

Rozmieszczanie kamer bezpieczeństwa

Przemysław Kopański

Mateusz Forc

6 maja 2017

Spis treści

1	Treść zadania	1
2	Przyjęte założenia	2
3	Przestrzeń przeszukiwań	2
4	Funkcja celu	3
4.1	Przykład	4
5	Metaheurystyka	5
6	Przewidywane wyniki pracy	5

1 Treść zadania

Jak optymalnie rozmieścić kamery monitoringu w ustalonym pomieszczeniu (rzut z góry), aby minimalną liczbą kamer móc obserwować dowolne miejsce (z uwzględnieniem maksymalnej dopuszczalnej odległości od kamery).

2 Przyjęte założenia

- wszystkie kamery są takie same (mają taki sam zasięg)
 - zasięg kamery jest kołem o stałym promieniu
 - promień zasięgu kamery wynosi 2 (możliwa interpretacja - średnica kamery wynosi 4 metry)
- rzut pomieszczenia reprezentowany jest przez zbiór punktów
 - punkty mają współrzędne odpowiadające I ćwiartce wykresu

$$x \leq 0, y \leq 0$$

- punkty podawane są jako lista, która reprezentuje zamknięty wielokąt - muszą one zostać podane we właściwej kolejności, tak aby można je było jednoznacznie połączyć (każde dwa kolejne punkty łączone są w odcinek)

3 Przestrzeń przeszukiwań

Zbiór kamer wraz z ich pozycjami.

Pierwszy zbiór zawiera [obszar rzutu/obszar jednej kamery] kamer rozmieszczonych losowo wewnątrz wielokąta. Do kolejnego stanu możemy przejść poprzez dodanie/usunięcie kamery lub przemieszczenie jednej z aktualnie umieszczonych kamer. Do zbioru kamer nie można wstawić kamery, która jest na zewnątrz obserwowanego pomieszczenia.

4 Funkcja celu

Dostępna informacja:

k_{min} - minimalna teoretyczna liczba kamer dla aktualnie rozpatrywanego rzutu

Parametry zadania:

d_k - koszt użycia kamery

d_p - wartość pokrycia 1% powierzchni

Funkcja: $f(p, k) = d_p * p - \frac{100d_k}{k_{min}} * \max(k, k_{min})$

gdzie:

p - % pokrycia dla danego stanu

k - ilość użytych kamer w danym stanie

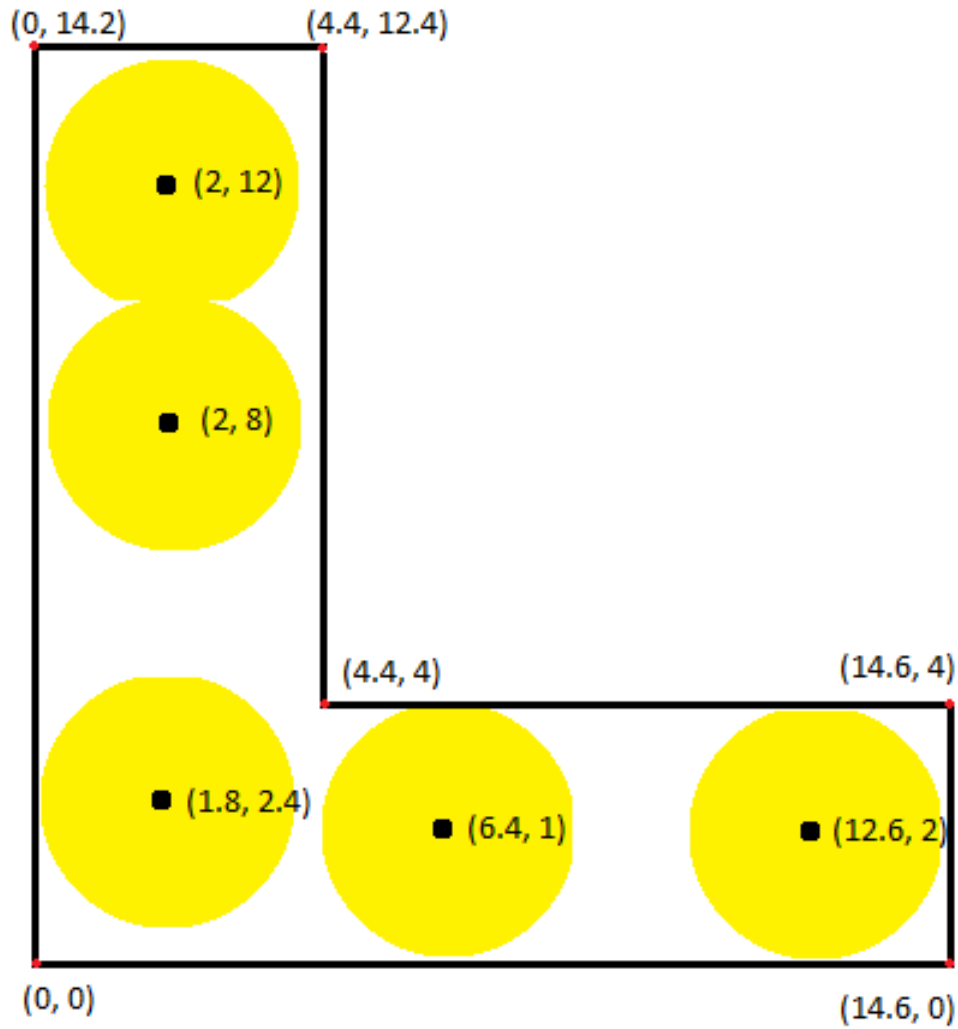
Zadanie polega na maksymalizacji funkcji f.

Przykładowo

$d_p = 1d_k = 1$

Dla podanych parametrów algorytm będzie znajdował „złoty środek pomiędzy” procentem pokrycia a ilością użytych kamer. Odpowiednio przeskalowując podane parametry i trzymając odpowiedni stosunek pomiędzy tymi wartościami, możemy sterować na czym bardziej nam zależy, jeżeli np $d_p = 2d_k = 1$ to będziemy w stanie zaakceptować dwukrotną ilość kamer w zamian za dwukrotnie większe pokrycie.

4.1 Przykład



Podane pomieszczenie ma pole powierzchni równe 103.38.

Zasięgi kamer są między sobą rozłączne, a ich pole wynosi 62.83.

Minimalna teoretyczna liczba kamery wynosi 9.

Pokrycie dla danego stanu wynosi 60.8%.

Dla parametrów: $d_p = 1$ $d_k = 1$

wartość funkcji wynosi $60.8 - 100 = -39.2$

Podane rozwiązanie posiada zbyt małą liczbę kamer.

5 Metaheurystyka

Zastosowany zostanie tabuizowany algorytm symulowanego wyżarzania. Ze względu na to, że stworzenie funkcji heurystycznej do badanej przestrzeni jest obliczalnie trudne, użycie metody A^* jest niewskazane. Algorytmy wspinaczkowe nie sprawdzą się w opisywanej przestrzeni ze względu na dużą liczbę ekstremów lokalnych. Stosując algorytm symulowanego wyżarzania zapewnione jest, że algorytm nie ‘utknie’ w ekstremum. Wraz z rosnącą liczbą iteracji można zmniejszać temperaturę, w celu znalezienia coraz lepszego rozwiązania. Dodatkowym mechanizmem pozwalającym uniknąć zakotwiczenia w ekstremum lokalnym jest tabuizacja.

6 Przewidywane wyniki pracy

Dla kilkunastu zadanych rzutów ilustrujących różne warianty pomieszczeń np. długie, wąskie i kręte, duże otwarte zostaną przeprowadzone symulacje w celu odnalezienia i zaprezentowania parametrów funkcji celu, które pozwalają implementacji na znalezienie możliwie najlepszego rozwiązania dla danego rodzaju przypadku.

Wyniki zostaną zaprezentowane jako zestawy rzutów oraz wykresów ilustrujących pokrycie w zależności od parametrów: d_p i d_k wraz z wyróżnionym zestawem parametrów dla każdego zestawu, który ilustruje najlepsze rezultaty.