# TỐI ƯU HÓA LỢI NHUẬN CHO BÀI TOÁN ĐỊNH TUYẾN PHƯƠNG TIỆN LẬP LỊCH DRONE SONG SONG

## PROFIT OPTIMIZATION FOR THE PARALLEL DRONE SCHEDULED VEHICLE ROUTING PROBLEM

**Đặng Thị Hương Giang**

*Khoa Điện tử và Kỹ thuật máy tính, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp*

Đến Tòa soạn ngày 24/05/2024, chấp nhận đăng ngày 15/10/2024

**Tóm tắt:** Việc sử dụng drones cho công đoạn giao hàng chặng cuối và việc phối kết hợp chúng cùng với tuyến xe tải với mục đích cải thiện chất lượng dịch vụ và tăng lợi nhuận vận chuyển, đã đặt ra thêm nhiều lớp Bài toán định tuyến phương tiện mới. 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, việc phân giao khách hàng cho xe tải hay drones được tính toán kỹ lưỡng. Chúng tôi đã nghiên cứu mục tiêu lợi nhuận của quy trình giao hàng và đề xuất cải tiến phương pháp SISRs (Slack Induction by String Removals) giải bài toán định tuyến phương tiện lập lịch drone song song nhằm đưa ra lời giải với lợi nhuận tối đa.

**Từ khóa:** SISRs, Drone, quy nạp chùng, lập lịch.

**Abstract:** Using drones for the end-stage delivery and combining them together with trucks in purpose of service quality improvement and transport profit increase, it sets forth more classes of new Vehicle Routing Problems. We consider the Parallel Drone Scheduled Vehicle Routing Problem (PDSVRP) with profit objective. At this problem, allocating customers to trucks or drones will be calculated thoroughly. We researched the profit objective of delivery process and proposed an improvement of the Slack Induction by String Removals method (SISRs) to solve the parallel drone scheduled vehicle routing problem in order to give a solution with maximum profit.

**Keywords:** SISRs, Drone, slack induction, scheduling.

### 1. GIỚI THIỆU

Bài toán PDSTSP (*parallel drone scheduling travelling salesman problem*) được Murray và Chu (2015) [1] giới thiệu, 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 drone giúp giảm thiểu thời gian hoàn tất nhiệm vụ giao phát và quay về kho hàng. Mbiadou Saleu và cộng sự (2018) [2] đã đưa ra phép siêu suy nghiệm gồm hai bước: Bước mã hóa để chuyển cách giải PDSTSP thành tuần tự khách hàng; và bước giải mã để phá vỡ tuần tự khách hàng thành lộ trình cho xe tải và drones. Dell's Amico và cộng sự (2020) [3] đã đề xuất một phép siêu suy nghiệm có cùng ý tưởng với các bước như trong tài liệu Mbiadou Saleu và cộng sự [2]. Tuy nhiên, thay vì lập trình động, phương pháp MILP được sử dụng cho bước giải mã.

Ở bài toán PDSTSP, thông số cần quan tâm nhất là tốc độ phục vụ khách hàng và chất lượng phục vụ của hệ thống vận chuyển. Trong các hoạt động logistic, một yếu tố quan trọng khác cần được xem xét đó là lợi nhuận thu được của hoạt động vận chuyển. Chưa có nghiên cứu nào về lợi nhuận thu được của hệ thống vận chuyển, chỉ có một số tài liệu đã

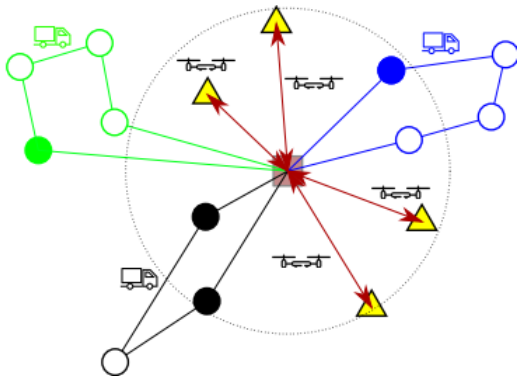
khảo cứu hàm mục tiêu làm giảm thiểu tổng chi phí vận chuyển, như: Mathew và cộng sự [4] đã xem xét Bài toán giao hàng hỗn hợp (HDP - *Heterogeneous Delivery Problem*); Hà và cộng sự [5] đã nghiên cứu Bài toán người chào hàng với drone chi phí tối thiểu (TSP-D - *Traveling Salesman Problem with Drone*) là phiên bản Bài toán FSTSP *(Flying Sidekick Traveling Salesman Problem)* được Murray và Chu [1] giới thiệu. 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 [6] giới thiệu.

Nghiên cứu xem xét khía cạnh lợi nhuận của hệ thống giao phát bằng cách xem xét hàm mục tiêu sao cho tăng tổng lợi nhuận thu được của hệ thống xe tải và drones. Hơn nữa, tác giả còn mở rộng bài toán PDSTSP bằng cách bổ sung thêm hai ràng buộc. Ràng buộc thứ nhất là tổng trọng lượng gói hàng được phân bổ cho mỗi xe tải hoặc drone không vượt quá sức chở của nó. Thứ hai là tổng thời gian làm việc của mỗi phương tiện không vượt quá giá trị cho trước. Như vậy, bài toán nghiên cứu sẽ là Bài toán định tuyến phương tiện lập lịch drone song song PDSVRP (*Parallel Drone Scheduling Vehicle Routing Problem*).

Nghiên cứu có những đóng góp như sau: (1) giới thiệu biến thể mới Bài toán định tuyến phương tiện xe tải và drone giao phát các gói hàng cho khách hàng; (2) lập công thức toán học cho bài toán; (3) Đề xuất phương pháp SISR cải tiến dựa trên phương pháp SISR của Christiaens và Berghe [7]. Phần tiếp theo trình bày cách xác định bài toán và những mô tả chi tiết về phép suy nghiệm. Cuối cùng là kết quả thí nghiệm và các phân tích, đánh giá.

## 2. XÂY DỰNG MÔ HÌNH TOÁN HỌC CỦA BÀI TOÁN

Bài toán PDSVRP tác giả đề xuất gồm đội *n* xe tải và nhóm *m* drones phối hợp 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. Mô hình bài toán PDSVRP được mô tả như trong Hình 1: Đường nét liền biểu thị tuyến xe tải, mũi tên đỏ biểu thị tuyến drones.



**Hình 1. Mô hình bài toán PDSVRP[8]**

### 2.1. Xác định bài toán và những chú giải

Bài toán được biểu thị trên đồ thị có hướng G={N,C}, tập hợp node N={0, 1,…, n}, node kho hàng 0, và tập hợp khách hàng C={0, 1,…, n}. Tập khách hàng xe tải giao hàng $C_T \subseteq C$. Đặt $C_F = C / C_T$ biểu thị khách hàng phù hợp với cả hai phương thức giao phát xe tải và drone. Trọng lượng gói hàng của khách hàng $i \in C$ là $\omega_i$.

Tại kho hàng, đội V gồm *n* xe tải và nhóm D với *m* drones có khả năng đáp ứng các nhiệm vụ giao phát. Khoảng cách xe tải từ node $i \in N$ tới node $j \in N$ với $(j \neq i)$ là $d_{ij}$. Đặt $d_i^{'} = d_{0i}^{'} + d_{i0}^{'}$ là tổng khoảng cách bay của drone phục vụ khách hàng $j \in C_F$. Với cùng cách thức đó, $t_{ij}$ là thời gian xe tải đi trên cạnh (i,j), và t'$_i$ là thời gian drone phục vụ khách hàng $i \in C_F$. Đặt P và P' lần lượt là lợi nhuận thu được của xe tải và drone trên mỗi đơn vị

khoảng cách. Sức chở xe tải chịu ràng buộc bởi Q. Chúng tôi giả định thiết lập như sau:

- Một drone chỉ mang một gói hàng và quay về kho hàng sau mỗi lần giao phát.
- Thời gian nhận, trả các gói hàng và hoán đổi pin là không đáng kể.
- Mỗi drone sẽ phục vụ một khách hàng một lần trong phạm vi khoảng bay giới hạn.
- Mỗi phương tiện (xe tải và drone) giới hạn thời gian làm việc tối đa là T và e.

Mục tiêu của bài toán PDSVRP này là tìm tập hợp tuyến đường cho đội hỗn hợp xe tải và drone có lợi nhuận thu được tối ưu.

### 2.2. Lập công thức toán cho PDSVRP

Lập công thức sẽ sử dụng các biến sau đây:

- $x_{ij}$: bằng 1 nếu node $i \in N$ đến node $i \in N$ $(i \neq j)$ bởi xe tải, còn lại là 0;
- $y_i^k$: bằng 1 nếu node $i \in C_F$ được phục vụ bởi drone $k \in D$, còn lại là 0;
- • $u_i \in \mathbb{R}$: các biến liên tục là trọng lượng xe tải tới node $i \in C$;
- $z_{ij} \in \mathbb{R}$: biểu thị thời gian xe tải từ node $i \in N$ đến node $i \in N$ $(i \neq j)$.

Bài toán PDSVRP có thể được lập công thức như sau:

$$\max \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} . P + \sum_{i \in D} d_i^{'} y_i^k . P^{'} \quad (1)$$

$$\sum_{i \in C} x_{0i} \leq h \quad (2)$$

$$\sum_{\substack{i \in N \\ i \neq j}} x_{ij} + \sum_{k \in D} y_j^k = 1 \ \ \forall j \in N \quad (3)$$

$$\sum_{\substack{i \in N \\ i \neq j}} x_{ij} + \sum_{k \in D} y_j^k = 1 \ \ \forall j \in C_F \quad (4)$$

$$\sum_{\substack{i \in N \\ i \neq j}} x_{ij} = 1 \ \ \forall j \in C_T \quad (5)$$

$$u_i - u_j + Q x_{ij} \leq Q - \omega_j \quad (6)$$
$$\forall i \in C, j \in \{C : j \neq i\}$$

$$\sum_{i \in C_F} y_j^k t_j^{'} \leq e \ \ \forall k \in D \quad (7)$$

$$\sum_{\substack{i \in N \\ l \neq i}} z_{li} + \sum_{\substack{k \in N \\ k \neq i}} t_{ik} x_{ik} = \sum_{\substack{j \in N \\ j \neq i}} z_{ij} \ \ \forall j \in C \quad (8)$$

$$z_{0i} = t_{0i} \, x_{0i} \ \ \forall i \in C \quad (9)$$

$$z_{i0} \leq T \, x_{i0} \ \ \forall i \in C \quad (10)$$

$$x_{ij} \in \{0,1\} \ \ \forall (i, j \in N, i \neq j) \quad (11)$$

$$y_i^k \in \{0,1\} \ \ \forall i \in C_F, k \in D \quad (12)$$

$$0 \leq u_i \leq Q \quad \forall i \in C \quad (13)$$

$$0 \leq z_{ij} \leq T \quad \forall (i, j) \in N, i \neq j \quad (14)$$

Hàm mục tiêu (1) để tìm lợi nhuận tối đa của quy trình. Ràng buộc (2) số lượng xe tải khởi hành từ kho hàng được cho trước. Ràng buộc (3) để đảm bảo tính liên tục của lộ trình xe tải. Ràng buộc (4) mỗi khách hàng được ghé thăm đúng một lần, hoặc là bằng xe tải hoặc là bằng drone. Ràng buộc (5) khách hàng chỉ dùng xe tải. Ràng buộc (6) là ràng buộc sức chở. Tổng thời gian vận hành của mỗi drone sẽ chịu giới hạn theo ràng buộc (7), ràng buộc (8)-(10) đảm bảo thời gian làm việc tối đa của xe tải không vượt quá giá trị T xác định trước. Cuối cùng, các ràng buộc (11)-(14) sẽ cụ thể hóa các miền biến thiên quyết định.

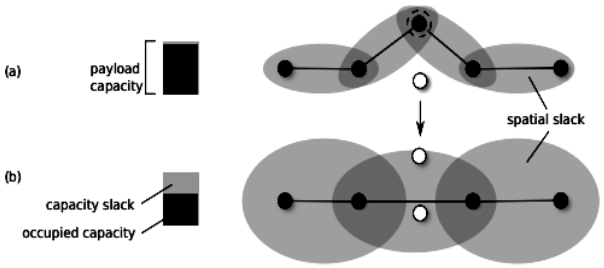
## 3. THUẬT TOÁN SISRs CHO BÀI TOÁN PDSVRP LỢI NHUẬN TỐI ĐA

Phương pháp SISRs cải tiến của tác giả được xây dựng trên cơ sở phương pháp SISRs do Christiaens và Berghe (2020) [7] đã giới thiệu. Phương pháp SISRs dựa trên phép quy nạp chùng và quy trình phá hủy - tái tạo.

### 3.1. Thuật toán SISRs (Slack Induction by String Removals)

Cách giải khởi tạo: phương pháp thực hiện lặp lại hai pha sau: pha Phá hủy ($R^-$) một phần cách giải hiện tại bị phá hủy và sau đó được Tái tạo ($R^+$) bằng cách giải hoàn chỉnh hơn.

Trong Hình 2a, khu vực khả thi của lộ trình tô màu xám, node đen là khách hàng trong lộ trình, node trắng là khách hàng ngoài lộ trình. Về nguyên tắc, khu vực khả thi sẽ mở rộng mỗi khi có khách hàng bị loại khỏi lộ trình và khi đó lộ trình có thể bổ sung khách hàng mới. Hình 2b, quy nạp chùng đạt được bằng việc loại bỏ khách hàng (node đen thành node trắng), phần chùng sức chở và phần chùng không gian của lộ trình được mở rộng. Cho phép có lựa chọn mới (node trắng) vào lộ trình.




**Hình 2. Minh họa phép quy nạp chùng**

### 3.2. Thuật toán SISRs cải tiến

Thuật toán SISRs cải tiến được xây dựng trên cơ sở thuật toán SISRs với một số cải tiến sau: bổ sung quy trình tìm kiếm cục bộ, cùng cơ chế nhiễu loạn để tránh sự hội tụ sớm.

***Tìm kiếm cục bộ***

Sử dụng thuật toán tham lam[9] để xây dựng cách giải từ điểm xuất phát. Trước hết hàm này sẽ chèn các node sử dụng drone theo trật tự giảm dần về mức lợi nhuận. 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.

***Thuật toán 1***. *bestInsertion*

**Input:** VRP problem instance *ins*
**Output:** bestInsertion $s^{best}$

```
1: function bestInsertion (ins)
2:       s₀ ← InsertionConstruction
3:       s* ← s₀, s^bcst ← s₀
4:   repeat
5:     s = Ruin&Recreate(s*)
6:    if profit(s)>profit(s*)-Δ x ln((U(0.1)) then
7:              s* <– s
8:       if profit(s*) > profit(sbest) then
9:                  s^best ← s*
10:           Δ←Δ x €:
11:   until Terminal condition met
```

Ở Thuật toán 1, tạo cách giải đầu tiên thông qua toán tử bestInsertion (dòng 2), khởi tạo cách giải hiện tại *s** và cách giải tốt nhất $s^{best}$ (dòng 3). quy trình lặp nhằm cải thiện cách giải (dòng 4-11). Mỗi lần lặp, cách giải lân cận s được tạo ra từ cách giải hiện tại *s** qua quy trình Ruin&Recreate (dòng 5).

*Cơ chế nhiễu loạn*

Cơ chế nhiễu loạn

Nhằm tìm ra phương án tốt nhất, cơ chế nhiễu loạn được áp dụng thông qua hàm perturbation(s), như được mô tả trong thuật toán 2.

Thuật toán 2. Perturbation

**Input**: *s*

p ←randInt(1,*d*);

*s** ← randomRemove (*s*, *p*);

*s** ←bestInsertion(*s*');

**return** *X'*

Nhằm đa dạng hóa các cách giải, hàm perturbation(s)) như được mô tả trong Thuật

toán 2, trước hết sẽ tách số lượng khách hàng (được tạo ra ngẫu nhiên giữa 1 và d) khỏi cách giải hiện tại s và chạy toán tử bestInsertion(*s*') để cải thiện cách giải này. Sau đó, pha tái lập được xử lý thông qua phép suy nghiệm trên SISRsnew được trình bày trong thuật toán 3.

***Thuật toán 3:***

**Input:** PDSVRP problem instance ins
**1: function SISRsnew(*ins*)**
2: $s_0 \leftarrow$ SolutionConstruction
3: $s^* \leftarrow s_0$
4: **repeat**
5: $s \leftarrow$ Ruin&Recreate($s^*$)
6: **if** profit(s) > profit($s^*$) x (1 + Δ) **then**
7: $s^* \leftarrow$ bestInsertion(s)
8: **if** profit($s^*$) > profit($s^{best}$) **then**
9: $s^{best} \leftarrow s^*$
10: **if** $Inte_{imt}$ iterations without improving $s^{best}$ **then**
11: Perturbate($s^*$)
12: $\Delta \leftarrow \Delta$ x $\epsilon$
13: **until** Terminal condition met
14: **return** $s^{best}$

Ở Thuật toán 3, quá trình khởi tạo từ dòng 1 đến dòng 5. Dòng 6 kiểm tra cách giải nếu thỏa mãn ngưỡng sẽ qua quy trình bestInsertion (dòng 7). Ở đây, Δ là giá trị khởi tạo của $\Delta_{ini}$. Cách giải tìm được sẽ được lưu như là cách giải hiện tại cho những lần lặp tiếp theo. dòng 8-9 tìm cách giải tốt nhất $s^{best}$. Khi số lần lặp không có cải thiện đạt tới $Iter_{imp}$, pha nhiễu loạn Pertubate(s*) được thực hiện (dòng 11-13).

## 4. KẾT QUẢ THÍ NGHIỆM

Kiểm tra tính hiệu quả của thuật toán, thông số được thiết lập trong Bảng 1.

**Bảng 1. Thiết lập thông số cho SISRsnew**

| Thông số | Ký hiệu | Giá trị |
|---|---|---|
| Khoảng cách cho phép | e | 0.6 h |
| Sức chở cho phép (xe tải) | $Q_{truck}$ | 1300 kg |
| Sức chở cho phép (drone) | $Q_{drone}$ | 2.27 kg |
| Giới hạn thời gian tuyến | T | up to 8h |
| Lợi nhuận xe tải trên 1 km | C | 0.03 $/km |
| Lợi nhuận drone trên 1 km | C’ | 1.25 $/km |
| Số lượng khách hàng | n | [10, 750] |

Các thuật toán này được thực hiện trên nền C++ được soạn với GCC 9.3.0. Toàn bộ thử nghiệm được chạy trên AMD Ryzen 3600 @3.8GHz CPU, 16GB RAM, máy tính cá nhân Xubuntu 20.04.

Bảng 2 và 3 trình bày kết quả thu được của thí nghiệm. trong đó: phương pháp SISRs giá trị của các giải pháp tốt nhất được tìm thấy (Cột “z”), thời gian thực hiện tính bằng giây (Cột “time”). “$z_{init}$” là giá trị mục tiêu trung bình và. Tương tự, SISRsnew: giá trị các giải pháp tốt nhất (Cột “$z_{best}$”), thời gian thực hiện trung bình tính bằng giây (Cột “time”), số lượng khách hàng đủ điều kiện sử dụng drone (Cột # DE) trong ví dụ, số lượng khách hàng được drone phục vụ (Cột # D) và số lượng xe tải đang sử dụng (Cột # trucks).

**Bảng 2. Phương pháp SISRs và SISRsnew trường hợp 30 khách hàng**

| Instance | SISRs | | $z_{init}$ | SISRsnew | | # DE | # D | # trucks |
|---|---|---|---|---|---|---|---|---|
| | *z* | *Time* | | $z_{best}$ | *Time* | | | |
| 30-r-0-c | 228.67 | 169.06 | 228.67 | 228.67 | 6.98 | 23 | 13 | 3 |
| 30-r-0-e | 350.05 | 1817.10+ | 351.12 | 350.05 | 5.13 | 14 | 5 | 5 |
| 30-r-0-r | 299.76 | 1536.89+ | 312.07 | 299.76 | 6.45 | 14 | 8 | 4 |
| 30-r-1-c | 158.30 | 2.79 | 158.30 | **163.25** | 7.32 | 26 | 16 | 1 |

| Instance | SISRs | | $z_{init}$ | SISRsnew | | # DE | # D | # trucks |
|---|---|---|---|---|---|---|---|---|
| | *z* | *Time* | | $z_{best}$ | *Time* | | | |
| 30-r-l-e | 314.41 | 651.71 + | 314.32 | 314.41 | 7.56 | 16 | 8 | 4 |
| 30-r-l-r | 264.75 | 2310.15+ | 266.45 | 264.75 | 6.97 | 21 | 11 | 3 |
| 30-r-2-c | 191.57 | 41.60 | 191.57 | 191.57 | 8.78 | 29 | 15 | 2 |
| 30-r-2-e | 306.05 | 956.03+ | 346.07 | 306.05 | 6.52 | 17 | 9 | 4 |
| 30-r-2-r | 271.26 | 789.46+ | 313.22 | 271.26 | 6.89 | 17 | 9 | 4 |
| 30-c-0-c | 122.15 | 38.87 | 122.15 | 122.15 | 10,02 | 24 | 11 | 2 |
| 30-c-0-e | 189.12 | 41.87 | 189.12 | **194.15** | 8.75 | 11 | 3 | 3 |
| 30-c-0-r | 142.72 | 326.63 | 142.72 | 142.72 | 9.02 | 21 | 5 | 3 |
| 30-c-l-c | 79.56 | 4.07 | 79.56 | 79.56 | 9.23 | 26 | 16 | 1 |
| 30-c-l-e | 184.49 | 3600.00 | 184.49 | **190.12** | 7.87 | 20 | 7 | 3 |
| 30-c-l-r | 121.28 | 1.11 | 121.28 | 121.28 | 7.23 | 24 | 13 | 1 |
| 30-c-2-c | 107.44 | 6.17 | 107.44 | 107.44 | 8,14 | 26 | 18 | 2 |
| 30-c-2-e | 184.82 | 3600.00 | 197.36 | 186.82 | 7.45 | 19 | 3 | 3 |
| 30-c-2-r | 128.55 | 31.42 | 128.55 | 128.55 | 9.03 | 25 | 10 | 2 |
| 30-rc-0-c | 121.06 | 15.81 | 121.06 | **125.32** | 8.52 | 26 | 14 | 2 |
| 30-rc-0-e | 231.10 | 3600.00 | 251.10 | 231.10 | 7.89 | 14 | 6 | 3 |
| 30-rc-0-r | 186.73 | 2491.38+ | 202.11 | 186.73 | 8.15 | 22 | 5 | 3 |
| 30-rc-l-c | 125.07 | 7.51 | 125.07 | 125.07 | 8.13 | 27 | 16 | 1 |

**Bảng 3. Phương pháp SISRs và SISRsnew trường hợp 50 khách hàng**

| Instance | SISRs | | $z_{init}$ | SISRsnew | | # DE | # D | # trucks |
|---|---|---|---|---|---|---|---|---|
| | *z* | *Time* | | $z_{best}$ | Time | | | |
| 50-r-0-c | 290.13 | 17.15 | 297.95 | 290.13 | 17.10 | 41 | 15 | 3 |
| 50-r-0-e | - | 2163.21+ | 435.21 | **436.23** | 18.23 | 22 | 8 | 5 |
| 50-r-0-r | - | 2418.11 + | 353.81 | **354.12** | 18.34 | 29 | 14 | 4 |
| 50-r-l-c | 287.21 | 3600.00 | 280.84 | 287.21 | 17.12 | 41 | 15 | 4 |
| S0-r-l-e | 376.97 | 3600.00 | 370.02 | 376.97 | 16.78 | 24 | 10 | 3 |
| 50-r-l-r | - | 2572.35+ | 355.04 | **358.11** | 19.78 | 33 | 13 | 4 |
| 50-I--2-c | 282.09 | 3600.00 | 278.73 | 282.09 | 15.12 | 32 | 15 | 4 |
| 50-r-2-e | - | 2387.22+ | 500.12 | **512.36** | 17.45 | 25 | 4 | 6 |
| 50-r-2-r | - | 2633.51 + | 380.61 | **381.49** | 19.67 | 29 | 12 | 6 |
| 50-c-0-c | 189.04 | 3600.00 | 151.73 | 189.04 | 18.58 | 44 | 15 | 2 |
| 50-c-0-e | 283.26 | 3600.00 | 255.69 | **267.14** | 16.81 | 30 | 10 | 4 |
| 50-c-0-r | 162.17 | 3563.53+ | 153.15 | 162.17 | 16.17 | 36 | 17 | 2 |
| 50-c-l-c | 181.76 | 859.09 | 116.83 | **190.16** | 16.54 | 42 | 18 | 2 |
| 50-c-l-e | 183.57 | 3600.00 | 177.55 | **185.61** | 14.58 | 41 | 6 | 3 |
| 50-c-l-r | 199.24 | 1512.49+ | 196.76 | 199.24 | 18.91 | 41 | 17 | 3 |
| 50-c-2-c | 177.16 | 73.52 | 154.70 | 177.16 | 20.26 | 40 | 15 | 2 |

| Instance | SISRs | | $z_{init}$ | SISRsnew | | # DE | # D | # trucks |
|---|---|---|---|---|---|---|---|---|
| | *z* | *Time* | | $z_{best}$ | Time | | | |
| 50-c-2-e | 294.25 | 2684.94+ | 292.78 | 294.25 | 17.52 | 23 | 13 | 4 |
| 50-c-2-r | 186.60 | 2288.27+ | 186.17 | **197.51** | 18.73 | 34 | 15 | 3 |
| 50-rc-0-c | 246.80 | 53.87 | 236.97 | 246.80 | 20.90 | 43 | 16 | 2 |
| 50-rc-0-e | - | 2679.98+ | 345.61 | **348.56** | 15.63 | 32 | 5 | 4 |
| 50-rc-2-e | - | 3032.47+ | 487.22 | **489.15** | 15.71 | 26 | 8 | 7 |
| 50-rc-2-r | 277.20 | 3600.00 | 272.75 | 277.20 | 16.34 | 29 | 16 | 4 |

Từ Bảng 2, chúng tôi nhận thấy SISRsnew có thể tìm ra giải pháp tối ưu cho 4 trên 22 trường hợp với 30 khách hàng trong vòng chưa đầy 3 phút. Khi số lượng khách hàng tăng lên 50, Bảng 3 cho thấy SISR trung bình mất nhiều thời gian hơn. Trong các trường hợp không thành công, SISRs buộc phải chấm dứt do tình trạng hết bộ nhớ trên 11 instance (được đánh dấu "+" trong cột *time*) và không đưa ra được giải pháp khả thi nào trong 7 trường hợp (được đánh dấu "-").

SISRsnew có hiệu suất tốt vì nó có thể tìm thấy tất cả các giải pháp tối ưu được SISRs tìm thấy. Ngoài ra, nó còn tìm ra giải pháp tốt hơn SISRs thực hiện (in đậm trong Cột $z_{best}$).. Trong cả hai bảng, thời gian chạy trung bình nhanh, dưới 21 giây. Tỷ lệ khách hàng được drone phục vụ trên tổng số khách hàng đủ điều kiện sử dụng drone thay đổi trong khoảng rộng từ 12,50% đến 82,35%. Điều này một lần nữa khẳng định tính đa dạng của các trường hợp được tạo ra và cho thấy vấn đề phân bổ khách hàng cho xe tải hoặc drone sẽ là bài toán khó giải quyết.

## 5. KẾT LUẬN

Trong nghiên cứu, chúng tôi đã tìm cách để tối ưu lợi nhuận vận chuyển của đội hỗn hợp phương tiện xe tải và drone trong hệ thống giao phát. Chúng tôi đã đề xuất SISRs cải tiến để tránh sự hội tụ sớm tại các điểm tối ưu cục bộ bằng cách bổ sung quy trình tìm kiếm cục bộ, cùng cơ chế nhiễu loạn.

Nhìn chung, kết quả tìm được trong nhiều thí nghiệm lên tập trường hợp đặt ra lựa chọn ngẫu nhiên mới đã xác nhận được tính hiệu quả và nhất quán của thuật toán.

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] Mbiadou Saleu, R.; Deroussi, L.; Feillet, D.; Grangeon, N.; Quilliot, A.: An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. 72, 459–474, (Networks 2018),.

[3] Mauro Dell'Amico, Roberto Montemanni, and Stefano Novellani.: Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. Annals of Operations Research, pages 1–16, (2020).

[4] 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).

[5] Quang Minh Ha, Yves Deville, Quang Dung Pham, and Minh Hoàng Hà.: On the min-cost traveling salesman problem with drone. Transportation Research Part C: Emerging Technologies, 86:597–621, (2018).

[6] 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).

[7] Jan Christiaens and Greet Vanden Berghe: Slack induction by string removals for vehicle routing problems. Transportation Science, (2020).

[8] Minh Anh Nguyễn, Giang Thị-Hương Đặng, Minh Hoàng Hà*, Minh-Triển Phạm "The min-cost parallel drone scheduling vehicle routing problem", European Journal of Operational Research.

[9] Dell'Amico, M., Montemanni, R., and Novellani, S. " Matheuristic algorithms for the parallel drone scheduling traveling salesman problem". Annals of Operations Research, pages 1–16, (2020).

*Thông tin liên hệ:* **Đặng Thị Hương Giang**

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

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