כדי להבין כיצד פונקציית האופטימיזציה משתמשת בזוויות ההתקנה ובטווח הראייה, חיוני לפרק את המודלים המתמטיים ואת המבנה הלוגי של הבעיה. להלן הסבר שלב אחר שלב:

1. משתני החלטה

𝑥

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

איקס

ijdekt

‏

: משתנה בינארי שהוא 1 אם יש מצלמה במיקום

𝑖

i עם כיוון אופקי

𝑗

j, כיוון אנכי

𝑑

ד, גובה

𝑒

ה, וזווית ראייה (AOV)

𝑡

t, ו-0 אחרת.

𝑦

𝑘

y

ק

‏

: משתנה בינארי שהוא 1 אם המטרה במיקום

𝑘

k מכוסה על ידי כל מצלמה, ו-0 אחרת.

2. פרמטרים

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

: פרמטר בינארי שהוא 1 אם המטרה במיקום

𝑘

k נראה מהמצלמה במיקום

𝑖

אני עם תצורה

(

𝑗

,

𝑑

,

𝑒

,

𝑡

)

(j,d,e,t), ו-0 אחרת.

𝑁

𝐶

NC: מספר עמדות המצלמה.

𝑁

ℎ

𝐷

NhD: מספר כיוונים אופקיים.

𝑁

𝑣

𝐷

NvD: מספר כיוונים אנכיים.

𝑁

𝐸

NE: מספר גבהים.

𝑁

𝐴

NA: מספר תצורות AOV.

𝑁

𝑇

NT: מספר עמדות יעד.

𝐶

𝑉

𝑅

CVR: שיעור כיסוי מינימלי.

3. תפקוד אובייקטיבי

פונקציית המטרה ממזערת את המספר הכולל של המצלמות בשימוש:

לְצַמְצֵם

∑

𝑖

=

1

𝑁

𝐶

∑

𝑗

=

1

𝑁

ℎ

𝐷

∑

𝑑

=

1

𝑁

𝑣

𝐷

∑

𝑒

=

1

∑

𝑡

=

1

𝑁

𝐴

𝑥

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

למזער∑

i=1

NC

‏

∑

j=1

NhD

‏

∑

d=1

NvD

‏

∑

e=1

‏

∑

t=1

NA

‏

איקס

ijdekt

‏

4. אילוצים

ודא שכל מטרה מכוסה במצלמה אחת לפחות:

∑

𝑖

=

1

𝑁

𝐶

∑

𝑗

=

1

𝑁

ℎ

𝐷

∑

𝑑

=

1

𝑁

𝑣

𝐷

∑

𝑒

=

1

𝑁

𝐸

∑

𝑡

=

1

𝑁

𝐴

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

⋅

𝑥

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

≥

𝑦

𝑘

∀

𝑘

∑

i=1

NC

‏

∑

j=1

NhD

‏

∑

d=1

NvD

‏

∑

e=1

NE

‏

∑

t=1

NA

‏

V

ijdektk

‏

⋅x

ijdekt

‏

≥y

ק

‏

∀k

קישור כיסוי לקיומה של מצלמה:

∑

𝑖

=

1

𝑁

𝐶

∑

𝑗

=

1

𝑁

ℎ

𝐷

∑

𝑑

=

1

𝑁

𝑣

𝐷

∑

𝑒

=

1

𝑁

𝐸

∑

𝑡

=

1

𝑁

𝐴

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

⋅

𝑥

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

≤

𝑁

𝐶

⋅

𝑦

𝑘

∀

𝑘

∑

i=1

NC

‏

∑

j=1

NhD

‏

∑

d=1

NvD

‏

∑

e=1

NE

‏

∑

t=1

NA

‏

V

ijdektk

‏

⋅x

ijdekt

‏

≤NC⋅y

ק

‏

∀k

הקפידו על שיעור כיסוי מינימלי:

∑

𝑘

=

1

𝑁

𝑇

𝑦

𝑘

≥

𝑁

𝑇

⋅

𝐶

𝑉

𝑅

∑

k=1

NT

‏

y

ק

‏

≥NT⋅CVR

5. שימוש בזוויות התקנה וטווח ראייה

זוויות ההתקנה (אופקיות

𝑗

j, אנכי

𝑑

ד) וה-AOV (

𝑡

t ) משולבים בפרמטר הנראות

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

. כך:

בחישוב

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

כיוונים אופקיים ואנכיים: פרמטר הנראות

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

בודק אם המטרה

𝑘

k נופל בתוך הזווית האופקית והאנכית של התצוגה של המצלמה ממיקום

𝑖

אני עם אוריינטציות

𝑗

י ו

𝑑

ד.

גובה: הגובה

𝑒

e משפיע על הכיסוי האנכי וניתן להשתמש בו כדי לחשב אם היעד

𝑘

k נמצא בטווח האנכי של המצלמה.

זווית ראייה (AOV)

𝑡

t: זה קובע את היקף שדה הראייה. AOV רחב יותר יכול לכסות יותר שטח אבל עם פחות פרטים, בעוד AOV צר יותר מכסה פחות שטח אבל עם יותר פרטים.

דוגמא של

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

תַחשִׁיב:

כדי לקבוע

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

, אתה עשוי להשתמש בחישובים גיאומטריים או באלגוריתמים להטלת קרניים כדי לבדוק אם נקודת יעד

𝑘

k נמצא ב-FOV של מצלמה הממוקמת במיקום

𝑖

i עם התצורות הנתונות. זה כרוך בדרך כלל:

חישוב אזור הכיסוי או הנפח של המצלמה בהתבסס על מיקומה, כיוונים ו-AOV.

בודקים אם נקודת היעד

𝑘

k נמצא בשטח מחושב זה.

דוגמה ליישום ב-Python (באמצעות PuLP):

להלן סקיצת יישום ב-Python באמצעות PuLP המשלבת את הרעיונות האלה:

import pulp

import numpy as np

def calculate\_visibility\_matrix(NC, NhD, NvD, NE, NA, NT):

# Placeholder: Replace this with actual visibility calculations

V = np.zeros((NC, NhD, NvD, NE, NA, NT))

# Assuming some logic to determine visibility

for i in range(NC):

for j in range(NhD):

for d in range(NvD):

for e in range(NE):

for t in range(NA):

for k in range(NT):

# Example condition for visibility

if is\_visible(i, j, d, e, t, k):

V[i][j][d][e][t][k] = 1

return V

def is\_visible(i, j, d, e, t, k):

# Replace with actual logic to determine if target k is visible

# from camera at position i with orientations j, d, height e, and AOV t

return True

def objective\_function(prob, x, NC, NhD, NvD, NE, NA):

prob += pulp.lpSum(x[(i, j, d, e, t)]

for i in range(NC)

for j in range(NhD)

for d in range(NvD)

for e in range(NE)

for t in range(NA)), "Minimize number of cameras"

def inequality\_constraint1(prob, x, y, v, NC, NhD, NvD, NE, NA, NT):

for k in range(NT):

prob += pulp.lpSum(v[i][j][d][e][t][k] \* x[(i, j, d, e, t)]

for i in range(NC)

for j in range(NhD)

for d in range(NvD)

for e in range(NE)

for t in range(NA)) >= y[k], f"Coverage\_constraint\_for\_target\_{k}"

def inequality\_constraint2(prob, x, y, v, NC, NT, NhD, NvD, NE, NA):

for k in range(NT):

prob += pulp.lpSum(v[i][j][d][e][t][k] \* x[(i, j, d, e, t)]

for i in range(NC)

for j in range(NhD)

for d in range(NvD)

for e in range(NE)

for t in range(NA)) <= NC \* y[k], f"Link\_coverage\_to\_cameras\_{k}"

def inequality\_constraint3(prob, y, NT, CVR):

prob += pulp.lpSum(y[k] for k in range(NT)) >= NT \* CVR, "Minimum\_coverage\_rate"

# Problem definition

prob = pulp.LpProblem("Camera\_Optimization", pulp.LpMinimize)

# Parameters

NC = 3 # Number of camera positions

NhD = 2 # Number of horizontal orientations

NvD = 1 # Number of vertical orientations

NE = 1 # Number of heights

NA = 1 # Number of camera types

NT = 5 # Number of target positions

CVR = 0.9 # Minimum coverage rate

# Calculate visibility matrix

v = calculate\_visibility\_matrix(NC, NhD, NvD, NE, NA, NT)

# Decision variables

x = pulp.LpVariable.dicts("x", ((i, j, d, e, t) for i in range(NC)

for j in range(NhD)

for d in range(NvD)

for e in range(NE)

for t in range(NA)), cat='Binary')

y = pulp.LpVariable.dicts("y", range(NT), cat='Binary')

# Objective function

objective\_function(prob, x, NC, NhD, NvD, NE, NA)

# Add constraints

inequality\_constraint1(prob, x, y, v, NC, NhD, NvD, NE, NA, NT)

inequality\_constraint2(prob, x, y, v, NC, NT, NhD, NvD, NE, NA)

inequality\_constraint3(prob, y, NT, CVR)

# Solve the problem

prob.solve()

# Print the results

print("Status:", pulp.LpStatus[prob.status])

for var in prob.variables():

print(var.name, "=", var.varValue)

print("Optimal number of cameras:", pulp.value(prob.objective))

סיכום:

זוויות וטווח ראייה: אלה קריטיים לקביעת הראות ממצלמה לנקודת יעד.

מטריצת נראות

𝑉

𝑖

𝑗

𝑑

𝑒

𝑘

𝑡

𝑘

V

ijdektk

‏

: מקודד את מידע הראות בהתבסס על תצורת המצלמה ומיקום היעד.

מטרה ואילוצים: בעיית האופטימיזציה ממזערת את מספר המצלמות תוך הבטחה שכל המטרות מכוסות בהתאם לאילוצים שצוינו.

גישה זו מבטיחה שהמספר והמיקומים האופטימליים של המצלמות מחושבים כדי לכסות את כל נקודות היעד תוך כיבוד האילוצים הנתונים.