Zachodniopomorski
Uniwersytet Technologiczny
w Szczecinie

Wydział
Elektryczny

Zmodyfikowana strategia próbkowania w algorytmie RRT

Michał Kubicki

Seminarium KIiA PAN O/Poznań,
26.04.2024, Szczecin

Ogólny opis problemu

Planowanie ścieżki sprowadza się do znalezienia ciągłego bezkolizyjnego ruchu pomiędzy konfiguracją początkową oraz końcową w pewnym otoczeniu roboczym.

W najprostszym przypadku otoczenie robocze jest statyczne i dobrze zdefiniowane, tj. nie występują w nim obiekty dynamiczne, a informacje o aktualnym stanie tego środowiska są dokładne.

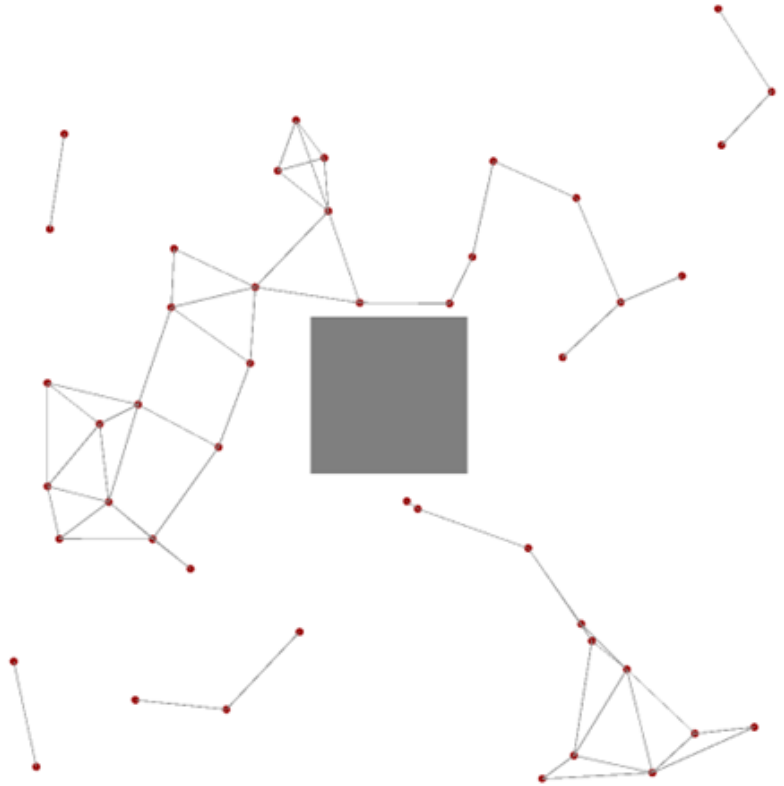
Typowe metody, które starają się rozwiązać ten problem to algorytmy próbkujące przestrzeń roboczą, geometryczne oraz dyskretyzujące przestrzeń roboczą.

Metody próbkujące przestrzeń konfiguracyjną

Metoda **PRM** (**Probabilistic Roadmap**) oraz metoda **RRT** (**Rapidly Exploring Random Trees**) należą do metod próbkujących przestrzeń roboczą. Starają się znaleźć punkty w wolnej przestrzeni konfiguracyjnej, a następnie łączyć je ze sobą w celu uzyskania połączeń pomiędzy kolejnymi dozwolonymi stanami.

Generalnie używane są w przypadku wielowymiarowych, obszernych, skomplikowanych przestrzeni konfiguracyjnych, o których wiedza może być niepełna.

W przypadku planowania ścieżki, połączenie pomiędzy poszczególnymi węzłami będzie oznaczało bezkolizyjny odcinek.

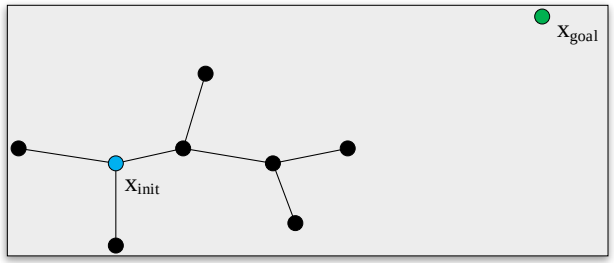


*Metoda **PRM** i tworzona przez nią struktura grafu*

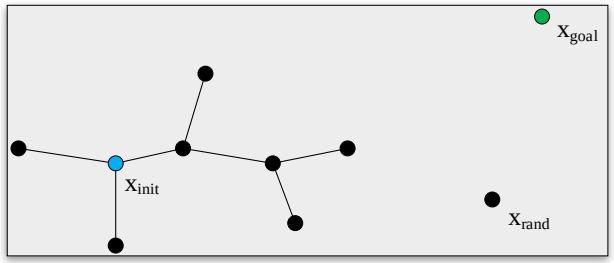
Źródło: Elbanhawi, M., & Simic, M. (2014). Sampling-based robot motion planning: A review. In IEEE Access. https://doi.org/10.1109/ACCESS.2014.2302442

Rapidly Exploring Random Tree (RRT)

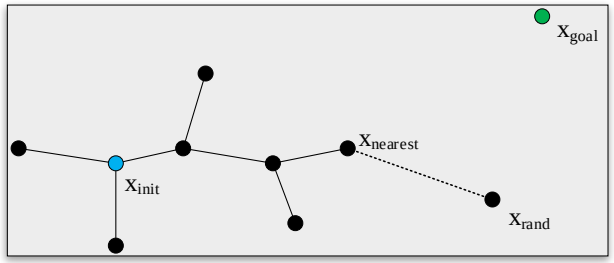
- Oparta jest o strukturę drzewa
- Probabilistycznie zupełna (tj. jeżeli istnieje rozwiązanie to w nieskończonym czasie zostanie ono osiągnięte)
- Single-Query, jednocześnie generowany jest pojedynczy punkt
- Nie gwarantuje optymalności rozwiązania



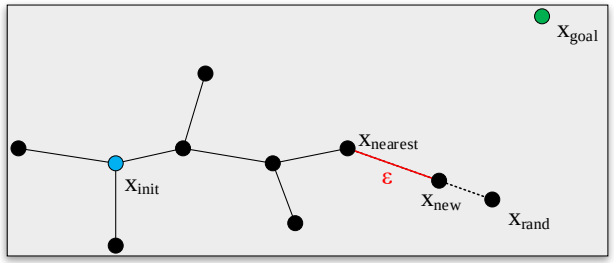
Drzewo początkowe



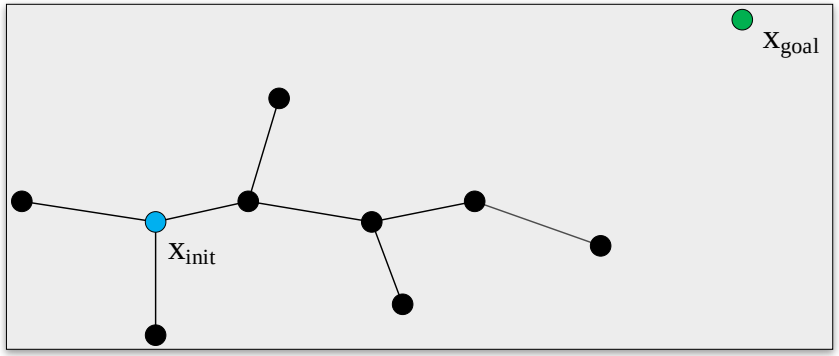
Losowanie nowego punktu x_{rand}



Znalezienie najbliższego istniejącego punktu w drzewie $x_{nearest}$



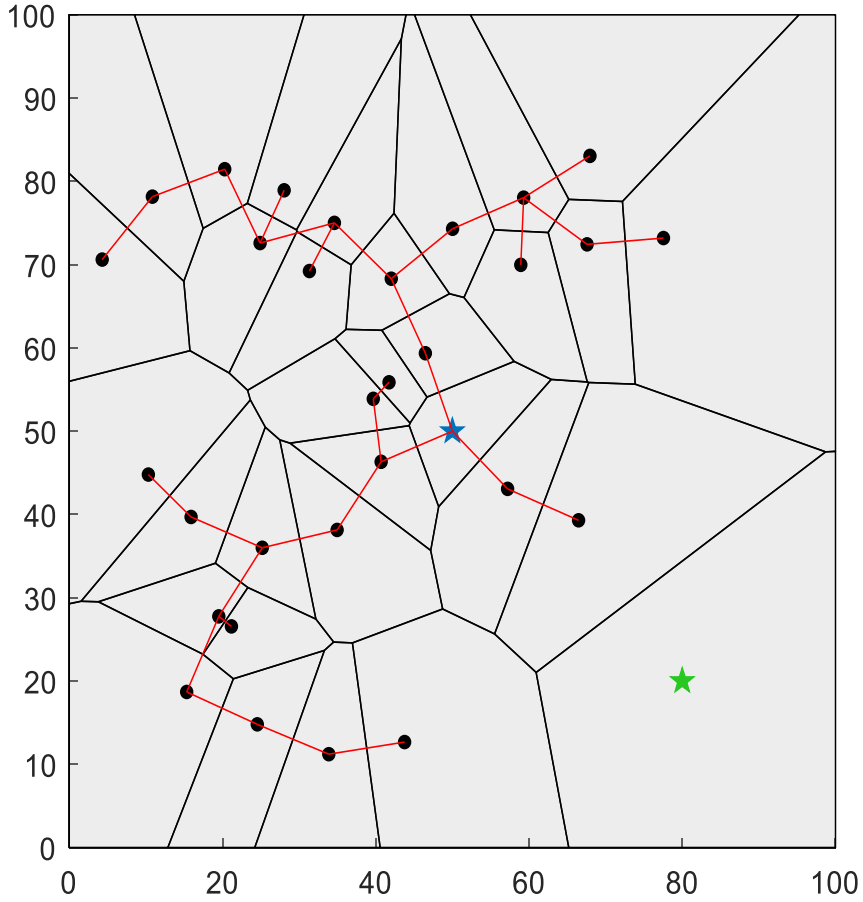
Próba rozszerzenia drzewa o punkt wskazany przez funkcję sterującą x_{new}



Struktura drzewa po dodaniu nowego punktu

Diagram Woronoja

Diagram Woronoja



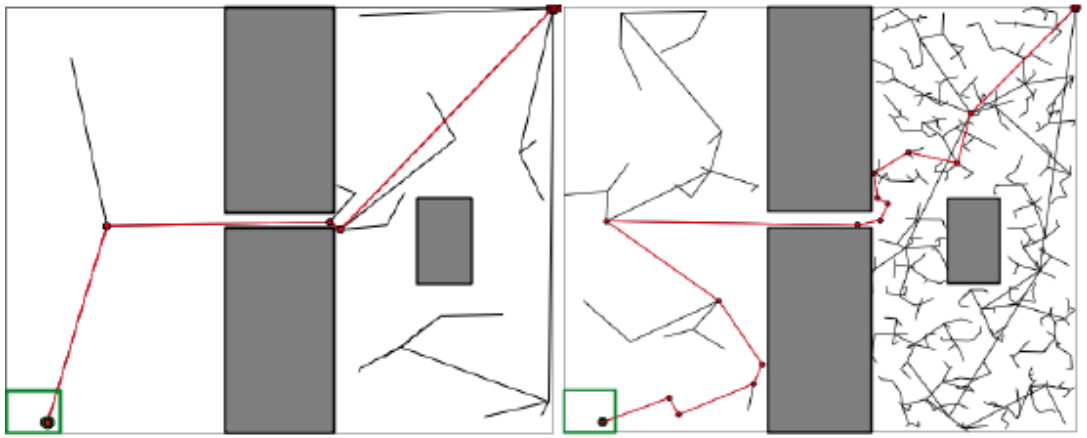
Poszczególne węzły drzewa są punktami, wobec których konstruowany jest diagram Woronoja. Poszczególne obszary P składają się z punktów, które leżą najbliżej względem pojedynczego węzła drzewa.

Prowadzi to do sytuacji, w której punkty losowane w przestrzeni konfiguracyjnej w sposób jednorodny mają szanse znaleźć się w obszarze skojarzonym z węzłem drzewa proporcjonalne do jego rozmiaru.

Jednocześnie, taki wylosowany punkt będzie starał się tworzyć połączenie z węzłem skojarzonym z danym obszarem.

Problemy napotykane przez RRT

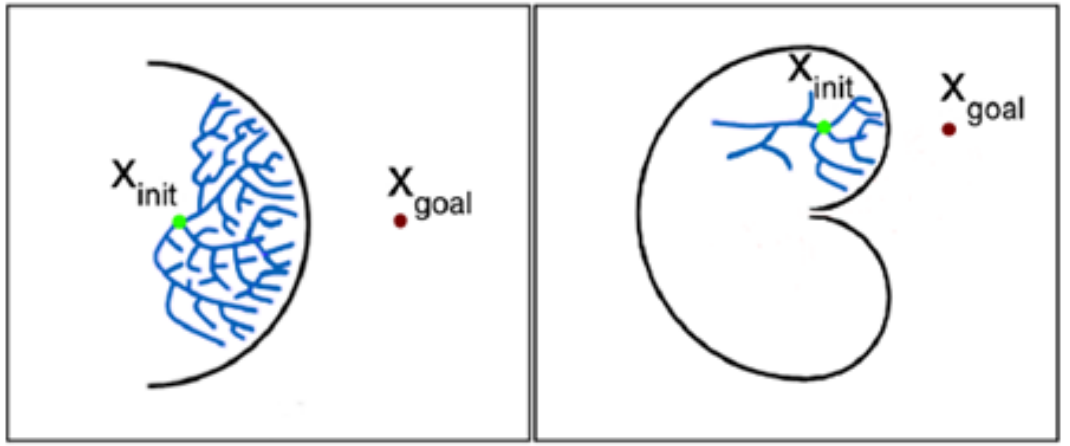
Duża wariancja rozwiązań



Czas oraz jakość znalezionego rozwiązania zależą od czynników losowych

Źródło: Elbanhawi, M., & Simic, M. (2014). Sampling-based robot motion planning: A review. In IEEE Access. https://doi.org/10.1109/ACCESS.2014.2302442

Środowiska, które ograniczają swobodę budowy drzewa



Dwa problematyczne przypadki
Po lewej: *saddle*, po prawej: *bugtrap*

Źródło: Elbanhawi, M., & Simic, M. (2014). Sampling-based robot motion planning: A review. In IEEE Access. https://doi.org/10.1109/ACCESS.2014.2302442

Biasowanie rozrostu drzewa RRT

Ze względu na to, że algorytm RRT sam w sobie nie dąży do osiągnięcia punktu końcowego x_{goal}, to stosuje się tzw. **biasowanie**, które ukierunkowuje rozrost drzewa w pożądanych kierunkach. Metody takie wykorzystują funkcję kosztu, która często bazuje na dystansie do punktu końcowego.

Przykładową implementacją jest algorytm ***hRRT*** (**heuristically-guided RRT**), który ocenia utworzone ścieżki do poszczególnych węzłów drzewa i ich dystans do punktu końcowego.

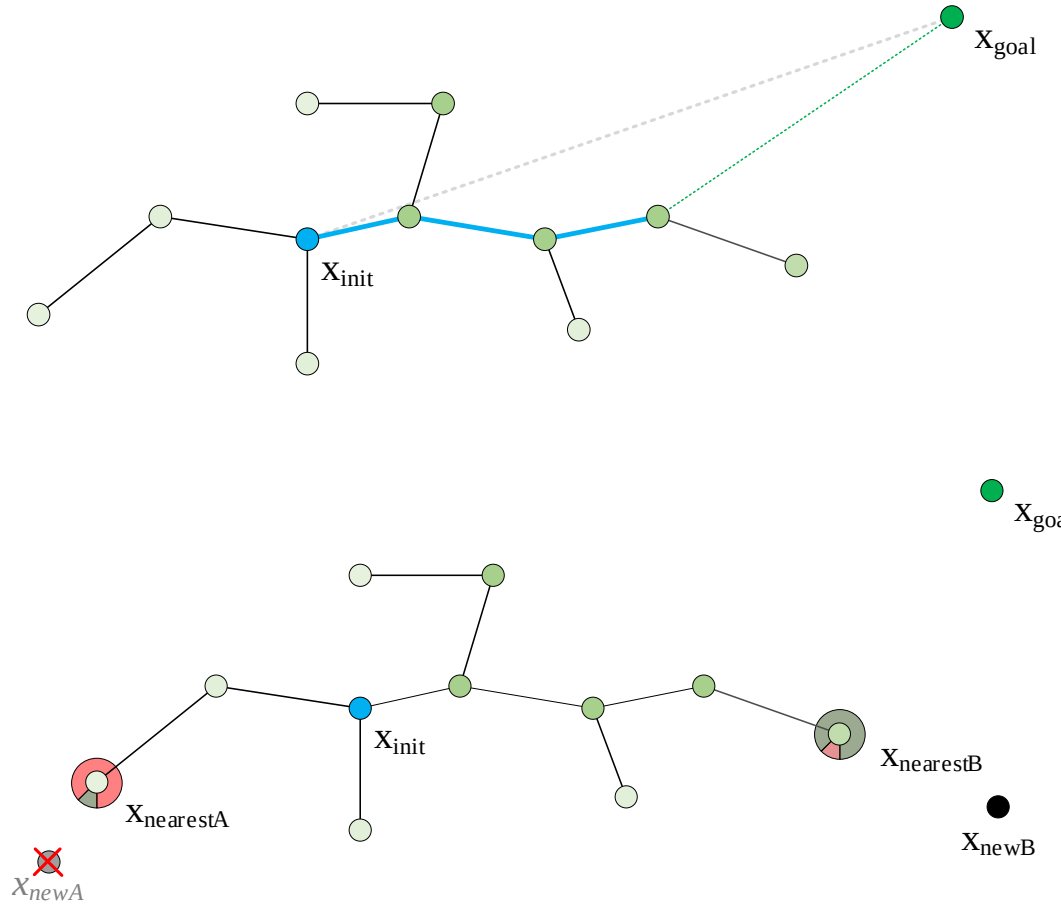
$$m_{quality} = 1 - \frac{C_{vertex} - C_{opt}}{C_{max} - C_{opt}}$$

Losowanie nowego punktu w przestrzeni roboczej odbywa się tak długo, aż wylosowana zostanie wartość r, która przekroczy kryterium jakości.

$$r \sim U(0,1)$$

$$\mathrm{r} < \max(m_{quality}, p)$$

Gdzie p jest parametrem w zakresie [0,1]. Dla wartości 1 algorytm zachowuje się jak RRT.



Proponowana metoda (GAVB-RRT)

Proponowana metoda **GAVB-RRT** (**Graph Assisted Voronoi Bias – RRT**) bazuje na metodzie RRT i podobnie jak ona priorytetyzuje próbkowanie dużych obszarów wyznaczonych grafem Woronoja, lecz skupia się na regionach, które są perspektywicznie. W tym celu wykorzystuje grafy oraz informacje o napotkanych kolizjach.

Algorytm RRT

Wejście: punkt początkowy x_{init}, punkt końcowy x_{goal}, maksymalna ilość punktów n_{max}
$V \leftarrow \{x_{start}\}$; $E \leftarrow \emptyset$;
C $\leftarrow \emptyset$;
for i=1:n_{max}
    $x_{rand} \leftarrow$ Sample()
if not IsCollisionFree(x_{rand})
    $x_{nearest} \leftarrow$ FindNearest(V, x_{rand})
    $x_{new} \leftarrow$ Steer(x_{near}, x_{rand})
    if not IsCollisionFree(x_{new})
        $V \leftarrow V \cup x_{start}$
        $E \leftarrow E \cup \{(x_{nearest}, x_{new})\}$
    end if
else
    C $\leftarrow C \cup x_{rand}$
end for

Algorytm GAVB-RRT

Wejście: punkt początkowy x_{init}, punkt końcowy x_{goal}, maksymalna ilość punktów n_{max}
$V \leftarrow \{x_{start}\}$; $E \leftarrow \emptyset$;
C $\leftarrow \emptyset$;
for i=1:n_{max}
    $P \leftarrow$ ChooseRegion(V, C)
    $x_{rand} \leftarrow$ Sample(P)
if not IsCollisionFree(x_{rand})
    $x_{nearest} \leftarrow$ FindNearest(V, x_{rand})
    $x_{new} \leftarrow$ Steer(x_{near}, x_{rand})
    if not IsCollisionFree(x_{new})
        $V \leftarrow V \cup x_{start}$
        $E \leftarrow E \cup \{(x_{nearest}, x_{new})\}$
    else
        C $\leftarrow C \cup x_{new}$
        x_{near}.collisions $\leftarrow x_{near}$.collisions + 1
    end if
else
    C $\leftarrow$ C $\cup x_{rand}$
end for

Wykorzystanie grafów

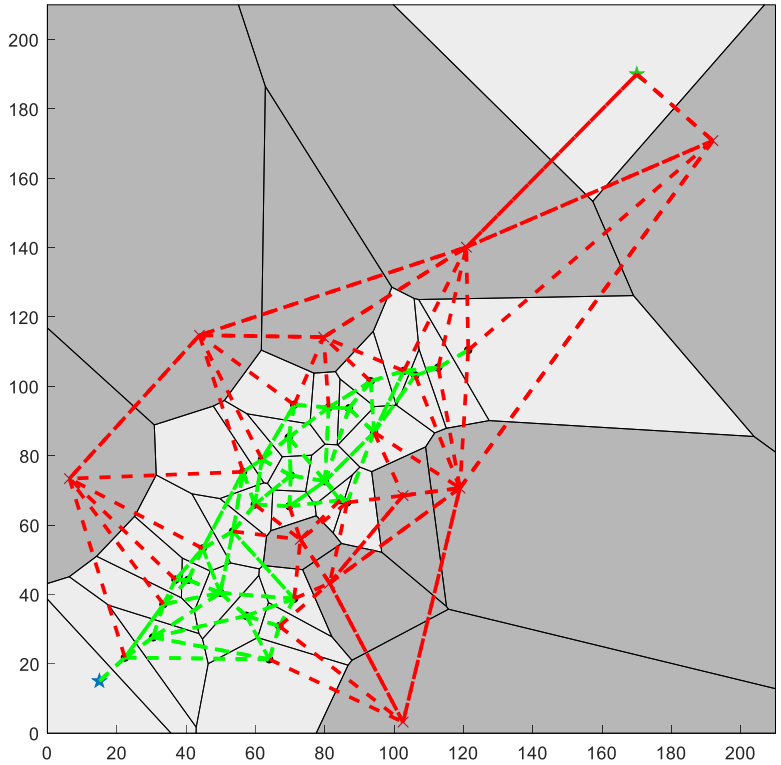
Tworzone są dwa grafy, jeden związany z x_{goal}, który zawiera wszystkie regiony będące przeszkodami widoczne z tego punktu oraz przylegający region wolny.

Drugi związany z x_{init} zawiera wszystkie regiony wolne.

Każda krawędź grafu przyjmuje wartość 1.

Algorytm rozpoznaje dwa przypadki:

1. W przypadku ustalenia, że istnieje bezpośrednie połączenie pomiędzy x_{init} i x_{goal} automatycznie wszystkie punkty są generowane z regionu P_{goal}.
2. W przeciwnym przypadku szukane są:
   A. wolne regiony wraz z sąsiadującymi regionami kolizyjnymi, które mają najmniejszy dystans do P_{goal}
   B. sąsiadujące bezpośrednio ze sobą regiony kolizyjne i wolne o największej łącznej powierzchni



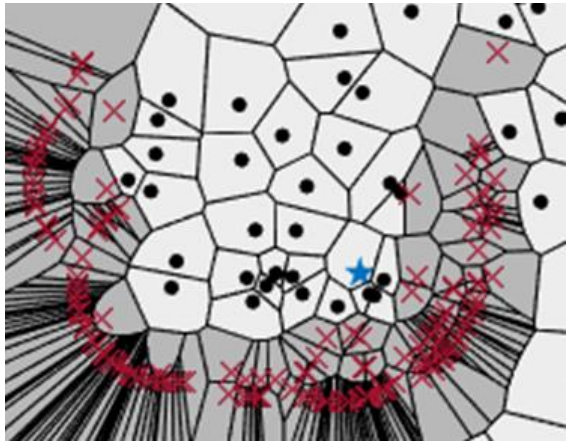
Zmienny maksymalny dystans generacji punktów

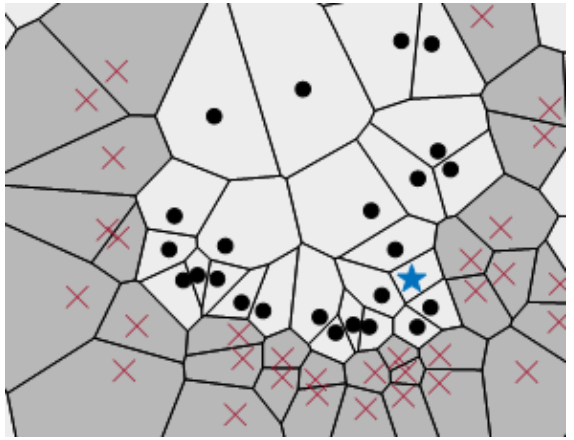
Wartość zmiennej ε, która jest maksymalnym dystansem generacji nowego punktu x_{new} zależy od poprzednich prób, w których wykorzystywany był węzeł $x_{nearest}$. Kolejne kolizje, które nastąpiły z tego węzła są zliczane (c) i wpływają na ε w sposób następujący:

$$\varepsilon(x_{nearest}) = \varepsilon_0^{\frac{1}{c+1}} \; dla \; \varepsilon_0 > 0$$

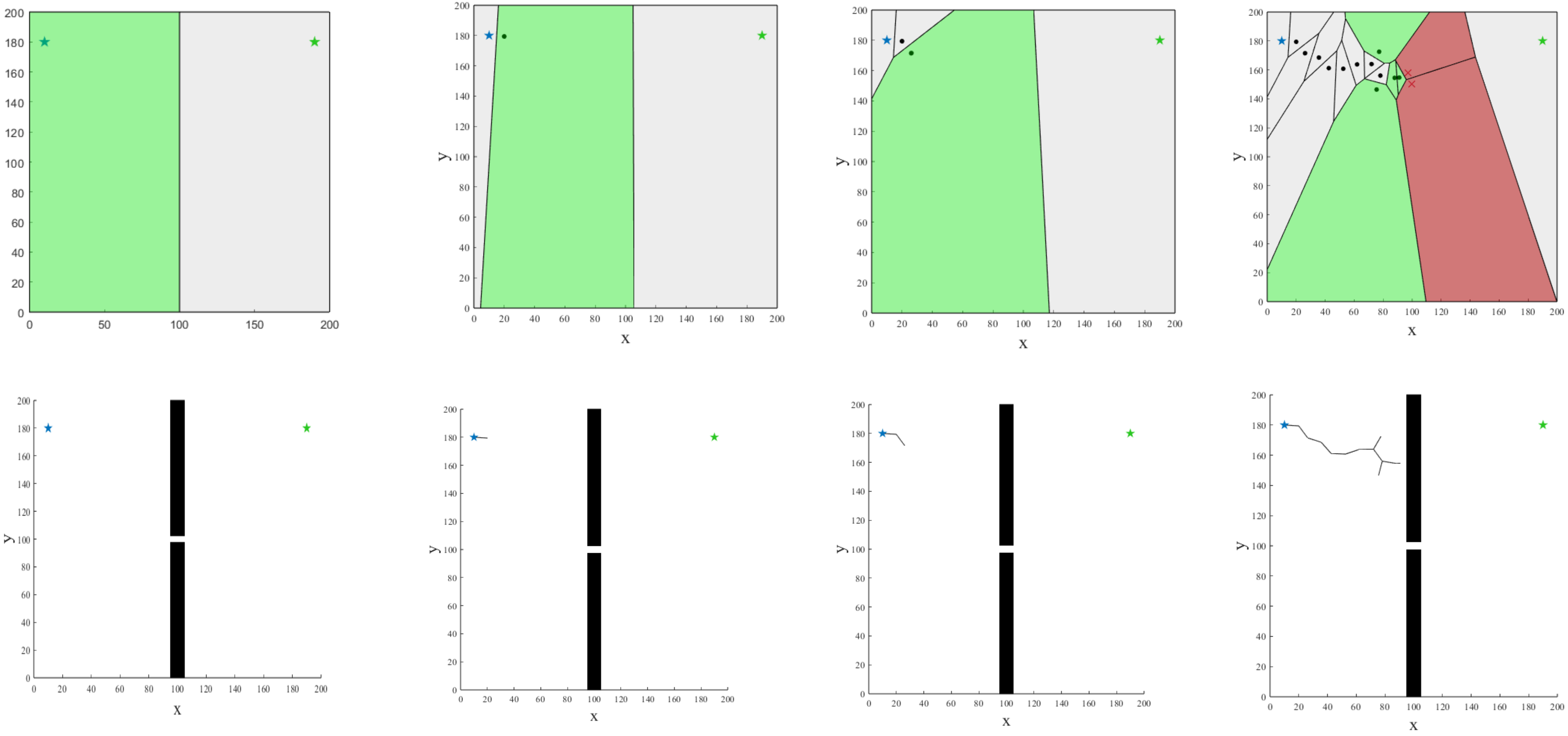
Ma to na celu ograniczyć sytuację, w której kolejne kolizyjne punkty x_{new} wyznaczane są w stałym dystansie od $x_{nearest}$.

Powoduje to dalszą fragmentację przestrzeni, ale jednocześnie nie zmienia znacząco układu obszarów kolizyjnych, w tym ich grafu, oraz całkowitej przylegającej powierzchni do obszaru przynależącego do $x_{nearest}$.

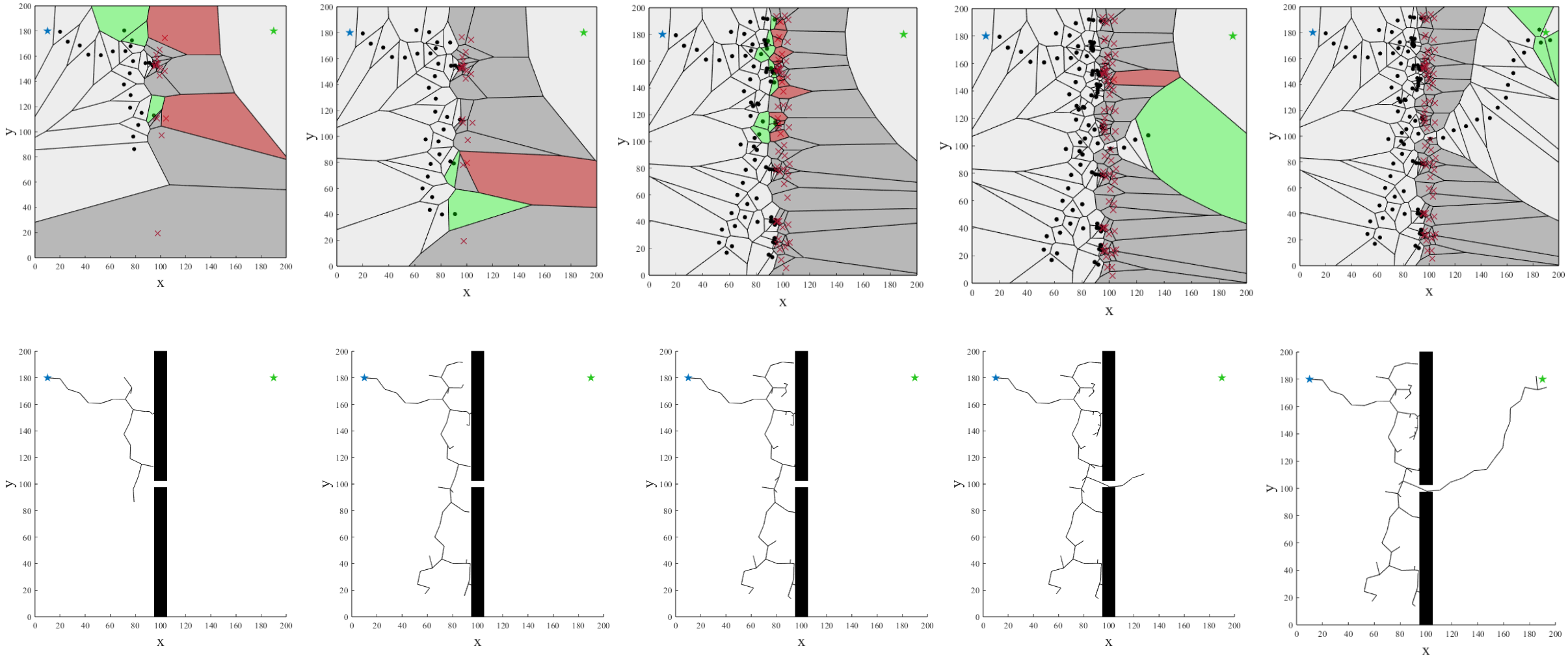




Rozrost drzewa w GAVB-RRT

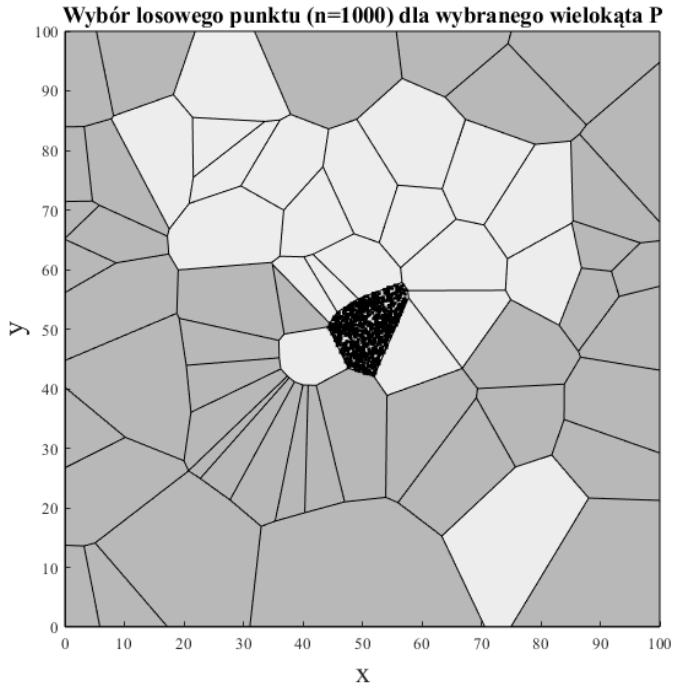


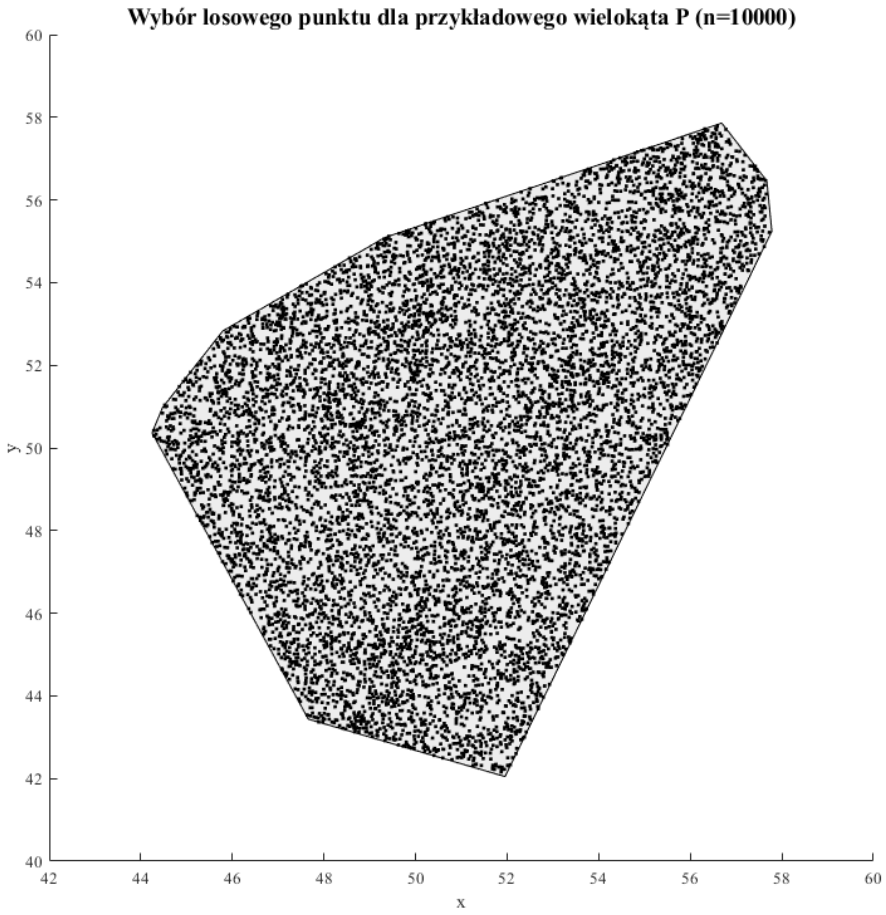
Rozrost drzewa w GAVB-RRT



Losowanie punktu (wielokąt)

Wybór punktu p_{next} wobec którego będzie odbywać się rozszerzanie drzewa jest ograniczone do wybranego regionu wielokąta P. Punkt losowany jest jednokrotnie w pojedynczej iteracji.

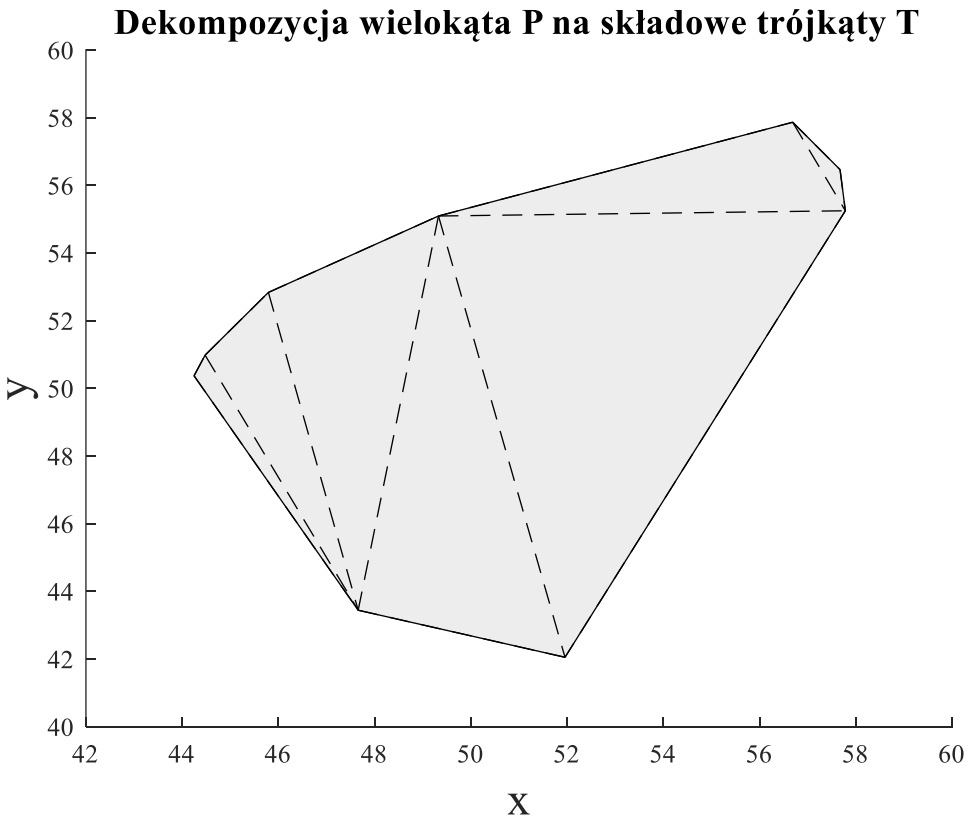




Dekompozycja P

Aby wybrać punkt x_{new} obszar P jest rozbijany na składowe trójkąty dzięki wykorzystaniu triangulacji Delunaya.

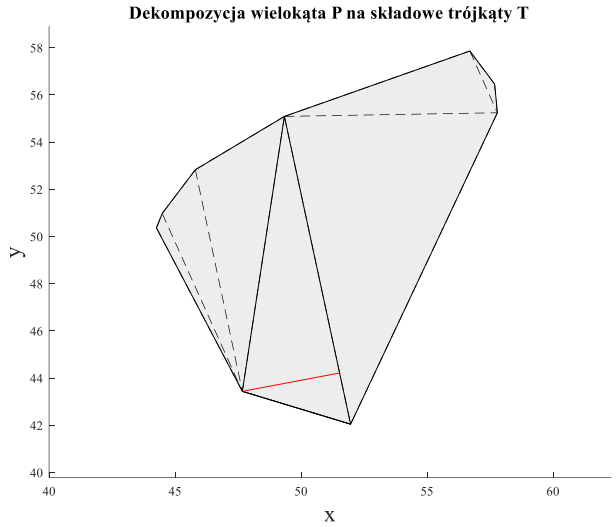
Suma obszarów poszczególnych trójkątów jest sumowana i normalizowana. Realizacja zmiennej losowej pozwala na wybranie jednego z trójkątów.

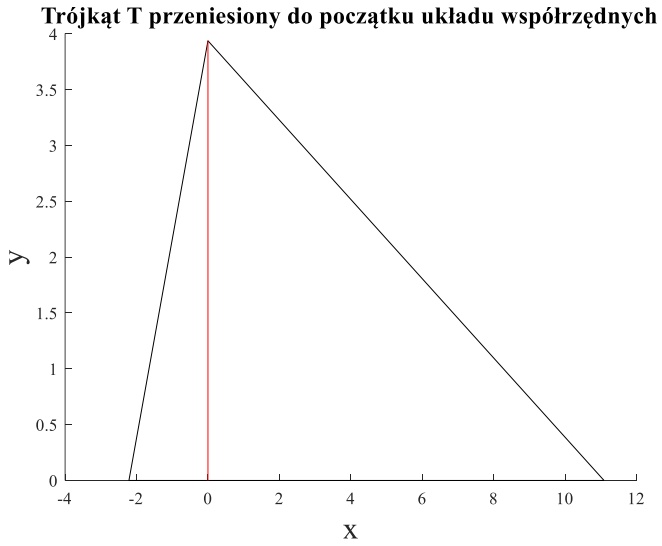


Trójkąt jednostkowy

Wybrany pojedynczy trójkąt jest sprowadzany do początku układu współrzędnych z wykorzystaniem operacji translacji T oraz rotacji R. Najdłuższy bok trójkąta znajduje się na osi x. Wysokość trójkąta powiązana z tym bokiem opisana jest odcinkiem OH [(0,0); (0,h)]. Skalowany jest następnie do trójkąta jednostkowego.

Wysokość dzieli trójkąt na dwa trójkąty prostokątne. Są rozpatrywane oddzielnie. Prawdopodobieństwo wylosowania punktu jest proporcjonalne do ich obszaru.





Losowanie punktu (trójkąt)

Trójkąty są konwertowane do trójkątów jednostkowych. Realizacja nowego losowego punktu

$$p_T = (p_{Tx}, p_{Ty})$$

w takim trójkącie odbywa się z wykorzystaniem następującego równania

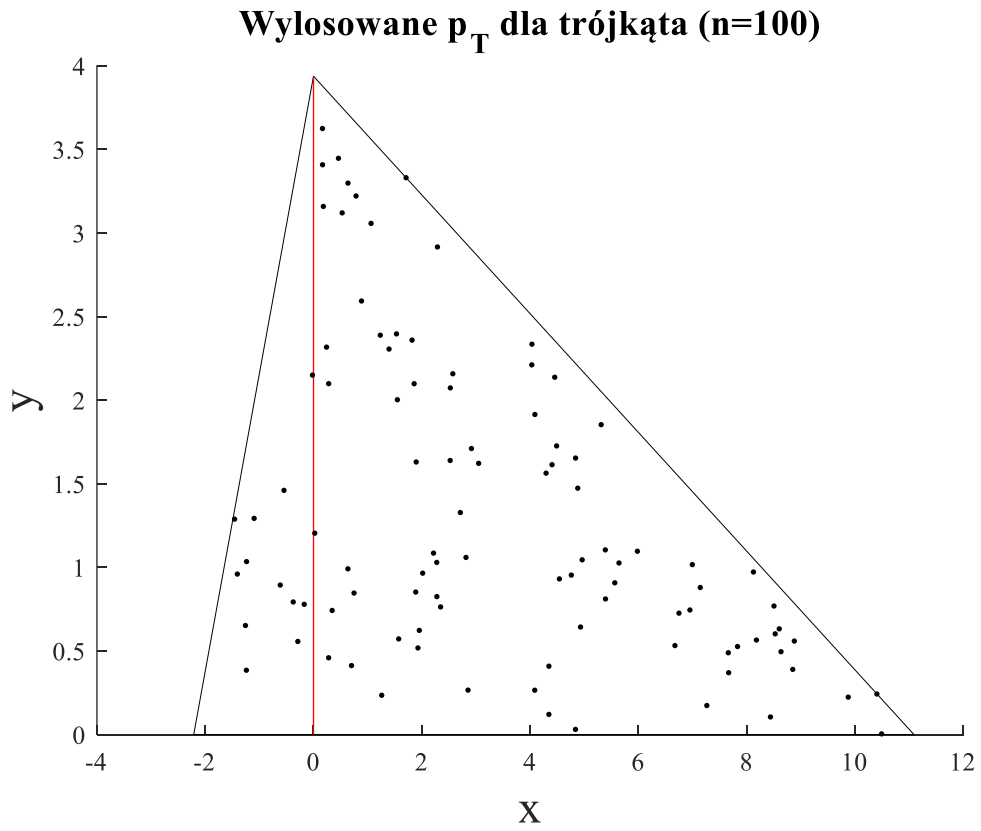
$$p_{Tx} = U(0,1)$$
$$p_{Ty} = U(0,1)$$

Przy czym dla $p_{Tx} + p_{Ty} > 1$

$$p_{Tx} = 1 - p_x$$
$$p_{Ty} = 1 - p_y$$

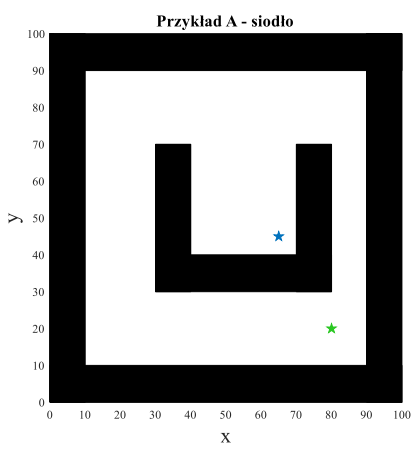
Punkt jest następnie mnożony przez długość w oraz wysokość h trójkąta. Powrót odbywa się poprzez odwrócenie operacji rotacji i translacji

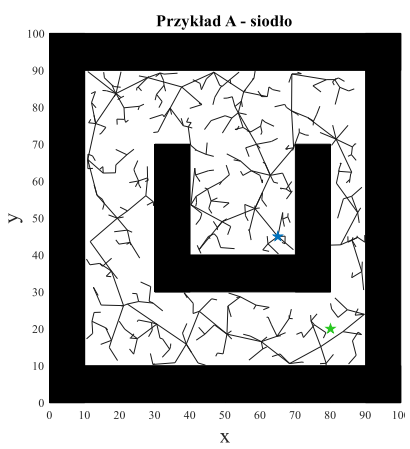
$$p_{new} = R^{-1} p_T + T$$

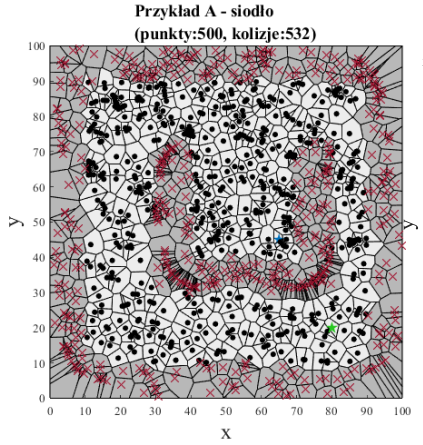


Przykładowe środowiska

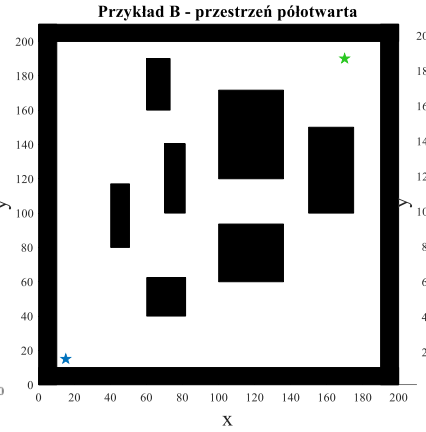
Przykład A (Siodło)

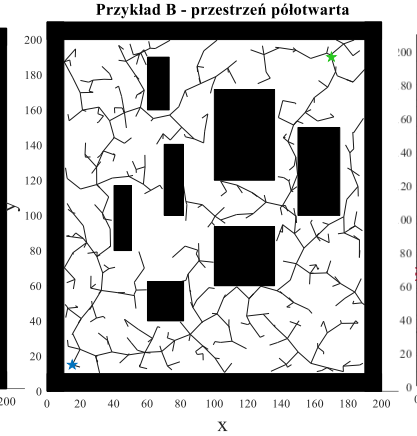


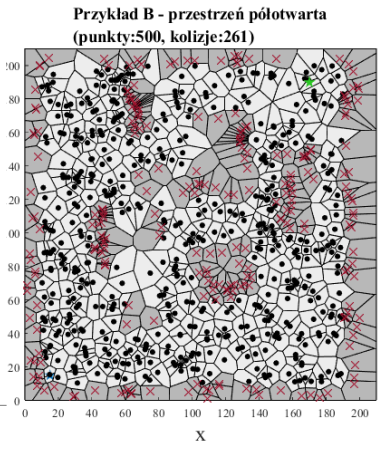




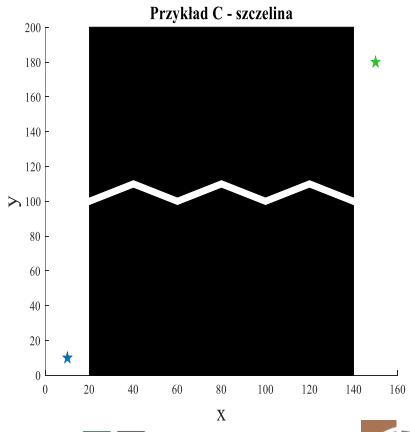
Przykład B (Przestrzeń półotwarta)

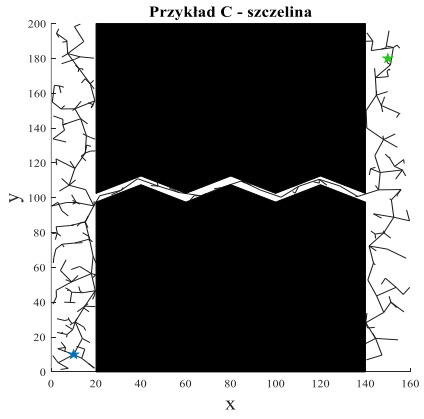


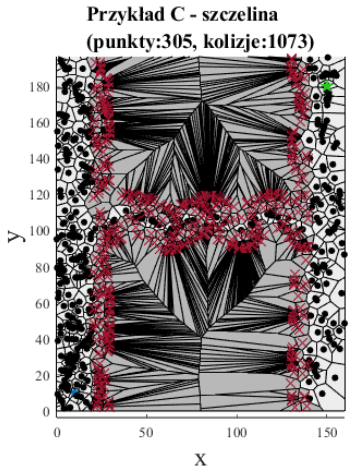




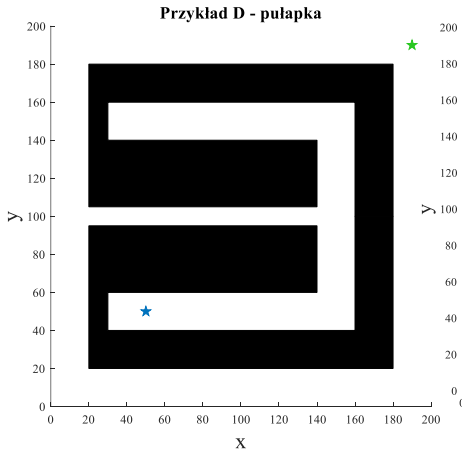
Przykład C (Szczelina)

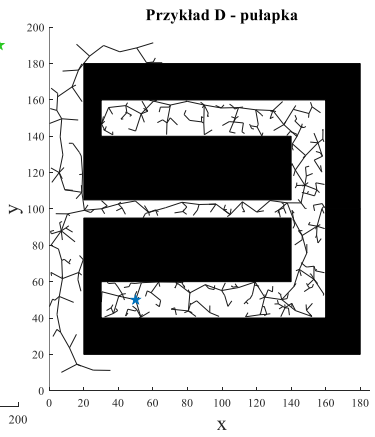


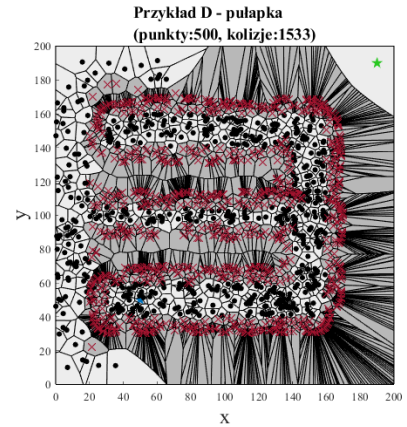




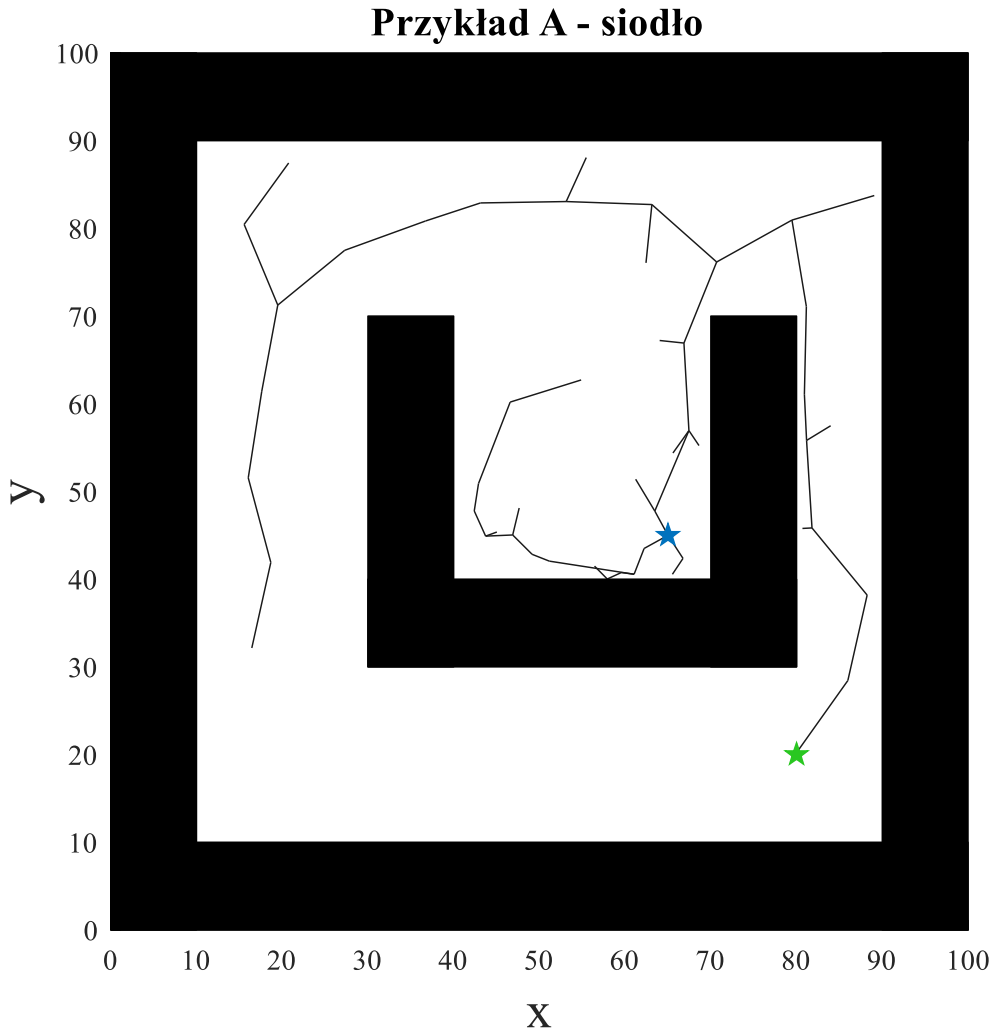
Przykład D (Pułapka)

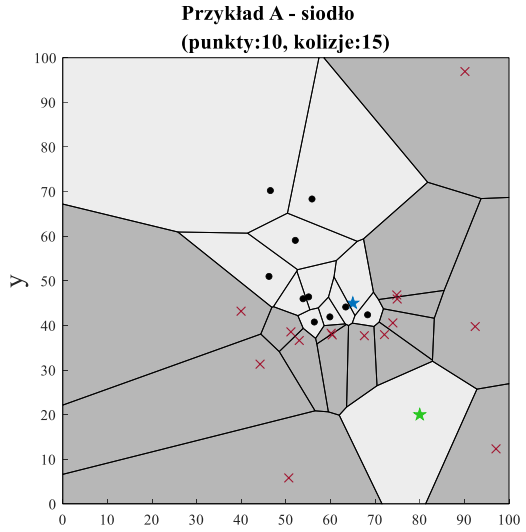


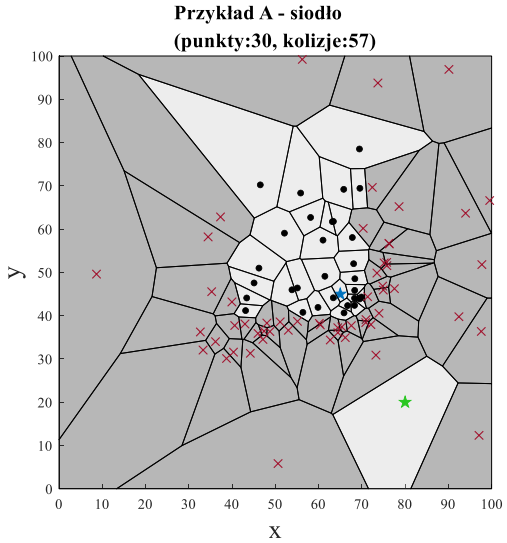


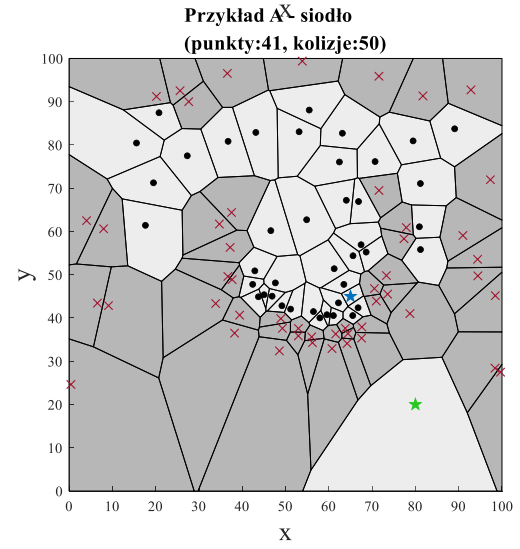


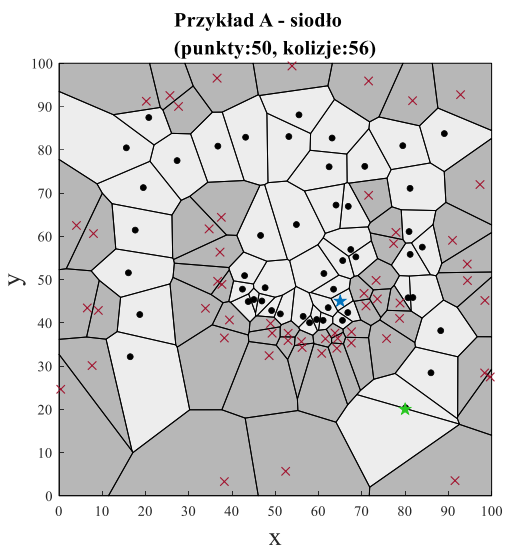
Przykład A (siodło)



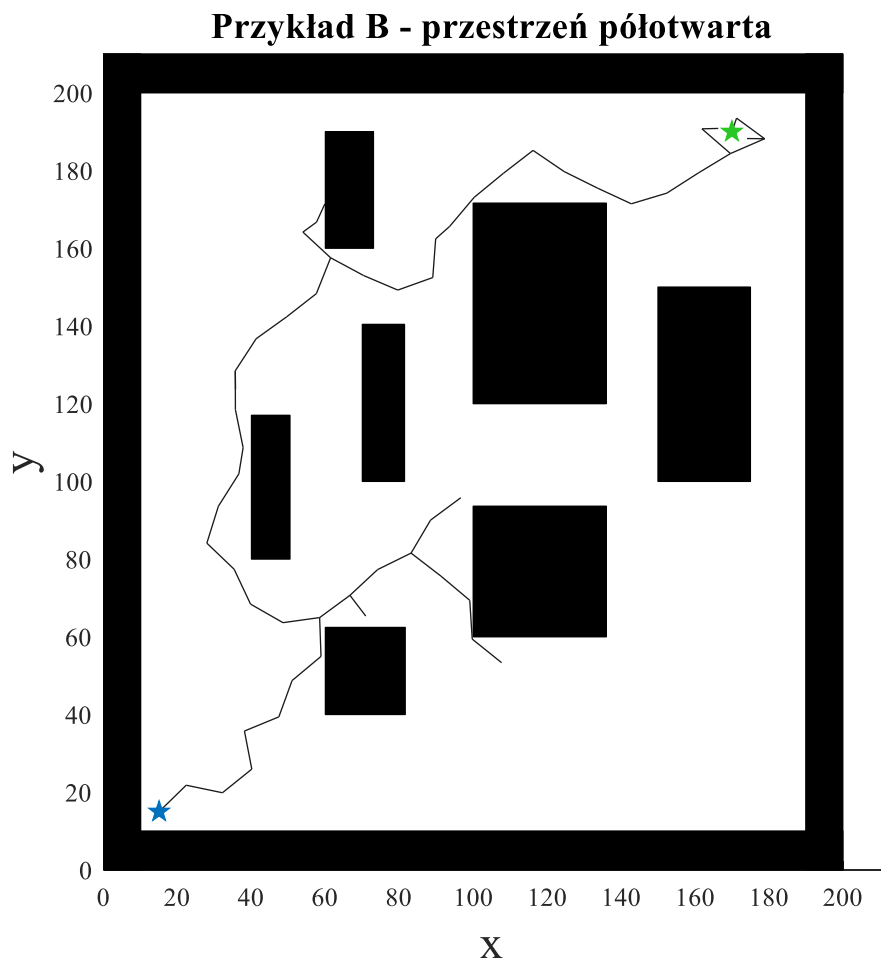


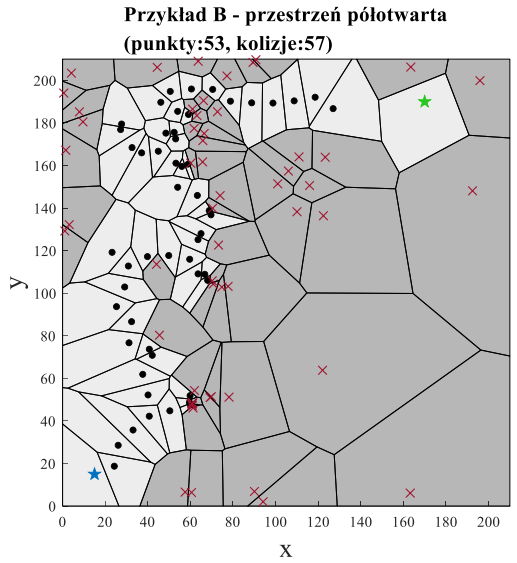


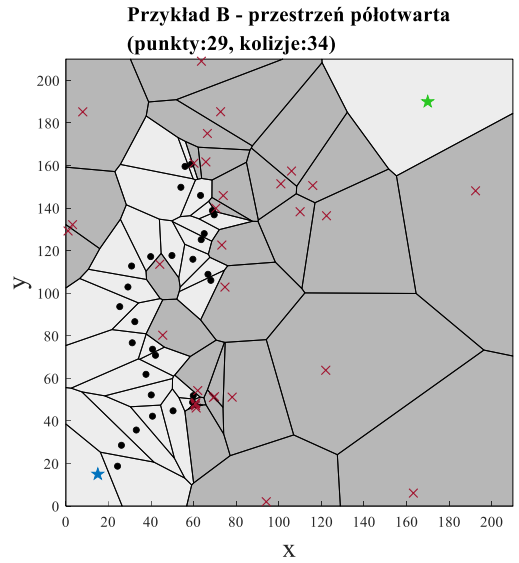


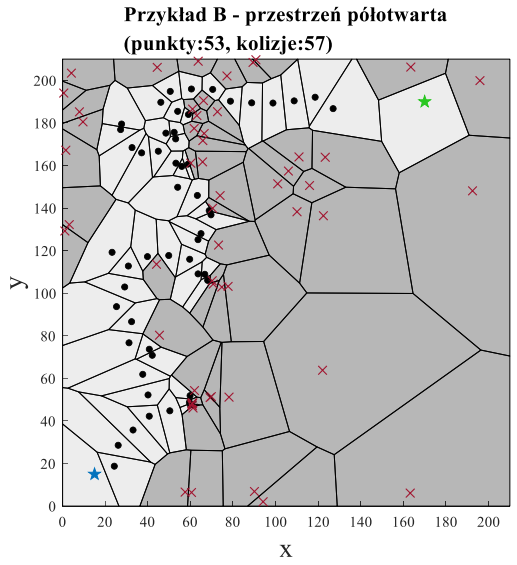


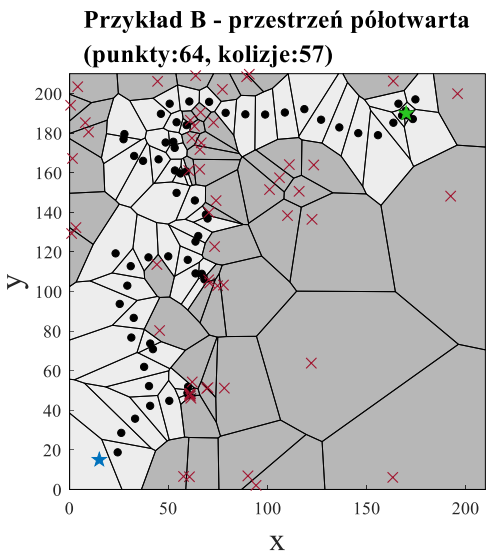
Przykład B (przestrzeń półotwarta)



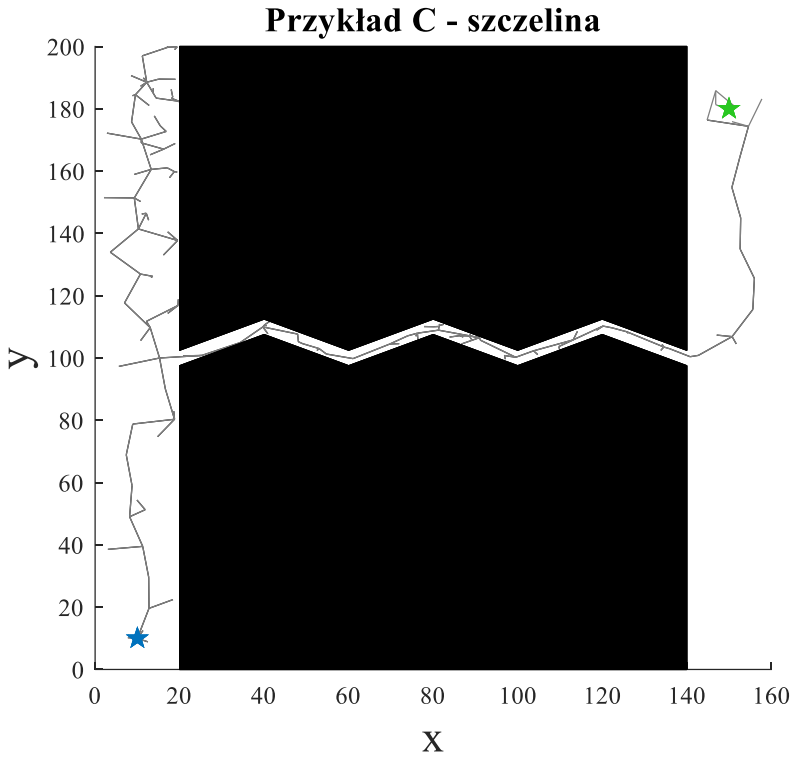


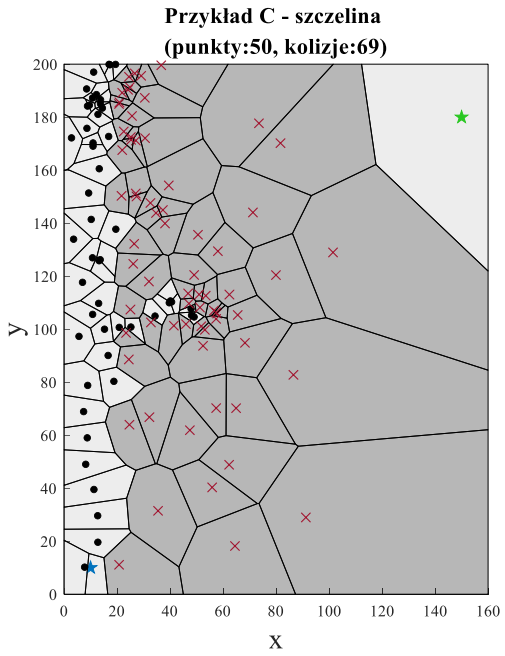


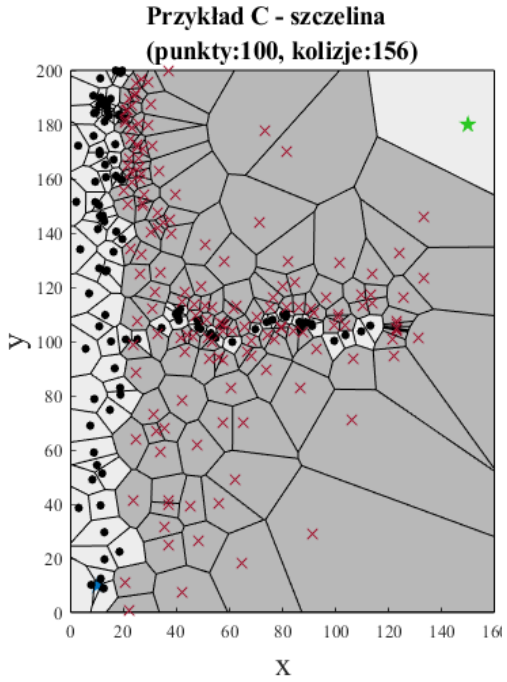


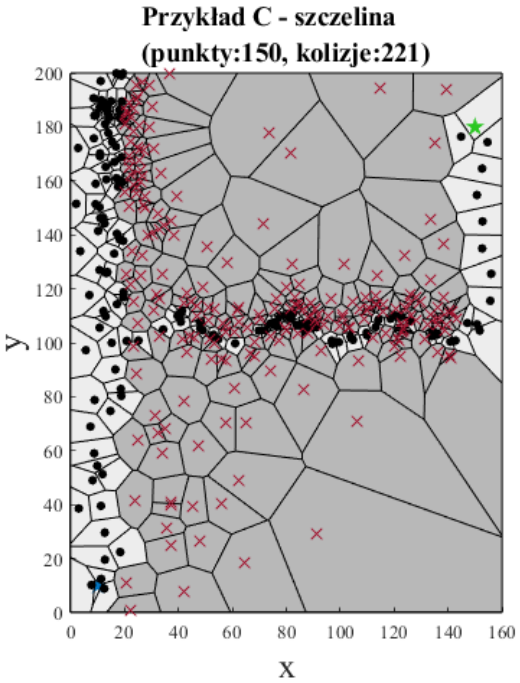


Przykład C (szczelina)

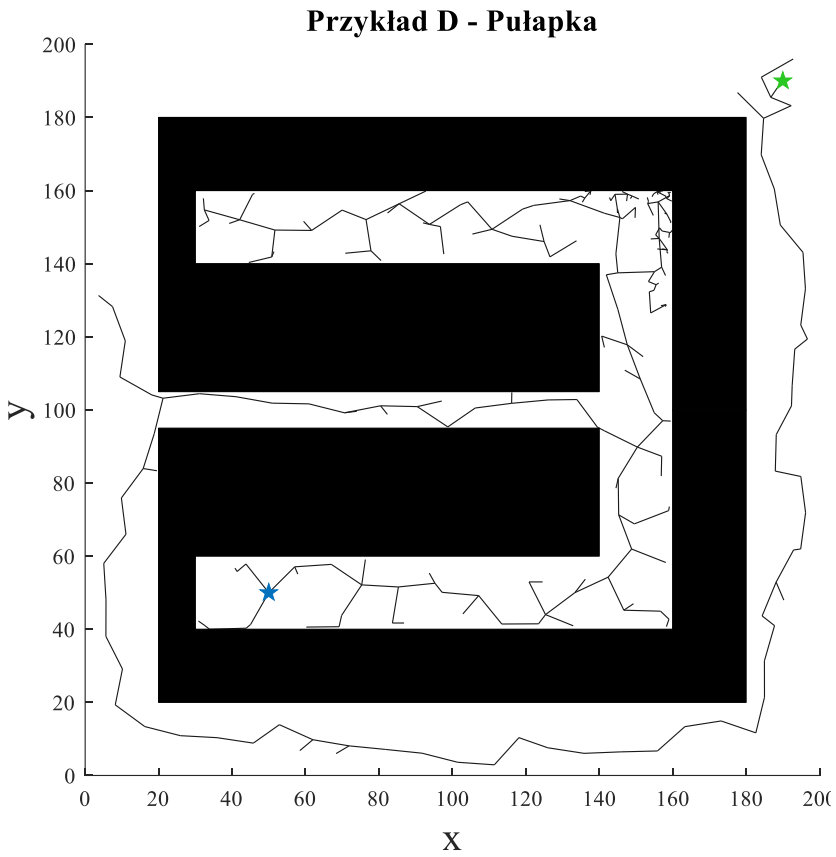


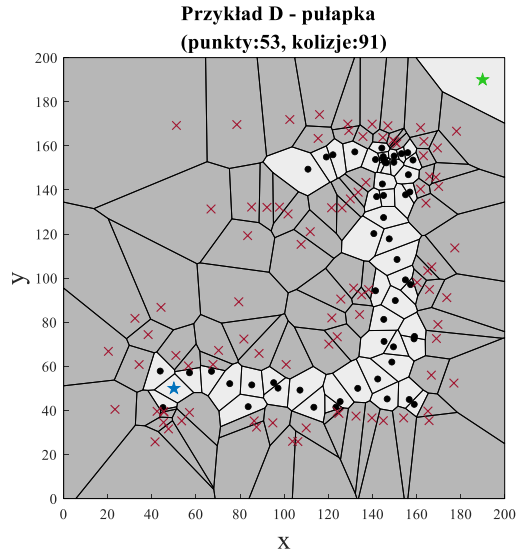


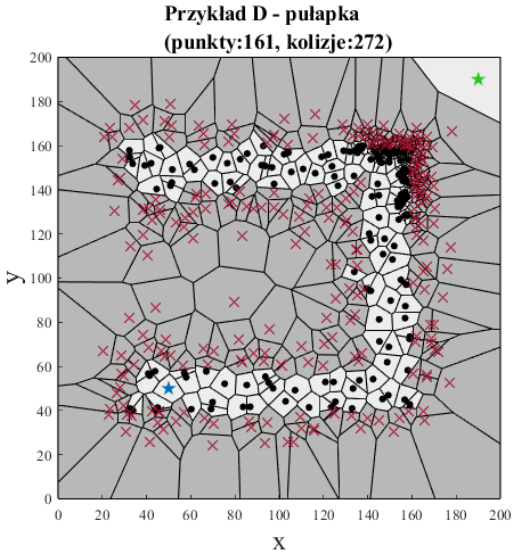


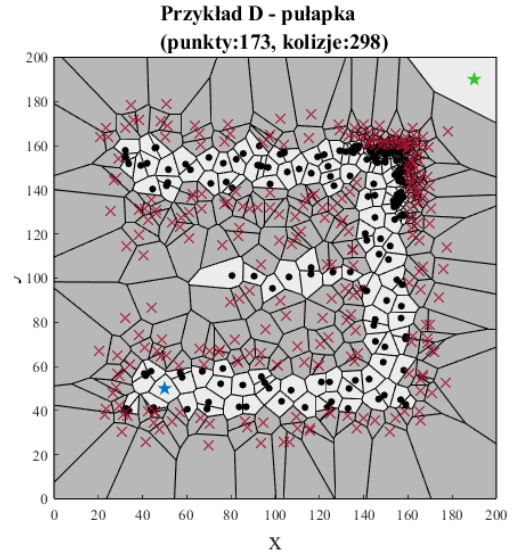


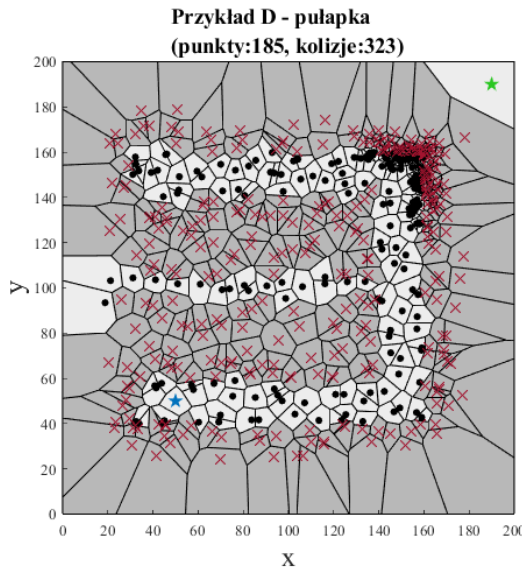
Przykład D (pułapka)

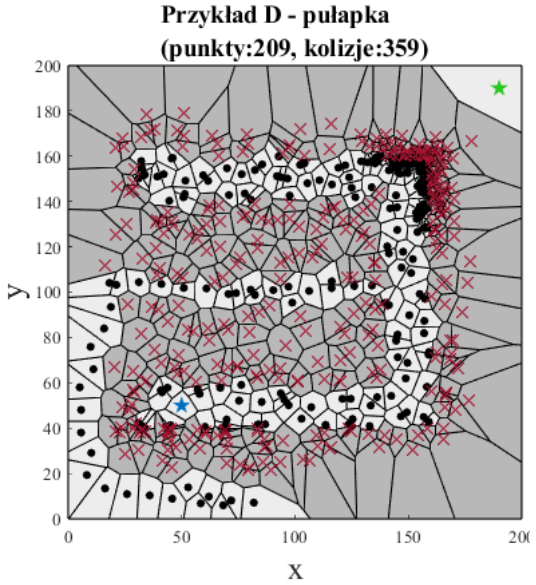


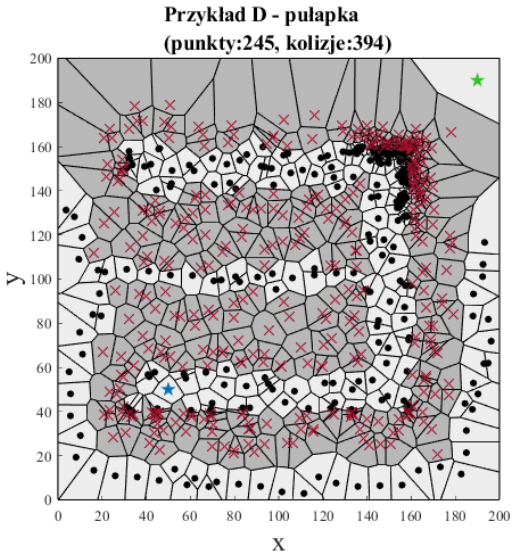






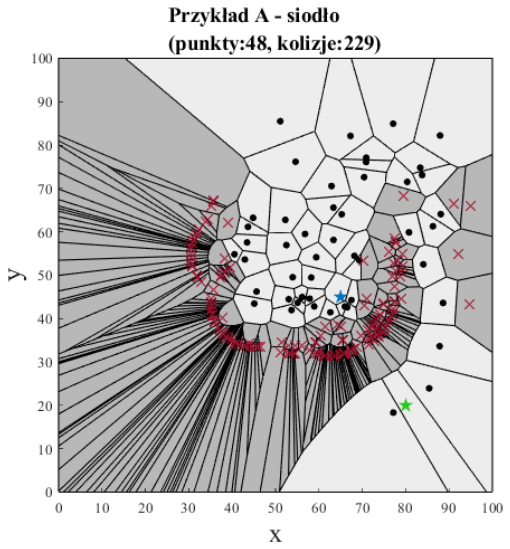


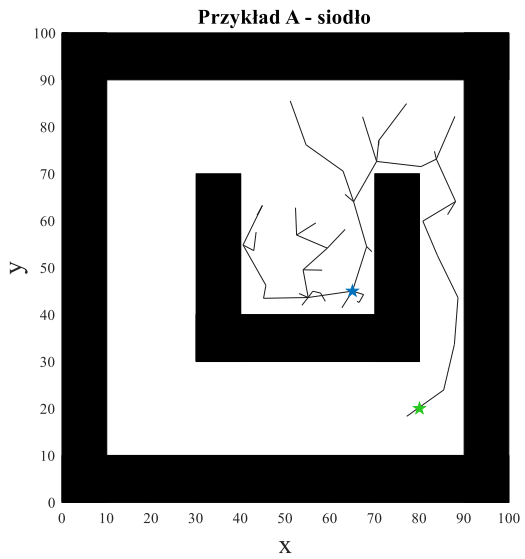




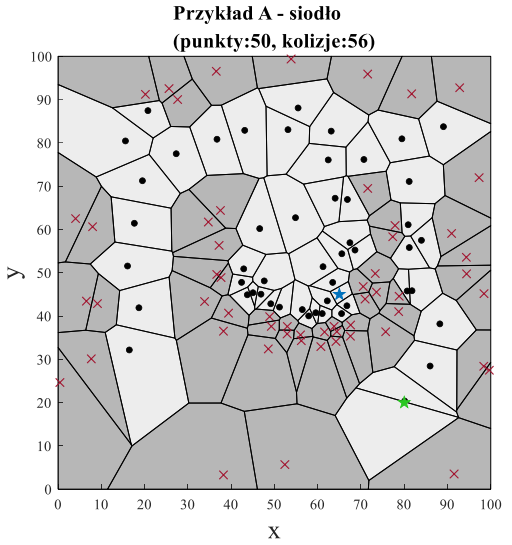
Przykład A – porównanie

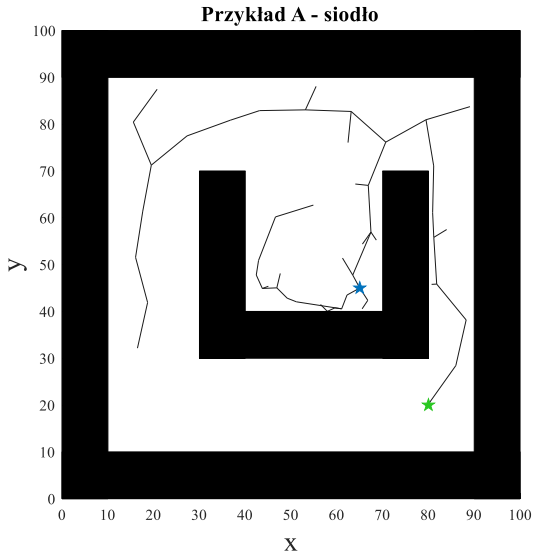
hRRT





GAVB-RRT

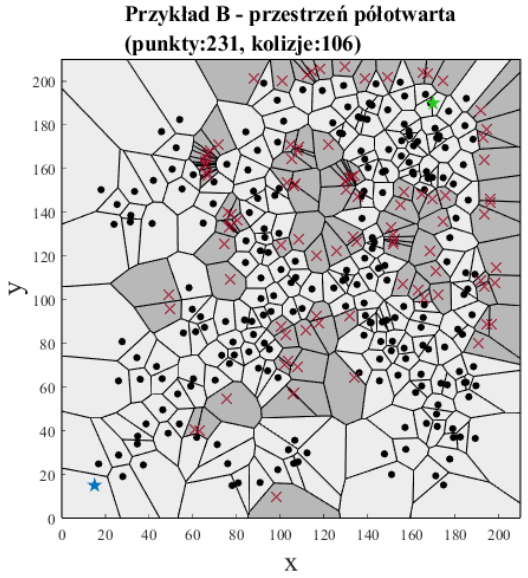


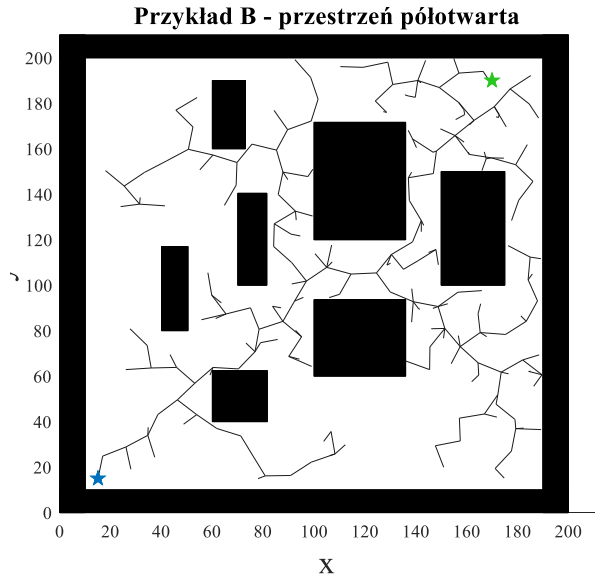


	hRRT	GAVB-RRT
Liczba węzłów	48	50
Liczba kolizji	229	56

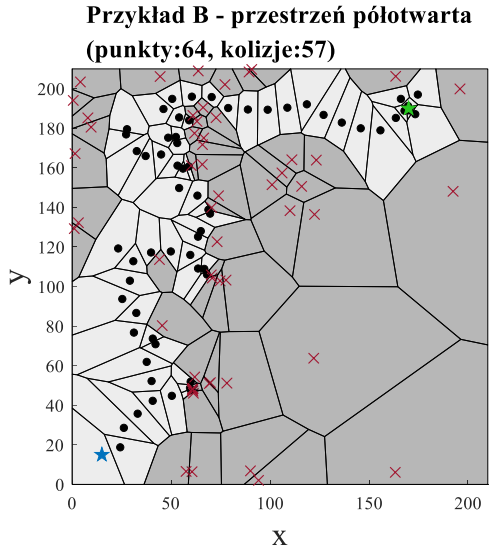
Przykład B - porównanie

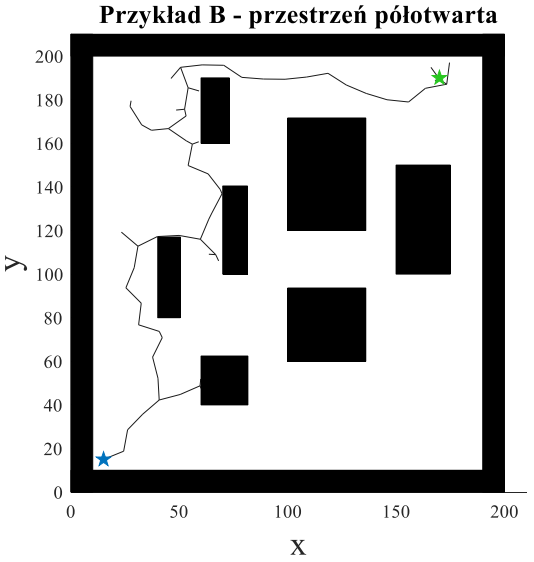
hRRT





GAVB-RRT

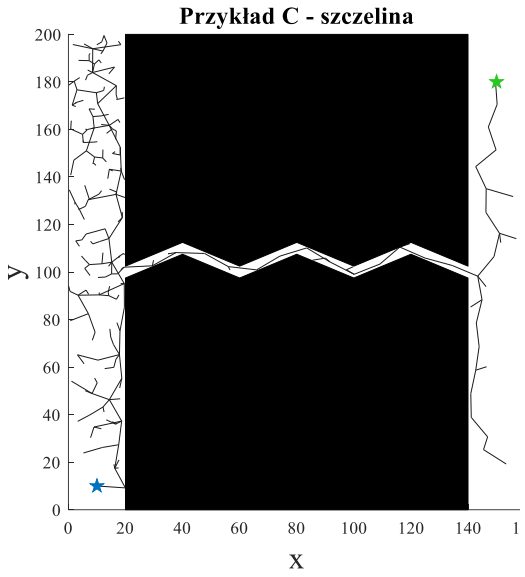


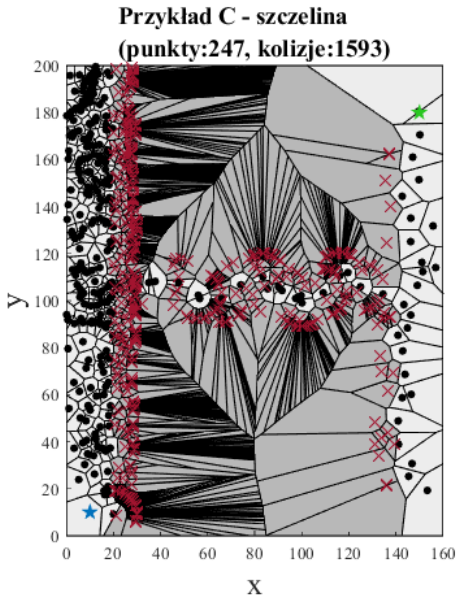


	hRRT	GAVB-RRT
Liczba węzłów	231	64
Liczba kolizji	106	57

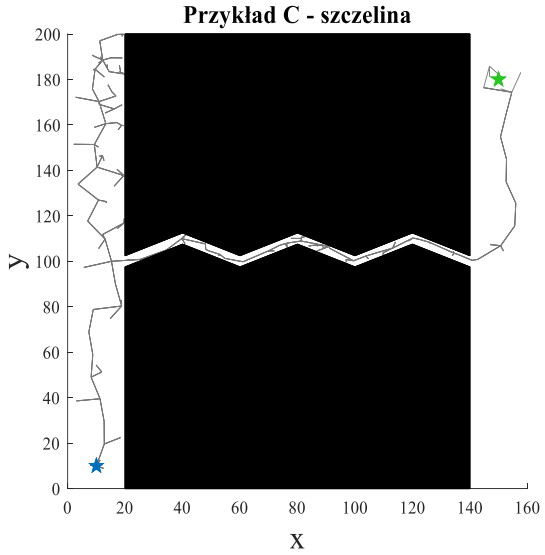
Przykład C - porównanie

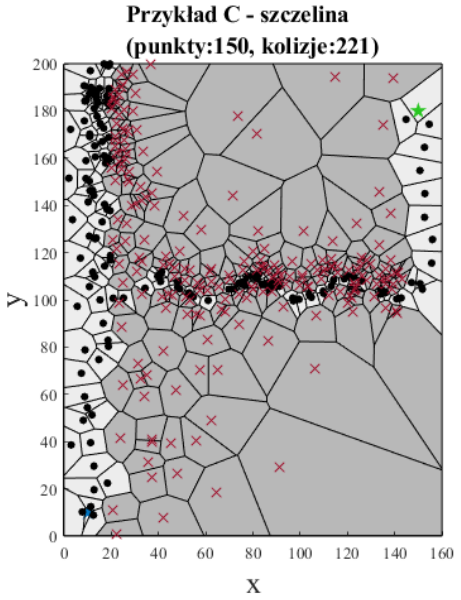
hRRT





GAVB-RRT

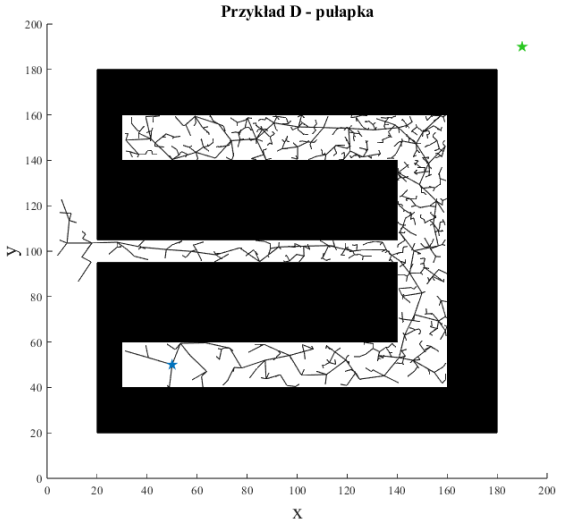


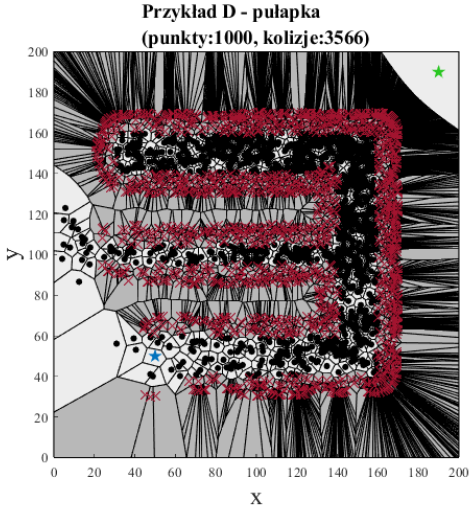


	hRRT	GAVB-RRT
Liczba węztów	247	157
Liczba kolizji	1593	221

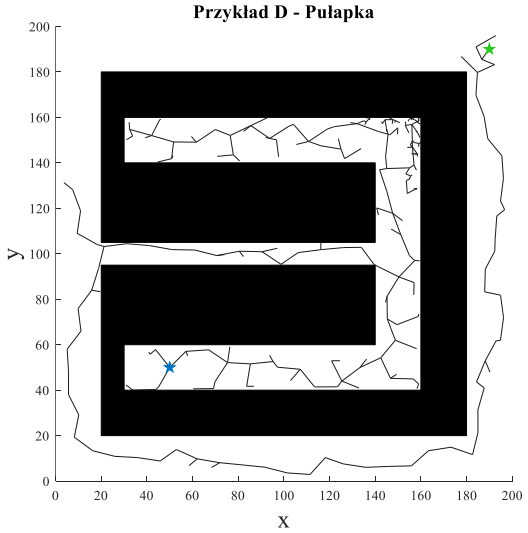
Przykład D – porównanie

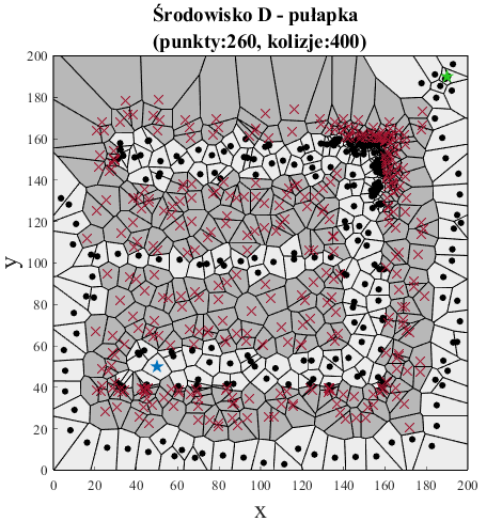
hRRT





GAVB-RRT

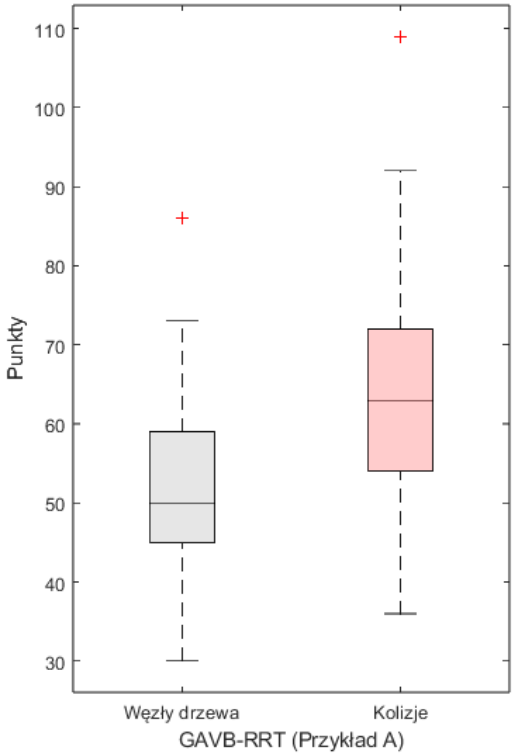


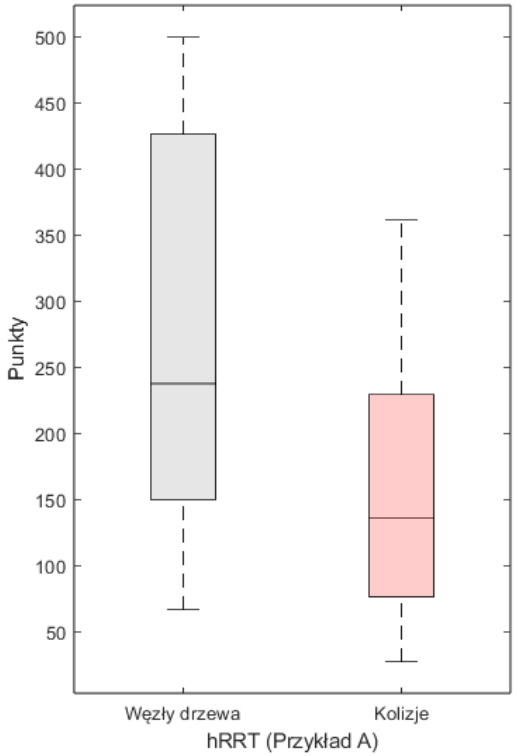


	hRRT	GAVB-RRT
Liczba węzłów	**1000**	260
Liczba kolizji	3566	400

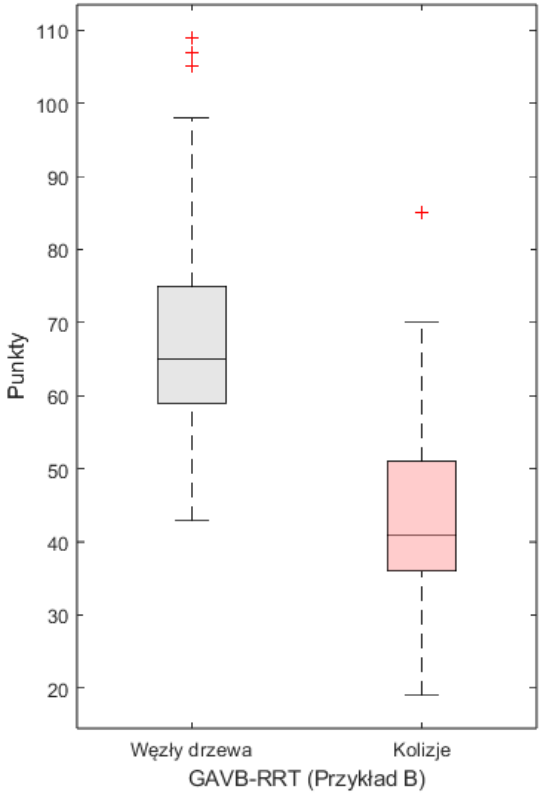
Porównanie metod dla wielu realizacji

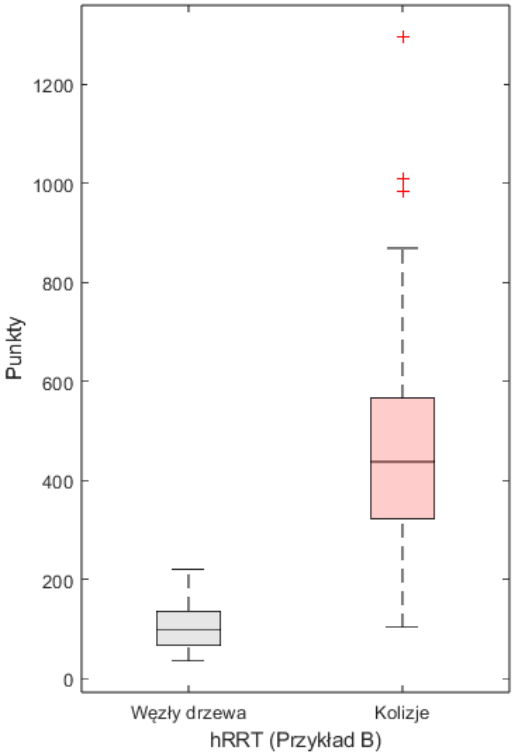
Przykład A (siodło)





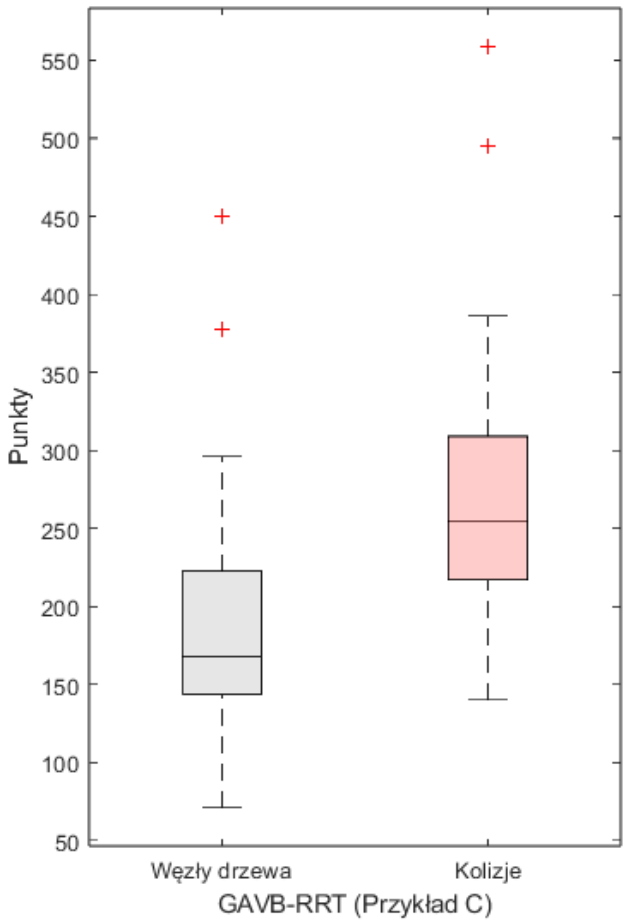
Przykład B (środowisko półotwarte)

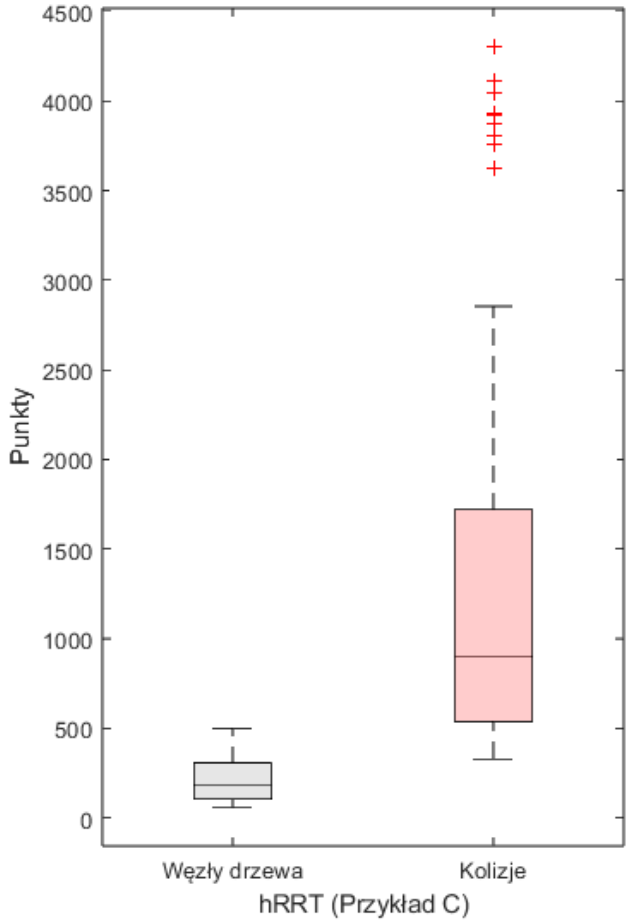




Porównanie metod dla wielu realizacji

Przykład C (szczelina)





Wnioski

- Opracowana metoda pozwala na osiągnięcie celu w mniejszej ilości kroków niż klasyczny algorytm RRT, a także heurystyczna wersja RRT (hRRT)
- Wykorzystanie informacji o przeszłych kolizjach pozwala na znaczące ograniczenie ich wpływu na eksplorację przestrzeni konfiguracyjnej
- Algorytm radzi sobie też w skomplikowanych przestrzeniach roboczych, warstwa eksploracyjna pozwala na wybrnięcie z minimów
- Powtarzalność wyników jest większa niż w porównywanym algorytmie hRRT

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Wydział Elektryczny

Dziękuję za uwagę