

一种改进蚁群算法的无人机避险方法仿真研究

吴学礼^{1,2}, 贾云聪¹, 张建华^{1,2}, 甄 然^{1,2}

(1.河北科技大学电气工程学院,河北石家庄 050018;2.河北省生产过程自动化工程技术研究中心,河北石家庄 050018)

摘要:随着低空空域的逐渐开放以及无人机产业的高速发展,无人机数量不断上升,无人机间随时有发生冲突的可能,需要一种可靠的冲突解脱技术使无人机可以避免危险。针对无人机冲突解脱问题,提出基于改进蚁群算法的无人机冲突解脱方法:采用参数自适应调整策略,根据解的质量,动态调整参数值,防止算法早熟,提高收敛精度;在算法状态转移规则中引入扰动因子,加快算法初期收敛。算法测试实验结果显示,改进蚁群算法收敛精度更高。仿真实验表明,改进算法可以帮助两无人机及时脱离危险。该算法作为一种通用优化算法,也可应用到目标识别、路径规划等问题中,具有重要的研究意义与广泛的应用价值。

关键词:机器人控制;无人机;冲突解脱;蚁群算法;参数自适应调整;扰动因子

中图分类号:TP273 文献标志码:A

Simulation study of UAV conflict resolution based on an improved ant colony algorithm

WU Xueli^{1,2}, JIA Yuncong¹, ZHANG Jianhua^{1,2}, ZHEN Ran^{1,2}

(1.School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, Hebei 050018, China; 2. Hebei Provincial Research Center for Technologies in Process Engineering Automation, Shijiazhuang, Hebei, 050018, China)

Abstract: With the gradual opening of the low-altitude airspace and the rapid development of Unmanned Aerial Vehicle(UAV) industry, the users of UAV are increasing continuously and the conflicts could occur at any time. It is necessary to develop a reliable UAV conflict resolution algorithm to avoid the danger. This paper proposes an UAV conflict resolution algorithm based on the improved ant colony algorithm with two advantages. Firstly, the algorithm adopts adaptive parameters adjustment strategy, which adjusts the parameters value dynamically according to the quality of the solution, prevents the algorithm premature convergence and improves the accuracy. In addition, the disturbance factors is introduced to the state transition rules of random selected path in order to accelerate the initial convergence. The simulation results have shown that the improved algorithm displays obvious superiority in convergence precision, helping the two UAVs avoiding dangers in time. The algorithm described in this paper could be applied to target identification, path planning and other issues as a general optimized algorithm, which is of

收稿日期:2017-12-16;修回日期:2018-03-06;责任编辑:李 穆

基金项目:河北省自然科学基金(F2015208128, F2014208119);河北省发改委项目(9130002017001);河北省科技厅项目(17212102D)

第一作者简介:吴学礼(1961—),男(满族),黑龙江齐齐哈尔人,教授,博士,主要从事控制科学与工程方面的研究。

E-mail:wuxueli@hebust.edu.cn

吴学礼,贾云聪,张建华,等.一种改进蚁群算法的无人机避险方法仿真研究[J].河北科技大学学报,2018,39(2):166-175.

WU Xueli, JIA Yuncong, ZHANG Jianhua, et al.Simulation study of UAV conflict resolution based on an improved ant colony algorithm [J].Journal of Hebei University of Science and Technology,2018,39(2):166-175.

great significance and wide application.

Keywords: robot control; UAV; conflict resolution; ant colony optimization; parameter adaptive adjustment; disturbance factor

近年来,由于无人机应用日趋广泛,致使低空空域变得越来越拥堵,无人机的飞行安全受到威胁。避障系统对确保无人机安全自主地飞行,提高空域流量,高效完成任务有着重要的现实意义。冲突解脱算法是无人机避障系统的核心,目前国内外对无人机冲突解脱算法的研究还并不完善,如何在代价最小的情况下实现无人机的冲突解脱,有待进一步研究。

当前,有许多传统方法解决无人机冲突解脱问题,如 Petri 网络算法^[1]和基于数据融合的模糊规划^[2]等,但是这些方法的计算效率通常较低,因此,国内外学者提出了几种基于元启发式算法的解决问题方案:文献[3—4]采用了遗传算法,但是遗传算法在搜索解脱路径时,虽然初期收敛较快,但收敛精度低,导致解脱路线不够优化;文献[5]采用了模拟退火算法,该算法稳定性高,但是寻得高质量近似解所花费时间较多;文献[6]采用了粒子群算法,该算法虽然收敛较快,但是容易陷入早熟,并且局部寻优能力较差;文献[7—8]采用了基本蚁群算法,从中可以看到蚁群算法同样存在易陷入早熟,且算法前期由于信息素匮乏,导致初期收敛速度慢的缺点,但是与其他几种智能算法对比,蚁群算法性能更均衡,后期搜索速度快并具有很强的鲁棒性,可以更好地应对动态问题并且蚁群算法易与其他算法或者策略相结合,以改善算法性能。

本文主要针对无人机避障算法进行研究,针对蚁群系统算法结果易早熟,收敛精度不高等问题,在现有研究基础上提出了一种改进算法,证明了算法的收敛性,并将其应用于两无人机冲突解脱问题,在保证无人机安全飞行的基础上,减少因冲突带来的消耗,通过 Matlab 仿真验证算法的有效性。

1 蚁群算法及拓展

1.1 基本蚁群算法

蚁群算法(ant colony optimization, ACO)是由 DORIGO 等^[9]首次提出的,蚁群中每个蚂蚁信息互通,每个蚂蚁的信息在蚁群内部不断更新并互相学习交流,使整体不断优化,算法通过外激素的浓度对最优解进行筛选^[10]。

问题的解在蚁群中通过不断迭代逐步被优化。对于各种复杂优化问题,蚁群算法通过定义解的分量来解决。开始时,解的分量对应于每只蚂蚁,备选解以重复增加解的分量的方式来产生^[11]。蚂蚁通过状态转移规则和正反馈机制,当处于选择点上时,决定哪个解的分量要加到它的当前部分解。构造完一个完整的解后,解的分量会被蚂蚁涂上信息素,指导接下来整体和个体的活动^[12]。

ACO 中包含了多个蚂蚁,每个蚂蚁的行为由状态转移规则和信息素更新规则决定。

1) 状态转移规则:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{r \in \text{allowed}_k} \tau_{ir}^\alpha(t) \eta_{ir}^\beta(t)}, & j \in \text{allowed}_k; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

2) 信息素更新规则:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t), \quad (2)$$

$$\Delta \tau_{ij}^k = \begin{cases} Q/C^k(t), & k \in [i, j]; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

式(1)中, $\tau_{ij}(t)$ 表示在第 t 次迭代时,路径 (i, j) 上的信息素浓度, $\eta_{ij}(t) = 1/d_{ij}$ 为启发函数,表示蚂蚁从 i 走到 j 的期望程度; α 和 β 分别为累计信息重要程度因子和启发函数重要程度因子; allowed_k 表示蚂蚁 k 下一步允许访问节点的集合。式(2)中 ρ 为信息素的保留比例,是一个取值范围在 0 和 1 之间的常数。 $\Delta \tau_{ij}^k$ 为蚂蚁 k 在 t 次迭代中,若经过边 (i, j) ,则在 (i, j) 上增加的信息素量,其值由式(3)确定,其中 Q 是一个常数, $C^k(t)$ 是蚂蚁 k 在 t 次迭代时所走的长度^[13]。

1.2 蚁群算法的拓展

虽然 ACO 中的正反馈机制能使较好的解得到不断优化,但是由于算法初期信息素积累差异不明显,导致蚁群算法初期收敛速度较慢,并且对于规模较大的问题,ACO 容易出现停滞现象,不能对解空间进行进一步搜索,不利于发现更好的解,为了解决这些问题,国际上一些专家提出了多种方法对蚁群算法进行改进,期望提高算法收敛速度和精度,改善停滞问题。

在文献[14]中,蚁群算法开创者 DORIGO 对基本蚁群算法进行改进,提出了蚁群系统算法(ant colony system, ACS)。ACS 对基本蚁群算法的状态转移规则和信息素更新规则进行了改进。对状态转移规则的改进如式(4)所示。

$$s = \begin{cases} \arg \max_{j \in \text{allowed}_k} \{ \tau(i, j) \cdot \eta(i, j) \}, & \text{if } q \leq q_0; \\ p_{ij}^k, & \text{else.} \end{cases} \quad (4)$$

将常量 q_0 加入到状态转移规则中,采用贪婪的路径选取方式还是随机探索新路径的主次关系由 q_0 大小决定,其是一个固定值,根据经验取 0.9, q 是在 $[0, 1]$ 区间均匀分布的随机数,在第 t 次迭代时大概率选择 $[\tau(i, j)] \cdot [\eta(i, j)]$ 最大的节点,这就使蚂蚁探索范围变小,减少前期算法搜索时间,并且使用全局与局部信息素双重更新规则。

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}^{gb}(t), \quad (5)$$

$$\tau_{ij} = (1 - \xi) \cdot \tau_{ij} + \xi \cdot \tau_0. \quad (6)$$

式中: $\Delta\tau_{ij}^{gb}(t) = (C_{gb})^{-1}$, $(C_{gb})^{-1}$ 为当前全局最优路径长度; τ_0 是每条边的初始信息素; ρ 和 ξ 均为信息素挥发系数,是取值范围在 $[0, 1]$ 之间的常数。

首先,对截止当前迭代次数时,算法所找到的属于全局最优解的路径进行更新,如式(5)所示;然后,对蚂蚁走过的,不属于全局最优解的路径进行局部信息素更新,如式(6)所示。在 ACS 中,通过设定此种优先级,即只有构成全局最优路径的那些才有机会增加其信息素水平,其他边的信息素,则由于挥发作用逐渐降低,那些属于最佳路径的信息素水平就会明显高于其他的边,对找寻最优路径就变得目的性更强。局部外激素更新规则采用负反馈,目的是降低已经搜索过的路径被选择的机会,降低早熟现象发生概率。每当一只蚂蚁由一个节点移动到另一个节点时,该路径上的信息素都按照式(6)被相应地消除一部分,实现一种信息素的局部调整。但是由于信息素的全局更新作用,再经过几次搜索以后,所有属于最佳路径的边,其信息素水平远远高于其他边相差一个数量级。因此,信息素的局部更新作用不能有效地阻止搜索陷入局部最优化。另外,由于信息素的局部更新在每一步搜索之后都要进行,因此,消耗了大量的计算时间。

另外,在蚁群算法初始化参数值时,一般都是凭经验设置。经过国内外专家多次研究测试发现,参数 α 和 β 对蚁群算法中蚂蚁行为有重要影响,蚂蚁的行为强烈依赖于给定参数的值。

文献[15]针对 ACS 进行了详细分析,发现无论在寻优能力还是寻优速度上,ACS 与基本蚁群算法相比都有明显的提升,并且通过 TSP 中 Eil51 问题对 ACS 进行大量实验测试,确定了 ACS 参数的最佳取值范围: $\alpha \in [1.0, 3.0]$, $\beta \in [2.0, 4.0]$, $\rho \in [0.5, 0.8]$,但是 ACS 在解决实际问题中,仍然存在易停滞,收敛速度不理想的情况。

文献[16]提出了参数 α, β 自适应调整策略,通过建立 α 与 β 的互锁关系,即 $\alpha + \beta = M$, M 为固定值,针对不同迭代次数 NC , 2 个参数会对应取不同的值,以应对不同时期算法不同的特点,如式(7),式(8)所示:

$$\alpha = \begin{cases} \alpha_1, & NC \leq NC_1; \\ \alpha_2, & NC_1 < NC \leq NC_2; \\ \alpha_3, & NC_2 < NC \leq NC_{\max}; \end{cases} \quad (7)$$

$$\beta = \begin{cases} \beta_1, & NC \leq NC_1; \\ \beta_2, & NC_1 < NC \leq NC_2; \\ \beta_3, & NC_2 < NC \leq NC_{\max}. \end{cases} \quad (8)$$

文献[17]通过分析研究认为,可以在算法初期对参数 α 与 β 设置相对较小的值,扩大算法初期搜索范围,当算法在寻得当前最优解后 N 次循环没有变化则认为算法陷入局部最优,此时 α 与 β 按照式(9)、式(10)进行调整。

$$\alpha(t+1) = \begin{cases} \xi_1 \alpha(t), & \text{if } \xi_1 < \alpha_{\max}; \\ \alpha_{\max}, & \text{otherwise;} \end{cases} \quad (9)$$

$$\beta(t+1) = \begin{cases} \xi_2 \beta(t), & \text{if } \xi_2 < \beta_{\max}; \\ \beta_{\max}, & \text{otherwise.} \end{cases} \quad (10)$$

式中: ξ_1, ξ_2 均为大于 1 的常数; $\alpha_{\max}, \beta_{\max}$ 为 α 和 β 所能达到的上界。

以上 2 种参数自适应方法虽然在一定程度上改善了算法性能,但是 2 种都各自增加了 2 个参数的设定,使算法产生了更多不确定因素。

2 改进的蚁群算法

蚁群算法有鲁棒性强、全局搜索、并行分布计算、易与其他问题结合等优点,并在实践中体现出了优越性,蚁群算法是解决飞机避障,路径规划及 TSP 问题的较好的优化算法之一,但是仍然存在算法初期收敛慢,易停滞陷入早熟等问题。在对第 1 部分所提及的方法进行分析与研究的基础上,本文将 ACS 与自适应参数调节进行融合,并且引入扰动因子,提出了一种新的改进蚁群算法。

2.1 算法的改进

2.1.1 参数自适应调节策略

α 和 β 是 2 个重要参数, α 是轨迹相对重要性,反映了蚂蚁在寻优过程中积累信息素,指导后来蚂蚁们在搜索中的相对重要程度,代表了蚂蚁在搜索路径中随机作用大小, α 越大搜索随机性越弱,越小则易陷入盲目搜索。 β 代表能见度相对重要性,反映了优化过程中的确定因素,是先验知识在指导蚂蚁搜索过程中的相对重要程度, β 越小搜索随机性越大, β 越大收敛速度加快但易陷入局部最优。 α 和 β 的取值是至关重要的,直接影响算法性能。

针对此问题,根据自然选择规则,在算法的状态转移规则中对参数 α 和 β 采用自适应参数适配方法,构造参数解集 $\Phi_k(t) = [\alpha_k(t), \beta_k(t)]$,更新公式如式(11)、式(12)所示:

$$\Phi_{\text{winner}}(t+1) = [\alpha_{\text{winner}}(t) + \Delta, \beta_{\text{winner}}(t)]; \quad (11)$$

$$\Phi_{\text{loser}}(t+1) = [\alpha_{\text{winner}}(t), \beta_{\text{winner}}(t)]. \quad (12)$$

每次迭代之后,该方法评估每只蚂蚁构建路径的质量,如果某只蚂蚁找到了当前迭代的最优路径,结构参数根据式(11)更新,而式(12)用于找到了最差路径的蚂蚁。也就是说,找到当前迭代最优路径的蚂蚁会被奖励,找到最差路径的蚂蚁会被惩罚,根据获胜者失败者反馈机制,在这个机制下,只要蚂蚁找到当前最佳路径, α 的幅度就会增加 $\Delta = 0.01$,随着算法的进行,获胜的蚂蚁对信息素浓度的感应变得更加敏感,并产生具有相同结构参数值的后代,而作为失败者的蚂蚁被获胜者的新生代所代替。在整个蚂蚁种群的 α 值可以增加的情况下,本文只允许获胜者蚂蚁增加 α 的幅度,以加强定向搜索。

文献[15]中大量实验测试结果表明,当 $\alpha \geq 1$ 作为安全下限时,算法能够达到令人满意的性能,并且 [2.0, 4.0] 是 β 的合理范围,因此,在所提出的改进算法中,每个蚂蚁在开始构造路径时具有相同初始值 $\alpha = 1$,同时 β 的值是在区间 [2.0, 4.0] 中随机的,并且相同的 α 起始值允许蚂蚁通过识别可能导致更优解 β 的值进行公平的竞争。

算法利用自然选择规则来寻找构造参数值的自适应趋势,从而能够识别出合适的值,在整个搜索过程中以这种方式逐渐地增加 α 的值,使得蚂蚁在开始阶段继续探索搜索空间(对应于较小的 α),有助于防止算法提前收敛到搜索空间的小区域。另外,逐渐增加参数 α 的值将强调更好的搜索空间区域,所以,搜索将被引导到后期的更优的路径(对应于大的 α 值),提高算法的收敛精度。

2.1.2 引入扰动因子

在 ACS 中, q_0 是状态转移规则控制参数,决定了 ACS 以贪婪方式还是随机方式选择路径的概率,一般取固定值 0.9,在算法初期可以大概率通过贪婪方式去选择路径,有助于前期信息素积累,加快前期收敛速度。但是由于算法前期收敛加快,导致算法容易停滞并且采用随机选择方式去拓展新路径概率较低,减弱了参数自适应调节的作用。

针对这个问题,本文在 ACS 的基础上,引入扰动因子 w_0 代替 q_0 , w_0 具体公式如式(13)所示:

$$w_0 = \frac{1}{e^{t/b}}, \quad n = 1, 2, \dots, NC, \quad (13)$$

式中: t 为迭代次数; b 为正的常数;在算法刚开始时, w_0 大概率大于 q ,所以算法前期路径以贪婪选择方式

为主,使信息素可以迅速积累,加速算法前期收敛,随着迭代次数增加, ω_0 小于 q 的概率变大,随机选择路径规则为主,此时参数自适应调节机制开始发挥作用,当 α 较小时,算法扩展搜索范围,丰富路径多样性,防止因前期加速收敛可能会导致的早熟问题,当 α 逐渐增大,定向搜索加强,所以算法收敛精度得以提升。这样将概率合理分配于各阶段,实际上是加速算法收敛,增加了改进算法的全局搜索能力,使算法具备了跳出早熟的能力。

则改进算法的状态转移规则为

$$s = \begin{cases} \arg \max_{j \in \text{allowed}_k} \{ \tau(i, j) \cdot \eta(i, j) \}, & \text{if } q \leq \omega_0; \\ p_{ij}^k, & \text{if } q > \omega_0; \end{cases} \quad (14)$$

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha_k(t)} \cdot [\eta_{ij}(t)]^{\beta_k(t)}}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^{\alpha_k(t)} \cdot [\eta_{is}(t)]^{\beta_k(t)}}, & j \in \text{allowed}_k; \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

2.2 改进算法收敛性证明

由于蚁群算法具有很强的鲁棒性,所以,本文对改进的蚁群算法进行收敛性证明。

证明思路如下:首先证明对于任意迭代次数 t ,搜索区域 A 中任意一段路径 $(i, j) \in A$ 上的信息素浓度是有界的,接下来再证明改进的算法能以任意趋近于 1 的概率找到最优路径。

命题:对任意迭代次数 n 中的任意路径 $(i, j) \in A$, $\tau_{ij}(n)$ 存在上界 τ_{\max} 和下界 τ_{\min} , 即:

$$\tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max}, \quad (16)$$

式中, $\tau_{ij}(t)$ 是路径 (i, j) 上的信息素浓度。

证明:全局信息素更新规则和局部信息素更新规则可以被写成如式(17)形式:

$$a_{h+1} = (1 - \varphi) \cdot a_h + \varphi b, \quad h \geq 1, \quad (17)$$

式中, a_{h+1} 和 a_h 代表 $\tau_{ij}(t+1)$ 和 $\tau_{ij}(t)$, 相应的, $b = g(s^*)$, τ_0 , $g(s^*)$ 为全局最优解的信息素浓度, $\varphi = \rho$, ξ 。那么可以容易的通过归纳得到:

$$a_h = (1 - \varphi)^{(h-1)} a_1 + \sum_{i=0}^{h-2} (1 - \varphi)^i \varphi b = (1 - \varphi)^{(h-1)} a_1 + b[1 - (1 - \varphi)^{h-1}], \quad (18)$$

因为, $1 > \varphi > 0$, 所以随着 $h \rightarrow \infty$, a_h 趋近于 b 。

下面,就序列 $\{a_h\}$ 的单调性进行讨论, a_h 的导数为

$$a'_h = (a_1 - b)(1 - \varphi)^{(h-1)} \ln(1 - \varphi), \quad (19)$$

因为 $\ln(1 - \varphi) < 0$, 在算法开始时,根据信息素更新规则,信息素浓度的值 $\tau_{ij} = \tau'$ 肯定要比 τ_0 大,但是比全局最优路径信息素浓度 $g(s^*)$ 小,并且局部信息素更新规则会通过更新逐步降低蚂蚁访问过边的浓度。因此假设在最坏的情况下,边 (k, l) 一直在蒸发信息素,从没有获得过加强,那么信息素的浓度也绝不会小于 τ_0 , 所以信息素最小值为 $\tau_{\min} = \tau_0$, 所以 $\tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max} = g(s^*)$ 。

证毕。

定理 令 $P^*(n)$ 为算法在前 n 次迭代中至少找出 1 个最优解的概率,则对于一个绝对小的 $\varepsilon > 0$ 而言,当 n 足够大时,有:

$$P^*(n) \geq 1 - \varepsilon, \quad (20)$$

$$\lim_{n \rightarrow \infty} P^*(n) = 1. \quad (21)$$

证明:根据伪随机比例法则,假设路径 (i, j) 没有相关的最大信息素路径,选择路径 (i, j) 的概率为随机选择路径的概率,即 $P(q > q_0) \cdot p_{ij}$, 所以在任意 1 次迭代 n 中,做出任何特定选择概率为

$$P_{ij}(n) = P(q > q_0) \cdot \frac{[\tau_{ij}(n)]^\alpha [\eta_{ij}]^\beta}{[\sum_{(i,j)} \tau_{ij}(n)]^\alpha [\eta_{ij}]^\beta} \geq P(q > q_0) \cdot \left(\frac{\gamma}{\Gamma}\right)^\beta \cdot \left[\frac{\tau_{ij}(n)}{\sum_{(i,j)} \tau_{ij}(n)}\right]^\alpha \geq P(q > q_0) \cdot \left(\frac{\gamma}{\Gamma}\right)^\beta \cdot \left[\frac{\tau_{\min}}{|V| \tau_{\max}}\right]^\alpha = P'_{\min}, \quad (22)$$

式中, $|V|$ 表示固定节点的可行后继的最大数目。任何产生的解 s' , 包括最优解 $s^* \in S^*$, 生成的概率为

$$\hat{P} \geq (P'_{\min})^t > 0. \quad (23)$$

式中 $l < +\infty$ 为序列的最大长度。所以,

$$P^*(n) \geq 1 - (1 - \hat{P})^n, \tag{24}$$

因此 $\lim_{n \rightarrow \infty} P^*(n) = 1$, 此外, 对于任意小的 $\epsilon > 0$ 而言, 当 n 足够大时, 有:

$$P^*(n) \geq 1 - \epsilon, \tag{25}$$

证毕。

2.3 改进蚁群算法的性能分析

TSP 问题是最经常研究的组合优化问题之一^[18], 它是一种 NP 难题但有一个相对简单的定义, 使人们能够将注意力集中在算法上, 因此 TSP 问题事实上已经成为测试启发式算法的标准问题。本文依然选择 TSP 问题作为测试改进蚁群算法性能的工具。

根据文献[19]所述, 最佳性能指标为评价蚁群算法优劣程度的基本指标。最佳性能指标由相对误差 E_o 定义, 其公式如式(26)所示:

$$E_o = \frac{g_b - g^*}{g^*} \times 100\%, \tag{26}$$

式中: g_b 为算法经过多次运行得到的最优化的值; g^* 为所解决的问题理论上的最优值。最佳性能指标用来衡量蚁群算法对问题的最佳优化程度, 其值越小意味着算法优化性能越好, 收敛精度越高。

下面将 ACS 与改进的蚁群算法分别求解 TSP 问题中的 Oliver30 问题和 Eil51 问题, 并对其性能进行比较分析。其中 2 种算法初始参数设置见表 1。

表 1 2 种算法初始参数设置

Tab.1 Initial parameter settings for both algorithms

问题	算法	α	β	ξ	ρ	q_0	m	NC	τ_0
Oliver30	ACS	1	3	0.5	0.6	0.9	20	500	0.01
Oliver30	改进蚁群	1	[2,4]随机	0.5	0.6	$(1/e)t$	20	500	0.01
Eil51	ACS	1	3	0.5	0.6	0.9	50	1 000	0.01
Eil51	改进蚁群	1	[2,4]随机	0.5	0.6	$(1/e)t$	50	1 000	0.01

分别用 ACS 与改进蚁群算法求解 Oliver30 问题和 Eil51 问题, 各运行 10 次。在 10 次运行中, ACS 对 Oliver30 问题求解的最优路径仿真图以及最优路线长度如图 1、图 2 所示, 用改进蚁群算法求解的 Oliver30 问题的最优路径以及最优路线长度如图 3、图 4 所示。

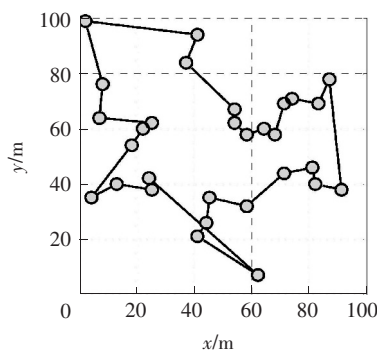


图 1 ACS 求解 Oliver30 最优路径

Fig.1 Optimal path of Oliver30 solved by ACS

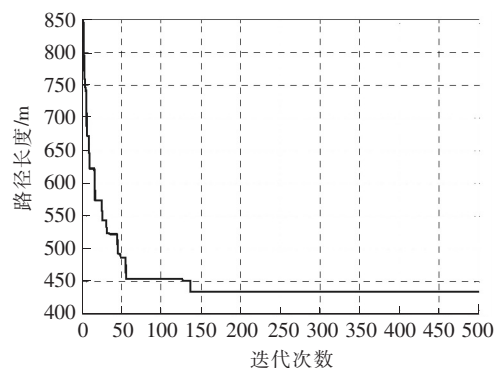


图 2 ACS 求解 Oliver30 最优路径长度

Fig.2 Optimal path's length of Oliver30 solved by ACS

从图 2 中可以看出 ACS 求解 Oliver30, 在算法前期收敛较快, 但是在迭代次数 137 次时算法就停止继续收敛, 而对于改进的蚁群算法 (见图 4), 则没有出现相较于 ACS 过久的停滞, 算法在 95 次迭代以后仍然逐步向最优解收敛。对于解决规模更大的 Eil51 问题, ACS 的最优路径与路线长度如图 5、图 6 所示, 改进蚁群算法如图 7、图 8 所示。

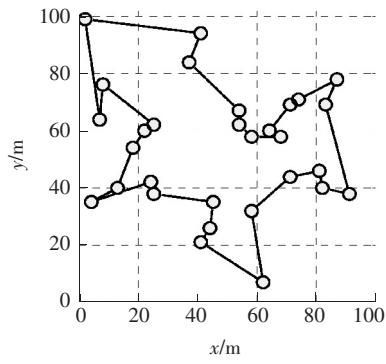


图3 改进蚁群算法求解 Oliver30 最优路径

Fig.3 Optimal path of Oliver30 solved by improved ant colony algorithm

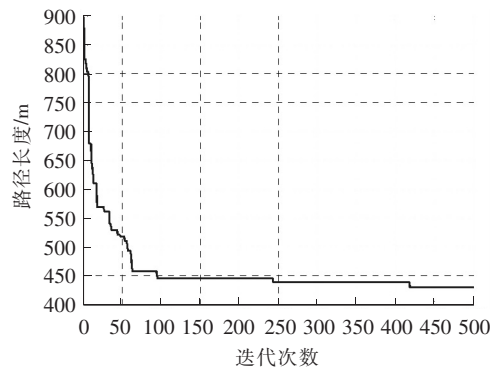


图4 改进蚁群算法求解 Oliver30 最优路径长度

Fig.4 Optimal path's length of Oliver30 solved by improved ant colony algorithm

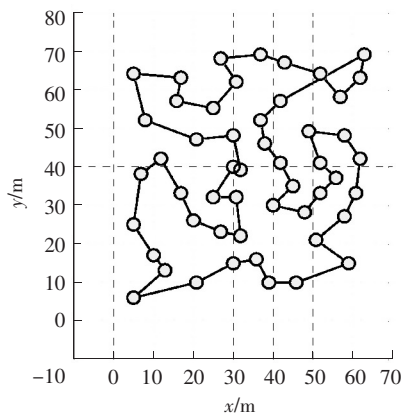


图5 ACS 求解 Eil51 最优路径

Fig.5 Optimal path of Eil51 solved by ACS

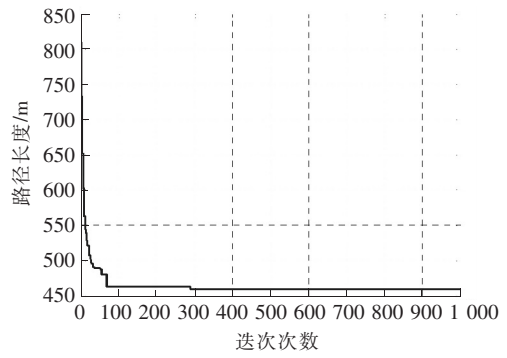


图6 ACS 求解 Eil51 最优路径长度

Fig.6 Optimal path's length of Eil51 solved by ACS

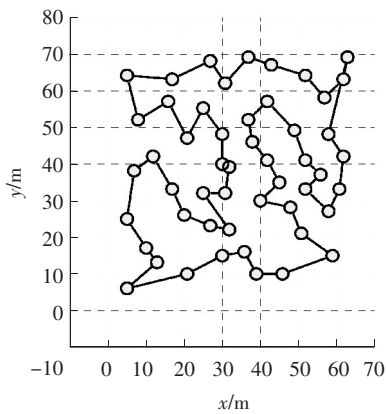


图7 改进蚁群算法求解 Eil51 最优路径

Fig.7 Optimal path of Eil51 solved by improved ant colony algorithm

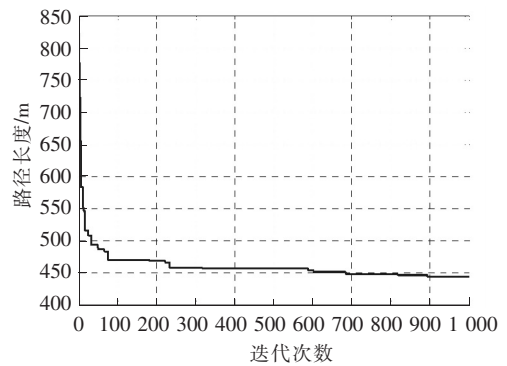


图8 改进蚁群算法求解 Eil51 最优路径长度

Fig.8 Optimal path's length of Eil51 solved by improved ant colony algorithm

通过图6和图8对比,可以更明显地看出,改进蚁群算法在规定迭代次数内并没有出现明显停滞,相较于ACS,改进蚁群算法对Eil51问题规划出的路径更合理,路径长度更短。实验详细测试对比结果见表2。

表 2 2 种算法性能测试结果

Tab.2 Test results of the performance of the two algorithms

问题	最优值/m	ACS 最优值/m	改进算法最优值/m	ACS 相对误差/%	改进算法相对误差/%
Oliver30	423.735 6	433.105 2	427.466 6	2.211 2	0.880 5
Ei151	426	458.306 2	442.129 0	7.583 6	3.786 2

从表 2 中可以看出,随着解决问题规模的扩大,算法的相对误差都有上升,对应算法性能都出现一定程度的下降,但是改进的蚁群算法得出的最优路径长度与相对误差方面均小于 ACS,即改进过后的算法与 ACS 相比性能要更好,是一种有效的改进算法。

3 冲突解脱问题建模及仿真模拟

3.1 冲突解脱问题建模

规划避险路径是无人机避险问题的目的,并且要使无人机的避险路径与原航迹相比延误距离之和最小,从而实现航迹最短。在规划避险路径之前,将两冲突无人机航线分别分成 k 等份,所以,定义优化目标函数为

$$f = \min(\sum_{i=1}^k d_{Ai} + \sum_{i=1}^k d_{Bi}), \tag{27}$$

式中, $i=1,2,\dots,k$, d_{Ai} 与 d_{Bi} 为无人机 A 和 B 在第 i 步时,与原飞行路线的距离。并且为了确保两无人机不相撞,在任意时刻无人机 A 和 B 之间的距离满足:

$$\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \geq r_{eq}, \tag{28}$$

式中, r_{eq} 为无人机间最小间隔,由于各种无人机性能差别较大,所以根据每架无人机性能特点,制定了适合无人机的最小安全间隔,具体公式如式(29)所示:

$$r_{eq} = \sqrt[3]{\frac{V_1(V_f V_a + V_f V_d + V_b V_a + V_b V_d)}{4}} \times \tau, \tag{29}$$

式中: V_f 为最大前进速度; V_b 为最大后退速度; V_a 为最大垂直上升速度; V_d 为最大下降速度; V_1 为最大水平横向速度; τ 为监视系统刷新速率^[20]。

3.2 仿真验证

本文对两无人机的避险问题进行 Matlab 仿真,以在同一海拔平面为例。首先构建一个 200 m × 200 m 的区域,并设置无人机 A 和 B 的起始点分别为(100,0)和(0,100),它们分别从各自起点向对向飞行。假设两无人机性能相同,若各自均保持匀速,直线飞行,那么两无人机将在(100,100)处相冲突,如图 9 所示。

以对避险算法要求较高森林防火无人机为例,算法中,参数分别初始化为 $\alpha = 1, \xi = 0.5, \rho = 0.6, m = 60, NC = 100, \tau_0 = 0.01$ 。本文采用基于 ADS-B 制式的无人机监视设备,刷新频率为 0.5 s/次,森林防火无人机其最大水平飞行速度为 $V_f = V_b = V_1 = 45$ m/s,最大上升下降速度 $V_a = V_d = 30$ m/s,所以根据式(29)可知,两无人机间最小安全间隔 $r_{eq} = 19.655 6$ m,为方便计算,取整为 20 m。将本文所提出的改进蚁群算法应用到无人机避险问题中,进行 Matlab 仿真验证。

改进蚁群算法的冲突解脱轨迹与两无人机距离如图 10、图 11 所示。从图 10 中可以看出,无人机 A 由于改进蚁群算法,调整了航迹,由图 11 可知,两无人机在开始时逐渐接近,在第 9 步时达到了最小安全距离,但是随后成功实现解脱,避免了相撞的危险。

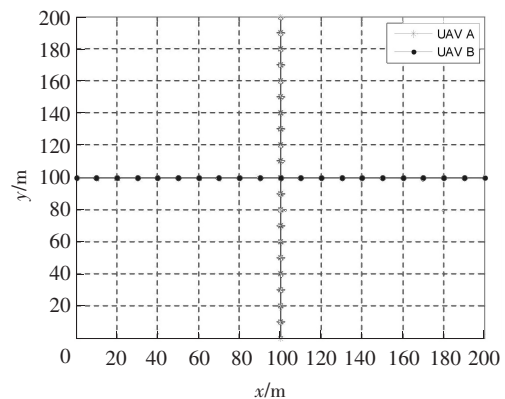


图 9 无人机原航路
Fig.9 UAV original route

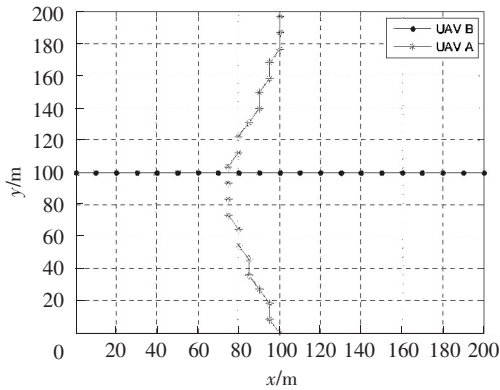


图 10 改进蚁群算法调整后的路径

Fig.10 Improved ant colony algorithm adjusted path

图 12 为在冲突解脱算法过程中迭代次数与两无人机延误距离之间的关系,可以看出算法在保证两机大于最小安全间隔的基础上,在第 34 次迭代时,能够找到延误距离相对较小的路径,减少了因两无人机冲突带来的飞行消耗。仿真结果表明改进蚁群算法能够较好地解决两无人机冲突解脱问题。

4 结 语

针对无人机冲突解脱问题在 ACS 的基础上提出一种改进的蚁群算法,对算法参数采用自适应调整策略,算法根据解的质量,动态调整参数值,使算法具备了跳出早熟的能力,提高了收敛精度;在算法状态转移规则中引入扰动因子,加快算法前期收敛速度。证明了改进算法的收敛性,并采用 Oliver30 和 Eil51 两个问题对 ACS 和改进的蚁群算法进行了测试对比,测试结果表明改进算法在最佳性能方面有了提升。将改进的算法应用到无人机冲突解脱中进行 Matlab 仿真,仿真结果表明改进的蚁群算法成功使 2 架即将发生碰撞的无人机脱离危险。本文仅研究了两无人机冲突解脱情况,今后将对多无人机冲突解脱问题进行更加深入的研究。

参考文献/References:

- [1] MOHANTA J C, PARHI D R, PATEL S K. Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation [J]. Computers & Electrical Engineering, 2011, 37(6):1058-1070.
- [2] 蒋忠中, 汪定伟. 物流配送车辆路径优化的模糊规划模型与算法[J]. 系统仿真学报, 2006, 18(11):3301-3304.
JIANG Zhongzhong, WANG Dingwei. Fuzzy programming model and algorithm of logistics distribution vehicle routing problem[J]. Journal of System Simulation, 2006, 18(11):3301-3304.
- [3] ROBERGE V, TARBOUCHI M, LABONTE G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning[J]. IEEE Transactions on Industrial Informatics, 2012, 9(1):132-141.
- [4] TSAI C C, HUANG H C, CHAN C K. Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation[J]. IEEE Transactions on Industrial Electronics, 2011, 58(10):4813-4821.
- [5] 裴志刚, 李华星, 王庆胜. 模拟退火遗传算法在飞行冲突解脱中的应用[J]. 交通信息与安全, 2005, 23(1):115-117.
PEI Zhigang, LI Huaxing, WANG Qingsheng. Application of simulated annealing genetic algorithm in flight conflicts resolution[J]. Computer and Communications, 2005, 23(1):115-117.
- [6] 甄然, 司超, 吴学礼, 等. 基于改进粒子群算法的飞行器冲突解脱方法研究[J]. 河北科技大学学报, 2016, 37(5):491-496.
ZHEN Ran, SI Chao, WU Xueli, et al. Aircraft conflict relief method based on improved particle swarm algorithm research[J]. Journal of Hebei University of Science and Technology, 2016, 37(5):491-496.

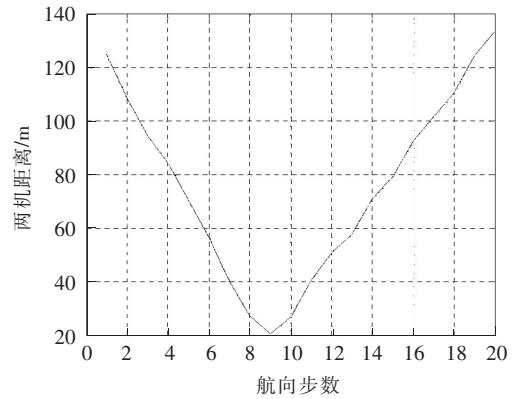


图 11 两无人机间距离

Fig.11 Distance between two UAVs

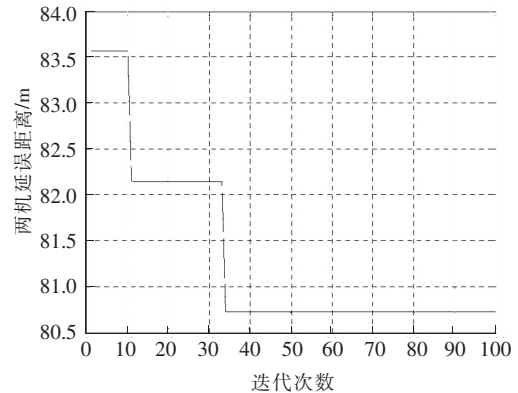


图 12 延误距离随着迭代次数的变化

Fig.12 Delay distance varies with the iterations

- [7] ZHANG W, GONG X, HAN G, et al. An improved ant colony algorithm for path planning in one scenic area with many spots[J]. IEEE Access, 2017,5:13260-13269.
- [8] LISSOVOI A, WITT C. Runtime analysis of ant colony optimization on dynamic shortest path problems[J]. Theoretical Computer Science, 2015, 561:73-85.
- [9] DORIGO M, BIRATTARI M, STUZLE T. Ant colony optimization[J]. Computational Intelligence Magazine IEEE, 2004, 1(4):28-39.
- [10] GUO P, ZHU L. Ant colony optimization for continuous domains[C]// Eighth International Conference on Natural Computation. [S.l.]: IEEE, 2012:758-762.
- [11] YU B, YANG Z Z, YAO B. An improved ant colony optimization for vehicle routing problem[J]. European Journal of Operational Research, 2009, 196(1):171-176.
- [12] 屈鸿, 黄利伟, 柯星. 动态环境下基于改进蚁群算法的机器人路径规划研究[J]. 电子科技大学学报, 2015, 44(2):260-265.
QU Hong, HUANG Liwei, KE Xing. Research of improved ant colony based robot path planning under dynamic environment[J]. Journal of University of Electronic Science and Technology of China, 2015, 44(2):260-265.
- [13] FUELLERER G, DOERNER K F, HARTL R F, et al. Ant colony optimization for the two-dimensional loading vehicle routing problem [J]. Computers & Operations Research, 2009, 36(3):655-673.
- [14] DORIGO M, GAMBARDILLA L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1):53-66.
- [15] JIANG L Y, ZHANG J. Analysis of parameters in ant colony system[J]. Computer Engineering & Applications, 2007,43(20):31-36.
- [16] LIU Y, CHEN Z, WANG X, et al. Research on adaptive ant colony algorithm in robot hole making path planning[J]. International Journal of Control & Automation, 2017, 10(5):189-198.
- [17] 黄志华. 基于自适应蚁群算法的旅行商问题的求解[J]. 嘉应学院学报(自然科学), 2017, 35(2):18-23.
HUANG Z H, MATHEMATICS S O, UNIVERSITY J. Solution of traveling salesman problem based on adaptive ant colony algorithm [J]. Journal of Jiaying University(Natural Science), 2017, 35(2):18-23.
- [18] 吴华锋, 陈信强, 毛奇凰. 基于自然选择策略的蚁群算法求解 TSP 问题[J]. 通信学报, 2013(4):165-170.
WU Huangfeng, CHEN Xinqiang, MAO Qihuang. Improved ant colony algorithm based on natural selection strategy for solving TSP problem[J]. Journal on Communications, 2013(4):165-170.
- [19] 杭海存, 郭爱煌, 舒文杰. 基于 LEACH 与蚁群算法的 WSN 路由机制及性能分析[J]. 传感技术学报, 2008, 21(10):1735-1738.
HANG Haicun, GUO Aihuang, SHU Wenjie. Performance analysis of WSN routing scheme based on LEACH and ant algorithm[J]. Chinese Journal of Sensors and Actuators, 2008, 21(10):1735-1738.
- [20] XIANG J, YANG L, LUO Z. Flight safety measurements of UAVs in congested airspace[J]. Journal of Aeronautics, 2016, 29(5): 1355-1366.