

Tomasz SZATKIEWICZ

POLITECHNIKA KOSZALIŃSKA, WYDZIAŁ MECHANICZNY, KATEDRA MECHANIKI PRECYZYJNEJ,
Raclawicka 15-17, 75-620 Koszalin

System wyznaczania ścieżek przemieszczeń dla grupy autonomicznych mobilnych robotów-agentów rozpoznawczo-inspekcyjnych

Dr inż. Tomasz SZATKIEWICZ

Autor jest adiunktem w Katedrze Mechaniki Precyzyjnej Wydziału Mechanicznego Politechniki Koszalińskiej. Specjalizuje się w problematyce wyznaczania

i optymalizacji trajektorii przemieszczeń w zautomatyzowanych systemach technologicznych. Ponadto, zajmuje się badaniami nad szeroko pojętymi systemami sztucznej inteligencji, a w szczególności sztucznymi sieciami neuronowymi, systemami agentowymi

i wykorzystaniem sztucznej inteligencji w zastosowaniach robotyki przemysłowej.



e-mail: tomasz.szatkiewicz@tu.koszalin.pl

Streszczenie

W artykule opisano oryginalny system wieloagentowy, którego podstawowym celem jest rozpoznanie, a następnie przeprowadzanie inspekcji nieznanego obszaru przy wykorzystaniu grupy autonomicznych mobilnych robotów-agentów. Wnioski z przeprowadzonych symulacji systemu pozwalają na rozpoznanie obszaru wiedzy, dotyczącego szczególnych aspektów systemów wieloagentowych takich jak min: metody ustalania ścieżek przemieszczeń agentów mobilnych czy wpływ liczby agentów na globalną efektywność systemu.

Słowa kluczowe: systemy agentowe, inspekcja obszaru, rozpoznawanie obszaru, ustalanie trajektorii przemieszczeń.

A system of path determination for a group of autonomous, mobile inspection robots-agents

Abstract

The paper describes an original multi-agent system [1, 2, 10, 11] whose primary purpose is to recognize and, subsequently, to inspect continuously a limited area. The agents are autonomous mobile robots, equipped with a set of sensors that are capable of scanning the examined area and collecting information on significant features. The main decision problem necessary to be solved in the system is to determine the paths for the robotic agents in such a way that the area is recognized in the shortest time, and the inspection process is the most effective from the point of view of the state of knowledge about the significant features of the examined area. The agents should act collectively in such a way as to balance their load and, hence, to optimize the system performance. In the paper there are described in detail the consecutive steps of the algorithm (Fig. 1), and, in particular, the methodology for determining the direction of motion of robotic agents (Subsections 5.3 and 5.4). The presented multi-agent system was implemented in the Webots environment [4]. Educational robots of the e-puck class were used as robots-agents (Fig. 2), [3]. Implementation of the system made it possible to perform a series of experiments which allowed drawing interesting conclusions (Section 6) on the effectiveness of the system in achieving its primary objectives. At the end of the paper the experiment is illustrated (Fig. 7), and also the growth chart of recognition of the area for each robotic agent is presented (Fig. 8).

Keywords: multi-agent systems, movement path determination, trajectory optimization, field inspection, field recognition.

1. Wstęp

Systemy wieloagentowe znajdują szerokie zastosowania w istotnych dziedzinach wiedzy i inżynierii i są obecnie przedmiotem badań wielu ośrodków naukowych na całym świecie [10, 11]. Możliwość zastosowania zespołów mobilnych robotów realizujących wspólne zadania w różnych aplikacjach związanych min. z przemysłem, transportem, usługami, spowodowała konieczność opracowania algorytmów, realizujących specyficzne zadania związane z funkcjonowaniem zespołów robotów (ang. swarm) [8, 9]. Do tych szczególnych zadań należą między innymi takie problemy jak: ustalenie skutecznych, efektywnych i bezkolizyjnych trajektorii przemieszczeń dla często dużych zespołów autonomicznych robotów mobilnych, działających we wspólnym obszarze [12, 13], optymalny przydział zadań dla poszczególnych robotów-agentów, ustalenie protokołu komunikacyjnego, przesyłania, przetwarzania i przechowywania danych.

W artykule autor opisuje metodę ustalania ścieżek dla grupy autonomicznych, mobilnych robotów inspekcyjnych wyposażonych w sensory. Głównym celem zespołu robotów jest w pierwszym etapie - rozpoznanie nieznanego i ograniczonego obszaru działania, a następnie - jego stała inspekcja, mająca na celu utrzymanie jak najbardziej aktualnych informacji o istotnych parametrach badanego obszaru. Opisana w artykule metoda ustalania ścieżek stanowi połączenie i modyfikację metod pola potencjałów, opisaną szerzej w pracach [5, 6, 7], oraz techniki podziału podobszarów działań każdego robota przez generowany diagram Voronoi [5].

2. Systemy rozpoznania i inspekcji obszaru

Rozpoznanie i inspekcja obszaru, obejmuje zestaw czynności, których celem jest uaktualnienie stanu wiedzy na temat jego szczególnych cech. Rozpoznanie i inspekcji dokonuje autonomiczny, mobilny robot-agent, wyposażony w zestaw czujników lub systemów, służących do

pomiaru określonych cech badanego obszaru. Agent inspekcyjny może działać pojedynczo lub w grupie i wtedy taka grupa stanowi system wieloagentowy [1, 2].

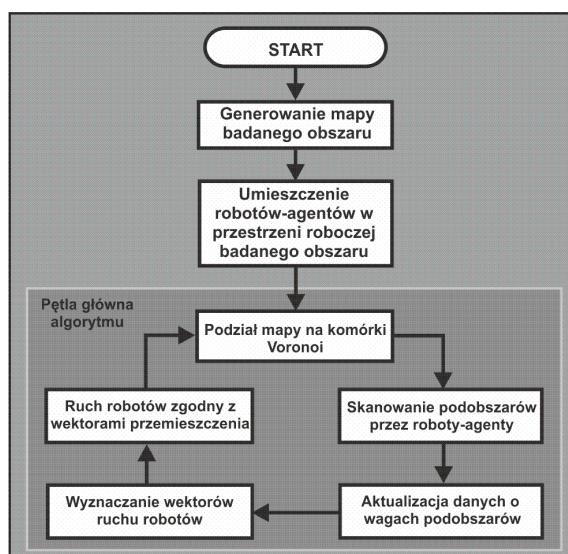
Jeżeli głównym zadaniem projektowanego systemu agentowego jest rozpoznanie i inspekcja określonego obszaru poprzez zebranie lub uaktualnienie informacji o określonych cechach obszaru, każdy z agentów powinien być wyposażony w:

- system zbierania i ewentualnie przetwarzania informacji o istotnych cechach badanego obszaru,
- system lokalizacji względem badanego obszaru i innych agentów w obrębie systemu,
- system gromadzenia i przesyłania danych między agentami i centralnym systemem gromadzenia i przetwarzania danych,
- cechy konstrukcyjne umożliwiające ruch po badanym obszarze,
- zestaw algorytmów stanowiących indywidualną „inteligencję” robota-agenta, rozwiązujących istotne problemy decyzyjne, a w szczególności ustalających kierunek ruchu robota w celu dokonania kolejnego pomiaru istotnych cech badanego obszaru.

Poddawany rozpoznaniu i inspekcji obszar jest podzielony na podobszary, których wielkość odpowiada zasięgowi czujników zbierających informacje, zainstalowanych na robotach-agentach. W każdym kroku działania systemu następuje skanowanie odpowiedniego podobszaru zajmowanego aktualnie przez agenta inspekcyjnego a następnie jego przemieszczenie do innego podobszaru. Cel działania systemu jest dwuetapowy: po pierwsze, system dąży do rozpoznania badanego obszaru, czyli przynajmniej jednokrotnego zebrania informacji ze wszystkich podobszarów składowych badanego obszaru, po drugie, po osiągnięciu celu pierwszego, system dąży do utrzymania jak najbardziej aktualnej skumulowanej informacji ze wszystkich podobszarów składowych badanego obszaru.

3. Etapy działania systemu

Wieloagentowy system rozpoznania i inspekcji obszaru działa według następującego schematu kolejno realizowanych po sobie zadań:



Rys. 1. Algorytm działania wieloagentowego systemu rozpoznania i inspekcji obszaru

Fig. 1. Algorithm for field recognition and inspection multiagents system

4. Generowanie mapy obszaru

W pierwszym kroku algorytmu generowana jest mapa badanego obszaru, która poddawana jest podziałowi na n sześciokątnych podobszarów (patrz rys. 3). Wielkość sześciokątnych podobszarów, dyskretyzujących badany obszar zależy od:

- zasięgu sensorów robotów – w taki sposób, że zasięg sensorów powinien pokrywać cały sześciokątny podobszar,
- maksymalnej prędkości ruchu robotów – w taki sposób, aby robot w pojedynczym kroku algorytmu mógł przemieścić się do centrum sąsiedniego podobszaru w stosunku do tego, na którym się aktualnie znajduje.

Każdemu z podobszarów przypisywana jest jedna z dwóch możliwych klas, określających jej stan względem kryterium wiedzy o badanych właściwościach obszaru:

- *podobszar nierozpoznany* – podobszar, który nie został jeszcze poddany inspekcji przez robota-agenta,
- *podobszar rozpoznany* – podobszar, który przynajmniej raz został poddany inspekcji w poprzednich krokach realizowanego procesu.

Dodatkowo, dla każdego z n podobszarów definiuje się wagę w_n , opisującą stopień aktualności informacji o szczególnych cechach poddawanych inspekcji, której wartość zmienia się zgodnie z przyjętą funkcją, różną w zależności od tego, czy podobszar należy do klasy *podobszar nierozpoznany* lub *podobszar rozpoznany*. Odpowiednio dla całego badanego obszaru definiuje się jego wagę skumulowaną W_s , której wartość jest średnią z wartości wag podobszarów w_n . Wartości wag w_n są wykorzystywane do wyznaczania wektora kierunku ruchu robota w późniejszej fazie działania systemu. Początkowa wartość wagi w_n każdego podobszaru jest jednakowa, niezależnie od jego aktualnej klasy i wynosi 1, jednak w każdym cyklu działania algorytmu wartości wag w_n zmieniają się zgodnie z funkcją przypisaną dla danej klasy podobszaru.

5. Umieszczenie agentów – robotów w przestrzeni roboczej

W badanym obszarze umieszcza się pewną liczbą robotów-agentów inspekcyjnych, wyposażonych w sensory umożliwiające zbieranie informacji z poszczególnych podobszarów składowych, system przesyłania danych i pozycjonowania, oraz mechanizm umożliwiający przemieszczenie robota-agenta. Liczba robotów-agentów ma istotny wpływ na efektywność systemu rozpoznania i inspekcji i teoretycznie powinna być jak największa, jednak można wykazać, że istnieje pewien poziom nasycenia agentami, powyżej którego system traci efektywność. Wyznaczenie tego poziomu jest przedmiotem osobnych badań. W przeprowadzonym eksperymencie wykorzystano roboty mobilne klasy e-puck [3], wyposażone w szereg sensorów i interfejs komunikacyjny (rys. 2).



Rys. 2. Robot e-puck wraz z sensorami i głównymi podzespołami
Fig. 2. E-puck bot with main sensors and subsystems

6. Pętla główna algorytmu

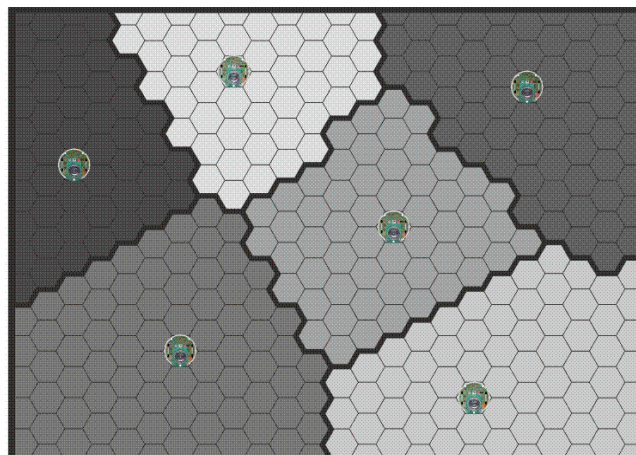
W kolejnych krokach pętli głównej algorytmu, każdy z robotów-agentów inspekcyjnych skanuje obszar składający się ze wszystkich sąsiednich podobszarów w stosunku do tego, na którym się aktualnie znajduje. Algorytm wykonywany jest dwuetapowo. W pierwszym etapie rozpoznania, głównym celem robotów-agentów jest przynajmniej jednokrotne skanowanie wszystkich podobszarów składowych, czyli rozpoznanie całego obszaru roboczego. Po zakończeniu tego etapu znane są dane z inspekcji całego obszaru roboczego. Na tym etapie system może zakończyć działanie lub, może przejść do etapu inspekcji, w którym celem robotów – agentów jest ciągle zbieranie informacji o istotnych cechach badanego obszaru, realizowane w taki sposób aby minimalizować wartość współczynnika W_s , który określa poziom aktualności informacji o istotnych (skanowanych) cechach całego obszaru. Głównym problemem decyzyjnym w opisanej strategii jest problem wyznaczania kierunku ruchu robotów w taki sposób aby w sposób ciągły minimalizować wartość współczynnika W_s badanego obszaru.

6.1. Podział mapy na sektory Voronoi'a

W pierwszym kroku pętli głównej algorytmu dokonuje się podziału mapy badanego obszaru na komórki Voronoi, tworzone na podstawie aktualnego położenia robotów-agentów (rys. 3).

Do wyznaczenia przynależnych każdemu robotowi obszarów Voronoi, można zastosować jedną z dwóch metod:

- algorytm Fortune'a (o złożoności obliczeniowej $O(n \log n)$, gdzie n – liczba robotów) i metrykę euklidesową.
- algorytm przeglądu zupełnego (o złożoności obliczeniowej $O(mn)$, gdzie m – liczba robotów i n – liczba sześciokątnych podobszarów), wyznaczający przynależność poszczególnych sześciokątnych podobszarów do obszarów Voronoi na podstawie ich geometrycznych odległości od robotów-agentów.



Rys. 3. Podział badanego obszaru na komórki Voronoi
Fig. 3. Division of the investigated field into Voronoi cells

Obydwa algorytmy wymagają zastosowania dodatkowych procedur naprawczych, ustalających przynależność poszczególnych sześciokątnych podobszarów, w przypadku, gdy częściowo należą do więcej niż jednego obszaru Voronoi.

Podział ten ma na celu wyznaczenie podobszarów działań dla każdego z robotów-agentów w aktualnym kroku algorytmu. Każdy z robotów-agentów, w późniejszym etapie algorytmu ustala kierunek ruchu na podstawie współczynników wag w_n podobszarów należących do przedzielonej mu komórki Voronoi.

6.2. Skanowanie podobszarów przez roboty

W każdym kroku pętli głównej algorytmu roboty-agenty skanują wszystkie podobszary sąsiadujące z tym, na którym się aktualnie znajdują. Operacja skanowania skutkuje zmianą wag w_n lub też klas, do których należą sąsiednie podobszary. W przypadku gdy podobszar należy do klasy *podobszaru nierozpoznanego*, po jego pierwszym skanowaniu i stosownej zmianie wagi w_n , klasa podobszaru zmienia się na *podobszar rozpoznany*.

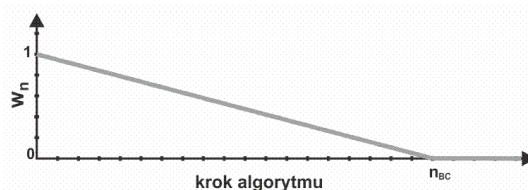
Wielkość podobszarów dzielących przestrzeń całego badanego obszaru powinna być tak dobrana, aby zasięg skanerów badanych cech pozwalał na skan całego podobszaru składowego.

Pojęcie skanowania jest tutaj pojęciem umownym i jest uogólniając rozumiane jako operacja zbierania informacji o cechach badanego obszaru istotnych dla użytkownika systemu.

6.3. Aktualizacja danych o podobszarach i wartości wskaźników

Na tym etapie, który następuje bezpośrednio po skanowaniu podobszarów przez roboty-agenty, dokonywana jest aktualizacja współczynników wag w_n wszystkich podobszarów zgodnie z odpowiednią funkcją zmiany wagi przynależną do klasy podobszaru.

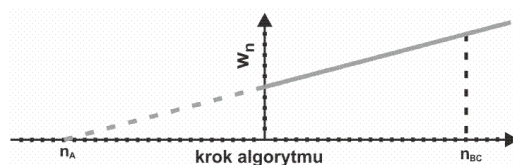
Wartości współczynników wag w_n dla podobszarów należących do klasy *podobszarów rozpoznanych* wyznacza się stosując malejącą funkcję liniową (rys. 4). Wartości wag w_n dla podobszarów rozpoznanych maleją w trakcie następowania po sobie kolejnych faz, w których nie nastąpiło skanowanie danego podobszaru, w przedziale od 1 do 0, gdzie 0 jest osiągnięte po n_{bc} krokach pętli głównej. Przyjęta wartość n_{bc} określa poziom szybkości redukcji wartości wag podobszarów i jest liczbą kroków, po której informacje o *podobszarze rozpoznany* uznawane są za nieaktualne (można zmienić ich klasę na *podobszar nierozpoznany*). Wartość współczynnika n_{bc} należy dobierać uwzględniając wielkość badanego obszaru, liczbę podobszarów i liczbę robotów-agentów.



Rys. 4. Postać funkcji zmiany wagi w_n dla podobszarów badanych
Fig. 4. Form of the weight w_n transition function for known subfields

Waga podobszarów rozpoznanych w_n nie może być mniejsza niż 0, stąd też każde $w_n < 0$ jest traktowane jako $w_n = 0$. W przypadku, gdy podobszar zostanie ponownie przeskanowany, jego waga w_n ponownie przyjmuje wartość równą 1.

Do wyznaczenia wartości wag w_n dla podobszarów należących do klasy *podobszar nierozpoznany* stosuje się funkcję liniową rosnącą. Wartości wag tych podobszarów ulegają zmianie w zakresie od 1 w pierwszym kroku do plus nieskończoności.



Rys. 5. Postać funkcji zmiany wagi w_n dla podobszarów nierozpoznanych
Fig. 5. Form of the weight w_n transition function for unknown subfields

Zwiększanie wartości wagi w_n dla niezbadanych podobszarów w kolejnych cyklach algorytmu ma na celu silne zdeterminowanie robotów-agentów do szybkiego zbadania (skanowania) wszystkich podobszarów i zakończenia pierwszego etapu działania algorytmu, czyli przynajmniej jednokrotnego przeskanowania wszystkich podobszarów składowych całego obszaru.

6.4. Wyznaczenie wektorów kierunku ruchu robotów

Pierwszym celem systemu jest całkowite rozpoznanie badanego obszaru przez grupę robotów-agentów inspekcyjnych, które działając zbiorowo, powinny mieć wbudowane mechanizmy współpracy w celu jak najefektywniejszej realizacji postawionego przed nimi zadania.

W kolejnych iteracjach pętli głównej algorytmu roboty–agenty przemieszczają się aby dokonać skanowania nowego podobszaru. Metodyka ustalania kierunku ruchu pojedynczego robota–agenta powinna uwzględniać jego indywidualne cele w inspekcji swojego podobszaru działania ale również uwzględniać aktualną pozycję, kierunek a także wielkość i klasy podobszarów innych robotów-agentów z grupy.

Każdy z robotów–agentów w swoim obszarze działania wyznaczonym za pomocą komórek Voronoi ma zbiór podobszarów należących do jednej z klas: *podobszar rozpoznany* lub *podobszar nierozpoznany*, i podobszary te w zależności od swojej aktualnej klasy powinny oddziaływać na robota-agenta w dwojaki sposób:

- przyciągać – w przypadku podobszarów nierozpoznanych,
 - odpychać – w przypadku podobszarów rozpoznanych,
- przy czym siła tego przyciągania i odpychania zależy od wartości wag poszczególnych podobszarów w taki sposób, że wraz z postępem działania algorytmu wzrasta przyciągania podobszarów nierozpoznanych i maleje odpychanie podobszarów rozpoznanych.

Aby zapewnić takie działanie, należy wyznaczyć dwa wektory odpowiedzialne za przyciąganie do podobszarów nierozpoznanych - , oraz odpychanie od podobszarów badanych - . Wektory te wyznaczane są z odpowiednich wzorów:

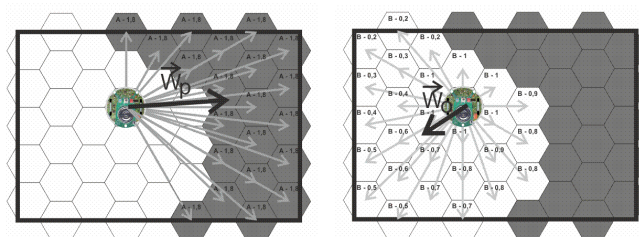
(1)

(2)

gdzie:

- wektor cząstkowy skierowany z aktualnej pozycji robota – agenta do geometrycznego centrum podobszaru nierozpoznanego o indeksie n ,
- wektor cząstkowy skierowany z aktualnej pozycji robota – agenta do geometrycznego centrum podobszaru rozpoznanego o indeksie n ,
- i - odpowiednio wagi podobszarów rozpoznanych i nierozpoznanych.

Sposób wyznaczania wektorów cząstkowych i i odpowiednio wektorów wypadkowych i pokazano na rysunku 6.



Rys. 6. Metoda wyznaczania wektorów i
Fig. 6. The method for determining vectors and

Wektor wypadkowy określający pożądany kierunek ruchu robota wyznaczany jest według wzoru:

(3)

Ujemny znak wektora zapewnia działanie odpychające od podobszarów zbadanych w wyznaczaniu wypadkowego wektora kierunku ruchu robota-agenta .

W początkowych krokach algorytmu, roboty mają za zadanie rozpoznanie nierozpoznanego obszaru, omijając zarazem obszary już rozpoznane. Stąd też wartość wektora A jest zbliżona lub nieznacznie wyższa od wartości wektora B . Na późniejszym etapie, gdy składowych podobszarów nierozpoznanych jest stosunkowo niewiele, sytuacja wymaga od robotów poruszanie się po obszarach już uprzednio rozpoznanych, dlatego też wagi podobszarów rozpoznanych są zmniejszane, a wagi podobszarów nierozpoznanych są zwiększane, co ma na celu wymuszenie na systemie przemieszczanie robotów-agentów w kierunku podobszarów dotychczas nierozpoznanych i szybkie rozpoznanie badanego całego obszaru. W przypadku gdy obszar działania danego robota-agenta zawiera wyłącznie podobszary sklasyfikowane jako rozpoznane, do wyznaczenia kierunku ruchu wykorzystuje się wyłącznie wektor odpychający . W takiej sytuacji zasada działania wektora odpychającego polega na wypychaniu robota z obszaru zawierającego podobszary skanowane stosunkowo niedawno

i kierowanie go w kierunku podobszarów skanowanych stosunkowo dawno, o mniejszych wartościach wag w_n .

6.5. Ruch robotów po wektorach przemieszczenia

Wektor pożądanego kierunku ruchu robota wyznacza jedynie teoretyczny kierunek ruchu robota. Praktycznie, robot–agent powinien skierować się do odpowiedniego podobszaru sąsiedniego z tym, który aktualnie zajmuje. Dlatego też robot–agent,

w przypadku, gdy obszar jest podzielony na sześciokątne podobszary może przemieścić się jedynie w sześciu możliwych kierunkach w taki sposób, aby w kolejnym kroku znaleźć się w geometrycznym centrum jednego z sąsiednich podobszarów.

Opisana metoda wyznaczania wektora kierunku ruchu jest wyjątkowo odporna na występowanie niekorzystnych minimów lokalnych, występujących często w klasycznej metodzie potencjałów i skutkujących zatrzymaniem robota lub jego krążeniem po niewielkim obszarze. Brak występowania minimów lokalnych jest wynikiem następujących faktów:

- wyjątkowo niewielkim prawdopodobieństwem wystąpienia wektora o zerowej wartości,
- nawet w przypadku niewielkiej wartości długości wektora, robot podąża w kierunku sąsiedniego sześciokątnego podobszaru, na który wskazuje kierunek wektora,
- zmiany w każdym kroku pętli głównej algorytmu zbioru wektorów tworzących wektor, ze względu na zmieniający się pod wpływem przemieszczania robotów kształt komórek Voronoi dla każdego robota,
- cyklicznej zmianie wartości wag w_n dla podobszarów tworzących wektor.

Istnieje możliwość łatwego wprowadzenia możliwości występowania przeszkód w badanym obszarze, które stanowią mogą odrębną klasę podobszaru – *zajętego przez przeszkodę*. W takim przypadku, gdy wektor przemieszczenia wskazuje na podobszar zajęty przez przeszkodę, występuje konieczność prostej modyfikacji algorytmu, polegającej na kierowaniu robota do najbliższego podobszaru nie będącego podobszarem zajęty przez przeszkodę.

7. Wyniki symulacji

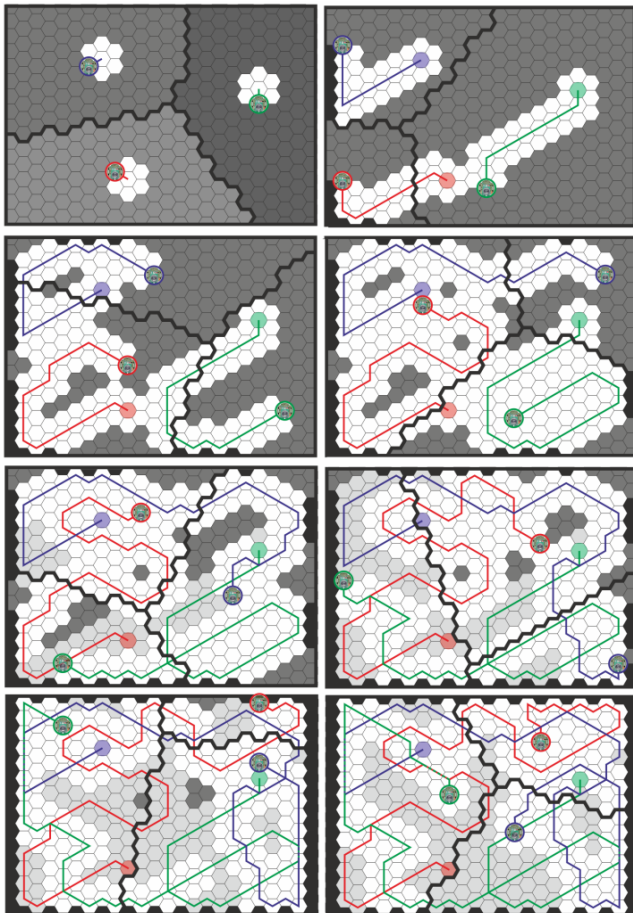
W ramach eksperymentu praktycznego przeprowadzono szereg symulacji działania algorytmu w środowisku symulacji Webots [4]. Plan symulacji zakładał przeprowadzenie eksperymentów na określonej powierzchni obszaru z wykorzystaniem odpowiednio: jednego, dwóch, trzech i czterech robotów-agentów oraz dla każdego z eksperymentów wyznaczenie współczynników efektywności badania obszaru w zależności od liczby kroków koniecznych do rozpoznania całego badanego obszaru.

W każdym kroku symulacji zapisywano dwie wartości dotyczące efektywności badania obszaru przez każdego robota. Pierwszą wartością jest liczba podobszarów skanowanych po raz pierwszy przez każdego robota, co pozwala wyznaczyć liczbę efektywnych kroków symulacji dla pierwszego etapu rozpoznania obszaru. Druga wartość opisuje aktualny, procentowy udział każdego robota w rozpoznawaniu całego obszaru. Dzięki temu, można określić dynamikę wzrostu poziomu rozpoznania obszaru przez każdego robota a tym samym efektywność całego systemu.

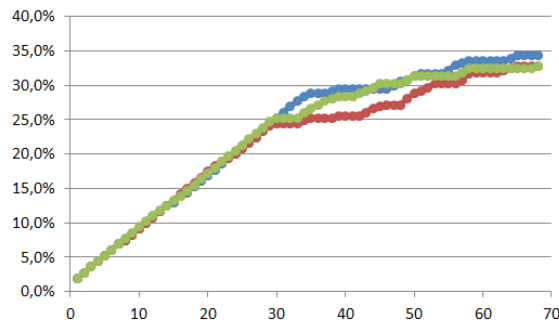
Na rysunku 7 pokazano przebieg eksperymentu dla trzech robotów kolejno w 1, 10, 20, 30, 40, 50, 60 i ostatnim – 68 kroku symulacji.

Analiza rysunku 7 pozwala na wyciągnięcie wniosków o efektywności algorytmu. Zaznaczone ścieżki, po których poruszają się roboty w trakcie eksperymentu pokazują, że istnieje kooperacja między robotami, które działają w sposób kolektywny w celu rozpoznania całego obszaru. Każdy z robotów, w każdym kroku symulacji ma swój indywidualny obszar badań wyznaczony komórkami Voronoi, jednak, obszary te zmieniają się w taki sposób, aby równoważyć obciążenie poszczególnych robotów w realizacji zadania rozpoznania obszaru.

Na rysunku 8 pokazano wykres dynamiki wzrostu rozpoznawania badanego obszaru przez każdego z robotów osobno.



Rys. 7. Przebieg eksperymentu dla 1, 10, 20, 30, 40, 50, 60 i 68 kroku symulacji
 Fig. 7. State of the experiment in the 1st, 10th, 20th, 30th, 40th 50th 60th and 68th step



Rys. 8. Wykres dynamiki wzrostu rozpoznawania obszaru przez każdego robota osobno w trakcie symulacji
 Fig. 8. Dynamics of each robotic agent field recognition level during simulation process

Wykres ten obrazuje dynamikę efektywności rozpoznawania obszaru, rozumianą jako skumulowaną liczbę skanowanych po raz pierwszy podobszarów w trakcie procesu symulacji. Z analizy wykresu wynika, że do około 30 kroku symulacji, każdy robot rozpoznawał nowe – nierozpoznane wcześniej podobszary z taką samą efektywnością i dopiero powyżej 30 kroku roboty–agenty przechodziły przez obszary już wcześniej rozpoznane. Zbieżność wykresów dynamiki wzrostu efektywności rozpoznawania dla każdego robota pokazuje że każdy z robotów wykonał proporcjonalną część zadania globalnego co jest pozytywnym.

8. Wnioski

Prezentowany system wykazał bardzo dużą efektywność zarówno w rozpoznawaniu jak i ciągłej inspekcji badanego obszaru. Realizowane zadanie należy do trudnych zadań związanych z systemami wieloagentowymi i ma szerokie zastosowania w wielu dziedzinach wiedzy. Praktyczne zastosowania opisanego systemu występują w takich dziedzinach jak: zastosowanie w technice wojskowej, pomiary dużych powierzchni, rozminowywanie terenu, roje robotów czy nawet nowoczesne systemy koszenia trawników. Badania efektywności systemu przeprowadzono pod względem wielu aspektów takich jak między innymi: wpływ liczby robotów-agentów na efektywność rozpoznawania i inspekcji obszaru w funkcji jego wielkości, wpływ liczby podobszarów, które każdy robot pokonuje w pojedynczym kroku na dynamikę rozpoznawania, wpływ liczby robotów-agentów na poziom współczynnika efektywności inspekcji w drugim etapie działania systemu. Kompleksowe wyniki, analizy i wnioski z eksperymentu symulacji opisanego systemu, ze względu na swoją obszerność nie mogły być zamieszczone w niniejszym artykule.

9. Literatura

- [1] Dasgupta P.: Multi-agent coordination techniques for multi-robot task allocation and multi-robot area coverage. Collaboration Technologies and Systems (CTS), International Conference on Digital Object Identifier 2012.
- [2] Dasgupta P.: Multi-Robot Task Allocation for Performing Cooperative Foraging Tasks in an Initially Unknown Environment. Innovations in Defense Support Systems - 2, Springer, Studies in Computational Intelligence, vol. 338, s. 5-20, 2011.
- [3] Mondada F., Bonani M., Raemy X., Pugh J., Cianci C., Klaptocz A., Magnenat S., Zufferey J.C., Floreano D., Martinoli, A.: The e-puck, a Robot Designed for Education in Engineering. Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, s. 59-65.
- [4] Michel O.: Webots: Professional Mobile Robot Simulation, International Journal of Advanced Robotic Systems, vol. 1, Num. 1, s. 39-42.
- [5] Ellips M., Amin-Naseri M.R.: A Voronoi Diagram-Visibility Graph-Potential Field Compound Algorithm for Robot Path Planning. Journal of Robotic Systems, vol. 21, s. 275-300, 2004.
- [6] Barracuand J., Langlois B., Latombe J.: Numerical potential field techniques for robot path planning. IEEE Transactions on systems man and cybernetics, vol. 22, s. 224-241, 1992.
- [7] Ge S.S., Cui Y.J.: Dynamic Motion Planning for Mobile Robots Using Potential Field Method. Autonomous Robots vol. 13, Issue: 3, s. 207 – 222, 2002.
- [8] Kim Dong H., Shin Seiichi.: Self-organization of Decentralized Swarm Agents Based on Modified Particle Swarm Algorithm. Journal of Intelligent and Robotic Systems, vol. 46, Issue 2, s. 129 – 149, 2006.
- [9] Aydin Mehmet Emin.: Coordinating metaheuristic agents with swarm intelligence. Journal of Intelligent Manufacturing, vol. 23, Issue: 4, s. 991 – 999, 2012.
- [10] Ephrati E., Jeffrey S.: A heuristic technique for multi-agent planning. Annals of Mathematics and Artificial Intelligence, vol. 20, Issue: 1-4, s. 13 – 67, 1997.
- [11] Tehrani Nik, Nejad Hossein, Sugimura Nobuhiro, Iwamura Koji, Tanimizu Yoshitaka.: Multi agent architecture for dynamic incremental process planning in the flexible manufacturing system. Journal of Intelligent Manufacturing, vol. 21, Issue: 4, s. 487 – 499, 2010.
- [12] Neto A., Macharet A., Douglas G., Mario F. M.: On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles. Journal of Intelligent and Robotic Systems, vol. 57, Issue: 1-4, s. 123 – 141, 2010.

-
- [13]Ioannidis K., Sirakoulis G.Ch., Andreadis I.: Cellular ants: A method to create collision free trajectories for a cooperative robot team. Robotics and Autonomous Systems, vol. 59, Issue: 2, s. 113-127, 2011.

otrzymano / received: 13.05.2013

przyjęto do druku / accepted: 01.10.2013 artykuł recenzowany / revised paper