2020.09.24

[LAB SEMINAR]

The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery

Chase C. Murray*, Amanda G. Chu

OPTLAB HUFS
한국외국어대학교 최적화연구실
석사과정 강문정

Introduction

- 소형 택배에 활용되는 드론의 예시

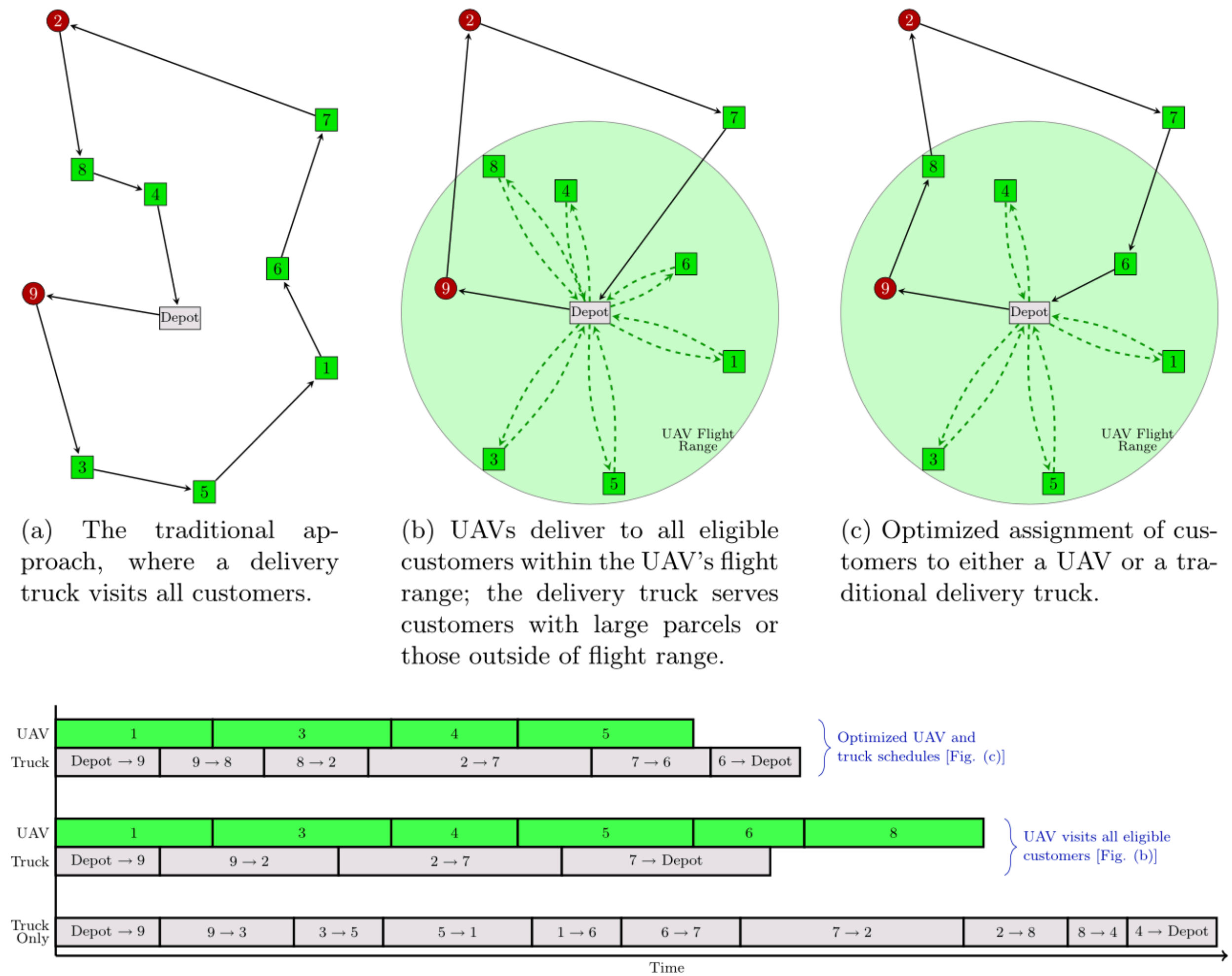
(a) Amazon's *Prime Air* UAV (source: amazon.com)

(b) DHL's *Parcelcopter* (source: dhl.com)

Fig. 1. UAVs under evaluation for small parcel delivery.

- 드론을 이용한 배송은 라스트 마일 (Last Mile, 상품이 최종 목적지까지 배송되기 위한 과정)에 활용 가능성↑
- 그러나 드론에 대한 규제 및 기술적 문제가 존재함 : 허가 받은 지역에서의 비행, 드론 배터리의 개선 및 안전성, 드론 GPS 연결 개선, 드론의 장애물 탐지 및 회피 수행 능력 등
- 기술적 문제를 해결하기 위한 연구는 많이 이루어지고 있지만 운영상의 문제를 해결하는 연구는 부족함

Introduction : Traditional delivery



(a) The traditional approach, where a delivery truck visits all customers.

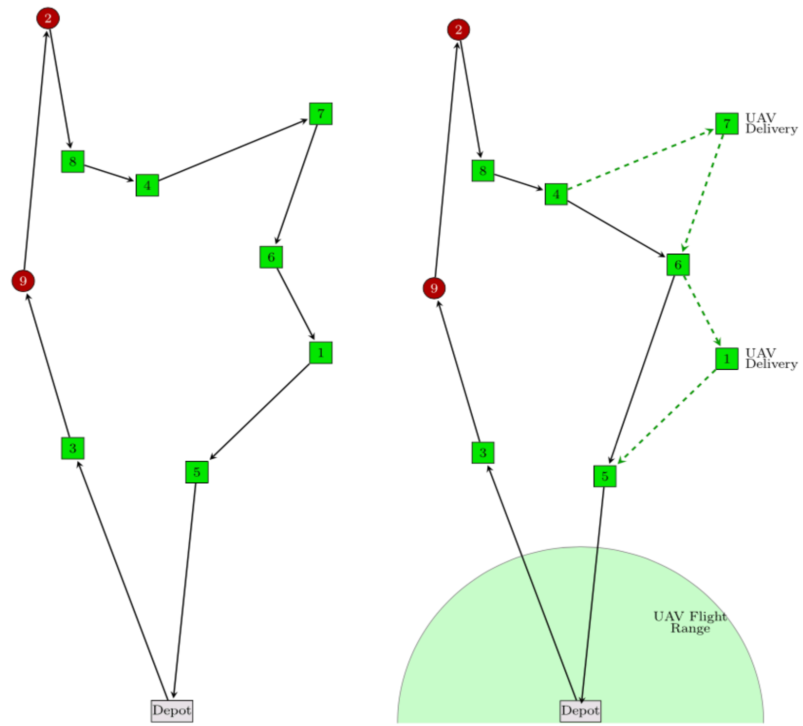
(b) UAVs deliver to all eligible customers within the UAV's flight range; the delivery truck serves customers with large parcels or those outside of flight range.

(c) Optimized assignment of customers to either a UAV or a traditional delivery truck.

(d) A comparison of delivery schedules for the three systems depicted above.

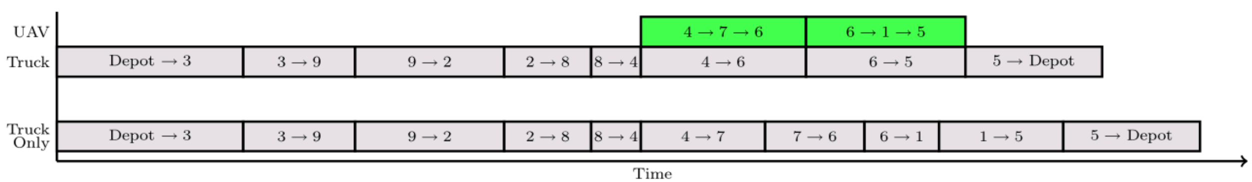
Fig. 2. Customer deliveries are made by either a traditional delivery truck or via UAV. Customers 2 and 9 (circular nodes) are ineligible to be served via UAV (e.g., due to parcel weight restrictions).

Introduction : FSTSP (Flying Sidekick Traveling Salesman Problem)



(a) An optimal truck delivery sequence, without the aid of a UAV.

(b) The UAV is launched from a delivery truck, delivering parcels to two eligible customers.



(c) A comparison of delivery schedules for the two scenarios depicted above.

Fig. 3. In cases where the distribution center (depot) is not conducive to direct UAV deliveries, a truck/UAV tandem may reduce delivery times. Customers 2 and 9, depicted by circular nodes above, are ineligible to be served via UAV.

FSTSP Notation

- **Parameters**

C : Set of all customers. $C = \{1,2, \dots, c\}$
C' : The subset of customers that may be serviced by the UAV. $C' \subseteq C$
N : Set of all nodes in the network. $N = \{0,1, \dots, c+1\}$
N_0 : $N_0 = \{0,1, \dots, c\}$
N_+ : $N_+ = \{1,2, \dots, c+1\}$
τ_{ij} : The time required for the truck to travel from node $i \in N_0$ to node $j \in N_+$. $\tau_{0,c+1} \equiv 0$.
τ'_{ij} : The analogous travel time for the UAV.
S_L : The time required by the truck driver to prepare the UAV for *launch*.
S_R : The time required by the truck driver to recover the UAV upon *rendezvous*.
e : The flight endurance of the UAV.
P : The set of tuples of the form $\langle i,j,k \rangle$.
→ ⟨the launch point i, the delivery point j, the rendezvous point k⟩ $i \in N_0$, $j \in \{C' : j \neq i\}$,
$k \in \{N_+ : k \neq j, k \neq i, \tau'_{ij} + \tau'_{jk} \leq e\}$
M : A sufficiently large number.

- **Decision variables**

x_{ij} : 1 if the truck travels from node $i \in N_0$ to $j \in N_+$ where $j \neq i$, 0 otherwise.
y_{ijk} : 1 if the UAV is launched from node $i \in N_0$, travels to node $j \in C$, and returns to a truck or the ending depot at node $k \in \{N_+ : \langle i,j,k \rangle \in P\}$, 0 otherwise.
t_j : The time at which the truck arrives at node $j \in N_+$. $t_j \geq 0$, $t_0 = 0$.
t'_j : The time at which the UAV arrives at node $j \in N_+$. $t'_j \geq 0$, $t'_0 = 0$.
p_{ij} : 1 if customer $i \in C$ is visited at some time before customer $j \in \{C : j \neq i\}$ in the truck's path, 0 otherwise. $p_{01} = 1 \ \forall j \in C$.
u_i. : Specifies the position of node $i \in N_+$ in the truck's route. $1 \leq u_i \leq c+2$.

FSTSP Mathematical Formulation

$$\text{Min} \quad t_{c+1} \quad (= \min\{\max\{t_{c+1}, t'_{c+1}\}\} \because (14), (15))$$

(1) Minimize the latest time at which either the truck or the UAV return to the depot. (1)

$$\text{s.t.} \quad \sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} + \sum_{\substack{i \in N_0 \\ i \neq j}} \sum_{\substack{k \in N_+ \\ \langle i,j,k \rangle \in P}} y_{ijk} = 1 \quad \forall j \in C \qquad (2)$$

Each customer to be visited exactly once. (2)

$$\sum_{j \in N_+} x_{0j} = 1 \qquad (3)$$

The truck departs from the depot exactly once. (3)

$$\sum_{i \in N_0} x_{i,c+1} = 1 \qquad (4)$$

The truck to return to the depot exactly once. (4)

$$u_i - u_j + 1 \leqslant (c+2)(1 - x_{ij}) \quad \forall i \in C, j \in \{N_+ : j \neq i\} \qquad (5)$$

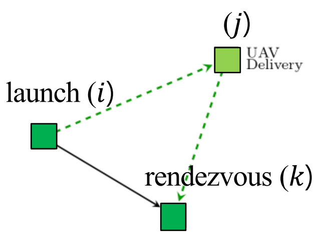
Subtour elimination constraints for the truck. (5)

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} = \sum_{\substack{k \in N_+ \\ k \neq j}} x_{jk} \quad \forall j \in C \qquad (6)$$

A truck visiting node j must also depart from j. (6)

$$\sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i,j,k \rangle \in P}} y_{ijk} \leqslant 1 \quad \forall i \in N_0 \qquad (7)$$

The UAV may launch from any particular node, including the depot, at most once. (7)



$$\sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle i,j,k \rangle \in P}} y_{ijk} \leqslant 1 \quad \forall k \in N_+ \qquad (8)$$

The UAV may rendezvous at any particular node, (including customers and the ending depot) at most once. (8)

$$2y_{ijk} \leqslant \sum_{\substack{h \in N_0 \\ h \neq i}} x_{hi} + \sum_{\substack{l \in C \\ l \neq k}} x_{lk} \quad \forall i \in C, j \in \{C : j \neq i\}, k \in \{N_+ : \langle i,j,k \rangle \in P\} \qquad (9)$$

If the UAV launches from *customer* i and is collected by the truck at node k, then the truck must be assigned to both nodes i and k. (9)

$$y_{0jk} \leqslant \sum_{\substack{h \in N_0 \\ h \neq k}} x_{hk} \quad \forall j \in C, k \in \{N_+ : \langle 0,j,k \rangle \in P\} \qquad (10)$$

If the UAV launches from *the starting depot* 0 and is collected at node k, then the truck must be assigned to node k. (10)

$$u_k - u_i \geqslant 1 - (c+2)\left(1 - \sum_{\substack{j \in C \\ \langle i,j,k \rangle \in P}} y_{ijk}\right) \quad \forall i \in C, k \in \{N_+ : k \neq i\} \qquad (11)$$

If the UAV launches from customer i and is collected at node k, then the truck must visit i before k. (11)

FSTSP Mathematical Formulation

$$t'_i \geqslant t_i - M\left(1 - \sum_{\substack{j\in C\\ j\neq i}} \sum_{\substack{k\in N_+\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) \quad \forall i \in C \tag{12}$$

$$t'_i \leqslant t_i + M\left(1 - \sum_{\substack{j\in C\\ j\neq i}} \sum_{\substack{k\in N_+\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) \quad \forall i \in C \tag{13}$$

The truck and the UAV are time-coordinated when the UAV is *launched* from *customer* node i. (12,13)

$$t'_k \geqslant t_k - M\left(1 - \sum_{\substack{i\in N_0\\ i\neq k}} \sum_{\substack{j\in C\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) \quad \forall k \in N_+ \tag{14}$$

$$t'_k \leqslant t_k + M\left(1 - \sum_{\substack{i\in N_0\\ i\neq k}} \sum_{\substack{j\in C\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) \quad \forall k \in N_+ \tag{15}$$

The truck and the UAV are time-coordinated when the UAV *returns* to the truck at node k. (14,15)

$$t_k \geqslant t_h + \tau_{hk} + s_L\left(\sum_{\substack{l\in C\\ l\neq k}} \sum_{\substack{m\in N_+\\ \langle k,l,m\rangle\in P}} y_{klm}\right) + s_R\left(\sum_{\substack{i\in N_0\\ i\neq k}} \sum_{\substack{j\in C\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) - M(1 - x_{hk})$$

$$\forall h \in N_0, k \in \{N_+ : k \neq h\} \tag{16}$$

Suppose the truck travels from $h \in N_0$ to $k \in N_+$.
If the UAV were launched from k, S_L must be incorporated.
If, prior to launching from k, S_R must be also incorporated. (16)

$$t'_j \geqslant t'_i + \tau'_{ij} - M\left(1 - \sum_{\substack{k\in N_+\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) \quad \forall j \in C', i \in \{N_0 : i \neq j\} \tag{17}$$

If the UAV launches from node i, then its arrival time at some node j must incorporate the travel time from i to j. (17)

$$t'_k \geqslant t'_j + \tau'_{jk} + s_R - M\left(1 - \sum_{\substack{i\in N_0\\ \langle i,j,k\rangle\in P}} y_{ijk}\right) \quad \forall j \in C', k \in \{N_+ : k \neq j\} \tag{18}$$

If the UAV is retrieved by the truck at node k, then the arrival time at k must incorporate the travel time from j to k plus the recovery service time at k, S_R. (18)

$$t'_k - (t'_j - \tau'_{ij}) \leqslant e + M(1 - y_{ijk}) \quad \forall k \in N_+, j \in \{C : j \neq k\}, i \in \{N_0 : \langle i,j,k\rangle \in P\} \tag{19}$$

The UAV's flight endurance. (19)

$$u_i - u_j \geqslant 1 - (c+2)p_{ij} \quad \forall i \in C, j \in \{C : j \neq i\} \tag{20}$$

$$u_i - u_j \leqslant -1 + (c+2)(1 - p_{ij}) \quad \forall i \in C, j \in \{C : j \neq i\} \tag{21}$$

$$p_{ij} + p_{ji} = 1 \quad \forall i \in C, j \in \{C : j \neq i\} \tag{22}$$

Determine the proper values of p_{ij}. (20,21,22)

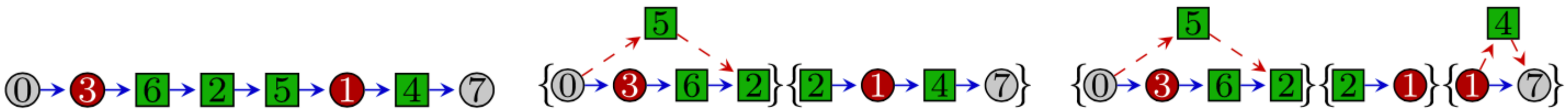
$$t'_l \geqslant t'_k - M\left(3 - \sum_{\substack{j\in C\\ \langle i,j,k\rangle\in P\\ j\neq l}} y_{ijk} - \sum_{\substack{m\in C\\ m\neq i\\ m\neq k\\ m\neq l}} \sum_{\substack{n\in N_+\\ \langle l,m,n\rangle\in P\\ n\neq i\\ n\neq k}} y_{lmn} - p_{il}\right)$$

$$\forall i \in N_0, k \in \{N_+ : k \neq i\}, l \in \{C : l \neq i, l \neq k\} \tag{23}$$

Suppose the UAV launches from i and returns to k. Further, suppose that the UAV later launches from l ($p_{il} = 1$). Constraint (23) will prevent the launch time from l, t'_l, from preceding the return time to k, t'_k. (23)

FSTSP Heuristic : A route and re-assign heuristic

- MILP 수리 모형이 10명의 customer 문제도 풀기 어렵기 때문에 휴리스틱 솔루션을 제안함



(a) The delivery truck is initially assigned to visit all customers. truckSubRoutes = truckRoute = $\{0, 3, 6, 2, 5, 1, 4, 7\}$. Cprime = $C' = \{6, 2, 5, 4\}$.

(b) UAV-eligible customer 5 has been removed from the truck's route and is assigned to the UAV. The UAV is launched from the depot (node 0) and is recovered by the truck at node 2. There are now two truck subroutes, $\{0, 3, 6, 2\}$ and $\{2, 1, 4, 7\}$; Cprime = $\{6, 4\}$.

(c) Customer 4 is assigned to the UAV. There are now three truck subroutes, and Cprime = $\{6\}$.

Fig. 4. A notional example to demonstrate the FSTSP heuristic. Nodes 0 and 7 represent the depot. Customers in C' (UAV-eligible) are shown in boxed nodes, while the remaining circular nodes represent UAV-ineligible customers.

1. TSP를 풀고 truckRoute와 truckSubRoute를 초기화한다. 이 때, 각 노드에서 트럭이 도착하는 시간(t)을 구한다.

2. Cprime(드론으로 배송 가능한 노드 집합)에 대해 각 노드에서의 도착 시간을 이용하여 드론으로 배송할 경우의 savings를 구한다.

   → Fig.4b.에서 5번을 제거할 경우 savings = min{$(\tau_{25} + \tau_{51} - \tau_{21})$, $((\tau_{03} + \tau_{36} + \tau_{62}) - (0 + \tau'_{05} + \tau'_{52} + S_R))$}

   - $(\tau_{25} + \tau_{51} - \tau_{21})$: 트럭 경로가 줄어든만큼 시간 절약
   - $(\tau_{03} + \tau_{36} + \tau_{62})$: 2번까지 트럭으로 걸리는 시간
   - $(0 + \tau'_{05} + \tau'_{52} + S_R)$: 0번에서5번을 지나 2번까지 드론으로 걸리는 시간 (0: 드론이 launch 하는 노드에서의 t)

3. 어떤 고객을 truckSubRoute 중 하나에 추가할 때 트럭으로 서비스하면 CalcCostTruck 드론으로 하면 CalcCostUAV.

   → 4번 고객을 {0,3,6,2}에 추가할 경우 CalcCostTruck, {2,1,4,7}에 추가할 경우 CalcCostUAV을 구하여 업데이트 한다.

4. 서비스를 트럭으로 하는지 드론으로 하는지에 따라 truckRoute, truckSubRoute, t를 업데이트한다.

5. 고객 할당이 개선되지 않을 경우(더 이상 배송 시간을 줄일 수 없을 경우) 알고리즘을 종료한다.

The Parallel drone scheduling TSP (PDSTSP)

- **Parameters**

  V : Set of identical UAVs

  C'' : UAV customers that are also within the UAV's range from the depot . $C'' \subseteq C'$
  (customer $i \in C'$ is in set C'' if $\tau'_{0,i} + \tau'_{i,c+1} \leq e$)

- **Decision variables**

  $\hat{x}_{i,j}$: 1 if the truck travels from node $i \in N_0$ to node $j \in \{N_+ : j \neq i\}$, 0 otherwise.

  $\hat{y}_{i,v}$: 1 if customer $i \in C''$ is served by UAV $v \in V$, 0 otherwise.

  $\hat{u}_i$: Specifies the position of node $i \in N_+$ in the truck's route. $1 \leq \hat{u}_i \leq c + 2$.

$\text{Min} \quad z$	(33)	Minimize the latest return time to the depot for both the UAV and the truck. (33)
$\text{s.t.} \quad z \geqslant \sum_{i \in N_0} \sum_{\substack{j \in N_+ \\ j \neq i}} \tau_{i,j} \hat{x}_{i,j}$	(34)	Provide lower bounds on z, based on truck. (34)
$z \geqslant \sum_{i \in C''} (\tau'_{0,i} + \tau'_{i,c+1}) \hat{y}_{i,v} \quad \forall v \in V$	(35)	Provide lower bounds on z, based on UAV assignments. (35)
$\sum_{\substack{i \in N_0 \\ i \neq j}} \hat{x}_{i,j} + \sum_{\substack{v \in V \\ j \in C''}} \hat{y}_{j,v} = 1 \quad \forall j \in C$	(36)	Each customer is visited exactly once, either by the truck or a UAV. (36)
$\sum_{j \in N_+} \hat{x}_{0,j} = 1$	(37)	The truck to leave the depot exactly once. (37)
$\sum_{i \in N_0} \hat{x}_{i,c+1} = 1$	(38)	The truck returns to the depot. (38)
$\sum_{\substack{i \in N_0 \\ i \neq j}} \hat{x}_{i,j} = \sum_{\substack{k \in N_+ \\ k \neq j}} \hat{x}_{j,k} \quad \forall j \in C$	(39)	Specify that a truck entering a customer node must also leave that node. (39)
$\hat{u}_i - \hat{u}_j + 1 \leqslant (c+2)(1 - \hat{x}_{i,j}) \quad \forall i \in C, \; j \in \{N_+ : j \neq i\}$	(40)	A standard subtour elimination constraint. (40)

PDSTSP Heuristic

- PDSTSP는 TSP와 PMS(parallel identical machine scheduling) problem with a minimal makespan objective 가 융합된 문제

- PDSTSP에서의 makespan은 모든 고객을 방문하고 depot까지 돌아오는 시간을 뜻함

- TSP와 PMS 모두 NP-hard 이기 때문에 휴리스틱을 제안함

1. Depot 근처에 있는 고객은 모두 드론으로 배송하고 나머지 고객은 트럭으로 배송하는 것으로 초기화한다.

2. TSP를 풀어 모든 고객을 방문하는 트럭의 경로 및 makespan을 구하고, PMS를 풀어 드론과 고객을 할당하고 makespan을 구한다.

3. 드론 배송의 makespan이 트럭 배송의 makespan보다 클 경우, 드론으로 배송하는 고객은 트럭으로 배송할 후보가 된다.

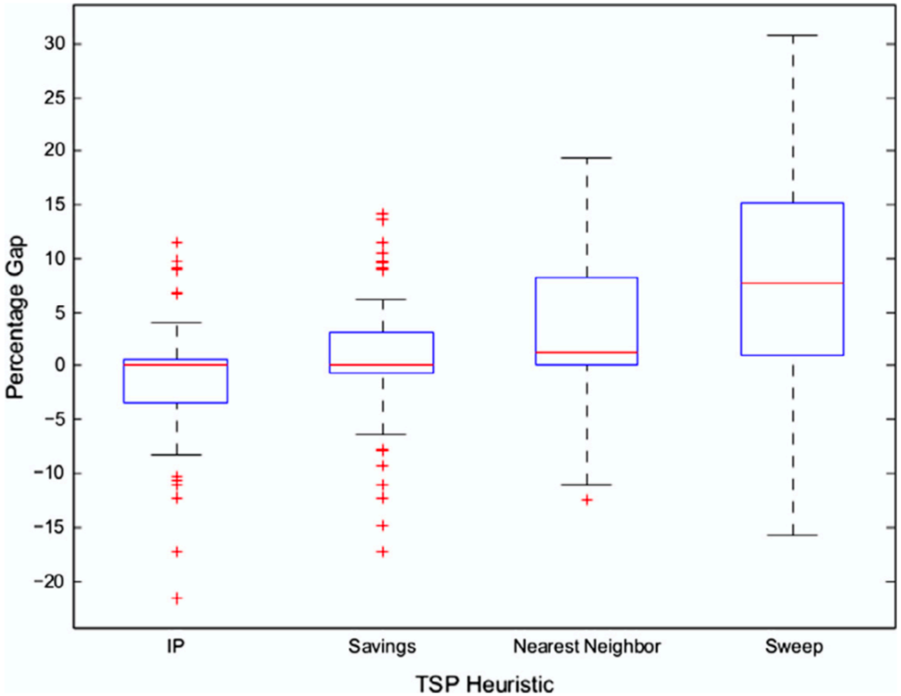
4. Net savings가 가장 큰 값(makespan을 가장 많이 줄일 수 있는 값)이 되도록 재할당하며, 더 이상 변화가 없는 경우 swap()을 실행한다.

5. Swap() 함수에서는 드론으로 배송하는 고객과 트럭으로 배송하는 고객의 모든 쌍을 바꾸면서 makespan을 개선할 수 있는지 확인한다.

6. 고객 재할당을 반복하며 Makespan을 개선하는 솔루션을 얻을 수 없는 경우 알고리즘을 종료한다.

Empirical results : Analysis of the FSTSP Heuristic framework

FSTSP에서 다양한 TSP를 적용했을 때의 효과를 비교



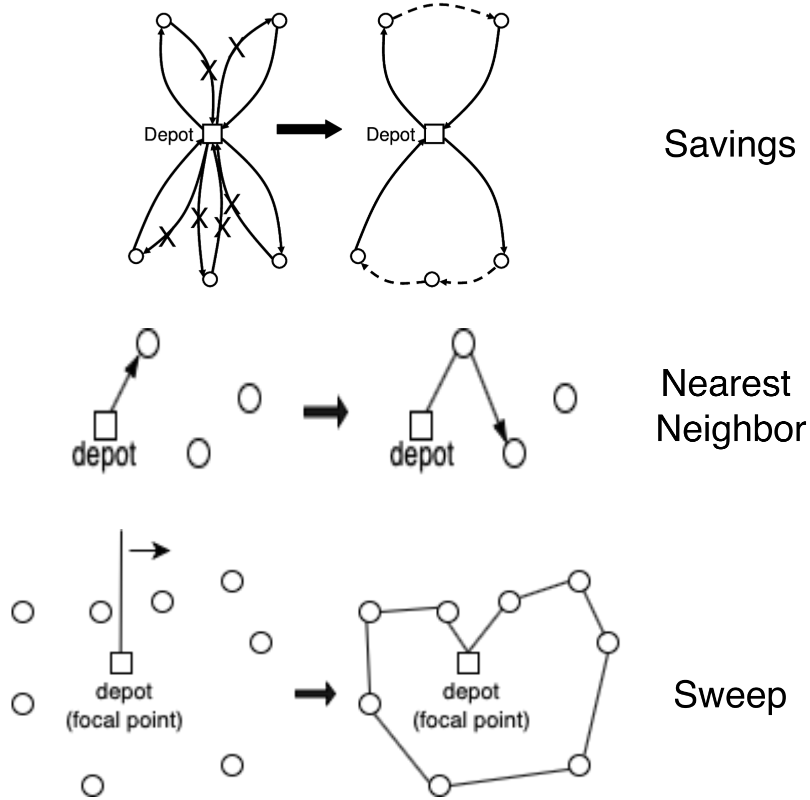
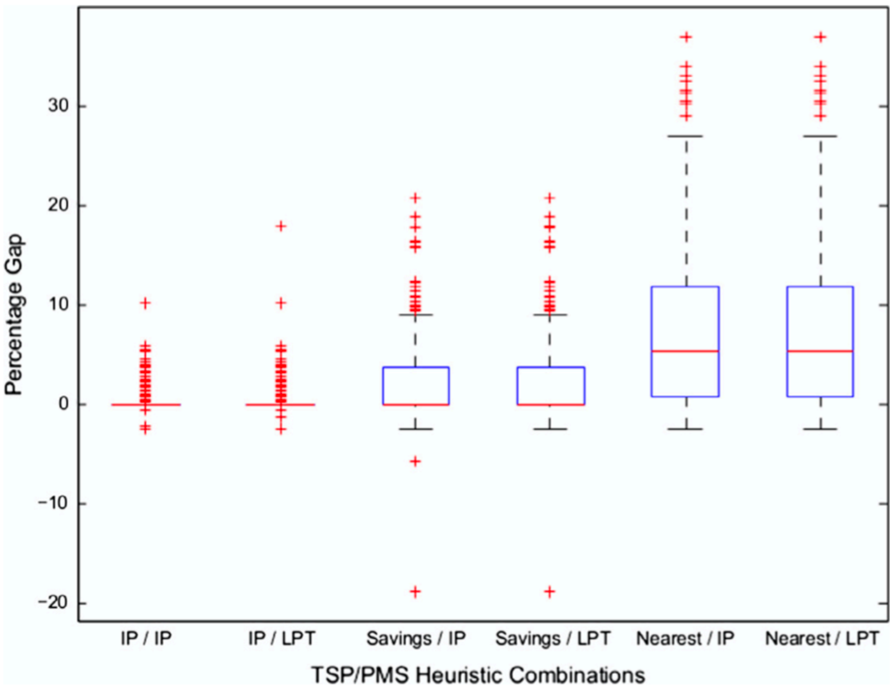


Table 1
A summary of the FSTSP heuristic's performance with various TSP solution approaches.

TSP solution approach	Gap (%)			Runtime (s)		
	Avg	Min	Max	Avg	Min	Max
IP	−1.16	−21.73	11.59	5.026	0.380	31.540
Savings	0.33	−17.33	14.07	0.004	0.001	0.006
Nearest neighbor	2.91	−12.45	19.26	0.004	0.001	0.006
Sweep	8.33	−15.70	30.93	0.004	0.001	0.006
FSTSP formulation				1800.000	1800.000	1800.000

Empirical results : Analysis of the PDSTSP Heuristic framework

10-20명의 고객에 대한 PDSTSP 솔루션 비교



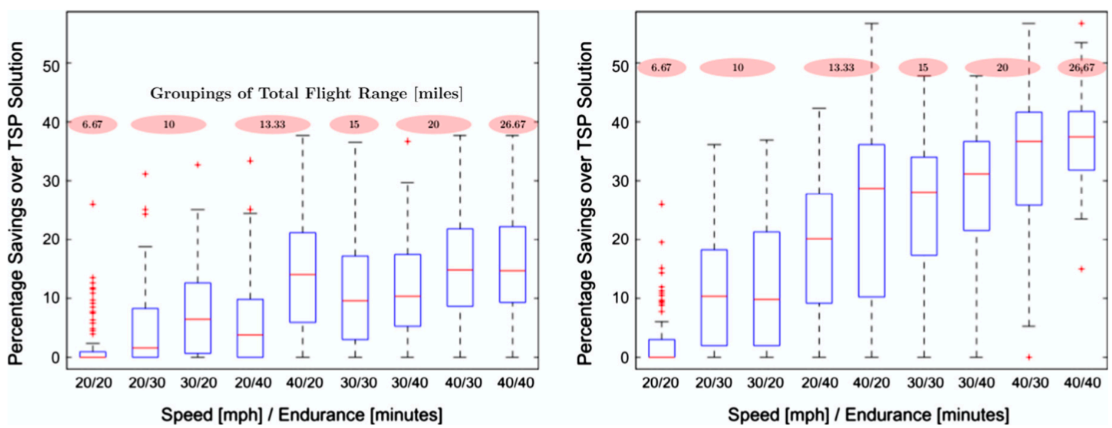
- PDSTSP는 TSP와 PMS가 결합되어 있음
  - TSP : IP, Savings, Nearest
  - PMS : IP, LPT(longest processing time first heuristic)
- UAV 근처에 있는 고객 비율(20,40,60,80%)을 다르게 하여 문제를 풀었고 트럭은 1대, 드론은 최대 3대까지 사용함
- Gap은 Gurobi와 제시한 휴리스틱의 솔루션 차이 (음수 값은 휴리스틱 솔루션이 더 좋다는 것을 의미함)
- 솔루션 품질과 풀이 시간을 모두 고려한다면 Savings/LPT가 가장 좋은 결과를 나타냄

Table 2
Details of heuristic performance on small PDSTSP instances.

Solution approach		10 Customers				20 Customers			
		Gap (%)		Runtime (s)		Gap (%)		Runtime (s)	
TSP	PMS	Avg	Max	Avg	Max	Avg	Max	Avg	Max
IP	IP	0.12	10.13	2.4856	29.97	0.22	5.53	495.272	21510.61
IP	LPT	0.12	10.13	2.3093	28.85	0.31	18.00	498.057	21521.31
Savings	IP	1.57	20.68	0.2373	8.26	3.79	18.83	3.721	80.68
Savings	LPT	1.58	20.68	0.0003	0.01	3.90	18.83	0.008	0.07
Nearest	IP	5.46	37.17	0.2335	8.26	10.63	34.21	3.137	65.83
Nearest	LPT	5.46	37.17	0.0003	0.01	10.68	34.21	0.002	0.02
PDSTSP formulation				0.3194	2.02			77.775	180.00

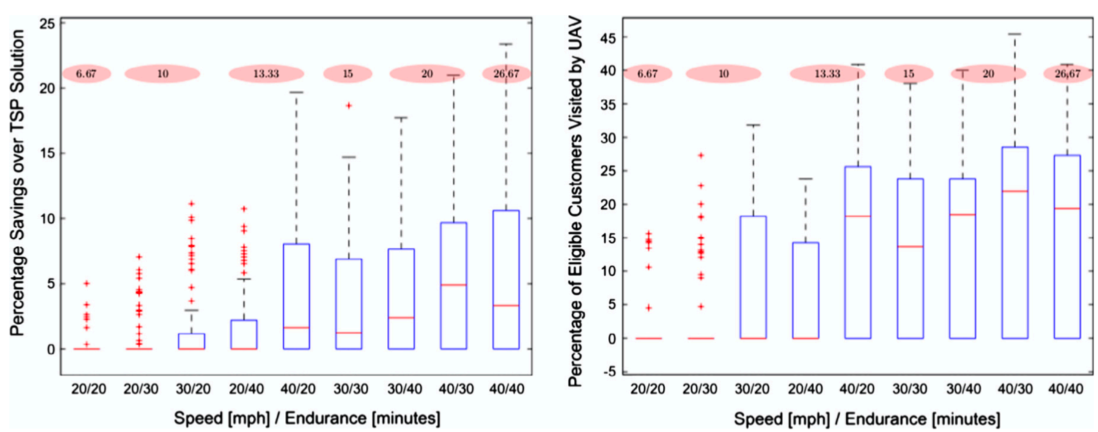
Empirical results : Speed vs Endurance

- 드론의 속도와 배터리 지속성은 중요한 요소
- Savings/LPT로 실험한 결과 속도가 더 중요한 요소임을 확인함



(a) One available UAV (b) Three available UAVs

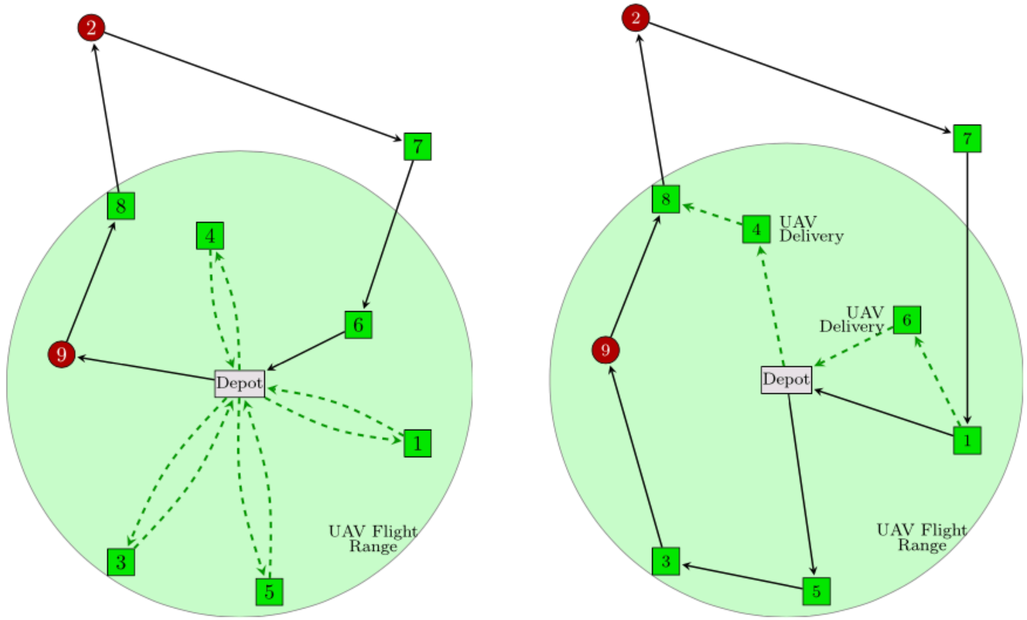
Fig. 7. Impacts of speed and endurance for the PDSTSP. Numbers above each speed/endurance pair represent groupings of equivalent total flight distances.



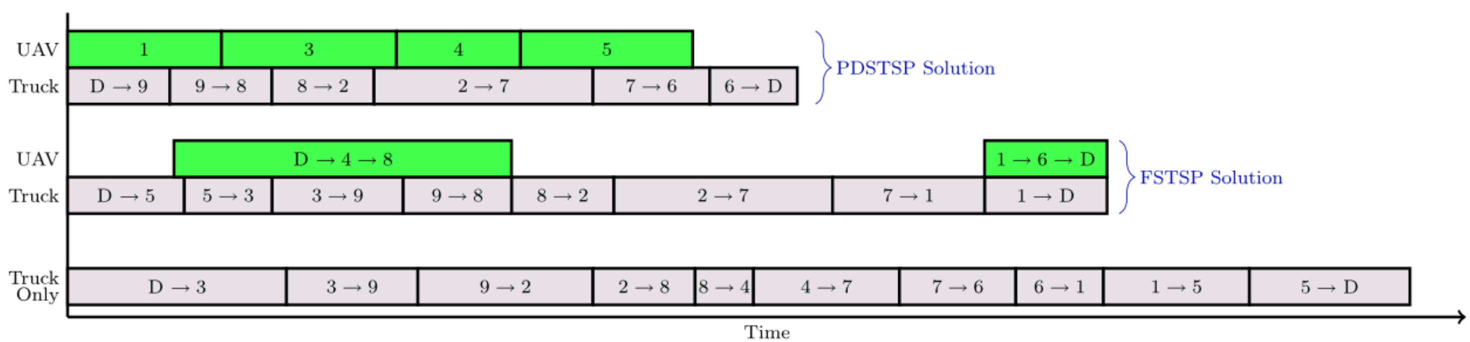
(a) Improvement over TSP solution (b) Percentage of eligible customers visited by UAV

Fig. 8. Impacts of speed and endurance for the FSTSP. Numbers above each speed/endurance pair represent groupings of equivalent total flight distances.

Empirical results : Contrasting the FSTSP and PDSTSP



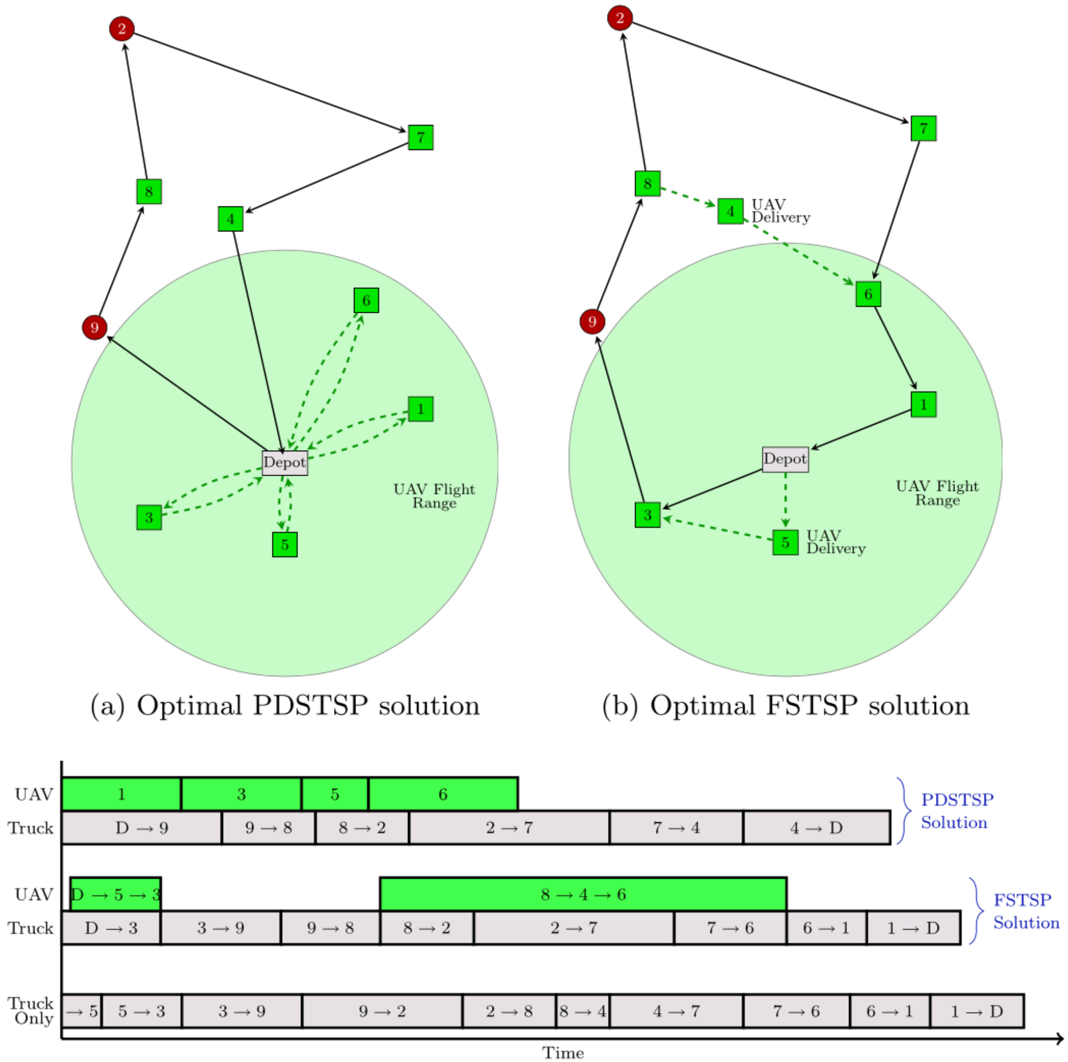
(a) Optimal PDSTSP solution (b) Optimal FSTSP solution



(c) A comparison of delivery schedules for the two systems depicted above. The "Truck Only" schedule represents an optimal TSP solution for a single truck serving all customers.

Fig. 9. Six customers are within the UAV's flight radius from a centrally-located depot. Customers 2 and 9 (circular nodes) are ineligible to be served via UAV (e.g., due to parcel weight restrictions).

Empirical results : Contrasting the FSTSP and PDSTSP



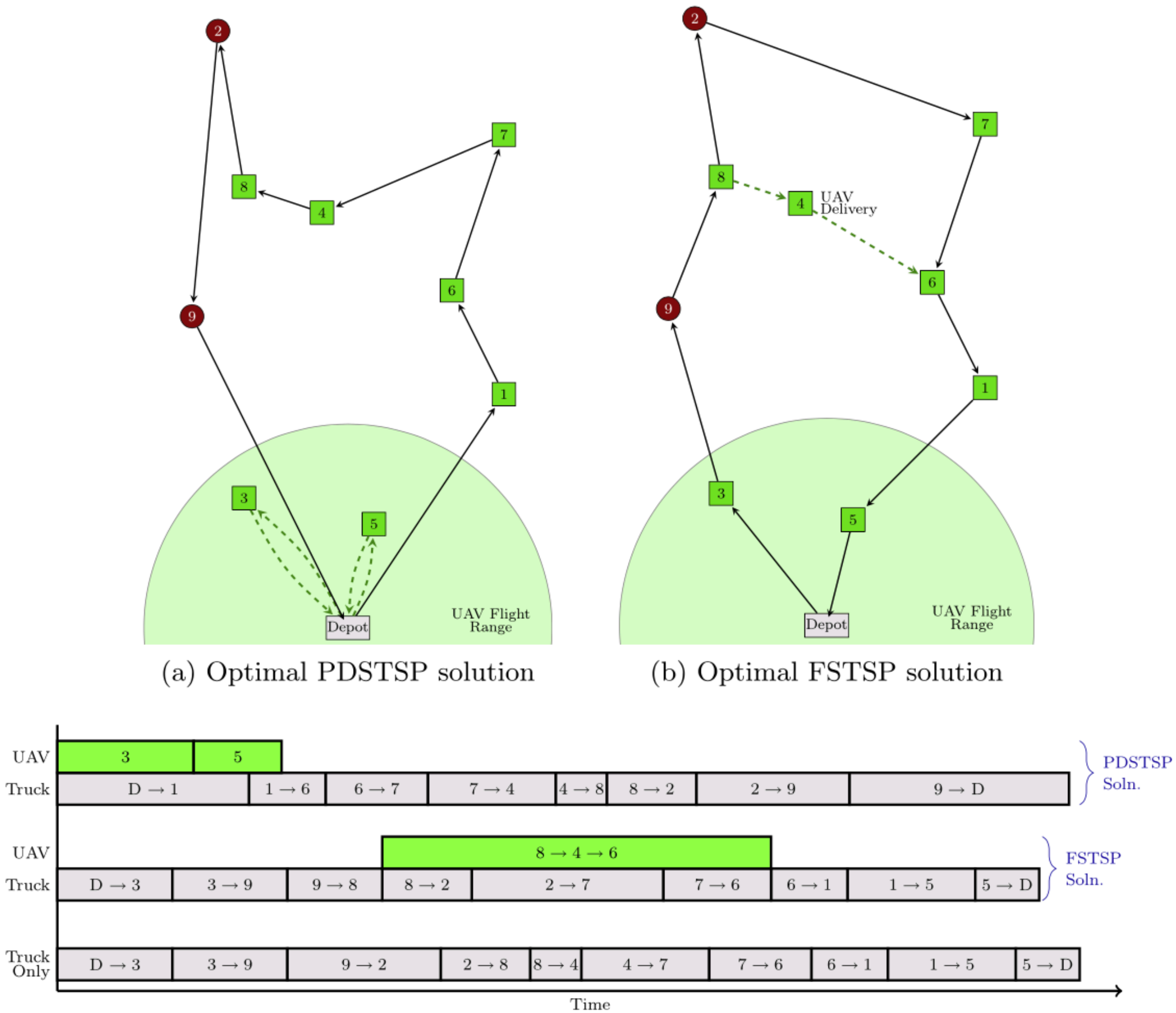
(a) Optimal PDSTSP solution

(b) Optimal FSTSP solution

(c) A comparison of delivery schedules for the two systems depicted above. The "Truck Only" schedule represents an optimal TSP solution for a single truck serving all customers.

Fig. 10. Four customers are within the UAV's flight radius from a slightly offset depot. Customers 2 and 9 (circular nodes) are ineligible to be served via UAV (e.g., due to parcel weight restrictions).

Empirical results : Contrasting the FSTSP and PDSTSP



(a) Optimal PDSTSP solution

(b) Optimal FSTSP solution

(c) A comparison of delivery schedules for the two systems depicted above. The "Truck Only" schedule represents an optimal TSP solution for a single truck serving all customers.

Fig. 11. Only two customers are within the UAV flight radius from a remotely-located depot. Customers 2 and 9 (circular nodes) are ineligible to be served via UAV (e.g., due to parcel weight restrictions).

FSTSP & PDSTSP vs HDTRP

- **FSTSP**
  - 드론을 트럭에 실어서 배송하며 드론은 배송 후 트럭(또는 depot)으로 돌아옴
  - 트럭과 드론이 동시에 움직이며 배송하는 것이 가장 큰 장점
  - 그러나 드론을 날리고 회수하는 과정에서 불확실성이 크기 때문에 위험할 수 있음 (교통상황 등으로 인해 트럭의 움직임을 예측하기 어려움)

- **PDSTSP**
  - Depot에서 한 개 이상의 드론이 비행 범위 내에 있는 고객에게 배송
  - 트럭과 드론을 synchronization 하지 않아도 되는 것이 장점
  - 그러나 drone station을 설치하고 유지하는 것은 상당한 비용이 요구됨

- **HDTRP**
  - 배터리 용량과 드론 속도가 각각 다른 heterogeneous drone을 사용함
  - " waiting node”를 통해 트럭이 대기함으로써 위의 방법들의 단점을 보완할 수 있음
  - 트럭과 드론을 동시에 움직이지 않고 트럭을 대기시켜 불확실성을 줄이는 좀 더 현실적인 방법
  - 드론을 트럭에 실어서 배송하기 때문에 drone station이 따로 필요없음
  - 위의 방법들과 달리 휴리스틱이 아닌 exact algorithm을 제시함