AKADEMIA GÓRNICZO-HUTNICZA
Wydział Informatyki, Elektrotechniki i Telekomunikacji

Projekt z przedmiotu Teoria Obliczeń i Złożoności Obliczeniowej II

Prowadzący: dr inż. Piotr Faliszewski
Autor: Alicja Salamon

1. Wstęp

Zadanie rozwiązałam dzięki wykorzystaniu algorytmu opartego przede wszystkim o przegląd wszystkich możliwości `-bruteforce`. Został on jednak wzbogacony o pewne dodatkowe funkcjonalności, których celem było skrócenie czasu wykonania dla większości przypadków.

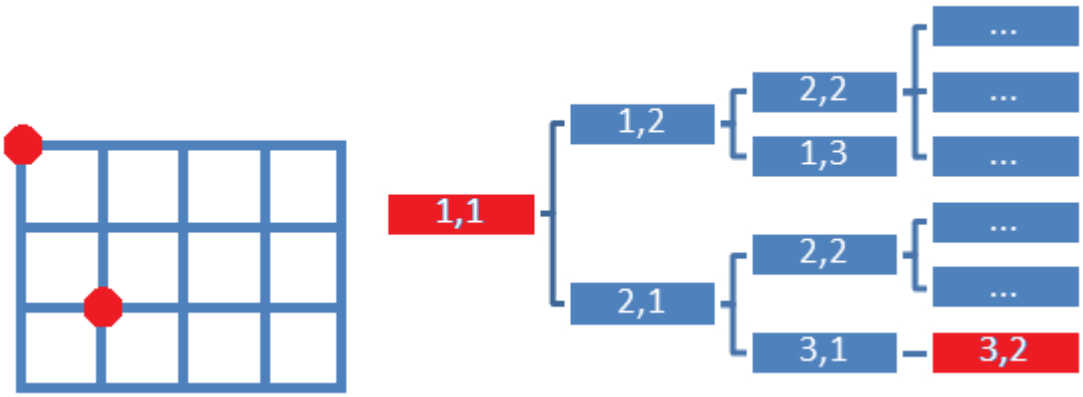
2. Opis rozwiązania

2.1 Algorytm DFS

Podstawą algorytmu jest przegląd wszystkich możliwości ruchów Agenta za pomocą rekurencyjnego przeszukiwania w głąb.

```
szukaj_drogi(Punkt a, Punkt b):
  if(a==b)
    nrAgenta++;
    agent=tablicaAgentow[nrAgenta]
    if(agent==null)
      return;
    else
      szukaj_drogi(agent.poczatek, agent.koniec);
    nrAgenta--;
  else
    for(Wektor w : MozliweWektoryRuchu)
      wynik.push_back(Trasa(Punkt(a,b), w));
      szukaj_drogi(a+w, b);
      wynik.pop_back();
```

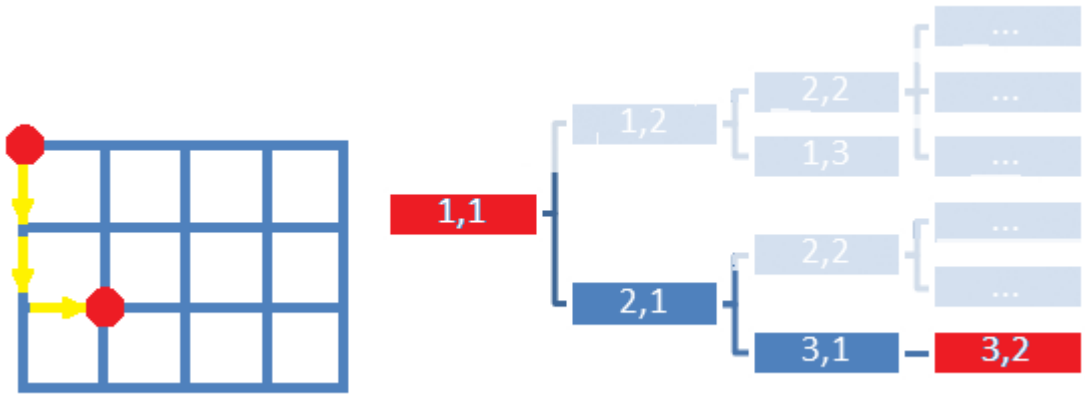
Poniższy przykład ilustruje, że jeśli jedyną odpowiednią drogą z punktu (1,1) do (3,2) okaże się ścieżka (1,1) →(3,2)→ (1,1) → (3,2) , to dla tego agenta zostanie zbudowane odpowiednie drzewo przeszukiwań możliwych ścieżek.



2.2 Sortowanie wektorów przesunięć

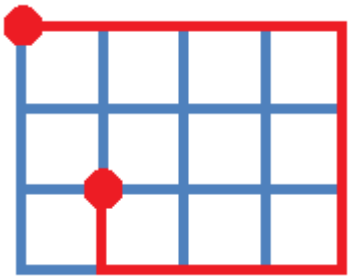
Ten prosty algorytm został zmodyfikowany poprzez dodanie sterowania wyborem kolejnych kierunków (północ, południe, wschód, zachód) zagłębiania się w drzewo przeszukiwań. Nie jest to zawsze ta sama, ustalona kolejność. Wektory są sortowane tak, aby zaczynać od tego, który wskazuje w kierunku punktu docelowego.

Na poniższym przykładzie widać sytuację (poprawna ścieżka to (1,1) →(3,2)→ (1,1) → (3,2)), w której podejście to znacznie ograniczyło wielkość drzewa.



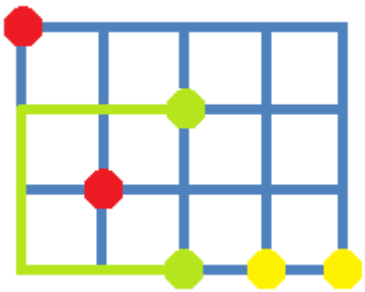
Oczywiście zdarzają się przypadki, kiedy wykonanie ruchu zbliżającego agenta do punktu przeznaczenia nie jest opłacalne, lecz generalnie poprawia to czas wykonania programu.

Przykład obok pokazuje sytuację, kiedy modyfikacja ta zwiększy rozmiar drzewa. Czerwonym kolorem zaznaczona jest jedyna ścieżka, która stanowić może rozwiązanie.



2.3 Sprawdzanie kolizji

Sprawdzenie, czy obrane ścieżki nie kolidują ze sobą (nie przecinają się) zostaje wykonane tak, żeby jak najwcześniej ograniczyć zagłębianie się niżej w rekurencyjne drzewo. Tzn. sprawdzenie kolizji nie jest dokonywane po znalezieniu wszystkich ścieżek łączących początek trasy z jej końcem, ale na każdym etapie zagłębiania się. Każdy ruch poprzedzony jest ustaleniem możliwych wektorów ruchów – nie prowadzących w wybrane już wcześniej punkty rozwiązania ani punkty początkowe i końcowe następnych agentów.



Ten przykład ilustruje jak bardzo można ograniczyć drzewo przeszukiwania. Po wyznaczeniu ścieżki zielonej, a przed wyznaczeniem ścieżki żółtej (i stosując opisane wcześniej sortowanie wektorów) droga ścieżki czerwonej nie wymaga żadnego rozbudowywania drzewa.