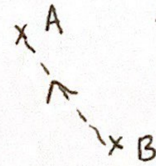
$\frac{n(n-1)}{2}$ trajectories

↳ each traj has a timecost



cost calculation callback

11/3/2023

time for reward calc

$\delta_{gu} = t = T +$ traj time cost $+$ comm upload time

time since simulation start

reward $= \chi_g \gamma^{(\delta_{gu} - \delta_{max})}$

Branch-and-Bound for mTSP

Deployment Model:

- N (includes depot) clusters (cities),
- U UAVs (salesmen),
- All UAVs should start and end at the takeoff/landing zone (depot),
- Each cluster must be visited only once by only one UAV.

GRAPH DEFINITIONS:

$V \triangleq \{1, 2, 3, \ldots, N\}$ vertex set of nodes (clusters/cities),

$S = V - \{1\}$ ↳ depot vertex / depot node / takeoff-landing zone,

$$C \triangleq \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1N} \\ c_{21} & c_{22} & \cdots & c_{2N} \\ \vdots & \vdots & & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NN} \end{bmatrix}$$

$c_{ij} \in C$: Distance from node i to node j, $\forall i,j \in V$,

$c_{ii} = 0$ (same node), $\forall i \in V$,

$C^T \stackrel{?}{=} C$

not necessarily true { If $c_{ij} = c_{ji}$, $\forall i,j \in V$, then C is symmetric.

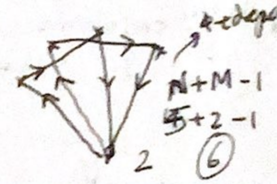
↳ So, C can be asymmetric

↳ Also, Euclidean if C satisfies the Euclidean norm; else C is non-Euclidean.

$E \triangleq \{(i,j) : 1 \le i < j \le N\}$

Arc set

Subtour: $\{(i_1, i_2), (i_2, i_3), \ldots, (i_k, i_1)\}$,
$i_p \neq i_q$, $p \neq q$, and $\forall 1 \le p, q \le k$, is a subtour (cycle) of size k.



Immediate subtour : $\{(1, i), (i, 1)\}, \ \forall i \in S.$

PROBLEM FORMULATION

U fixed

$$Z^* = \min_{X} \left\{ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} x_{ij} + \underbrace{\sum_{i \in S} c_{i1} x_{i1}}_{\text{- reward; not needed}} \right\} \text{ s.t. } \quad (P)$$

All U UAVs depart from depot: $\sum_{j \in S} x_{1j} = U, \quad (1)$ $\quad \sum_{i \in S} x_{i1} = U, \quad (2)$ All U UAVs return to depot

any other city/cluster visited by only one UAV $\left\{ x_{j1} + \sum_{i<j} x_{ij} + \sum_{i>j} x_{ji} = 2, \quad \forall j \in S, \quad (3) \right.$

sub-tour elimination $\left\{ \sum_{\substack{i,j \in S_k \\ i<j}} x_{ij} \le |S_k| - 1, \quad \forall S_k \subseteq S; \quad (4) \right.$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} x_{ij} = N-1, \quad \text{solution should contain } (N+M-1) \text{ arcs} \quad (5)$$

$$x_{ij} = 0, 1, \quad \forall 1 \le i < j \le N, \quad (6)$$

$$x_{i1} = 0, 1, \quad \forall i \in S. \quad (7)$$

LAGRANGIAN:

Lagrangian multiplier

$$\underset{x_{ij} \in X}{\text{maximize}} \left\{ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} r_{ij} x_{ij} + \sum_{j=2}^{N} \lambda_j \left(2 - \sum_{i=1}^{j-1} x_{ij} - \sum_{i=j+1}^{N} x_{ji} - x_{j1} \right) \right\}$$

s.t. to (1), (2), (4)-(7)

$$\mathcal{L}(\lambda) = \max_{X} \left\{ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \tilde{r}_{ij} x_{ij} + \cancel{\sum_{i=2}^{N} r_{i1}} \sum_{j=2}^{N} 2\lambda_j \right\} \quad (\varphi)$$

$$\tilde{r}_{ij} = r_{ij} - \lambda_i - \lambda_j, \quad \lambda_1 = 0.$$

2 × × 3 $\quad r$

Depot 1

N=3

$$\sum_{i=1}^{2}\sum_{j=i+1}^{3} r_{ij}x_{ij} + \sum_{j=2}^{3}\lambda_j\left(2-\sum_{i=1}^{j-1}x_{ij}-\sum_{i=j+1}^{N}x_{ji}-x_{j1}\right)$$

$$r_{12}x_{12}+r_{13}x_{13}+\cancel{r_{22}x_{22}}+r_{23}x_{23}+$$

$$\lambda_2(2-x_{12}-x_{23}-x_{21})$$

$$+$$

$$\lambda_3(2-x_{13}-x_{23}-x_{31})$$

$$r_{12}x_{12}+r_{13}x_{13}+\cancel{r_{22}x_{22}}+r_{23}x_{23}+2\lambda_2-x_{12}\lambda_2-x_{23}\lambda_2-x_{21}\lambda_2$$
$$+2\lambda_3-x_{13}\lambda_3-x_{23}\lambda_3+x_{31}\lambda_3+\cancel{r_{21}x_{21}}+r_{31}x_{31}$$

not there because not needed remove in P

$$(r_{12}-\lambda_2 \cancel{-\lambda_1})\lambda_{12}+(r_{13}-\lambda_3 \cancel{-\lambda_1})x_{13}$$
$$0 \quad + 2\lambda_2+2\lambda_3$$

$$-\lambda_1\cancel{x_{12}}+\cancel{\lambda_1 x_{12}}-\cancel{\lambda_1 x_{13}}+\cancel{\lambda_1 x_{13}}$$

$\lambda_1=0$

$$+$$

$$(r_{23}-\lambda_2-\lambda_3)x_{23}$$

$$2\lambda_2+2\lambda_3+$$

$$(r_{12}-\lambda_2)x_{12}+(r_{13}-\lambda_3)x_{13}+(r_{23}-\lambda_2-\lambda_3)x_{23}-x_{21}\lambda_2-x_{31}\lambda_3$$
$$+x_{21}r_{21}+x_{31}r_{31}$$

add them

$$x_{21}(r_{21}-\lambda_2)$$
$$+x_{31}(r_{31}-\lambda_3)$$

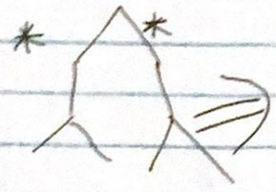
$$\mathcal{L}(\underline{\lambda}) \overset{\max}{\underset{x}{=}} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\tilde{r}_{ij}x_{ij}+\sum_{i=2}^{N}\tilde{r}_{i1}x_{i1}+\sum_{j=2}^{N}2\lambda_j.$$

$\downarrow$ SGD type alg gives $\underline{\lambda}^{(i)}$

For a fixed set of multipliers:

(DCMST) (maximum)

① On this graph, solve the degree-constrained minimum spanning tree using $\tilde{R}$ ($\tilde{x}_{ij}$). Node 1 can have exactly U edges adjacent to it.



Cost of this DCMST $= W(\underline{\lambda})$.

② Using R (x_{ij}), select the $\cancel{M}$ U minimal cost edges adjacent to node 1 as return arcs for $\forall$ UAVs.

Cost of this $= U(\underline{\lambda})$.

③ $$\tilde{\mathcal{L}}(\underline{\lambda}) = W(\underline{\lambda}) + U(\underline{\lambda}) + 2\sum_{j=2}^{N} \lambda_j.$$

Make sure there are at most $(M-1)$ immediate cycles.

If from the $\cancel{M}$ U edges, together with the MST tree edges, have at most $(\cancel{M}\, U-1)$ immediate cycles okay, go to step ③; else, replace the highest-cost edge among the $\cancel{M}$ U edges in ② with the $(\cancel{M}\, U+1)^{th}$ lowest cost edge adjacent to node 1.

$$\tilde{L}(\underline{\lambda}) \leq Z^*$$

choose $\underline{\lambda}$ to get tightest bound

$$\tilde{\mathcal{L}}(\underline{\lambda}^*) = \max_{\underline{\lambda}} \left\{ \tilde{L}(\underline{\lambda}) \right\}$$

choose $\underline{\lambda}$ such that you push the lower bound $\tilde{L}(\underline{\lambda})$ as close to Z^*.

<u>Subgradient</u>

$$\gamma_j^k = 2 - \left(x_{j1} + \sum_{i=1}^{j-1} x_{ij} + \sum_{i=j+1}^{N} x_{ji} \right), \ \forall j \in S.$$

$$\lambda_j^{k+1} = \lambda_j^k + t_k \gamma_j^k \text{, where}$$

t_k → step-size

$$t_k = \frac{\vartheta_k (\bar{Z} - \tilde{L}(\underline{\lambda}^k))}{\|\bar{\gamma}^k\|^2}$$

ϑ_k: user-determined step-size (typically, starts with 2 and halved every (10-20 successive iterations) in which we have no improvement in $\tilde{L}(\underline{\lambda}^k)$)

$\bar{Z}$: approx. to Z^*, usually set to an upper bound on Z^*

① Generate an initial feasible solution ⇒ Cost is $\bar{Z}$.
(maybe, a successive node insertion initialization)

② Initialize $\underline{\lambda}$ (maybe $\lambda_i = $ second minimum in $\{c_{ij} \, (-r_{ij}), \ j = 1, 2, \ldots, N, \ j \neq i\}$) $\forall i \in S$

③ Set $\beta = 1$, if cost data is integer, else 0.

④ Compute the solution to the Lagrangian problem. $\tilde{L}(\underline{\lambda})$
(make sure the solution is feasible)

⑤ Terminate if $\tilde{L}(\underline{\lambda}) \geq \bar{Z} - \beta$, otherwise continue to ⑥

⑥ Update $\underline{\lambda}$ using subgradients. Repeat 2 – 6 until no significant improvement in $\tilde{L}(\underline{\lambda})$

⑦ Sensitivity analyses → B&B

Once we have λ^*, $\tilde{\mathcal{L}}(\lambda^*)$, use implicit enumeration to perform sensitivity analyses

Reduce size:

any subset of arcs forced to be in the solution can be condensed

$$10 \xrightarrow{1} \cancel{2} \xrightarrow{1} \cancel{18} \xrightarrow{1} 3$$

forced to be 1 by sensitivity analyses

$$10 \longrightarrow 3$$

$c_{10,3}$

Σc_{ij} $(c_{10,2} + c_{2,18} + c_{18,3})$

Remove subtours:

If any subset of arcs "forced" in the solution form a sub-tour

Add the costs of these arcs to the obj as a constant, remove all nodes in the subtour from further analysis except depot, and reduce U by 1.

But, we do not need problem reduction here.
(U and M are small enough)

Choose an appropriate upper bound for the main problem Z_f.

Separation & Branching

$B: \{b_1, b_2, \ldots, b_{N+U-1}\} = \{(i_1, j_1), \ldots, (i_{N+U-1}, j_{N+U-1})\}$

are the set of edges in a particular solution to Q

W_k : variable corresponding to b_k, $b_k \in B$.

$L_k : \{ i \mid i = i_k \,/\, i = j_k,\ i \neq 1 \}$

p_k : change in soln value if $W_k = 0$

q_k : change in soln value if $W_k = 1$

d_i : degree of node i in B

N_k : number of variables in B forced to zero if $W_k = 1$.

For a given solution B: compute all p_k and q_k values.

Use p_k and q_k to identify the separation variable.

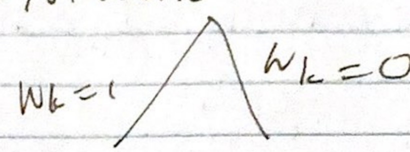
Choose W_k in decreasing order of importance:

for branch repetition

1. highest value of $\lceil q_k + Z(\underline{\lambda}) \rceil$?
2. $N_k > 0$?
3. $\begin{cases} \max[0, d_{j_k} - 2] + \max[0, d_{i_k} - 2], & i_k \neq 1, j_k \neq 1 \\ \max[0, d_{j_k} - 2], & i_k = 1 \\ \max[0, d_{i_k} - 2], & j_k = 1 \end{cases}$
4. highest value of $\lceil p_k + Z(\underline{\lambda}) \rceil$?
5. highest value of $p_k + Z(\underline{\lambda})$.

Once W_k is chosen as the branch separation variable,

$$W_k = \begin{cases} 1, & \text{if } \lceil \tilde{L}(\underline{\lambda}) + q_k \rceil \leq \lceil p_k + \tilde{L}(\underline{\lambda}) \rceil \\ 0, & \text{otherwise} \end{cases}$$



START

Read cost data C

Generate initial feasible solution (successive node insertion procedure) → use the cost as upper bound $\bar{Z}$

(9)

Initialize $\underline{\lambda}$, $\underline{\lambda}^*$, β

β: 1 if cost is integer, 0 otherwise

$\underline{\lambda}$, $\underline{\lambda}^*$: 2nd min in $\{c_{ij}\}_{j \neq i}$

Solve the Lagrangian problem

(DCMST → find M/U minimal cost edges adjacent to depot $U(\underline{\lambda})$ → at most M-1/U-1 immediate cycles CHECK & UPDATE → $L(\underline{\lambda}) = W(\underline{\lambda}) + U(\underline{\lambda}) + 2\sum_{j=2}^{N} \lambda_j$)

$W(\lambda)$

~~Yes~~

Is the solution close to feasibility? (number of nodes not at their reqd. degree is less than a preset number, use a heuristic to generate a feasible solution using the solution of the Lagrangian problem) Update $\bar{Z}$

Is $\tilde{L}(\underline{\lambda}) \geq \bar{Z} - \beta$?

Yes → End

No → Is $\tilde{L}(\underline{\lambda}) \geq L(\underline{\lambda}^*)$?

Yes → Set $\lambda^* = \lambda$ → Update subgradient & opt parameters

No → Update subgradient & opt parameters

$\tilde{L}(\underline{\lambda}^*)$ changed enough?

Yes → (back to Solve the Lagrangian problem)

No → (A)

A

↓ artificial

compute upper bound Z_f ← maybe use $\bar{Z}$ before?

↓

Set $\hat{\lambda} = \lambda = \lambda^*$

↓

Apply separation & branching rules

↓

Descend in the branch-and-bound tree

↓

Solve Lagrangian

↓

Is the solution feasible?

↓

$\tilde{Z}(\underline{\lambda}) \geq Z_f - \beta$?

↓ Yes

Backtrack on branch & bound tree

↓

other branch examined? → No

↓ Yes

Branch-and-bound search completed? → No

↓ Yes

Is $\bar{Z} = Z_f$? $\xrightarrow{\text{Yes}}$ $\boxed{Z^* = \bar{Z}}$ → end

↓ no

print: rerun needed → end

No

$\tilde{L}(\underline{\lambda}) > \tilde{L}(\hat{\underline{\lambda}})$?

↓ Yes

Set $\hat{\lambda} = \lambda$

↓

No

Update λ and subgradient parameters

↓

Sufficient Iterations? → Yes

↓ No

Sufficient Improvment in $\tilde{Z}(\underline{\lambda})$ → Yes

↓ No

Set $\lambda = \hat{\lambda}$