

一种基于混合进化算法的大规模空地协同路径规划方法

杜文博、翟若楠、胡兆琨、郭通、张学军

发明背景

近年来，随着社会经济的高速发展，各个领域对无人机的需求与日俱增。尤其是电商行业的迅速发展，人们对物流服务质量有着越来越高的要求与期待。而新冠疫情的冲击，使得社区团购、线上买菜等多种多样的购物形式不断涌现，更是加强了人们对高服务高效率的物流配送的期待。相比于传统的卡车配送，无人机在配送速度和道路选择方面具有显著优势，但是其缺点也同样不可忽略。例如无人机的载重量太小，续航里程不足，这使得无人机很难独立完成整个配送流程。而若将无人机和卡车协同配合，两者的优势互补，可以极大提高配送效率，提升物流的服务水平。一种能协同多辆卡车和多架无人机的路径规划方法将会发挥不可估量的作用。

物流配送中的多辆卡车多架无人机路径规划问题具有求解规模大、约束条件多，求解难度指数增长等困难。在现实场景中，需要在尽可能短的时间内完成整个物流配送过程。同时，还需要考虑无人机的续航里程、卡车回收无人机等多重约束，因此该问题为一个组合优化的问题。

现有的卡车和无人机调度问题主要是针对单车单机模型、单车多机模型，关于多车多机模型研究较少，而多车多机模型无疑更适用于未来的实际应用当中。此外，现有算法达到未考虑无人机可以连续服务多个用户点和可以降落在不同卡车上等情况。

发明内容

本发明是一种适用于物流配送中多辆卡车和多架无人机协同配送的路径规划方法，整体算法思路参照混合遗传算法，考虑到多车多机问题模型的复杂性，传统的单一编码方式难以进行求解，故创新性地采用总路径编码、卡车路径编码、无人机路径编码三者结合的编码方式，并设计了四种局部搜索算子，能在较大规模和较多约束的复杂条件下，在短时间内给出可行有效的调度方案。

具体步骤包括：

步骤一、构建物流配送中多卡车多架无人机路径规划的数学模型。具体包括：首先构建有向图 $G = \langle V, E \rangle$ ， V 代表节点， E 代表弧； $C_0 = \{1, 2, 3, 4, \dots, c\}$ 代表用户点集。另外，尽管仓库点只有一个，但将仓库点分配到两个唯一的节点号 $0, c+1$ 。定义包含起点的节点集为 $C_1 = C_0 \cup \{0\}$ ；定义包含终点的节点集为 $C_2 = C_0 \cup \{c+1\}$ 。其次，构建最小化目标函数。最后一辆卡车或最后一架无人机尽可能快得回到仓库，公式为 $\min\{\max[\max\{t_{k,c+1}^T, k \in T\}, \max\{t_{d,c+1}^D, d \in D\}]\}$ ，其中 $t_{m,c+1}^T$ 为第 k 辆卡车回到仓库点的时间， $t_{n,c+1}^D$ 为第 d 架无人机回到仓库点的时间。确定决策变量 u_i, p_{ij} ，其中 u_i 为节点 i 被访问的顺序，以消除子环路，而 p_{ij} 为0-1变量，若其值为1，则表示在卡车路径中，节点 i 比节点 j 先访问。确定约束条

件。约束条件主要分为以下几类：服务路径限制、无人机服务限制，无人机卡车交互限制。

(1) 服务路径限制的表达式为

$$\begin{aligned}
\sum_{k \in K} z_{0jk} &= m \quad \forall j \in C_0 \\
\sum_{d \in D} y_{0jd} &= n \quad \forall j \in C_0 \\
\sum_{k \in K} z_{i0k} &= m \quad \forall i \in C_0 \\
\sum_{d \in D} y_{i0d} &= n \quad \forall i \in C_0 \\
\sum_{\substack{i,j \in E \\ k \in K}} x_{ijk} + \sum_{\substack{i,j \in E \\ d \in D}} y_{ijd} + \sum_{\substack{i,j \in E \\ k \in K}} z_{ijk} &= 1 \quad \forall j \in C_0 \\
\sum_{\substack{i,j \in E \\ k \in K}} x_{ijk} + \sum_{\substack{i,j \in E \\ d \in D}} y_{ijd} + \sum_{\substack{i,j \in E \\ k \in K}} z_{ijk} &= 1 \quad \forall i \in C_0
\end{aligned}$$

其中 x_{ijk} 为 0-1 变量，若其值为 1，则表示第 k 辆卡车独立经过弧 ij ，否则为 0； z_{ijk} 为 0-1 变量，若其值为 1，则表示第 k 辆卡车搭载无人机经过弧 ij ，否则为 0； y_{ijk} 为 0-1 变量，若其值为 1，则表示第 d 架无人机经过弧 ij ，否则为 0； K 为卡车集， D 为无人机集；前四个式子约束所有卡车和无人机离开仓库并返回仓库，后两个式子约束用户点的出入度均为 1。

(2) 无人机服务限制的表达式为 $\sum_{i,j \in E} y_{ijd} \tau_{ij}^D \leq L \quad \forall d \in D$ ，其中 τ_{ij}^D 表示无人机从节点 i 到节点 j 所需要的时间， L 为无人机最大续航时间。该约束条件保证无人机飞行时间不超过其最大续航时间。

(3) 无人机卡车交互限制的表达式为

$$\begin{aligned}
t_{d,i}^D &\geq t_{k,i}^T - M \left(1 - \sum_{\substack{j \in C_0 \\ i \neq j}} y_{ijd} \right) \quad \forall i \in C_1 \\
t_{d,i}^D &\leq t_{k,i}^T + M \left(1 - \sum_{\substack{j \in C_0 \\ i \neq j}} y_{ijd} \right) \quad \forall i \in C_1
\end{aligned}$$

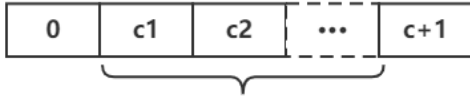
$$t_{d,j}^D \geq t_{k,j}^T - M \left(1 - \sum_{\substack{i \in C_0 \\ i \neq j}} y_{ijd} \right) \quad \forall j \in C_2$$

$$t_{d,j}^D \leq t_{k,j}^T + M \left(1 - \sum_{\substack{i \in C_0 \\ i \neq j}} y_{ijd} \right) \quad \forall j \in C_2$$

前两个式子约束卡车和无人机在节点*i*同步分离，即若卡车和无人机的出发时间必须相同。需要注意的是，这里的第 *d* 架无人机是从第 *k* 辆卡车上起飞。同理，后两个式子约束卡车和无人机在节点*j*相遇时到达时间相同。需要注意的是，这里的第 *d* 架无人机不一定从第 *k* 辆卡车上起飞，即当卡车（或无人机）先到达节点*j*时，必须要等待无人机（或卡车）到达。

步骤二、构建总路径编码、卡车路径编码和无人机路径编码。具体为，设共有 *m* 辆卡车和 *n* 架无人机；在总路径编码中起点为 0，终点为 *c* + 1，，其余各点为用户点的节点号；在卡车路径编码中，总共有 *m* 组以 0 和 *c* + 1 为两端的片段，分别表示 *m* 辆卡车的路径，其中的数值为用户点在总路径的相对位置；在无人机路径编码中的无人机起飞点和降落点为卡车路径编码中的相对位置，其余非负数值为用户点在总路径路径编码中的相对位置，而-1 为一级区分符，用以区分不同无人机；-2 为二级区分符，用以区分同一无人机的不同行程编码如下图所示。

总路径编码



用户点

卡车路径编码



*m*组卡车路径

卡车路径编码

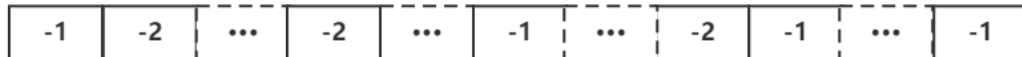


图 1 编码示意图

步骤三、初始化算法基础参数。具体包括种群规模 *popsi*ze，用户点个数 *c*，卡车数量 *m*，每辆卡车上的无人机数量 *n*，初始温度 *t*₀，降温方式 $t_k = \frac{t_0}{1+k}$ ，终止温度 *t*_{*F*}，内循环次数 *N*，无人机飞行时间限制 *L*，交叉概率系数 *P*_{*c*}，变异概率系数 *P*_{*m*}。

步骤四、初始化个体参数，即随机生成 mTSP 问题的解。具体操作为在总路径编码中的起点和终点之间随机生成不重复的整数 $c_i \in [1, c]$ ；在卡车路径编码中每组卡车路线中随机生成不重复的整数 $k_i \in [1, c]$ ，代表用户点在总路径编码中的相对位置。此时需要确保每个用户点都被分配到某一卡车回路中，以确保每个用户点有且只被访问一次。无人机路径编码产生 $n+1$ 组一级标识符-1，用以区分不同无人机路线。

步骤五、计算个体适应度。具体为，先判断个体是否满足约束条件，即是否为可行解，对可行解，其适应度计算公式为 $F = e^{-af}$ ，其中 $f = \min\{\max[\max\{t_{k,c+1}^T, k \in T\}, \max\{t_{d,c+1}^D, d \in D\}]\}$ ，为目标函数， a 为常数，根据问题规模进行适当选择。记录并保存当前的最优解 s_{best}

步骤六、采用轮盘赌选择算子对个体进行选择。具体操作为，计算当代所有个体的适应度，求出每个个体的选择概率 $P_i = \frac{F_i}{\sum_{j=1}^N F_j}$ 和累计概率 $q_i = \sum_{j=1}^i P_j$ ，生成 popsize 个 $[0, 1]$ 之间的随机数 r 。若 $r < q_1$ ，则选择个体 1；否则选择个体 k ，使得 $q_{k-1} < r \leq q_k$ 。

步骤七、采用顺序交叉算子对选中个体进行交叉。具体为，根据交叉概率 P_c 选择父代个体，在选中的两个个体的总路径编码中任意选择一段连续的片段进行交换，剩余基因按照所在个体的原先顺序依次排列。采用总路径编码的使得交叉操作更加快速方便，而卡车路径编码选择用户点在总路径编码中的相对位置，使得在总路径编码交叉后，卡车路径编码就可以代表一段新的卡车路线。

步骤八、进行局部搜索，即对产生的解进行扰动。局部搜索采用三种算子，分别为 Swap, DroneInsertion 和 TruckInsertion。

- (1) Swap 算子的具操作为在总路径编码中随机选择两个用户点进行位置调换，产生领域新解，记录可行的最优解 s_1 。
- (2) DroneInsertion 算子是将原先由卡车服务的用户点变为无人机服务，具体操作为将卡车路径编码中的一个点直接插入到无人机路径编码中，若插入后的两侧均为区分符，说明会生成一条新的无人机路径，则在该点两段选择无人机的起飞点和降落点，再在起飞点的左侧和降落点的右侧生成二级区分符-2。需要注意的是由于起飞点和降落点的数值是卡车路径编码中的相对位置，为了避免不可行解，降落点的数值必须大于起飞点的数值。若插入后两侧不均为区分符，则说明该用户点插入到一条已有的无人机路径中，记录可行的最优解 s_2 。
- (3) TruckInsertion 算子是将某一卡车回路中的用户点插入到另一卡车回路中，具体操作为将 $0 \sim c + 1$ 之间的某点插入到另一组 $0 \sim c + 1$ 之中，记录可行的最优解 s_3 。

步骤九、根据 Metropolis 准则判断是否接受新解。将局部搜索前的解记为 s_{now} ，选择 s_1, s_2, s_3 的适应度的最大值记作 s_{new} ，若 $F(s_{new}) > F(s_{now})$ ，则 s_{new} 加入到新群体中，并且 $s_{now} = s_{new}$ ；否则根据 Metropolis 准则判断是否接受新解。

步骤十、判断该温度下解是否达到稳定。若内循环次数达到 N 则，则认为该温度下解已达到稳定。

步骤十一、若已稳定，则返回步骤五；否则返回步骤八，进行迭代。

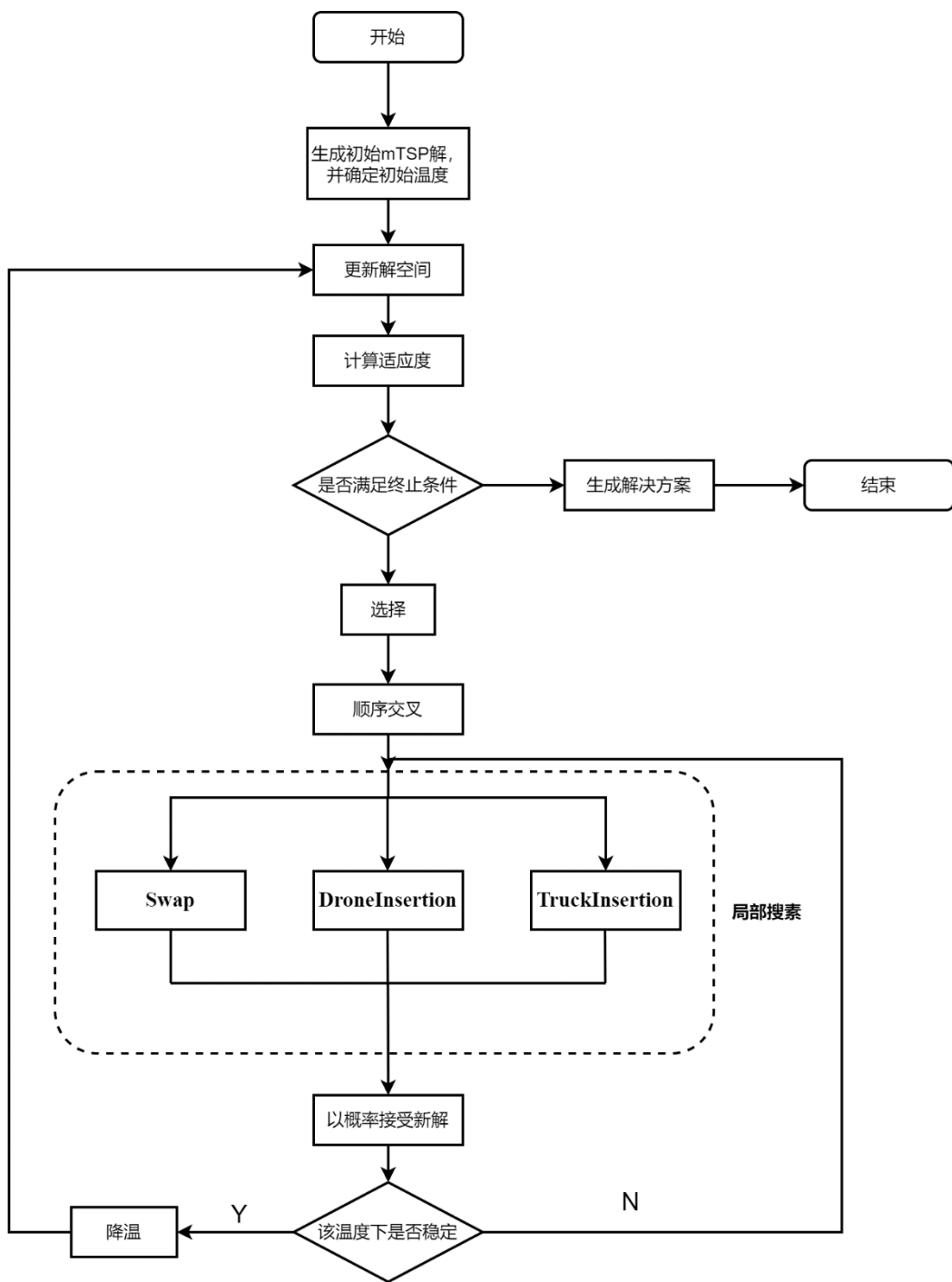


图 2 算法流程图