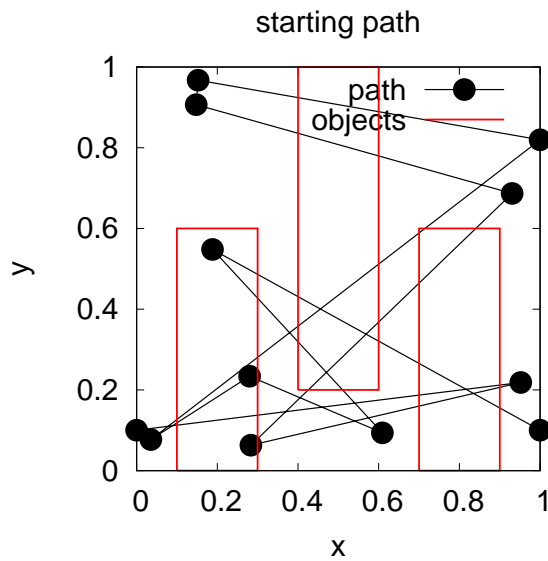


Monte Carlo: wyznaczanie najkrótszej ścieżki metodą symulowanego wyżarzania

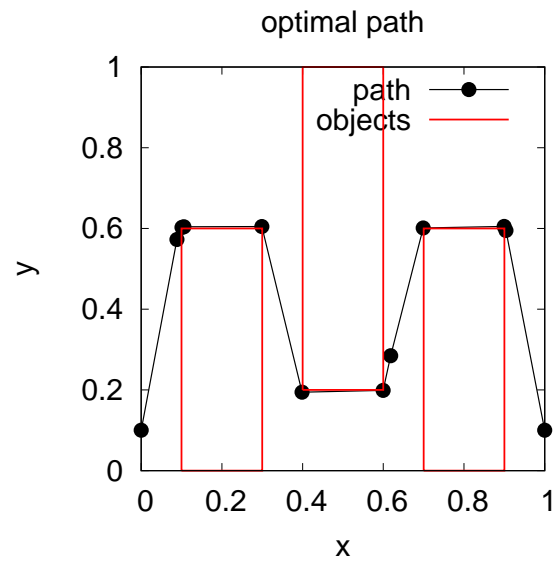
17 listopada 2022

1 Wstęp

a)



b)



Rysunek 1: Geometria układu. Kolorem czerwonym zaznaczony jest obrys przeszkód, które należy ominąć. Kolorem czarnym zaznaczona jest ścieżka: a) początkowa losowo wygenerowana ścieżka i b) znaleziona przy użyciu metody MC ścieżka optymalna.

Na zajęciach wykorzystamy metodę symulowanego wyżarzania (jedną z metod MC) do znalezienia optymalnej ścieżki w układzie z przeszkodami tj. ścieżki o najkrótszej długości omijającej przeszkody (np. budynki w mieście). Przykładowa geometria układu z trzema przeszkodami i ze znalezioną ścieżką pokazana jest na rysunku 1.

2 Metoda symulowanego wyżarzania

2.1 ścieżka

Pojedynczą ścieżkę definiuje zestaw n węzłów

$$path = \{\vec{r}_0, \vec{r}_1, \vec{r}_2, \dots, \vec{r}_{n-2}, \vec{r}_{n-1}\} \quad (1)$$

$$\vec{r}_i = [x_i, y_i] \quad (2)$$

Węzły: \vec{r}_0 i \vec{r}_{n-1} są ustalone gdyż stanowią wejście i wyjście z układu, położenia pozostałych węzłów mogą się zmieniać.

2.2 funkcja celu/kosztu (ang. fitness function)

W dowolnej metodzie MC definiujemy funkcję celu/kosztu, której wartość minimalizujemy. Naszą funkcję zdefiniujemy jako długość całkowitą ścieżki (l) sumując długości poszczególnych odcinków ($d_{i-1,i} = \|\vec{r}_i - \vec{r}_{i-1}\|_2$ - norma euklidesowa)

$$l = \sum_{i=1}^{n-1} d_{i-1,i} \quad (3)$$

$$d_{i-1,i} = \int_{\vec{r}_{i-1}}^{\vec{r}_i} [1 + V(\vec{r})] ds \quad (4)$$

$$ds = \sqrt{dx^2 + dy^2} \quad (5)$$

$$V(\vec{r}) = \begin{cases} V_{max} > 0 & \iff \vec{r} \in \bigcup_k Obstacle_k \\ 0 & \iff \vec{r} \notin \bigcup_k Obstacle_k \end{cases} \quad (6)$$

Długość ścieżki (l) jest sumą długości odcinków $d_{i-1,i}$ które dostajemy całkując element drogi ds . Aby uwzględnić wpływ przeszkód (*Obstacles*), wprowadzamy funkcję kary (funkcja $V(\vec{r})$), której wartość jest duża jeśli ścieżka przechodzi przez przeszkodę (kara) lub równa 0 jeśli ją omija (brak kary). Całkowanie wkładu od pojedynczego odcinka wykonujemy numerycznie np. metodą trapezów dla n_w węzłów

$$d_{i-1,i} = \sum_{k=1}^{n_w} [1 + V(\vec{r}_k)] \cdot c_k \cdot \Delta s \quad (7)$$

$$\Delta s = \sqrt{\Delta x^2 + \Delta y^2} \quad (8)$$

$$c_k = \begin{cases} \frac{1}{2} & \iff k = 1 \vee k = n_w \\ 1 & \iff k = 2, 3, \dots, n_w - 1 \end{cases} \quad (9)$$

$$\Delta x = \frac{x_i - x_{i-1}}{n_w - 1} \quad (10)$$

$$\Delta y = \frac{y_i - y_{i-1}}{n_w - 1} \quad (11)$$

$$\vec{r}_k = [x_k, y_k] \quad (\text{węzły całkowania}) \quad (12)$$

$$x_k = x_{i-1} + \Delta x \cdot (k - 1), \quad k = 1, 2, \dots, n_w \quad (13)$$

$$y_k = y_{i-1} + \Delta y \cdot (k - 1), \quad k = 1, 2, \dots, n_w \quad (14)$$

$$(15)$$

2.3 optymalizacja długości ścieżki

Poszukiwanie optymalnej ścieżki odbywa się w sposób stochastyczny.

Uwaga: oznacznie

$$U_1 \sim U(0, 1) \quad (16)$$

należy rozumieć jako losowanie liczby U_1 przy użyciu generatora liczb pseudolosowych o rozkładzie jednorodnym w przedziale $(0, 1)$.

W jednym kroku metody dokonujemy próby zmiany kolejno położenia wszystkich węzłów swobodnych ścieżki ($i = 2, 3, \dots, n - 2$). Dla i – tego węzła wyznaczamy wartość funkcji celu przed i po zmianie (zmiana wpływa tylko na długości dwóch sąsiednich odcinków $d_{i-1,i}$ oraz $d_{i,i+1}$)

$$x_i^{old} = x_i \quad (17)$$

$$y_i^{old} = y_i \quad (18)$$

$$x_i^{new} = x_i^{old} + \delta_x \cdot (2 \cdot U_1 - 1), \quad U_1 \sim U(0, 1) \quad (19)$$

$$y_i^{new} = y_i^{old} + \delta_y \cdot (2 \cdot U_2 - 1), \quad U_2 \sim U(0, 1) \quad (20)$$

$$p_i^{old} = \exp\left(-\frac{d_{i-1,i}^{old} + d_{i,i+1}^{old}}{T}\right) \quad (21)$$

$$p_i^{new} = \exp\left(-\frac{d_{i-1,i}^{new} + d_{i,i+1}^{new}}{T}\right) \quad (22)$$

$$(23)$$

gdzie: T - temperatura, δ_x, δ_y to maksymalne przesunięcia w kierunkach x i y , a następnie określamy prawdopodobieństwo akceptacji położenia nowego węzła

$$U_3 \sim U(0, 1) \quad (24)$$

$$p_{acc} = \min\left\{1, \frac{p_i^{new}}{p_i^{old}}\right\} \quad (25)$$

$$\begin{cases} U_3 \leq p_{acc} & \rightarrow \vec{r}_i = \vec{r}_i^{new} & (AKCEPTACJA) \\ U_3 > p_{acc} & \rightarrow \vec{r}_i = \vec{r}_i^{old} & (ODRZUCAMY) \end{cases} \quad (26)$$

Jeśli parametr T będzie miał małą wartość, wówczas preferowane będą zmiany powodujące skrócenie ścieżki, która omija przeszkody. Należy oczekiwać, iż po wielu takich próbach długość ścieżki zbliży się do optymalnej (najkrótszej możliwej).

2.4 temperatura

Parametr T (temperatura) odpowiada za określenie wartości prawdopodobieństwa akceptacji gorszego wyniku ($p_i^{old} > p_i^{new}$) czyli gdy nowe położenie węzła powoduje zwiększenie funkcji celu (lepsza funkcja celu - niższa - zawsze jest akceptowana). Duża wartość T oznacza duże prawdopodobieństwo akceptacji gorszego wyniku, mała wartość obniża to prawdopodobieństwo. Ponieważ proces jest iteracyjny, wartość temperatury $T \in [T_{min}, T_{max}]$ powiążemy z numerem aktualnej iteracji, startując od wartości maksymalnej (T_{max}).

$$T = T_{max} - (T_{max} - T_{min}) \left(\frac{m}{M}\right)^\mu \quad (27)$$

$$T_{max} = T_0 > T_1 > T_2 > \dots > T_M = T_{min} \quad (28)$$

gdzie: M -całkowita liczba iteracji, m -aktualny numer iteracji, wykładnik μ określa szybkość zmiany T (np. $\mu = 1, \frac{1}{2}$ etc.). Parametr μ decyduje jak długo utrzymuje się wysoka temperatura, dla $\mu \ll 1$ temperatura szybko spada w kolejnych iteracjach, natomiast dla $\mu > 1$ wysoka temperatura utrzymuje się dość długo.

2.5 algorytm

Nie należy się spodziewać, iż uda nam się znaleźć optymalne rozwiązanie zmieniając położenia węzłów w pojedynczej ścieżce. Jednak nasze szanse wzrosną jeśli algorytm symulowanego wyżarzania wykonamy na grupie N_{pop} takich ścieżek, a ponieważ algorytm łatwo zrównoleglić, można to zrobić w tej samej pętli iteracyjnej (pętla z licznikiem ścieżek j) co pokazuje poniższy algorytm

```

inicjalizacja: n, Npop, δx, δy, Tmin, Tmax, M,
               nodes[Npop][n][2], fitness[Npop],
               optimal_path[n][2], actual_fitness, optimal_fitness,
               p1 = p2 = 0

for(m=0; m<M; m++){
  T=wzór (27)
  for(j=0; j< Npop; j++){
    for(i=1; i<(n-1); i++){
      xiold = nodes[j][i][0] <- ściezka j-ta, węzeł i-ty
      yiold = nodes[j][i][1]
      xinew =wzór (19)
      yinew =wzór (20)
      di-1old, diold, di-1new, dinew = wzór (7)
      piold, pinew = wzory (21) (22)
      pacc =wzór (25)
      U3 ~ U(0,1)
      if(U3 ≤ pacc){<- akceptujemy nowe polozenie wezla
        nodes[j][i][0] = xinew
        nodes[j][i][1] = yinew
        p1 ++
      }else{<- przywracamy stare polozenie (jesli zmienilismy)
        nodes[j][i][0] = xiold
        nodes[j][i][1] = yiold
        p2 ++
      }
    }
    fitness[j]=wzór(3)
  }

// szukamy najlepszego rozwiazania w danej iteracji
actual_fitness=1010
for(j=0; j<Npop; j++){
  if(actual_fitness>fitness[j]){
    actual_fitness=fitness[j]
    nr_actual=j
  }
}

```

```

}

// aktualnie najlepsze rozwiązanie porównujemy z rozwiązaniem globalnym
if(actual_fitness < optimal_fitness){
    optimal_fitness=actual_fitness
    optimal_path [] []=nodes[nr_actual] [] []
}

// co 200 iteracji kontrolujemy stan symulacji
if(m%200 == 0){
    prob_av=p1/(p1+p2)    <-  aktualne srednie prawdopodobienstwo akceptacji
    p1 = p2 = 0 <-reset
    zapis do pliku:  m, T, prob_av, actual_fitness, optimal_fitness
}
}

```

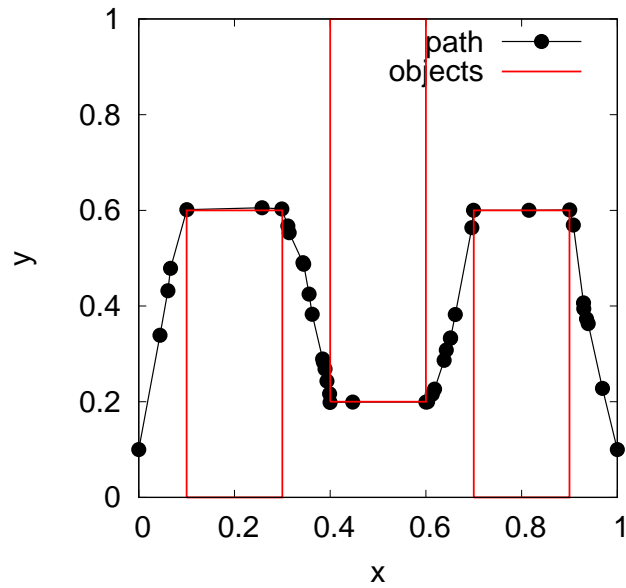
W algorytmie użyto zbioru N_{pop} ścieżek, których położenia węzłów zapisane są w tablicy *nodes*. W trakcie symulacji staramy się kontrolować wyniki, dlatego co 200 iteracji należy wypisać na ekran (i do pliku) parametry: numer iteracji, temperaturę, najlepszą w danej iteracji długość ścieżki oraz najlepszą znaną dotychczas (globalną). Ponadto na bieżąco obliczane jest średnie prawdopodobieństwo akceptacji zmiany położenia (uśredniane jest po 200 iteracjach oraz wszystkich ścieżkach i węzłach).

3 Zadania do wykonania

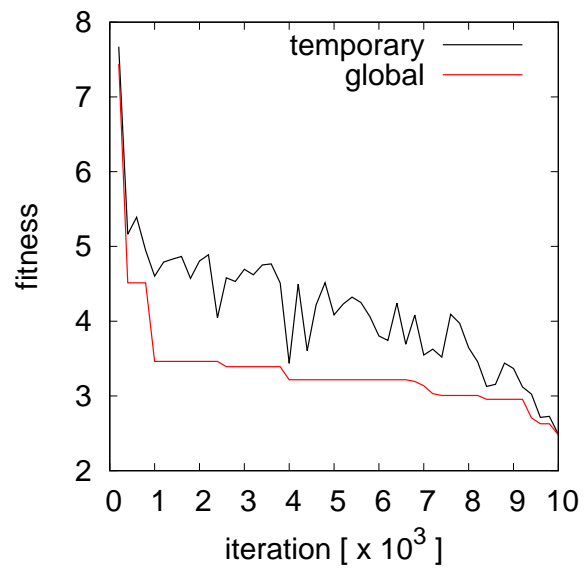
- Obszar obliczeniowy i parametry geometryczne przeszkód definiujemy tak jak na Rys.1. Przyjmujemy parametry startowe symulacji
 - obszar obliczeniowy: $x_{min} = y_{min} = 0, x_{max} = y_{max} = 1.0$
 - węzły ustalone (wejście/wyjście): $\vec{r}_0 = [0.0, 0.1], \vec{r}_{n-1} = [1.0, 0.1]$
 - $\delta x = \delta y = 0.03, T_{min} = 10^{-5}, T_{max} = 1.0, N_{pop} = 50, n = 20, M = 10^4, \mu = 1.0$
 - $n_w = 100, V_{max} = 10.0$
 - liczba przeszkód $n_{obj} = 3$, przeszkody definiujemy w postaci prostokątów więc wystarczy podać położenie dwóch przeciwległych wierzchołków (Lewy-Dolny oraz Prawy-Górny)
 - obiekt 1: $R_{1,LD} = [0.1, 0.0], R_{1,PG} = [0.3, 0.6]$
 - obiekt 2: $R_{2,LD} = [0.4, 0.2], R_{2,PG} = [0.6, 1.0]$
 - obiekt 3: $R_{3,LD} = [0.7, 0.0], R_{3,PG} = [0.9, 0.6]$
- Zaprogramować algorytm symulowanego wyżarzania jak w sekcji 2.5. Węzły wewnętrzne dla każdej ścieżki ($i = 1, 2, \dots, n - 2$) rozmieścić w obszarze obliczeniowym w sposób losowy przy użyciu generatora o rozkładzie jednorodnym - węzły mogą wylądować na przeszkodach, ale to nie stanowi problemu dla metody.
- Sporządzić wykres dowolnej ścieżki startowej, na rysunku zaznaczyć przeszkody. (10 pkt)
- Wykonać symulację dla parametrów startowych, sporządzić wykresy: zmian temperatury, aktualnej i globalnej najlepszej wartości funkcji celu w zależności od numeru iteracji. Narysować układ z przeszkodami i najlepszą znaną ścieżkę. (70 pkt)

5. Powtórzyć symulację dla parametrów: $n = 10$ i $n = 40$. Narysować najlepsze znalezione ścieżki. (20 pkt)

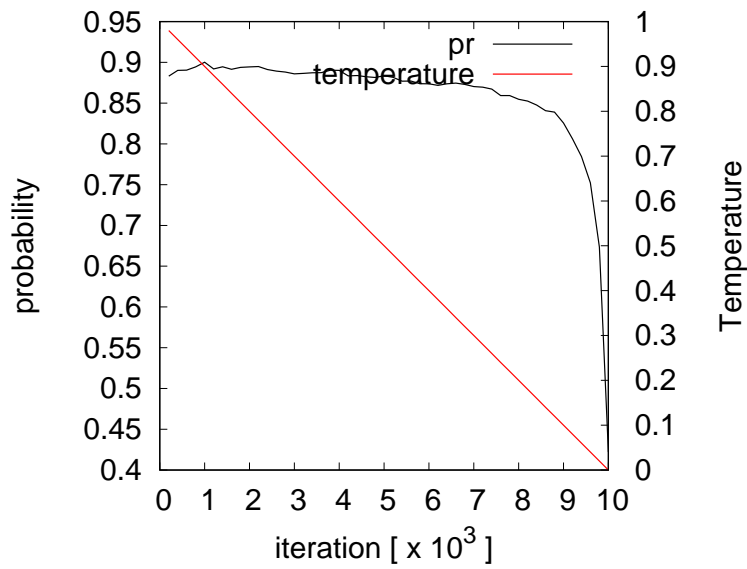
4 Przykładowe wyniki



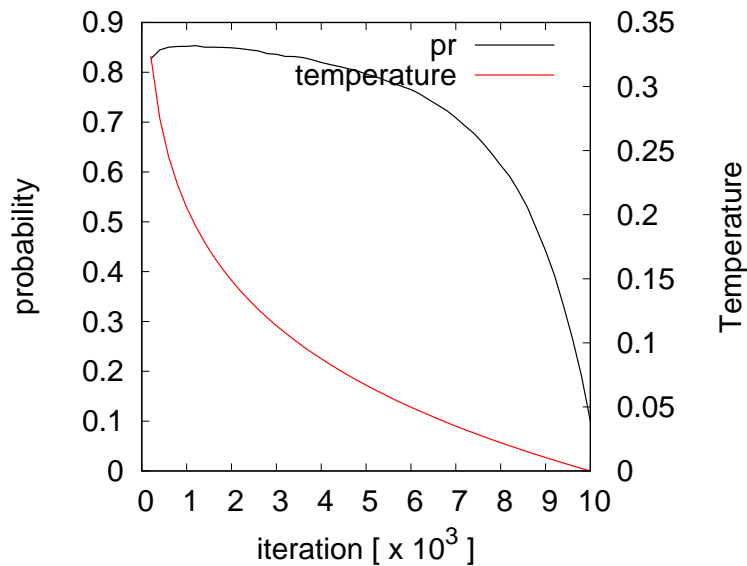
Rysunek 2: Ścieżka zoptymalizowana dla $n = 40$ węzłów.



Rysunek 3: Zmiany funkcji celu w trakcie symulacji.



Rysunek 4: Zmiany temperatury oraz średnie prawdopodobieństwo akceptacji nowego położenia węzła w trakcie symulacji dla $\mu = 1$.



Rysunek 5: Zmiany temperatury oraz średnie prawdopodobieństwo akceptacji nowego położenia węzła w trakcie symulacji dla $\mu = 0.1$.