

TỐI ƯU HÓA LỢI NHUẬN CHO BÀI TOÁN NGƯỜI GIAO HÀNG LẬP LỊCH DRONE SONG SONG

PROFIT OPTIMIZATION FOR THE PARALLEL SCHEDULING TRAVELLING SALESMAN PROBLEM

Đặng Thị Hương Giang

Khoa Điện tử, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

Đến Tòa soạn ngày 07/04/2023, chấp nhận đăng ngày 27/05/2023

Tóm tắt: Sự gia tăng gần đây trong việc sử dụng phương tiện bay không người lái drone cho việc giao hàng, đã đáp ứng yêu cầu tối ưu hóa sự vận hành của quy trình giao hàng. Chúng tôi xem xét đến bài toán tối ưu hóa Định tuyến Lập lịch Drone Song song xem xét đến lợi nhuận. Trong bài toán này, xe tải giao hàng và đội drones được sử dụng kết hợp. Xe tải chịu trách nhiệm phục vụ khách hàng ở những nơi vượt quá tầm bay của drone hay những khách hàng có gói hàng nặng. Chúng tôi đã nghiên cứu mục tiêu lợi nhuận của quy trình giao hàng và đề xuất dùng phương pháp SA-ILS (Simulated Annealing - Iterated Local Search) giải bài toán người giao hàng lập lịch drone song song nhằm đưa ra lời giải với lợi nhuận tối đa.

Từ khóa: SA, SA-ILS, Drone, thuật toán mô phỏng luyện kim, lập lịch.

Abstract: The increasingly larger interest with Drone unmaned aerial vehicle in application to the cargo delivery has gradually met the needs of optimizing the operation of delivery process. We focus on researching the optimization problem of Parallel Drone Scheduling Vehicle Routing (PDSVRP) with profit objective. At this problem, it uses combination between truck and drone fleets. The truck is responsible for making delivery to customer beyond the drone's flight distance or heavy packages. We researched the profit objectives of delivery process and recommended the SA-ILS method (Simulated Annealing-Iterated Local Search) for solving the parallel drone scheduling travelling salesman problem for a solution with maximum profit.

Keywords: SA, SA-ILS, Drone, annealing algorithms, scheduling.

1. GIỚI THIỆU

Bài toán người chào hàng lập lịch drone song song PDSTSP (*Parallel drone scheduling travelling salesman problem*) được lấy nguyên mẫu của Murray và Chu (2015) [1], mở ra hướng nghiên cứu mới về định tuyến cho xe tải và drone. Mục tiêu của bài toán PDSTSP là nhằm tìm được đường đi Hamilton cho xe tải và tập hợp các chuyến đi cho drone mà theo đó giúp giảm thiểu thời gian hoàn tất nhiệm vụ giao phát và quay về kho hàng.

Ở bài toán PDSTSP, tốc độ phục vụ khách hàng, chất lượng phục vụ của hệ thống vận chuyển và chi phí vận chuyển được quan tâm xem xét. Một số tài liệu đã khảo cứu hàm mục tiêu làm giảm thiểu tổng chi phí vận chuyển, như là: Mathew và cộng sự [2] đã xem xét Bài toán Giao hàng Hỗn hợp HDP (*heterogeneous delivery problem*); Hà và cộng sự [3] đã nghiên cứu Bài toán người chào hàng với drone chi phí tối thiểu TSP-D (*traveling salesman problem with drone*). Hàm mục tiêu tương tự cũng được nghiên cứu trong HVDRP

(*hybrid vehicle-drone routing problem*) cho các dịch vụ đón và giao hàng được Karak và Abdelghany [4] giới thiệu. Mbiadou Saleu và cộng sự (2018) [5] đã đưa ra phép siêu suy nghiệm gồm hai bước: Bước mã hóa để chuyển cách giải PDSTSP khả thi thành tuần tự khách hàng; và bước giải mã để phá vỡ tuần tự khách hàng thành lộ trình cho xe tải và các chuyến đi cho drones. NM.Anh và cộng sự (2021) đã dùng thuật toán *SISSRs* để giải bài toán chi phí tối thiểu cho mô hình kết hợp xe tải và drone [8].

Việc giao hàng bằng drone hứa hẹn là một cuộc cách mạng trong ngành kinh doanh logistics, song một vài những rào cản kỹ thuật đáng lưu tâm vẫn cần phải khắc phục trước khi drone đạt được sự chấp nhận rộng rãi trong ngành thương mại này. Trong khi nhiều nghiên cứu nhắm vào việc khắc phục những vấn đề kỹ thuật đặt ra với giao hàng bằng drone, thì chỉ có một số ít nghiên cứu đề cập đến những thách thức vận hành. Khác với kiểu giao hàng bằng xe tải truyền thống, kiểu giao hàng bằng drone có một số ưu điểm chính:

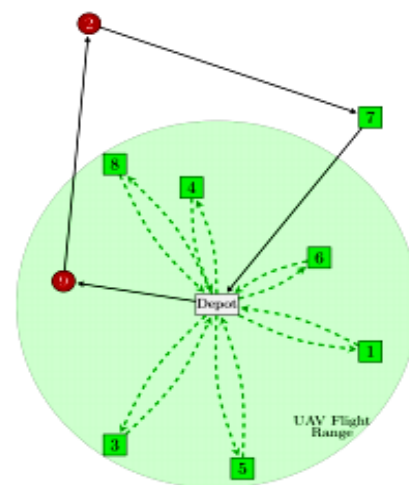
- Có thể điều khiển mà không cần phi công.
- Tránh được tắc đường do có thể bay bên trên.
- Nhanh hơn xe tải.
- Chi phí đi lại thấp hơn xe tải truyền thống.

Tuy nhiên, vì drones hoạt động dùng năng lượng pin, khoảng cách di chuyển tối đa và khả năng chuyên chở của nó hạn chế. Ngược lại, xe tải có ưu thế về năng lực di chuyển trong khoảng cách xa. Nó có thể mang được hàng hóa nặng và cồng kềnh nhiều kích cỡ, nhưng lại chậm và có chi phí cao hơn nhiều.

Ưu nhược điểm của xe tải và drone sẽ bù đắp cho nhau. Nên việc sử dụng kết hợp đội

drones và xe tải trong bài toán người giao hàng lập lịch drone song song với chi phí tối thiểu PDSVRP (min-cost Parallel Drone Scheduling Vehicle Routing Problem) mang lại hiệu quả đáng mong đợi. Đội xe tải và nhóm drones được phối hợp để giao phát các gói hàng tới tập hợp khách hàng, mỗi khách hàng được phục vụ một lần. Xe tải sẽ thực hiện nhiệm vụ trong phạm vi một chuyến đi từ kho hàng, qua tất cả các vị trí khách hàng được phân bổ và quay lại kho hàng. Drones sẽ đi lại giữa kho hàng và khách hàng, mỗi chuyến đi là một lần giao phát. Tổng thời gian làm việc của mỗi phương tiện (cả xe tải và drones) không vượt quá giới hạn nhất định. Mô hình bài toán PDSVRP được mô tả như trong hình 1: Đường nét liền biểu thị tuyến xe tải, đường đứt nét biểu thị tuyến drones.

Mặc dù có một số nghiên cứu về việc giải bài toán PDSTSP với chi phí tối thiểu hay thời gian làm việc tối thiểu, nhưng không có nghiên cứu nào đã thực hiện làm tối đa hóa mục tiêu lợi nhuận cho bài toán. Với mục tiêu lợi nhuận của bài toán giao hàng, mỗi khách hàng đều được xem xét với một lợi nhuận và nghiên cứu này nhằm tạo ra được kế hoạch thu thập lợi nhuận cao nhất trong phạm vi ràng buộc thời gian làm việc tối đa của xe tải và drone.



Hình 1. Mô hình lập lịch cho xe tải và tập drone [9]

2. XÂY DỰNG MÔ HÌNH TOÁN HỌC CỦA BÀI TOÁN

Bài toán PDSTP được biểu thị trên đồ thị có hướng $G=\{N,E\}$, tập hợp node $N=\{0, 1, \dots, n, n+1\}$ có node kho hàng (node 0) và bản sao của nó (node $n+1$) và tập hợp khách hàng $C=\{1, 2, \dots, n\}$. Tập khách hàng chỉ dùng xe tải $C_T \subseteq C$. Như vậy, đặt $C_F \subseteq C / C_T$ là tập khách hàng phù hợp với cả hai phương thức giao phát xe tải và drone. Mỗi khách hàng $i \in C$ sẽ gắn với lợi nhuận p_i .

Có một xe tải và một tập hợp m drones (D) có khả năng đáp ứng các nhiệm vụ giao phát tại kho hàng. Khoảng cách yêu cầu đối với xe tải để chuyển dịch từ node $i \in C$ tới node $j \in N$ với ($j \neq i$) được xác định bằng d_{ij} và d'_{ij} là khoảng cách bay từ node i tới j của drone. trên thực tế drones sẽ bay tới bay đi giữa kho hàng và khách hàng, đặt $d'_i = d'_{0i} + d'_{i0}$ là tổng khoảng cách bay của drone để phục vụ khách hàng $i \in C_F$. Với cùng cách thức đó, t_{ij} biểu thị thời gian tiêu tốn bởi xe tải để đi lại trên cạnh $(i, j) \in E$, và t'_i là thời gian bay phải tiêu tốn khi drone phục vụ khách hàng $i \in C_F$, (bao gồm thời gian cho các hoạt động: bay đi bay về giữa kho hàng và khách hàng, bốc và dỡ gói hàng và thay pin (nếu có) cho drone).

Giả định:

- Một drone chỉ mang một gói một lần và phải quay về kho hàng sau mỗi lần giao phát.
- Dữ liệu đầu vào, chẳng hạn như số lần lái và bay tương ứng của xe tải và drone, được giả định là tất định.
- Sức chở của xe tải là đủ lớn để có thể phục vụ được tất cả khách hàng.

Mục tiêu của bài toán này là nhằm tìm được tập hợp các tuyến cho đội hỗn hợp xe tải và drone làm tối đa hóa tổng lợi nhuận có được của tất cả các phương tiện và làm thỏa mãn

các ràng buộc sau đây:

- Mỗi drone sẽ phục vụ một khách hàng một lần bằng cách thực hiện các chuyến bay tới bay lui trong phạm vi khoảng bay giới hạn.
- Mỗi phương tiện (cả xe tải và drone) sẽ phụ thuộc vào thời gian làm việc tối đa T .

Các tham số sử dụng:

y_i : bằng 1 nếu xe tải di chuyển qua node $i \in N$, bằng 0 nếu xe tải không qua node i .

x_{ij} : bằng 1 nếu xe tải đi từ node $i \in N$ tới node $j \in N$, trường hợp còn lại bằng 0.

z_{ij} : bằng 1 nếu khách hàng $i \in C_F$ được phục vụ bằng drone $j \in D$, trường hợp còn lại bằng 0.

τ_i : biến liên tục biểu thị thời gian xe tải tới khách hàng $i \in C$.

Bài toán PDSTSP với lợi nhuận tối đa có thể được diễn giải như sau:

Tối đa hóa

$$\sum_{i \in C} p_i y_i \sum_{i \in C_F} \sum_{j \in D} p_j z_{ij} \quad (1)$$

Ràng buộc

$$\sum_{i \in C_T} y_i + \sum_{i \in D, i \in C_F} z_{ij} = 1 \quad \forall i \in C_F \quad (2)$$

$$y_i = 1 \quad \forall i \in \{0, n+1\} \quad (3)$$

$$\sum_{i \in N \setminus \{0\}} x_{0i} = 1 \quad (4)$$

$$\sum_{i \in N \setminus \{n+1\}} x_{i,n+1} = 1 \quad (5)$$

$$\sum_{i \in N \setminus \{i\}} x_{ij} = y_i \quad \forall i \in N \setminus \{n+1\} \quad (6)$$

$$\sum_{i \in N \setminus \{i\}} x_{ij} = y_i \quad \forall i \in N \setminus \{0\} \quad (7)$$

$$T_i + t_{ij} \leq T_j + (1 - x_{ij}) \times (T + t_{ij}) \quad (8)$$

$\forall i \in N, j \in N, i \neq j$

$$T_{n+1} \leq T \quad (9)$$

$$\sum_{i \in C_F} z_{i1} t_i' \leq T \quad (10)$$

$$\sum_{i \in C_F} z_{i,j} t_i' \leq \sum_{i \in C_F} z_{i,j-1} t_i' \quad \forall i \in D \setminus \{1\} \quad (11)$$

$$y_i \in \{0,1\} \quad \forall i \in N \quad (12)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in N, j \in N \quad (13)$$

$$z_{ij} \in \{0,1\} \quad \forall i \in C_F, j \in D \quad (14)$$

$$T_i \in \{0,T\} \quad \forall i \in N \quad (15)$$

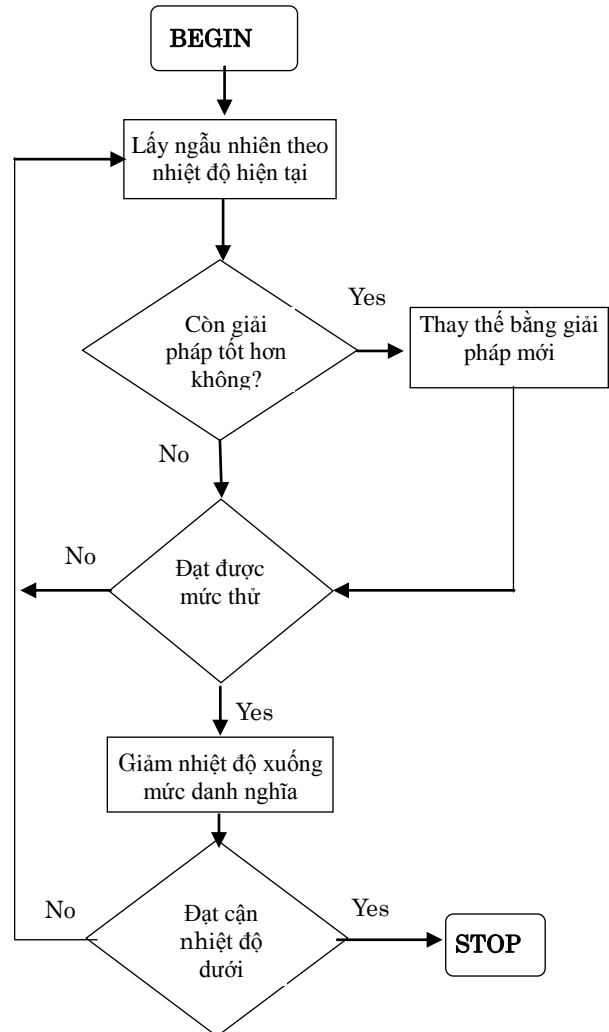
Hàm mục tiêu 1 tối đa hóa lợi nhuận. Ràng buộc 2 đảm bảo mỗi khách hàng chỉ được phục vụ bằng 1 xe tải hoặc 1 drone. Ràng buộc 3 đảm bảo rằng xe tải đi qua kho hàng và bản sao của nó. Ràng buộc 4 và 5 yêu cầu xe tải bắt đầu và kết thúc tuyến đi tại kho hàng. Sự duy trì đảm bảo tuyến chạy của xe tải được thỏa mãn theo ràng buộc 6 và 7. Ràng buộc 8 biểu thị mối quan hệ giữa biến τ và x . Ràng buộc này cũng đảm bảo rằng không có lộ trình phụ nào trên tuyến của xe tải. Ràng buộc 9, 10 là các ràng buộc thời gian làm việc tối đa của xe tải và drone. Ràng buộc 11: drone j phải kết thúc thời gian làm việc không chậm hơn drone $j-1$. Cuối cùng, các ràng buộc 12, 13, 14 và 15 nhằm xác định miền của các biến.

3. THUẬT TOÁN MÔ PHÒNG LUYỆN KIM CẢI TIẾN SA-ILS (*Simulated Annealing - Iterated Local Search*)

3.1. Thuật toán Mô phỏng luyện kim SA

Thuật toán SA được đề xuất năm 1983 bởi Scott Kirkpatrick [7]. Tên gọi và ý tưởng của thuật toán bắt nguồn từ kỹ thuật luyện kim trong ngành luyện kim, một kỹ thuật sử dụng một cách khéo léo việc nung nóng rồi làm nguội vật liệu kim loại nhằm làm tăng kích thước và giảm những chỗ khuyết, lõm của chúng. Nhiệt độ làm cho các nguyên tử trong

vật liệu trở nên mất liên kết, rời khỏi vị trí ban đầu của chúng và di chuyển ngẫu nhiên vào những nơi có năng lượng cao hơn; việc làm nguội sẽ làm tăng khả năng nhận được một trạng thái có nội năng thấp hơn so với ban đầu. Lưu đồ thuật toán như sau:



Hình 2. Lưu đồ giải thuật SA

Các bước như sau:

Bước 1: Khởi tạo bộ đếm $i = 0$, nhiệt độ ban đầu T_0 , trạng thái ban đầu x

Bước 2: (Quá trình Metropolis TPM): Tạo trạng thái x' theo cơ chế nhiễu loạn.

Bước 3: Nếu thỏa mãn bước 2, hệ thống đạt trạng thái cân bằng thì chuyển sang bước 4, ngược lại thì quay về bước 2.

Bước 4: Quá trình giảm nhiệt độ để hệ thống đạt trạng thái cân bằng

Bước 5: Quá trình hội tụ: xác định nhiệt độ tại trạng thái cân bằng và cho kết quả nghiệm, ngược lại tăng $i = i+1$ và quay bước 2.

3.2. Thuật toán Mô phỏng luyện kim cải tiến SA-ILS

Để giải bài toán PDSTSP tối đa lợi nhuận tác giả cải tiến thuật toán SA bằng cách bổ sung thêm bước tìm kiếm cục bộ ILS (Iterated Local Search) nhằm khắc phục nhược điểm không thoát được điểm tối ưu cục bộ của thuật toán SA tìm đến điểm tối ưu toàn cục của bài toán.

Gọi Δ là sai số của hàm mục tiêu giữa cách giải mới và cách giải hiện có, và T là nhiệt độ hiện tại. T_0 là nhiệt độ khởi tạo của T . Nhiệt độ T giảm trong quá trình tìm kiếm thông qua quy trình làm mát được kiểm soát bởi hệ số α . $e^{\frac{\Delta}{T}}$ là xác suất giới hạn cho việc chấp nhận cách giải mới.

Thuật toán SA-ILS đề xuất được trình bày trong Thuật toán 1.

Thuật toán 1. Thuật toán SA-ILS

```

 $X \leftarrow \text{bestInsertion}(\emptyset);$ 
 $x_{best} \leftarrow X;$ 
 $T \leftarrow T_0;$ 
 $d \leftarrow 1;$ 
 $iter \leftarrow 0;$ 
while  $iter < maxIter$  do
     $iter \leftarrow iter + 1;$ 
     $X \leftarrow \text{perturbation}(X, d);$ 
     $X' \leftarrow \text{localSearch}(X);$ 
     $A \leftarrow \text{profit}(X') - \text{profit}(X);$ 

```

```

if  $d < dmax$  then
     $d \leftarrow d + 1;$ 
    if  $\text{rand}(0,1) < e^{\frac{\Delta}{T}}$  then
         $X < X';$ 
        if  $\text{profit}(X) > \text{profit}(X_{best})$  then
             $X_{best} \leftarrow X;$ 
         $d \leftarrow 1;$ 

```

$T \leftarrow \alpha \times T;$

Return X_{best}

Hàm $\text{bestInsertion}(X)$ sử dụng thuật toán tham lam [6] để xây dựng cách giải từ điểm xuất phát ($X \neq \emptyset$). Trước hết hàm này sẽ chèn các node sử dụng drone theo một trật tự giảm dần về mức lợi nhuận trên chi phí thời gian $\frac{p_i}{t_i}$.

Quy trình này được lặp lại với tất cả các drone. Khách hàng còn lại sau đó được được chèn vào tuyến xe tải để tối đa hóa lợi nhuận. Thuật toán dừng lại khi không có thêm việc chèn nào có thể xảy ra. Lợi nhuận theo công thức (1) mỗi lần chèn có thể được đánh giá theo thời gian không đổi.

Cơ chế nhiễu loạn

Nhằm tìm ra phương án tốt nhất, cơ chế nhiễu loạn được áp dụng thông qua hàm $\text{perturbation}(X, d)$, như được mô tả trong thuật toán 2.

Thuật toán 2. Perturbation

```

Input:  $X, d$ 
 $p \leftarrow \text{randInt}(1, d);$ 
 $X' \leftarrow \text{randomRemove}(X, p);$ 
 $X' \leftarrow \text{localSearch}(X);$ 
 $X' \leftarrow \text{bestInsertion}(X);$ 
return  $X'$ 

```

Trước hết sẽ tách số lượng (được tạo ra ngẫu nhiên giữa 1 và d) khách hàng khỏi cách giải

hiện tại X và chạy toán tử $localSearch(X')$ để tìm giải pháp tốt nhất cho nhóm khách hàng này. Sau đó, pha tái lập được xử lý thông qua hàm $bestInsertion()$ nhằm tạo ra lộ trình giao hàng có lợi nhuận tốt nhất tại thời điểm hiện tại.

Tại mỗi lần lặp của phương pháp ILS (tìm kiếm cục bộ lặp lại), quy trình $perturbation()$ được sử dụng với d là số lượng khách hàng được tách ra. d được khởi tạo là 1 và tăng 1 cho mỗi lần tìm kiếm mà không có cải thiện nào cho đến khi đạt được d_{max} . Khi cách giải tốt nhất được tìm thấy, quy trình ILS dừng lại cho nhóm khách hàng này và d sẽ được tái khởi tạo bằng 1 cho nhóm khách hàng được tách ra tiếp theo.

Nhằm thực tế hóa các cách giải, trong khoảng thời gian gọi $bestInsertion()$, nhiều sẽ được cộng vào cách tính toán lợi nhuận trên chi phí. Lợi nhuận trên chi phí sau khi cộng nhiều như sau:

$$poc_i = \frac{p_i \div \max_j p_j}{cost_i \div \max_j p_j} + r \quad (16)$$

$$\text{Với } r \sim N(0,1) \quad (17)$$

Quy trình tìm kiếm cục bộ (ILS).

Hai vùng lân cận sau đây được sử dụng trong hàm tìm kiếm cục bộ:

Hoán đổi: sẽ trao đổi sự phân giao cho hai phương tiện.

Di chuyển: tách khách hàng khỏi phương tiện hiện tại của nó và phân giao nó tới một phương tiện khác.

4. KẾT QUẢ THÍ NGHIỆM

Chúng tôi sử dụng thuật toán SA và SA-ILS ở trên để giải các trường hợp thí nghiệm với số lượng khách hàng $n \in \{10, 30\}$; thời gian làm việc tối đa $T \in \{0, 5, 1\}$ và kích cỡ của đội drone $m \in \{1, 3, 5\}$. Mô hình này được thực hiện trên nền C++ thông qua thư viện IBM ILOG CPLEX 12.10 Concert và chạy trên máy macOS 11.1 với bộ xử lý 2,3 GHz Dual-Core Intel Core i5. Giới hạn thời gian của mỗi trường hợp được đặt là 30 phút. Chúng tôi đã thực hiện phép suy nghiệm SA và SA-ILS với cùng một dạng thức. Mô hình đặt ra 10-khách hàng và 30-khách hàng; các thông số đặt ra: $maxIter$ được thiết lập với 25000 lần lặp; nhiệt độ ban đầu T_0 được đặt là $-\max Profit \div \ln(0.5)$; d_{max} được đặt là 10 và hệ số kiểm soát nhiệt độ α được đặt là 0,99975. Phép suy nghiệm được chạy 10 lần và mục tiêu tốt nhất và mục tiêu trung bình sẽ được ghi chép. So sánh chi tiết có thể thấy trong bảng 1 và bảng 2, bảng 3, bảng 4.

Bảng 1. Phương pháp SA và SA-ILS các trường hợp 10-khách hàng, $T=0.5$

Instance	T = 0.5.					
	SA		Best	SA-ILS		
	Profit	R.Gap%		Avg	Time	Gap%
10-c-c-0	43.9503	0.00	43.9503	43.9503	0.09	0.00
10-c-c-1	46.0901	0.00	46.0901	46.0901	0.08	0.00
10-c-c-2	28.9319	0.00	28.9319	28.9319	0.08	0.00
10-c-e-0	20.1762	0.00	20.1762	20.1762	0.07	0.00
10-c-e-1	7.02177	0.00	7.02177	7.02177	0.10	0.00
10-c-e-2	33.9582	0.00	33.9582	33.9582	0.14	0.00

Instance	T = 0.5.					
	SA		Best	SA-ILS		
	Profit	R.Gap%		Avg	Time	Gap%
10-c-r-0	53.5087	0.00	53.5067	53.5067	0.16	0.00
10-c-r-1	35.1715	0.00	35.1745	35.1715	0.09	0.00
10-c-r-2	26.5277	0.00	26.5277	26.5277	0.10	0.00
10-r-c-0	33.4198 OS	0.00	33.4198	33.119s	0.12	0.00
10-r-c-1	52.9405	0.00	52.9405	52.9405	0.24	0.00
10-r-c-2	33.5392	0.00	33.5392	33.5392	0.16	0.00
10-r-e-0	22.7788	0.00	22.7786	22.7786	0.08	0.00
10-r-e-1	16.793	0.00	16.793	16.793	0.07	0.00
10-r-e-2	14.5689	0.00	14.5689	14.5689	0.11	0.00
10-r-r-0	27.0319	0.00	27.0319	27.0319	0.07	0.00
10-r-r-1	29.091	0.00	29.091	29.091	0.09	0.00
10-r-r-2	20.1295	0.00	20.1295	20.1295	0.06	0.00
Avg	22.6105	0.0000	30.3128	28.4559	0.11	0.00

Bảng 2. Phương pháp SA và SA-ILS các trường hợp 10-khách hàng, T=1

Instance	T= 1					
	SA		Best	SA-ILS		
	Profit	R.Gap%		Avg	Time	Gap%
10-c-c-0	60.5285	0.00	60.5285	60.5285	0.18	0.00
10-c-c-1	60.5285	0.00	60.5285	60.5285	0.15	0.00
10-c-c-2	54.1568	0.00	54.1568	54.1569	0.23	0.00
10-c-e-0	53.5067	0.00	53.5067	53.5067	0.22	0.00
10-c-e-1	30.3665	0.00	30.3665	30.3665	0.10	0.00
10-c-e-2	50.5364	0.00	50.5364	50.5364	0.15	0.00
10-c-r-0	60.5285	0.00	60.5285	60.5285	0.14	0.00
10-c-r-1	53.5067	0.00	53.5067	53.5067	0.11	0.00
10-c-r-2	60.5285	0.00	00.5285	60.5285	0.13	0.00
10-r-c-0	60.5285	0.00	60.5285	60.5285	0.13	0.00
10-r-c-1	52.9405	0.00	52.9405	52.9405	0.13	0.00
10-r-c-2	52.5113	0.00	52.5113	52.5113	0.12	0.00
10-r-e-0	37.3475	0.00	37.3475	37.3475	0.10	0.00
10-r-e-1	39.0133	0.00	39.0133	39.0133	0.10	0.00
10-r-e-2	24.9903	0.00	24.9903	21.9903	0.08	0.00

Instance	T = 1					
	SA		Best	SA-ILS		
	Profit	R.Gap%		Avg	Time	Gap%
10-r-r-0	36.5883	0.00	36.5883	36.5883	0.09	0.00
10-r-r-1	15.2771	0.00	15 2771	45.2771	0.09	0.00
10-r-r-2	26.5174	0.00	26.5174	26.5174	0.07	0.00
Avg	46.1056	0.0000	41.8942	46.3840	0.13	0.00

Bảng 3. Phương pháp SA và SA-ILS các trường hợp 30-khách hàng, T=0.5

Instance	T = 0.5					
	SA		Best	SA-ILS		
	Profit	R.Gap%		Avg	Time	Gap%
30-c-c-0	34.1299	288.06	34.1299	34.1299	0.13	0.00
30-c-c-1	95.0054	25.93	96.5282	96.5282	0.33	-1.58
30-c-c-2	89.0257	2S.11	89.0257	89.0257	0.45	0.00
30-c-c-0	26.7939	300.82	28.698	28.698	0.15	-6.63
30-c-e-1	38.282	233.47	38.6051	38.6651	0.24	-0.84
30-c-e-2	60.6913	159.38	60.6913	60.6913	0.22	0.00
30-c-r-0	49.4799	177.53	51.8841	51.8841	0.20	-4.63
30-c-r-1	60.5002	151.93	60.5003	60.5002	0.30	0.00
30-c-r-2	53.1642	98.51	53.1642	53.1642	0.27	0.00
30-r-c-0	42.3231	0.00	42.3231	42.3231	0.24	0.00
30-r-c-1	86.3708	54.19	88.0827	88.0827	0.48	-1.94
30-r-c-2	60.4699	141.36	60.4699	60.4698	0.27	0.00
30-r-e-0	63.8935	105.45	63.8935	63.8935	0.24	0.00
30-r-e-1	41.6073	184.04	41.6073	41.6073	0.20	0.00
30-r-e-2	61.4658	89.07	61.4658	61.4658	0.28	0.00
30-r-r-0	52.3789	102.69	52.3789	52.3789	0.26	0.00
30-r-r-1	49.1944	202.67	49.1944	49.1944	0.29	0.00
30-r-r-2	57.8256	171.27	57.8256	57.8256	0.25	0.00
Avg.	56.8112	138.1317	57.2482	57.2515	0.27	-0.87

Bảng 4. Phương pháp SA và SA-ILS các trường hợp 30-khách hàng, T=1

Instance	T = 1					
	SA		Best	SA-ILS		
	Profit	R.Gap%		Avg.	Time	Gap%
30-c-c-0	129.648	24.61	129.648	129.648	0.58	0.00
30-c-c-1	143.496	12.59	151.414	151.413	0.53	-5.23
30-c-c-2	133.244	21.25	140.102	140.102	0.59	-4.90
30-c-e-0	67.6403	138.85	72.803	72.803	0.28	-7.09
30-c-e-1	112.118	13.86	116.292	114.929	0.46	-3.59
30-c-e-2	98.8922	63.37	98.8922	98.8922	0.30	0.00
30-c-r-0	122.144	32.27	126.756	126.756	0.60	-3.64
30-c-r-1	133.621	20.91	134.973	134.973	0.62	-1.00
30-c-r-2	117.721	37.24	122.855	122.855	0.56	-4.18
30-r-c-0	100.167	61.29	103.912	103.912	0.39	-3.00
30-r-c-1	137,505	17.49	139.909	139.909	0.60	-1.72
30-r-c-2	113.318	97.44	116.112	116.112	0.51	-2.41
30-r-c-0	81.8263	97,44	81.8263	81.8263	0.40	0.00
30-r-e-1	93.0449	73.63	102.041	102.041	0.40	-8.82
30-r-e-2	102.714	57.29	102.714	102.714	0.32	0.00
30-r-r-0	81.6401	97.89	88.2174	88.2174	0.32	-7.46
30-r-r-1	114.088	41,61	120.769	120.769	0.42	-5.53
30-r-r-2	90.3602	78.79	90.7372	90.7372	0.32	-0.42
Avg.	7734.6090	47.1539	113.3318	113.2561	0.4244	-3.28

Trong khi SA hiệu quả với những mô hình có quy mô nhỏ, việc tìm được cách giải tối ưu trường hợp quy mô lớn thường không chắc chắn. SA-ILS hiệu quả đối với cả mô hình số lượng khách hàng nhỏ và lớn. Từ bảng 1, 2, 3, 4, ta thấy: đối với trường hợp 10 khách hàng cả hai thuật toán SA và SA-ILS đều tìm được kết quả tương đồng. Nhưng khi số lượng khách hàng tăng lên 30, thuật toán SA-ILS có nhiều trường hợp lợi nhuận tìm được cao hơn đáng kể (các trường hợp in đậm trong bảng). Việc thay thế các phương pháp truyền thống bằng sử dụng SA-ILS cho thấy hiệu quả rõ rệt trong việc đem lại lời giải tốt cho bài toán triển khai hệ thống giao hàng thông minh kết hợp xe tải và drone.

5. KẾT LUẬN

Trong báo cáo này, tác giả đã giải quyết vấn đề tối đa hóa lợi nhuận cho bài toán người giao hàng lập lịch drone song song với việc đề xuất dùng phương pháp SA-ILS.

Khi sử dụng phép suy nghiệm này để giải những trường hợp lớn hơn, đã đạt được những cách giải có chất lượng nhất quán với thời gian tính toán rất ngắn (tác giả đã thử nghiệm với trường hợp tập 400-khách hàng, thời gian tính toán chỉ mất 30 giây).

Với kết quả này, chúng tôi hi vọng các giải thuật tối ưu này có thể áp dụng để tính toán các khía cạnh khác của bài toán lập lịch drone song song.

TÀI LIỆU THAM KHẢO

- [1] Murray, C.C. and Chu, A.G. "*The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery*". Transportation Research Part C: Emerging Technologies, 54:86–109, (2015).
- [2] Mathew, N., Smith, S.L., & Waslander, S.L. "*Planning paths for package delivery in heterogeneous multirobot teams*". IEEE Transactions on Automation Science and Engineering, 12, 1298–1308, (2015).
- [3] Ha, Q.M., Deville, Y., Pham, Q.D., and Ha, M.H. "*On the min-cost traveling salesman problem with drone*". Transportation Research Part C: Emerging Technologies, 86:597–621, (2018).
- [4] Aline Karak and Khaled Abdelghany: "*The hybrid vehicle-drone routing problem for pick-up and delivery services*". Transportation Research Part C: Emerging Technologies, 102:427 – 449. ISSN 0968-090X, (2019).
- [5] Mbiadou Saleu, R.G., Deroussi, L., Feillet, D., Grangeon, N., and Quilliot, A. "*An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem*". Networks, 72(4):459–474, (2018).
- [6] S. Kirkpatrick, C.D. Gelatto, and M.P. Vecchi, "*Optimization by Simulated Annealing*" Science, vol. 220, pp. 671-680, May 1983
- [7] V. Cerny, "*Thermodynamical Approach to The Traveling Salesman Problem: An Efficient Simulated Algorithm*". Optim. Theory Applicat., vol. 45, no. 1, pp. 41-51, 1985.
- [8] Minh Anh Nguyen, Giang Thi-Huong Dang, Minh Hoàng Hà, Minh-Trien Pham, "The min-cost parallel drone scheduling vehicle routing problem ", *European Journal of Operational Research*, 2021.

Thông tin liên hệ:

Đặng Thị Hương Giang

Điện thoại: 0912506182 - Email: dthgiang@uneti.edu.vn

Khoa Điện tử, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp.