传算法与蚁群优化相结合的优势,适合在对算法的稳定性有较高要求的稀移植裁场景中使用。

如图 8 所示,2 种算法在处理复杂优化问题时,都能够有效平衡探索与利用,从而在合理的迭代次数内提供高质量的解决方案。ACGO 算法在初始下降非常迅速,这表明算法在早期迭代中快速找到了更优的解。随着迭代次数增加,适应度的下降速率减慢,逐渐趋于稳定。GACO 算法适应度下降很快,但整体适应度范围较低,显示出更高效的优化效果。2 种算法都表现出快速的初始收敛速度,但 GACO 算法显示出较快的达到较低适应度的能力。GACO 算法的最终适应度低于 ACGO 算法,表明 GACO 算法提供的解决方案更高效。由图 8B 可知,适应度在迭代后期有些波动,表明算法在探索新解和利用已知解之间做了一些权衡,而图 8A 中 ACGO 算法的适应度曲线则相对平滑,显示出较高的稳定性。总之,尽管其在迭代过程中表现出了较多的波动,GACO 算法总体性能优于 ACGO 算法,ACGO 算法虽然稳定,但在最终适应度方面较差。



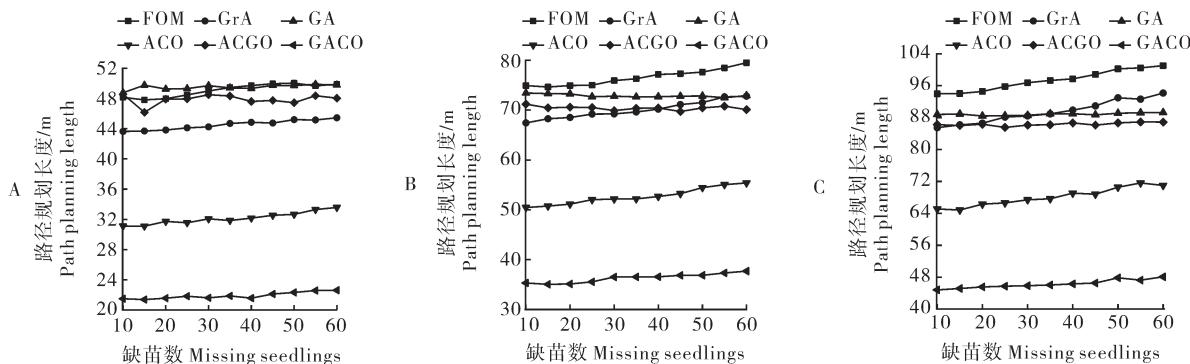
A:ACGO;B:GACO.

图 8 交互算法迭代优化过程中的性能趋势

高。计算时间的对比结果表明,FOM 方法由于采用固定顺序方式,不涉及路径优化计算,因此无需额外计算时间。GrA 方法计算时间为 0.42 s, 表现出较快的计算能力。GA 和 ACO 方法计算时间为 4.11 和 4.23 s, 计算量相对适中。ACGO 方法的计算时间最长,为 22.50 s, 这是由于其优化过程中涉及较复杂

的遗传-蚁群协同计算。GACO方法的计算时间为6.84 s,相较于ACGO显著降低,同时在路径优化质量

上保持较优性能。因此,GACO方法在本研究的不同规模移栽场景下表现出较强的普适性和平衡性。



A:200-72孔各算法规划路径长度变化对比 Comparison of path length changes of each algorithm planning in 200-72 holes; B:200-105孔各算法规划路径长度变化对比 Comparison of path length changes of each algorithm planning in 200-105 holes; C:200-128孔各算法规划路径长度变化对比 Comparison of path length changes of each algorithm planning in 200-128 holes.

图9 200孔向不同规格穴盘移栽路径优化长度对比

Fig.9 Comparison of optimized lengths of transplanting paths for 200-hole plug trays of different specifications

ACGO算法在某些移栽场景下仍具有一定的优化能力,尤其是在处理中等规模问题时表现良好,但在大孔数向小孔数移栽的场景下,贪婪算法因其局部决策的优势反而在路径优化中表现更好。相比之下,GACO算法凭借其高效的全局搜索和局部优化能力,在不同孔数组合和不同缺苗数下都展现出了更高的普适性和稳定性。因此,在需要处理复杂稀植移栽路径规划问题时,GACO算法显然是一个更为优越的选择。

ACGO算法的计算时间在多穴盘孔场景中大幅增加,这与其结合了遗传算法的全局搜索和蚁群算法的局部优化有关。遗传算法的计算开销较高,再加上蚁群算法在大规模场景下需要频繁进行局部优化,因此导致其时间成本大幅上升。GACO算法的计算时间相较于ACGO在多穴盘孔场景中要低得多,通过遗传算法优化蚁群算法的关键参数,有效减少了蚁群算法在大规模问题中的时间开销。虽然计算时间有所增加,但GACO在大规模问题中的计算效率仍然在可接受范围内。

### 3 讨 论

针对现有研究的不足,本研究提出了一种交互优化路径规划方法,包括ACGO和GACO算法,用于温室穴盘苗稀植移栽路径规划。这2种方法结合了GA算法的全局搜索能力和ACO算法的局部优化优势,通过互补机制克服单一算法的局限,提高路径规划的效率和稳定性。试验结果表明,在72孔穴盘

到32孔穴盘缺苗数为9的条件下,ACGO算法与固定顺序法规划路径长度相比优化幅度为13.9%,GACO算法路径优化幅度为59.3%,极大地提升了移栽效率。在移植钵苗少于目的穴盘孔数的情况下,GrA、GA和ACGO算法的优化路径性能并不优于固定顺序法,而ACO和GACO算法始终展示出较好的路径优化性能。计算时间方面,ACGO算法的平均计算时间为13.50 s,显著较长,反映了在进行更复杂的局部搜索优化时的计算负担。GACO算法的平均计算时间为5.15 s,时间较长,但其提供了最稳定的解决方案。GACO算法在处理时间成本和优化效果间具有有效平衡,其时间成本在可接受的范围内。算法稳定性方面,GACO算法的相对标准差(RSD)最优,RSD中值约为1%,较ACGO算法更加紧凑,显示出卓越的稳定性和可靠性。ACGO算法的RSD值与ACO算法相近,表现出较高的稳定性,说明其在多次试验中的优化结果较为一致。相比之下,GA和GrA算法的RSD值较大,反映出其解的波动性较高,在部分场景下可能存在收敛不稳定的问题。

进一步研究显示,ACGO算法在中等规模(72-32孔)的移栽问题中表现出较好的优化潜力和稳定性,但在200孔移栽至72孔和105孔的场景中,其计算时间显著增加,且优化效果较贪婪算法逊色。这可能是由于ACGO算法在大规模路径优化问题中,局部搜索优化的计算量急剧增加,导致收敛速度下降。相比之下,GACO算法在不同孔数组合和缺苗

数情况下均展现了更高的普适性和稳定性,并且在迭代过程中展现出更快的收敛速度和全局搜索能力。其优化策略使得在面对更复杂的路径规划问题时,能够保持较优的搜索效率,避免局部最优问题。因此,GACO算法适用于更大规模的稀移植裁任务,并具有较好的应用推广价值。

## 参考文献 References

- [1] 刘蒙滋,姜凯,王秀,等.蔬菜自动移栽机研究现状与展望[J].农机化研究,2024,46(10):1-8.LIU M Z,JIANG K,WANG X, et al. Research status and prospect of vegetable automatic transplanter[J].China industrial economics,2024,46(10):1-8 (in Chinese with English abstract).
- [2] 张佳敏,韩孟红,徐强辉,等.蔬菜高效移栽机械化技术探索与优化[J].农机科技推广,2023(4):44-46.ZHANG J M, HAN M H, XU Q H, et al. Exploration and optimization of vegetable efficient transplanting mechanization technology [J]. Agriculture machinery technology extension, 2023 (4) : 44-46 (in Chinese).
- [3] WU X L, GAO D L. A study on greedy search to improve simulated annealing for large-scale traveling salesman problem [C]//Advances in Swarm Intelligence. Cham: Springer International Publishing, 2017:250-257.
- [4] BLUM C. Ant colony optimization: introduction and recent trends[J].Physics of life reviews,2005,2(4):353-373.
- [5] 李腾辉,周德强,何冯光,等.基于遗传算法优化模糊PID的甘蔗收获机切割器控制系统[J].华中农业大学学报,2023, 42(2):243-250.LI T H, ZHOU D Q, HE F G, et al. Control system of sugarcane harvester cutter based on fuzzy PID optimized by genetic algorithm [J]. Journal of Huazhong Agricultural University, 2023, 42(2):243-250 (in Chinese with English abstract).
- [6] IWATA H, NINOMIYA S. AntMap: constructing genetic linkage maps using an ant colony optimization algorithm [J]. Breeding science, 2006, 56(4):371-377.
- [7] 贺磊盈,杨太玮,武传宇,等.基于贪心遗传算法的穴盘苗补栽路径优化[J].农业机械学报,2017,48(5):36-43. HE L Y, YANG T W, WU C Y, et al. Optimization of replugging tour planning based on greedy genetic algorithm [J]. Transactions of the CSAM, 2017, 48(5):36-43 (in Chinese with English abstract).
- [8] LI X J, WANG W B, LIU G H, et al. Optimizing the path of plug tray seedling transplanting by using the improved A\* algorithm [J/OL]. Agriculture, 2022, 12(9) : 1302 [2024-06-30]. <https://doi.org/10.3390/agriculture12091302>.
- [9] TONG J H, QIU Z A, ZHOU H L, et al. Optimizing the path of seedling transplanting with multi-end effectors by using an improved greedy annealing algorithm [J/OL]. Computers and electronics in agriculture, 2022, 201: 107276 [2024-06-30]. <https://doi.org/10.1016/j.compag.2022.107276>.
- [10] LEE Z J, SU S F, CHUANG C C, et al. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment [J]. Applied soft computing, 2008, 8 (1) : 55-78.
- [11] HU J P, YAN X Y, MA J, et al. Dimensional synthesis and kinematics simulation of a high-speed plug seedling transplanting robot [J]. Computers and electronics in agriculture, 2014, 107:64-72.
- [12] 童俊华,蒋焕煜,武传宇.基于贪心算法的温室钵苗稀移植裁路径优化[J].农业机械学报,2016,47(3):8-13.TONG J H, JIANG H Y, WU C Y. Optimization of seedlings lower density transplanting path based on greedy algorithm [J]. Transactions of the CSAM, 2016, 47(3) : 8-13 (in Chinese with English abstract).
- [13] TONG J H, WU C Y, JIANG H Y, et al. Optimizing the path of seedling low-density transplanting by using greedy genetic algorithm [J]. Computers and electronics in agriculture, 2017, 142:356-368.
- [14] 徐守江.基于蚁群算法的穴盘苗自动移钵路径优化[J].江苏农业科学,2015,43(12):475-477. XU S J. Optimization of automatic pot-moving path of plug seedlings based on ant colony algorithm [J]. Jiangsu agricultural sciences, 2015, 43(12):475-477 (in Chinese).
- [15] 童俊华,蒋焕煜,周鸣川.基于遗传算法的穴盘苗自动移钵路径优化[J].农业机械学报,2013,44(4):45-49.TONG J H, JIANG H Y, ZHOU M C. Optimization of seedling transplanting path based on genetic algorithm [J]. Transactions of the CSAM, 2013, 44 (4) : 45-49 (in Chinese with English abstract).
- [16] 冯莉,吕修凯,崔生乐,等.基于双向迪杰斯特拉算法移栽机补栽路径规划及仿真[J].中国农机化学报,2023,44(3) : 177-182.FENG L, LÜ X K, CUI S L, et al. Path planning and simulation of transplanting machine replenishment based on two-way Dijkstra algorithm [J]. Journal of Chinese agricultural mechanization, 2023, 44(3) : 177-182 (in Chinese with English abstract).
- [17] SCHMITT L M. Theory of genetic algorithms[J]. Theoretical computer science, 2001, 259(1/2):1-61.
- [18] BASEL J, WILLEMAIN T R. Random tours in the traveling salesman problem: analysis and application [J]. Computational optimization and applications, 2001, 20(2):211-217.
- [19] DORIGO M, BIRATTARI M, STUTZLE T. Ant colony optimization [J]. IEEE computational intelligence magazine, 2006, 1(4):28-39.

## Optimizing path of sparse planting and transplanting plug seedlings based on genetic-ant colony interactive algorithms

CAI Jimeng, WANG Weibing, QU Jiahao, GUO Xiaolong, LI Guodong, WU Xiaoyu

*College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832000, China*

**Abstract** The ant colony-genetic optimization algorithm (ACGO) and genetic-ant colony optimization algorithm (GACO) interactive algorithms based on genetic algorithm and ant colony algorithm was proposed to optimize the path of sparse planting transplanting to solve the problem of low efficiency in planning the path of transplanting plug seedlings to low-density plug trays. The fixed sequence method and five other algorithms were used to calculate the length of transplanting path for plug trays with holes from 72-32, 72-50, 128-50, and 128-32 holes through simulation experiments. Comparative analyses were conducted on the performance of algorithms in optimizing the length of transplanting path and the time of calculation. The stability of the algorithm was evaluated by the relative standard deviation. The results showed that the GACO algorithm reduced the average length of path in transplanting plug tray with 72 to 32 holes by 59.3% compared with the fixed sequence method, with an average time of calculation of 5.15 seconds and a relative standard deviation of approximately 1.5%. The ACGO algorithm reduced the average length of path by 19.2%, with an average time of calculation of 13.50 seconds and a relative standard deviation of approximately 1%. The results of further studies showed that the optimization effect of ACGO algorithm in the scenarios of transplanting from 200 holes to 72 holes and 105 holes was weaker than that of greedy algorithm, while GACO algorithm had higher universality and stability under different combinations of hole numbers and numbers of missed seedling. It is indicated that both interactive algorithms have improved the performance of the original algorithm, but the GACO algorithm performs better in dealing with problems in planning path of complex sparse planting and transplanting plug seedlings. It will provide a strong reference basis for optimizing the path of sparse planting and transplanting plug seedlings.

**Keywords** plug seedlings; sparse planting and transplanting; path optimization; interactive algorithm; genetic-ant colony optimization algorithm

(责任编辑:陆文昌)