

Moving-Target TSP in two-orthogonal-axes

Pseudocode-BF

Felix Greuling (666020)

21. Oktober 2019

Algorithmus 1 Brute-Force-Algorithmus für zwei-orthogonale Achsen beim bewegende Ziele in TSP

Input: Ziele Z , Ursprung $z_{ursprung}$, Verfolgergeschwindigkeit v_{max}

Output: Ziele Z in der Tour-Reihenfolge, inklusive Retour zum Ursprung

$B \rightarrow$ Permutationsbaum, bei n Zielen muss es n Kinderknoten geben.

$\tau_{min} \leftarrow \infty$

Sei *aktuelleZiele* das Array mit den aktuellen Indizes von Z , initialisiert mit einsen

Sei k die aktuelle Tiefe im Baum

Sei t das Zeit-Array welches für die aktuelle Ziel-Reihenfolge die Abfangzeit angibt

$k \leftarrow 1$

while *aktuelleZiele*[0] $\leq n$ und *aktuelleZiele*[n] $\leq n$ **do**

$current \leftarrow Z[aktuelleZiele[k]]$

$prev \leftarrow Z[aktuelleZiele[k-1]]$

$t[current] \leftarrow t[prev] + \pi[prev \rightarrow current]$

if $t[current] \geq \tau_{min}$ oder mindestens ein Ziel der verbliebenen Indizes wird zwischen *current* und *prev* abgefangen **then**

if *aktuelleZiele*[k] = n **then**

 Dekrementiere k

 Inkrementiere *aktuelleZiele*[k]

else

 Inkrementiere *aktuelleZiele*[k]

end if

else

if $k < n$ **then**

 Inkrementiere k

else

$t[current] \leftarrow t[current] + \pi[current \rightarrow ursprung]$

if $t[current] < \tau_{min}$ **then**

$\tau_{min} \leftarrow t[current]$

 Dekrementiere k

 Inkrementiere *aktuelleZiele*[k]

end if

end if

end if

end while
