

DOI: 10.11991/ykj.202011008

面向无人机路径规划的多目标粒子群优化算法

苏子美, 董红斌

哈尔滨工程大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001

摘 要: 针对无人机路径规划中方案单一的问题, 本文提出一种基于集分解的多目标综合学习粒子群优化算法框架 (MOCS-PSO/D), 该算法使用基于分解的多目标优化框架 (MOEA/D), 结合基于集的粒子群优化 (S-PSO) 和综合学习粒子群优化 (CLPSO), 对 CLPSO 和 PSO 的速度更新公式进行改进, 直接获得更多样的路径规划方案, 同时可以降低调用无人机数量。该算法将通过仿真实验与最近邻随机混合算法、遗传算法和基于集的综合学习粒子群优化算法 (CS-PSO) 对比, 且在算法的收敛性、多样性上进行分析。

关键词: 智能系统; 无人机; 路径规划; 贪心策略; 多目标优化; 粒子群优化; 进化算法; 自适应

中图分类号: U675.79

文献标志码: A

文章编号: 1009-671X(2021)03-0012-10

Multi-objective particle swarm optimization algorithm for UAV path planning

SU Zimei, DONG Hongbin

College of Information and Communication Engineering Harbin Engineering University, Harbin 150001, China

Abstract: Aiming at the problem of single scheme in UAV (Unmanned Aerial Vehicle) path planning, this paper proposes a multi-objective comprehensive learning particle swarm optimization algorithm framework based on set decomposition (MOCS-PSO/D). The algorithm uses the decomposition based multi-objective optimization framework (MOEA/D), combines set based particle swarm optimization (S-PSO) and comprehensive learning particle swarm optimization (CLPSO), so as to directly obtain more diverse path planning schemes, at the same time, in order to reduce the number of UAVs, the local search strategy is improved. The algorithm will be compared with the nearest neighbor random mixed hybrid algorithm, genetic algorithm and set based comprehensive learning particle swarm optimization (CS-PSO) through simulation experiments, and will be carried out in terms of convergence and diversity of solutions analysis.

Keywords: intelligent system; UAV; path planning; greedy strategy; multi-objective optimization; particle swarm optimization; evolutionary algorithm; adaptive

无人机路径规划的应用领域正变得越来越广。在军事领域方面无人机可以在战场物资投放、作战资源调度等方面使用^[1-5], 无人机参与作战可大量减少战争伤亡, 大大提高作战效率与成功率。在民用方面使用无人机进行快递配送、救灾物资投放和喷洒田地^[6]也成为新的目标, 因此在此方面所应用的技术具有巨大的市场潜力。

无人机在以上方面应用所产生的问题都可以归结为无人机路径规划问题, 这类问题是一个 NP 难问题 (NP-hard problem), 目前使用进化算法

可以在这类问题上得到很好的解决^[7-8]。文献 [2] 针对该问题提出了两阶段策略, 使用贪心算法与遗传算法作为一、二阶段进行优化; 文献 [5] 使用了单目标进化算法应用于无人机追捕罪犯; 文献 [9] 使用极端拥挤的 NSGA-II (EC-NSGA-II) 作为第一阶段, 求得近似 Pareto 前沿的极端解作为多目标遗传算法的初始粒子; 文献 [10] 将 MOEA/D (基于分解的多目标优化) 引入粒子群优化算法中以更好地平衡多个目标函数解; 文献 [11] 将基于集的粒子群优化算法 (S-PSO) 和综合学习粒子群优化 (CLPSO) 引入无人机路径规划问题中, 其中, 使用多目标粒子群优化算法求得的解, 其多样性好且收敛速度快, 因此在无人机路径规划上表现会更加有前景^[12-13]。

收稿日期: 2020-11-15.

基金项目: 黑龙江省自然科学基金 (LH2020F023).

作者简介: 苏子美, 女, 硕士研究生.

董红斌, 男, 教授.

通信作者: 董红斌, E-mail: donghongbin@hrbeu.edu.cn.

1 无人机路径规划问题

假设无人机飞行在足够高的上空可以忽略障碍且用户位置不变的条件下,无人机路径规划的多种问题都可以归结为带时间窗的车辆路径规划(vehicle routing problems with time window, VRPTW)问题。在VRPTW问题中的每个实例下,存在一个无人机基地和若干待满足需求的用户,且在固定的位置。每架无人机从无人机基地出发,经过可满足用户后回到无人机基地。当一个路径规划方案可以使所有用户按时得到满足,且无人机可回到无人机基地,则说明路径规划方案可行。

1.1 VRPTW问题的形式化表示

VRPTW可以形式化表示为一个完整有向带权图 $G=(C,E)$ 中的优化问题,如下所示。

1) 顶点集 C : 顶点 C 包括无人机基地点和用户点。其中当 $i=0$ 时, c_0 为图的根,表示无人机基地。当 $i \in [1,n]$, n 为用户数, c_i 表示用户。

2) 边集 E : 每条边分别表示2个顶点之间的链接,并且有一个距离属性。形式化的表示即在图 G 中,边集 $E=\{\langle c_i, c_j \rangle | c_i, c_j \in C, i \neq j\}$,每个边 $\langle c_i, c_j \rangle$ 由从变量 c_i 到变量 c_j 之间的欧几里得距离 d_{ij} 表示,其中 $d_{ij}=d_{ji}$,且距离与时间需要做等价替换。

对于每个用户 $c_i \in C, i \in [1,n]$ 的变量细节如表1所示。

对于无人机基地 c_0 的变量细节如表2所示。

表1 用户细节变量

变量	描述
q_i	需求
s_i	服务时间
e_i	最早可以开始服务用户 c_i 的时间
$l_i^{complete}$	最晚可以开始服务用户 c_i 的时间最晚结束服务用户 c_i 的时间

表2 无人机基地细节变量

变量	描述
q_0	$q_0=0$
s_0	$s_0=0$
e_0	任何一架无人机最早可以被派遣的时间
l_0	任何一架无人机最晚返回到无人机基地 c_0 的时间

设无人机基地的无人机 $v_r \in V$,其中 $\{r \in [1, |V|] | r \in \mathbb{N}\}$ 。设计能服务所有用户的路线需要二进制决策变量 a'_{ij} 。

$$a'_{ij} = \begin{cases} 1, & \text{如果无人机 } v_r \text{ 直接从 } c_i \text{ 到 } c_j \\ 0, & \text{其他} \end{cases} \quad (1)$$

二进制决策变量 b'_i 由服务于用户 c_i 的无人机 v_r 来表示。

$$b'_i = \begin{cases} 1, & \text{如果用户 } c_i \text{ 由无人机 } v_r \text{ 进行服务} \\ 0, & \text{其他} \end{cases} \quad (2)$$

以RC101为例,如图1所示,集中的点代表无人机基地位置,其余点代表用户位置,不同颜色代表不同的无人机出行路线,此图1表明调用了8架无人机完成了此次任务。

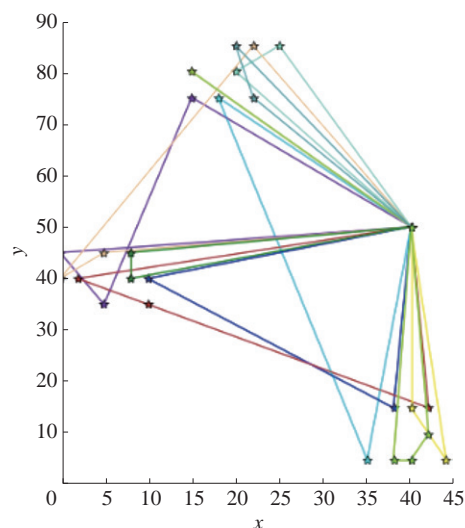


图1 RC101 路径规划图

1.2 解的形式

该问题中顶点集 $C=\{0, 1, \dots, n\}$, 0表示无人机基地, $1 \sim n$ 代表用户。每个解 x_i 包含一组可行路径 $R_i=\{R_i^1, \dots, R_i^{N_{vi}}\}$ 。其中 R_i^j 表示解 i 中的第 j 条路径,由从无人机基地到一组可服务用户再回到无人机基地的顶点序列组成,并且 N_{vi} 表示在解 x_i 中使用的无人机数量。

在使用该算法求解VRPTW时,会得到一组可行解,每个解 x_i 包含一组可行路线 $R_i=\{R_i^1, \dots, R_i^{N_{vi}}\}$ 。其中 R_i^j 表示解 i 中的第 j 条路线,并且 N_{vi} 表示在解 i 中使用的无人机数量。解可以由有向带权图、邻接表、邻接矩阵进行表示,如图2—图4所示。以有5个用户为例,顶点集表示为 $C=\{0, 1, 2, \dots, 5\}$,解的路线集表示为 $R_i=\{R_i^1, R_i^2\}$,其中 $R_i^1=(0, 3, 4, 0)$, $R_i^2=(0, 2, 1, 5, 0)$ 。

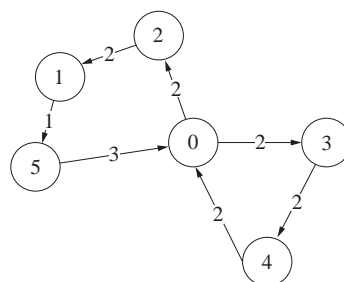


图2 示例有向带权图

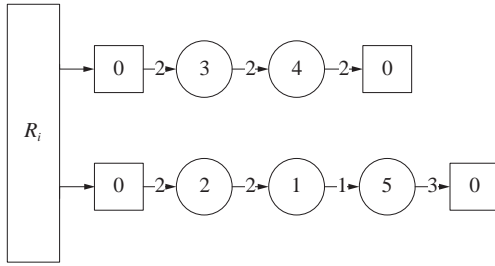


图3 示例邻接图

0	0	1	1	0	0
0	0	0	0	0	1
0	1	0	0	0	0
0	0	0	0	1	0
1	0	0	0	0	0
1	0	0	0	0	0

图4 示例邻接矩阵

1.3 约束条件

在 VRPTW 问题中, 有 6 个约束条件, 其中: i 为路段起点, j 为路段终点, n 为用户加无人机基地总数, r 为调用的第 r 架无人机, $|V|$ 代表调用无人机数。

1) 边与顶点的约束: 恰好有一条路径进入和离开与用户相关联的每个顶点, 这样后面的公式表示才是成立的。

$$\sum_{i=0}^n a'_{ij} = b'_j = 0, \forall r = 1, 2, \dots, |V|, \forall j = 1, 2, \dots, n$$

$$\sum_{j=0}^n a'_{ij} = b'_i = 0, \forall r = 1, 2, \dots, |V|, \forall i = 1, 2, \dots, n \quad (3)$$

2) 每名用户只能使用一架无人机, 且所有路线从无人机基地开始。

$$\sum_{r=1}^{|V|} b'_i = 1, \forall i = 1, 2, \dots, n$$

$$\sum_{r=1}^{|V|} b'_0 = |V| \quad (4)$$

(3) 无人机总容量约束: 表示每架无人机的负载不得超过其承载能力。

$$\sum_{i=0}^n b'_i \times q_i \leq Q_r, \forall r = 1, 2, \dots, |V| \quad (5)$$

式中 q_i 为用户 c_i 的需求。

4) 无人机工作最长时间约束。

$$\sum_{i=0}^n a'_{ij} d_{ij} + \sum_{i=0}^n a'_{ij} w_j + \sum_{i=1}^n a'_{ij} s_i < T_r$$

$$\forall r = 1, 2, \dots, |V|, \forall j = 1, 2, \dots, n \quad (6)$$

$$w_{c_j} = \max \{0, e_j - t_j\}, \forall j = 1, 2, \dots, n$$

式中: w_j 为无人机 v 等待用户 j 的时间; e_j 为最早可以开始服务用户 c_j 的时间。

5) 每个用户完成时间的约束。

$$\sum_{i=0}^k a'_{ij} d_{ij} + \sum_{i=1}^k a'_{ij} s_j < t_{\text{complete } i}$$

$$\forall r = 1, 2, \dots, |V|, \forall j, k = 1, 2, \dots, n \quad (7)$$

式中: d_{ij} 为用户 c_i 和 c_j 之间的旅行时间; s_j 为服务用户 c_j 的时间。

6) 每个用户的时间窗约束。

$$w_{c_j} \leq t_j \leq l_j, \forall j = 1, 2, \dots, n$$

$$t_j = t_i + w_j + s_i + d_{ij}, \forall i = 0, 1, \dots, n \quad (8)$$

$$\forall j = 1, 2, \dots, n, i \neq j$$

式中: t_j 为可以到用户 c_j 的时间; l_j 为无人机 v 最晚可以开始服务用户 c_j 的时间。

1.4 目标函数

根据 VRPTW 的特点, 设定了包含冲突的 5 个目标函数。

1) 调用无人机数 f_1 与路线总距离 f_2 。

$$\min f_1 = |V| \quad (9)$$

$$\min f_2 = \sum_r \sum_{i=0}^n \sum_{j=0}^n a'_{ij} d_{ij} \quad (10)$$

式 (9) 表示调用无人机数, 式 (10) 表示路线总距离, 在文献 [9] 中证明无人机数量与路线总距离成正比, 但路线总距离与无人机等待时间为一对冲突函数。

2) 路线总时间 f_3 。

$$\min f_3 = \max \left\{ \sum_{1 \leq r \leq |V|} \sum_{i=0}^n \sum_{j=0}^n a'_{ij} d_{ij} + \sum_{i=0}^n \sum_{j=1}^n a'_{ij} w_j + \sum_{i=1}^n \sum_{j=1}^n a'_{ij} s_j \right\} \quad (11)$$

式 (11) 表示路线总时间。总时间与车辆数量在文献 [9] 中被证明为一对冲突函数。

3) 无人机等用户总时间 f_4 与用户等无人机总时间 f_5 。

$$\min f_4 = \max_{1 \leq r \leq |V|} \left\{ \sum_{j=1}^n \max \{b'_j \times (e_j - t_j), 0\} \right\} \quad (12)$$

$$\min f_5 = \max_{1 \leq r \leq |V|} \left\{ \sum_{j=1}^n \max \{b'_j \times (t_j - e_j), 0\} \right\} \quad (13)$$

式中: $|V|$ 为使用的无人机架数; n 为第 r 架无人机服务的用户数。

式 (12) 表示无人机等用户总时间, 式 (13) 表示用户等无人机总时间, 这 2 个公式是本文提出的一对冲突函数。

1.5 多目标优化

多目标优化问题可以用函数表示为

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))$$

$$\mathbf{x} \in \Omega, f_i(\mathbf{x}) \in R^M, \forall i = 1, 2, \dots, M \quad (14)$$

式中: R^M 为目标空间; Ω 为决策空间; 目标函数 F 可将决策空间映射到目标空间。

与单目标优化问题不同, 多目标优化中的不同目标之间一般需要存在冲突, 因此用单目标优

化算法往往效果不佳。

在多目标优化问题中, $\forall 2$ 个解 $\mathbf{x}, \mathbf{y} \in \Omega$, 当且仅当对于 $\forall_i \in \{1, 2, \dots, M\}$, $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ 且 $\exists j \in \{1, 2, \dots, M\}$, $f_j(\mathbf{x}) < f_j(\mathbf{y})$, 则 \mathbf{x} 占优, \mathbf{x} 支配 \mathbf{y} , 与 \mathbf{y} 相比, \mathbf{x} 是 Pareto 最优解。使用多目标优化算法可以求得 Pareto 最优集, Pareto 最优集是所有 Pareto 最优解的集合。

2 两阶段 MOCS-PSO/D 算法

本文提出的 MOCS-PSO/D 框架是基于 MOEA/D 的框架, 将多个目标函数进行归一化处理, 并使用 S-PSO 来对 VRPTW 问题的解进行基于集合和概率的表示, 同时结合 CLPSO 来对粒子进行更新。最后, 根据实际的应用情况, 对局部搜索策略进行调整, 以减少调用的无人机架数。

两阶段算法流程如图 5 所示。

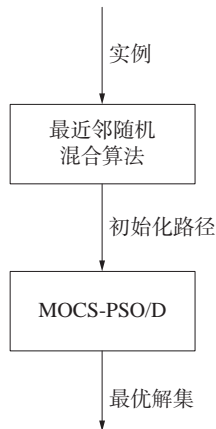


图5 两阶段算法流程图

2.1 粒子速度和位置的表达

在 VRPTW 问题中, 利用邻接矩阵来表示每个粒子的位置, 离散的位置表示方式可以适用于 S-PSO 的位置更新和速度更新公式中。每个粒子的位置由邻接矩阵表示, 粒子 h 在步骤 k 的位置可以由 x_k^h 表示, 粒子的位置可以扩展的表示成 $x_k^h = [x_k^{h_1}, \dots, x_k^{h_m}]$, 每个维表示一个边集, 每个边集有 2 个边, 每个边由从无人机基地和用户集中选取 2 个点组成, 其中 δ 为当前维度, δ 相邻的前后 2 个点 $c_i, c_j \in \{1, 2, \dots, \delta-1, \delta+1, \dots, m\}$, 且 $c_i \neq c_j$ 。该边集由式 (15) 表示。

$$x_k^{h\delta} = [\langle c_i, \delta \rangle, \langle \delta, c_j \rangle] \quad (15)$$

根据每个粒子的位置可以画出一个完整的有向哈密顿圆。其中每个有向哈密顿圆的集合代表要调度的每架无人机在满足约束条件下的出行路线。

速度由边与其出现概率表示, 每个粒子的第 m 维的速度分量用 $u_k^{h\delta}$, $\forall \delta \in [1, m]$ 表示。粒子 h 在步

骤 k 时在 m 维空间上的速度定义为: $u_k^h = [u_k^{h_1}, u_k^{h_2}, \dots, u_k^{h_m}]$ 。 δ 维的边概率集由式 (16) 表示。

$$u_k^{h\delta} = \left\{ \frac{\langle c_i, c_j \rangle}{p(c_i, c_j)} \mid \langle c_i, c_j \rangle \in A^\delta \right\} \quad (16)$$

集合 A^δ 中的边都是与有向完全图 G 中的状态 δ 相邻的状态组成的状态对。其中, 边 $\langle c_i, c_j \rangle$ 出现的概率用 $p(c_i, c_j)$ 表示, $p(c_i, c_j) \in [0, 1]$ 。在后续对边 $\langle c_i, c_j \rangle$ 进行更新时, 将根据此概率进行选择。如果 $p(c_i, c_j) = 0$, 则表示 $u_k^{h\delta}$ 忽略边 $\langle c_i, c_j \rangle$ 。

2.2 第一阶段 (最近邻随机混合算法)

初始化阶段将混合使用最近邻算法和随机算法两种启发式算法进行路径初始化。

NNH 是一种贪心的初始化种群候选解的方法, 是在实际设计算法时常用的一种方法。将用户集 C 和无人机基地集 D 的数据进行读取, 利用 NNH 建立可行解。具体方法如下。

1) 初始化从无人机基地开始的路线, 递归地将最近的可满足用户添加到路线中, 直到不存在可满足用户, 则回到无人机基地。已满足的用户移除用户集。

2) 如果用户集 C 不为空, 则转到 1), 否则终止算法。

随机算法将每次选择的用户更改为随机可满足用户即可。也可将 2 种算法混合使用。

2.3 第二阶段 (MOCS-PSO/D 算法)

本文针对 VRPTW 问题的 MOCS-PSO/D 框架将阶段一得到的结果作为初始值进行优化。

MOCS-PSO/D 框架的执行详细过程如下。

1) 初始化: 初始化种群大小 P_{size} 、目标函数个数 N 、每个目标函数上的采样个数 H , 邻居个数 T , 最大迭代次数 g_{max} , 学习率 c_1, c_2 , 随机数 σ_1, σ_2 , 线性速度更新权重 w , 外部更新集 E_p 。生成均匀权重向量, 同时使用第一阶段中的解作为初始化粒子。

2) 求值和规范化: 求得所有粒子以获得目标向量。更新每个目标的上下限。相应地, 规范化每个粒子的目标向量。

3) 学习者和存档更新: 使用粒子的权重向量和其邻域中所有新生成的解以 MOEA/D 方式更新每个粒子的学习者。使用所有新解基于 Pareto 优势更新外部存档 E_p 。

4) 粒子群更新: 对每一个粒子, 在 CLPSO 的速度更新公式中执行基于元素的运算来更新其速度, 然后构造一个新的解来更新其位置。

5) 局部搜索策略: 再对每一个粒子进行局部搜索策略, 不断选择用户最少的路线, 试图将

其中的用户插入其他路线中, 依旧可满足约束条件。

6) 额外粒子群更新: 如果某粒子多次没有进行更新, 则对其进行额外粒子群更新策略, 使用基于元素的算法运算来更新其速度, 然后构造一个新的解来更新其位置。

7) 终止: 如果满足停止条件, 则停止并输出外部储存集 E_p ; 否则, 转到 2)。

2.3.1 速度更新

CLPSO 算法是 PSO 算法的一个变种, 其采用一种新的速度更新规则来防止解的过早收敛, 在每次更新时只采用局部最优解进行更新, 而不使用全局最优。用于更新 VRPTW 的 S-PSO 算法中每次迭代 k 次时每个粒子的位置。CLPSO 速度更新规则为

$$u_{k+1}^{h_s} = \omega \times u_k^{h_s} + \sum_{i=0}^{n-1} (c_1 \times \sigma_1 \times (B_{\text{rand}_i} x_k^{h_s} - x_k^{h_s})) \quad (17)$$

式中: w 为随迭代次数线性变化的惯性权重系数; c_1 为向邻域学习的学习率; σ_1 为在 $[0, 1]$ 上的随机数; 在本算法中, $B_{\text{rand}_i} x_k^{h_s}$ 为使用 MOEA/D 框架后的邻域中随机选择的 2 个邻居, 最后将计算出的最大速度赋值给 $u_{k+1}^{h_s}$ 。

S-PSO 是基于集合和概率定义的, 因此将其用于 VRPTW 问题中需要对其中的运算符进行重新定义。

算子 1: 权重系数与速度算子相乘由式 (18) 定义, 该公式使用公式 (19) 确定边 $\langle c_i, c_j \rangle$ 更新后存在的概率 $p'(c_i, c_j)$, 其中 $p(c_i, c_j)$ 为原来存在的概率。

$$\omega \times u_k^{h_s} = \{\langle c_i, c_j \rangle / p'(c_i, c_j) | \langle c_i, c_j \rangle \in A^\delta\} \quad (18)$$

$$p'(c_i, c_j) = \begin{cases} 1, & \text{if } w \times p(c_i, c_j) > 1 \\ w \times p(c_i, c_j), & \end{cases} \quad (19)$$

算子 2: 位置与位置算子相减由式 (20) 定义。

$$x_k^{h_s} - x_k^{h'_s} = M^\delta = \{\langle c_i, c_j \rangle | \langle c_i, c_j \rangle \in x_k^{h_s} \text{ and } \notin x_k^{h'_s}\} \quad (20)$$

算子 3: $c \times \sigma \times$ (位置与位置算子相减) 算子由式 (21) 定义。该公式将使用给定的 crisp 集合 M^δ 转化为具有概率的边 $\langle c_i, c_j \rangle$ 集合, 式 (22) 利用 $c \times \sigma$ 用来得到更新后的速度值。

$$c \times \sigma \times M^\delta = \{\langle c_i, c_j \rangle / p'(c_i, c_j) | \langle c_i, c_j \rangle \in M^\delta\} \quad (21)$$

$$p'(c_i, c_j) = \begin{cases} 1, & \text{if } \langle c_i, c_j \rangle \in M^\delta \text{ and } c \times \sigma > 1 \\ c \times \sigma, & \text{if } \langle c_i, c_j \rangle \in M^\delta \text{ and } 0 \leq c \times \sigma \leq 1 \\ 0, & \text{if } \langle c_i, c_j \rangle \notin M^\delta \end{cases} \quad (22)$$

算子 4: 速度与速度算子相加由式 (23) 定义, 即边 $\langle c_i, c_j \rangle$ 被选择的概率将以最大值为准则, 以从多个粒子中保留下边 $\langle c_i, c_j \rangle$ 被选择的最大概率。

$$u_k^{h_s} + u_k^{h'_s} = \{\langle c_i, c_j \rangle / \max(p_k^{h_s}(c_i, c_j), p_k^{h'_s}(c_i, c_j)) | \langle c_i, c_j \rangle \in A^\delta\} \quad (23)$$

为了更加清晰的进行表示, 用定义算子的应用实例进行说明。假设:

$$u_k^1 = \{\langle 2, 1 \rangle / 0.5, \langle 1, 3 \rangle / 0.2, \langle 4, 1 \rangle / 0.3, \langle 2, 1 \rangle / 0.1\}$$

$$x_k^1 = \{\langle 5, 1 \rangle, \langle 1, 3 \rangle\}$$

$$B_0 x_k^1 = \{\langle 2, 1 \rangle, \langle 1, 3 \rangle\}$$

$$B_1 x_k^1 = \{\langle 4, 1 \rangle, \langle 1, 3 \rangle\}$$

$$w = 0.7, c_1 = 2, \sigma_1 = 3$$

可以得到

$$\omega \times u_k^1 = \{\langle 2, 1 \rangle / 0.35, \langle 1, 3 \rangle / 0.14, \langle 4, 1 \rangle / 0.21, \langle 2, 1 \rangle / 0.07\}$$

$$c_1 \times \sigma_1 \times (B_{\text{rand}_0} x_k^1 - x_k^1) = \{\langle 1, 3 \rangle / 0.6\}$$

$$c_1 \times \sigma_1 \times (B_{\text{rand}_1} x_k^1 - x_k^1) = \{\langle 4, 1 \rangle / 0.6, \langle 1, 3 \rangle / 0.6\}$$

最终 u_{k+1}^1 则可表示为,

$$u_{k+1}^1 = \omega \times u_k^1 + \sum_{i=0}^{n-1} (c_1 \times \sigma_1 \times (B_{\text{rand}_i} x_k^1 - x_k^1)) = \{\langle 2, 1 \rangle / 0.35, \langle 1, 3 \rangle / 0.6, \langle 4, 1 \rangle / 0.6, \langle 2, 1 \rangle / 0.07\}$$

通过这个实例可以看出, 速度将根据 $B_i x_k^1$ 学习, 从而提高了 $B_i x_k^1$ 中的边 $\langle 1, 3 \rangle$ 和 $\langle 4, 2 \rangle$ 的概率。

此外针对综合学习策略的缺陷, 设置了额外粒子群更新策略, 该策略将学习邻域内的粒子及外部储存集 E_p 中解的信息。其中速度更新策略如式 (24) 所示

$$u_{k+1}^{h_s} = \omega \times u_k^{h_s} + \sum_{i=0}^{n-4} (c_1 \times \sigma_1 \times (B_i x_k^{h_s} - x_k^{h_s})) + \sum_{i=0}^{n-14} (c_2 \times \sigma_2 \times (EP_i x_k^{h_s} - x_k^{h_s})) \quad (24)$$

2.3.2 位置更新

当粒子位置的边成立时, 初始值将设置为 $[0, 1]$ 的随机值, 表示可以作为出行路段。位置更新的规则根据式 (25) 进行完成, 这里的 '+' 号将根据 S-PSO 的特点重新进行构造。 x_{k+1}^h 将由 x_k^h 和 u_{k+1}^h 的 crisp 集合将在满足约束条件的情况下, 经过构造得到。

$$x_{k+1}^h = x_k^h + u_{k+1}^h \quad (25)$$

式中: 速度 u_{k+1}^h 先将通过式 (26) 进行更新, 从而转化为一个 crisp 集合。

$$\text{Cut}(u_{k+1}^h) = \left\{ \langle c_i, c_j \rangle \mid \frac{\langle c_i, c_j \rangle}{p(c_i, c_j)} \in u_{k+1}^h \text{ and } p(c_i, c_j) \geq \text{rand} \right\} \quad (26)$$

式中: rand 为 $[0, 1]$ 的随机数, 当且仅当 $p(c_i, c_j) \geq \text{rand}$ 时, $\langle c_i, c_j \rangle / p(c_i, c_j)$ 将保存在速度集中, 说明当 $p(c_i, c_j)$ 较大时, 在经过转换后被保存在 crisp 集合中的概率更大。在实际计算中, 当同一边 $\langle c_i, c_j \rangle$ 被多次选中后, 即会增加进入 crisp 集合的概率。

x_{k+1}^h 的构造性方式如下, 其中可到达点即为在当前点满足约束的条件下可到达的节点。

1) 对于每架无人机, 在 x_{k+1}^h 上的路径初始化为 0。

2) 为了构造无人机出行路径,该方法将选择一个从无人机基地的可到达点。再依次向后寻找可到达点,直到所有点都不满足,则回到无人机基地。

其中,可到达点的选择将根据以下3个 crisp 集合依次选择:

$$S_U = \{c_i | \langle c_i, c_j \rangle \in u_{k+1}^h, \langle c_i, c_j \rangle \text{ 满足约束 } \Omega\}$$

$$S_X = \{c_i | \langle c_i, c_j \rangle \in x_k^h, \langle c_i, c_j \rangle \text{ 满足约束 } \Omega\}$$

$$S_A = \{c_i | \langle c_i, c_j \rangle \in A, \langle c_i, c_j \rangle \text{ 满足约束 } \Omega\}$$

如果 S_U 、 S_X 、 S_A 中都没有可到达点,则将调用新一架无人机出行,且将 $\langle c_i, 0 \rangle$ 和 $\langle 0, c_j \rangle$ 添加到 x_{k+1}^h 中。

3 实验结果与分析

3.1 数据介绍

本文使用的数据是 Solomon 在 1984 中介绍的 VRPTW 的基准数据集,该数据集是基于几个标准路径测试问题数据集生成的;测试使用的 RC1 类包含随机和集群地理位置用户的混合;实验数据的用户规模为 25。其中用户数据集的属性包括用户编号、横纵坐标、需求量、时间窗和服务时间,无人机数据集的属性包括横纵坐标、最大容量和最大工作时间。

3.2 对比实验设置

实验都在运行 30 次取平均值后进行结果统计。

为了对比的公平性,对以下对比算法进行了处理。

1) 第一阶段采取最近邻随机混合策略 (NNH+RANDOM): 取实验的随机 15 个粒子与其他算法进行比较。

2) 第二阶段采取单目标遗传算法 (GA) 和基

于集的单目标 CS-PSO(基于集的综合学习粒子群优化算法): 取随机 15 次实验的粒子与其他算法进行比较,该算法使用的单目标适应度函数为

$$F(x_k^h) = \alpha f_1 + \beta f_2 + \gamma \frac{(f_4 + f_5)}{f_3} \quad (27)$$

3) 第二阶段采取多目标算法 MOCS-PSO/D: 随机取一次实验的外部储存集解与其他算法进行比较。

3.3 运行时间对比

本实验将对 4 种算法的收敛速度进行对比。如表 3 所示。

表3 4种算法平均一次实验的运行时间

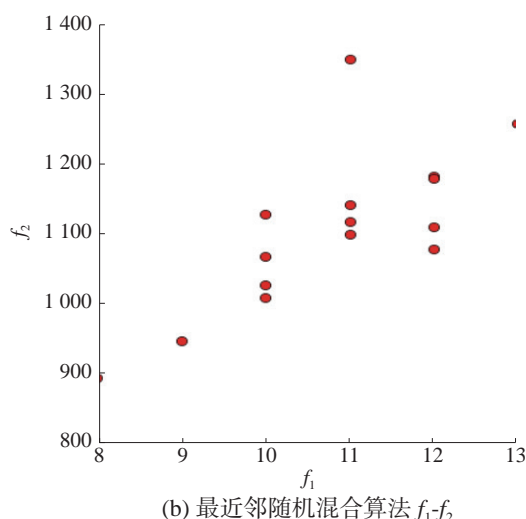
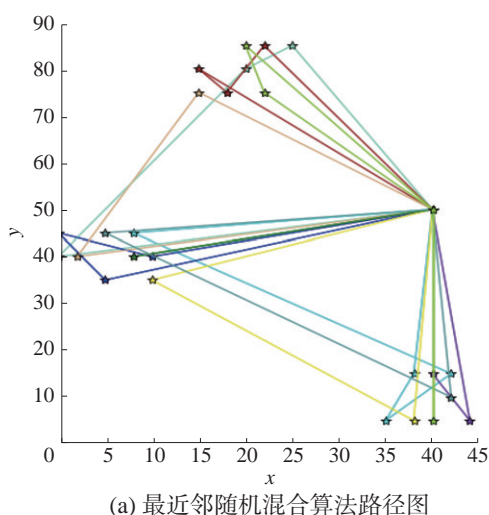
算法	NNH+RANDOM	GA	CS-PSO	MOCS-PSO/D
平均运行时间/s	24	213	174	457

该实验数据表明,第一阶段的 Rand+NNH 算法可以仅消耗少量时间即可以搜索到可行解,为第二阶段的算法提供初始粒子。

GA 与 CS-PSO 算法相比,运行消耗时间相对较长,因此选择对 CS-PSO 算法进行改进。CS-PSO 算法与 MOCS-PSO/D 算法相比时间消耗较少,但考虑到 CS-PSO 算法每次实验只能得到 1 个最优解,而 MOCS-PSO/D 算法一次实验则可以得到 15 个最优解,从获得解效率来说,MOCS-PSO/D 具有比较大的优势,说明 MOCS-PSO/D 具有很好的收敛性。

3.4 算法性能对比

本实验将从解的多样性方面进行对比。以 RC101 数据样本为实验数据,图 6—图 9 分别为 NNH+RANDOM、GA、CS-PSO 和 MOCS-PSO/D 的实验图,其中从左到右依次为其中一个粒子的路径图、3 组目标 $f_1 - f_2$ 、 $f_1 - f_3$ 、 $f_4 - f_5$ 对比的散点图,其中路程和时间已做标准化处理,单位为“1”。



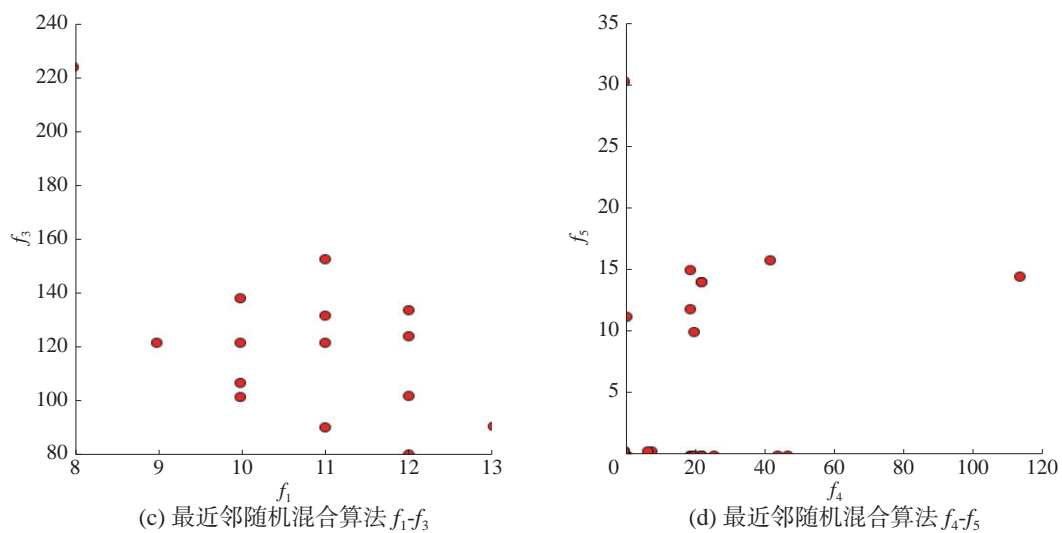


图6 NNH+RANDOM 算法结果

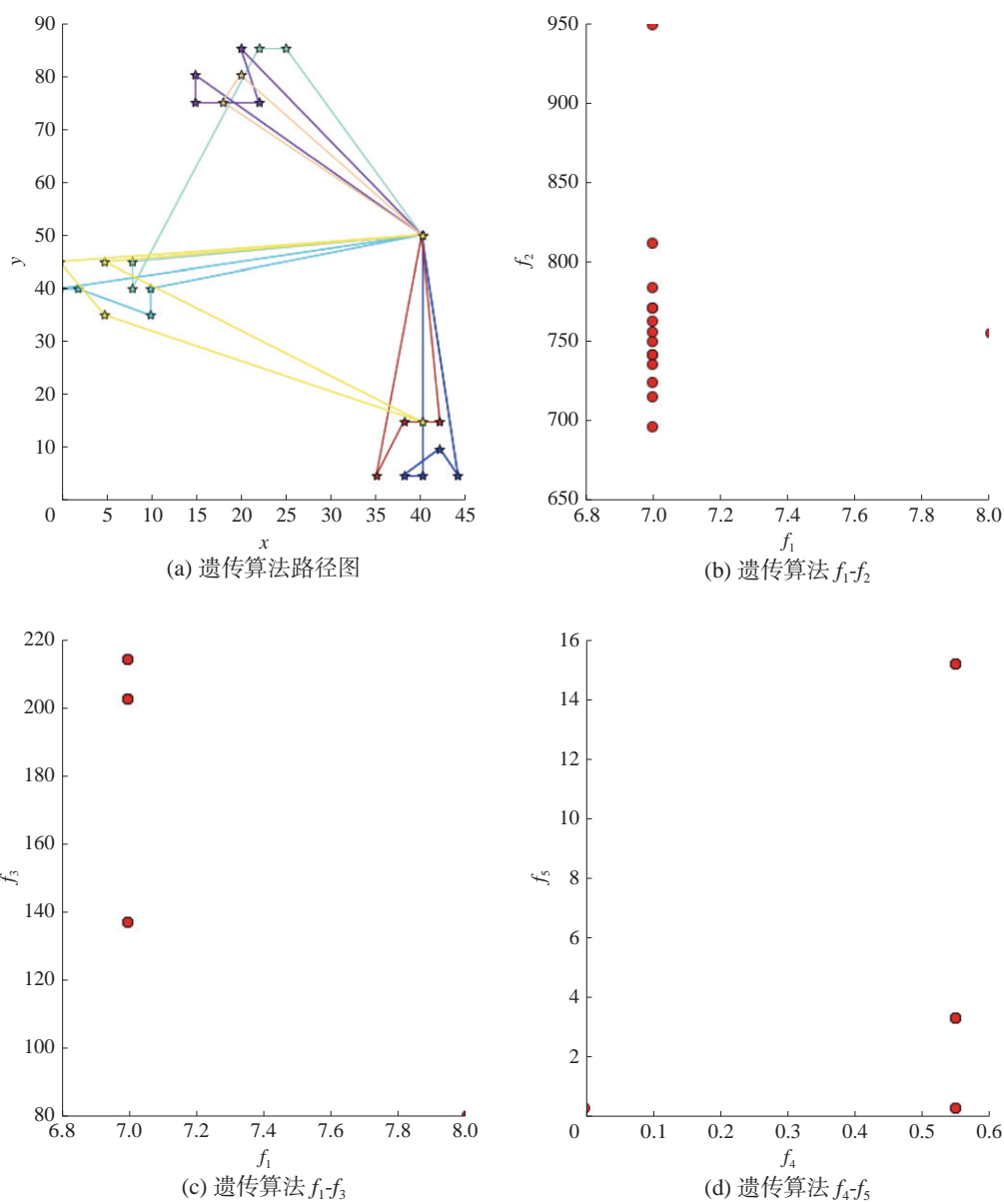


图7 GA 算法结果

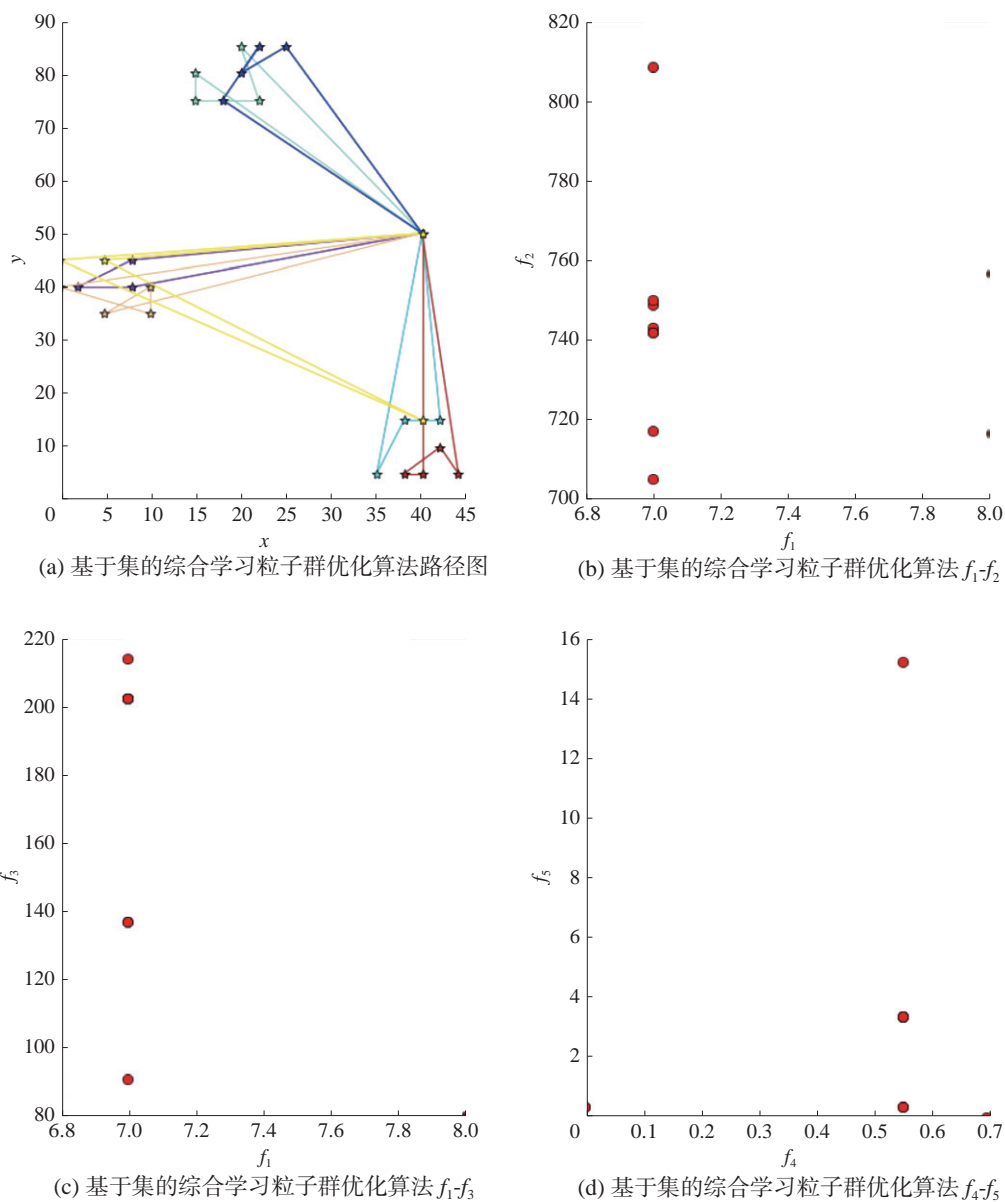
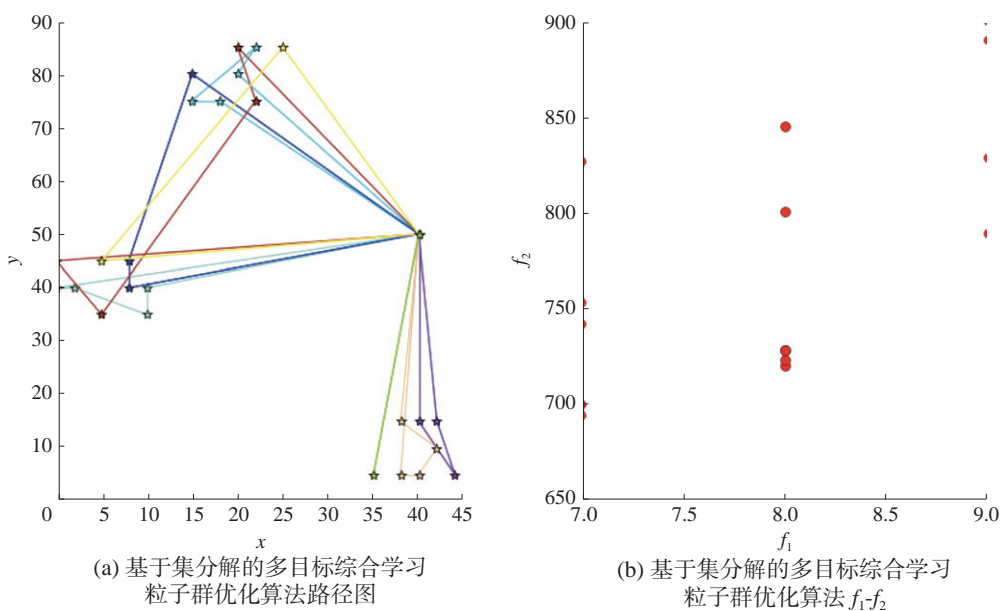


图8 CS-PSO 算法结果



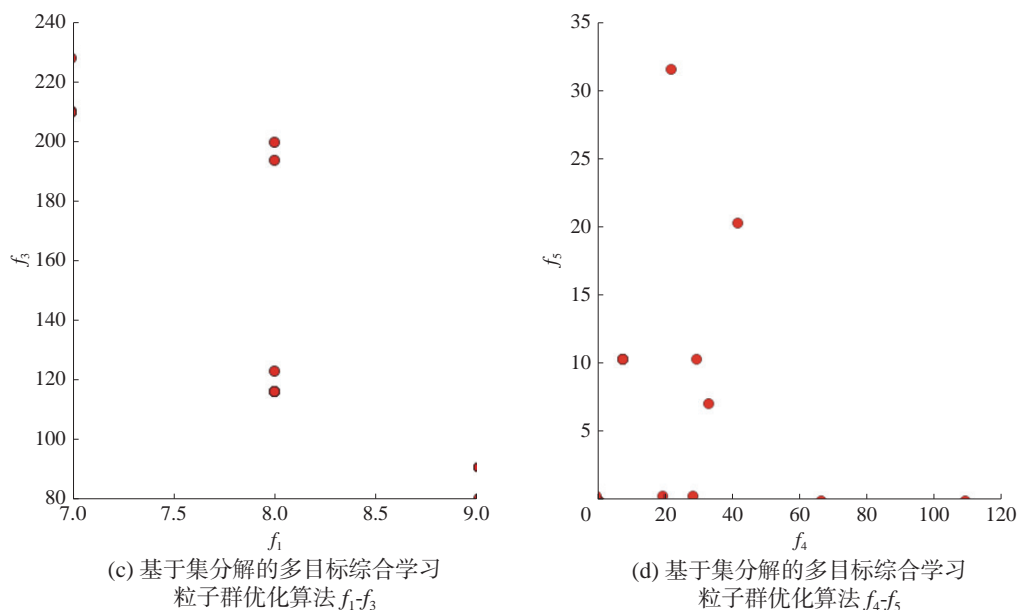


图9 MOCS-PSO/D 算法结果

由 4 组实验图的 (a) 图可以看出, 4 种算法都可以有效地找到可行路径, 且无人机群可以服务全部的 25 名用户。由 4 组实验图的 (b) 图可以看出, GA、CS-PSO 和 MOCS-PSO/D 都可以显著地降低调用无人机数量。同时, 可以从 MOCS-PSO/D 算法中发现 f_1-f_2 为成正比的目标。最后, 由 4 组实验图的 (c)(d) 图可以看出, MOCS-PSO/D 算法搜索到的解的多样性更好, 而 GA、CS-PSO 算法得到的解比较单一。同时, 可以从 MOCS-PSO/D 算法中发现 f_1-f_3 、 f_4-f_5 为两对冲突的目标。

综上, MOCS-PSO/D 算法具有很好的多样性, 且可以搜索到质量更高的 Pareto 最优解。

4 结论

在面向的无人机路径规划问题中, 针对 VRPTW 问题使用提出的两阶段算法, 第一阶段为随机贪心混合策略, 第二阶段使用的 MOCS-PSO/D 算法在 CS-PSO 算法的基础上增加了 MOEA/D 框架, 改变了 CLPSO 和 PSO 的速度更新公式, 使其解可以自适应的在 5 个目标函数上达到均衡, 更加符合实际的应用场景, 当某路径规划方案不能执行时, 可立刻选择其他备选方案。同时, 与随机贪心混合策略、GA 和 CS-PSO 相比, 在算法收敛速度、解的多样性上都有很好的表现。

但是本文算法的解求得的调用无人机数量依

旧较多, 由于该目标为主要目标函数, 未来将对该方面进行研究。

参考文献:

- [1] 李绍斌, 姜大立, 付高阳, 等. 战场物资无人机配送研究[J]. 国防科技, 2019, 40(3): 98-104.
- [2] 皇甫莹丽, 程威. 灾害救援中无人机资源优化配置仿真研究[J]. 计算机仿真, 2019, 36(2): 55-58.
- [3] PRAG K. Computational Logistics of the Vehicle Routing Problem with Time Windows[D]. University of the Witwatersrand, 2018: 27-64.
- [4] OTTO A, AGATZ N, CAMPBELL J, et al. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey[J]. Networks, 2018, 72(4): 411-458.
- [5] 庞强伟, 胡永江, 李文广, 等. 多无人机协同侦察任务规划方法研究综述[J]. 电讯技术, 2019, 59(6): 741-748.
- [6] 吴柏林. 多目标粒子群优化算法及其应用研究[D]. 成都: 电子科技大学, 2019.
- [7] 刘畅, 谢文俊, 张鹏, 等. 多目标群多基地多无人机协同任务规划[J]. 弹箭与制导学报, 2019, 39(1): 119-124.
- [8] 杨轻, 杨忠, 许昌亮, 等. 改进 PSO 算法及其无人机电力巡线规划应用[J]. 应用科技, 2019, 46(3): 80-85.
- [9] YU Xue, CHEN Weineng, GU Tianlong, et al. Set-Based Discrete Particle Swarm Optimization Based on Decomposition for Permutation-Based Multiobjective Combinatorial Optimization Problems[J]. IEEE Transactions on Cybernetics, 2018, 47(7): 2139-2153.

(下转第 26 页)

- [9] TIAN Zhi, SHEN Chunhua, CHEN Hao, et al. FCOS: fully convolutional one-stage object detection[C]//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision. Seoul, Korea (South), 2019: 9626-9635.
- [10] LAW H, DENG Jia. CornerNet: detecting objects as paired keypoints[J]. International journal of computer vision, 2020, 128(3): 642-656.
- [11] ZHOU Xingyi, WANG Dequan, KRÄHENBUHL P. Objects as points[J]. arXiv: 1904.07850, 2019.
- [12] MA Ningning, ZHANG Xiangyu, ZHENG Haitao, et al. ShuffleNet V2: practical guidelines for efficient CNN architecture design[C]//Proceedings of the 15th European Conference on Computer Vision. Munich, Germany, 2018: 122-138.
- [13] 李启运, 纪庆革, 洪赛丁. FastFace: 实时鲁棒的人脸检测算法 [J]. 中国图象图形学报, 2019, 24(10): 1761-1771.
- [14] 伍凯, 朱恒亮, 郝阳阳, 等. 级联回归的多姿态人脸配准[J]. 中国图象图形学报, 2017, 22(2): 257-264.
- [15] YANG Shuo, LUO Ping, LOY C C, et al. WIDER FACE: a face detection benchmark[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, 2016: 5525-5533.
- [16] ZHANG Shifeng, ZHU Xiangyu, LEI Zhen, et al. FaceBoxes: a CPU real-time face detector with high accuracy[C]//Proceedings of 2017 IEEE International Joint Conference on Biometrics. Denver, CO, USA, 2017.
- [17] ZHANG Kaipeng, ZHANG Zhanpeng, LI Zhifeng, et al. Joint face detection and alignment using multitask cascaded convolutional networks[J]. IEEE signal processing letters, 2016, 23(10): 1499-1503.
- [18] LI Haoxiang, LIN Zhe, SHEN Xiaohui, et al. A convolutional neural network cascade for face detection[C]//Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA, 2015.

本文引用格式:

段燕飞, 刘胤田, 王瑞祥, 等. 非约束环境下的实时人脸检测方法 [J]. 应用科技, 2021, 48(3): 21-26.

DUAN Yanfei, LIU Yintian, WANG Ruixiang, et al. Real-time face detection method in an unconstrained environment[J]. Applied science and technology, 2021, 48(3): 21-26.

(上接第 20 页)

- [10] ZHENG Yujun, DU Yichen, LING Haifeng, et al. Evolutionary Collaborative Human-UAV Search for Escaped Criminals[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(2): 217-231.
- [11] WANG Jiahai, WENG Taiyao, ZHANG Qingfu. A Two-Stage Multiobjective Evolutionary Algorithm for Multiobjective Multidepot Vehicle Routing Problem With Time Windows[J]. IEEE Transactions on Cybernetics, 2019, 49(7): 2467-2478.
- [12] HUI Y, YUJIA W, QIANG C, et al. Multi - Objective Particle Swarm Optimization Based on Multi - population Dynamic Cooperation[J]. Electronic Science and Technology, 2019, 32(10): 28-33., 2019, 30(2): 28-33.
- [13] JIA Yahui, CHEN Weineng, GU Tianlong, et al. A Dynamic Logistic Dispatching System With Set-Based Particle Swarm Optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 48(9): 1607-1621.

本文引用格式:

苏子美, 董红斌. 面向无人机路径规划的多目标粒子群优化算法 [J]. 应用科技, 2021, 48(3): 12-20, 26.

SU Zimei, DONG Hongbin. Multi-objective particle swarm optimization algorithm for UAV path planning[J]. Applied science and technology, 2021, 48(3): 12-20, 26.