

**Politechnika Śląska  
Wydział Mechaniczny Technologiczny  
Katedra Podstaw Konstrukcji Maszyn**

**Wawrzyniec PANFIL**

**System  
sterowania  
grupą  
inspekcyjnych  
robotów mobilnych  
bazujący  
na zachowaniach**

**Gliwice 2012**

*Recenzenci*

Prof. dr hab. inż. Jan Maciej Kościelny, Politechnika Warszawska

Prof. dr hab. inż. Wojciech Cholewa, Politechnika Śląska

*Redaktor zeszytów*

Wojciech Cholewa

*Redaktor techniczny*

Marek Wyleźoń

*Projekt okładki*

Wojciech Cholewa, Marek Wyleźoń

*BibTeX*

```
@BOOK{, title = {System sterowania grupą inspekcyjnych robotów  
mobilnych bazujący na zachowaniach}, publisher =  
{Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn}, year  
= {2012}, author = {Panfil, W.}, volume = {145}, series =  
{Zeszyty}, address = {Gliwice}}
```

Znaczące fragmenty pracy zrealizowano w ramach projektu badawczego promotorskiego N N514 406336 finansowanego przez MNiSW.

**ISBN 978-83-60759-20-2**

*Wydawca*

Katedra Podstaw Konstrukcji Maszyn

Wydział Mechaniczny Technologiczny

Politechnika Śląska

ul. Konarskiego 18a, 44-100 Gliwice

tel. (32) 237-14-67, fax (32) 237-13-60

<http://kpk.m.polsl.pl>

---

Druk i oprawa (z dostarczonych materiałów)  
Błękitne Studio Graficzne, ul. Chopina 6, 44-100 Gliwice

# Od autora

Zeszyt został opracowany na podstawie rozprawy doktorskiej, którą wykonałem pod kierunkiem prof. dra hab. Wojciecha Moczulskiego. Publiczna obrona rozprawy odbyła się 6 grudnia 2011 roku przed Komisją powołaną przez Radę Wydziału Mechanicznego Technologicznego. Znaczące fragmenty pracy zrealizowano w ramach projektu badawczego promotorskiego N N514 406336 finansowanego przez MNiSW. Kierownikiem projektu był promotor rozprawy.

Składam serdeczne podziękowania recenzentom rozprawy doktorskiej, prof. dr. hab. inż. Janowi M. Kościelnemu oraz prof. dr. hab. inż. Wojciechowi Cholewie, za cenne uwagi, które starałem się uwzględnić w niniejszym zeszycie.

Pragnę wyrazić swoją wdzięczność mojemu promotorowi prof. dr. hab. Wojciechowi Moczulskiemu za cenne wskazówki, inspirację i wsparcie podczas przygotowywania rozprawy doktorskiej, a także za intensywną i wszechstronną opiekę naukową podczas studiów magisterskich i doktoranckich. Składam serdeczne podziękowania Koleżankom i Kolegom z Katedry Podstaw Konstrukcji Maszyn Politechniki Śląskiej za okazaną mi pomoc i życzliwość w czasie wykonywania pracy.

Niniejszą książkę dedykuję mojej najbliższej rodzinie, w szczególności mojej żonie Alicji i synowi Adasiowi, dziękując za ich wsparcie, wyrozumiałość i cierpliwość.

*Gliwice, maj 2012*

*Wawrzyniec Panfil*



# Spis treści

<b>Od autora</b>	<b>3</b>
<b>Wykaz ważniejszych oznaczeń i skrótów</b>	<b>9</b>
<b>Rozdział 1. Wstęp</b>	<b>13</b>
1.1. Podstawowe pojęcia	16
1.2. Cel rozprawy	17
1.3. Tezy rozprawy	18
1.4. Zakres rozprawy	18
<b>Rozdział 2. Sterowanie behawioralne oraz metody alokacji zadań w systemach wielorobotowych</b>	<b>21</b>
2.1. Sterowanie deliberatywne, reaktywne i hybrydowe	21
2.2. Sterowanie behawioralne	23
2.3. Koordynacja zachowań	23
2.3.1. Konkurencyjne mechanizmy koordynacji zachowań	24
2.3.2. Kooperatywne mechanizmy koordynacji zachowań	25
2.3.3. Architektury behawioralnych systemów sterowania	26
2.4. Taksonomia systemów wielorobotowych	30
2.5. Wymagania systemów wielorobotowych	30
2.6. Kooperacja a koordynacja w systemach wielorobotowych	31
2.6.1. Sterowanie nadrzędne oraz zdecentralizowane	32
2.7. Systemy wieloagentowe a wielorobotowe	33
2.8. Alokacja zadań w systemach wielorobotowych	34
2.9. Podsumowanie	36
<b>Rozdział 3. System sterowania grupą robotów inspekcyjnych</b>	<b>37</b>
3.1. Założenia	37
3.2. Formalna reprezentacja systemu	39
3.2.1. Konfiguracja misji	39
3.2.2. Model środowiska	40
3.2.3. Model zasobów inspekcyjnych	42
3.2.4. Model robota	43
3.2.5. Model zadania	45

3.2.6.	Zbiór robotów R. . . . .	47
3.2.7.	Zbiór zadań Z. . . . .	48
3.2.8.	Działanie systemu . . . . .	48
3.3.	Metody alokacji zadań oparte o Contract-Net Protocol . . . . .	52
3.3.1.	Parametry misji . . . . .	52
3.3.2.	Przebieg procesu alokacji zadań . . . . .	53
3.3.3.	Kryteria oceny ofert. . . . .	56
3.3.4.	Model oferty mechanizmu aukcyjnego . . . . .	57
3.3.5.	Kolejkowa alokacja zadań . . . . .	60
3.3.6.	Alokacja zadań według kolejki robotów . . . . .	61
3.4.	Wielokryterialne metody alokacji zadań . . . . .	62
3.4.1.	Algorytmy wyznaczania wartości kryteriów . . . . .	62
3.4.2.	Pozycja konkurencyjna robota dla zadania . . . . .	64
3.4.3.	Atrakcyjność zadania dla robota. . . . .	65
3.4.4.	Wielokryterialna ocena przystosowania robota do zadania. . . . .	70
3.4.5.	Metody przydzielania zadań stosujące wielokryterialną ocenę przystosowania robotów do zadań MCFV. . . . .	70
3.5.	Behawioralne sterowanie ruchem robotów . . . . .	74
3.5.1.	Koordinator ruchu robota. . . . .	74
3.5.2.	Koordinator zachowań . . . . .	76
3.5.3.	Zachowania. . . . .	77
<b>Rozdział 4.</b>	<b>Badania weryfikacyjne</b>	<b>81</b>
4.1.	Sposób prezentacji wyników badań. . . . .	81
4.2.	Reprezentatywność wyników badań . . . . .	83
4.3.	Miary oceny działania systemu . . . . .	84
4.4.	Obiekt badań symulacyjnych. . . . .	85
4.4.1.	Roboty inspekcyjne . . . . .	85
4.4.2.	Parametry behawioralnego podsystemu sterowania ruchem robotów . . . . .	87
4.4.3.	Przykładowe zadania inspekcyjne . . . . .	88
4.5.	Plan badań weryfikacyjnych . . . . .	88
4.5.1.	Środowiska testowe oraz konfiguracje eksperymentów . . . . .	89
4.5.2.	Warianty eksperymentów. . . . .	90
4.6.	Eksperymenty weryfikacyjne . . . . .	91
4.6.1.	Środowisko 1 . . . . .	91
4.6.2.	Środowisko 2 . . . . .	102
4.7.	Podsumowanie badań weryfikacyjnych . . . . .	107
<b>Rozdział 5.</b>	<b>Podsumowanie</b>	<b>109</b>
5.1.	Wnioski . . . . .	110
5.2.	Kierunki dalszych badań . . . . .	111

**Bibliografia** 118

**Streszczenie** 119

**Summary** 120





# Wykaz ważniejszych oznaczeń i skrótów

$b$	oferta na wykonanie zadania
$B$	zbiór ofert $b$ jednej aukcji
$bl$	poziom naładowania akumulatorów robota (zapas energii)
$\mathbb{C}$	konfiguracja misji
$d$	odległość (parametr oferty)
$dev^{insp}$	urządzenie inspekcyjne (zasób inspekcyjny) wymagane przez zadanie lub posiadane przez robota
$E$	zbiór krawędzi należących do $TO$
$ec^{insp}$	zużycie energii zasobu inspekcyjnego $dev^{insp}$ przypadające na jednostkę czasu
$ec^{sb}$	jednostkowe zużycie energii robota wynikające z działania podstawowych układów robota
$ec^{trav}$	jednostkowe zużycie energii robota wynikające z działania układów napędowych robota
$ec^{ex}$	jednostkowe zużycie energii robota wynikające z działania zasobów inspekcyjnych robota
$ee^{nb}(et^{nb})$	energia zużyta (poświęcony czas) przez robota w trakcie jego bezczynności, tj. w trakcie oczekiwania na przydział zadania
$ee^{trav}(et^{trav})$	energia zużyta (poświęcony czas) przez robota w trakcie dojazdu do podjętych zadań
$ee^{ex}(et^{ex})$	energia zużyta (poświęcony czas) przez robota w trakcie wykonywania zadań
$ere$	szacowana ilość energii potrzebnej na realizację zadania
$ere^{r \rightarrow z}, ert^{r \rightarrow z}, ur^{r \rightarrow z}$	pozycja konkurencyjna robota dla wybranego zadania ze względu na kryterium: energii potrzebnej na realizację zadania, czasu potrzebnego na realizację zadania, liczby nadmiarowych zasobów inspekcyjnych robota

$ere^{z \rightarrow r}, ert^{z \rightarrow r},$	atrakcyjność zadania dla wybranego robota ze względu na kryterium:
$ur^{z \rightarrow r}, zc^{z \rightarrow r}, zp^{z \rightarrow r}$	energii potrzebnej na realizację zadania, czasu potrzebnego na realizację zadania, liczby nadmiarowych zasobów inspekcyjnych robota, zmiany koncentracji zadań, zysku z wykonania zadania
$ert$	szacowany czas potrzebny na realizację (dojazd i wykonanie) zadania
$id^r$	identyfikator robota
$id^z$	identyfikator zadania
$L$	wektor reprezentujący graf $TO$ w pamięci komputera
$let$	czas ostatniego zdarzenia dotyczącego robota (przydział, porzucenie oraz wykonanie zadania, aktywowanie robota)
$mcfv$	wielokryterialna ocena przystosowania robota do zadania
$MCFV$	zbiór metod stosujących wielokryterialną ocenę przystosowania robota do zadania
$MP$	parametry misji
$MRDS$	Microsoft <sup>®</sup> Robotics Developer Studio
$r$	robot
$R, R^{act}, R^{ex},$	odpowiednio zbiory robotów: wszystkich biorących udział w misji do danej chwili czasu, aktywnych, realizujących zadania, nieaktywnych, nie-
$R^{nact}, R^{nb}, R^{new}$	zajętych (kolejka robotów), wprowadzanych do systemu w danej chwili czasu
$r^{cd}$	indeks robota, który zrealizował wybrane zadanie $z$
$r^{ex}$	indeks robota realizującego wybrane zadanie $z$
$rcp$	zbiór parametrów dot. pozycji konkurencyjnej robota dla wybranego zadania
$rcp^{r \rightarrow z}$	pozycja konkurencyjna robota dla wybranego zadania
$res^{insp}$	zbiór zasobów inspekcyjnych posiadanych przez robota lub wymaganych przez zadanie
$t$	aktualny czas misji
$\tau = (t^{in}, t^{ex}, t^{al},$	zbiór parametrów związanych z czasem dotyczących zadania $z$ (czas
$t^{st}, t^{left}, t^{await}, t^{act}, t^f)$	wprowadzenia do systemu $t^{in}$ , potrzebny na realizację $t^{ex}$ , ostatniego przydziału $t^{al}$ , ostatniego rozpoczęcia realizacji $t^{st}$ , pozostałego czasu do wykonania $t^{left}$ , oczekiwania $t^{await}$ , wykonywania $t^{act}$ oraz zrealizowania $t^f$ zadania)
$td$	dystans przebyty przez robota
$TO$	ograniczona topografia przestrzeni
$ur$	liczba zasobów inspekcyjnych robota niepotrzebnych do realizacji wybranego zadania
$v$	wierzchołek należący do $TO$

$V$	zbiór wierzchołków należących do $TO$
$V^r$	zbiór wierzchołków należących do $TO$ , które robot musi odwiedzić, żeby dojechać do przydzielonego zadania
$vel^{act}, vel^{trg}, vel^{max}$	aktualna, docelowa, maksymalna prędkość liniowa ruchu robota
$w_{i,j}$	waga (odległość) pomiędzy dwoma wierzchołkami $v_i$ i $v_j$ należącymi do $TO$
$x, y, \theta$	współrzędne określające położenie robota i zadania (bez $\theta$ ) w trójwymiarowej przestrzeni rzeczywistej
$x^{act}, y^{act}, \theta^{act}, \theta^{trg}$	aktualne położenie robota oraz docelowa orientacja
$z$	zadanie
$Z$	zbiór wszystkich zadań misji
$Z^{cd}$	zbiór zadań zrealizowanych
$Z_r^{cd}$	zbiór zadań zrealizowanych przez robota $r$
$Z^{nc}$	zbiór zadań niewykonanych
$Z^q$	zbiór zadań oczekujących na przydział (kolejka zadań)
$z^{ex}$	aktualnie realizowane (dojazd lub wykonywanie) zadanie przez robota
$Z^{ex}$	zbiór zadań wykonywanych
$z_a$	zbiór parametrów dot. atrakcyjności zadania dla wybranego robota
$z_a^{z \rightarrow r}$	atrakcyjność zadania dla wybranego robota
$z_C$	zmiana koncentracji zadań dla robota wynikająca z wykonania wybranego zadania



# Rozdział 1

## Wstęp

We współczesnym świecie wiele obiektów użyteczności publicznej, zarówno zamkniętych jak i otwartych, wymaga ciągłego nadzorowania, monitorowania, czy też prowadzenia w ich obrębie szeroko rozumianych działań inspekcyjnych. Sytuacja taka spowodowana jest przede wszystkim wzrastającymi wymogami bezpieczeństwa, będącymi konsekwencją nasilających się oraz przybierających coraz to nowsze formy zagrożeń. Źródła takich zagrożeń są różnorodne, jak np. wycieki z instalacji chemicznych, niepożądane elementy na pasie startowym lotniska, pola minowe, niepowołani, czy też wrogo nastawieni ludzie oraz wiele innych.

Wykonywanie wyżej wspomnianych działań przez ludzi zazwyczaj jest niebezpieczne, wiąże się z narażaniem ich zdrowia, a nawet życia, a ponadto bywa kosztowne ze względu na czas potrzebny do realizacji takich działań. Dlatego też, od kilku dziesięcioleci, jest to powodem wykorzystywania robotów do takich zadań. Jako przykłady niech posłużą roboty do inspekcji rurociągów służących do transportu różnego rodzaju (w tym szkodliwych dla zdrowia) mediów (Giergiel *i in.*, 2010; Moczulski *i in.*, 2007; Panfil *i in.*, 2007), rozbrajające bomby (Bigaj *i in.*, 2010), dokonujące inspekcji obiektów zamkniętych (Chojecki *i in.*, 2004; Czupryniak i Trojnecki, 2010), oznaczające tereny zaminowane, eksplorujące powierzchnię Marsa, czy też bezzałogowe pojazdy wojskowe HMMWV (High Mobility Multipurpose Wheeled Vehicle) (Balch i Arkin, 1998).

W przypadku zadań o małej skali zastosowanie jednego robota jest wystarczające. Jednakże istnieje wiele zadań, gdzie korzyści wypływające z działania grupowego są czynnikiem przemawiającym za stosowaniem systemów wielorobotowych. W trakcie budowy systemów wielorobotowych wykorzystuje się analogie do organizacji zespołów ludzkich (Horling i Lesser, 2005; Gasser, 2001), czy też zbiorowisk lub grup zwierząt. Wiele zadań (prac), ze względów bezpieczeństwa, ekonomicznych, nie może być wykonywanych przez jedną osobę (robota). Stąd naturalną konsekwencją jest angażowanie grupy pracowników (robotów), z których każdy posiada odrębne kwalifikacje. Dzięki temu pracownicy (roboty) spełniają w grupie określone role, wnoszą do grupy specyficzne umiejętności, jak również uzupełniają się wzajemnie. Ważną cechą grupy może być posiadanie przez każdego jej członka zestawu tych samych umiejętności, co pozwala na migrację zadań pomiędzy członkami grupy w zależności od obciążenia poszczególnych pracowników (robotów), jak również w momencie powstania nieprzewidzianych okoliczności. Dzięki temu

wzrasta elastyczność, jak i zdolności przystosowawcze (adaptacyjność) grupy.

Zadania, często zwane misjami, realizowane mogą być przez jednego robota, bądź ich grupę. Liczność takiej grupy zdeterminowana jest zwykle charakterem wykonywanych zadań, a przez to postawionymi wymaganiami, i może wynosić od kilku do nawet kilkuset. Istnieje wiele przesłanek przemawiających za stosowaniem systemów wielorobotowych (Arkin, 1998; Balch i Arkin, 1998; Cao *i in.*, 1997; Fierro *i in.*, 2006; Jones i Matarić, 2006):

- Znacząca liczba zadań nie może zostać zrealizowana przez jednego robota,
- Zastosowanie kilku robotów pozwala na kontynuowanie wykonywanego zadania nawet w przypadku częściowego lub całkowitego braku zdadności do działania jednego, czy kilku robotów,
- Postawione zadanie może zostać zrealizowane szybciej przez grupę robotów, roboty nie muszą być uniwersalne – dotyczy to zarówno możliwości poszczególnych robotów (rodzaju wykonywanych zadań), jak i ich konfiguracji sprzętowych oraz programowych,
- Budowa grupy mało skomplikowanych, mniej „inteligentnych” robotów jest często mniej kosztowna oraz łatwiejsza w porównaniu z budową jednego wielozadaniowego, uniwersalnego robota,
- Poprzez komunikację oraz wymianę informacji zostają zwiększone możliwości sensoryczne pojedynczych robotów, jak i całej grupy,
- Zespołowe współdziałanie grupy robotów może zwiększyć ich zdolność do realizacji specyficznych zadań podczas misji, jak np. pokonywanie przeszkód (schody, rowy i in.).

Do podstawowych zastosowań, dla których prowadzi się obecnie badania nad rozwojem systemów wielorobotowych, zaliczyć należy:

- Gromadzenie (ang. *foraging* (Arkin, 1998)) losowo rozrzuconych obiektów w jedno miejsce (Parker, 1998) , np. zbieranie śmieci, odpadów, piłek golfowych,
- Zadania polegające na wykonywaniu pracy nad obiektem w miejscu znajdowania się tego obiektu (ang. *consuming* (Arkin, 1998)), np. montaż/demontaż obiektu, rozminowywanie,
- Zadania polegające na dokładnym eksplorowaniu wyznaczonego terenu (ang. *grazing* (Arkin, 1998)), np. monitorowanie terenu, koszenie trawy, odkurzanie mieszkań, budowanie mapy obszaru (Zlot *i in.*, 2002) itp.,
- Podążanie grupy w zadanej formacji (ang. *flocking* (Arkin, 1998)) (Balch i Arkin, 1998),
- Transportowanie obiektu przez kilka robotów. Zazwyczaj zadanie to sprowadza się do gromadzenia obiektów w danym miejscu przez grupę robotów poruszających się w zadanym szyku. Przykładami mogą być koordynowane przesuwanie (Gerkey i Matarić, 2002) (Arkin, 1998)) (Balch i Arkin, 1998), czy też przenoszenie obiektów,

- Rozmieszczenie sieci sensorycznej (Batalin i Sukhatme, 2002).

Większość z wymienionych zastosowań wymaga realizacji zadań, które z reguły są rozproszone w czasie, przestrzeni i funkcjonalności (Parker, 1998). Stąd często korzystnie jest, aby grupa wykonująca te zadania składała się z heterogenicznych robotów, zarządzanych przez odpowiedni system sterowania. Dużym wyzwaniem jest właściwe opracowanie takiego systemu, tj. rozwiązanie szeregu problemów (zapewnienie spójności systemu, dekompozycja i alokacja<sup>1</sup> zadań w grupie, interakcja między robotami) nie pojawiających się w systemach jednorobotowych. Niewłaściwie opracowany system sterowania może stać się przyczyną nadmiernego skomplikowania, zamiast uproszczenia, rozwiązania zadania.

W prowadzonych badaniach (Parker, 1998; Sariel, 2007) dotyczących budowy systemów wielorobotowych stawiane są często następujące pytania charakterystyczne również dla systemów wieloagentowych ((Parker, 1998) idąc za (Bond i Gasser, 1988)):

- W jaki sposób sformułować, opisać, zdekomponować i przydzielić zadania pośród grupy inteligentnych agentów (robotów)?
- Jak skonstruować mechanizm komunikacji i interakcji agentów (robotów)?
- Jak zapewnić spójność działania agentów (robotów) ?
- Jak zapewnić agentom (robotom) możliwość rozpoznawania i rozwiązywania konfliktów?
- Jak zapewnić działanie systemu w nieprzewidzianych sytuacjach?

Są to podstawowe pytania, na które należy sobie odpowiedzieć w trakcie budowy systemów sterowania. Istotnym jest również zwrócenie uwagi na fakt, że roboty mobilne mogą być traktowane jako agenty - w przeciwieństwie do komputerowych (programowych, abstrakcyjnych) agentów - mające postać fizyczną i wzajemnie oddziałujące z otaczającym je środowiskiem (Arkin, 1998). Mimo daleko posuniętych badań w zakresie budowy robotów autonomicznych i modelowania współpracy w zespołach robotów, istnieje jeszcze wiele problemów, których rozwiązanie pozwoli na coraz szersze stosowanie grup robotów mobilnych w praktycznych zadaniach. Przykładem takiego problemu jest alokacja zadań w grupie robotów. W literaturze (Zaborowski, 2008; Gerkey, 2003) problem alokacji zadań często formułowany jest jako problem optymalnego przydziału w następujący sposób:

*Danych jest  $n$  pracowników (robotów), z których każdy może w danej chwili wykonywać jedno zadanie oraz  $m$  zadań do wykonania, każde wymagające jednego pracownika (roboty). Dla każdej pary (pracownik, zadanie) istnieje funkcja określająca przystosowanie pracownika (roboty) do wykonania zadania. Jeśli pracownik (robot) nie jest przystosowany do wykonania zadania, wtedy funkcja przyjmuje wartość zerową. Rozwiązanie problemu polega na znalezieniu takiego przydziału zadań pracownikom (robotom), aby uzyskać maksymalną wydajność zespołu pracowników (robotów) z uwzględnieniem wydajności poszczególnych pracowników (robotów) i priorytetów zadań.*

<sup>1</sup>Znaczenie pojęcia *alokacja* wyjaśnione jest w rozdz. 1.1.

Dotychczas na potrzeby alokacji zadań w systemach wielorobotowych stosowanych było wiele rozwiązań, z których można wyróżnić trzy główne podejścia:

- Oparte o metody programowania liniowego (Nuzhet i Burchan, 2006),
- Stosujące behawioralne systemy sterowania, np. ALLIANCE (Parker, 1998), BLE (Werger i Matarić, 2001),
- Bazujące na modelach ekonomicznych, np. aukcje (Gerkey i Matarić, 2002) i negocjacje (Kalra *i in.*, 2005).

W powszechnie stosowanych aukcyjnych metodach alokacji zadań stosuje się mechanizm aukcjonowania zadań oparty o Contract-Net Protocol (Smith, 1980), w którym zadania rozważane są wg kolejności pojawienia się w systemie i dla każdego kolejnego zadania poszukuje się najlepiej przystosowanego robota. Wydaje się, że w systemach wielorobotowych korzystne byłoby zastosowanie podejścia, w którym aukcjonuje się roboty – dla każdego kolejnego robota w kolejce robotów znajduje się najbardziej atrakcyjne dla niego zadanie. Pozwoliłoby to przede wszystkim na uniknięcie sytuacji, w których w wyniku przydziału zadania wynikającego z jego pozycji w kolejce zadań robot musi wykonać zadanie, które dla niego jest mniej korzystne (zużyje więcej energii, pokona większy dystans) niż inne zadanie, którego miejsce wykonania znajduje się bliżej robota, ale będące na dalszej pozycji w kolejce zadań.

Innym problemem zauważonym w działaniu metod aukcyjnych jest jednostronna ocena przystosowania robota do zadania – dla danego zadania poszukiwany jest robot mogący najlepiej (ze względu na czas wykonania, zużytą energię) te zadanie wykonać. Wydaje się, że jednoczesna dwustronna ocena przystosowania robota do zadania (dla zadania poszukuje się najlepiej przystosowanego robota, a dla robota najbardziej atrakcyjne dla niego zadanie) pozwoli na efektywniejsze (skróci się czas realizacji misji, zmniejszy się sumaryczne zużycie energii robotów, czy też sumaryczny przebyty przez nie dystans) realizowanie misji przez grupę robotów. Inspiracją propozycji autora jest lektura dotycząca portfelowych metod analizy rynku, a w szczególności metody stosującej macierz General Electric (macierz McKinseya) (Lisiński, 2004; Stabryła, 2000).

## 1.1. Podstawowe pojęcia

### Kolejka zadań

W każdej chwili działania systemu może istnieć zbiór zadań oczekujących na przydzielenie ich robotom. W metodzie alokacji zadań stosującej aukcjonowanie zadań istotna jest kolejność wprowadzania zadań do zbioru zadań, a więc zbiór ten należy traktować jako zbiór uporządkowany, w którym porządek elementów wynika z kolejności ich wprowadzania do zbioru. Dla wspomnianej metody zadania rozważane są zgodnie z zasadą działania kolejki (FIFO – First In, First Out). Dlatego zdecydowano się stosować pojęcie *kolejka zadań*.



### **Kolejka robotów**

Określenie jest podobne, jak w przypadku pojęcia *kolejka zadań*, z tym że pojęcie *kolejka robotów* dotyczy zbioru robotów niezajętych w danej chwili działania systemu grupy robotów. Dla kilku metod alokacji zadań opisywanych w pracy znaczenie ma kolejność wprowadzania robotów do i pobierania ich ze zbioru robotów, dlatego w tym przypadku zdecydowano się również na traktowanie zbioru robotów niezajętych jako kolejki.

### **Realizowanie zadania**

Robot w wyniku procesu alokacji może otrzymać zadanie do zrealizowania. W pracy realizowanie zadania traktowane jest jako dojazd robota do miejsca wykonywania zadania, po którym następuje wykonywanie zadania.

### **Wykonywanie zadania**

Jest to proces rozpoczynający się w momencie dojazdu robota do miejsca wykonywania zadania, a kończy się, kiedy robot wykonujący zadanie znajduje się w miejscu jego wykonywania przez czas określony przez jeden z parametrów zadania określający czas wykonywania zadania.

### **Alokacja zadań**

W literaturze anglojęzycznej dotyczącej robotów mobilnych często używa się sformułowania *task allocation* (Dahl *i in.*, 2009; Sariel i Balch, 2005; Gerkey i Mataric, 2003a). Stąd w pracy postanowiono stosować pojęcie *alokacja zadań*, które określa proces aukcjonowania zadań lub robotów oraz następujący po nim proces przydzielania zadań. Należy zaznaczyć, że nie jest możliwe przydzielanie zadań robotom bez wcześniejszego aukcjonowania.

## **1.2. Cel rozprawy**

Celem rozprawy było opracowanie podstaw teoretycznych i praktyczna weryfikacja metodyki konstruowania systemów sterowania grupą robotów mobilnych przeznaczonych do realizacji działań inspekcyjnych. Opracowana metodyka umożliwia tworzenie oprogramowania dla różnorodnych grup robotów, zaś w szczególności przewiduje się jej zastosowanie w konstruowaniu systemów sterowania dla grup robotów projektowanych i konstruowanych w Katedrze Podstaw Konstrukcji Maszyn Politechniki Śląskiej w ramach projektów badawczych realizowanych wspólnie z Instytutem Technologii Eksploatacji – Państwowym Instytutem Badawczym w Radomiu.

### 1.3. Tezy rozprawy

W oparciu o przedstawiony problem badawczy oraz zdefiniowany cel rozprawy, sformułowano następujące tezy:

1. *Alokacja zadań w grupie robotów bazująca na aukcjonowaniu robotów zapewnia efektywniejsze realizowanie misji w porównaniu do alokacji zadań bazującej na aukcjonowaniu zadań.*

2. *Metody alokacji zadań w grupie robotów mobilnych bazujące na wielokryterialnej analizie stanu robotów oraz ich otoczenia umożliwiają wzrost efektywności realizacji misji w porównaniu z powszechnie stosowanymi metodami aukcyjnymi.*

### 1.4. Zakres rozprawy

Rozprawa skupia się na opisie formalnym oraz weryfikacji systemu sterowania grup robotów mobilnych wykonujących zadania inspekcyjne. Opisana metodyka postępowania może być zastosowana po jej adaptacji do konstruowania systemów sterowania grup robotów przeznaczonych do innych zadań.

W rozdziale pierwszym zamieszczono wstęp przedstawiający szerokie spektrum zastosowań systemów wielorobotowych oraz wyszczególniono podstawowe problemy jakie pojawiają się podczas budowy takich systemów, ze szczególnym uwypukleniem problemu efektywnego rozdziału zadań w grupie robotów heterogenicznych. Rozdział ponadto zawiera cel, tezy oraz wyjaśnienie podstawowych pojęć stosowanych w rozprawie.

Rozdział drugi obejmuje syntetyczną prezentację obszaru badawczego. W pierwszej części tego rozdziału opisane zostały stosowane obecnie podejścia do sterowania ruchem robotów, tj. reaktywne, deliberatywne i hybrydowe. Dodatkowo przedstawione jest podejście bazujące na zachowaniach, z dokładną charakterystyką mechanizmów koordynacji zachowań. W drugiej części rozdziału zawarto opis zagadnień związanych z konstruowaniem systemów wielorobotowych, skupiając się na stosowanych metodach aukcjonowania i przydzielania zadań w grupie robotów. Ponadto opisano zidentyfikowaną przez autora potrzebę opracowania nowych metod alokacji zadań.

Rozdział trzeci poświęcony został głównemu celowi rozprawy. Zawartość tego rozdziału można podzielić na trzy części. W pierwszej przyjęto podstawowe założenia, jak również wprowadzono podstawowe modele systemu sterowania. W drugiej części rozdziału uzupełniono opis formalny systemu tak, aby umożliwić prowadzenie procesów aukcjonowania oraz przydzielania zadań robotom. Zaprezentowany został opis zastosowania w systemie istniejących aukcyjnych metod alokacji zadań, jak również metod opracowanych przez autora stosujących wielokryterialną ocenę przystosowania robotów do zadań. Ostatnia część rozdziału dotyczy zastosowanego na potrzeby sterowania ruchem robotów podejścia behawioralnego, wykorzystującego pola potencjalne jako podstawę algorytmów sterowania.

Rozdział czwarty dotyczy weryfikacji działania systemu oraz wykazania słuszności sformułowanych tez. Na początku tego rozdziału przedstawiony jest obiekt badań wery-

fikacyjnych, tj. grupa robotów wirtualnych będących odpowiednikami robotów rzeczywistych. Następnie wprowadzono miary służące ocenie działania systemu, jak również zaproponowano plan badań weryfikacyjnych. Eksperymenty weryfikacyjne przeprowadzono w dwóch etapach. W pierwszym etapie porównano działanie systemu stosującego aukcjonowanie zadań i aukcjonowanie robotów. Testy zostały wykonane dla mało licznych konfiguracji robotów i zadań. W drugim etapie przeprowadzono badania porównawcze metod alokacji zadań stosujących wielokryterialną ocenę przystosowania robotów do zadań oraz znanych aukcyjnych metod alokacji zadań. W tym wypadku testy wykonano dla wielu zadań i licznych grup robotów. Rozdział kończy podsumowanie przeprowadzonych badań weryfikacyjnych.

Praca zakończona jest podsumowaniem całokształtu badań przeprowadzonych przez autora. Przedstawiono także wnioski oraz kierunki dalszych badań.



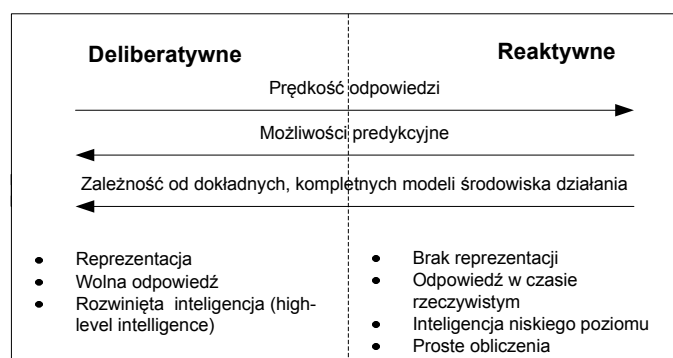
## Rozdział 2

# Sterowanie behawioralne oraz metody alokacji zadań w systemach wielorobotowych

W niniejszym rozdziale zamieszczony został syntetyczny przegląd istniejących rozwiązań w zakresie podejść stosowanych w sterowaniu ruchem robotów, podstawowych aspektów systemów wielorobotowych, w tym metod alokacji zadań stosowanych w takich systemach.

### 2.1. Sterowanie deliberatywne, reaktywne i hybrydowe

Dotychczas na potrzeby sterowania robotów mobilnych opracowanych zostało wiele podejść - od podejścia deliberatywnego do reaktywnego. Rys.2.1 przedstawia szerokie spektrum systemów sterowania robotów mobilnych, z których większość wykorzystuje elementy charakterystyczne dla podejścia deliberatywnego lub reaktywnego, lub też dla obu podejść - wtedy taki system klasyfikuje się jako hybrydowy.



Rys. 2.1: Spektrum systemów sterowania robotów mobilnych (Arkin, 1998)

Deliberatywne systemy sterowania robotów mobilnych, aby właściwie przewidzieć re-

zultaty swoich działań, wymagają znacznej wiedzy dotyczącej środowiska, w którym te roboty operują. Poza tym, aby wnioskowanie oparte na modelu świata rzeczywistego było efektywne, konieczne jest przyjęcie wielu założeń, tj. wiedza służąca wnioskowaniu powinna charakteryzować się spójnością, pewnością oraz niezawodnością. Wiedza niedokładna, czy też przestarzała, jest źródłem niewłaściwego działania układu wnioskowania systemu sterowania. Otoczenie, w którym zazwyczaj funkcjonują roboty mobilne, charakteryzuje się dużą zmiennością, stąd planowanie oparte o przestarzałe informacje może być nieefektywne, a nawet niebezpieczne. Dlatego modele środowiska działania robotów budowane są w oparciu o wiedzę *a priori* o tym środowisku, jak również o aktualne informacje napływające z układów sensorycznych robotów. Deliberatywne systemy sterowania charakteryzują się hierarchiczną strukturą, w której zadanie sterowania jest dzielone przez coraz niższe poziomy (warstwy) systemu na coraz mniejsze (bardziej szczegółowe) podzadania - horyzonty, zarówno przestrzenne jak i czasowe, stają się coraz krótsze. Komunikacja oraz wymiana informacji pomiędzy warstwami systemu odbywa się w określony z góry sposób.

Na drugim biegunie stosowanych systemów sterowania robotów mobilnych znajdują systemy reaktywne. W przeciwieństwie do systemów deliberatywnych, systemy reaktywne nie operują na żadnej wiedzy dotyczącej środowiska otaczającego robota, a więc też nie planują trajektorii ruchu. Działanie algorytmów sterowania systemów reaktywnych opiera się na prostej zasadzie *akcji i reakcji*, np.:

Jeśli nastąpi zderzenie z prawej strony, skręć w lewo,  
Jeśli nastąpi zderzenie z lewej strony, skręć w prawo,  
W przeciwnym przypadku, jedź prosto.

a więc do sterowania wykorzystywane są tylko bieżące odczyty układów sensorycznych robota. Do zalet (Turek, 2009) systemów reaktywnych zalicza się ich prostotę implementacji, wysoką prędkość odpowiedzi oraz przejrzystość algorytmów sterowania, a także niską złożoność obliczeniową. Z kolei jako wady (Matarić, 2007) podaje się najczęściej:

- Wąskie pole zastosowań, np. zadania polegające na bezkolizyjnej eksploracji zadanego obszaru,
- Wykluczenie możliwości uczenia systemu,
- Brak jakiegokolwiek pamięci systemu,
- Niedoskonałości algorytmów sterujących, np. ruch oscylacyjny robota w pobliżu wielu znajdujących się blisko siebie przeszkód.

Pośrodku spektrum systemów sterowania robotów mobilnych (Rys. 2.1) znajdują się architektury hybrydowe, które łączą w sobie elementy sterowania deliberatywnego, jak również reaktywnego. Architektury hybrydowe często określa się jako *architektury trójwarstwowe* składające się z:

- Niskiej warstwy (reaktywna),
- Wysokiej warstwy (planująca),

- Warstwy pośredniej (łączącej dwie warstwy wcześniejsze).

Najtrudniejszym zadaniem podczas budowy hybrydowego systemu sterowania jest takie opracowanie jego warstwy pośredniej (Matarić, 2007), aby:

- Kompensowała ograniczenia charakterystyczne dla systemów deliberatywnych oraz reaktywnych,
- Operowała na różnych reprezentacjach otoczenia robota,
- Uwzględniała różne wymagania czasowe warstwy wysokiej oraz niskiej.

## 2.2. Sterowanie behawioralne

Ostatnim omawianym podejściem stosowanym w sterowaniu robotów mobilnych jest podejście bazujące na zachowaniach (ang. *behavior-based*). Systemy behawioralne lokuje się pomiędzy systemami reaktywnymi a hybrydowymi. Systemy behawioralne, podobnie jak systemy reaktywne, nie operują na wiedzy *a priori* o otoczeniu, ale uwzględniają jego reprezentację. Składają się one z szeregu modułów (zachowań), których nazwy, a co za tym idzie, zadania, które wykonują, wynikają z funkcjonalności systemu. Przykładami zachowań są *omijaj-przeszkody*, *podążaj-wzdłuż-ściany*, *buduj-mapę*. Działanie systemów behawioralnych polega głównie na właściwej koordynacji zachowań.

W trakcie budowy behawioralnych systemów sterowania należy znaleźć odpowiedzi na następujące pytania (Arkin, 1998):

- jakie są podstawowe zachowania pozwalające na sterowanie robotem?
- jakie elementy składowe system powinien zawierać (zachowania, złożenia zachowań i in.)?
- jak właściwie koordynować zachowania?
- jakie są związki pomiędzy zachowaniami a układami sensorycznymi i wykonawczymi robota?

## 2.3. Koordynacja zachowań

Behawioralny system sterowania, aby kontrolować działanie robota mobilnego, wykorzystuje zazwyczaj kilka lub więcej zachowań. W celu właściwego sterowania robotem konieczna jest koordynacja tych zachowań. Ogólnie przyjmuje się, że odpowiedź systemu  $\rho$  definiowana jest w następujący sposób (Arkin, 1998):

$$\rho = \mathbf{C}(\mathbf{G} * \mathbf{B}(\mathbf{S})) \text{ lub } \rho = \mathbf{C}(\mathbf{G} * \mathbf{RSP}) \quad (2.1)$$

gdzie, w danej chwili czasu  $t$ :

- $\mathbf{S} = [s_1, s_2, \dots, s_n]$  jest wektorem bodźców istotnych dla kolejnych zachowań,
- $\mathbf{B} = [\beta_1, \beta_2, \dots, \beta_n]$  określa aktywne zachowania,

- $\mathbf{G} = [g_1, g_2, \dots, g_n]$  jest wektorem współczynników charakterystycznych dla kolejnych zachowań. Współczynniki te określają relatywną „istotność” zachowań,
- $\mathbf{RSP} = [rsp_1, rsp_2, \dots, rsp_n]$  jest wektorem odpowiedzi kolejnych zachowań  $\mathbf{B}$  na bodźce  $\mathbf{S}$ .

Odpowiedź systemu  $\rho$  w danej chwili czasu  $t$  jest wynikiem koordynacji  $\mathbf{C}$  odpowiedzi  $\mathbf{RSP}$  zachowań  $\mathbf{B}$  na bodźce  $\mathbf{S}$ . Odpowiedzi zachowań  $\mathbf{RSP}$  skalowane są za pomocą współczynników  $\mathbf{G}$  pozwalających na określanie wzajemnej „istotności” zachowań. Globalna odpowiedź systemu  $\rho$  reprezentowana jest w tej samej postaci, co odpowiedzi poszczególnych zachowań  $rsp_i$ , np.  $[\varphi, v]$ , gdzie  $v$  określa żądaną prędkość a  $\varphi$  kierunek ruchu robota.

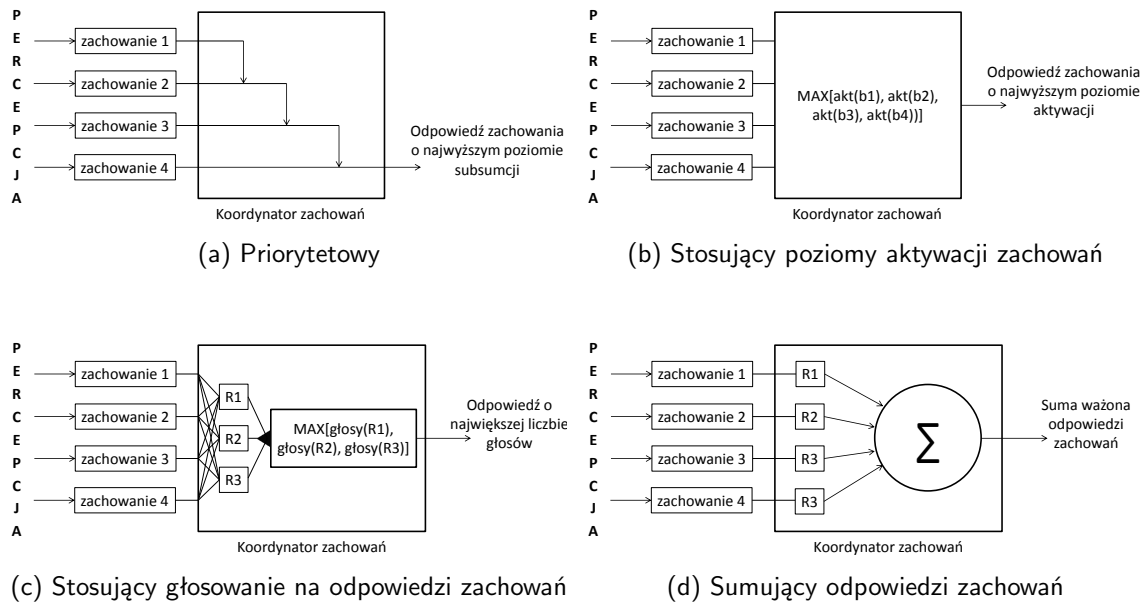
Funkcja koordynacji  $\mathbf{C}$  może być zdefiniowana w dowolny sposób, jednakże istnieje kilka powszechnie używanych metod koordynacji zachowań. Dzielą się one na dwie podstawowe grupy: konkurencyjne i kooperatywne. W przypadku metod konkurencyjnych spośród wszystkich odpowiedzi  $\mathbf{RSP}$  wybierana jest tylko jedna odpowiedź  $rsp_i$  i następnie zwracana ona jest jako globalna odpowiedź systemu  $\rho$ . Z kolei metody kooperatywne łączą odpowiedzi wszystkich zachowań. Najpopularniejszą metodą łączenia odpowiedzi zachowań jest dodawanie wektorów odpowiedzi.

### 2.3.1. Konkurencyjne mechanizmy koordynacji zachowań

Konkurencyjne mechanizmy koordynacji zachowań działają według zasady *zwycięzca bierze wszystko*, zgodnie z którą jako odpowiedź systemu sterowania przekazywana jest odpowiedź jednego, zwycięskiego, zachowania. Funkcja koordynacji, wybierająca jedną odpowiedź, może przyjmować różne formy, a do najczęściej dotychczas stosowanych należą (Arkin, 1998):

- Koordynacja ze stałymi priorytetami zachowań (Rys. 2.2a) – pomiędzy zachowaniami istnieje stała hierarchia realizowana poprzez mechanizmy blokowania (ang. *inhibition*) lub zamiany (ang. *suppression*) sygnałów niższych warstw poprzez sygnały warstw wyższych. Jest to podstawowa zasada architektury sterowania zwanej *Subsumption architecture* (Brooks, 1986),
- Koordynacja w oparciu o poziomy aktywacji zachowań (Rys. 2.2b) (Laue i Röfer, 2004) – wybór zwycięskiego zachowania następuje na podstawie jego poziomu aktywacji, zależnego od aktualnego celu działania robota i informacji napływających z systemu sensorycznego. Wybrane zostaje zachowanie o najwyższym poziomie aktywacji, a między zachowaniami nie istnieje stała hierarchia,
- Koordynacja poprzez głosowanie (Rys. 2.2c) – zachowania posiadają określoną liczbę głosów (istnieje hierarchia zachowań), które przeznaczają na jedną z predefiniowanych odpowiedzi systemu (akcji). Przykładami zachowań (Rosenblatt, 1997) mogą być *goal-seeking*, *obstacle-avoidance* i in., a przykładami odpowiedzi systemu *hard-left*, *soft-right*, *straight-ahead*. Akcja mająca w danej chwili czasu najwięcej głosów steruje silnikami robota.





Rys. 2.2: Mechanizmy koordynacji zachowań (Arkin, 1998)

### 2.3.2. Kooperatywne mechanizmy koordynacji zachowań

Alternatywą dla konkurencyjnych mechanizmów koordynacji zachowań są mechanizmy kooperatywne. Ich podstawową cechą jest jednoczesne łączenie odpowiedzi wielu zachowań. Tutaj istotnym zadaniem w trakcie budowy systemu sterowania jest opracowanie formalizmu pozwalającego na łączenie odpowiedzi zachowań.

Powszechnie stosowanym (Arkin, 1989; Koren i Borenstein, 1991; MacKenzie *i in.*, 1997) środkiem reprezentacji zachowań są pola potencjalne (ang. *potential-field*) pozwalające na generowanie gładkich trajektorii ruchu robotów. Badania nad zastosowaniem pól potencjalnych w robotyce zapoczątkowane zostały przez M. A. Arbibą i D. House'a (Arbib, 1981; Arbib i House, 1987) podczas modelowania mechanizmu pożerania ofiary przez żabę w wypełnionym przeszkodami (płotem) środowisku. Zaproponowali oni następujące pola wektorowe: atrakcyjne (przyciągające) dla ofiary i nieatrakcyjne (odpychające) dla płotu. Połączenie tych pól dało model zachowania w znacznej mierze zgodny z eksperymentalnymi obserwacjami zachowania żaby.

Jako odpowiedź w dowolnym punkcie każdego pola potencjalnego generowany jest wektor, którego zwrot wskazuje żądany kierunek ruchu robota, a wartość określa żądaną prędkość ruchu. Funkcja generująca wektor odpowiedzi może być dowolna, jednak w literaturze można znaleźć kilka powszechnie stosowanych funkcji.

W przypadku kooperatywnych mechanizmów łączenia zachowań wykorzystujących pola potencjalne koordynacja polega na ważonym sumowaniu wektorów odpowiedzi każdego z pól (Rys. 2.2d). Wagami są współczynniki  $g_i$  charakterystyczne dla każdego zachowania. Pozwalają one na hierarchizację zachowań. Należy podkreślić, że w danej chwili nie jest generowane całe pole potencjalne dla każdego zachowania, lecz tylko wektor odpowiedzi dla jednego punktu, w którym znajduje się aktualnie robot. Dzięki temu

metoda charakteryzuje się znaczną szybkością.

Z racji tego, że wektory pól potencjalnych generowane są w dowolnym punkcie przestrzeni działania robota, liczba możliwych reakcji robota jest nieskończona. Jest to przyczyną kilku problemów (Koren i Borenstein, 1991):

- utknięcie robota w punkcie spowodowane występowaniem minimum lokalnego, np. wewnątrz przeszkody w kształcie litery U,
- niemożność przejazdu robota pomiędzy dwoma znajdującymi się blisko siebie przeszkodami,
- ruch oscylacyjny robota w pobliżu przeszkody,
- ruch oscylacyjny robota w trakcie przejazdu wąskim korytarzem.

Celem zapobieżenia tym problemom stosowane są przykładowo pola potencjalne czasowo zależne, pola adaptacyjne, pola generujące wektory losowe, pola (*avoid-past*) wykorzystujące mapy najbliższego otoczenia robota zawarte w tzw. pamięci krótkotrwałej (ang. *Short-Term Memory, STM*) (Balch i Arkin, 1993; Arkin i Balch, 1997) pozwalające na omijanie dopiero co odwiedzonych miejsc, i in.

### 2.3.3. Architektury behawioralnych systemów sterowania

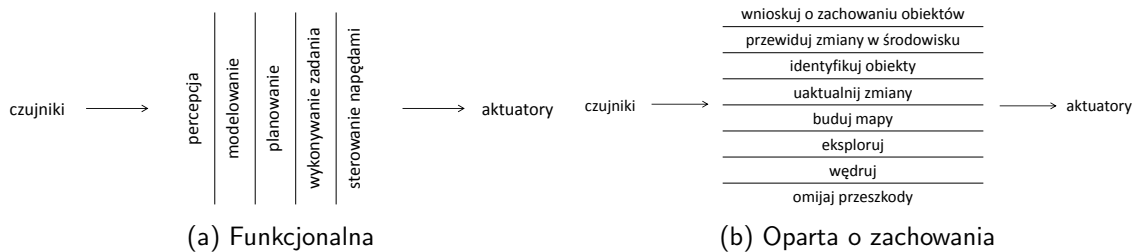
Poniżej przedstawione są jedne z pierwszych architektur sterowania behawioralnych systemów sterowania robotów mobilnych. Pierwszą z nich jest *Subsumption Architecture* (Brooks, 1986) a drugą *Motor Schemas* (Arkin, 1987). Mimo że zostały one opracowane w latach 80-tych XX wieku, są często podstawą innych współczesnych architektur, w tym systemów wielorobotowych. Przykładami mogą być architektury BLE (ang. *Broadcast of Local Eligibility*) (Werger i Matarić, 2000), ALLIANCE (Parker, 1998), czy też architektura systemu sterowania grupą robotów biorących udział w zawodach ligi RoboCup (Vail i Veloso, 2003),

#### Subsumption architecture

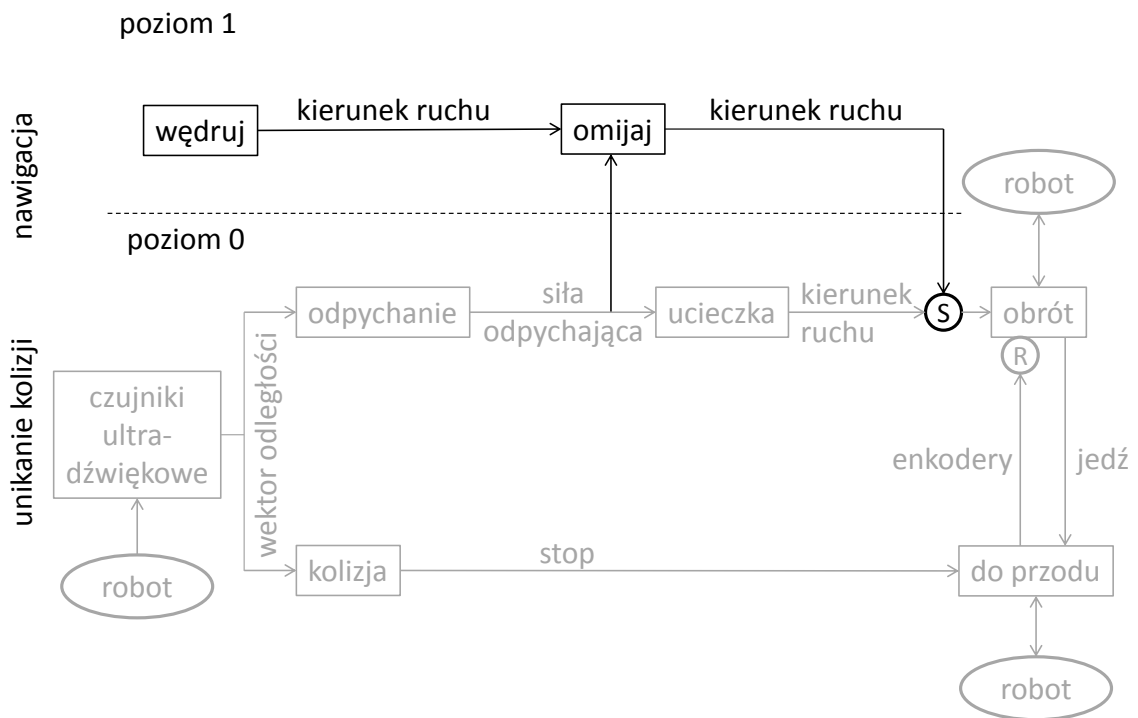
Jako pierwszą całościowo reaktywną behawioralną architekturę sterowania robotów mobilnych uznaje się *Subsumption Architecture*<sup>1</sup> opracowaną przez R. Brooksa (Brooks, 1986), która wykorzystuje konkurencyjny mechanizm koordynacji zachowań (Rys. 2.2a). Jej powstanie było odpowiedzią na wady wcześniejszych, deliberatywnych, architektur sterowania: niska niezawodność, trudność rozbudowy systemu sterowania, konieczność modelowania. Podstawową cechą odróżniającą tę architekturę od wcześniejszych jest metoda dekompozycji systemu sterowania (Rys. 2.3). Działanie pierwszych (Rys. 2.3a) systemów sterowania robotów mobilnych polegało na wykonywaniu następujących po sobie czynności wynikających z funkcjonalnej dekompozycji systemu. Dane pochodzące z układu sensorycznego przepływają kolejno przez moduły funkcjonalne (detekcji, reprezentacji/modelowania, planowania, wykonywania działań, sterowania) systemu sterowania.

<sup>1</sup>Wg słownika PWN termin subsumpcja oznacza podporządkowanie pojęcia mniej ogólnego bardziej ogólnemu, stąd w pracy używane jest pojęcie *architektura uogólniająca*.

Pomiędzy modułami mogą występować pętle sprzężenia zwrotnego. Niewłaściwe funkcjonowanie dowolnego modułu uniemożliwia działanie całego systemu - system podatny jest na jednopunktowe uszkodzenia.



Rys. 2.3: Dekompozycja systemu sterowania (Brooks, 1986)



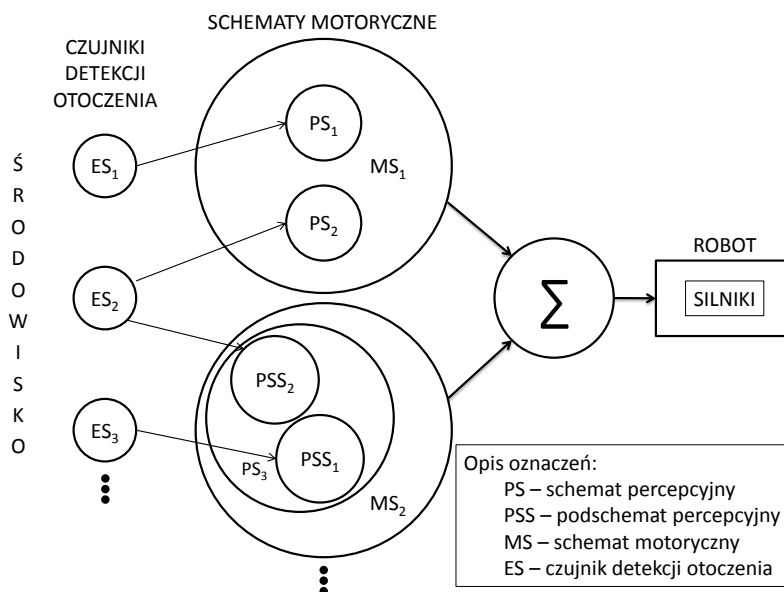
Rys. 2.4: Zasada działania architektury Brooks'a (Brooks, 1986)

Z kolei architektura Brooks'a zakłada dekompozycję systemu opartą o zachowania (Rys. 2.3b) mające na celu realizację określonych zadań (*task-achieving behaviors* (Brooks, 1986)). Zachowania te, zwane poziomami kompetencji (*levels of competence* (Brooks, 1986)), są reprezentowane jako oddzielne warstwy systemu i działają w sposób niesynchronizowany. Każda warstwa przyjmuje postać rozszerzonej maszyny stanowej, gdzie wejścia/wyjścia stanów mogą być blokowane (*inhibited*) lub zastępowane (*suppressed*) przez sygnały innych (wyższych) warstw systemu. Dodatkowo każdy stan posiada zestaw zmiennych określających jego stan wewnętrzny, który w trakcie działania systemu w wyniku wystąpienia jakiegoś zdarzenia może zostać przywrócony do stanu początkowego. Najbardziej istotną cechą architektury Brooks'a jest mechanizm subsumpcji

występującej pomiędzy kolejnymi warstwami systemu. Budowę systemu sterowania rozpoczyna się od przygotowania najprostszych zachowań (np. *omijaj-przeszkodę* - robot obraca się w momencie wykrycia przeszkody), które następnie są podporządkowywane wyższym (ogólniejszym) zachowaniom (np. *wędruj* - robot porusza się w sposób dowolny z jednoczesnym omijaniem przeszkód). Subsumpcja polega na tym, że (Rys. 2.4): (i) zachowania wyższych warstw korzystają z informacji dostarczanych przez warstwy niższe; (ii) sygnały wyższych warstw mogą zastępować i/lub blokować sygnały warstw niższych.

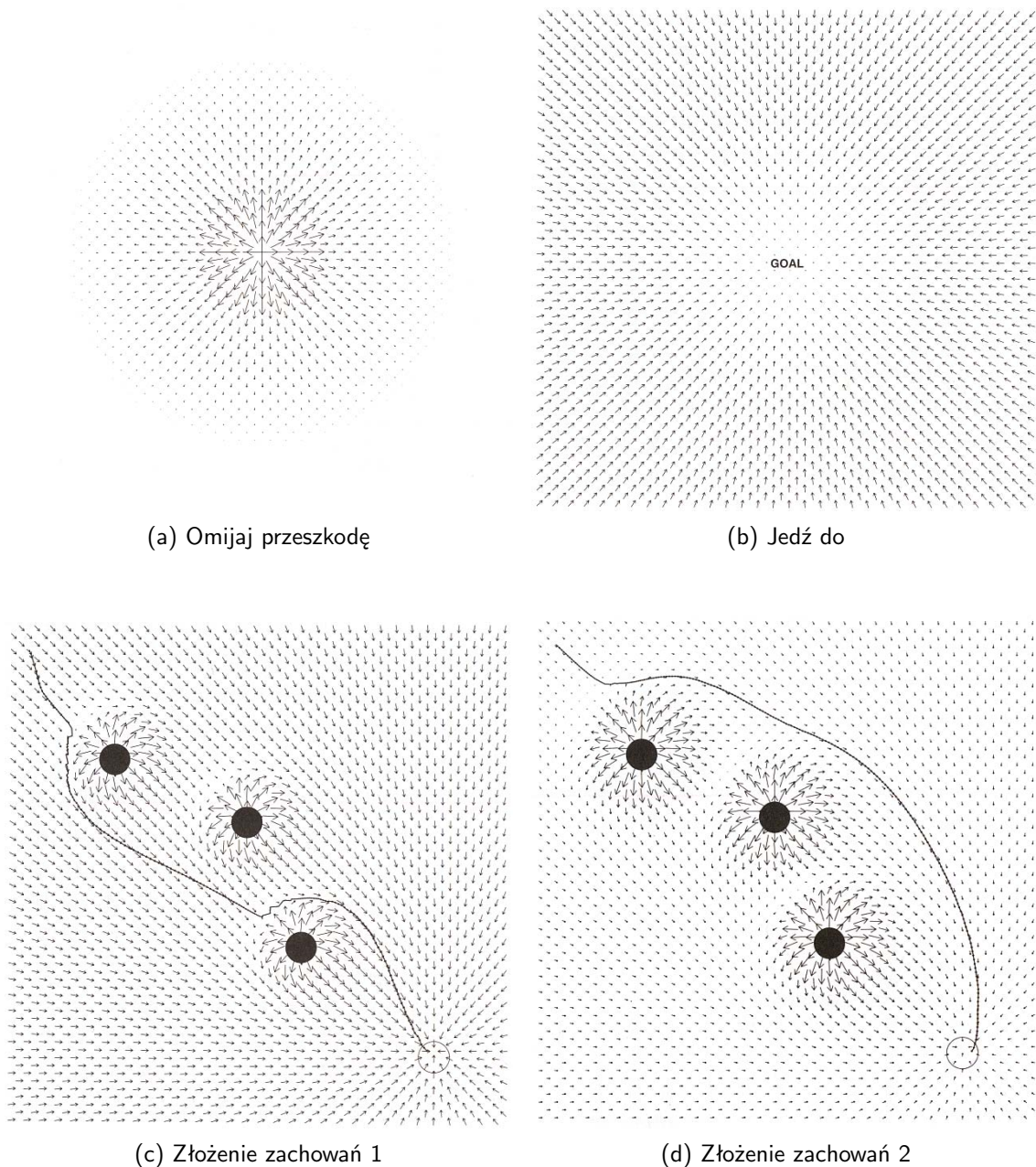
## Motor Schemas

Architektura sterowania *Motor Schemas* opracowana została przez R. Arkina (Arkin, 1987) i jest ona reaktywnym komponentem hybrydowej architektury AuRA (Arkin i Balch, 1997). *Motor Schemas* zalicza się do kooperatywnych mechanizmów koordynacji zachowań (Rys. 2.2d). Schematy motoryczne, zwane również zachowaniami, zawierają schematy percepcyjne (ang. *Perceptual Schema*) (Rys. 2.5), których celem jest dostarczanie zachowaniom informacji dotyczącej otoczenia robota. Każde zachowanie otrzymuje tylko informacje będące istotne dla jego działania. Schematy motoryczne występują na różnych poziomach systemu sterowania: zadaniem schematów wyższych poziomów jest realizowanie celów systemu planującego działanie robota, a schematy niskich poziomów odpowiedzialne są za bezpośrednie sterowanie układami wykonawczymi robota i oparte są o pola potencjalne.



Rys. 2.5: Motor schemas (Arkin, 1998)

Z modułowości zawartej w *Motor Schemas* wynika wiele korzyści (Arkin, 1998): (i) możliwe jest konstruowanie złożonych schematów motorycznych; (ii) schematy niskiego poziomu jak również ich złożenia mogą być wielokrotnie zaimplementowane na różnych poziomach systemu sterowania; (iii) schematy motoryczne mogą korzystać z kilku sche-



Rys. 2.6: Przykłady zachowań oraz ich złożzeń (Arkin, 1998)

matów percepcyjnych (przykładowo schemat percepcyjny dokonujący detekcji obecności człowieka na podstawie dwóch zagnieżdżonych schematów percepcyjnych: schematu dokonującego detekcji ruchu z wykorzystaniem termowizji, który uruchamia - w momencie wykrycia ruchu - schemat analizujący obraz wizyjny); (iv) te same zachowania w zależności od celu działania mogą wykorzystywać różne schematy percepcyjne (zachowanie *omijaj-przeszkodę* służy omijaniu robota lub statycznych przeszkód i w tym celu korzysta odpowiednio ze schematu percepcyjnego *wykryj-robota* lub *wykryj-przeszkodę*).

Rys. 2.6 przedstawia przykładowe pola potencjalne będące podstawą działania schematów motorycznych *omijaj-przeszkodę* (Rys. 2.6a) oraz *jedź-do-punktu* (Rys. 2.6b).

Rys. 2.6c oraz rys. 2.6d przedstawiają złożenia powyższych schematów. Złożenie zachowań polega na sumowaniu wektorów odpowiedzi każdego aktywnego zachowania (Rozdz. 2.3.2). W zależności od wartości współczynników  $g_i$  zachowań dominującym zachowaniem jest *jeźdź-do-punktu* (Rys. 2.6c) lub *omijaj-przeszkodę* (Rys. 2.6d).

## 2.4. Taksonomia systemów wielorobotowych

Systemy wielorobotowe posiadają wiele zastosowań, a najważniejsze z nich podano we wstępie pracy. Z mnogości istniejących rozwiązań oraz potencjalnych zastosowań grup robotów wynika różnorodność architektur systemów sterowania grup robotów. (Turek, 2009) idąc za (Dudek *i in.*, 1996) wprowadza taksonomię systemów wielorobotowych, zgodnie z którą systemy takie można podzielić ze względu na:

- Liczbę robotów w systemie – wyróżnia się systemy do sterowania jednym, kilkoma lub nieograniczoną liczbą robotów,
- Różnorodność robotów w systemie – grupa może składać się z robotów homo- lub heterogenicznych,
- Zasięg komunikacji – ograniczony, nieograniczony,
- Topologia komunikacji – określa sposób przekazywania informacji w systemie (rozgłaszanie, adresowanie, i inne),
- Przepustowość komunikacji – ograniczona lub nie,
- Możliwości obliczeniowe robotów.

## 2.5. Wymagania systemów wielorobotowych

Dla systemów wielorobotowych wyróżnia się dwie grupy wymagań: (i) wymagania funkcjonalne różne dla każdego systemu, wynikające głównie z zastosowań systemu; (ii) wymagania pozafunkcjonalne podobne dla większości systemów (Turek, 2009). Do najważniejszych wymagań określających właściwości, jakimi powinien charakteryzować się system (nie tylko system sterowania) robota (-ów), należą (Parker, 1994; Brooks, 1986; Turek, 2009):

- Rozszerzalność (ang. *extensibility*) – możliwość rozbudowywania systemu o kolejne moduły, podsystemy, możliwość dodawania nowych typów robotów, realizowania nowych typów zadań,
- Skalowalność (ang. *scalability*) – zdolność do przetwarzania coraz większej ilości informacji, zadań, jak również zdolność do sterowania grupą o zmieniającej się liczności,
- Niezawodność (ang. *reliability*) – niska podatność na uszkodzenia mechaniczne, elektroniczne, programowe,
- Odporność na uszkodzenia (ang. *fault tolerance*) – zdolność do kontynuowania działania w przypadku uszkodzeń systemu,

- Wielokrotne wykorzystanie elementów systemu (ang. *reusability*) – możliwość wykorzystania tych samych komponentów, modułów systemu dla różnych robotów.

## 2.6. Kooperacja a koordynacja w systemach wielorobotowych

Koordynacja i kooperacja są to pojęcia, które należy wyraźnie rozróżnić, szczególnie w kontekście systemów wielorobotowych. Koordynacja jest definiowana na wiele sposobów (Sariel, 2007): (i) jako zarządzanie zależnościami pomiędzy czynnościami ; (ii) umiejętność podejmowania przez agenta decyzji dotyczących własnych działań z uwzględnieniem działań innych agentów; jako proces mający na celu spójne działanie grupy agentów, a odbywający się na podstawie wnioskowania każdego agenta opartego o obecne i przewidywane działania swoje i innych agentów . Z kolei kooperacja jest rozumiana jako działanie zespołu zmierzające do osiągnięcia wspólnego celu, z tym że członkowie zespołu współpracują ze sobą a nie współzawodniczą. W (desJardins *i in.*, 1999) dosyć wyraźnie rozróżniono kooperację od koordynacji poprzez wprowadzenie dwóch terminów: kooperatywne rozproszone planowanie (ang. *Cooperative Distributed Planning, CDP*) oraz negocjacyjne rozproszone planowanie (ang. *Negotiated Distributed Planning, NDP*). W podejściu CDP agenty mają wspólny cel, wspólną reprezentację. Zdarza się, że podstawowym celem działania agentów jest opracowanie jednego, nadrzędnego planu, jednak zazwyczaj celem jest współdziałanie agentów, mających własne (indywidualne) plany, które są na bieżąco aktualizowane. Proces ten odbywa się na podstawie ciągłej wymiany informacji pomiędzy agentami o ich planach. Z kolei podejście NDP zakłada, że nadrzędnym celem działania każdego agenta jest osiągnięcie na drodze negocjacji jak największych indywidualnych korzyści.

Bardzo ineteresującym obszarem badań w zakresie koordynacji systemów wielorobotowych jest tzw. ścisła koordynacja (ang. *tight coordination*). Jak podaje (Kalra, 2007)), ścisła koordynacja zachowań robotów jest jednym z najtrudniejszych podejmowanych obecnie problemów w zakresie sterowania grupami robotów, szczególnie w warunkach rzeczywistych. Zadania rozwiązywane w ramach badań nad takimi systemami skupiają się głównie na rozwiązywaniu problemów rozproszonego planowania działań w grupie agentów (robotów). Z uwagi na działanie robotów mobilnych w środowiskach odznaczających się dużą zmiennością konieczne jest posiadanie przez roboty indywidualnych zdolności reagowania na nieprzewidywane zdarzenia, jak również zdolność grupy do częstego modyfikowania planu jej działania. Przykładami zadań wymagających ścisłej koordynacji robotów jest koordynowane przenoszenie, przesuwanie obiektów (Gerkey i Matarić, 2002), czy też układanie kostki Rubika przez dwa roboty przemysłowe (Zieliński *i in.*, 2006).

## 2.6.1. Sterowanie nadrzędne oraz zdecentralizowane

### Nadrzędne systemy sterowania

Nadrzędne systemy sterowania wykorzystują pojedynczego agenta jako centralną jednostkę planującą działania całego zespołu. Jest to model architektury, który można zaliczyć do kategorii *sense-model-plan-act* (Parker, 1998), zwykle składający się z trzech poziomów (warstw): planowania, sterowania nadrzędnego i sterowania reaktywnego. W nadrzędnym systemie sterowania agent w trakcie procesu planowania musi brać pod uwagę informacje o środowisku działania zespołu, o stanie robotów, jak również o wzajemnych interakcjach pomiędzy członkami grupy robotów. Podstawową zaletą nadrzędnych systemów sterowania jest możliwość znalezienia rozwiązania optymalnego dla postawionego zadania (misji). Jednak, jak podaje (Kalra, 2007), w praktycznych zastosowaniach znalezienie rozwiązania optymalnego w akceptowalnym czasie możliwe jest tylko w przypadku grup składających się z co najwyżej kilku robotów. Z tego powodu stosuje się inne rozwiązania, np. systemy zdecentralizowane, które pozwalają na znacznie szybsze znalezienie rozwiązania bliskiego optymalnemu. Nadrzędne systemy działają wolno, ponieważ ciągle gromadzone informacje są przesyłane ze wszystkich robotów do centralnej jednostki sterującej, gdzie następuje modyfikacja planu działania zespołu, a następnie zachodzi konieczność przesłania z powrotem zaktualizowanej informacji do robotów. Proces ten zazwyczaj obejmuje środowisko odznaczające się dużą niepewnością, słabą przewidywalnością, a ponadto powinien zapewniać podejmowanie decyzji w czasie rzeczywistym<sup>2</sup> i dlatego pociąga za sobą znaczne wymagania obliczeniowe jednostki centralnej i komunikacyjne całego systemu. Wadą systemów nadrzędnych jest znaczna podatność systemu na wielo- , a nawet jednopunktowe uszkodzenia

### Zdecentralizowane systemy sterowania

#### Kooperacja nieświadomiona

Bardzo popularnym modelem kooperacji stosowanym w budowie systemów sterowania robotów jest model oparty o inteligencję roju (ang. *swarm intelligence*), w którym inteligencja grupy nie wynika z działania mechanizmów współpracy w grupie, a raczej z interakcji pomiędzy pojedynczymi jej członkami. Kooperacja w takich systemach określana jest często jako *emergent cooperation*. Inspiracją dla takich systemów są systemy biologiczne (zwierzęce), np. ławice ryb. Właściwie skonstruowany system oparty o inteligencję roju może być najprostszym rozwiązaniem dla jakiegoś zadania, jednak zwykle systemy takie przeznaczone do wykonywania wybranych zadań – są mało uniwersalne (Gerkey i Mataric, 2003b). Podejście oparte na inteligencji roju stosowane jest zwykle w budowie systemów sterowania bardzo licznych grup robotów homogenicznych posiadających te same lub zbliżone umiejętności.

<sup>2</sup>Zgodnie z (Kasprzyk, 2006), aby sterownik pracował w czasie rzeczywistym, jego reakcja w postaci obliczonego sterowania w odpowiedzi na zmianę wejścia musi wystąpić w określonym czasie, akceptowalnym z punktu widzenia wymagań stawianych temu sterowaniu.



### Kooperacja uświadomiona

Drugim modelem kooperacji szeroko stosowanym na potrzeby budowy systemów wielorobotowych jest model kooperacji uświadomionej (ang. *intentional cooperation*). W takich systemach w grupie występuje bezpośrednia komunikacja między jej członkami, dzięki której powstaje plan działania całej grupy. Ten rodzaj współpracy jest bardziej zbliżony do współpracy w zespołach ludzkich, aniżeli w grupach zwierząt. Systemy zdecentralizowane oparte o kooperację uświadomioną stosowane są raczej dla grup robotów heterogenicznych. Bardzo popularnym kierunkiem badań w robotyce jest konstruowanie systemów wielorobotowych współpracujących z zespołami ludzkimi, albo nawet mogących konkurować z ludźmi – silnie rozwijany jest projekt RoboCup, którego dalekosiężnym celem (zaplanowanym na 2050. rok) jest opracowanie grupy robotów (drużyny piłkarskiej) potrafiącej zwyciężyć *ludzką* drużynę piłkarską (Burkhard *i in.*, 2002).

## 2.7. Systemy wieloagentowe a wielorobotowe

Badania służące rozwojowi systemów wieloagentowych (ang. *Multi-Agent Systems, MAS*) oraz systemów wielorobotowych (ang. *Multi-Robot Systems, MRS*) prowadzone są od wielu lat niezależnie. Ponadto, w ostatnim czasie badania te zbiegają do coraz szerszej implementacji metod charakterystycznych dla systemów wieloagentowych w systemach wielorobotowych (Gerkey i Mataric, 2004; Scerri *i in.*, 2003), co nie jest zadaniem łatwym przede wszystkim ze względu na różną postać agentów/robotów oraz środowisko ich działania (Sariel, 2007) – roboty funkcjonują w świecie rzeczywistym, podczas gdy agenty, będące twórcami abstrakcyjnymi, są elementami programów komputerowych. Stąd w systemach wielorobotowych nie można stosować bezpośrednio (bez żadnych modyfikacji) założeń przyjmowanych dla systemów wieloagentowych, oraz w przypadku systemów wielorobotowych należy uwzględniać aspekty świata rzeczywistego. Często w literaturze robotyki określane są mianem *situated agent*<sup>3</sup> (Dias, 2004). Zasadniczym wyzwaniem w trakcie budowy systemu sterowania robota(-ów) jest zapewnienie systemowi możliwości prowadzenia obliczeń i interakcji z otoczeniem. R. Brooks (Brooks, 1991) wymienia następujące aspekty podejścia behawioralnego związane z operowaniem robotów w świecie rzeczywistym<sup>4</sup>:

- umiejscowienie (ang. *situatedness*) - robot nie operuje na abstrakcyjnej reprezentacji świata rzeczywistego, a ponadto jest obiektem umiejscowionym w tym świecie,
- upostaciowienie (ang. *embodiment*) - robot ma postać fizyczną, co pociąga za sobą konieczność modelowania dynamicznych interakcji pomiędzy robotem a obiektami znajdującymi się w jego otoczeniu,
- przypadkowość (ang. *emergence*) - zwłaszcza w przypadku behawioralnych systemów sterowania inteligencja nie jest własnością robota, czy też jego otoczenia, a wynika z ich wzajemnych interakcji.

<sup>3</sup><http://www.ryerson.ca/~dgrimsha/courses/cps720/agentDef.html>

<sup>4</sup>tłumaczenie własne

Istnieje wiele różnic pomiędzy robotami a agentami programowymi (Dias, 2004). Zadania wykonywane przez roboty są inne w stosunku do tych realizowanych przez agenty. Czas, zasięg działania robotów - szczególnie mobilnych - ograniczone są posiadanymi zasobami energetycznymi i komunikacyjnymi, podczas gdy zasoby agentów zazwyczaj są nieograniczone. Podobnie roboty mają ograniczone możliwości detekcji otoczenia ze względu na cechy systemów sensorycznych, tj. zasięg, opóźnienia, szумы, itp. Roboty są znacznie bardziej podatne na uszkodzenia. Różnorodność układów (mechaniczne, napędowe, elektroniczne, komunikacyjne, sensoryczne), a w konsekwencji źródeł możliwych niesprawności, jest przyczyną spadku niezawodności robota. Ponadto zwiększona częstotliwość uszkodzeń robotów - w stosunku do uszkodzeń agentów - jest spowodowana przez niesprzyjające warunki pracy (niska/wysoka temperatura, nadmierna wilgotność, zapylenie) robotów, oddziaływania dynamiczne pomiędzy robotami a ich otoczeniem, jak i zwykle niższą niezawodność poszczególnych elementów (np. komputerów). Z uwagi na przytoczone różnice, sterowanie systemów wielorobotowych i wieloagentowych w wielu aspektach jest odmienne.

## 2.8. Alokacja zadań w systemach wielorobotowych

Optymalny przydział zadań w grupie robotów jest jednym z ważniejszych problemów badawczych, nie tylko w robotyce mobilnej. Dyscypliną ściśle związaną z tym zagadnieniem są badania operacyjne. Poszukiwanie rozwiązań za pomocą metod znanych z badań operacyjnych jest czasochłonne ze względu na znaczną złożoność obliczeniową algorytmów stosowanych przez te metody. Również przybliżone metody programowania liniowego, np. metoda węgierska, metoda simplex, w zastosowaniach systemów wielorobotowych służących wykonywaniu dużej liczby zadań nie sprawdzają się ze względu na ich dużą złożoność obliczeniową.

Metody znane z badań operacyjnych, czy też metody programowania liniowego byłyby przydatne dla systemów, w których od samego początku znana byłaby konfiguracja robotów i zadań w misji, a dodatkowo pozostawałaby ona niezmienna w ciągu trwania tej misji (Zaborowski, 2008). Wtedy uzasadnione byłoby stosowanie tych metod. Jednak w praktycznych zastosowaniach, w których w trakcie misji pojawiają się nowe zadania, a roboty ulegają uszkodzeniom, przez co porzucają wykonywane przez siebie zadanie, metody te są nieprzydatne. Dlatego prowadzone są badania nad metodami i systemami pozwalającymi na szybkie znajdowanie rozwiązań bliskich rozwiązaniom optymalnym. Można wyróżnić dwa główne nurty badań, tj.:

- Badania nad systemami, w których przydział zadań w grupie robotów jest realizowany z zastosowaniem podejścia behawioralnego,
- Badania w ramach których rozwijane są ekonomiczne (aukcjonowanie, negocjacje) metody alokacji zadań.

Poniżej zamieszczono krótki opis dwóch przykładowych znanych systemów sterowania stosujących podejście behawioralne (ALLIANCE) oraz aukcyjne (Murdoch).

ALLIANCE (Parker, 1998) jest behawioralną rozproszoną architekturą systemu sterującego robotami za pomocą zbioru zachowań służących wykonywaniu określonych zadań typu *znajdź-objekty*, *zbieraj-objekty*, *raportuj-postęp-zadania*. Aktywacja lub deaktywacja tych zachowań następuje za pomocą skojarzonych z nimi tzw. zachowań motywacyjnych (ang. *motivational behaviors*) na podstawie dwóch parametrów, tj. *niecierpliwości* (ang. *impatience*) i *przyzwolenia* (ang. *acquiescence*). Pierwszy parametr wyraża niezadowolenie robota wynikające z braku postępu w realizacji wybranego zadania przez innego robota – im większa wartość tego parametru, tym większa motywacja robota do przejścia tego zadania. Z kolei drugi parametr wyraża gotowość robota do porzucenia wykonywanego przez niego zadania. Wartość tego parametru rośnie wraz z czasem, w którym robot nie odnotowuje postępu realizowanego przez siebie zadania, np. na skutek awarii lub utknięcia w jakimś miejscu. Poziom aktywacji zachowania motywacyjnego zależy od tych dwóch parametrów. Dodatkowo zachowania motywacyjne mają możliwość wpływania (obniżania) na wartość poziomu aktywacji innych zachowań. Przekroczenie określonej wartości progowej przez poziom aktywacji wybranego zachowania motywacyjnego powoduje aktywowanie skojarzonego z nim zachowania (zbioru zachowań) służącego wykonywaniu określonych zadań.

Przykładem systemu stosującego mechanizm aukcyjny bazujący na Contract-Net Protocol (Smith, 1980) jest system Murdoch (Gerkey i Matarić, 2002). System ten był stosowany w zadaniu koordynowanego przesuwania obiektu (pudła) przez trzy roboty, z których jeden pełnił rolę koordynatora tego zadania, a pozostałe dwa miały za zadanie przesuwać to pudło. Charakterystyczną cechą tego systemu jest możliwość dokonania hierarchicznej dekompozycji zadań.

Działanie systemów aukcyjnych polega na przydzielaniu zadań po przeprowadzeniu dla nich aukcji. Dla każdego pojawiającego się w systemie zadania przeprowadzana aukcja składa się z następujących kroków:

- Ogłoszenie aukcji – menadżer aukcji (system nadrzędny, robot) wysyła ogłoszenie (zapytanie ofertowe) o zadaniu do wykonania do innych robotów. Ogłoszenie zawiera informacje o zadaniu, m. in. typ zadania, identyfikator, czas wprowadzenia, wymagane umiejętności (zasoby) robotów,
- Wyznaczenie oceny przystosowania – ogłoszenie zawiera również informację o stosowanym kryterium oceny przystosowania robota do zadania, na podstawie którego każdy robot biorący udział w aukcji ma oszacować swoje „dopasowanie” do zadania,
- Złożenie oferty – każdy robot, jeśli może podjąć się zadania będącego przedmiotem aukcji, przesyła wiadomość zwrotną (ofertę na wykonanie zadania) do menedżera aukcji. Oferta zawiera informację o dopasowaniu robota do zadania,
- Zamknięcie aukcji – po określonym czasie aukcja zostaje zamknięta, menadżer na podstawie analizy ocen przystosowania robotów do zadania wyłania zwycięzcę. Zwycięzca otrzymuje ograniczony czasowo kontrakt na wykonanie zadania. Pozostałe roboty mogą brać udział w licytacji innych zadań,

- Monitorowanie zadania/odnawianie kontraktu – menedżer aukcji, której przedmiotem było to zadanie, nadzoruje jego postęp aż do chwili, kiedy to zadanie zostanie wykonane.

Podstawą wyboru robota, któremu przydzielane jest zadanie, jest kryterium mówiące o przystosowaniu robota do zadania. Najczęściej kryteriami są czas, energia potrzebne na dotarcie do miejsca wykonywania zadania i jego wykonanie, czy też dystans dzielący robota i miejsce wykonywania zadania. W przypadku systemu Murdoch kryterium wyboru robota mającego pchać jeden z brzegów pudła był procentowy udział koloru (jaki miało pudło) w obrazie rejestrowanym przez kamerę robota. Oceny przystosowania robotów do zadań mogą być również ocenami wielokryterialnymi.

## 2.9. Podsumowanie

Rozdział ten miał na celu przedstawienie obszaru badań tej pracy. Zwrócono uwagę na następujące zagadnienia związane z systemami wielorobotowymi, tj.:

- Podejścia stosowane w sterowaniu ruchem robotów, począwszy od reaktywnego, poprzez behawioralne i hybrydowe, aż po deliberatywne,
- Podstawowe architektury behawioralnych systemów sterowania,
- Mechanizmy koordynacji zachowań,
- Podejścia stosowane w sterowaniu grupami robotów,
- Podstawowe metody alokacji zadań w grupie robotów.

Analiza wybranych pozycji literatury pozwala na stwierdzenie, że w przypadku heterogenicznej grupy robotów inspekcyjnych, która jest obiektem badań niniejszej pracy, istnieje możliwość skonstruowania systemu sterowania, który mógłby połączyć zalety wielu znanych rozwiązań i zapewnić sprawną możliwość działania tej grupy przy ograniczonych zasobach obliczeniowych będących środowiskiem sprzętowym działania systemu sterowania. Tak sformułowane zadanie zostało podjęte przez autora i opisane w kolejnych rozdziałach pracy.

## Rozdział 3

# System sterowania grupą robotów inspekcyjnych

W niniejszym rozdziale zamieszczono opracowany przez autora formalny opis systemu sterowania grupą robotów inspekcyjnych oraz opis metod alokacji zadań, z których większość zostało zaproponowanych przez autora.

### 3.1. Założenia

Nie jest możliwe zbudowanie uniwersalnego systemu, który pozwalałby na sterowanie wszystkimi znanymi rodzajami robotów wykonującymi dowolne zadania w dowolnym środowisku. Dlatego, przed zdefiniowaniem modelu systemu sterowania, należy przyjąć założenia, które w głównej mierze wynikają z funkcjonalności systemu, a także uwzględniają ograniczenia podsystemów robotów.

#### Przeznaczenie systemu

- System ma umożliwiać prowadzenie misji przez grupę heterogenicznych robotów inspekcyjnych – rozważa się zadania typu monitorowanie terenu, detekcja szkodliwych gazów, usuwanie niebezpiecznych substancji, poszukiwanie osoby rannej, zbieranie niebezpiecznych ładunków,
- System ma wspomagać operatora w podejmowaniu decyzji – zakłada się, że w sytuacjach wysokiego zagrożenia, kiedy w krótkim czasie pojawia się wiele zadań do wykonania, w wyniku przeciążenia informacyjnego operator będzie dokonywał nieefektywnych (dalekich od optymalnych) przydziałów zadań w grupie robotów.

#### System

- System jest systemem nadrzędnym – istnieje jednostka centralna zarządzająca zbiorami robotów i zadań, która ma dostęp do wszystkich informacji zawartych w modelach robotów i zadań,
- Podstawą systemu alokacji zadań jest mechanizm aukcyjny,

- Nie istnieją problemy komunikacyjne – zakłada się, że nie istnieją ograniczenia charakterystyczne dla stosowanych obecnie w robotyce mobilnej systemów komunikacyjnych, tj. ograniczony zasięg, straty danych, przerwy w komunikacji.

### Środowisko działania robotów

- Środowisko działania robotów jest zdeterminowane – dla każdej misji znana jest mapa środowiska,
- Środowisko działania robotów jest trójwymiarowe – w implementacji systemu zaprezentowanej w pracy w środowisku symulacyjnym roboty modelowane są jako trójwymiarowe obiekty, będące wirtualnymi odpowiednikami rzeczywistych robotów - zachowane są ich wymiary, masy. Również inne obiekty (budynki, przeszkody) środowiska są obiektami trójwymiarowymi,<sup>1</sup>
- Topografia przestrzeni działania robotów jest ograniczona do dwóch wymiarów – położenie robotów definiowane jest za pomocą trzech  $(x, y, \theta)^2$ , a położenie miejsc wykonywania zadań za pomocą dwóch  $(x, y)$  współrzędnych. Ponadto w topografii uwzględnia się wyróżnione punkty, pomiędzy którymi przemieszczają się roboty. Siatka tych punktów uwzględnia dostępną mapę środowiska, a służy głównie wyznaczaniu ścieżek ruchu robotów.

### Roboty

- Grupa składa się z robotów heterogenicznych – zróżnicowanie robotów wynika z ich mobilności (napęd gąsienicowy/kołowy, różne prędkości maksymalne ruchu), posiadanych zasobów energetycznych, zużycia zasobów energetycznych. Ponadto zakłada się, że każdy z robotów wyposażony jest w urządzenia (zasoby) umożliwiające wykonywanie zadań inspekcyjnych. Liczba i rodzaj tych zasobów określone są indywidualnie dla każdego robota,
- Roboty mogą być różnych typów, gdzie typ wskazuje na przeznaczenie (rolę) robota. Ponadto typ robota determinuje zbiór zasobów inspekcyjnych robota,
- Roboty mogą pojawić się w systemie w dowolnej chwili, w dowolnej liczbie i dowolnego typu,
- Roboty wyposażone są w system lokalizacji pozwalający na określenie położenia (pozycja i orientacja). W pracy nie są rozpatrywane algorytmy lokalizacji,
- Do sterowania ruchem robotów wykorzystywane jest podejście behawioralne,
- Nie ma możliwości doładowywania akumulatorów robota w trakcie misji - robot działa do momentu wyczerpania się akumulatorów. Ostatnia zarejestrowana pozy-

<sup>1</sup>W większości prowadzonych dotychczas badań z zastosowaniem symulatorów (np. (Gnatowski, 2005; Zaborowski, 2008; MacKenzie *i in.*, 1997)) przestrzeń działania robotów była dwuwymiarowa, roboty reprezentowane były za pomocą punktów (obiektów o zerowej masie i zerowych wymiarach).

<sup>2</sup>Wielkość  $\theta$  określa kąt obrotu wokół osi prostopadłej do płaszczyzny  $XY$ .

cja robota przed wyładowaniem akumulatorów uważana jest za aktualną pozycję robota nieaktywnego.

### Zadania

- Istnieje jeden zbiór zadań dla całej grupy robotów,
- Zadanie jest określone przez parametry: miejsce wystąpienia, szacowany czas wykonywania, potrzebne zasoby robotów (energetyczne, sprzętowe),
- Nie są rozważane problemy realizacji zadań — zakłada się, że zadanie jest zrealizowane, jeśli robot je wykonujący znajduje się w miejscu występowania zadania przez czas równy szacowanemu czasowi wykonywania zadania,
- Każde zadanie może być wykonane tylko przez jednego robota – nie zakłada się wykonywania zadań przez kilka robotów, których suma zasobów inspekcyjnych pozwala na wykonie zadania,
- Zadania w systemie mogą pojawić się w dowolnym momencie, w dowolnej liczbie i różnych typów (wymagających różnych zbiorów zasobów inspekcyjnych).

## 3.2. Formalna reprezentacja systemu

### 3.2.1. Konfiguracja misji

W każdej chwili działania wejścia systemu (Rys. 3.1) reprezentowane są za pomocą zależności:

$$\mathbb{C} = (TO, R, Z, MP), \quad (3.1)$$

gdzie:

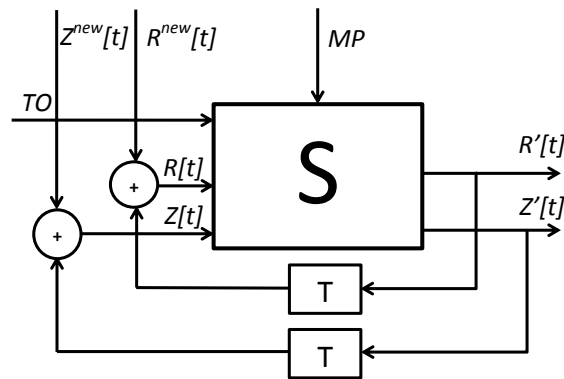
$TO$  - ograniczona topografia przestrzeni (3.3),

$R$  - zbiór robotów biorących udział w misji (3.15),

$Z$  - zbiór zadań do wykonania w trakcie misji (3.18),

$MP$  - parametry misji (3.21).

Zależność ta jest zapisem formalnym aktualnej w każdej chwili **konfiguracji misji**. Ograniczona topografia przestrzeni  $TO$ , zawierająca informacje o węzłach, w których mogą znajdować się zadania oraz pomiędzy którymi przemieszczają się roboty, pozostaje niezmienna wraz z postępem misji, natomiast zawartości zbiorów  $R$  i  $Z$  podlegają zmianom wynikającym ze zdarzeń zachodzących w systemie. Parametry misji  $MP$  przechowują dane wykorzystywane przez metody aukcjonowania i przydziału zadań. Szczegółowy opis wszystkich parametrów konfiguracji misji  $\mathbb{C}$  zawarty jest w dalszej części pracy.



Rys. 3.1: Model systemu sterowania grupą robotów inspekcyjnych

### 3.2.2. Model środowiska

W ogólnym przypadku przestrzenią działania robotów jest trójwymiarowa przestrzeń rzeczywista, a więc do jednoznacznego opisu ich położenia wymaganych jest sześć zmiennych (3 współrzędne do określenia położenia, 3 kąty do określenia orientacji). Parametrami opisującymi miejsce występowania zadań są również współrzędne tej przestrzeni. Aby móc określić położenie robotów i zadań w punktach tej przestrzeni, czy też zapewnić robotom możliwość przemieszczania pomiędzy tymi punktami, należy zdefiniować topografię przestrzeni (Zaborowski, 2008). Zgodnie z przyjętymi założeniami systemu (Rozdz. 3.1) topografia przestrzeni opisana jest dwoma parametrami  $(x, y)$  za pomocą poniższej zależności:

$$T = X \times Y = \{(x, y) : x, y \in \mathbb{R}\}, \quad (3.2)$$

Celem określenia położenia robota konieczne jest wprowadzenie dodatkowego parametru  $\theta$  określającego orientację robota. W przypadku zadań istotna jest tylko ich pozycja  $(x, z)$ , zaś orientacja  $(\theta)$  nie ma znaczenia, dlatego nie uwzględnia się jej w modelu zadania (3.12).

#### Ograniczona topografia przestrzeni

Kryteriami stosowanymi podczas aukcjonowania i przydziału zadań są czas i energia potrzebne na realizację zadań, których wartości wynikają m. in. z odległości pomiędzy miejscami realizacji zadań a robotami. Model topografii przestrzeni zdefiniowany za pomocą (3.2) pozwala na prawidłowe funkcjonowanie systemu sterowania w przypadku, gdy przestrzeń ta nie jest zajmowana przez inne obiekty (poza robotami) - punkt początkowy i końcowy aktualnej ścieżki robota znajdują się zawsze na jego linii widzenia (ang. *line of sight*). W przypadku, gdy przestrzeń  $T$  pomniejszona jest o obszary zajęte przez te obiekty, konieczne jest uwzględnienie w modelu  $T$  dodatkowych ograniczeń.

W związku z powyższym wprowadzono pojęcie *ograniczona topografia przestrzeni* i oznaczono je przez  $TO$ . Poza uwzględnieniem dodatkowych obiektów w przestrzeni działania robota,  $TO$  redukuje możliwe miejsca realizacji zadań tylko do wybranych punktów, a także możliwe trasy robotów do odcinków między tymi punktami. Reprezentacją  $TO$



(3.3) w modelu systemu jest graf nieskierowany ważony będący grafem geometrycznym, w którym wierzchołki  $v_i$  opisane są współrzędnymi przestrzeni  $T$ , a waga  $w_{i,j}$  krawędzi  $e_{i,j}$  równa jest odległości pomiędzy wierzchołkami tej krawędzi.

$$TO = (V, E, w), \quad (3.3)$$

gdzie:

$V$  – zbiór wierzchołków grafu ( $V = \{v_i : v_i = (x_i, y_i), i \in \{0\} \cup \mathbb{N}, x, y \in \mathbb{R}\}$ ),

$E \subset V \times V$  – zbiór krawędzi grafu będący rodziną dwuelementowych podzbiorów wierzchołków  $V$  (3.4),

$w$  – funkcja przypisująca krawędzi  $e_{i,j} \in E$  wagę  $w_{i,j}$  będącą odległością między wierzchołkami tej krawędzi (3.5).

$$E = \{e_{i,j} = (v_i, v_j) : v_i, v_j \in V, i, j \in \{0, 1, \dots, |V| - 1\}, i \neq j\} \quad (3.4)$$

$$w : E \rightarrow \mathbb{R}^+ \\ w_{i,j} = w(e_{i,j}) = \begin{cases} 0 & , \text{jeśli } i = j \\ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} & , \text{jeśli } e_{i,j} = \{v_i, v_j\} \in E \\ \infty & , \text{jeśli } e_{i,j} = \{v_i, v_j\} \notin E \end{cases} \quad (3.5)$$

### Wybór środka reprezentacji ograniczonej topografii przestrzeni

Podstawowymi środkami reprezentacji grafów stosowanymi w algorytmach grafowych (przeszukiwanie grafów, poszukiwanie najkrótszej ścieżki i inne) są macierz sąsiedztwa, lista sąsiedztwa oraz macierz incydencji. Każda z tych struktur danych ma swoje wady i zalety w stosunku do konkretnych zastosowań. Najważniejsze cechy ułatwiające wybór któregoś z wymienionych środków są zebrane w Tab. 3.1<sup>3</sup>. Graf będący reprezentacją  $TO$  opisywanego systemu jest grafem rzadkim, tzn.  $|E| \ll |V|^2$ . W takim przypadku – ze względu na niską złożoność pamięciową ( $O(n)$ ,  $n$ -liczba krawędzi grafu) – zalecanym środkiem reprezentacji grafu jest lista sąsiedztwa. Dodatkową zaletą, przemawiającą za zastosowaniem listy sąsiedztwa, jest możliwość szybkiej identyfikacji wierzchołków połączonych krawędzią z danym wierzchołkiem (wiersz *Przejrzenie sąsiadów danego wierzchołka* w Tab. 3.1). Ma to szczególny wpływ na czas realizacji algorytmów wymagających szybkiego określenia sąsiadów wybranego wierzchołka, na przykład algorytm Dijkstry,  $A^*$ .

W związku z powyższym na potrzeby kodu systemu sterowania jako środek reprezentacji grafu *ograniczonej topografii przestrzeni* wybrano listę sąsiedztwa. W przypadku opisywanego systemu lista ta jest wektorem  $L$  (3.6) o wymiarze  $n$ , gdzie  $n$  oznacza liczbę wierzchołków grafu  $TO$ , czyli  $|V|$  (3.3). Indeksy elementów wektora  $L$  odpowiadają numerom wierzchołków grafu  $TO$ . Każdy element wektora jest listą dwuelementowych

<sup>3</sup>Dane zawarte w tabeli zaczerpnięto z <http://www.cs.put.poznan.pl/arybarczyk/GrafReprezentacje.htm>

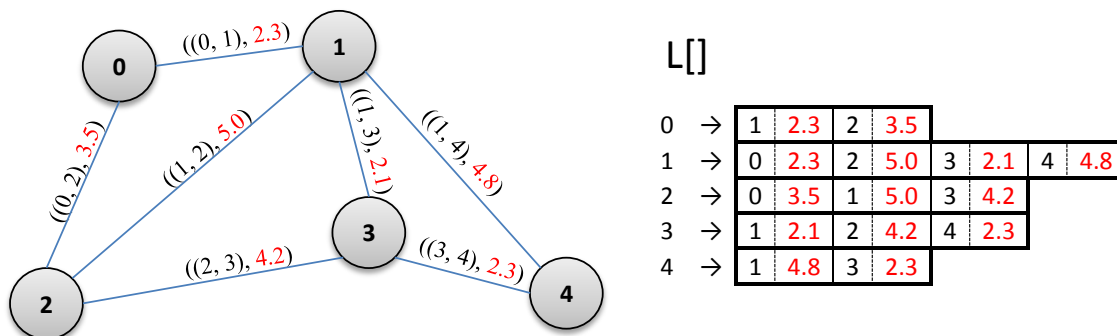
struktur zawierających numery wierzchołków i odległości do nich, z którymi dany wierzchołek o  $i$ -tym indeksie posiada krawędzie. Przykład ilustrujący sposób konstruowania wektora  $L$  zamieszczony jest na Rys.3.2.

Tab. 3.1: Porównanie reprezentacji grafów

Cecha	Macierz sąsiedztwa	Lista sąsiedztwa	Macierz incydencji
Złożoność pamięciowa	$O(m^2)$	$O(n)$	$O(m * n)$
Przejrzenie wszystkich krawędzi	$O(m^2)$	$O(n)$	$O(n)$
Przejrzenie sąsiadów danego wierzchołka	$O(m)$	max. $O(m)$ średnio $O(n/m)$	$O(n)$
Sprawdzenie istnienia jednej krawędzi	$O(1)$	max. $O(m)$ średnio $O(n/m)$	$O(m)$

$m = |V|, n = |E|$

$$L_i = \begin{cases} \{(j, w_{i,j})\} & , \text{jeśli } w_{i,j} \in \mathbb{R}^+ \\ \emptyset & , \text{w przeciwnym przypadku} \end{cases} \quad (3.6)$$



Rys. 3.2: Lista sąsiedztwa

Weryfikacja systemu sterowania przeprowadzona została z wykorzystaniem testowych środowisk symulacyjnych, dla których wygenerowane zostały grafy *ograniczonej topografii przestrzeni*. Szczegółowy opis algorytmu pozwalającego na otrzymanie grafu  $TO$  na podstawie wirtualnego modelu środowiska symulacyjnego znajduje się w (Wiglenda, 2009).

### 3.2.3. Model zasobów inspekcyjnych

Zbiór  $res^{insp}$  (3.7) zawiera informacje o posiadanych przez robota lub wymaganych dla realizacji zadania zasobach, tj. urządzeniach, które umożliwiają prowadzenie zadań inspekcyjnych. Każdy element tego zbioru zawiera parametr przechowujący informację o typie urządzenia ( $dev^{insp}$ ), może przyjmować przykładowe wartości: `webcam` – kamera wideo, `presence` – czujnik obecności, `temperature` – czujnik temperatury otoczenia, `manipulator` – urządzenie pozwalające na przenoszenie niebezpiecznych ładunków, `CO`

– czujnik tlenku węgla. Drugi z parametrów określa wartość zużycia energii zasobu (urządzenia) przypadającą na jednostkę czasu. Dla większości urządzeń (szczególnie czujników) możliwe jest łatwe określenie jednostkowego zużycia energii, jednak dla niektórych należy przyjąć założenia upraszczające. Przykładem może być manipulator, którego zużycie energii w znacznej mierze zależy od liczby aktywnych w danej chwili napędów jego układu napędowego. W takim przypadku właściwym rozwiązaniem wydaje się przyjęcie średniego jednostkowego zużycia energii, wyznaczonego przykładowo na podstawie dostępnych danych z historii eksploatacji urządzenia lub po przeprowadzeniu odpowiednich eksperymentów.

$$res^{insp} = \{r_i^{insp} = (dev_i^{insp}, ec_i^{insp}) : ec_i^{insp} \in \mathbb{R}^+, i \in \{0\} \cup \mathbb{N}\} \quad (3.7)$$

Zawartość zbioru  $res^{insp}$  określona jest indywidualnie dla każdego robota oraz zadania. W wielu praktycznych zastosowaniach można wyróżnić typy zadań inspekcyjnych (np. detekcja szkodliwych gazów, patrolowanie wyznaczonego obszaru) oraz role robotów (np. robot rozminowujący, sprzątający), jakie pełnią w systemie. Dlatego zasadnym postępowaniem jest przyjęcie zestawów zasobów inspekcyjnych (co zrobiono w części weryfikacyjnej systemu) pozwalających na realizację wybranych typów zadań, czy też charakterystycznych dla konkretnych ról robotów. Oczywiście, jeden robot może pełnić więcej ról, np. patrolowanie i sprząkanie. Ponadto, role robotów mogą się całkowicie lub częściowo pokrywać.

### 3.2.4. Model robota

Jednym z elementów aktualnej w danej chwili konfiguracji misji opisywanego systemu są roboty. Poniżej zebrane zostały charakterystyczne dla każdego robota parametry będące źródłem istotnych informacji dla działania systemu.

$$\begin{aligned} r &= (id^r, p, vel, res, ec, ee, et, td, let, z^{ex}, Z_r^{cd}) & (3.8) \\ &= (id^r, \\ &\quad (p = (x^{act}, y^{act}, \theta^{act}, \theta^{trg}, V^r)), \\ &\quad (vel = (vel^{act}, vel^{trg})), \\ &\quad (res = (res^{insp}, bl, vel^{max})), \\ &\quad (ec = (ec^{sb}, ec^{trav}, ec^{ex})) \\ &\quad (ee = (ee^{nb}, ee^{trav}, ee^{ex})) \\ &\quad (et = (et^{nb}, et^{trav}, et^{ex})) \\ &\quad td, let, z^{ex}, (Z_r^{cd} \subset Z^{cd})) \end{aligned}$$

gdzie:

$id^r \in \{0\} \cup \mathbb{N}$  – indeks robota,

$p$  – wektor określający aktualne  $(x^{act}, y^{act}, \theta^{act})$  i docelowe  $(V_0^r, \theta^{trg})$  w danej chwili położenie robota, oraz kolejkę punktów do odwiedzenia przez robota  $V^r \subset V$ ,

$vel \in R$  – wektor zawierający informacje o aktualnej  $vel^{act}$  i docelowej  $vel^{trg}$  prędkości liniowej robota,

$res$  – zasoby robota,

$ec$  – zużycie energii przypadające na jednostkę czasu, a wynikające z działania podstawowych układów robota  $ec^{sb}$ , dojazdu do miejsca wykonywania zadania  $ec^{trav}$ , wykonywania zadania  $ec^{ex}$ ,

$ee$  – sumaryczna energia zużyta w trakcie oczekiwania robota na przydział zadań  $ee^{nb}$ , dojazd do nich  $ee^{trav}$  oraz ich wykonywanie  $ee^{ex}$ ,

$et$  – sumaryczny czas oczekiwania robota na przydział zadań  $et^{nb}$ , dojazd do miejsca ich wykonywania  $et^{trav}$  oraz ich wykonywanie  $et^{ex}$ ,

$dt$  – przebyty dystans wynikający z dojazdu do miejsc wykonywania zadań,

$let$  – czas ostatniego zdarzenia (por. rozdz. 3.2.8),

$z^{ex}$  – aktualnie wykonywane zadanie (przyjmuje wartość null, jeśli robot nie dojeżdża do żadnego miejsca wykonywania zadania ani nie wykonuje jakiegokolwiek zadania) ( $z^{ex} \in Z^{ex}$ ),

$Z_r^{cd}$  – zbiór zadań zrealizowanych przez robota.

Każdy robot posiada unikalny indeks  $id^r$  jednoznacznie identyfikujący go w grupie. Będący w użyciu indeks nie może zostać przypisany kolejnemu dodawanemu do systemu robotowi, nawet w sytuacji kiedy robot oznaczony tym indeksem nie funkcjonuje już w systemie.

Wektor  $V^r$  zawiera podzbiór wierzchołków grafu  $TO$  i jest kolejką punktów (ścieżką), które robot  $r$  powinien odwiedzić, aby dojechać do miejsca, w którym wykonywane jest zadanie.  $V^r$  wyznaczany jest za pomocą algorytmu A\* w momencie wygrania przez agenta reprezentującego robota aukcji dotyczącej wybranego zadania. Ścieżka generowana jest na podstawie aktualnej pozycji robota  $(x^{act}, y^{act})$ , lokalizacji zadania  $(x^z, y^z)$  (3.12) oraz aktualnej konfiguracji  $TO$ . Pierwszy element wektora  $V_0^r$  wskazuje aktualny punkt docelowy ruchu robota. Współrzędne  $x_0^r, y_0^r$  tego punktu oraz  $x^{act}, y^{act}, vel^{max}$  służą jako wejścia zachowania *jeźdź-do-punktu*, które oblicza swoją składową wektora wyjściowego zachowań określonego przez dwójkę  $(vel^{trg}, \theta^{trg})$ . Ostatni element zbioru  $V^r$  wskazuje na lokalizację zadania.

Istotnymi dla funkcjonalności grupy są indywidualne dla każdego robota zasoby  $res$ . Mają one szczególne znaczenie w procesach aukcjonowania oraz przydziału zadań. Zasoby inspekcyjne bezpośrednio stanowią o możliwości prowadzenia przez robota wybranych działań inspekcyjnych - na podstawie zawartości zbioru  $res^{insp}$  podejmowana jest decyzja o przystąpieniu agenta robota do aukcji dotyczącej wybranego zadania. Z kolei aktualny poziom naładowania akumulatorów  $bl$ , prędkość maksymalna robota  $vel^{max}$  oraz zużycia energii  $ec$  mają wpływ na wartość kryteriów oceny ofert, mówiących (wartości kryteriów) o przystosowaniu robota do zadania. Procesy aukcjonowania oraz przydziału zadań szczegółowo omówione są w rozdz. 3.3 oraz 3.4.

Na zużycie energii robota  $ec$  przypadające na jednostkę czasu wpływ mają trzy składowe:

- Energia  $ec^{sb}$  zużywana na działanie podstawowych układów robota, tj. jednostki sterującej, układów I/O, systemu lokalizacji, systemu detekcji otoczenia,
- Energia  $ec^{trav}$  zużywana w trakcie dojazdu robota do miejsc wykonywanych zadań, potrzebna do zasilania układów napędowych robota,
- Energia  $ec^{ex}$  zużywana przez urządzenia (zasoby) inspekcyjne  $res_r^{insp}$  robota w trakcie wykonywania zadań inspekcyjnych. Przyjmuje się, że jeśli liczba zasobów inspekcyjnych robota  $|res_r^{insp}|$  jest większa od liczby zasobów inspekcyjnych wymaganych przez zadanie  $|res_z^{insp}|$ , to zużycie energii  $ec^{ex}$  jest sumą zużyć energii zasobów wymaganych przez zadanie, a nie są uwzględniane zasoby nadmiarowe robota.

Na potrzeby badań weryfikacyjnych w modelu robota zawarte zostały następujące grupy parametrów:

- Sumaryczny czas oczekiwania  $et^{nb}$  oraz zużyta w tym czasie energia  $ee^{nb}$ ,
- Sumaryczny czas dojazdu do zadań  $et^{trav}$ , zużyta w tym czasie energia  $ee^{trav}$  oraz przebyty dystans  $td$ ,
- Sumaryczny czas wykonywania zadań  $et^{ex}$  oraz zużyta w tym czasie energia  $ee^{ex}$ .

Dla powyższych parametrów zachodzą zależności:

$$ee^{nb} = et^{nb} \cdot ec^{sb}, \quad (3.9)$$

$$ee^{trav} = et^{trav} \cdot (ec^{sb} + ec^{trav}), \text{ gdzie } et^{trav} = td \cdot vel^{max}, \quad (3.10)$$

$$ee^{ex} = et^{ex} \cdot ec^{sb} + \sum_{l=0}^{N-1} (ec^{ex}(Z_r^{cd} \cdot res^{insp}) \cdot Z_r^{cd} \cdot \tau \cdot t^{ex}), \quad (3.11)$$

$$\text{gdzie } N = |Z_r^{cd}|.$$

### 3.2.5. Model zadania

Zadania w modelu systemu opisywane są za pomocą następujących parametrów:

$$z = (id^z, v, r^{ex}, r^{cd}, \tau, res^{insp}), \quad (3.12)$$

gdzie:

$id^z \in \{0\} \cup \mathbb{N}$  – indeks zadania,

$v = (x, y), v \in V$  – wierzchołek grafu  $TO$  określający miejsce wykonywania zadania w przestrzeni  $T$ ,

$r^{ex}$  – liczba wskazująca indeks robota wykonującego zadanie,

$r^{cd}$  – liczba wskazująca indeks robota, który zrealizował zadanie,

$\tau = (t^{in}, t^{ex}, t^{al}, t^{st}, t^{left}, t^{await}, t^{act}, t^f)$  – czas wprowadzenia do systemu  $t^{in}$ , potrzebny na wykonanie  $t^{ex}$ , ostatniego przydziału  $t^{al}$ , ostatniego rozpoczęcia wykonywania  $t^{st}$ , pozostałego czasu do wykonania  $t^{left}$ , oczekiwania  $t^{await}$ , wykonywania  $t^{act}$  oraz zrealizowania  $t^f$  zadania,

$res^{insp}$  – objaśnienie w rozdz. 3.2.3.

Podobnie jak roboty, każde zadanie jest jednoznacznie identyfikowane w systemie przez unikalny indeks  $id^z$ , który raz nadany jednemu zadaniu nie może być powtórnie przypisany innemu zadaniu.

W założeniach systemu przyjęto, że zadania mogą pojawić się tylko w wyróżnionych punktach przestrzeni  $T$ , tj. węzłach  $v_i \in V$  należących do grafu  $TO$ . Założenie takie upraszcza i skraca czas działania algorytmu wyznaczania najkrótszej ścieżki pomiędzy robotem a miejscem wykonywania zadania, nie mając jednocześnie – dzięki małemu odstępowi siatki  $TO$  w stosunku do odległości pokonywanych przez roboty – znaczącego wpływu na funkcjonalność systemu.

$$r^{ex} = \begin{cases} -2 & \text{zadanie zostało wykonane} \\ -1 & \text{zadanie nie zostało wykonane} \\ & \text{i nie jest przydzielone żadnemu robotowi} \\ k & \text{zadanie przydzielone jest robotowi o indeksie } k \end{cases} \quad (3.13)$$

$$r^{cd} = \begin{cases} -1 & \text{zadanie nie zostało wykonane } (r^{ex} > -2) \\ k & \text{zadanie zostało zrealizowane przez robota o indeksie } k \end{cases} \quad (3.14)$$

W chwili pojawienia się zadania w systemie parametr  $r^{ex}$  przyjmuje wartość  $-1$  (patrz 3.60), którą zachowuje aż do chwili powierzenia zadania robotowi, kiedy to  $r^{ex}$  przyjmuje wartość indeksu  $k$  robota. Jeśli zadanie zostanie wykonane przez robota  $k$ , to zostaje usunięte z kolejki zadań do wykonania i  $r^{ex} = -2$ . Jeśli wykonywanie zadania zostanie przerwane (awaria robota, porzucenie zadania), parametr  $r^{ex}$  przyjmuje z powrotem wartość  $-1$  i zadanie wprowadzone jest ponownie na koniec kolejki zadań do wykonania. Parametr  $r^{cd}$  przyjmuje wartość  $-1$  w momencie wprowadzenia zadania do systemu. W chwili zrealizowania zadania  $r^{cd}$  przyjmuje wartość indeksu  $k$  robota, który je wykonał.

Podstawowym kryterium oceny działania systemu jest kryterium czasowe – oceniane są czasy oczekiwania, zakończenia wykonywania pojedynczych czy też wszystkich zadań. Dlatego do modelu zadania wprowadzono szereg parametrów związanych z czasem. Umownie przyjęto  $t_0 = 0$  jako chwilę, w której następuje rozpoczęcie działania systemu. W chwili  $t_0$  wprowadzone do systemu zostają wszystkie roboty. Zadania mogą być wprowadzone przed (parametr  $t^{in} = 0$ ) i po (parametr  $t^{in}$  przyjmuje wartość czasu, jaki upłynął od rozpoczęcia działania systemu) chwili  $t_0$ . W pracy nie przywiązuje się wagi do procesu wykonywania zadań, a jedynie przyjmuje się założenie, że zadania uznaje się za zrealizowane, jeśli robot je wykonujący przebywa w miejscu wykonywania tego zadania przez okres równy zdefiniowanemu czasowi potrzebnemu na wykonanie zadania  $t^{ex}$ . Dodatkowo do modelu wprowadzono parametry  $t^{al}$ ,  $t^{st}$  oraz  $t^{act}$  oznaczające odpowiednio

czasu ostatniego przydziału, ostatniego rozpoczęcia realizacji oraz wykonywania zadania. W chwili ostatniego przydziału zadania  $t_{al} = t$ . W momencie, kiedy  $t - t^{st} = t^{ex}$ , zadanie zostaje uznane za wykonane i usunięte z kolejki zadań do wykonania, a  $t^f = t$  (wcześniej  $t^f = -1$ ). W przypadku, gdy zadanie było wykonywane, ale nie zostało zrealizowane, parametr  $t^{act}$  zostaje powiększony o wartość równą  $t - t^{st}$ , a zadanie traktowane jest tak, jakby nie było w ogóle wykonywane i umiejscowione zostaje na końcu kolejki zadań. Ostatnim parametrem związanym z czasem jest  $t^{await}$ , który w chwili rozpoczęcia realizacji zadania przyjmuje wartość  $t^{st} - t^{in}$ .

Parametr  $res^{insp}$  określa jakie zasoby inspekcyjne (opis w rozdziale 3.2.3) robota są konieczne do wykonania zadania. Robot nie posiadający takich zasobów nie może uczestniczyć w procesach aukcjonowania, przydziału, ani realizacji zadania.

### 3.2.6. Zbiór robotów $R$

W każdej chwili działania systemu można wprowadzić do niego dowolną liczbę robotów. Oznacza to, że możliwe jest prowadzenie misji, w której wszystkie roboty funkcjonują od jej początku, a także misji, podczas której do systemu wprowadza się kolejne roboty w miarę na ich zapotrzebowanie, co znacznie zbliża działanie systemu do jego rzeczywistych potencjalnych zastosowań. W pracy, aby zweryfikować proponowane metody, wystarczyło przeprowadzić eksperymenty, w których wszystkie roboty były wprowadzane do systemu z początkiem misji ( $t = 0$ ).

Wszystkie wprowadzone do systemu roboty reprezentowane są przez zbiór  $R$ , będący sumą dwóch rozłącznych podzbiorów robotów aktywnych  $R^{act}$  oraz robotów nieaktywnych  $R^{nact}$  ( $R^{act} \cup R^{nact} = R$  i  $R^{act} \cap R^{nact} = \emptyset$ ). Zbiór  $R_{act}$  jest sumą dwóch kolejnych rozłącznych podzbiorów robotów niezajętych  $R^{nb}$  oraz robotów wykonujących zadanie  $R^{ex}$  ( $R^{nb} \cup R^{ex} = R^{act}$  i  $R^{nb} \cap R^{ex} = \emptyset$ ). Zbiór  $R^{nb}$  jest **kolejką robotów** oczekujących na przydział zadania. W momencie wprowadzenia robota  $r$  do systemu umieszczony jest on na końcu kolejki robotów  $R^{nb}$  ( $R^{nb} = R^{nb} \cup \{r\}$ ).

$$R = R^{act} \cup R^{nact}, \text{ gdzie } R^{act} \cap R^{nact} = \emptyset \quad (3.15)$$

$$R^{act} = R^{nb} \cup R^{ex}, \text{ gdzie } R^{nb} \cap R^{ex} = \emptyset \quad (3.16)$$

Część opisywanych w dalszej części pracy metod podczas procesów aukcjonowania robotów i przydziału zadań uwzględnia tylko roboty niezajęte, a część wszystkie aktywne roboty. W przypadku metod opartych na kolejkowym aukcjonowaniu robotów i przydziale im zadań ważne jest, aby w pamięci komputera zbiór  $R^{nb}$  reprezentowany był jako kolejka, a więc traktowany jest jako ciąg, którego elementami są niezajęte roboty (3.17).

$$R^{nb} = \{R_i^{nb} : i \in \{0\} \cup \mathbb{N}\} \quad (3.17)$$

### 3.2.7. Zbiór zadań $Z$

Podobnie jak roboty, zadania do systemu mogą być wprowadzone w dowolnej chwili i w dowolnej liczbie (zarówno w danej chwili, jak i podczas całej misji). Wszystkie wprowadzone zadania określone są przez zbiór  $Z$ . Oprócz tego system korzysta ze zbiorów zadań wykonanych  $Z^{cd}$  i niewykonanych  $Z^{ncd}$  będących podzbiorami  $Z$ , takimi że  $Z^{cd} \cup Z^{ncd} = Z$  i  $Z^{cd} \cap Z^{ncd} = \emptyset$  (3.18). Z kolei zbiór zadań niewykonanych  $Z^{ncd}$  (3.19) jest sumą dwóch rozłącznych zbiorów zadań pozostających w **kolejce zadań** i oczekujących na wykonanie  $Z^q$  oraz zadań będących aktualnie wykonywanych  $Z^{ex}$  ( $Z^q \cup Z^{ex} = Z^{ncd}$  i  $Z^q \cap Z^{ex} = \emptyset$ ).

$$Z = Z^{cd} \cup Z^{ncd}, \text{ gdzie } Z^{cd} \cap Z^{ncd} = \emptyset \quad (3.18)$$

$$Z^{ncd} = Z^q \cup Z^{ex}, \text{ gdzie } Z^q \cap Z^{ex} = \emptyset \quad (3.19)$$

Z punktu widzenia funkcjonalności systemu najważniejszym zbiorem jest kolejka zadań  $Z^q$ , ponieważ to na nim wykonywane są operacje aukcjonowania i z niego przydzielane są zadania robotom. W przypadku aukcjonowania zadań (rozdział 3.3.5) istotne jest, aby w pamięci komputera zbiór  $Z^q$  reprezentowany był jako kolejka - zadania w trakcie przydzielania rozpatrywane są w kolejności ich występowania w tej kolejce, a więc zbiór  $Z^q$  traktowany jest jako ciąg, którego elementami są zadania oczekujące na przydział (3.20).

$$Z^q = \{Z_i^q : i \in \{0\} \cup \mathbb{N}\} \quad (3.20)$$

### 3.2.8. Działanie systemu

Podstawą działania systemu są wprowadzone w poprzednich rozdziałach modele środowiska, robotów oraz zadań, a także parametry misji (Rozdz. 3.3.1) sterujące metodami alokacji zadań. Działanie systemu polega na obsłudze zdarzeń, które mogą nastąpić (Rys. 3.3, Alg.3.1) <sup>4</sup>:

1. Na zewnątrz systemu <sup>5</sup>:

- Uruchomienie systemu,
- Aktywowanie zadania i/lub robota (aktywacja),
- Zatrzymanie systemu,

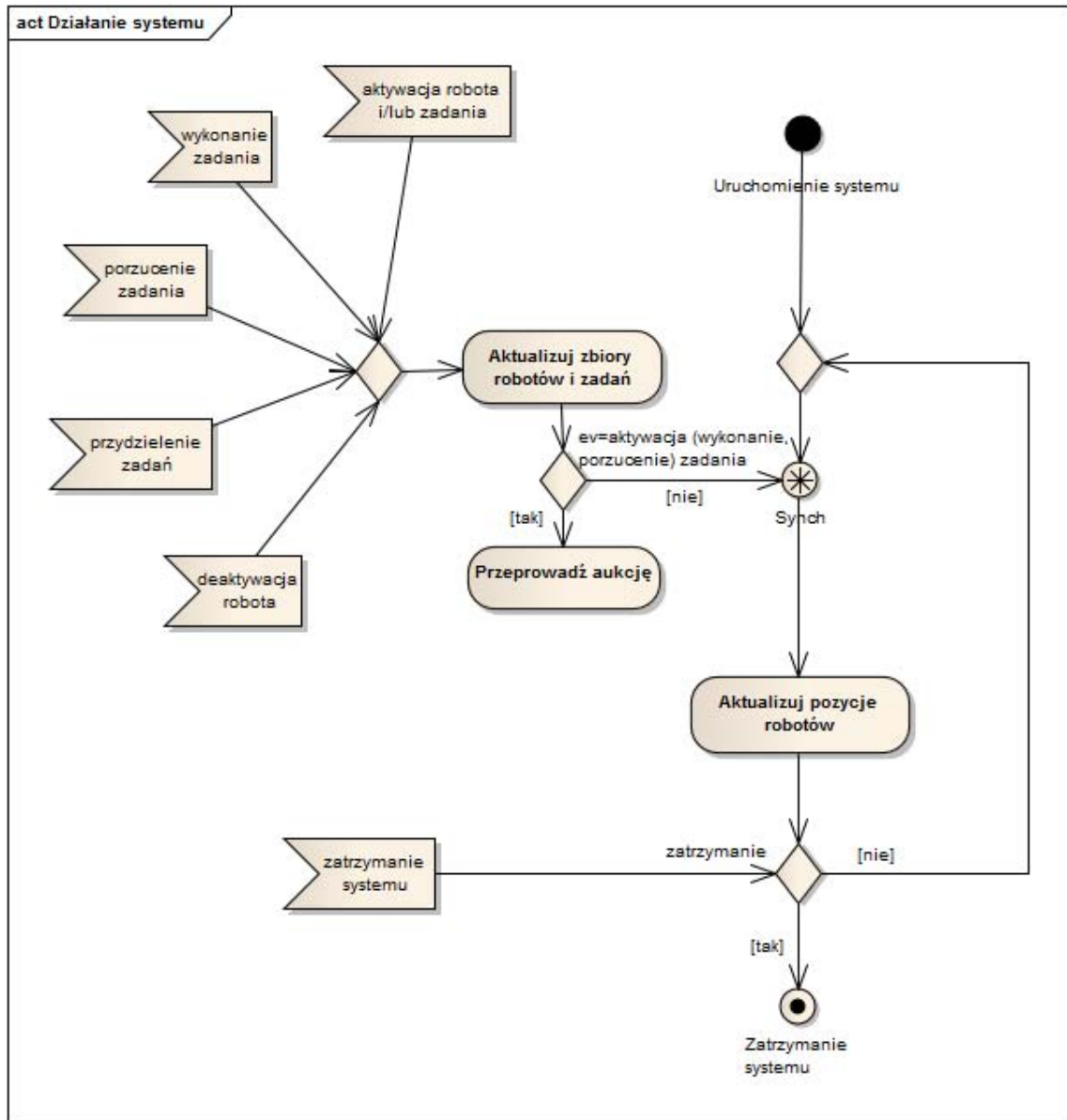
<sup>4</sup>W nawiasach podano wyrażenia użyte w algorytmie 3.1, jakie przyjmuje argument *ev*.

<sup>5</sup>Sformułowanie "na zewnątrz systemu" oznacza, że zdarzenie nie jest wywoływane przez opisywany w pracy system, a może być wywołane np. przez operatora systemu za pomocą aplikacji użytkownika dołączonej do systemu (co zostało zrobione w ramach badań weryfikacyjnych (Rozdz. 4)) albo innego – zewnętrznego – systemu.



## 2. Wewnątrz systemu:

- Przydzielenie zadania (przydział),
- Wykonanie zadania (wykonanie),
- Porzucenie zadania (porzucenie),
- Deaktywacja robota (deaktywacja).



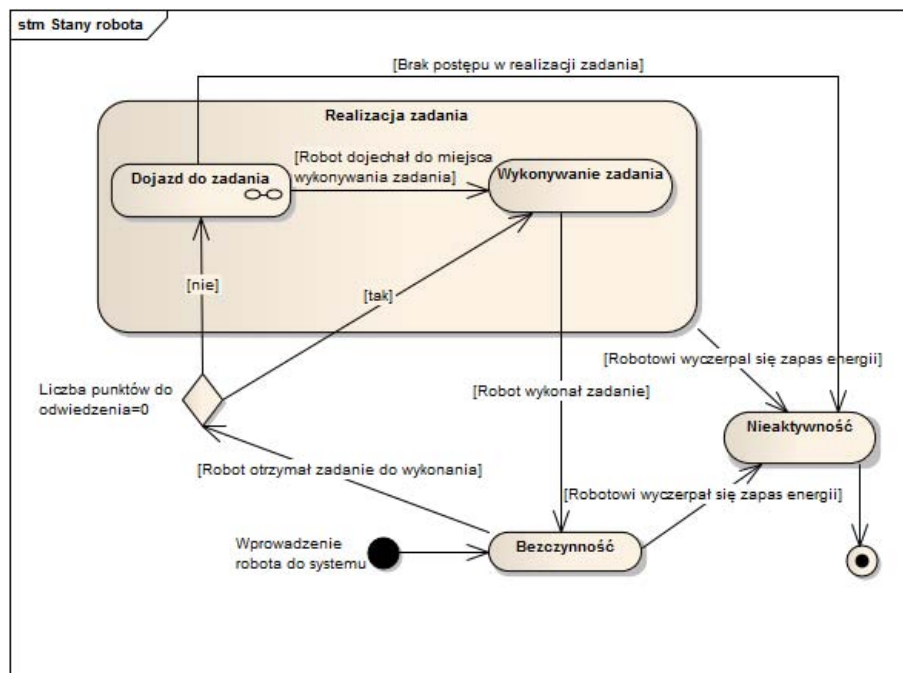
Rys. 3.3: Działanie systemu sterowania grupą robotów inspekcyjnych

Wymienione zdarzenia – poza uruchomieniem i zatrzymaniem systemu – są źródłem operacji na zbiorach robotów  $R$  i zadań  $Z$  (Alg. 3.1). W momencie wprowadzenia zadania  $z$  do systemu jest ono włączane do  $Z^q$  ( $Z^q = Z^q \cup \{z\}$ ). Przydzielenie zadania należącego do zbioru  $Z^q$  skutkuje przeniesieniem go do zbioru zadań wykonywanych  $Z^{ex}$  ( $Z^q =$

$Z^q \setminus \{z\}$ ,  $Z^{ex} = Z^{ex} \cup \{z\}$ ), w którym pozostaje aż do momentu zakończenia jego wykonywania ( $t - t_z^{st} = t_z^{ex}$ ), kiedy to zostaje przeniesione do zbioru zadań zrealizowanych  $Z^{cd}$  ( $Z^{ex} = Z^{ex} \setminus \{z\}$ ,  $Z^{cd} = Z^{cd} \cup \{z\}$ ). W przypadku porzucenia zadanie wraca do zbioru zadań oczekujących na przydzielenie  $Z^q$ , z tym że na koniec kolejki zadań reprezentującej ten zbiór ( $Z^{ex} = Z^{ex} \setminus \{z\}$ ,  $Z^q = Z^q \cup \{z\}$ ).

Aktywowanie robota skutkuje wprowadzeniem robota  $r$  do zbioru robotów niezajętych ( $R^{nb} = R^{nb} \cup \{r\}$ ). W chwili przydziału robotowi zadania jest on przeniesiony do zbioru  $R^{ex}$  ( $R^{nb} = R^{nb} \setminus \{r\}$ ,  $R^{ex} = R^{ex} \cup \{r\}$ ). Po wykonaniu zadania następuje odwrotna operacja ( $R^{ex} = R^{ex} \setminus \{r\}$ ,  $R^{nb} = R^{nb} \cup \{r\}$ ). Robot może przejść w stan nieaktywności w trzech przypadkach: (i) wyczerpania się energii w stanie bezczynności ( $R^{nb} = R^{nb} \setminus \{r\}$ ,  $R^{nact} = R^{nact} \cup \{r\}$ ), (ii) wyczerpania się energii w trakcie dojazdu lub wykonywania zadania ( $R^{ex} = R^{ex} \setminus \{r\}$ ,  $R^{nact} = R^{nact} \cup \{r\}$ ), (iii) porzucenia zadania w wyniku braku postępu w jego realizacji<sup>6</sup> ( $R^{ex} = R^{ex} \setminus \{r\}$ ,  $R^{nact} = R^{nact} \cup \{r\}$ ).

Rys. 3.4 przedstawia stany, w jakich może znaleźć się robot. Stan *Dojazd do zadania* szerzej omówiony jest w rozdz. 3.5.2.



Rys. 3.4: Diagram stanów robota

Rozpoczęcie procesu aukcjonowania może nastąpić w wyniku trzech zdarzeń (Rys. 3.3):

- aktywowania robotów i/lub zadań,

<sup>6</sup>Brak postępu w realizacji zadania może być skutkiem wywrócenia się robota lub *utknięcia* algorytmów sterowania ruchem robota w minimum lokalnym. Zgodnie z założeniami nie uwzględnia się możliwości braku postępu w realizacji zadania wynikającego z uszkodzeń lub awarii robota, co należałoby zrobić prowadząc eksperymenty w warunkach rzeczywistych.

- wykonania zadania – robot, który wykonał zadanie, staje się robotem niezajętym, a więc może podjąć się innych zadań,
- porzucenia zadania – porzucone zadanie ponownie rozpatrywane jest jako zadanie oczekujące na przydział (umieszczane jest na końcu kolejki zadań).

Metody oraz algorytmy alokacji (aukcjonowanie i przydzielanie) zadań szczegółowo omówione są w rozdziałach 3.3 oraz 3.4.

---

**Algorytm 3.1** Aktualizacja wybranych zbiorów modelu systemu
 

---

1: **Funkcja** Aktualizuj zbiory robotów i zadań

argumenty:  $r, z$  - odpowiednio: robot  $r$  i zadanie  $z$  biorące udział w zdarzeniu

$R^{nb}$  - zbiór robotów niezajętych,  $R^{new}$  - zbiór robotów do aktywowania,

$R^{ex}$  - zbiór robotów będących w trakcie wykonywania zadań

$Z^q$  - kolejka zadań,  $Z^{new}$  - zbiór zadań do aktywowania

$Z^{ex}$  - zbiór zadań wykonywanych

$MP$  - parametry misji

$ev$  - zdarzenie

zwracane:  $R^{nb}, R^{ex}, Z^{ex}, Z^q$

2: **Jeżeli**  $ev = \text{aktywacja}$

3:     **Dla wszystkich**  $r' \in R^{new}$

4:          $R^{nb} \leftarrow R^{nb} \cup \{r'\}$

5:     **Dla wszystkich**  $z' \in Z^{new}$

6:          $Z^q \leftarrow Z^q \cup \{z'\}$

7: **W przeciwnym razie, jeżeli**  $ev = \text{przydział}$

8:      $R^{nb} = R^{nb} \setminus \{r\}, R^{ex} = R^{ex} \cup \{r\}$

9:      $Z^q = Z^q \setminus \{z\}, Z^{ex} = Z^{ex} \cup \{z\}$

10: **W przeciwnym razie, jeżeli**  $ev = \text{wykonanie}$

11:      $R^{ex} = R^{ex} \setminus \{r\}, R^{nb} = R^{nb} \cup \{r\}$

12:      $Z^{ex} = Z^{ex} \setminus \{z\}, Z^{cd} = Z^{cd} \cup \{z\}$

13: **W przeciwnym razie, jeżeli**  $ev = \text{porzucenie}$

14:      $R^{ex} = R^{ex} \setminus \{r\}, R^{nact} = R^{nact} \cup \{r\}$

15:      $Z^{ex} = Z^{ex} \setminus \{z\}, Z^q = Z^q \cup \{z\}$

16: **W przeciwnym razie, jeżeli**  $ev = \text{deaktywacja}$

17:      $R^{nb} = R^{nb} \setminus \{r\}, R^{nact} = R^{nact} \cup \{r\}$

18: **Jeżeli**  $ev = \text{aktywacja} \parallel ev = \text{wykonanie} \parallel ev = \text{porzucenie}$

19:     **Wywołaj** *Przeprowadź aukcję*

20:     argumenty:  $R^{nb}, R^{ex}, Z^q, MP$

---

### 3.3. Metody alokacji zadań oparte o Contract-Net Protocol

W pracy na potrzeby działania systemu zaimplementowano powszechnie stosowane metody alokacji zadań oparte o Contract-Net Protocol ((Smith, 1980))<sup>7</sup>, a także opracowano oraz zaimplementowano metody oparte o wielokryterialną ocenę przystosowania robotów do zadań (metody MCFV<sup>8</sup>(ang. *Multi-Criterial Fitness Value*)). Stąd rozdział ten spełnia dwa cele:

- Prezentuje implementację metod CNP w systemie,
- Wprowadza pojęcia oraz algorytmy wspólne dla wszystkich omawianych w pracy metod alokacji zadań.

#### 3.3.1. Parametry misji

Na konfigurację misji  $\mathbb{C}$  (Rozdz. 3.2.1) - oprócz robotów  $R$  i ich przestrzeni działania  $TO$  oraz zadań  $Z$  - składają się parametry misji  $MP$  opisane zależnością:

$$MP = (am, m, auaa, cf, mcfv\_am), \quad (3.21)$$

gdzie:

$am$  – określa metodę alokacji zadań,

$m$  – pozwala na wybór kryterium oceny ofert,

$auaa$  – decyduje o udziale w alokacji tylko robotów niezajętych  $R^{nb}$  lub wszystkich robotów aktywnych  $R^{nb} \cup R^{ex}$ ,

$cf$  – określa metodę obliczania wartości kryteriów oceny dla metod MCFV,

$mcfv\_am$  – decyduje o wyborze jednej z metod MCFV przydziału zadań.

Zbiór tych parametrów służy do sterowania systemem alokacji zadań. Pierwszy z parametrów określa metodę aukcjonowania i przydziału zadań: (i) jedną z dwóch standardowych metod (Rozdz. 3.3.5 oraz 3.3.6), polegających na aukcjonowaniu zadań ( $am = zadania$ ) lub aukcjonowaniu robotów ( $am = roboty$ ); (ii) jedną z wielokryterialnych metod aukcjonowania ( $am = mcfv$ ) (Rozdz. 3.4). Kolejny parametr  $m$  daje informację o kryterium oceny ofert stosowanym podczas aukcjonowania za pomocą dwóch pierwszych właśnie wymienionych metod. Rozważane są dwa kryteria (Rozdz. 3.3.3): czas ( $m = ET$ ) lub energia ( $m = EE$ ) potrzebne na realizację zadania. W obu wariantach dąży się do minimalizacji wartości tych parametrów.

Kolejne parametry zależności (3.21) odnoszą się do wielokryterialnych metod aukcjonowania zadań omówionych szczegółowo w rozdziale 3.4. W pracy zaproponowano

<sup>7</sup>Na potrzeby skrócenia zapisu w dalszej części pracy zamiast sformułowania "metody alokacji zadań oparte o Contract-Net Protocol" stosowane będzie sformułowanie "metody CNP"

<sup>8</sup>Zamiast sformułowania "metody stosujące (oparte o) wielokryterialną ocenę przystosowania robotów do zadań" stosowane będzie sformułowanie "metody MCFV"

dwie grupy metod, tj. metody, które podczas aukcjonowania, a następnie przydziału zadań, biorą pod uwagę tylko roboty niezajęte  $R^{nb}$  w chwili prowadzenia aukcji ( $auaa = \text{fałsz}$ ), oraz metody uwzględniające wszystkie aktywne roboty  $R^{nb} \cup R^{ex}$  ( $auaa = \text{prawda}$ ), a więc także te wykonujące zadania w chwili prowadzenia aukcji. Parametr  $cf$  odpowiedzialny jest za wybór algorytmu obliczania wartości kryteriów dla wielokryterialnych metod aukcjonowania. Pierwszy algorytm ( $cf = \text{MIN}$ ) podczas obliczania pozycji konkurencyjnej robota dla zadania lub atrakcyjności zadania dla robota jako punkt odniesienia dla każdego z kryteriów przyjmuje odpowiednio najlepszego robota lub najlepsze zadanie względem danego kryterium. Drugi z algorytmów ( $cf = \text{MIN-MAX}$ ) stosuje dwa punkty odniesienia, tj. najgorszego i najlepszego robota lub najgorsze i najlepsze zadanie. Dokładny opis tych algorytmów znajduje się w rozdziale 3.4.1. Ostatni z parametrów  $mcfv\_am$  przechowuje informację o metodzie przydziału zadań. Zaproponowano trzy metody: (i) roboty rozważane są zgodnie z ich kolejnością w kolejce robotów, każdy kolejny robot otrzymuje zadanie o największej dla siebie wartości MCFV ( $mcfv\_am = \text{Queue}$ ); (ii) roboty rozważane zgodnie z malejącą średnią swoich MCFV ( $mcfv\_am = \text{BestMeanOfMCFVs}$ ); (iii) spośród wszystkich wartości MCFV w danej aukcji, wybierana jest największa i zgodnie z nią następuje przydział zadania robotowi ( $mcfv\_am = \text{BestMCFV}$ ). Wszystkie trzy metody działają aż do przydziału wszystkich zadań lub wykorzystania wszystkich niezajętych robotów.

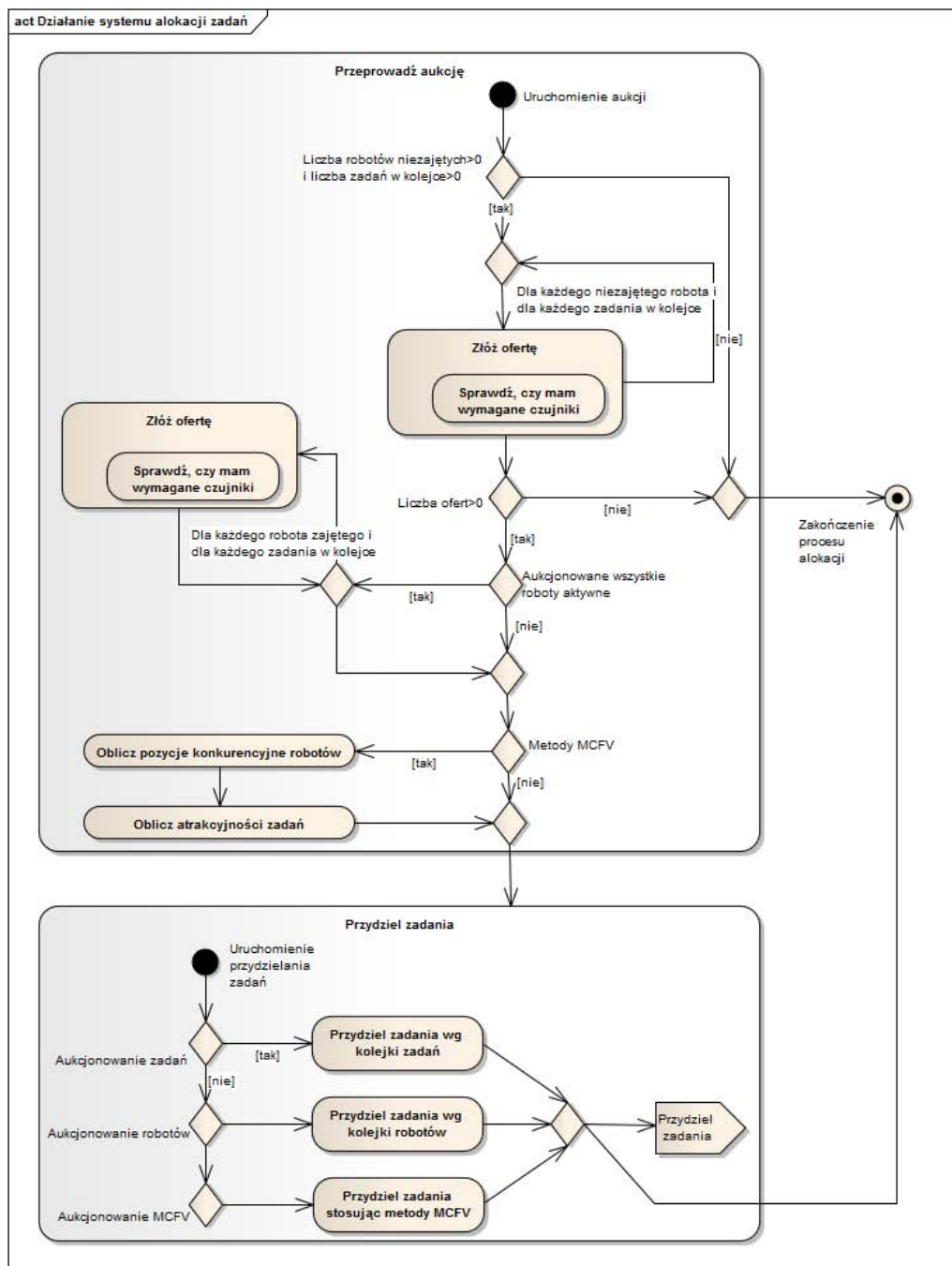
### 3.3.2. Przebieg procesu alokacji zadań

Następstwem trzech podstawowych zdarzeń systemu (aktywowanie robotów i zadań, wykonanie zadania, porzucenie zadania) jest uruchomienie procesu alokacji (Rys. 3.5), który dzieli się na:

- Proces aukcjonowania (Alg. 3.2),
- Proces przydzielania zadań (Alg. 3.3).

Proces aukcjonowania rozpoczyna się pod warunkiem, że zbiór robotów niezajętych  $R^{nb}$  oraz zbiór zadań do wykonania  $Z^a$  są niepuste (Alg. 3.2(3)). Dzieje się tak również dla tych metod MCFV, dla których do aukcji – oprócz robotów niezajętych – przystępują również roboty zajęte  $R^{ex}$ , z tego względu, że nawet w przypadku wygrania aukcji przez zajętego robota  $r$  dla wybranego zadania  $z$ , następuje *fikcyjny* przydział zadania - robot  $r$  jako potencjalnie najlepszy wykonawca zadania  $z$  blokuje je, a nadal wykonuje swoje aktualnie wykonywane zadanie (parametr  $z^{ex}$  modelu robota (3.8)). Dlatego, jeśli system nie ma do dyspozycji ani jednego niezajętego robota, nie nastąpi żaden *faktyczny* przydział zadania, a zatem przeprowadzenie aukcji nie ma sensu.

W chwili rozpoczęcia procesu aukcjonowania dla wszystkich zadań kolejki zadań do wszystkich robotów niezajętych wysyłane są zapytania o złożenie ofert (Alg. 3.4). Jeśli liczba ofert złożonych przez roboty niezajęte jest większa od 0 i poza robotami niezajętymi aukcjonowane są roboty zajęte (Alg. 3.4(13–14)), wtedy do nich (robotów zajętych) również wysyłane są zapytania ofertowe o wykonanie zadań. Może się zdarzyć, że od ro-



Rys. 3.5: Działanie systemu alokacji zadań

botów niezajętych nie wpłynie żadna oferta, wtedy – podobnie jak wcześniej – nie ma sensu dalsze prowadzenie aukcji.

W przypadku alokacji zadań stosującej metody MCFV, przed samym przydzielaniem zadań, obliczone zostają pozycje konkurencyjne robotów dla zadań (Alg. 3.2(23)), atrakcyjności zadań dla robotów (Alg. 3.2(26)) i wartości  $mcfv$  dla każdej pary  $(r, z)$ , dla której robot  $r$  złożył ofertę  $b$  dla zadania  $z$ .

Ostatnim krokiem procesu aukcjonowania jest uruchomienie (Alg. 3.2(29)) procesu przydzielania zadań. Przydział zadań (Alg. 3.3) następuje w oparciu o:

- Aukcjonowanie zadań (Rozdz. 3.3.5),

- Aukcjonowanie robotów (Rozdz. 3.3.6),
- Metody MCFV(Rozdz. 3.4).

Wynikiem każdorazowego procesu alokacji zadań jest zdarzenie systemu  $ev =$  przydział, w następstwie którego następuje aktualizacja zbiorów robotów  $R$  i zadań  $Z$  (Rys. 3.3).

---

### Algorytm 3.2 Aukcja

---

```

1: Funkcja Przeprowadź aukcję
argumenty:  $R^{nb}$  - zbiór robotów niezajętych
            $R^{ex}$  - zbiór robotów będących w trakcie wykonywania zadań
            $Z^q$  - kolejka zadań
            $MP$  - parametry misji
2:  $B \leftarrow \emptyset$ 
3: Jeżeli  $|R^{nb}| = 0$  lub  $|Z^q| = 0$ 
4:   Przerwij działanie
5: W przeciwnym razie
6:   Dla wszystkich  $z \in Z^q$ 
7:     Dla wszystkich  $r \in R^{nb}$ 
8:       Wywołaj Złóż ofertę
9:       argumenty:  $r, z, MP$ 
10:      zwracane:  $b$ 
11:      Jeżeli  $b \neq \text{null}$ 
12:         $B \leftarrow B \cup \{b\}$ 
13:   Jeżeli  $|B| > 0$ 
14:     Jeżeli  $MP.aaaa = \text{prawda}$ 
15:       Dla wszystkich  $z \in Z^q$ 
16:         Dla wszystkich  $r \in R^{ex}$ 
17:           Wywołaj Złóż ofertę
18:           argumenty:  $r, z, MP$ 
19:           zwracane:  $b$ 
20:           Jeżeli  $b \neq \text{null}$ 
21:              $B \leftarrow B \cup \{b\}$ 
22:     Jeżeli  $MP.am = \text{mcfv}$ 
23:       Wywołaj Oblicz pozycje konkurencyjne robotów dla zadań
24:       argumenty:  $B, MP$ 
25:       zwracane:  $B$ 
26:       Wywołaj Oblicz atrakcyjności zadań dla robotów
27:       argumenty:  $B, MP$ 
28:       zwracane:  $B$ 
29:       Wywołaj Przydziel zadania
30:       argumenty:  $B, MP$ 
31:   W przeciwnym razie
32:     Przerwij działanie

```

---

**Algorytm 3.3** Przydział zadań1: **Funkcja** Przydział zadaniaargumenty:  $B$  - zbiór ofert robot/zadanie $MP$  - parametry misji $R^{nb}$  - zbiór robotów niezajętych $R^{ex}$  - zbiór robotów będących w trakcie wykonywania zadań $Z^q$  - kolejka zadań2: **Jeżeli**  $MP.am = \text{zadania}$ 3: **Wywołaj** Przydział zadania wg kolejki zadań4: argumenty:  $B, Z^q, MP$ 5: **W przeciwnym razie, jeżeli**  $MP.am = \text{mcfv}$ 6: **Wywołaj** Przydział zadania stosując MCFV7: argumenty:  $B, R^{nb}, R^{ex}, MP$ 8: **W przeciwnym razie, jeżeli**  $MP.am = \text{roboty}$ 9: **Wywołaj** Przydział zadania wg kolejki robotów10: argumenty:  $B, R^{nb}, MP$ **3.3.3. Kryteria oceny ofert**

Aukcyjne mechanizmy alokacji zadań wymagają określenia kryterium, zgodnie z którym ocenie podlegają oferty składane przez roboty, a następnie dokonywany jest przydział zadań. Do najczęściej stosowanych kryteriów należą czas oraz energia, jakie są potrzebne na wykonanie zadania. Stąd w metodach alokacji zadań opisywanych w pracy zastosowano te dwa kryteria.

**Czas**

Mając dane zadanie  $z$  oraz robota  $r$  mogącego to zadanie wykonać, zgodnie z przyjętymi modelami robotów i zadań, można wyznaczyć czas potrzebny na realizację zadania  $z$  przez robota  $r$  z poniższej zależności:

$$ert = \overbrace{d \cdot r.res.vel^{max}}^{①} + \overbrace{z.\tau.t^{ex}}^{②}, \quad (3.22)$$

gdzie:

① – szacowany czas potrzebny na dojazd robota do zadania,

② – szacowany czas potrzebny na wykonanie zadania przez robota <sup>9</sup>,

<sup>9</sup>Szacowany czas potrzebny na wykonanie zadania przy przyjętych założeniach systemu nie zależy od robota go wykonującego, a więc jest to stały składnik kryterium czasowego, który w przypadku aukcjonowania stosującego to kryterium jako podstawę oceny oferty mógłby być pominięty. Zdecydowano się jednak go wprowadzić, ponieważ kryterium czasowe jest czynnikiem kryterium energetycznego, którego wartość jest zależna jednocześnie od czasu potrzebnego na wykonanie zadania i robota wykonującego to zadanie (③ w 3.23 lub ⑦ w 3.24)



$d$  – odległość pomiędzy robotem a miejscem wykonywania zadania <sup>10</sup>.

### Energia

Dla zadania  $z$  oraz robota  $r$  mogącego to zadanie wykonać, zgodnie z przyjętymi modelami robotów i zadań, energia potrzebna na realizację zadania  $z$  przez robota  $r$  wyznaczona zostaje z poniższej zależności:

$$ere = \overbrace{ert \cdot r.ec.ec^{sb}}^{③} + \overbrace{d \cdot r.res.vel^{max} \cdot r.ec.ec^{trav}}^{④} + \overbrace{z.\tau.t^{ex} \cdot r.ec.ec^{ex}}^{⑤}, \quad (3.23)$$

gdzie:

③ – szacowana energia potrzebna na działanie podstawowych układów robota,

④ – szacowana energia potrzebna na działanie układów napędowych robota,

⑤ – szacowana energia potrzebna na działanie zasobów inspekcyjnych robota,

$d$  – j.w.

lub inaczej

$$ere = \overbrace{d \cdot r.res.vel^{max} \cdot (r.ec.ec^{trav} + ert \cdot r.ec.ec^{sb})}^{⑥} + \overbrace{z.\tau.t^{ex} \cdot (r.ec.ec^{ex} + ert \cdot r.ec.ec^{sb})}^{⑦}, \quad (3.24)$$

gdzie:

⑥ – szacowana energia potrzebna na dojazd robota do miejsca wykonywania zadania,

⑦ – szacowana energia potrzebna na wykonanie zadania przez robota,

$d$  – j.w.

#### 3.3.4. Model oferty mechanizmu aukcyjnego

W chwili rozpoczęcia aukcji dla wszystkich zadań w kolejce  $Z^q$  do wszystkich potencjalnych wykonawców (robotów aktywnych  $R^{act}$  lub tylko niezajętych  $R^{nb}$ ) wysyłane są zapytania ofertowe (Alg. 3.4) ze specyfikacją zadania do wykonania. Po otrzymaniu zapytania robot sprawdza, czy posiada zasoby inspekcyjne  $r^{insp}$  wymagane przez zadanie (Alg. 3.4(4), Alg.3.5) oraz oblicza jednostkowe zużycie energii  $ec^{ex}$  wynikające z używania tych zasobów inspekcyjnych w trakcie wykonywania zadania  $z$ . Jeśli robot nie posiada wymaganych zasobów inspekcyjnych, nie składa oferty na zadanie (Alg. 3.4(9)).

Ostatnim krokiem algorytmu składania ofert jest wyznaczanie wartości kryterium oceny ofert zgodnie z zależnościami 3.22 – 3.24. Obliczenie tych wartości może nastąpić na dwa sposoby zależne od tego, czy:

<sup>10</sup>Odległość  $d$  wyznaczana jest z zastosowaniem algorytmu A\*, na podstawie znajomości ograniczonej topografii przestrzeni  $TO$ , pozycji robota oraz miejsca wykonywania zadania.

- Robot składający ofertę jest niezajęty (Alg. 3.4(16)),
- W aukcji rozpatrywane są wszystkie roboty aktywne oraz robot składający ofertę realizuje (dojeżdża lub wykonuje) w danej chwili zadanie (Alg. 3.4(12)).

---

**Algorytm 3.4** Składanie ofert
 

---

1: **Funkcja** Złóż ofertę

argumenty:  $r$  - robot mający złożyć ofertę

$z$  - zadanie, na które składana jest oferta

$MP$  - parametry misji

zwracane:  $b$  - oferta

2:  $b.r \leftarrow r$

3:  $b.z \leftarrow z$

4: **Wywołaj** Sprawdź, czy mam wymagane zasoby inspekcyjne

5: argumenty:  $r, z$

6: zwracane:  $r', ps$

7: **Jeżeli**  $ps = \text{fałsz}$

8:  $b \leftarrow \text{null}$

9: **Przerwij działanie**

10: **W przeciwnym razie**

11:  $b.r.ec.ec^{ex} \leftarrow r'.ec.ec^{ex}$

12: **Jeżeli**  $MP.auaa$  i  $r.z^{ex} \neq \text{null}$

13:  $b.d \leftarrow \overbrace{\text{obliczOdległość}((r.p.x^{act}, r.p.y^{act}), (r.z^{ex}.v.x, r.z^{ex}.v.y))}^{(1)} +$   
 $\overbrace{\text{obliczOdległość}((r.z^{ex}.v.x, r.z^{ex}.v.y), (z.v.x, z.v.y))}^{(2)}$

14:  $b.ert \leftarrow \overbrace{b.d \cdot r.res.vel^{max}}^{(3)} + \overbrace{r.z^{ex}.t.t^{left}}^{(4)} + \overbrace{z.t.t^{ex}}^{(5)}$

15:  $b.ere \leftarrow \overbrace{b.ert \cdot r.ec.ec^{sb}}^{(6)} + \overbrace{b.d \cdot r.res.vel^{max} \cdot r.ec.ec^{trav}}^{(7)} +$   
 $\overbrace{b.z.t.t^{ex} \cdot b.r.ec.ec^{ex}}^{(8)} + \overbrace{r.z^{ex}.t.t^{left} \cdot r.ec.ec^{ex}}^{(9)}$

16: **W przeciwnym razie**

17:  $b.d \leftarrow \text{obliczOdległość}((r.p.x^{act}, r.p.y^{act}), (z.v.x, z.v.y))$

18:  $b.ert \leftarrow b.d \cdot r.res.vel^{max} + z.t.t^{ex}$

19:  $b.ere \leftarrow b.ert \cdot r.ec.ec^{sb} + b.d \cdot r.res.vel^{max} \cdot r.ec.ec^{trav} + z.t.t^{ex} \cdot b.r.ec.ec^{ex}$

20: **Jeżeli**  $b.ere > r.bl$

21: **Przerwij działanie**

---

W pierwszym przypadku obliczane wartości kryteriów oceny oferty dotyczą tylko zadania, na które składana jest oferta. Natomiast w drugim przypadku, mimo że robot składa ofertę na wykonanie zadania będącego przedmiotem aktualnego zapytania, to wartość kryterium oceny oferty na realizację tego zadania jest sumą wartości kryteriów oceny ofert:

- Aktualnie realizowanego zadania (tylko pozostałej części koniecznej do jego zrealizowania),
- Zadania będącego przedmiotem aktualnego zapytania ofertowego.

Symbole umieszczone w algorytmie 3.4 oznaczają<sup>11</sup>: ① – pozostały dystans pomiędzy aktualną pozycją robota a miejscem wykonywania zadania; ② – dystans pomiędzy miejscem wykonywania aktualnego zadania a miejscem wykonywania zadania, na które składana jest oferta; ③ – czas potrzebny na dojazd do miejsc wykonywania obu zadań; ④ – pozostały czas wykonywania aktualnego zadania; ⑤ – czas wykonywania zadania będącego przedmiotem oferty; ⑥ – energię potrzebną do zasilenia podstawowych układów robota w trakcie realizacji obu zadań; ⑦ – energię potrzebną do zasilenia układów napędowych robota w trakcie dojazdu do miejsc wykonywania obu zadań; ⑧ – energię potrzebną do zasilenia zasobów inspekcyjnych w trakcie wykonywania zadania, na które składana jest oferta; ⑨ – energię potrzebną do zasilenia zasobów inspekcyjnych w trakcie wykonywania aktualnego zadania.

---

### Algorytm 3.5 Sprawdzenie wymaganych zasobów inspekcyjnych

---

1: **Funkcja** Sprawdź, czy mam wymagane zasoby inspekcyjne  
 argumenty:  $r$  - robot mający złożyć ofertę  
                $z$  - zadanie, dla którego składana jest oferta  
                $MP$  - parametry misji  
 zwracane:  $r, ps$  - parametr określający, czy robot ma wymagane zasoby inspekcyjne  
 2:  $ps \leftarrow$  prawda  
 3: **Dla wszystkich**  $s \in z.res^{insp}$   
 4:     **Jeżeli**  $s \in r.res.res^{insp}$   
 5:          $r.ec.ec^{ex} \leftarrow r.ec.ec^{ex} + s.ec^{insp}$   
 6:     **W przeciwnym razie**  
 7:          $ps \leftarrow$  fałsz  
 8:     **Przerwij działanie**

---

Jeśli robot jest w stanie podjąć się danego zadania, tj. posiada wymagane przez zadanie zasoby inspekcyjne (Alg. 3.4(10)) oraz nadmiar energii pozwalający na realizację zadania (Alg. 3.4(20)), wysyła z powrotem ofertę  $b$ , której model jest następujący:

$$\begin{aligned}
 b &= (r, z, d, ert, ere, ur, zc, rcp, za, mcfv) & (3.25) \\
 &= (r, z, d, ert, ere, ur, zc, \\
 &\quad rcp = (rcp^{r \rightarrow z}, ert^{r \rightarrow z}, ere^{r \rightarrow z}, ur^{r \rightarrow z}), \\
 &\quad za = (za^{z \rightarrow r}, ert^{z \rightarrow r}, ere^{z \rightarrow r}, ur^{z \rightarrow r}, zc^{z \rightarrow r}, zp^{z \rightarrow r}) \\
 &\quad mcfv^{r \rightarrow z}),
 \end{aligned}$$

---

<sup>11</sup>Są to wartości szacowane.

gdzie<sup>12</sup>:

- $r, z$  – odpowiednio: robot składający ofertę i zadanie, na które składana jest oferta,
- $d$  – odległość pomiędzy robotem  $r$  a zadaniem  $z$ ,
- $ert$  – szacowany czas wykonania zadania  $z$ ,
- $ere$  – szacowana energia potrzebna na wykonanie zadania  $z$ ,
- $ur$  – liczba nadmiarowych zasobów inspekcyjnych robota  $r$  względem zadania  $z$ ,
- $zc$  – zmiana koncentracji zadań dla robota  $r$  wynikająca z wykonania zadania  $z$ ,
- $rcp$  – pozycja konkurencyjna robota  $r$  dla zadania  $z$ ,
- $za$  – atrakcyjność zadania  $z$  dla robota  $r$ ,
- $mcfv^{r \rightarrow z}$  – wielokryterialna ocena przystosowania robota  $r$  do zadania  $z$ .

### 3.3.5. Kolejkowa alokacja zadań

Proces alokacji zadań według ich kolejki przebiega w ten sposób, że dla kolejnych zadań w kolejce  $Z^q$  powtarzane są następujące kroki:

1. Menedżer aukcji wysyła do wszystkich robotów niezajętych  $R^{nb}$  zapytanie ofertowe ze specyfikacją zadania,
2. Każdy robot, jeśli może podjąć się zadania, oblicza wartości kryteriów oceny oferty i odsyła ofertę menedżerowi aukcji,
3. Po zebraniu wszystkich ofert następuje ich ocena względem kryterium ich oceny,
4. Robot, który złożył najbardziej korzystną ofertę, otrzymuje kontrakt na wykonanie zadania.

Zgodnie z przedstawionym algorytmem dla każdego kolejnego zadania uruchamiana jest osobna aukcja, po której następuje przydział tego zadania *najlepiej dopasowanemu* robotowi, pod warunkiem, że co najmniej jeden robot złożył ofertę. Taki sposób postępowania zazwyczaj zmniejsza liczbę składanych ofert w stosunku do liczby ofert powstających w wyniku działania algorytmu aukcjonowania zastosowanego w pracy (Alg. 3.2(6–12)), jednak przydział zadań dla danego zbioru zadań  $Z^q$  oraz zbioru robotów  $r^{nb}$  zawsze będzie taki sam. Zdecydowano się stosować taki algorytm, ponieważ w badaniach weryfikacyjnych zamieszczone zostały porównania wyników alokacji zadań za pomocą metod CNP i metod MCFV, a metody MCFV wymagają znajomości wszystkich możliwych ofert. Podsumowując, stosowany w pracy kolejkowy algorytm alokacji zadań składa się z następujących kroków:

1. Menedżer aukcji wysyła zapytanie ofertowe do każdego robota dla każdego zadania,
2. Każdy robot dla każdego zadania, jeśli może się go podjąć, oblicza wartości kryteriów oceny ofert i odsyła ofertę menedżerowi aukcji,

<sup>12</sup>Parametry elementów  $rcp$  oraz  $za$  szczegółowo opisane są w rozdz. 3.4.2 i rozdz. 3.4.3. W indeksach górnych tych parametrów zastosowano wyrażenia  $r \rightarrow z$  lub  $z \rightarrow r$ , co przy przyjętym sposobie zapisu modeli systemu nie jest konieczne, ale w dalszej części pracy poprawi jej czytelność.

3. Menedżer aukcji kolejno dla każdego zadania  $z$  w kolejce zadań  $Z^q$  (Alg. 3.6):
  - Ze zbioru wszystkich ofert  $B$  wybiera te (podzbiór  $B_z^{temp}$  (Alg. 3.6(5))), których przedmiotem było zadanie  $z$ ,
  - Z podzbioru ofert  $B_z^{temp}$  znajduje najkorzystniejszą ofertę (jeśli jest przynajmniej jedna) zgodnie z parametrem misji  $m$  i przydziela zadanie  $z$  robotowi, który złożył tę ofertę.

---

**Algorytm 3.6** Przydział zadań wg kolejki zadań
 

---

1: **Funkcja** Przydziel zadania wg kolejki zadań  
 argumenty:  $B$  - zbiór ofert robot/zadanie  
            $MP$  - parametry misji  
            $R^{nb}$  - zbiór robotów niezajętych  
            $Z_q$  - kolejka zadań

2:  $B_z^{temp} \leftarrow \emptyset$

3: **Dla wszystkich**  $z \in Z^q$

4:      $B_z^{temp} \leftarrow \{b \in B : b.z = z\}$

5:     **Jeżeli**  $|B_z^{temp}| > 0$

6:         Sortuj( $B_z^{temp}, MP.m$ )

7:          $r \leftarrow \{r' \in R^{nb} : r' = B_z^{temp}_{0,r}\}$

8:          $r.z^{ex} \leftarrow z //$

9:         **Wywołaj** Aktualizuj zbiory robotów i zadań

10:         argumenty:  $r, z, ev \leftarrow$  przydział

11:          $B \leftarrow B \setminus \{b \in B : b.r = r\}$

12:          $B \leftarrow B \setminus \{b \in B : b.z = z\}$

13:          $B_z^{temp} \leftarrow \emptyset$

---

### 3.3.6. Alokacja zadań według kolejki robotów

Druga zaimplementowana w systemie metoda oparta o CNP działa podobnie jak kolejkowa metoda alokacji zadań, z tym że podstawą działania tej metody nie jest rozpatrywanie zadań zgodnie z ich pozycją w kolejce zadań  $Z^q$ , lecz rozpatrywanie robotów według ich kolejności w kolejce robotów niezajętych  $R^{nb}$ . Proces aukcjonowania jest identyczny jak w przypadku poprzedniej metody, natomiast proces przydziału zadań przebiega inaczej. Cały proces alokacji zadań według kolejki robotów niezajętych składa się z podanych kroków:

1. Menedżer aukcji wysyła zapytanie ofertowe do każdego robota dla każdego zadania,
2. Każdy robot dla każdego zadania, jeśli może się go podjąć, oblicza wartości kryteriów oceny oferty i odsyła ofertę menedżerowi aukcji,
3. Menedżer aukcji kolejno dla każdego robota  $r$  w kolejce robotów niezajętych  $R^{nb}$  (Alg. 3.7):

- Ze zbioru wszystkich ofert  $B$  wybiera te (podzbiór  $B_r^{temp}$  (Alg. 3.7(3))), których oferentem był robot  $r$ ,
- Z podzbioru ofert  $B_r^{temp}$  znajduje najkorzystniejszą ofertę (jeśli jest przynajmniej jedna) zgodnie z parametrem misji  $m$  i przydziela zadanie  $z$  robotowi, który złożył tę ofertę.

---

**Algorytm 3.7** Przydział zadań wg kolejki robotów
 

---

1: **Funkcja** Przydziel zadania wg kolejki robotów  
 argumenty:  $B$  - zbiór ofert robot/zadanie  
 $MP$  - parametry misji  
 $R^{nb}$  - zbiór robotów niezajętych  
 $Z_q$  - kolejka zadań

2:  $B_r^{temp} \leftarrow \emptyset$

3: **Dla wszystkich**  $r \in R^{nb}$

4:      $B_r^{temp} \leftarrow \{b \in B : b.r = r\}$

5:     **Jeżeli**  $|B_r^{temp}| > 0$

6:         Sortuj( $B_r^{temp}, MP.m$ )

7:          $z \leftarrow \{z' \in Z^q : z' = B_r^{temp}_0.z\}$

8:          $r.z^{ex} \leftarrow z //$

9:         **Wywołaj Aktualizuj zbiory robotów i zadań**

10:         argumenty:  $r, z, ev \leftarrow$  przydział

11:          $B \leftarrow B \setminus \{b \in B : b.r = r\}$

12:          $B \leftarrow B \setminus \{b \in B : b.z = z\}$

13:      $B_r^{temp} \leftarrow \emptyset$

---

## 3.4. Wielokryterialne metody alokacji zadań

W rozdziale tym omówione są metody alokacji zadań w grupie robotów stosujące wielokryterialną ocenę przystosowania robotów do zadań. Ocena ta jest wyznaczana na podstawie dwóch ocen składowych, tj. pozycji konkurencyjnej robota dla zadania oraz atrakcyjności zadania dla robota.

### 3.4.1. Algorytmy wyznaczania wartości kryteriów

Pozycja konkurencyjna robota oraz atrakcyjność zadania są miarami oceny robota lub zadania, których wartości są wyznaczane na podstawie zbiorów kryteriów. Pozycja konkurencyjna robota jest oceną robota w grupie robotów. Podobnie atrakcyjność zadania jest oceną zadania w grupie zadań. Z uwagi na to, że są to oceny wielokryterialne, należy przeprowadzić normalizację wartości kryteriów oceny. Znormalizowane wartości kryteriów oceny "stanowią ocenę jakości obiektu lub efektywności jego działania w relacji do ustalonego wzorca" (Stabryła, 2000). W pracy rozważa się dwa algorytmy normalizacji

wartości kryteriów oceny, tj. algorytm MIN (opracowany na podstawie (Stabryła, 2000)) oraz algorytm MIN-MAX.

### Algorytm MIN

Zgodnie z algorytmem MIN, w celu obliczenia znormalizowanej wartości wybranego kryterium oceny, należy wyznaczyć wartość wzorcową. Dla zadanego zbioru wartości danej zmiennej  $x$ :

$$x = \{x_i : i \in \{0\} \cup \mathbb{N}, x_i \in \mathbb{R}^+\}, \quad (3.26)$$

wartość wzorcowa dla stymulanty wynosi:

$$x_{wz} = \max_i \{x_i\}, \quad (3.27)$$

wartość wzorcowa dla destymulanty wynosi:

$$x_{wz} = \min_i \{x_i\}. \quad (3.28)$$

Na podstawie wartości wzorcowej znormalizowana wartość kryterium oceny dla zmiennej będącej stymulantą wyraża się wzorem:

$$x_j^{crit} = \frac{\max_i \{x_i\}}{x_j} = \frac{x_{wz}}{x_j}, \quad (3.29)$$

natomiast wartość kryterium oceny dla zmiennej będącej destymulantą wyraża się wzorem:

$$x_j^{crit} = \frac{x_j}{\min_i \{x_i\}} = \frac{x_j}{x_{wz}}. \quad (3.30)$$

### Algorytm MIN-MAX

W algorytmie MIN-MAX normalizację zmiennej przeprowadza się w oparciu o wartości graniczne tej zmiennej, czyli dla tego samego zbioru z zależności 3.26 wartości danej zmiennej  $x$  wartość wzorcowa oraz antywzorcowa dla stymulanty wynosi:

$$x_{wz}^s = \max_i \{x_i\}, \quad (3.31)$$

$$x_{awz}^s = \min_i \{x_i\}, \quad (3.32)$$

wartość wzorcowa oraz antywzorcowa dla destymulanty wynosi:

$$x_{wz}^d = \min_i \{x_i\}, \quad (3.33)$$

$$x_{awz}^d = \max_i \{x_i\}. \quad (3.34)$$

Na podstawie wartości wzorcowej i antywzorcowej znormalizowana wartość kryterium oceny dla zmiennej będącej stymulantą wyraża się wzorem:

$$x_j^{crit} = \frac{x_j - \min_i \{x_i\}}{\max_i \{x_i\} - \min_i \{x_i\}} = \frac{x_j - x_{awz}^s}{x_{wz}^s - x_{awz}^s}, \quad (3.35)$$

natomiast wartość kryterium oceny dla zmiennej będącej destymulantą wyraża się wzorem:

$$x_j^{crit} = \frac{\max_i \{x_i\} - x_j}{\max_i \{x_i\} - \min_i \{x_i\}} = \frac{x_{awz}^d - x_j}{x_{awz}^d - x_{wz}^d}. \quad (3.36)$$

### 3.4.2. Pozycja konkurencyjna robota dla zadania

Zgodnie z przyjętymi w rozdz. 3.2 modelami, dla każdego zadania  $z \in Z^q$  - jeśli tylko została złożona przynajmniej jedna oferta wykonania zadania  $z$  - pozycja konkurencyjna każdego robota  $r$  dla tego zadania wyraża się zależnością:

$$b.rcp.rcp^{r \rightarrow z} = \frac{b.rcp.ert^{r \rightarrow z} + b.rcp.ere^{r \rightarrow z} + b.rcp.ur^{r \rightarrow z}}{3}, \quad (3.37)$$

gdzie zmienne w tym równaniu oznaczają pozycje konkurencyjne robota  $r$  dla zadania  $z$ :

$b.rcp.rcp^{r \rightarrow z}$  – ogólną,

$b.rcp.ert^{r \rightarrow z}$  – ze względu na kryterium czasu realizacji,

$b.rcp.ere^{r \rightarrow z}$  – ze względu na kryterium energii potrzebnej na realizację,

$b.rcp.ur^{r \rightarrow z}$  – ze względu na kryterium liczby nadmiarowych zasobów inspekcyjnych.

Pozycja konkurencyjna robota  $r$  dla zadania  $z$  pozwala na ocenę przystosowania robota do zadania w odniesieniu do wszystkich robotów, które złożyły oferty na wykonanie zadania  $z$ . Wartości pozycji konkurencyjnych, zarówno ogólnej jak i jej składowych, zawierają się w przedziale  $[0; 1] = \{b.rcp.rcp^{r \rightarrow z} \in \mathbb{R} : 0 \leq b.rcp.rcp^{r \rightarrow z} \leq 1\}$ . Im większa wartość pozycji konkurencyjnej  $b.rcp.rcp^{r \rightarrow z}$ , tym bardziej robot  $r$  jest przystosowany do zadania  $z$  na tle innych robotów, które mogą podjąć się tego zadania. W sytuacji, kiedy tylko jeden robot złoży ofertę na wybrane zadanie, wartości jego wszystkich pozycji konkurencyjnych równe są 1.

Dla zadanego zbioru zadań  $Z^q$  oraz zbioru ofert  $B$ , które zostały złożone na te zadania w pojedynczej aukcji, podzbiór ofert  $B_z^{temp}$  dla zadania  $z$  wyznacza się z zależności:

$$\forall z \in Z^q (\exists b \in B b.z = z \Rightarrow B_z^{temp} = \{b' \in B : b'.z = z\}), \quad (3.38)$$

a następnie na podstawie zawartości zbioru  $B_z^{temp}$  wyznacza się wartości pozycji konkurencyjnych robotów, które złożyły oferty na wykonanie zadania  $z$ , stosując równania (3.39, 3.41, 3.43) lub równania (3.40, 3.42, 3.44), odpowiednio dla algorytmów MIN lub MIN-MAX wyznaczania wartości kryteriów.

Ocena pozycji konkurencyjnych robotów prowadzona jest dla trzech kryteriów (czas, energia, liczba nadmiarowych zasobów inspekcyjnych), dla których dąży się do minimalizacji ich wartości. W działaniu systemu skutkuje to tym, że ze zbioru robotów mogących wykonać dane zadanie najwyższą pozycję konkurencyjną mają te, które potrzebują najmniej czasu na realizację tego zadania (czyli najszybciej dojadą do miejsca jego wykonywania, bo czas wykonywania zadania jest dla każdego robota taki sam), zużyją najmniej



energii i będą miały najmniej zasobów inspekcyjnych, których nie będą mogli użyć w trakcie wykonywania tego zadania.

Procedurę wyznaczania pozycji konkurencyjnych dla wszystkich robotów biorących udział w wybranej aukcji szczegółowo przedstawia Alg. 3.8 (Str. 68).

### Kryterium czasowe (destymulanta)

$$\forall_{b \in B_z^{temp}} b.rcp.ert^{r \rightarrow z} = \frac{\min_i \{B_z^{temp}.ert\}}{b.ert} \quad (3.39)$$

$$\forall_{b \in B_z^{temp}} b.rcp.ert^{r \rightarrow z} = \frac{\max_i \{B_z^{temp}.ert\} - b.ert}{\max_i \{B_z^{temp}.ert\} - \min_i \{B_z^{temp}.ert\}} \quad (3.40)$$

### Kryterium energetyczne (destymulanta)

$$\forall_{b \in B_z^{temp}} b.rcp.ere^{r \rightarrow z} = \frac{\min_i \{B_z^{temp}.ere\}}{b.ere} \quad (3.41)$$

$$\forall_{b \in B_z^{temp}} b.rcp.ere^{r \rightarrow z} = \frac{\max_i \{B_z^{temp}.ere\} - b.ere}{\max_i \{B_z^{temp}.ere\} - \min_i \{B_z^{temp}.ere\}} \quad (3.42)$$

### Kryterium liczby nadmiarowych zasobów inspekcyjnych (destymulanta)<sup>13</sup>

$$\forall_{b \in B_z^{temp}} b.rcp.ur^{r \rightarrow z} = \frac{\min_i \{B_z^{temp}.ur\} + 1}{b.ur + 1} \quad (3.43)$$

$$\forall_{b \in B_z^{temp}} b.rcp.ur^{r \rightarrow z} = \frac{\max_i \{B_z^{temp}.ur\} - b.ur}{\max_i \{B_z^{temp}.ur\} - \min_i \{B_z^{temp}.ur\}} \quad (3.44)$$

### 3.4.3. Atrakcyjność zadania dla robota

Dla każdego robota  $r \in R^{nb}$  lub  $r \in R^{ex}$  - jeśli złożył ofertę na wykonanie zadania  $z$  - atrakcyjność zadania  $z$  dla robota  $r$  wyraża się zależnością:

$$b.za.za^{r \rightarrow z} = \frac{b.za.ert^{z \rightarrow r} + b.za.ere^{z \rightarrow r} + b.za.ur^{z \rightarrow r} + b.za.zc^{z \rightarrow r} + b.za.zp^{z \rightarrow r}}{5}, \quad (3.45)$$

gdzie zmienne w tym równaniu oznaczają atrakcyjności zadania  $z$  dla robota  $r$ :

<sup>13</sup>W równaniu 3.43 w liczniku i mianowniku dodano 1, aby zapobiec sytuacji, kiedy dla zadania, dla którego obliczane są pozycje konkurencyjne robotów, istnieje oferta robota mającego zbiór zasobów inspekcyjnych równy zbiorowi zasobów inspekcyjnych wymaganych przez zadanie. W takiej sytuacji wyrażenie  $\min_i \{B_{z(i)}^{temp}.ur\}$  przyjmuje wartość 0, w następstwie czego pozycja konkurencyjna ze względu na liczbę nadmiarowych czujników każdego robota mogącego wykonać dane zadanie też przyjmuje wartość 0, niezależnie od liczby nadmiarowych zasobów inspekcyjnych. Dodatkowy problem pojawia się dla robota, dla którego  $b.ur = 0$ , kiedy to wartość  $b.rcp.ur^{r \rightarrow z}$  jest symbolem nieoznaczonym.

$b.za.za^{z \rightarrow r}$  – ogólną,

$b.za.ert^{z \rightarrow r}$  – ze względu na kryterium czasu realizacji,

$b.za.ere^{z \rightarrow r}$  – ze względu na kryterium energii potrzebnej do realizacji,

$b.za.ur^{z \rightarrow r}$  – ze względu na kryterium liczby nadmiarowych zasobów inspekcyjnych,

$b.za.zc^{z \rightarrow r}$  – ze względu na kryterium zmiany koncentracji zadań,

$b.za.zp^{z \rightarrow r}$  – ze względu na kryterium zysku z wykonania zadania.

Atrakcyjność zadania  $z$  dla robota  $r$  pozwala na ocenę przystosowania robota do zadania względem wszystkich zadań, na które robot złożył ofertę. Wartości atrakcyjności zadań, zarówno ogólnej jak i jej składowych, zawierają się w przedziale  $[0; 1] = \{b.za.za^{z \rightarrow r} \in \mathbb{R} : 0 \leq b.za.za^{z \rightarrow r} \leq 1\}$ . Im większa wartość atrakcyjności zadania  $b.za.za^{z \rightarrow r}$ , tym bardziej robot  $r$  jest przystosowany do zadania  $z$  na tle innych zadań, które może zrealizować. W sytuacji, kiedy robot  $r$  złoży tylko ofertę na zadanie  $z$ , atrakcyjność tego zadania  $b.za.za^{z \rightarrow r}$  równa jest 1.

Dla zadanych zbiorów robotów zajętych  $R^{ex}$  i niezajętych  $R^{nb}$  oraz zbioru ofert  $B$ , które te roboty złożyły na zadania znajdujące się w kolejce zadań  $Z^q$ , podzbiór ofert  $B_r^{temp}$  wyznacza się z zależności:

$$\forall_{r \in R^{nb} \cup R^{ex}} (\exists_{b \in B} b.r = r \Rightarrow B_r^{temp} = \{b' \in B : b'.r = r\}), \quad (3.46)$$

a następnie na podstawie zawartości zbioru  $B_r^{temp}$  wyznacza się atrakcyjności zadań, na które robot  $r$  złożył oferty, stosując równania (3.47, 3.49, 3.51, 3.53, 3.55) lub równania (3.48, 3.50, 3.52, 3.54, 3.56), odpowiednio dla algorytmów MIN lub MIN-MAX wyznaczania wartości kryteriów.

### Kryterium czasowe (destymulanta)

$$\forall_{b \in B_r^{temp}} b.za.ert^{z \rightarrow r} = \frac{\min_i \{B_r^{temp}_i.ert\}}{b.ert} \quad (3.47)$$

$$\forall_{b \in B_r^{temp}} b.za.ert^{z \rightarrow r} = \frac{\max_i \{B_r^{temp}_i.ert\} - b.ert}{\max_i \{B_r^{temp}_i.ert\} - \min_i \{B_r^{temp}_i.ert\}} \quad (3.48)$$

### Kryterium energetyczne (destymulanta)

$$\forall_{b \in B_r^{temp}} b.za.ere^{z \rightarrow r} = \frac{\min_i \{B_r^{temp}_i.ere\}}{b.ere} \quad (3.49)$$

$$\forall_{b \in B_r^{temp}} b.za.ere^{z \rightarrow r} = \frac{\max_i \{B_r^{temp}_i.ere\} - b.ere}{\max_i \{B_r^{temp}_i.ere\} - \min_i \{B_r^{temp}_i.ere\}} \quad (3.50)$$

**Kryterium liczby nadmiarowych zasobów inspekcyjnych (destymulanta)**

$$\forall_{b \in B_r^{temp}} b.za.ur^{z \rightarrow r} = \frac{\min_i \{B_r^{temp}{}_i.ur\} + 1}{b.ur + 1} \quad (3.51)$$

$$\forall_{b \in B_r^{temp}} b.za.ur^{z \rightarrow r} = \frac{\max_i \{B_r^{temp}{}_i.ur\} - b.ur}{\max_i \{B_r^{temp}{}_i.ur\} - \min_i \{B_r^{temp}{}_i.ur\}} \quad (3.52)$$

**Kryterium zmiany koncentracji zadań (destymulanta)**

$$\forall_{b \in B_r^{temp}} b.za.zc^{z \rightarrow r} = \frac{\min_i \{B_r^{temp}{}_i.zc\}}{b.zc} \quad (3.53)$$

$$\forall_{b \in B_r^{temp}} b.za.zc^{z \rightarrow r} = \frac{\max_i \{B_r^{temp}{}_i.zc\} - b.zc}{\max_i \{B_r^{temp}{}_i.zc\} - \min_i \{B_r^{temp}{}_i.zc\}} \quad (3.54)$$

**Kryterium zysku z wykonania zadania (stymulanta)**

$$\forall_{b \in B_r^{temp}} b.za.zp^{z \rightarrow r} = \frac{b.zp}{\max_i \{B_r^{temp}{}_i.zp\}} \quad (3.55)$$

$$\forall_{b \in B_r^{temp}} b.za.zp^{z \rightarrow r} = \frac{b.zp - \min_i \{B_r^{temp}{}_i.zp\}}{\max_i \{B_r^{temp}{}_i.zp\} - \min_i \{B_r^{temp}{}_i.zp\}} \quad (3.56)$$

Zastosowanie pierwszych trzech kryteriów powoduje, że dla wybranego robota wśród zadań, których ten robot może się podjąć, najbardziej atrakcyjne są te, na realizację których robot potrzebuje najmniej czasu i energii, oraz te, które w największym stopniu wykorzystają zasoby inspekcyjne robota.

Wartość atrakcyjności zadania ze względu na kryterium  $b.za.zc^{z \rightarrow r}$  obliczana jest na podstawie parametru  $zc$  będącego elementem każdej oferty. Parametr ten określa zmianę koncentracji zadań dla robota  $r$  wynikającą z realizacji przez niego wybranego zadania  $z$ . Jako koncentrację zadań dla wybranego robota przyjmuje się średnią odległość pomiędzy tym robotem a wszystkimi zadaniami, które ten robot może zrealizować. Zapisując za pomocą pseudokodu, zmianę koncentracji zadań określa się jako:

$$b.zc = \frac{\text{śr. odl. pomiędzy robotem } r \text{ a zadaniami, które może wykonać, po zrealizowaniu zadania } z}{\text{śr. odl. pomiędzy robotem } r \text{ a zadaniami, które może wykonać, przed zrealizowaniem zadania } z} \quad (3.57)$$

Przy tak zdefiniowanym parametrze  $zc$ , wzrost jego wartości ma negatywny wpływ na (obniża) atrakcyjność zadania, a więc jest on destymulantą. Wartość zmiany koncentracji zadań  $zc$  pozwala stwierdzić, czy wykonanie jednego zadania zbliża robota do pozostałych zadań. Kryterium  $b.za.zc^{z \rightarrow r}$  faworyzuje zadania, które są skupione, a obniża atrakcyjność zadań występujących pojedynczo.

Dla każdego zadania w momencie wprowadzenia go do systemu określony jest czas potrzebny na jego wykonanie  $t^{ex}$ . Kryterium  $b.za.zp^{z \rightarrow r}$  określa, jaki jest względny zysk (w stosunku do zysku wynikającego z wykonania innych zadań) z wykonania danego zadania. W działaniu systemu zastosowanie takiego kryterium skutkuje przykładowo tym, że z podobnie oddalonych od robota zadań najbardziej atrakcyjne będzie to o najdłuższym czasie potrzebnym na jego wykonanie, a więc wzrośnie stosunek czasu lub energii potrzebnych na wykonanie do czasu lub energii potrzebnych na dojazd, co jest skutkiem pożądanym.

---

**Algorytm 3.8** Wyznaczanie pozycji konkurencyjnej robotów dla zadań
 

---

1: **Funkcja** Oblicz pozycje konkurencyjne robotów dla zadań

argumenty:  $B$  - zbiór ofert w aukcji

$MP$  - parametry misji

zwracane:  $B$

2:  $B_z^{temp} \leftarrow \emptyset$

3:  $IDs \leftarrow \text{ZnajdźIdentyfikatoryZadań}(B)$

4: **Dla wszystkich**  $id \in IDs$

5:  $B_z^{temp} \leftarrow \{b \in B : b.z.id = id\}$

6:  $B \leftarrow B \setminus B_z^{temp}$

7: **Sortuj**( $B_z^{temp}, ert$ )

8:  $ert^{min} \leftarrow B_z^{temp}_0.ert$

9:  $ert^{max} \leftarrow B_z^{temp}_{|B_z^{temp}|-1}.ert$

10: **Sortuj**( $B_z^{temp}, ere$ )

11:  $ere^{min} \leftarrow B_z^{temp}_0.ere$

12:  $ere^{max} \leftarrow B_z^{temp}_{|B_z^{temp}|-1}.ere$

13: **Sortuj**( $B_z^{temp}, ur$ )

14:  $ur^{min} \leftarrow B_z^{temp}_0.ur$

15:  $ur^{max} \leftarrow B_z^{temp}_{|B_z^{temp}|-1}.ur$

16: **Dla wszystkich**  $b \in B_z^{temp}$

17:     **Jeżeli**  $MP.cf = \text{MIN}$

18:          $b.rcp.ert^{r \rightarrow z} \leftarrow ert^{min} / b.ert$

19:          $b.rcp.ere^{r \rightarrow z} \leftarrow ere^{min} / b.ere$

20:          $b.rcp.ur^{r \rightarrow z} \leftarrow (ur^{min} + 1) / (b.ur + 1)$

21:     **Jeżeli**  $MP.cf = \text{MIN-MAX}$

22:          $b.rcp.ert^{r \rightarrow z} \leftarrow (ert^{max} - b.ert) / (ert^{max} - ert^{min})$

23:          $b.rcp.ere^{r \rightarrow z} \leftarrow (ere^{max} - b.ere) / (ere^{max} - ere^{min})$

24:          $b.rcp.ur^{r \rightarrow z} \leftarrow (ur^{max} - b.ur) / (ur^{max} - ur^{min})$

25:          $b.rcp.rcp^{r \rightarrow z} \leftarrow (b.rcp.ert^{r \rightarrow z} + b.rcp.ere^{r \rightarrow z} + b.rcp.ur^{r \rightarrow z}) / 3$

26:  $B \leftarrow B \cup B_z^{temp}$

27:  $B_z^{temp} \leftarrow \emptyset$

---

**Algorytm 3.9** Wyznaczanie atrakcyjności zadań dla robotów

---

1: **Funkcja** Oblicz atrakcyjności zadań dla robotów  
argumenty:  $B$  - zbiór ofert w aukcji  
 $MP$  - parametry misji  
zwracane:  $B$

2:  $B_r^{temp} \leftarrow \emptyset$

3:  $IDs \leftarrow \text{ZnajdźIdentyfikatoryRobotów}(B)$

4: **Dla wszystkich**  $id \in IDs$

5:  $B_r^{temp} \leftarrow \{b \in B : b.r.id = id\}$

6:  $B \leftarrow B \setminus B_r^{temp}$

7:  $\text{Sortuj}(B_r^{temp}, ert); \quad ert^{min} \leftarrow B_r^{temp}_0.ert; \quad ert^{max} \leftarrow B_r^{temp}_{|B_r^{temp}|-1}.ert$

8:  $\text{Sortuj}(B_r^{temp}, ere); \quad ere^{min} \leftarrow B_r^{temp}_0.ere; \quad ere^{max} \leftarrow B_r^{temp}_{|B_r^{temp}|-1}.ere$

9:  $\text{Sortuj}(B_r^{temp}, ur); \quad ur^{min} \leftarrow B_r^{temp}_0.ur; \quad ur^{max} \leftarrow B_r^{temp}_{|B_r^{temp}|-1}.ur$

10:  $\text{Sortuj}(B_r^{temp}, z.tex); \quad tex^{min} \leftarrow B_r^{temp}_0.z.tex; \quad tex^{max} \leftarrow B_r^{temp}_{|B_r^{temp}|-1}.z.tex$

11:  $zc^{min} \leftarrow 0; \quad zc^{max} \leftarrow 0$

12: **Dla wszystkich**  $b \in B_r^{temp}$

13:  $b.zc \leftarrow \text{ObliczZmianęKoncentracjiZadań}(b, B_r^{temp})$

14: **Jeżeli**  $b.zc \geq zc^{max}$

15:  $zc^{max} \leftarrow b.zc$

16: **Jeżeli**  $b.zc \leq zc^{min}$

17:  $zc^{min} \leftarrow b.zc$

18: **Dla wszystkich**  $b \in B_r^{temp}$

19: **Jeżeli**  $MP.cf = \text{MIN}$

20:  $b.za.ert^{z \rightarrow r} \leftarrow (ert^{min})/(b.ert)$

21:  $b.za.ere^{z \rightarrow r} \leftarrow (ere^{min})/(b.ere)$

22:  $b.za.ur^{z \rightarrow r} \leftarrow (ur^{min} + 1)/(b.ur + 1)$

23:  $b.za.zc^{z \rightarrow r} \leftarrow (zc^{min})/(b.zc)$

24:  $b.za.zp^{z \rightarrow r} \leftarrow (b.z.tex)/(tex^{max})$

25: **Jeżeli**  $MP.cf = \text{MIN-MAX}$

26:  $b.za.ert^{z \rightarrow r} \leftarrow (ert^{max} - b.ert)/(ert^{max} - ert^{min})$

27:  $b.za.ere^{z \rightarrow r} \leftarrow (ere^{max} - b.ere)/(ere^{max} - ere^{min})$

28:  $b.za.ur^{z \rightarrow r} \leftarrow ((ur^{max} - b.ur)/(ur^{max} - ur^{min}))$

29:  $b.za.zc^{z \rightarrow r} \leftarrow (zc^{max} - b.zc)/(zc^{max} - zc^{min})$

30:  $b.za.zp^{z \rightarrow r} \leftarrow (b.z.tex - tex^{min})/(tex^{max} - tex^{min})$

31:  $b.za.za^{z \rightarrow r} \leftarrow (b.za.ert^{z \rightarrow r} + b.za.ere^{z \rightarrow r} + b.za.ur^{z \rightarrow r} + b.za.zc^{z \rightarrow r} + b.za.zp^{z \rightarrow r})/5$

32:  $B \leftarrow B \cup B_r^{temp}$

33:  $B_r^{temp} \leftarrow \emptyset$

---

### 3.4.4. Wielokryterialna ocena przystosowania robota do zadania

Dla każdej pary  $(r, z)$ , dla której na zadanie  $z$  złożona została oferta  $b$  przez robota  $r$ , wartość wielokryterialnej oceny przystosowania robota  $r$  do zadania  $z$  wyznaczona zostaje na podstawie znajomości:

- Pozycji konkurencyjnej robota dla zadania  $rcp^{r \rightarrow z}$  (3.37),
- Atrakcyjności zadania dla robota  $za^{z \rightarrow r}$  (3.45),

za pomocą równania:

$$mcfv^{r \rightarrow z} = \sqrt{(rcp^{r \rightarrow z})^2 + (za^{z \rightarrow r})^2}. \quad (3.58)$$

Można łatwo zauważyć, że wartość  $mcfv^{r \rightarrow z}$  jest odległością euklidesową pomiędzy:

- punktem  $(0,0)$  będącym początkiem prostokątnego układu współrzędnych, którego osie reprezentują pozycje konkurencyjne robotów i atrakcyjności zadania,
- a punktem należącym do tego układu o współrzędnych  $(rcp^{r \rightarrow z}, za^{z \rightarrow r})$ .

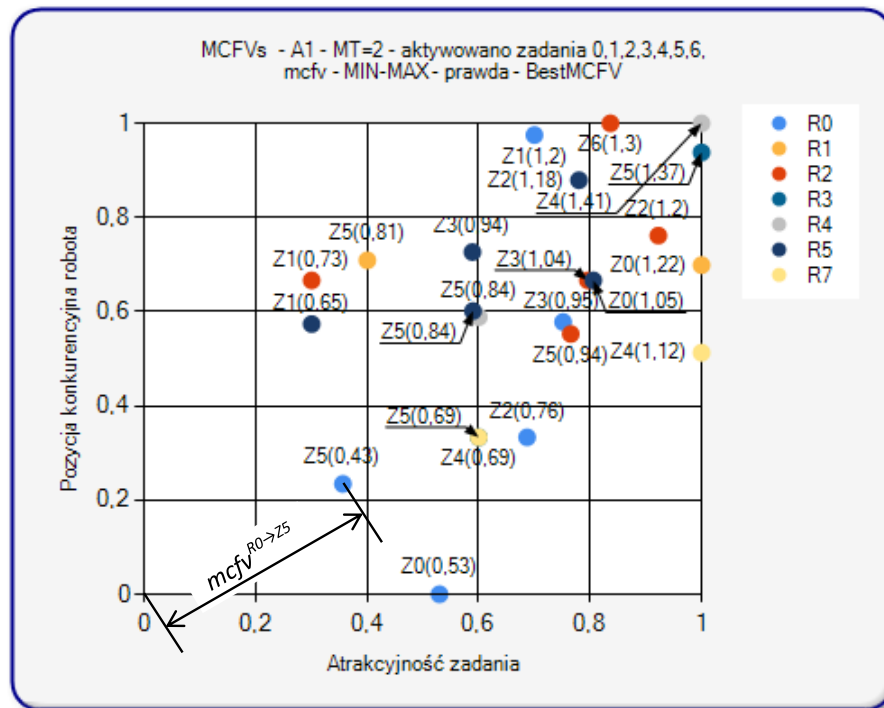
Wielokryterialna ocena przystosowania robota do zadania  $mcfv^{r \rightarrow z}$  jest miarą, na podstawie której w metodach MCFV następuje przydział zadań robotom. W każdej z tych metod, mimo że działają one według różnych algorytmów, dąży się do przydziału zadań o jak największej wartości  $mcfv^{r \rightarrow z}$ .

Poniżej zamieszczono przykładowy wykres (Rys. 3.6)<sup>14</sup> prezentujący wartości  $mcfv^{r \rightarrow z}$  dla wszystkich ofert złożonych w jednej aukcji przez osiem robotów dla sześciu zadań. Na osi odciętych odkładane są wartości atrakcyjności zadań, a na osi rzędnych wartości pozycji konkurencyjnych robotów. W tab. 3.2 zamieszczono dane, na podstawie których wygenerowany został wykres z rys. 3.6. W komórkach tabeli zawarte są informacje o wartości  $mcfv^{r \rightarrow z}$  (przed nawiasem) wynikającej z atrakcyjności  $za^{z \rightarrow r}$  zadania  $z$  dla robota  $r$  (pierwszy element wewnątrz nawiasu) oraz pozycji konkurencyjnej  $rcp^{r \rightarrow z}$  robota  $r$  dla zadania  $z$  (drugi element wewnątrz nawiasu). Pusta komórka w tabeli oznacza, że robot  $r$  nie złożył oferty na wykonanie zadania  $z$ . Na rys. 3.6 oraz w tab. 3.2 wyróżniono przykładową wartość  $mcfv^{r \rightarrow z}$  dla oferty złożonej przez robota R0 na wykonanie zadania Z5.

### 3.4.5. Metody przydzielania zadań stosujące wielokryterialną ocenę przystosowania robotów do zadań MCFV

Po każdej zakończonej aukcji, gdy są obliczone wartości  $mcfv^{r \rightarrow z}$  dla wszystkich ofert  $b \in B$ , następuje przydział zadań według jednej z trzech metod przydzielania zadań

<sup>14</sup>Wykres został wygenerowany za pomocą oprogramowania opracowanego w ramach badań prezentowanych w pracy. Oprogramowanie to implementuje opisywany w pracy system sterowania grupą robotów inspekcyjnych i służy jego weryfikacji.



Rys. 3.6: Przykładowy wykres wielokryterialnych ocen przystosowania robotów do zadań

Tab. 3.2: Przykładowe wartości atrakcyjności zadań, pozycji konkurencyjnych robotów oraz wielokryterialnych ocen przystosowania robotów do zadań - dane do wykresu z rys. 3.6

R\Z	Z0 (4)	Z1 (5)	Z2 (6)	Z3	Z5 (2)	Z6 (3)	Z4 (1)
R0 (5)	0,53 (0,53;0)	1,2 (0,7;0,98)	0,76 (0,69;0,33)	0,95 (0,75;0,58)	0,43 (0,36;0,23)		
R1 (4)	1,22 (1;0,7)				0,81 (0,4;0,71)		
R2 (3)		0,73 (0,3;0,67)	1,2 (0,92;0,76)	1,04 (0,8;0,67)	0,94 (0,77;0,55)	1,3 (0,84;1)	
R3 (2)					1,37 (1;0,94)		0,69 (0,6;0,33)
R4 (1)					0,84 (0,6;0,59)		1,41 (1;1)
R5 (6)	1,05 (0,81;0,67)	0,65 (0,3;0,57)	1,18 (0,78;0,88)	0,94 (0,59;0,73)	0,84 (0,59;0,6)		
R7 (0)					0,69 (0,6;0,33)		1,12 (1;0,51)

stosujących MCFV. Należy nadmienić, że dla każdej z tych metod wartości atrakcyjności zadań i pozycji konkurencyjnych robotów będących składowymi  $mcfv^{r \rightarrow z}$  mogą zostać obliczone za pomocą algorytmu MIN lub MIN-MAX (Rozdz. 3.4.1). Do omówienia tych metod posłużą algorytm 3.10 (str. 73).

### Metoda kolejkowa oraz metoda największej średniej wartości $mcfv^{r \rightarrow z}$

Obie opisywane w tym podrozdziale metody stosują algorytmy przydzielania zadań oparte o kolejkę robotów, z tym że w przypadku:

- metody kolejkowej ( $MP.mcfv\_am = Queue$ ) w trakcie procesu przydzielania zadań roboty rozpatrywane są zgodnie z ich kolejnością w kolejce robotów niezajętych  $R^{nb}$ ,
- metody największej średniej wartości  $mcfv^{r \rightarrow z}$  ( $MP.mcfv\_am = BestMeanOfMCFVs$ ) roboty biorące udział w aukcji szeregowane są malejąco według średniej wartości  $mcfv^{r \rightarrow z}$  dla wszystkich ofert

złożonych przez każdego robota.

Metoda kolejkowa zakłada, że do alokacji zadań mogą przystąpić tylko roboty niezajęte  $R^{nb}$ , ponieważ przydzielanie zadań przebiega zgodnie z kolejką robotów, a trudno jest określić jak taka kolejka miałyby być zbudowana z dwóch niezależnych zbiorów robotów zajętych i niezajętych. Dla metody największej średniej wartości  $mcfv^{r \rightarrow z}$  początkowa kolejność robotów nie ma znaczenia, zostają one uszeregowane dopiero po obliczeniu dla każdego z tych robotów średniej wartości  $mcfv^{r \rightarrow z}$  wszystkich złożonych przez niego ofert. Stąd przy zastosowaniu tej metody do alokacji zadań przystępują wszystkie roboty niezajęte, a mogą przystąpić również roboty zajęte - w ten sposób powstają dwa warianty tej metody. Rozważania zawarte w tym i poprzednim akapicie uwzględnione są w liniach 3-8 algorytmu 3.10.

Gdy jest określony uporządkowany zbiór robotów  $R^{al}$ , którym mogą być przydzielone zadania, rozpoczyna się proces przydzielania zadań zgodnie z kolejnością tych robotów. Proces ten składa się z następujących kroków:

1. Dla każdego robota  $r \in R^{al}$  wyznaczony zostaje zbiór ofert  $B_r^{temp}$  złożonych przez tego robota (Alg. 3.10 (10)),
2. Jeśli robot  $r$  złożył przynajmniej jedną ofertę ( $B_r^{temp} > 0$ ), określona zostaje oferta  $b$  o największej wartości  $mcfv^{r \rightarrow z}$ , a tym samym zadanie  $z$  będące przedmiotem tej oferty (Alg. 3.10 (12-13)),
3. Następuje przydział zadania, który jest przydziałem (Alg. 3.10 (14)):
  - *Fikcyjnym* – jeśli ofertę złożył robot zajęty. Robot kontynuuje aktualnie wykonywane przez siebie zadanie, a przy tym *blokuje* zadanie  $z$ ,
  - *Faktycznym* – jeśli ofertę złożył robot niezajęty. Jeśli ma miejsce *faktyczny* przydział zadania, następuje aktualizacja zbiorów robotów  $R$  i zadań  $Z$  (Alg. 3.1),
4. Ze zbioru wszystkich ofert  $B$  usunięte zostają – niezależnie od tego, czy przydział był *faktyczny*, czy *fikcyjny* – te oferty, które złożone zostały przez robota  $r$ , lub których przedmiotem było zadanie  $z$  (Alg. 3.10 (18-19)).

Powyższy proces przydzielania zadań powtarzany jest dla każdego  $r \in R^{al}$ , ale może się zdarzyć, że dla wybranego robota  $r$  nie nastąpi przydział zadania, mimo że złożył on oferty na jakieś zadania. Sytuacja taka może być skutkiem wcześniejszego przydziału wszystkich zadań, na które robot złożył oferty.

### **Metoda największej wartości $mcfv^{r \rightarrow z}$**

Proces przydzielania zadań stosujący metodę największej wartości  $mcfv^{r \rightarrow z}$  ( $MP.mcfv_{am} = \text{BestMCFV}$ ) przebiega odmiennie w stosunku do procesu przedstawionego w poprzednim rozdziale – kolejne przydziały zadań nie wynikają z kolejności robotów w kolejce, lecz z kolejności ofert w uporządkowanym zbiorze ofert. Oferty w zbiorze ofert posortowane są względem malejącej wartości  $mcfv^{r \rightarrow z}$ . Gdy dany jest



**Algorytm 3.10** Przydział zadań wg metod stosujących MCFV

---

1: **Funkcja** Przydziel zadania stosując metody MCFV  
argumenty:  $B$  - zbiór ofert robot/zadanie  
 $MP$  - parametry misji  
 $R^{nb}$  - zbiór robotów niezajętych  
 $R^{ex}$  - zbiór robotów będących w trakcie realizacji zadania

2:  $B_r^{temp} \leftarrow \emptyset$   
3:  $R^{al} \leftarrow R^{nb}$   
4: **Jeżeli**  $MP.aauua = \text{prawda}$  **i**  $MP.mcfv\_m \neq \text{Queue}$   
5:      $R^{al} \leftarrow R^{ex}$   
6: **Jeżeli**  $MP.mcfv\_am = \text{Queue}$  **lub**  $MP.mcfv\_am = \text{BestMeanOfMCFVs}$   
7:     **Jeżeli**  $MP.mcfv\_am = \text{BestMeanOfMCFVs}$   
8:          $R^{al} \leftarrow \text{ObliczŚredniąMCFVsNastępnieSortujMalejąco}(R^{al}, B)$   
9:     **Dla wszystkich**  $r \in R^{al}$   
10:          $B_r^{temp} \leftarrow \{b \in B : b.r = r\}$   
11:         **Jeżeli**  $|B_r^{temp}| > 0$   
12:              $\text{SortujMalejąco}(B_r^{temp}, \text{mcfv})$   
13:              $z \leftarrow B_r^{temp}_0.z$   
14:             **Jeżeli**  $r \in R^{nb}$   
15:                  $r.z^{ex} \leftarrow z$   
16:             **Wywołaj Aktualizuj zbiory robotów i zadań**  
17:                 argumenty:  $r, z, ev \leftarrow$  przydział  
18:                  $B \leftarrow B \setminus \{b \in B : b.r = r\}$   
19:                  $B \leftarrow B \setminus \{b \in B : b.z = z\}$   
20:              $B_r^{temp} \leftarrow \emptyset$   
21: **W przeciwnym razie, jeżeli**  $MP.mcfv\_am = \text{BestMCFV}$   
22:      $\text{SortujMalejąco}(B, \text{mcfv})$   
23:     **Dopóki**  $|B| > 0$   
24:          $z \leftarrow B_0.z$   
25:          $r \leftarrow B_0.r$   
26:         **Jeżeli**  $r \in R^{nb}$   
27:              $r.z^{ex} \leftarrow z$   
28:         **Wywołaj Aktualizuj zbiory robotów i zadań**  
29:             argumenty:  $r, z, ev \leftarrow$  przydział  
30:          $B \leftarrow B \setminus \{b \in B : b.r = r\}$   
31:          $B \leftarrow B \setminus \{b \in B : b.z = z\}$

---

uporządkowany zbiór ofert, proces przydzielania zadań przebiega według następujących kroków:

1. Z aktualnego zbioru ofert  $B$  rozpatrywana jest pierwsza oferta  $b$  (Alg. 3.10 (24-25)),
2. Zgodnie z parametrami  $b.r$  i  $b.z$  tej oferty następuje przydział zadania  $z$  robotowi  $r$ , który może być przydziałem *fikcyjnym* lub *faktycznym* (Alg. 3.10 (26)). Jeśli ma miejsce *faktyczny* przydział zadania, następuje aktualizacja zbiorów robotów  $R$  i zadań  $Z$ ,
3. Ze zbioru wszystkich ofert  $B$  usunięte zostają – niezależnie od tego, czy przydział był *faktyczny*, czy *fikcyjny* – te oferty, które złożone zostały przez robota  $r$ , lub których przedmiotem było zadanie  $z$  (Alg. 3.10 (30-31)).

Proces kończy się, kiedy w zbiorze ofert nie ma żadnej oferty ( $|B| = 0$ )

## 3.5. Behavioralne sterowanie ruchem robotów

Do sterowania ruchem robotów postanowiono zastosować podejście behawioralne, którego zachowania modelowane są za pomocą pól potencjalnych (Arkin, 1987). Zidentyfikowano kilka podstawowych zachowań, tj. *jedź-do-punktu*, *omijaj-innego-robota*, *omijaj-przeszkodę*, *losowe*, które pozwolą na przemieszczanie się robotów do miejsc wykonywanych zadań.

W założeniach systemu przyjęto, że każdy robot wyposażony jest w system lokalizacji pozwalający na określenie jego pozycji oraz orientacji bezwzględnej. W pracy nie są rozpatrywane algorytmy lokalizacji, a zakłada się, że jest ona wyznaczona poprawnie. Dodatkowo zakłada się, że każdy robot wyposażony jest w system detekcji otoczenia pozwalający na wykrywanie przeszkód w pobliżu robota. Przyjęto, że czujnikiem służącym do detekcji otoczenia będzie powszechnie stosowany w robotyce mobilnej laserowy czujnik skanujący, którego parametrami istotnymi dla działania sterownika behawioralnego są: kąt skanowania, rozdzielczość oraz zakres.

### 3.5.1. Koordynator ruchu robota

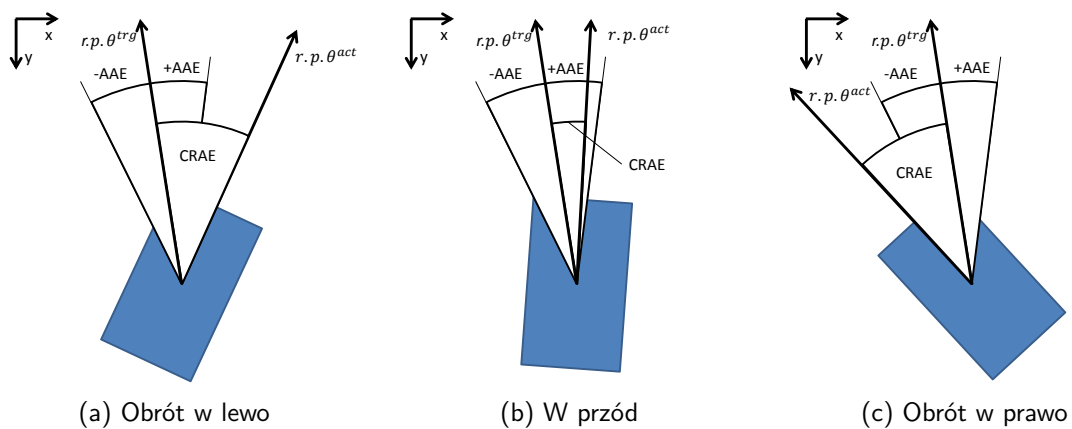
Działanie koordynatora ruchu robota przedstawione jest za pomocą diagramu maszyny stanowej z rys. 3.8. *Dojazd do zadania* (Rys. 3.4) jest stanem, w stosunku do którego stanem podrzędnym jest *Dojazd do wierzchołka v* (Rys. 3.8).

Cały czas, kiedy robot znajduje się w stanie *Dojazd do wierzchołka v*, obliczane są za pomocą koordynatora zachowań (Rozdz. 3.5.2) wartości docelowej orientacji  $r.p.\theta^{trg}$  oraz docelowej prędkości  $r.vel.vel^{trg}$  robota. Na podstawie docelowej  $r.p.\theta^{trg}$  oraz aktualnej  $r.p.\theta^{act}$  orientacji robota obliczane jest odchylenie jego orientacji *CRAE* (ang. *Current*

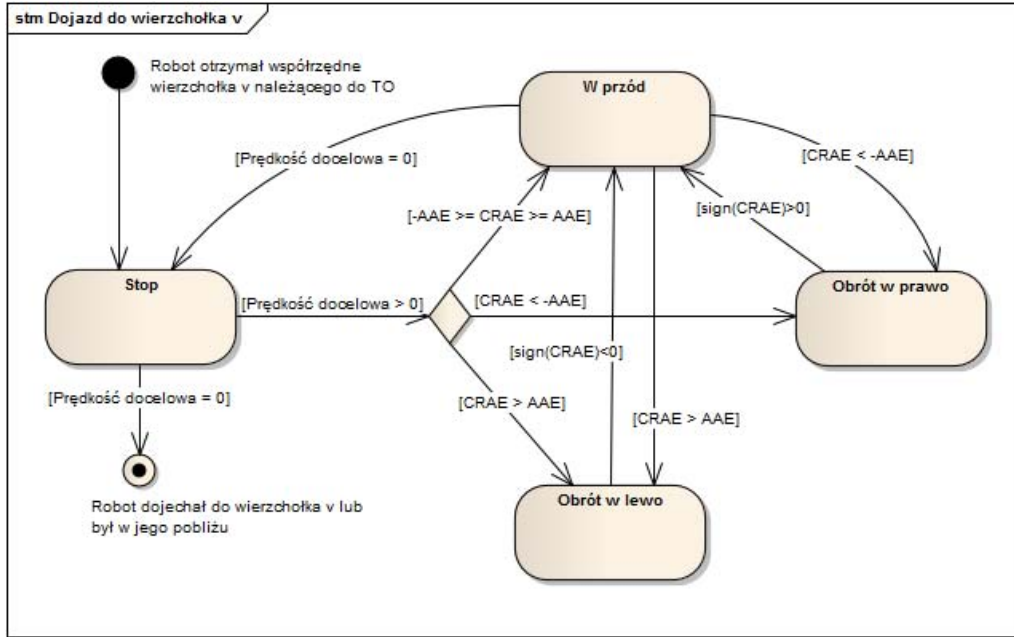
Robot Angle Error) za pomocą zależności:

$$CRAE = \begin{cases} r.p.\theta^{act} - r.p.\theta^{trg} & \text{jeśli } r.p.\theta^{act} \geq r.p.\theta^{trg} \text{ i } r.p.\theta^{act} - r.p.\theta^{trg} \leq \pi \\ 2\pi - r.p.\theta^{act} + r.p.\theta^{trg} & \text{jeśli } r.p.\theta^{act} \geq r.p.\theta^{trg} \text{ i } r.p.\theta^{act} - r.p.\theta^{trg} > \pi \\ r.p.\theta^{act} - r.p.\theta^{trg} & \text{jeśli } r.p.\theta^{act} < r.p.\theta^{trg} \text{ i } r.p.\theta^{trg} - r.p.\theta^{act} \leq \pi \\ 2\pi - r.p.\theta^{trg} + r.p.\theta^{act} & \text{jeśli } r.p.\theta^{act} < r.p.\theta^{trg} \text{ i } r.p.\theta^{trg} - r.p.\theta^{act} > \pi \end{cases} \quad (3.59)$$

Początkowo (Rys. 3.8) robot znajduje się w stanie *Stop*. Jeśli obliczona za pomocą koordynatora zachowań prędkość docelowa robota  $r.vel.vel^{trg}$  jest równa zero, oznacza to, że robot znajduje się w pobliżu miejsca reprezentowanego przez wierzchołek  $v$ . Wtedy robot oczekuje na wskazanie kolejnego wierzchołka należącego do kolejki wierzchołków  $r.V^r$ , której ostatni element wskazuje miejsce wykonywania aktualnie przydzielonego zadania. Jeśli takiego wierzchołka nie ma (robot znajduje się w miejscu wykonywania zadania), przechodzi on w stan *Wykonywanie zadania* (Rys. 3.4). Większa od zera prędkość docelowa robota wskazuje, że robot jest oddalony od miejsca reprezentowanego przez aktualny wierzchołek  $v$ . W tej sytuacji robot dojeżdża do tego miejsca i może pozostawać w jednym ze stanów *W przód*, *Obrót w prawo*, *Obrót w lewo*. Zmiana stanu robota następuje poprzez porównywanie aktualnego odchylenia orientacji robota  $CRAE$  z dopuszczalnym odchyleniem orientacji  $AAE$  (ang. *Acceptable Angle Error*) (Rys. 3.8, Rys. 3.7). W momencie dotarcia do miejsca reprezentowanego przez wierzchołek  $v$ , prędkość docelowa robota  $r.vel.vel^{trg}$  przyjmuje wartość równą zero i robot przechodzi w stan *Stop*. Uznaje się, że robot dotarł do miejsca reprezentowanego przez wierzchołek  $v$ , jeśli odległość pomiędzy robotem a tym miejscem jest mniejsza niż założony parametr  $GZR$  (ang. *Goal Zone Radius*).



Rys. 3.7: Stany robota zależne od jego odchylenia orientacji  $CRAE$



Rys. 3.8: Diagram stanów robota w trakcie dojazdu do wybranego wierzchołka  $v \in TO$

### 3.5.2. Koordynator zachowań

Parametrami mającymi wpływ na ruch robota są m. in. docelowa orientacja  $r.p.\theta^{trg}$  oraz prędkość  $r.vel.vel^{trg}$  robota. W pracy postanowiono zastosować zbiór zachowań opartych o pola potencjalne, których wektory odpowiedzi są wejściami koordynatora zachowań. Na wyjściu tego koordynatora otrzymywany jest znormalizowany wektor sumy wektorów odpowiedzi wszystkich aktywnych zachowań pomnożonych przez skojarzone z tymi zachowaniami współczynniki. Wektor odpowiedzi koordynatora zachowań, jak i odpowiedzi każdego zachowania opisany jest za pomocą pary  $(\theta, mag)$ , gdzie  $\theta$  wskazuje kierunek, a  $mag$  długość wektora odpowiedzi. Kierunek wektora odpowiedzi koordynatora zachowań określa docelową orientację  $r.p.\theta^{trg}$  robota, a jego długość pomnożona przez maksymalną prędkość robota  $r.res.v^{max}$  daje docelową prędkość  $r.vel.vel^{trg}$  robota.

Dla omówionych poniżej zachowań odpowiedź koordynatora zachowań można zapisać w postaci:

$$\rho = (r.p.\theta^{trg}, r.vel.vel^{trg}) = C \left( \begin{array}{l} g_{r \rightarrow v}^{move-to} * r_{sp_{r \rightarrow v}}^{move-to} \\ g_{r \rightarrow r_1}^{avoid-robot} * r_{sp_{r \rightarrow r_1}}^{avoid-robot} \\ g_{r \rightarrow r_2}^{avoid-robot} * r_{sp_{r \rightarrow r_2}}^{avoid-robot} \\ \dots \\ g_{r \rightarrow r_m}^{avoid-robot} * r_{sp_{r \rightarrow r_m}}^{avoid-robot} \\ g_{r \rightarrow o_1}^{avoid-obstacle} * r_{sp_{r \rightarrow o_1}}^{avoid-obstacle} \\ g_{r \rightarrow o_2}^{avoid-obstacle} * r_{sp_{r \rightarrow o_2}}^{avoid-obstacle} \\ \dots \\ g_{r \rightarrow o_n}^{avoid-obstacle} * r_{sp_{r \rightarrow o_n}}^{avoid-obstacle} \\ g_r^{random} * r_{sp_r}^{random} \end{array} \right)$$

gdzie  $m$  i  $n$  oznaczają liczby aktywnych zachowań *avoid-robot* oraz *avoid-obstacle*.

### 3.5.3. Zachowania

#### *jedź-do-punktu*

Podstawowym zachowaniem każdego robota jest zachowanie *jedź-do-punktu*, pozwalające na przemieszczanie się robota pomiędzy punktami ograniczonej topografii przestrzeni  $TO$ . Dla danego robota  $r$  i wybranego wierzchołka  $v$  parametry wektora odpowiedzi  $rsp_{r \rightarrow v}^{move-to}$  zachowania *jedź-do-punktu* wyznaczone są z poniższych zależności:

$$\theta_{r \rightarrow v}^{move-to} = \text{w kierunku punktu aktualnego} \quad (3.60)$$

wierzchołka docelowego  $v = r.V_0^r$ ,

$$mag_{r \rightarrow v}^{move-to} = 1. \quad (3.61)$$

#### *omijaj-innego-robota*

Zachowanie *omijaj-innego-robota* opracowane zostało na podstawie zachowania *avoid-static-obstacle* (Arkin, 1998). Każdy aktywny robot biorący udział w misji posiada tyle aktywnych zachowań *omijaj-innego-robota*, ile jest robotów w misji, pomniejszone o 1 (robot nie posiada zachowania pozwalającego na omijanie samego siebie), a więc pod uwagę brane są również roboty nieaktywne  $R^{nact}$ , dla których pamiętane są ich ostatnie pozycje, kiedy były jeszcze aktywne. Dla danego robota  $r \in R^{act}$  dla każdego omijanego robota  $r_i \in R \setminus \{r\}$  sposób wyznaczania  $\theta_{r \rightarrow r_i}^{avoid-robot}$  odpowiedzi  $rsp_{r \rightarrow r_i}^{avoid-robot}$  zachowania *omijaj-innego-robota* zilustrowany jest na rys. 3.9, natomiast wartość  $mag_{r \rightarrow r_i}^{avoid-robot}$  obliczana jest za pomocą zależności (3.62).

Dla sytuacji, kiedy robot  $r_i$  znajduje się z przodu robota  $r$ , tj. w 1. i 2. ćwiartce lokalnego układu współrzędnych robota  $r$  (Rys. 3.9a, Rys. 3.9b), długość wektora odpowiedzi  $mag_{r \rightarrow r_i}^{avoid-robot}$  obliczana jest za pomocą zależności (3.62). Natomiast dla sytuacji, kiedy robot  $r_i$  znajduje się z tyłu robota, tj. w 3. i 4. ćwiartce lokalnego układu współrzędnych robota  $r$  (Rys. 3.9b, Rys. 3.9d), długość wektora odpowiedzi  $mag_{r \rightarrow r_i}^{avoid-robot} = 0$ , a więc zachowanie *omijaj-innego-robota* można uznać za nieaktywne.

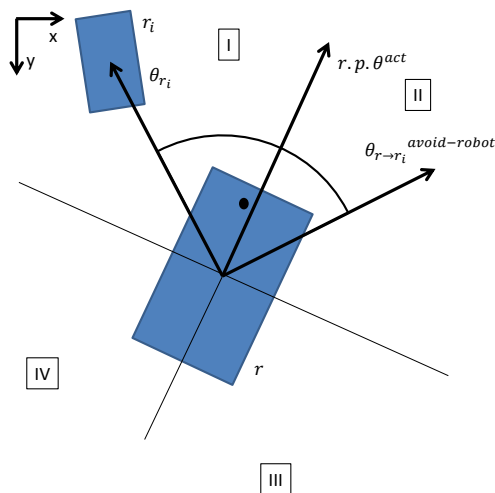
$$mag_{r \rightarrow r_i}^{avoid-robot} = \begin{cases} 0 & , \text{jeśli } d_{r \rightarrow r_i} > S_{r \rightarrow r'} \\ \frac{S_{r \rightarrow r'} - d_{r \rightarrow r_i}}{S_{r \rightarrow r'} - R_{r \rightarrow r'}} & , \text{jeśli } R_{r \rightarrow r'} < d_{r \rightarrow r_i} \leq S_{r \rightarrow r'} \\ \infty & , \text{jeśli } d_{r \rightarrow r_i} \leq R_{r \rightarrow r'} \end{cases} \quad (3.62)$$

gdzie:

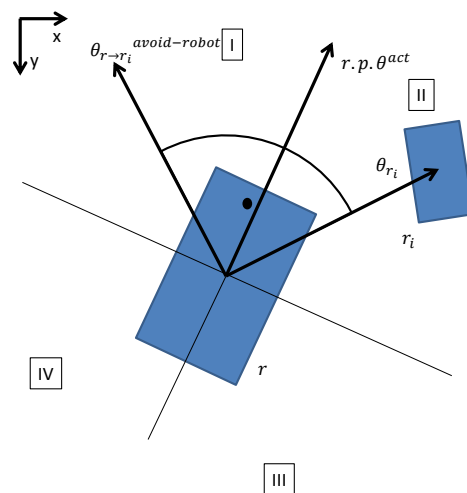
$d_{r \rightarrow r_i}$  – odległość pomiędzy robotem  $r$  a robotem aktualnie omijanym  $r_i$ ,

$S_{r \rightarrow r'}$  – maksymalna rozważana odległość pomiędzy robotem  $r$  a robotem omijanym  $r'$ ,

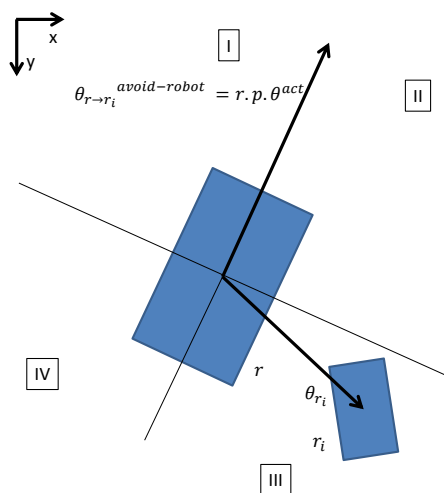
$R_{r \rightarrow r'}$  – minimalna rozważana odległość pomiędzy robotem  $r$  a robotem omijanym  $r'$ .



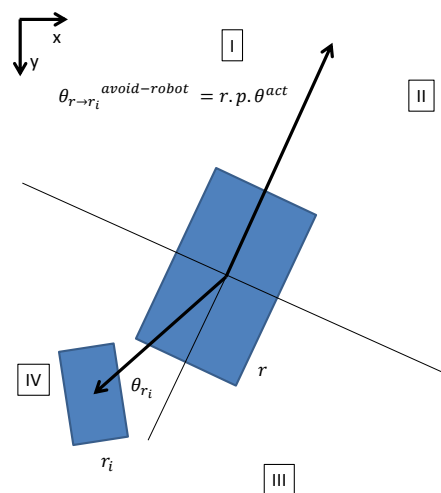
(a) Robot omijany  $r_i$  znajduje się w 1. ćwiartce lokalnego układu współrzędnych robota omijającego  $r$



(b) Robot omijany  $r_i$  znajduje się w 2. ćwiartce lokalnego układu współrzędnych robota omijającego  $r$



(c) Robot omijany  $r_i$  znajduje się w 3. ćwiartce lokalnego układu współrzędnych robota omijającego  $r$



(d) Robot omijany  $r_i$  znajduje się w 4. ćwiartce lokalnego układu współrzędnych robota omijającego  $r$

Rys. 3.9: Ilustracja zachowania *omijaj-innego-robota*

### **omijaj-przeszkodę**

Zachowanie *omijaj-przeszkodę* działa w oparciu o dane pochodzące z laserowego czujnika skanującego. Zakłada się, że robot posiada tyle aktywnych zachowań *omijaj-przeszkodę*, ile przeszkód można zidentyfikować na podstawie danych z czujnika skanującego. Przeszkody identyfikowane są w ten sposób, że dla aktualnego zbioru wyników pomiarów pochodzących z czujnika wyodrębniane są ciągi pomiarów, dla których wartość każdego kolejnego pomiaru jest mniejsza od zakresu czujnika. Odległość  $d_{r \rightarrow o_i}$  do zidentyfikowanej przeszkody określona jest przez pomiar o najmniejszej wartości z danego ciągu

pomiarów, a kierunek  $\theta_{r \rightarrow o_i}$  do przeszkody określany przez środkowy element tego ciągu. Dla tak wyznaczonych  $d_{r \rightarrow o_i}$  oraz  $\theta_{r \rightarrow o_i}$  parametry wektora odpowiedzi  $rsp_{r \rightarrow o_i}^{avoid-obstacle}$  zachowania *omijaj-przeszkodę* dla przeszkody  $o_i$  obliczane są za pomocą poniższych zależności:

$$\theta_{r \rightarrow o_i}^{avoid-obstacle} = \theta_{o_i}, \quad (3.63)$$

$$mag_{r \rightarrow o_i}^{avoid-obstacle} = \begin{cases} 0 & , \text{jeśli } d_{r \rightarrow o_i} > S_{r \rightarrow o} \\ \frac{S_{r \rightarrow o} - d_{r \rightarrow o_i}}{S_{r \rightarrow o} - R_{r \rightarrow o}} & , \text{jeśli } R_{r \rightarrow o} < d_{r \rightarrow o_i} \leq S_{r \rightarrow o} \\ \infty & , \text{jeśli } d_{r \rightarrow o_i} \leq R_{r \rightarrow o} \end{cases} \quad (3.64)$$

gdzie:

$d_{r \rightarrow o_i}$  – odległość pomiędzy robotem  $r$  a przeszkodą  $o_i$ ,

$S_{r \rightarrow o}$  – maksymalna rozważana odległość pomiędzy robotem  $r$  a przeszkodą  $o_i$ ,

$R_{r \rightarrow o}$  – minimalna rozważana odległość pomiędzy robotem  $r$  a przeszkodą  $o_i$ .

### **losowe**

Wadą behawioralnych systemów sterowania robotów mobilnych, których zachowania oparte są o pola potencjalne, jest możliwość *utknięcia* algorytmów sterowania robotem w minimum lokalnym. Dlatego zdecydowano się uwzględnić w systemie zachowanie *losowe* (Koren i Borenstein, 1991), którego wektor odpowiedzi przyjmuje losowy kierunek. Parametry opisujące wektor odpowiedzi  $rsp_r^{random}$  zachowania *losowe* oblicza się z poniższych zależności:

$$\theta_r^{random} = rand(0, 2\pi), \quad (3.65)$$

$$mag_r^{random} = \begin{cases} 0 & , \text{jeśli istnieje postęp w realizacji zadania} \\ 1 & , \text{w przeciwnym przypadku} \end{cases} \quad (3.66)$$

Działanie zachowania *random* polega na tym, że z określonym odstępem czasowym  $t^{rpt}$  sprawdza ono, czy robot będący w trakcie dojazdu przemieszcza się. Jeśli nie, parametr  $mag_r^{random}$  przyjmuje przez określony czas  $t^{random}$  (krótszy od odstępu sprawdzania) wartość 1. Operacja ta jest powtarzana kilkakrotnie. Jeśli robot nie zacznie się z powrotem przemieszczać, porzuca realizowane zadanie i staje się robotem nieaktywnym.





## Rozdział 4

# Badania weryfikacyjne

Implementacja oraz testowanie opracowanego systemu sterowania odbywały się w odpowiednio przygotowanym trójwymiarowym środowisku symulacyjnym. Prowadzenie badań w środowisku symulacyjnym (wirtualnym) umożliwiło weryfikację proponowanego systemu.

### 4.1. Sposób prezentacji wyników badań

Celem ułatwienia interpretacji wyników przeprowadzonych badań zaproponowano schematy oznaczeń zamieszczanych na wykresach aktualnych konfiguracji robotów i zadań. Schematy oraz przykładowe oznaczenia zebrane zostały w tabeli 4.1. Zadania oznaczane są kwadratami, a roboty kołami, których kolor wypełnienia zależy od typu zadania lub robota (Tab. 4.6, Tab. 4.4). Oznaczenia te umieszczone są odpowiednio w miejscu aktualnego przebywania robota oraz w miejscu, gdzie jest wykonywane zadanie.

Zadania znajdujące się w kolejce zadań oznaczane są cienko obramowanym kwadratem. Przykładowe oznaczenia takich zadań informują, że: ① – zadanie o identyfikatorze<sup>1</sup> 1 zostało wprowadzone do systemu w chwili  $t = 2$  i znajduje się w kolejce zadań (Q od *queue*); ② – w wyniku kolejkowej alokacji zadań nastąpił przydział znajdującego się w kolejce zadań  $Z^q$  zadania 5. robotowi 6. Cyfra 1 (parametr *kol* w schematach oznaczeń) na początku oznaczenia wskazuje, że był to pierwszy przydział, który w tym przypadku wynika z pozycji zadania 5. w kolejce zadań; ③ – zadanie 6. było na 4. pozycji w kolejce zadań i złożone zostały oferty na jego wykonanie, lecz wszystkim robotom, które złożyły te oferty, zostały przydzielone zadania znajdujące się na wyższych pozycjach w kolejce (stąd znak –).

Przykład oznaczenia ④ zadania realizowanego (robot dojeżdża do miejsca wykonywania tego zadania lub je wykonuje) informuje, że zadanie 1. jest realizowane (E od *executed*) przez robota 6., natomiast przykład ⑤ wskazuje, że zadanie 1. zostało wykonane (F od *finished*) przez robota 6. w chwili  $t = 39$ . Zadania realizowane i wykonane








---

<sup>1</sup>W dalszej części opisu zamiast sformułowań typu "zadanie (robot) o identyfikatorze 1, zadanie (robot) o identyfikatorze 2", itd., będą stosowane krótsze sformułowania typu "zadanie (robot) 1, zadanie (robot) 2", itd.

oznaczane są kwadratem z grubym obramowaniem, przy czym dla zadań wykonanych kwadrat ten jest obrócony o kąt  $45^\circ$ .

Roboty na wykresach oznaczane są wypełnionymi okręgami (za wyjątkiem robotów nieaktywnych oznaczanych krzyżykiem). Okręgi z cienkim obramowaniem oznaczają roboty niezajęte  $R^{nb}$ , zaś z grubym obramowaniem roboty zajęte  $R^{ex}$  (dojeżdżające do miejsca wykonywania zadania lub je wykonujące). Przykładowe oznaczenia robotów informują, że: ⑥ – robot 1. jest niezajęty (znajduje się w kolejce robotów niezajętych  $R^{nb}$  i oczekuje na aukcję); ⑦ – robot 1. złożył w trakcie aukcji oferty na wykonanie zadań, ale po tej samej aukcji wszystkie te zadania zostały przydzielone wcześniej (cyfra 0 na początku oznaczenia) innym robotom; ⑧ – robotowi 5. jako pierwszemu przydzielono zadanie 6. Jeśli alokacja zadań przebiegała według kolejki robotów, to cyfra 1 na początku oznaczenia wskazuje również na pozycję robota w kolejce robotów niezajętych  $R^{nb}$ . Pogrubiona czcionka oznaczenia informuje, że nastąpił faktyczny przydział zadania, zaś czcionka niepogrubiona wskazuje na fikcyjny przydział zadania (Rozdz. 3.4.5); ⑨ – robot 6. złożył oferty na wykonanie zadania(-ń) i był czwarty w kolejce do przydziału, ale wcześniej wszystkie zadania, na które złożył oferty, zostały przydzielone innym robotom. Zawartość informacyjna oznaczeń ⑩ – ⑮ jest podobna do zawartości wcześniejszych oznaczeń, z tym że oznaczenia te dotyczą robotów wykonujących (E od *executing*) jakieś zadania lub dojeżdżających (MT od *moving to*) do miejsca wykonywania tych zadań. Ostatni przykład oznaczenia ⑮ informuje, że robot 5. wyczerpał zapas energii w chwili  $t = 17$ .

Tab. 4.1: Schematy oznaczeń zamieszczonych na wykresach aktualnych konfiguracji robotów i zadań

Zadanie/robot	Symbol	Schemat oznaczenia	Przykłady oznaczeń
Zadanie w kolejce		$[kol]Zz.id^z(z.\tau.t^{in})-Q[->Rr.id^r]$	① Z1(2)-Q ② <b>1Z5(10)-Q-&gt;R6</b> , ③ -4Z6(10)-Q
Zadanie realizowane		$Zz.id^z(z.\tau.t^{in})-Er.id^r$	④ Z1(2)-E6
Zadanie wykonane		$Zz.id^z(z.\tau.t^{in})-Fr.id^r(z.t^f)$	⑤ Z1(2)-F6(39)
Robot niezajęty		$[kol]Rr.id^r-Q[->Zz.id^z]$	⑥ R1-Q, ⑦ 0R1-Q ⑧ <b>1R5-Q-&gt;Z6</b> , ⑨ -4R6-Q
Robot wykonujący zadanie		$[kol]Rr.id^r-Ez.id^z[->Zz.id^z]$	⑩ R1-E6, ⑪ 0R1-E6 ⑫ 1R5-E5->Z6
Robot dojeżdżający do zadania		$[kol]Rr.id^r-MTz.id^z[->Zz.id^z]$	⑬ R1-MT6, ⑭ 0R1-MT6 ⑮ 1R5-MT5->Z6
Robot nieaktywny		$Rr.id^r-EX(r.\tau.t^d)$	⑯ R5-EX(17)

## 4.2. Reprezentatywność wyników badań

Na podstawie analizy opisywanych w rozdz. 3 metod alokacji zadań w grupie robotów zidentyfikowano czternaście wariantów eksperymentów (patrz rozdz. 4.5.2). W celu zapewnienia rzetelnej oceny stosowanych i proponowanych metod alokacji zadań konieczne jest przeprowadzenie eksperymentów dla różnych środowisk działania robotów, a dla każdego takiego środowiska dla różnych konfiguracji robotów i zadań. Wyniki eksperymentów prowadzonych w warunkach rzeczywistych zapewne obciążone byłyby znacznym rozrzutem wynikającym z dużej losowości zjawisk występujących w środowisku działania robotów inspekcyjnych, a także niepełnej wiedzy o otoczeniu robotów. W przypadku eksperymentów symulacyjnych również wyniki obciążone są rozrzutem, ale znacznie mniejszym, który może wynikać m.in. z błędów numerycznych generowanych przez stosowane oprogramowanie, lub ze sposobu określania dyskretnych chwil czasu, w których wykonywane są obliczenia kolejnych kroków symulacji.

W związku z powyższym postanowiono przeprowadzić dla kilku konfiguracji po dziesięć eksperymentów, dla których dla wybranych parametrów będących miarami oceny działania systemu obliczono ich średnie wartości, odchylenia standardowe oraz błędy względne tych odchyleń w stosunku do średnich. Przykładowe wyniki zamieszczone zostały w tabelach 4.2 oraz 4.3.

Tab. 4.2: Wyniki dziesięciu eksperymentów dla przykładowej konfiguracji – wariant najlepszy: mcfv – MIN – true – BestMCFV

Nr testu	MT	TravT	AwaitT	TravD	TravE
1	102	255	693	192,2	673,9
2	103	259	694	192,5	695,2
3	104	263	700	192,2	692,4
4	105	265	707	192,0	696,3
5	106	260	717	192,2	683,4
6	106	260	726	192,4	683,4
7	107	263	756	191,5	687,4
8	104	261	699	191,8	684,8
9	103	258	695	191,8	679,4
10	102	257	692	191,7	677,6
Średnia	104,2	260,1	707,9	192,0	685,4
Odch. stand.	1,7	2,9	19,2	0,3	7,1
Błąd wzgl. [%]	1,6	1,1	2,7	0,2	1,0

Przeprowadzone testy powtarzalności upoważniają do stwierdzenia, że względne błędy obliczenia wartości kryteriów oceny metod są małe. W dalszych badaniach postanowiono zaniechać statystycznej oceny wyników symulacji i do oceny porównawczej poszczególnych metod wykorzystać wyniki uzyskane podczas co najwyżej dwukrotnego wykonania eksperymentu.

Tab. 4.3: Wyniki dziesięciu eksperymentów dla przykładowej konfiguracji – wariant najgorszy: zadania(ET)

Nr testu	MT	TravT	AwaitT	TravD	TravE
1	136	478	1025	383,8	1264,1
2	137	491	1040	387,1	1280,3
3	138	492	1036	386,5	1292,7
4	137	482	1026	386,3	1276,0
5	141	450	1063	395,9	1262,3
6	140	489	1034	388,6	1291,4
7	141	500	1041	396,4	1329,9
8	139	489	1037	393,6	1296,3
9	149	491	1041	387,3	1291,1
10	137	480	1030	389,3	1279,3
Średnia	139,5	484,2	1037,3	389,5	1286,3
Odch. std.	3,6	13,0	10,2	4,1	18,3
Błąd wzgl. [%]	2,6	2,7	1,0	1,1	1,4

### 4.3. Miary oceny działania systemu

Podczas budowy systemów sterowania robotów dąży się do tego, aby w wyniku działania tych systemów minimalizować zużycie energii, przebyty dystans, czas, które są potrzebne na realizację zadania (zadań), a maksymalizować np. liczbę zrealizowanych zadań. Dla opracowanego systemu wprowadzono kilka miar<sup>2</sup> będących podstawą jego oceny, tj.:

1. Ze względu na czas:

- **Całkowity czas realizacji misji  $MT$**  – czas od momentu rozpoczęcia misji aż do zakończenia wykonywania wszystkich zadań,
- **Sumaryczny (indywidualny) czas potrzebny na dojazd robotów do miejsca realizacji zadań  $TravT$ ,**
- Sumaryczny (indywidualny) czas oczekiwania zadań na rozpoczęcie ich wykonywania  $AwaitT$  – czas jaki upływa od chwili wprowadzenia zadania (zadań) do systemu aż do rozpoczęcia jego (ich) wykonywania,
- **Sumaryczny (indywidualny) czas *blokowania* zasobów inspekcyjnych robotów  $UnnResT$**  – czas jaki upłynął od momentu przydzielenia zadań robotom aż do chwili zakończenia wykonywania (ew. porzucenia) tych zadań, pomnożony przez liczbę nadmiarowych zasobów inspekcyjnych robotów w stosunku do wykonywanych zadań,
- Sumaryczny (indywidualny) czas bezczynności robotów  $NonBusyT$  – czas oczekiwania robotów na przydzielenie zadań,

2. Ze względu na energię:

<sup>2</sup>Pogrubioną czcionką wyróżniono najistotniejsze miary, za pomocą których przeprowadzono w rozdz. 4.6 dyskusję otrzymanych wyników badań.

- **Sumaryczną (indywidualną) energię zużytą na dojazd robotów do miejsca wykonywania zadań  $TravE$ ,**
- Sumaryczną (indywidualną) energię zużytą podczas bezczynności robotów  $NonBusyE$ ,
- Sumaryczną (indywidualną) energię zużytą w trakcie wykonywania zadań  $ExecE$ .

3. Inne:

- **Sumaryczny (indywidualny) przebyty dystans przez roboty  $TravD$ .**

## 4.4. Obiekt badań symulacyjnych

Badania weryfikacyjne postanowiono przeprowadzić w formie eksperymentu numerycznego za pomocą oprogramowania opracowanego przez autora pracy, w środowisku symulacyjnym opracowanym w ramach pracy magisterskiej (Wiglenda, 2010). W eksperymencie wykorzystano wirtualne roboty, których własności odpowiadają realnie istniejącym robotom inspekcyjnym. Trzy spośród robotów wirtualnych mają własności odpowiadające własnościom rzeczywistych robotów, opracowanych w Katedrze Podstaw Konstrukcji Maszyn Politechniki Śląskiej w ramach wykonywanych projektów badawczych i rozwojowych.

### 4.4.1. Roboty inspekcyjne

Implementacja oraz weryfikacja systemu przeprowadzone zostały z zastosowaniem czterech robotów inspekcyjnych, tj.:

- Inspectora (oznaczenie na wykresach ●) – czterokołowego robota mogącego patrolować zadany obszar, prowadzić detekcję pożarów, szkodliwych gazów, poszukiwać rannej osoby,
- Pathfinder (oznaczenie na wykresach ●) – czterokołowego robota przeznaczonego do patrolowania i poszukiwania rannych osób,
- Explorera (oznaczenie na wykresach ●) – robota gaśnicowego wykonującego te same zadania co Inspector, ale znacznie szybszego, mogącego dodatkowo przenosić niebezpieczne ładunki,
- Cleanera (oznaczenie na wykresach ○) – czterokołowego robota służącego patrolowaniu oraz usuwaniu szkodliwych substancji z podłoża.

Roboty Inspector<sup>3</sup>, Pathfinder<sup>4</sup> oraz Explorer, będące członkami grup robotów, powstały w wyniku współpracy kierowanego przez Prof. dra hab. Wojciecha Moczulskiego

<sup>3</sup>Robot Inspector powstał w ramach projektu nr PW-004/ITE/02/2006 – „Zespół specjalistycznych robotów mobilnych do inspekcji grupowej obiektów technicznych”

<sup>4</sup>Roboty Pathfinder oraz Explorer powstały w ramach projektu NB-115/RMT6/2010/PS/II.4.1/6/2010 – „Wielozadaniowe mobilne roboty wykorzystujące zaawansowane technologie”

zespołu pracowników Katedry Podstaw Konstrukcji Maszyn Politechniki Śląskiej oraz zespołu pracowników Instytutu Technologii Eksploatacji – Państwowego Instytutu Badawczego. Dane potrzebne do weryfikacji systemu zebrane są w tabeli 4.4, zaś szczegółowe informacje dotyczące przede wszystkim opracowanych rozwiązań konstrukcyjnych oraz systemów sterowania robota Inspector znaleźć można w (Moczulski *i in.*, 2008; Moczulski *i in.*, 2009; Januszka, 2008; Januszka *i in.*, 2008; Panfil i Moczulski, 2010a; Panfil i Moczulski, 2010b), natomiast robotów Pathfinder i Explorer w (Moczulski *i in.*, 2010; Moczulski *i in.*, 2011).



Rys. 4.1: Roboty inspekcyjne

Dla każdego z wymienionych robotów przygotowane zostały modele wirtualne oraz odpowiednie serwisy MRDS pozwalające na ich sterowanie w środowisku symulacyjnym MRDS. Ostatni robot inspekcyjny o nazwie Cleaner jest to robot oferowany przez firmę Coroware. Do przeprowadzenia eksperymentów zastosowano jego model wirtualny i model manipulatora (Rys. 4.1c) wraz z oprogramowaniem, które zostały dołączone do (Johns i Taylor, 2008).

Tab. 4.4: Dane charakterystyczne robotów inspekcyjnych

Robot	Parametry robota							$res^{insp}$
	Oznaczenie	Wymiary [m] (dł. × szer. × wys.)	Masa [kg]	$vel^{max}$ [m/s]	$ec^{sb}$ [A]	$ec^{trav}$ [A]	$bl$ [Ah]	
Inspector	I	0,30 × 0,28 × 0,20	5	0,7	2,2	3,5	5	W, T, P, G1, G2, G3
Pathfinder	P	0,35 × 0,30 × 0,30	5	1,0	1,2	5,5	7	W, T, P
Explorer	E	1,07 × 0,75 × 0,23	100	1,5	3,4	20	15	W, T, M, G1, G2, G3
Cleaner	C	0,30 × 0,32 × 0,25	5	0,5	2,0	2,0	3	W, C

### Zasoby inspekcyjne robotów

Każdy z robotów wyposażony jest w zbiór zasobów inspekcyjnych  $res^{insp}$  (3.7), którego zawartość jest podstawowym kryterium decydującym o złożeniu oferty przez robota na wykonanie zadania. W tabeli 4.5 zestawiono przykładowe zasoby inspekcyjne, których wymagały zadania, oraz w które wyposażone zostały roboty realizujące te zadania podczas misji będących przedmiotem eksperymentów weryfikacyjnych opisanych w pracy. Dodatkowo tabela 4.5 zawiera informacje o oznaczeniach zasobów inspekcyjnych stosowanych w systemie oraz jednostkowym zużyciu energii każdego z zasobów.

Tab. 4.5: Zasoby inspekcyjne

Zasób inspekcyjny	Oznaczenie $res^{insp}.dev^{insp}$	Zużycie energii $res^{insp}.ec^{insp}$ [A]
Kamera	W - <i>webcam</i>	1,0
Czujnik obecności	P - <i>presence</i>	0,2
Czujnik temperatury	T - <i>temperature</i>	0,2
Urządzenie czyszczące	C - <i>cleaner</i>	1,5
Czujniki gazów	G1, G2, G3 - <i>gas</i>	0,4
Manipulator	M - <i>manipulator</i>	5,0

#### 4.4.2. Parametry behawioralnego podsystemu sterowania ruchem robotów

Na potrzeby sterowania ruchem grupy robotów będącej obiektem badań symulacyjnych dobrane zostały następujące wartości parametrów:

##### 1. Dla koordynatora ruchu robotów

$$AAE = 15[^\circ], \quad GZR = 1.5[m],$$

##### 2. Dla zachowań

- *jedź-do-punktu*

$$g_{r \rightarrow v}^{move-to} = 0.8,$$

- *omijaj-innego-robota*

$$g_{r \rightarrow r'}^{avoid-robot} = 2.0, \quad R_{r \rightarrow r'} = 1.5[m], \quad S_{r \rightarrow r'} = 3.0[m],$$

- omijaj-przeszkodę

$$g_{r \rightarrow o}^{avoid-obstacle} = 2.0, \quad R_{r \rightarrow o} = 1.5[m], \quad S_{r \rightarrow o} = 3.0[m],$$

- losowe

$$g_r^{random} = 0.5, \quad t^{rpt} = 5[s], \quad t^{random} = 1[s].$$

### 4.4.3. Przykładowe zadania inspekcyjne

Misje prowadzone podczas eksperymentów symulacyjnych składały się z następujących zadań:

- Poszukiwanie osoby rannej (oznaczenie na wykresach ■),
- Detekcja zagrożenia pożarowego (oznaczenie na wykresach ■),
- Detekcja szkodliwych gazów (oznaczenie na wykresach ■),
- Usuwanie szkodliwej substancji (oznaczenie na wykresach ),
- Patrolowanie zadanego obszaru (oznaczenie na wykresach ■),
- Poszukiwanie ładunków niebezpiecznych (oznaczenie na wykresach ■).

Zgodnie z przyjętymi założeniami nie są rozważane problemy związane z wykonywaniem zadań, a w modelu systemu zadanie  $z$  uznaje się za zrealizowane, jeśli robot je wykonujący przebywał w miejscu jego wykonywania przez czas równy  $z.T.t^{ex}$ . W tabeli 4.6 dla każdego z ww. zadań podano jedynie oznaczenie oraz zasoby inspekcyjne wymagane do realizacji tego zadania. Czas potrzebny na wykonanie zadania  $z.T.t^{ex}$  oraz inne parametry zawarte w jego modelu są określone indywidualnie dla każdego zadania w momencie wprowadzania go do systemu.

Tab. 4.6: Przykładowe zadania inspekcyjne

Zadanie inspekcyjne	Oznaczenie	Wymagane zasoby insp. $res^{insp}.dev^{insp}$	Robot mogący podjąć się zadania
Poszukiwanie ranego	FH - <i>Find Human</i>	W, P	I, P
Detekcja pożaru	DF - <i>Detect Fire</i>	T, G1	I, E
Detekcja gazów	DG - <i>Detect Gases</i>	G1, G2, G3	I, E
Usuwanie szkodliwej substancji	HC - <i>Hazardous CleanUp</i>	C	C
Patrolowanie	WA - <i>Watch Area</i>	W	I, P, E, C
Zbieranie niebezpiecznych ładunków	TB - <i>Take Bomb</i>	W, M	E

## 4.5. Plan badań weryfikacyjnych

Weryfikacja systemu zrealizowana została za pomocą eksperymentów numerycznych, dla których przygotowano dwa środowiska testowe, a dla każdego z tych środowisk zdefiniowano kilka konfiguracji misji.



### 4.5.1. Środowiska testowe oraz konfiguracje eksperymentów

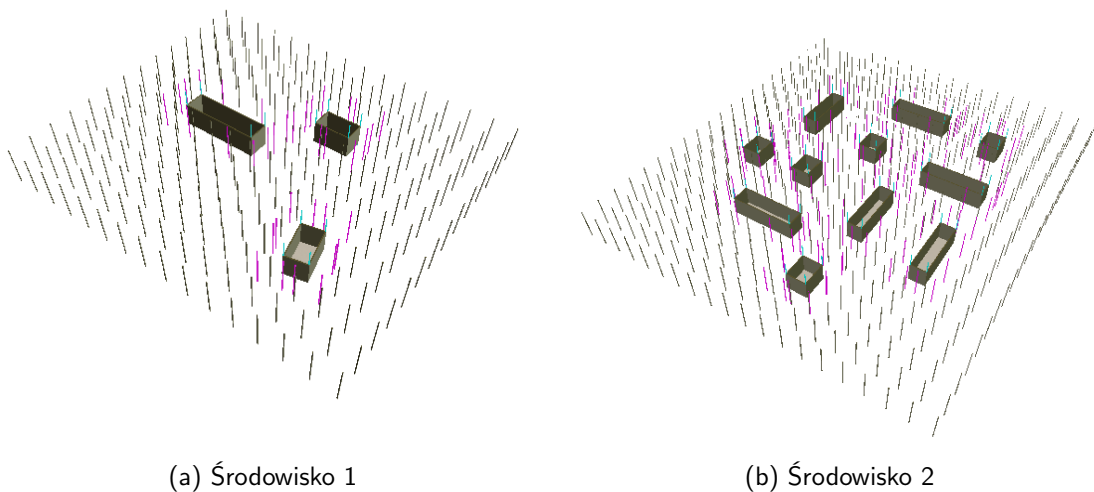
W celu przeprowadzenia weryfikacji systemu opracowano następujące środowiska testowe<sup>5</sup> oraz zdefiniowano dla nich konfiguracje robotów i zadań:

1. Środowisko 1 (Rys. 4.2a)

- Konfiguracja 1 (wygenerowana przez autora)
- Konfiguracja 2 (wygenerowana przez autora)
- Konfiguracja 3 (wygenerowana losowo na podstawie konfiguracji 2)
- Konfiguracja 4 (wygenerowana losowo na podstawie konfiguracji 2)

2. Środowisko 2 (Rys. 4.2b)

- Konfiguracja 5 (wygenerowana przez autora)
- Konfiguracja 6 (wygenerowana losowo na podstawie konfiguracji 5)



Rys. 4.2: Środowiska testowe. Widoczne są przeszkody (budynki) oraz siatki punktów reprezentujących wierzchołki grafu ograniczonej topografii przestrzeni

Na potrzeby badań weryfikacyjnych wprowadzono dwa różne (pod względem liczby przeszkód oraz ich rozmieszczenia) środowiska testowe, aby wykazać niezależność tak zdefiniowanego systemu w stosunku do środowiska, w którym działają roboty wyposażone w ten system. W przypadku, kiedy w modelu systemu nie byłaby uwzględniana wiedza o otoczeniu robotów (ograniczona topografia przestrzeni), zapewne istniałby duży wpływ środowiska na działanie systemu.

<sup>5</sup>Środowiska testowe widoczne na rys. 4.2 wygenerowane zostały za pomocą opracowanego w ramach pracy magisterskiej (Wiglenda, 2010)

### 4.5.2. Warianty eksperymentów

W rozdziale 3 rozważanych jest kilka metod (oraz ich wariantów) alokacji zadań w grupie robotów, dlatego w celu usystematyzowania przeprowadzonych badań wprowadzono symboliczne oznaczenie wariantu testu, które wyjaśnia wszystkie czynniki określone dla danego testu:

$$\overbrace{MP.am}^{①} \overbrace{[(MP.m)]}^{②} - \overbrace{[MP.cj]}^{③} - \overbrace{[MP.aaaa]}^{④} - \overbrace{[MP.mcfv\_am]}^{⑤} \quad (4.1)$$

gdzie parametry określają (podano dla przypomnienia):

- ① – metodę alokacji zadań. Parametr wymagany, przyjmuje wartości zadania, roboty, mcfv,
- ② – kryterium oceny ofert stosowane podczas aukcjonowania za pomocą metod CNP. Parametr wymagany dla tych metod, przyjmujący wartości ET (szacowany czas), EE (szacowana energia),
- ③ – algorytm wyznaczania atrakcyjności zadań i pozycji konkurencyjnych robotów. Parametr wymagany dla metod MCFV, przyjmuje wartości MIN lub MIN-MAX,
- ④ – zbiory robotów biorące udział w alokacji zadań. Parametr wymagany dla metod MCFV, przyjmuje wartości fałsz (roboty niezajęte) lub prawda (roboty zajęte i niezajęte),
- ⑤ – metodę MCFV alokacji zadań. Parametr wymagany dla metod MCFV, przyjmuje wartości Queue, BestMeanOfMCFVs lub BestMCFV.

Wyróżniono czternaście wariantów testów, którym odpowiadają oznaczenia wg 4.1:

- zadania(ET),
- zadania(EE),
- roboty(ET),
- roboty(EE),
- mcfv – MIN – fałsz – Queue,
- mcfv – MIN – fałsz – BestMeanOfMCFVs,
- mcfv – MIN – fałsz – BestMCFV,
- mcfv – MIN – prawda – BestMeanOfMCFVs,
- mcfv – MIN – prawda – BestMCFV,
- mcfv – MIN-MAX – fałsz – Queue,
- mcfv – MIN-MAX – fałsz – BestMeanOfMCFVs,
- mcfv – MIN-MAX – fałsz – BestMCFV,
- mcfv – MIN-MAX – prawda – BestMeanOfMCFVs,
- mcfv – MIN-MAX – prawda – BestMCFV.

## 4.6. Eksperymenty weryfikacyjne

W niniejszym rozdziale zamieszczone zostały wyniki eksperymentów przeprowadzonych dla sześciu konfiguracji w ramach dwóch przykładowych środowisk symulacyjnych (Rozdz. 4.5.1). Eksperymenty przeprowadzone dla Konfiguracji 1. służyły wykazaniu słuszności pierwszej tezy sformułowanej w pracy. Dla konfiguracji składającej się z jednego robota oraz trzech zadań przeprowadzono testy polegające na aukcjonowaniu zadań lub robotów. Z kolei testy przeprowadzone dla Konfiguracji 2-6 miały na celu wykazanie słuszności drugiej tezy sformułowanej w pracy. W ramach tych eksperymentów dokonano analizy wyników otrzymywanych z zastosowaniem wszystkich opisanych w rozdz. 3 metod dla różnych ich wariantów.

Do opisu eksperymentów każdej konfiguracji dołączono tabele zawierające informacje o zbiorach robotów i zadań tworzących tę konfigurację. Pod każdą taką tabelą znajduje się zrzut ekranu środowiska symulacyjnego prezentujący konfigurację robotów i zadań zgodnie z oznaczeniami omówionymi w rozdz. 4.1. Dodatkowo zamieszczone zostały tabele podsumowujące wszystkie warianty testów wykonanych dla danej konfiguracji. Tabele te zawierają wartości miar oceny efektywności działania systemu (Rozdz. 4.3). Dla wybranych wariantów testów zamieszczono wykresy przedstawiające konfiguracje robotów i zadań po zrealizowaniu misji wraz ze ścieżkami ruchu robotów. Wizualna analiza tych wykresów, a w szczególności przebiegów i długości ścieżek ruchu robotów, pozwala na wstępną ocenę porównawczą metod alokacji zadań. Wykresy te również niosą informacje o tym, który robot które zadanie kiedy wykonał. Dla każdego wariantu testu, dla którego zamieszczono wykres konfiguracji końcowej, wyróżniono wiersz w tabeli podsumowującej misję odpowiadający temu wariantowi.

### 4.6.1. Środowisko 1

Środowisko 1 zawiera przestrzeń otwartą wraz z kilkoma przeszkodami. Dla tego środowiska przeprowadzono eksperymenty dla czterech różnych konfiguracji robotów i zadań.

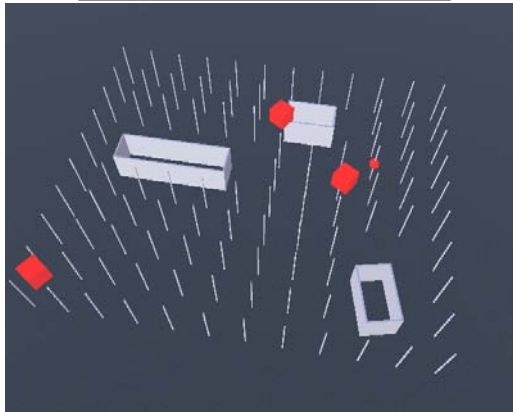
#### **Środowisko 1 – Konfiguracja 1 (wygenerowana przez autora)**

Eksperyment przeprowadzony dla Konfiguracji 1 miał na celu wykazanie słuszności pierwszej tezy postawionej w pracy, dlatego wykonano testy dla wariantu 1. oraz wariantu 3. W ramach tej konfiguracji w misji brał udział jeden robot (typu Inspector) oraz trzy zadania (typu FindHuman). Robot został wprowadzony do systemu w chwili jego (systemu) uruchomienia, natomiast zadanie 0. oraz zadanie 1. wprowadzono w chwili  $t = 1$ , a zadanie 2. w chwili  $t = 17$ .

Pierwsza aukcja miała miejsce w chwili wprowadzenia zadań 0 i 1 do systemu i przebiegła ona tak samo dla obu wariantów, tzn. robotowi 0. przydzielone zostało zadanie 0. W przypadku wariantu 1. wynika to z pozycji zadania 0. w kolejce zadań, natomiast w przypadku wariantu 2. wynika to z tego, że zadanie 0. było najbliżej robota 0. Jeśli

Tab. 4.7: Konfiguracja początkowa robotów (Środowisko 1, Konfiguracja 1)

Identyfikator robota	Robot	$x$	$y$	$t^{in}$
0	1	3	12	0



Tab. 4.8: Konfiguracja zadań (Środowisko 1, Konfiguracja 1)

Identyfikator zadania	Zadanie	$x$	$y$	$t^{in}$	$t^{ex}$
0	FH	3	3	1	10
1	FH	-15	-12	1	10
2	FH	-3	9	17	10

Tab. 4.9: Wyniki przeprowadzonych eksperymentów (Środowisko 1, Konfiguracja 1)

L.p.	Wariant	$MP.am$	$MP.cf$	$MP.aaaa$	$MP.mcfv.am$	$MT$	$TravT$	$TravD$	$TravE$	$UnnResT$
1	1	zadania(ET)				123	92	57,6	145,6	488
2	3	roboty(ET)				101	70	42,8	110,8	400

zadanie 2. byłoby wprowadzone do systemu w tej samej chwili, co zadania 0. i 1., to dla wariantu 1. wynik alokacji zadań byłby ten sam, a dla wariantu 2. byłby inny – robotowi 0. przydzielone zostałyby zadanie 2. ponieważ miejsce jego wykonywania znajduje się nieco bliżej robota 0. niż miejsce wykonywania zadania 0.

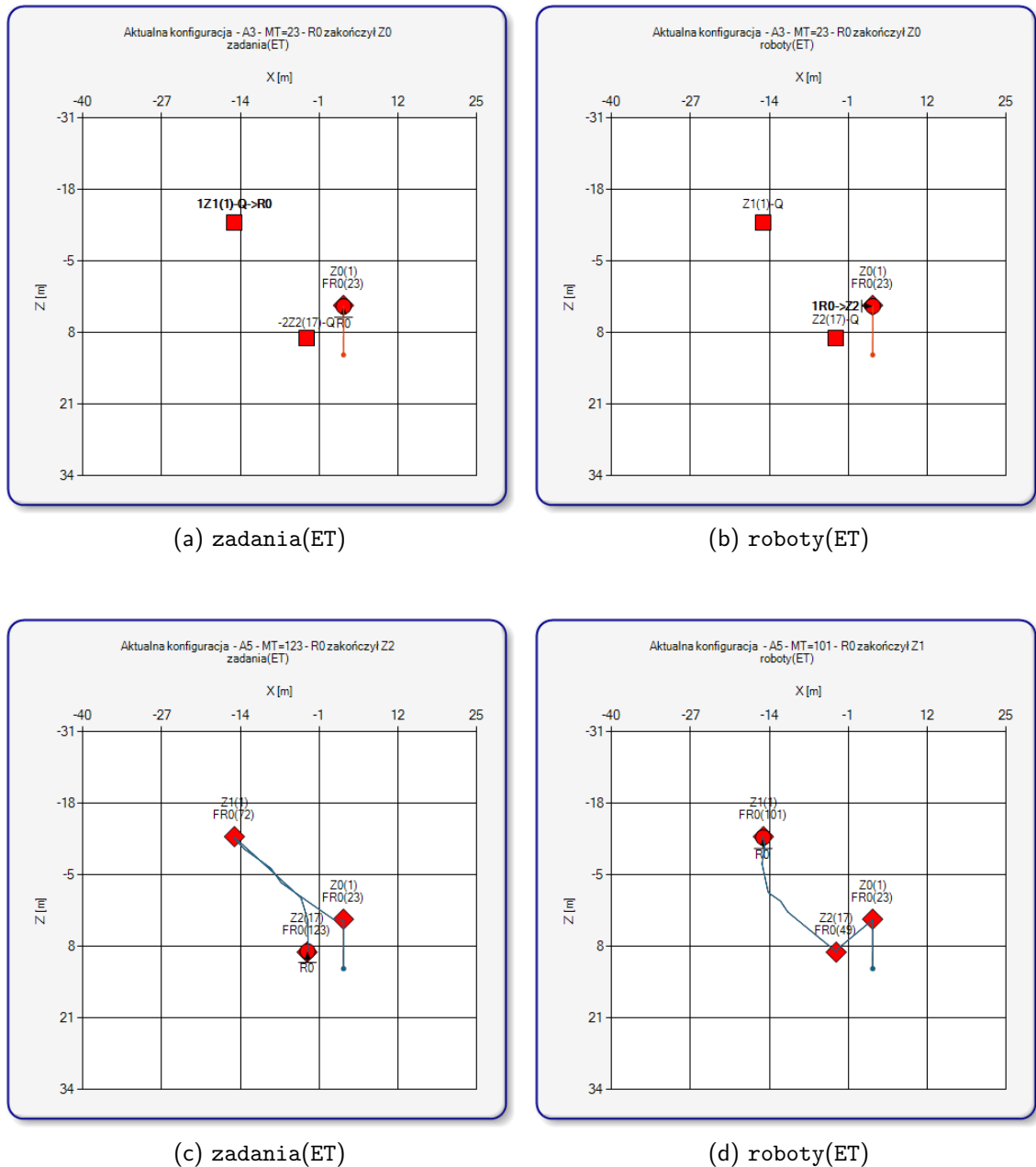
W chwili  $t = 17$  do systemu wprowadzone zostało zadanie 2. W tym momencie robot 0. wykonywał zadanie 0. (dla obu wariantów), więc nie mógł nastąpić przydział zadania. W chwili  $t = 23$  robot 0. zakończył wykonywanie zadania 0. W tym czasie w kolejce zadań znajdowały się dwa zadania. W wyniku przeprowadzonej aukcji (Rys. 4.3a i Rys. 4.3b):

- W przypadku wariantu 1. eksperymentu nastąpił przydział zadania 1. robotowi 0.,
- W przypadku wariantu 3. eksperymentu nastąpił przydział zadania 2. robotowi 0.

Następnie robotowi 0. po wykonaniu przez niego zadania 1. (wariant 1.) lub zadania 2. (wariant 3.) przydzielono odpowiednio zadanie 2. (wariant 1.) lub zadanie 1. (wariant 3.), po wykonaniu których kończy się misja. Przebiegi ścieżek ruchu robotów w ciągu całej misji przedstawiają wykresy z rys. 4.3c i rys. 4.3d.

Na podstawie analizy wyników zawartych w tab. 4.9 można zauważyć, że dla wszystkich miary oceny efektywności działania systemu następuje wzrost efektywności (wartości miar maleją) dla metody stosującej aukcjonowanie robotów w stosunku do metody stosującej aukcjonowanie zadań. Wzrost taki odnotowano nie tylko dla tej prostej konfiguracji,

lecz dla wszystkich – znacznie bardziej złożonych – konfiguracji, dla których wykonano testy (por. Tab. 4.12 – 4.24).



Rys. 4.3: Wykresy tras robotów dla wariantów (Środowisko 1, Konfiguracja 1)

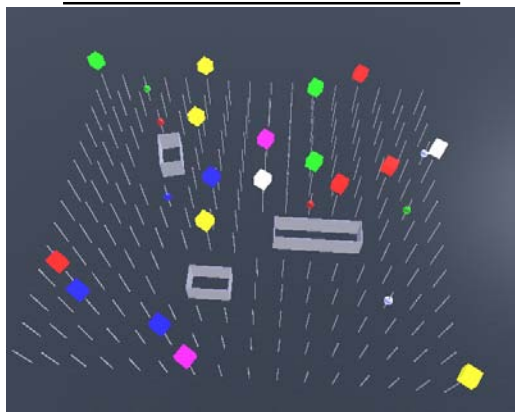
### Środowisko 1 – Konfiguracja 2 (wygenerowana przez autora)

Konfiguracja 2., dla której wykonano eksperymenty za pomocą Środowiska 1., została wygenerowana przez autora i składa się z ośmiu robotów i osiemnastu zadań. Wszystkie roboty zostały wprowadzone do systemu z chwilą jego uruchomienia, natomiast wszystkie zadania wprowadzono do systemu w chwili  $t = 2$ . Należy zauważyć, że dla kolejnych zadań zostały ustalone różne wartości czasów koniecznych na ich wykonywanie  $t^{ex}$ .

Szczegóły dotyczące Konfiguracji 2. zamieszczone zostały w tab. 4.10 oraz tab. 4.11.

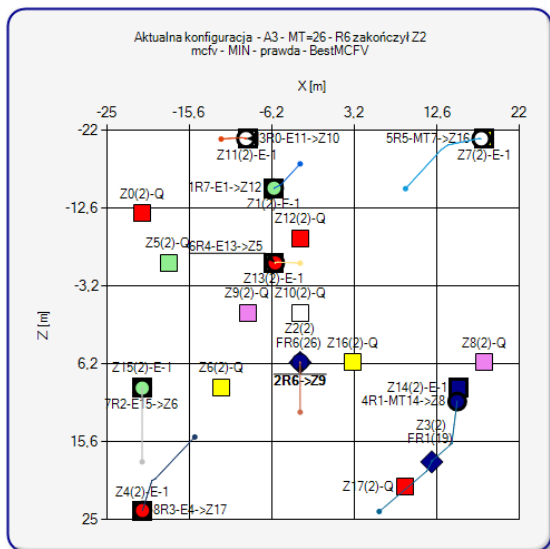
Tab. 4.10: Konfiguracja początkowa robotów (Środowisko 1, Konfiguracja 2)

Identyfikator robota	Robot	$x$	$y$	$t^{in}$
0	C	-12	-21	0
1	E	6	24	0
2	P	-21	18	0
3	I	-15	15	0
4	I	-3	-6	0
5	C	9	-15	0
6	E	-3	12	0
7	P	-3	-18	0

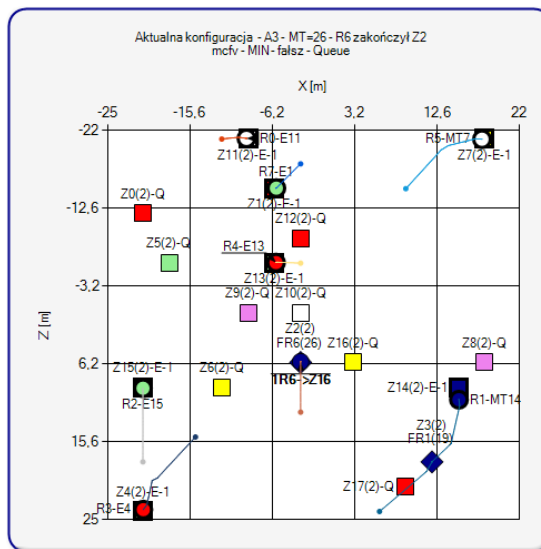


Tab. 4.11: Konfiguracja zadań (Środowisko 1, Konfiguracja 2)

Identyfikator zadania	Zadanie	$x$	$y$	$t^{in}$	$t^{ex}$
0	FH	-21	-12	2	10
1	FH	-6	-15	2	20
2	DF	-3	6	2	20
3	DF	12	18	2	10
4	DG	-21	24	2	45
5	DG	-18	-6	2	10
6	WA	-12	9	2	10
7	WA	18	-21	2	10
8	TB	18	6	2	10
9	TB	-9	0	2	10
10	HC	-3	0	2	25
11	HC	-9	-21	2	25
12	FH	-3	-9	2	25
13	DG	-6	-6	2	35
14	DF	15	9	2	20
15	WA	-21	9	2	20
16	WA	3	6	2	20
17	FH	9	21	2	5



(a) mcfv - MIN - prawda - BestMCFVs



(b) mcfv - MIN - fałsz - Queue

Rys. 4.4: Wykresy przedstawiające przykłady alokacji zadań będących wynikiem dwóch aukcji wywołanych przez zakończenie wykonywania zadania 2. przez robota 6. (Środowisko 1, Konfiguracja 2)

Tab. 4.12: Wyniki przeprowadzonych eksperymentów (Środowisko 1, Konfiguracja 2)

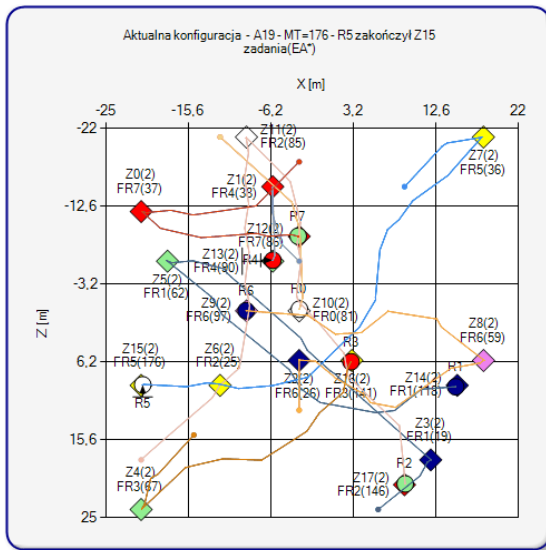
Lp.	Wariant	<i>MP.am</i>	<i>MP.cf</i>	<i>MP.aaaa</i>	<i>MP.mcfv.am</i>	<i>MT</i>	<i>TravT</i>	<i>TravD</i>	<i>TravE</i>	<i>UnnResT</i>
1	1	zadania(ET)				177	580	422,3	1357,3	2396
2	2	zadania(EE)				176	594	433,7	1399,1	2470
3	3	roboty(ET)				144	327	249,1	869,9	1904
4	4	roboty(EE)				154	282	217,5	667,8	1782
5	5	mcfv	MIN	falsz	Queue	109	276	198,8	639,6	1513
6	6	mcfv	MIN	falsz	BestMeanOfMCFVs	109	274	198,5	650,6	1512
7	7	mcfv	MIN	falsz	BestMCFV	110	276	199,9	639,4	1513
8	8	mcfv	MIN	prawda	BestMeanOfMCFVs	127	324	219,7	594,2	1385
9	9	mcfv	MIN	prawda	BestMCFV	128	307	205,4	552,8	1356
10	10	mcfv	MIN-MAX	falsz	Queue	110	275	198,8	632,1	1504
11	11	mcfv	MIN-MAX	falsz	BestMeanOfMCFVs	110	279	199,3	653,4	1526
12	12	mcfv	MIN-MAX	falsz	BestMCFV	110	276	198,5	643,7	1515
13	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	140	352	232,4	555,5	1414
14	14	mcfv	MIN-MAX	prawda	BestMCFV	127	307	205,9	544,0	1355
15	1	zadania(ET)				176	575	420,7	1350,1	2381
16	2	zadania(EE)				175	589	432,3	1372,2	2453
17	3	roboty(ET)				144	326	249,0	868,1	1903
18	4	roboty(EE)				156	290	219,6	690,4	1802
19	5	mcfv	MIN	falsz	Queue	110	273	198,7	629,1	1503
20	6	mcfv	MIN	falsz	BestMeanOfMCFVs	110	272	199,7	631,6	1506
21	7	mcfv	MIN	falsz	BestMCFV	110	274	198,6	631,3	1502
22	8	mcfv	MIN	prawda	BestMeanOfMCFVs	128	321	219,1	578,6	1377
23	9	mcfv	MIN	prawda	BestMCFV	128	311	205,2	574,6	1371
24	10	mcfv	MIN-MAX	falsz	Queue	110	273	199,6	624,2	1499
25	11	mcfv	MIN-MAX	falsz	BestMeanOfMCFVs	110	272	198,6	622,9	1494
26	12	mcfv	MIN-MAX	falsz	BestMCFV	109	270	198,0	629,5	1496
27	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	141	348	231,0	543,4	1405
28	14	mcfv	MIN-MAX	prawda	BestMCFV	128	308	205,5	560,1	1359

Tab. 4.12 zawiera zestawienie wyników dwóch serii omawianego eksperymentu. W dalszym ciągu przeprowadzono szczegółową dyskusję tych wyników. Na podstawie analizy tab. 4.12 oraz wykresów z rys. 4.5 można zauważyć, że dla wariantów eksperymentów:

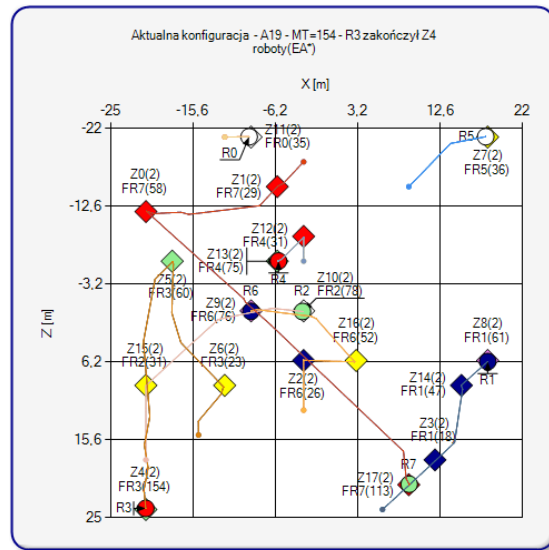
- 1 i 2 – wartości miar ocen efektywności działania systemu są zbliżone, a więc dla aukcjonowania zadań na rozdział zadań większy wpływ miało wzajemne rozmieszczenie przestrzenne robotów i zadań, aniżeli przyjęte kryterium oceny ofert,
- 1,2 i 3,4 – aukcjonowanie robotów daje znacznie lepsze wyniki w stosunku do aukcjonowania zadań,
- 3 i 4 – dla aukcjonowania robotów lepszym kryterium jest energia – tylko czas misji *MT* jest większy dla eksperymentu 4.,
- 1-4 i 5-14 – metody wielokryterialne dla większości miar oceny efektywności działania systemu dają lepsze wyniki w stosunku do metod stosujących aukcjonowanie robotów, a tym bardziej do metod stosujących aukcjonowanie zadań,
- 5-7 i 8-9 – metody oparte o algorytm MIN, stosujące aukcjonowanie wszystkich robotów w stosunku do metod stosujących aukcjonowanie robotów tylko niezajętych dają korzystniejsze wyniki dla miar *TravE*, *UnnResT*, natomiast gorsze dla *MT*, *TravT*, *TravD*,
- 10-12 i 13-14 – komentarz j.w., z tym że dotyczy metod stosujących algorytm MIN-MAX,
- 5 i 9 – dla metod MCFV (Rys. 4.4a, Rys. 4.4b) aukcjonowanie wszystkich robotów (wariant 9) spowodowało, że w momencie zakończenia wykonywania przez robota 6. zadania 2., oferta robota 6. na wykonanie zadania 16. była gorsza w stosunku do oferty złożonej przez innego robota (zajętego), w wyniku czego robot 6. podjął

się zadania 9., a zadanie 16. zostało chwilowo zablokowane. W chwili  $t = 37$  zadanie 16. zostało przydzielone robotowi 5., który następnie zmierzał do miejsca wykonywania tego zadania aż do  $t = 108$  (dla zadania 16.  $z.\tau.t^{ex} = 20, z.\tau.t^f = 128$ ), co znacznie wydłużyło parametry  $MT, TravT, TravD$ , ale mimo to dla tej metody alokacji zadań najniższe są wartości parametrów  $TravE, UnnResT$ , co jest spowodowane tym, że robot 5. ma niskie zużycie energii w trakcie jazdy  $ec^{trav}$  oraz tylko jeden nadmiarowy zasób inspekcyjny dla zadania 16.,

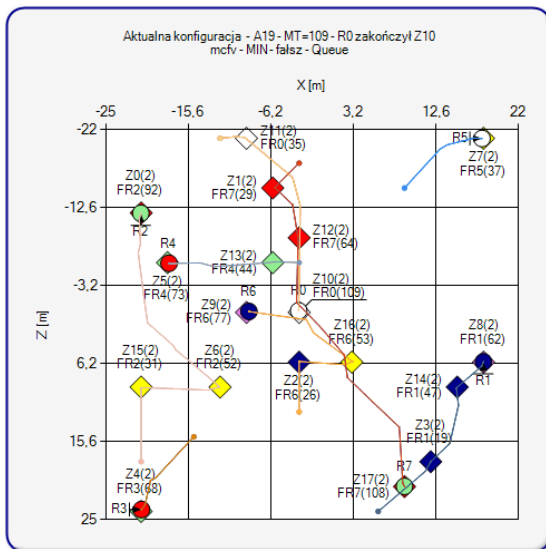
- 5-9 i 10-14 – wybór algorytmu (MIN lub MIN-MAX) wyznaczania wartości kryteriów dla metod MCFV nie miał wpływu na wartości wyników.



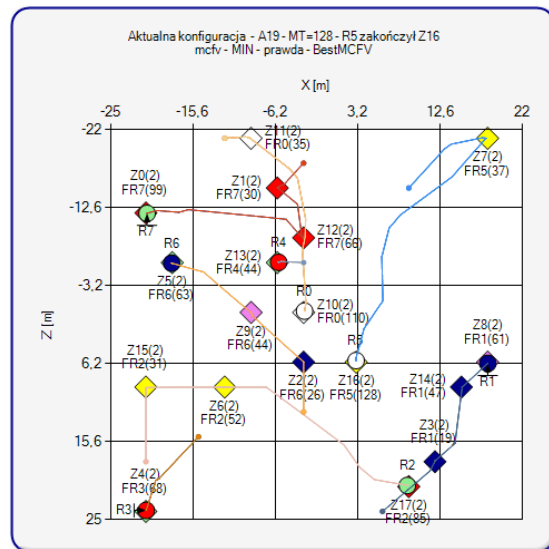
(a) zadania(EA)



(b) roboty(EA)



(c) mcfv - MIN - fałsz - Queue



(d) mcfv - MIN - prawda - BestMCFV

Rys. 4.5: Wykresy tras robotów dla wariantów (Środowisko 1, Konfiguracja 2)

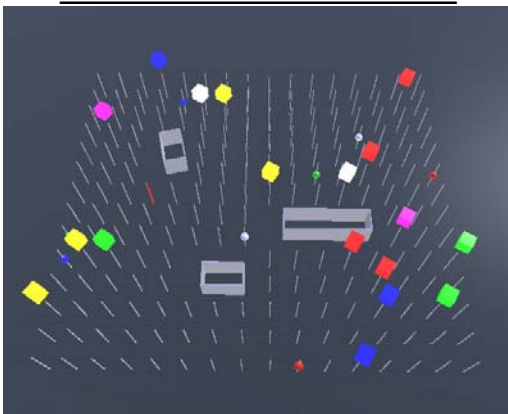


### Środowisko 1 – Konfiguracja 3 (losowa)

Konfiguracja 3. została wygenerowana na podstawie Konfiguracji 2. Liczby, typy, czasy wprowadzenia robotów i zadań, a także czasy wykonywania zadań są takie same jak w Konfiguracji 2., natomiast losowo zostały wygenerowane początkowe pozycje robotów i miejsca wykonywania zadań. Tab. 4.13 oraz tab. 4.14 zawierają szczegółowe informacje o tej konfiguracji, a rysunek umieszczony pod tab. 4.13 przedstawia Środowisko 1. wraz z początkową Konfiguracją 3.

Tab. 4.13: Konfiguracja początkowa robotów (Środowisko 1, Konfiguracja 3)

Identyfikator robota	Robot	$x$	$y$	$t^{in}$
0	W	-12	-12	0
1	E	-18	12	0
2	P	-6	-6	0
3	I	18	-3	0
4	I	-6	-21	0
5	W	3	3	0
6	E	6	24	0
7	P	3	24	0



Tab. 4.14: Konfiguracja zadań (Środowisko 1, Konfiguracja 3)

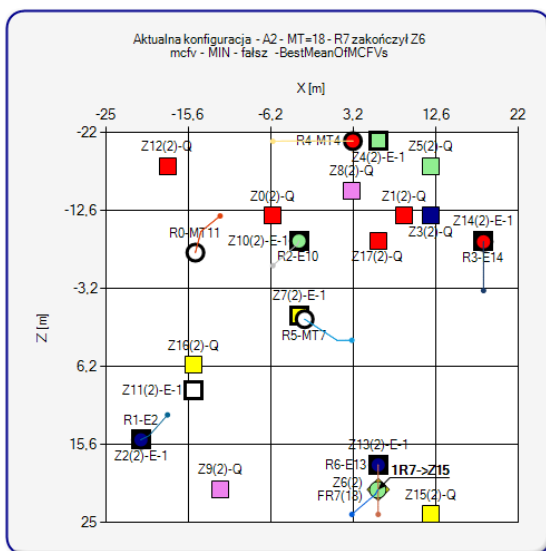
Identyfikator zadania	Zadanie	$x$	$y$	$t^{in}$	$t^{ex}$
0	FH	-6	-12	2	10
1	FH	9	-12	2	20
2	DF	-21	15	2	20
3	DF	12	-12	2	10
4	DG	6	-21	2	45
5	DG	12	-18	2	10
6	WA	6	21	2	10
7	WA	-3	0	2	10
8	TB	3	-15	2	10
9	TB	-12	21	2	10
10	HC	-3	-9	2	25
11	HC	-15	9	2	25
12	FH	-18	-18	2	25
13	DG	6	18	2	35
14	DF	18	-9	2	20
15	WA	12	24	2	20
16	WA	-15	6	2	20
17	FH	6	-9	2	5

Tab. 4.15 zawiera zestawienie wyników dwóch serii omawianego eksperymentu. W dalszym ciągu przeprowadzono szczegółową dyskusję tych wyników. Na podstawie analizy tab. 4.15 oraz wykresów z rys. 4.7 można zauważyć, że dla wariantów eksperymentów:

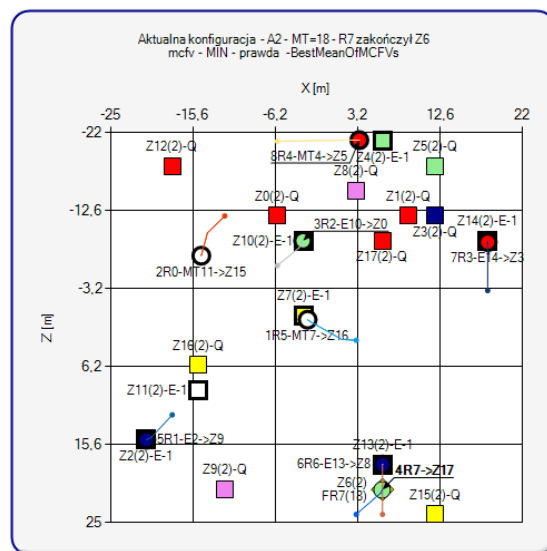
- 1 i 2 – wartości miar ocen efektywności działania systemu są zbliżone, a więc dla aukcjonowania zadań na rozdział zadań większy wpływ miało wzajemne rozmieszczenie przestrzenne robotów i zadań, aniżeli przyjęte kryterium oceny ofert,
- 1,2 i 3,4 – aukcjonowanie robotów daje lepsze wyniki w stosunku do aukcjonowania zadań, najlepsze dla kryterium zużycia energii. Czasy misji  $MT$  poszczególnych eksperymentów są zbliżone,
- 1-4 i 5-14 – metody wielokryterialne dają lepsze wyniki w stosunku do metod stosujących aukcjonowanie robotów, a tym bardziej do metod stosujących aukcj-

Tab. 4.15: Wyniki przeprowadzonych eksperymentów (Środowisko 1, Konfiguracja 3)

L.p.	Variant	MP.am	MP.cf	MP.auaa	MP.mcfv.am	MT	TravT	TravD	TravE	UnnResT
1	1	zadania(ET)				130	537	409,4	1372,6	2230
2	2	zadania(EE)				136	538	415,5	1439,1	2214
3	3	roboty(ET)				129	400	299,9	1199,2	2124
4	4	roboty(EE)				130	344	234,9	864,0	1979
5	5	mcfv	MIN	fałsz	Queue	110	303	227,1	791,7	1629
6	6	mcfv	MIN	fałsz	BestMeanOfMCFVs	111	285	210,8	764,0	1734
7	7	mcfv	MIN	fałsz	BestMCFV	109	281	210,6	761,5	1728
8	8	mcfv	MIN	prawda	BestMeanOfMCFVs	98	289	217,6	776,2	1629
9	9	mcfv	MIN	prawda	BestMCFV	112	302	212,0	701,6	1609
10	10	mcfv	MIN-MAX	fałsz	Queue	129	370	283,8	1089,9	1993
11	11	mcfv	MIN-MAX	fałsz	BestMeanOfMCFVs	128	367	282,8	1075,4	1980
12	12	mcfv	MIN-MAX	fałsz	BestMCFV	131	374	283,2	1090,7	2000
13	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	101	269	195,0	667,9	1506
14	14	mcfv	MIN-MAX	prawda	BestMCFV	108	279	198,0	674,1	1532
15	1	zadania(ET)				128	521	399,5	1333,5	2201
16	2	zadania(EE)				139	546	414,8	1457,0	2240
17	3	roboty(ET)				132	397	299,4	1188,8	2109
18	4	roboty(EE)				129	333	230,4	826,6	1946
19	5	mcfv	MIN	fałsz	Queue	110	305	227,9	799,0	1636
20	6	mcfv	MIN	fałsz	BestMeanOfMCFVs	109	279	210,3	738,2	1716
21	7	mcfv	MIN	fałsz	BestMCFV	109	320	248,3	931,5	1652
22	8	mcfv	MIN	prawda	BestMeanOfMCFVs	99	292	218,0	791,3	1643
23	9	mcfv	MIN	prawda	BestMCFV	111	301	212,9	700,5	1609
24	10	mcfv	MIN-MAX	fałsz	Queue	130	371	282,8	1086,6	1998
25	11	mcfv	MIN-MAX	fałsz	BestMeanOfMCFVs	129	371	283,6	1096,7	1995
26	12	mcfv	MIN-MAX	fałsz	BestMCFV	130	372	282,8	1083,6	1991
27	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	101	270	194,7	660,5	1503
28	14	mcfv	MIN-MAX	prawda	BestMCFV	107	278	198,6	671,7	1536



(a) mcfv – MIN – fałsz – BestMeanOfMCFVs



(b) mcfv – MIN – prawda – BestMeanOfMCFVs

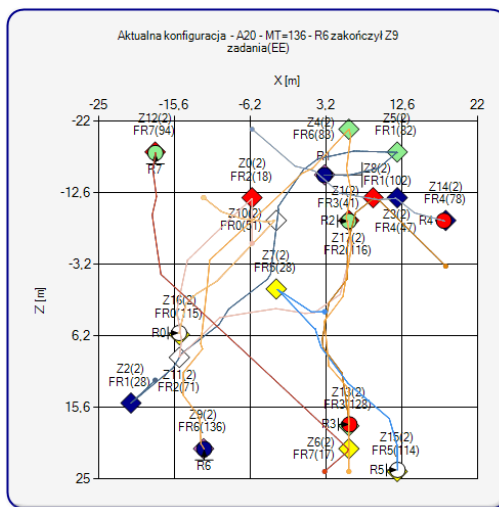
Rys. 4.6: Wykresy przedstawiające przykłady alokacji zadań będących wynikiem dwóch aukcji wywołanych przez zakończenie wykonywania zadania 6. przez robota 7.

(Środowisko 1, Konfiguracja 3)

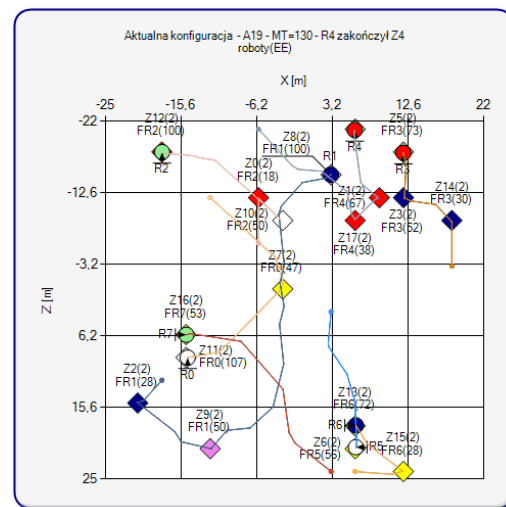
nowanie zadań. Wyjątkiem są metody aukcjonujące tylko roboty niezajęte dla algorytmu MIN-MAX (warianty 10-12), dla których wartości miar oceny są zbliżone do miar dla metod stosujących aukcjonowanie robotów,

- 13-14 – metody MCFV stosujące aukcjonowanie wszystkich robotów oraz algorytm MIN-MAX dały najlepsze wyniki,
- 6 i 8 – z rys. 4.7c oraz 4.7d można wywnioskować, że robot 7. (typu Pathfinder)

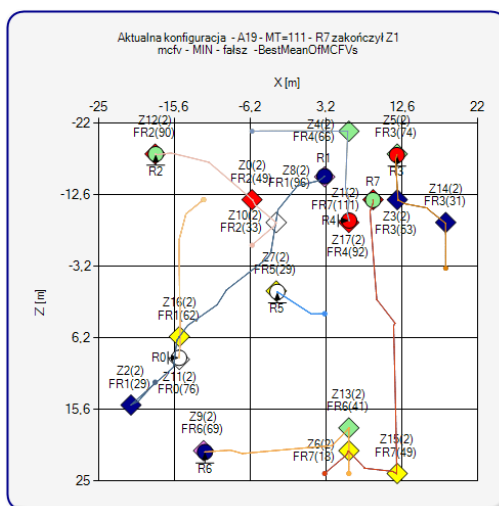
kończąc zadanie 6. przystąpił do aukcji i złożył ofertę na wykonanie m.in. zadania 15. (typu WatchArea). W tym czasie robot 6. (typu Explorer) wykonywał zadanie 13. W wariancie 6. robot 6. nie mógł złożyć oferty na wykonanie zadania 15., przez co pozycja konkurencyjna robota 7. była wysoka i przydzielone zostało mu zadanie 15. W wariancie 8. robot 6. mógł złożyć już taką ofertę, przez co osłabił pozycję konkurencyjną robota 7., w następstwie czego bardziej *przystosowany* robot 0. zablokował zadanie 15. Dzięki temu robot 7. pojechał w kierunku miejsce wykonywania innych zadań i zakończył zadanie 1. dla  $t = 84$  – dla wariantu 6. zakończył zadanie 1. dla  $t = 111$  będącego czasem misji. Podsumowując, dla tych dwóch wariantów aukcjonowanie wszystkich robotów przyczyniło się do skrócenia czasu misji.



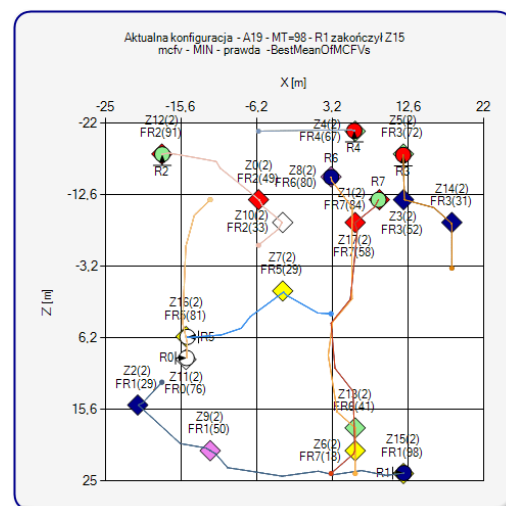
(a) zadania(EE)



(b) roboty(EE)



(c) mcfv - MIN - fałsz - BestMeanOfMCFVs



(d) mcfv - MIN - prawda - BestMeanOfMCFVs

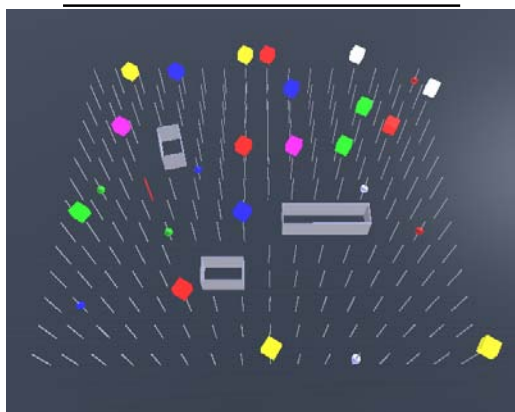
Rys. 4.7: Wykresy tras robotów dla wariantów (Środowisko 1, Konfiguracja 3)

### Środowisko 1 – Konfiguracja 4 (losowa)

Konfiguracja 4. została wygenerowana na podstawie Konfiguracji 2. Liczby, typy, czasy wprowadzenia robotów i zadań, a także czasy wykonywania zadań są takie same jak w Konfiguracji 2., natomiast losowo zostały wygenerowane początkowe pozycje robotów i miejsca wykonywania zadań. Tab. 4.17 oraz tab. 4.16 zawierają szczegółowe informacje o tej konfiguracji, a rysunek umieszczony pod tab. 4.16 przedstawia Środowisko 1. wraz z początkową Konfiguracją 4.

Tab. 4.16: Konfiguracja początkowa robotów (Środowisko 1, Konfiguracja 4)

Identyfikator robota	Robot	$x$	$y$	$t^{in}$
0	W	18	-9	0
1	E	-6	9	0
2	P	-3	21	0
3	I	-21	-21	0
4	I	3	-18	0
5	W	-3	-12	0
6	E	12	21	0
7	P	3	12	0

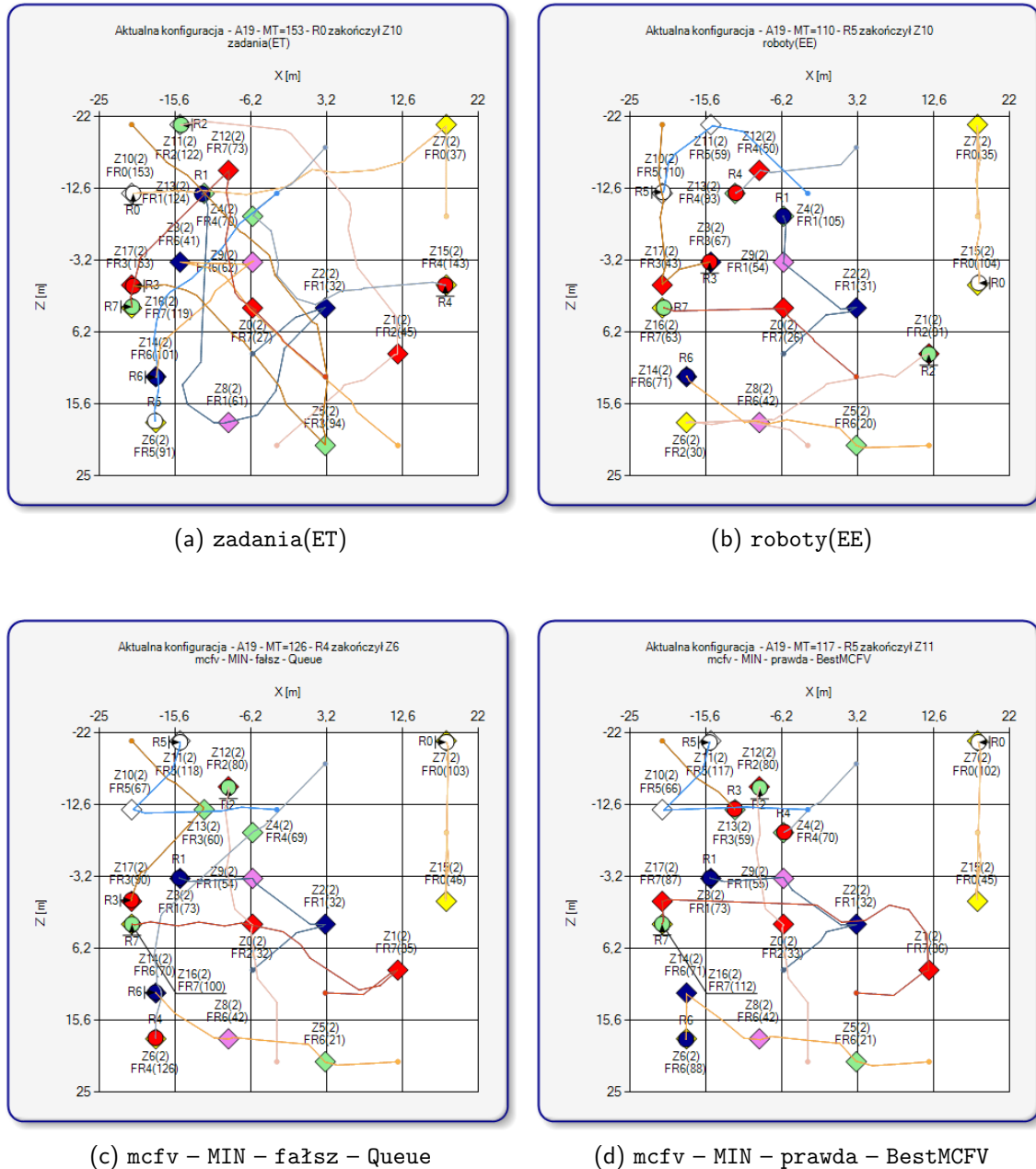


Tab. 4.17: Konfiguracja zadań (Środowisko 1, Konfiguracja 4)

Identyfikator zadania	Zadanie	$x$	$y$	$t^{in}$	$t^{ex}$
0	FH	-6	3	2	10
1	FH	12	9	2	20
2	DF	3	3	2	20
3	DF	-15	-3	2	10
4	DG	-6	-9	2	45
5	DG	3	21	2	10
6	WA	-18	18	2	10
7	WA	18	-21	2	10
8	TB	-9	18	2	10
9	TB	-6	-3	2	10
10	HC	-21	-12	2	25
11	HC	-15	-21	2	25
12	FH	-9	-15	2	25
13	DG	-12	-12	2	35
14	DF	-18	12	2	20
15	WA	18	0	2	20
16	WA	-21	3	2	20
17	FH	-21	0	2	5

Tab. 4.18 zawiera zestawienie wyników omawianego eksperymentu. W dalszym ciągu przeprowadzono szczegółową dyskusję tych wyników. Na podstawie analizy tab. 4.18 oraz wykresów z rys. 4.8 można zauważyć, że dla wariantów eksperymentów:

- 1 i 2 – wartości miar ocen efektywności działania systemu są różne, a więc dla aukcjonowania zadań korzystniejszym było kryterium energetyczne,
- 1,2 i 3,4 i 8,9 – aukcjonowanie robotów daje lepsze wyniki w stosunku do aukcjonowania zadań, najlepsze dla kryterium zużycia energii. Dla tej konfiguracji metoda stosująca aukcjonowanie robotów daje wyniki porównywalne z wynikami najlepszych metod spośród metod MCFV,
- 5-7 i 8,9 i 10-12, 13,14 – dla metod MCFV lepsze wyniki otrzymywane są podczas stosowania algorytmu MIN (w stosunku do algorytmu MIN-MAX),
- 5-7 i 8,9 i 10-12, 13,14 – dla metod MCFV lepsze wyniki otrzymywane są, jeśli aukcjonowane są wszystkie roboty aktywne,



Rys. 4.8: Wykresy tras robotów dla wariantów (Środowisko 1, Konfiguracja 4)

- 5-7 i 8,9 i 10-12 – zbliżone wyniki dla każdej z grup wariantów wskazują na identyczne przydziały zadań dla każdego wariantu w poszczególnych grupach, a więc mało znaczący był wpływ metody przydzielania zadań MCFV (Rozdz. 3.4.5), a znaczenie miały algorytmy wyznaczania wartości kryteriów dla metod MCFV oraz aukcjonowanie wszystkich robotów aktywnych.

Tab. 4.18: Wyniki przeprowadzonych eksperymentów (Środowisko 1, Konfiguracja 4)

L.p.	Wariant	<i>MP.am</i>	<i>MP.cf</i>	<i>MP.ouaa</i>	<i>MP.mcfv-am</i>	<i>MT</i>	<i>TravT</i>	<i>TravD</i>	<i>TravE</i>	<i>UnnResT</i>
1	1	zadania(ET)				153	663	476,3	1331,4	2765
2	2	zadania(EE)				152	558	382,2	1095,3	2399
3	3	roboty(ET)				142	416	312,9	1145,9	2200
4	4	roboty(EE)				110	354	240,6	789,6	1790
5	5	mcfv	MIN	fałsz	Queue	126	409	278,6	886,0	1961
6	6	mcfv	MIN	fałsz	BestMeanOfMCFVs	126	405	277,8	890,2	1954
7	7	mcfv	MIN	fałsz	BestMCFV	125	405	277,5	880,6	1952
8	8	mcfv	MIN	prawda	BestMeanOfMCFVs	116	347	243,7	839,8	1553
9	9	mcfv	MIN	prawda	BestMCFV	117	349	243,2	838,3	1560
10	10	mcfv	MIN-MAX	fałsz	Queue	151	449	298,6	927,1	2032
11	11	mcfv	MIN-MAX	fałsz	BestMeanOfMCFVs	150	445	298,5	920,5	2018
12	12	mcfv	MIN-MAX	fałsz	BestMCFV	150	449	300,1	937,2	2035
13	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	149	421	285,6	825,5	1765
14	14	mcfv	MIN-MAX	prawda	BestMCFV	127	368	256,6	835,2	1661

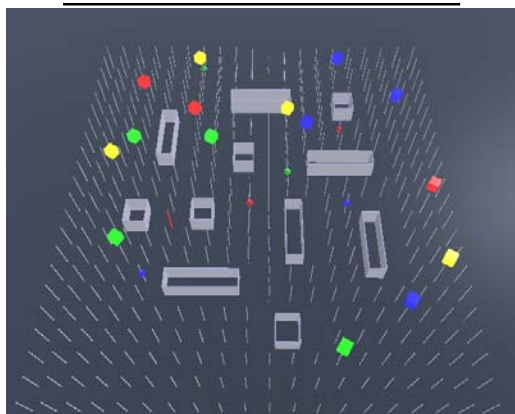
## 4.6.2. Środowisko 2

### Środowisko 2 - Konfiguracja 5 (wygenerowana przez autora)

Konfiguracja 5., dla której wykonano eksperymenty za pomocą Środowiska 2., została wygenerowana przez autora i składa się z sześciu robotów i piętnastu zadań. Wszystkie roboty zostały wprowadzone do systemu z chwilą jego uruchomienia, natomiast część zadań wprowadzono do systemu w chwili  $t = 2$ , a część w chwili  $t = 10$ , kiedy to roboty wykonywały zadania wprowadzone wcześniej. Należy zauważyć, że dla kolejnych zadań zostały ustalone różne wartości czasów koniecznych na ich wykonywanie  $t^{ex}$ . Szczegóły dotyczące Konfiguracji 5. zamieszczone zostały w tab. 4.19 oraz tab. 4.20.

Tab. 4.19: Konfiguracja początkowa robotów (Środowisko 2, Konfiguracja 5)

Identyfikator robota	Robot	$x$	$y$	$t^{in}$
0	E	-6	-12	0
1	P	-12	-3	0
2	I	-21	-12	0
3	I	-6	3	0
4	P	-36	12	0
5	E	6	18	0



Tab. 4.20: Konfiguracja zadań (Środowisko 2, Konfiguracja 5)

Identyfikator zadania	Zadanie	$x$	$y$	$t^{in}$	$t^{ex}$
0	WA	6	-24	2	15
1	FH	-21	12	2	20
2	FH	-27	21	2	30
3	DF	-33	-12	2	10
4	DF	-24	-21	2	10
5	DG	18	-9	2	15
6	DG	-15	21	2	15
7	WA	-33	12	10	15
8	WA	-12	24	10	15
9	FH	-6	-24	10	20
10	DG	3	21	10	10
11	DF	12	-18	10	25
12	DG	-15	9	10	10
13	WA	-21	-3	10	20
14	DF	-18	-6	10	25

Tab. 4.21 zawiera zestawienie wyników omawianego eksperymentu. W dalszym ciągu

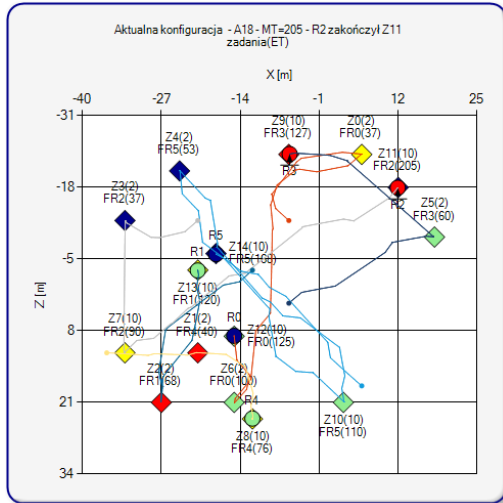
przeprowadzono szczegółową dyskusję tych wyników. Na podstawie analizy tab. 4.21 oraz wykresów z rys. 4.9 można zauważyć, że dla wariantów eksperymentów:

- 1 i 2 – wartości miar ocen efektywności działania systemu są zbliżone, a więc dla aukcjonowania zadań na rozdział zadań większy wpływ miało wzajemne rozmieszczenie przestrzenne robotów i zadań, aniżeli przyjęte kryterium oceny ofert,
- 1,2 i 3,4 – aukcjonowanie robotów daje znacznie lepsze wyniki w stosunku do aukcjonowania zadań,
- 3,4 i 10-14 – metody MCFV (poza metodą z wariantu 14.) stosujące algorytm wyznaczania wartości kryteriów MIN-MAX dają wyniki nieco gorsze niż metody stosujące aukcjonowanie robotów,
- 9 i 14 – najlepsze rezultaty daje metoda największej wartości  $mcfv^{r \rightarrow z}$  stosowana podczas aukcjonowania wszystkich robotów aktywnych niezależnie od użytego algorytmu wyznaczania wartości kryteriów (MIN lub MIN-MAX).

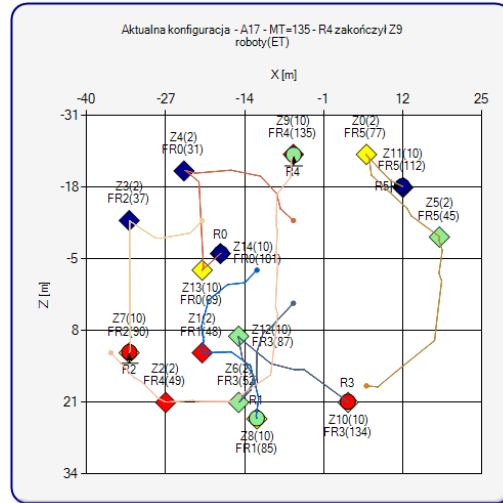
Tab. 4.21: Wyniki przeprowadzonych eksperymentów (Środowisko 2, Konfiguracja 5)

L.p.	Wariant	$MP.am$	$MP.cf$	$MP.aaaa$	$MP.mcfv.am$	$MT$	$TravT$	$TravD$	$TravE$	$UnnResT$
1	1	zadania(ET)				205	560	478,7	1829,6	2654
2	2	zadania(EE)				211	529	438,9	1649,2	2495
3	3	roboty(ET)				135	391	311,1	1025,9	1915
4	4	roboty(EE)				156	354	302,4	1133,4	1701
5	5	mcfv	MIN	falsz	Queue	153	356	299,4	1145,8	1688
6	6	mcfv	MIN	falsz	BestMeanOfMCFVs	134	339	256,2	976,1	1862
7	7	mcfv	MIN	falsz	BestMCFV	135	344	255,3	984,3	1879
8	8	mcfv	MIN	prawda	BestMeanOfMCFVs	142	340	275,0	1060,2	1595
9	9	mcfv	MIN	prawda	BestMCFV	142	337	275,7	1055,1	1585
10	10	mcfv	MIN-MAX	falsz	Queue	136	390	323,6	1081,9	1907
11	11	mcfv	MIN-MAX	falsz	BestMeanOfMCFVs	150	420	323,4	1056,8	2011
12	12	mcfv	MIN-MAX	falsz	BestMCFV	149	419	325,5	1059,9	2015
13	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	148	407	312,9	1056,8	1924
14	14	mcfv	MIN-MAX	prawda	BestMCFV	142	342	276,9	1078,6	1597

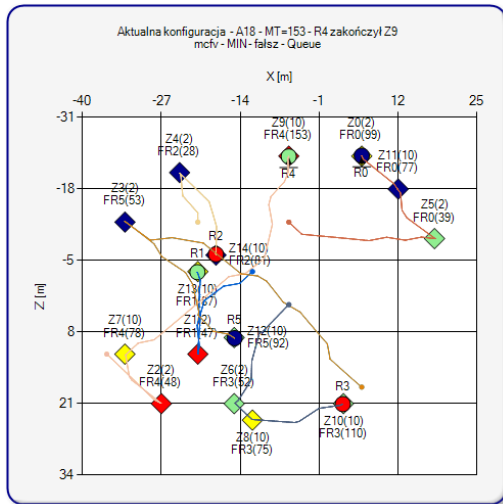




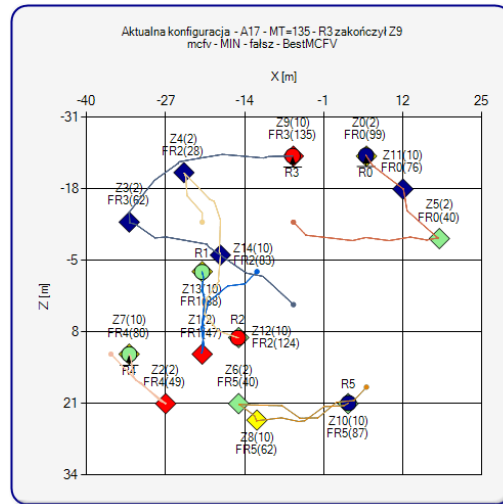
(a) zadania(ET)



(b) roboty(ET)



(c) mcfv - MIN - fałsz - Queue



(d) mcfv - MIN - prawda - BestMCFV

Rys. 4.9: Wykresy tras robotów dla wariantów (Środowisko 2, Konfiguracja 5)

### Środowisko 2 - Konfiguracja 6 (losowa)

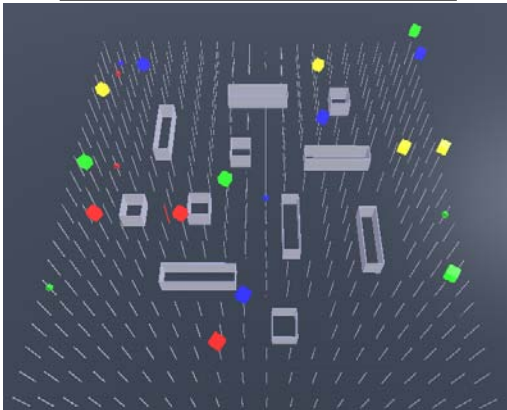
Konfiguracja 6. została wygenerowana na podstawie Konfiguracji 5. Liczby, typy, czasy wprowadzenia robotów i zadań, a także czasy wykonywania zadań są takie same jak w Konfiguracji 5., natomiast losowo zostały wygenerowane początkowe pozycje robotów i miejsca wykonywania zadań. Tab. 4.22 oraz tab. 4.23 zawierają szczegółowe informacje o tej konfiguracji, a rysunek umieszczony pod tab. 4.22 przedstawia Środowisko 2. wraz z początkową Konfiguracją 5.

Tab. 4.24 zawiera zestawienie wyników omawianego eksperymentu. W dalszym ciągu przeprowadzono szczegółową dyskusję tych wyników. Na podstawie analizy tab. 4.24 oraz wykresów z rys. 4.10 można zauważyć, że dla wariantów eksperymentów:



Tab. 4.22: Konfiguracja początkowa robotów (Środowisko 2, Konfiguracja 6)

Identyfikator robota	Robot	$x$	$y$	$t^{in}$
0	E	-6	0	0
1	P	-3	-27	0
2	I	-33	27	0
3	I	-12	24	0
4	P	9	30	0
5	E	-36	27	0



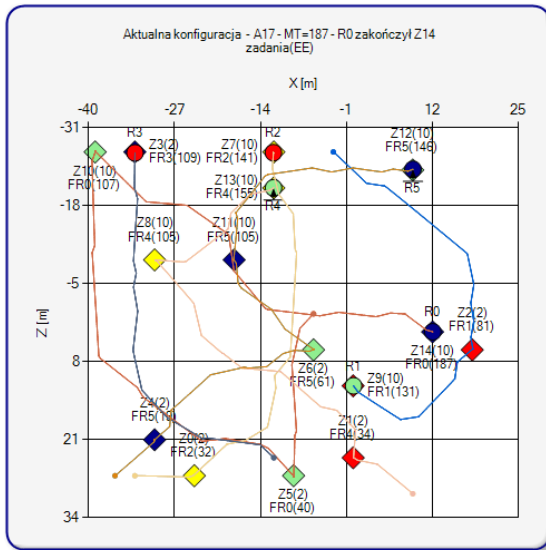
Tab. 4.23: Konfiguracja zadań (Środowisko 2, Konfiguracja 6)

Identyfikator zadania	Zadanie	$x$	$y$	$t^{in}$	$t^{ex}$
0	WA	-24	27	2	15
1	FH	0	24	2	20
2	FH	18	6	2	30
3	DF	-33	-27	2	10
4	DF	-30	21	2	10
5	DG	-9	27	2	15
6	DG	-6	6	2	15
7	WA	-12	-27	10	15
8	WA	-30	-9	10	15
9	FH	0	12	10	20
10	DG	-39	-27	10	10
11	DF	-18	-9	10	25
12	DG	9	-24	10	10
13	WA	-12	-21	10	20
14	DF	12	3	10	25

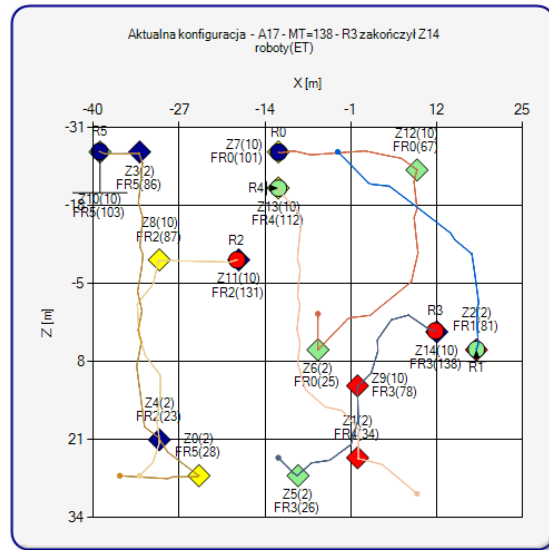
- 1-4 – wartości miar ocen efektywności działania systemu są różne, a korzystniejszym było kryterium czasowe,
- 1,3 – dla kryterium czasowego efektywniejszym jest aukcjonowanie robotów niż aukcjonowanie zadań,
- 2,3 – dla kryterium energetycznego efektywniejszym jest również aukcjonowanie robotów niż aukcjonowanie zadań,
- 3,9,14 – bardzo dobre wyniki otrzymano tylko dla wariantu 3. (aukcjonowanie robotów stosujące kryterium czasowe). Trochę lepsze (najlepsze dla wszystkich wariantów) rezultaty dają metody największej wartości  $mcfv^{r \rightarrow z}$  stosujące aukcjonowanie wszystkich robotów aktywnych niezależnie od użytego algorytmu wyznaczania wartości kryteriów (MIN lub MIN-MAX),
- 3,4 i 5 – wyniki dla wariantów 3.,4. są zbliżone (wartości jednych miar są lepsze, innych gorsze) do wyników wariantu 5. Z racji tego, że dla każdego z tych trzech wariantów zadania przydzielane są robotom niezajętym zgodnie z kolejką tych robotów, zauważa się (przynajmniej dla wariantu 5.) brak wzrostu efektywności działania systemu wynikający z zastosowania wielokryterialnej oceny przystosowania robotów do zadań.
- 5-9 i 10-14 – porównując wyniki odpowiadających sobie wariantów dla algorytmów MIN oraz MIN-MAX można zauważyć, że wyniki te są podobne (wartości jednych miar są lepsze, innych gorsze), a więc dla tej konfiguracji wybór algorytmu wyznaczania wartości kryteriów nie miał większego znaczenia.
- 5-8 i 9 – dla algorytmu MIN wariant, dla którego zastosowano aukcjonowanie

wszystkich robotów aktywnych oraz metodę największej wartości  $mcfv^{r \rightarrow z}$ , daje najlepsze (poza  $TravE$ ) wyniki,

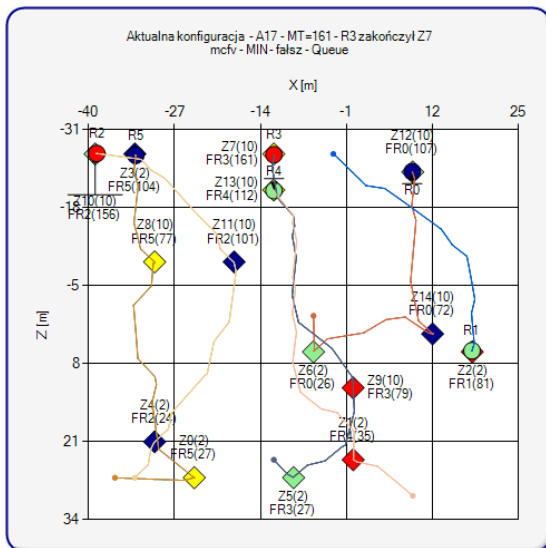
- 10-13 i 14 – dla algorytmu MIN-MAX wariant, dla którego zastosowano aukcjonowanie wszystkich robotów aktywnych oraz metodę największej wartości  $mcfv^{r \rightarrow z}$ , daje najlepsze wyniki dla wszystkich miar oceny efektywności działania systemu.



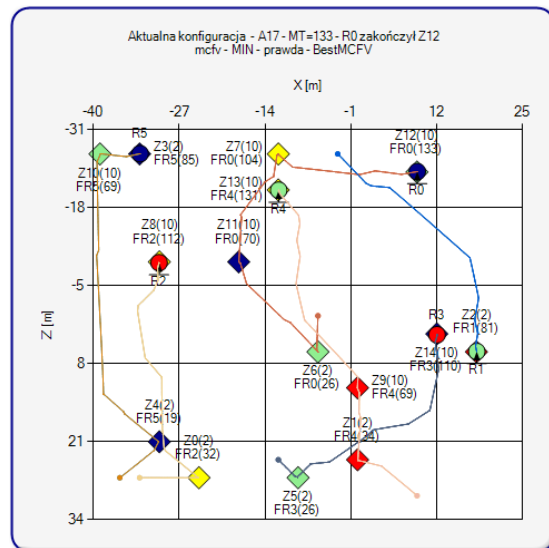
(a) zadania(EE)



(b) roboty(ET)



(c) mcfv - MIN - fałsz - Queue



(d) mcfv - MIN - prawda - BestMCFV

Rys. 4.10: Wykresy tras robotów dla wariantów (Środowisko 2, Konfiguracja 6)

Tab. 4.24: Wyniki przeprowadzonych eksperymentów (Środowisko 2, Konfiguracja 6)

L.p.	Wariant	<i>MP.am</i>	<i>MP.cf</i>	<i>MP.aaaa</i>	<i>MP.mcfv.am</i>	<i>MT</i>	<i>TravT</i>	<i>TravD</i>	<i>TravE</i>	<i>UnnResT</i>
1	1	zadania(ET)				144	450	391,7	1377,4	2289
2	2	zadania(EE)				187	609	547,2	1905,9	2673
3	3	roboty(ET)				138	402	339,2	1289,6	2154
4	4	roboty(EE)				160	472	395,7	1438,5	2345
5	5	mcfv	MIN	falsz	Queue	161	455	363,9	1229,4	2364
6	6	mcfv	MIN	falsz	BestMeanOfMCFVs	160	431	347,5	1111,4	2340
7	7	mcfv	MIN	falsz	BestMCFV	161	426	345,4	1078,9	2321
8	8	mcfv	MIN	prawda	BestMeanOfMCFVs	156	482	406,6	1061,1	2094
9	9	mcfv	MIN	prawda	BestMCFV	133	385	324,4	1127,5	2014
10	10	mcfv	MIN-MAX	falsz	Queue	151	456	371,8	1300,1	2329
11	11	mcfv	MIN-MAX	falsz	BestMeanOfMCFVs	149	440	363,6	1204,6	2293
12	12	mcfv	MIN-MAX	falsz	BestMCFV	171	455	372,4	1193,6	2368
13	13	mcfv	MIN-MAX	prawda	BestMeanOfMCFVs	172	477	400,1	1122,6	2036
14	14	mcfv	MIN-MAX	prawda	BestMCFV	131	384	323,1	1121,3	2000

## 4.7. Podsumowanie badań weryfikacyjnych

Przeprowadzone badania weryfikacyjne opisane w niniejszym rozdziale miały na celu:

1. Weryfikację poprawności działania systemu oraz zaproponowanych przez autora metod alokacji zadań w grupie robotów.
2. Wykazanie słuszności tez sformułowanych na początku pracy.

Za pomocą przeprowadzonych badań wykazano metodą indukcji niezupełnej, że dla wszystkich realizowanych misji metody alokacji zadań stosujące aukcjonowanie robotów dają lepsze wyniki w stosunku do metod alokacji zadań stosujących aukcjonowanie zadań. W ten sam sposób dowiedziono tezy mówiącej, że dla większości realizowanych misji metody alokacji zadań w grupie robotów mobilnych bazujące na wielokryterialnej analizie stanu robotów oraz ich otoczenia umożliwiają wzrost efektywności realizacji misji w porównaniu z metodami alokacji zadań bazującymi na aukcjonowaniu zadań lub aukcjonowaniu robotów.

Ponadto przeprowadzone badania weryfikacyjne systemu sterowania grupą robotów wykazały na drodze symulacji, że możliwa jest jego implementacja na grupie robotów wykonujących zadania inspekcyjne.

Prowadzenie eksperymentów weryfikacyjnych pozwoliło na zebranie wniosków dotyczących opracowanego sterownika behawioralnego:

- Roboty we właściwy sposób dojeżdżały do miejsc wykonywania zadań,
- Zdarzyły się sytuacje, w których w wyniku działania zachowania *omijaj-innego-robot* roboty zamiast się wyminąć, zaczęły przemieszczać się równolegle w tym samym kierunku, aż do chwili, kiedy zbyt oddaliły się od swoich wyznaczonych ścieżek. Wtedy się omijały i wracały na swoje ścieżki ruchu. Wyników alokacji zadań dla takich eksperymentów nie uwzględniano ze względu, że byłyby one wypaczone z powodu niewłaściwego działania sterownika behawioralnego,
- W pracy nie testowano systemu dla misji, dla których system nie miał wiedzy o środowisku (zawsze dostępna była mapa w postaci ograniczonej topografii przestrzeni). Wydaje się, że wyczerpujące i interesujące byłyby testy behawioralnego

podsystemu sterowania ruchem robotów dla misji, w których nie byłaby dostępna wiedza o otoczeniu robotów.

Analiza wyników otrzymanych ramach badań weryfikacyjnych prowadzi do następujących wniosków szczegółowych:

- Korzystniejszą metodą alokacji zadań jest metoda stosująca aukcjonowanie robotów, aniżeli metoda stosująca aukcjonowanie zadań,
- Dla wariantów 1-4 dla eksperymentów zrealizowanych dla Konfiguracji 2-4 kryterium energetyczne oceny ofert było nieznacznie bardziej korzystne niż kryterium czasowe, natomiast dla Konfiguracji 5-6 w większości przypadków bardziej korzystne było kryterium czasowe oceny ofert. Może to wskazywać, że dla tak skonstruowanego systemu w przypadku metod alokacji zadań stosujących aukcjonowanie zadań lub robotów większy wpływ ma konfiguracja robotów i zadań, aniżeli przyjęte kryterium oceny ofert,
- Najprostszym sposobem wykazania wzrostu efektywności działania systemu stosującego zaproponowane metody wielokryterialnej oceny przystosowania robotów do zadań MCFV w stosunku do systemu stosującego znane metody aukcjonowania CNP jest porównanie wyników eksperymentów dla wariantów 3-4. z wariantem 5., ponieważ dla tych wariantów kolejność przydzielania zadań wynika z pozycji robotów niezajętych w kolejce robotów  $R^{nb}$ . Dla wszystkich przeprowadzonych eksperymentów (poza eksperymentem dla Środowiska 1., Konfiguracji 4., warianty 4. i 5.) otrzymano korzystniejsze wyniki dla metody wielokryterialnej,
- Bardzo często (nie zawsze) najlepsze wyniki dawała metoda MCFV stosująca aukcjonowanie wszystkich robotów aktywnych, gdzie przydział zadań następował na podstawie największej wartości  $mcfv^{r \rightarrow z}$ ,
- Porównując wyniki otrzymane za pomocą metod stosujących algorytm MIN lub algorytm MIN-MAX można zauważyć, że dla odpowiadających sobie wariantów testów dla tych algorytmów w większości przypadków lepsze wartości poszczególnych miar oceny działania systemu otrzymywane są za pomocą algorytmu MIN,
- Porównując warianty testów 5-7 z 8-9 oraz 10-12 z 13-14 stwierdza się, że metody MCFV stosujące aukcjonowanie wszystkich robotów aktywnych dają lepsze wyniki w stosunku do metod MCFV stosujących aukcjonowanie tylko robotów niezajętych. Wyjątkiem są wyniki dla Środowiska 1., Konfiguracji 2., które nie potwierdzają tej reguły.

## Rozdział 5

# Podsumowanie

Przedmiotem badań opisanych w rozprawie był system sterowania grupą robotów bazujący na zachowaniach. Tematyka tych badań zainspirowana została kilkoma projektami badawczymi, wykonywanymi w Katedrze Podstaw Konstrukcji Maszyn Politechniki Śląskiej przez zespół, w skład którego wchodził autor. W ramach wzmiankowanych projektów zrealizowano kompletny proces projektowo-konstrukcyjny, a następnie wytworzono i uruchomiono dwa zespoły robotów mobilnych przeznaczone do realizowania zadań inspekcyjnych. Udział w tych pracach umożliwił autorowi identyfikację potrzeby rozwoju systemów sterowania, których celem byłaby alokacja zadań pomiędzy roboty tworzące grupę inspekcyjną. Tematyka rozprawy wypływa więc ściśle z potrzeby rozwiązania problemów sterowania taką grupą.

Głównym osiągnięciem autora było zaproponowanie ogólnego systemu sterowania grupą robotów, bazującego na zachowaniach. Podstawą do działań o charakterze teoretycznym była wnikliwa analiza opisów dotychczas stosowanych rozwiązań w kraju i za granicą. Szczególnym problemem rozważanym w pracy jest przydział zadań inspekcyjnych pomiędzy roboty tworzące grupę. Rozwiązania opisane w dostępnej literaturze bazują na aukcjonowaniu zadań. Autor w zasadniczy sposób rozwinął metody przydziału zadań, wprowadzając nową metodę aukcjonowania robotów, dualną do znanej metody aukcjonowania zadań. Ponadto wprowadzono optymalne metody przydziału zadań, bazujące na wielokryterialnej ocenie robotów i zadań. Stosowane kryteria dotyczą między innymi:

- czasu wykonywania misji przez grupę robotów,
- ilości energii koniecznej do zrealizowania misji (z podziałem na energię zużytą na dojazd do miejsca realizacji zadania, energię zużywaną przez roboty pozostające w stanie bezczynności oraz energię zużywaną na wykonanie zadań wchodzących w skład misji),
- nadmiarowości wyposażenia robotów w układy wykonawcze i pomiarowe.

Opis systemu jest na tyle ogólny, że możliwa jest jego implementacja dla różnych grup robotów, zadań i środowisk.

Znaczna część badań dotyczy weryfikacji systemu sterowania grupą robotów oraz metod alokacji zadań pomiędzy członków grupy. Weryfikację postanowiono przeprowadzić dla wirtualnej grupy robotów na drodze symulacji. Jako środowisko programowe wy-

brano Microsoft Robotics Developer Studio. Do weryfikacji zastosowano środowisko symulacyjne opracowane w zespole Katedry Podstaw Konstrukcji Maszyn. Autor opracował odpowiednie oprogramowanie stanowiące implementację metod aukcjonowania zadań. Przygotowane środowisko umożliwia wirtualne symulacje realizacji misji o znacznym stopniu złożoności. Oceny wartości kryteriów są wyliczane podczas poszczególnych symulacji oraz składowane w bazie danych w celu ich późniejszej szczegółowej analizy.

Właściwe badania weryfikacyjne prototypu systemu i metod rozdziału zadań zostały przeprowadzone dla grupy wirtualnych robotów, których cechy zostały określone na podstawie realnie istniejących robotów będących wynikiem projektów badawczych i rozwojowych zrealizowanych w Katedrze Podstaw Konstrukcji Maszyn. Ze względu na fakt, że postanowiono wykazać słuszność wysuniętych tez na drodze indukcji niezupełnej, opracowano plan badań weryfikacyjnych, obejmujący wykorzystanie dwóch konfiguracji środowisk testowych i konfiguracji misji. Stopień złożoności konfiguracji i misji testowych odpowiada potrzebom inspekcji rzeczywistych obiektów. Następnie przeprowadzono zaplanowane symulacje oraz dokonano szczegółowej analizy i oceny uzyskanych wyników.

Zdaniem autora, do najważniejszych, oryginalnych wyników pracy można zaliczyć:

1. Formalny opis systemu sterowania grupa robotów (Rozdz. 3),
2. Zbiór metod aukcjonowania zadań i robotów z wykorzystaniem wielokryterialnej oceny przystosowania robotów do zadań, a w szczególności nowatorskie zastosowanie miar pozycji konkurencyjnej robotów dla zadań i atrakcyjności zadań dla robotów do sterowania dynamicznym przydziałem zadań poszczególnym robotom podczas trwania misji (Rozdz. 3.4),
3. Zaplanowanie i przeprowadzenie szeroko zakrojonych badań symulacyjnych dla wielu wariantów wartości parametrów działania systemu sterowania (Rozdz. 4),
4. Implementację systemu sterowania w środowisku symulacyjnym,
5. Opracowanie odpowiedniego oprogramowania umożliwiającego automatyzację eksperymentów numerycznych.

## 5.1. Wnioski

Uzyskane wyniki weryfikacji systemu sterowania grupą robotów inspekcyjnych pozwalają na sformułowanie metodą indukcji niezupełnej następujących wniosków:

1. Metody aukcjonowania robotów pozwalają na lepszą alokację zadań pomiędzy roboty, niż metody aukcjonowania zadań.
2. Metody alokacji zadań z wykorzystaniem optymalizacji wielokryterialnej pozwalają na lepszy przydział zadań pomiędzy roboty wchodzące w skład grupy.
3. Metody MCFV stosujące algorytm MIN do wyliczania wartości kryteriów oceny pozycji konkurencyjnej robota i atrakcyjności zadania dają w większości przypadków lepsze wyniki (miary oceny działania systemu) w stosunku do metod MCFV stosujących algorytm MIN-MAX.

4. Metody MCFV stosujące aukcjonowanie wszystkich robotów aktywnych (zajętych i niezajętych) dają w przeważającej liczbie przypadków najlepsze wyniki ze wszystkich stosowanych i opracowanych metod.
5. Opracowany sterownik behawioralny jest rozwiązaniem wystarczającym do sterowania pojedynczym robotem mobilnym poruszającym się w dynamicznie zmiennym środowisku działania tego robota.

## 5.2. Kierunki dalszych badań

Po przeprowadzeniu badań opisanych w dysertacji autor zidentyfikował następujące kierunki badań, które chciałby podjąć w przyszłości:

1. Implementacja systemu sterowania oraz metod alokacji zadań na grupie realnie istniejących robotów inspekcyjnych.
2. W przypadku eksperymentu prowadzonego na rzeczywistych robotach i w rzeczywistym środowisku konieczne może być wykrywanie ogólnie rozumianych uszkodzeń robotów. W tym celu można stosować diagnostykę wspartą modelem obiektu (Kościelny, 2001; Korbicz *i in.*, 2002), gdzie modelowane może być np. poprawne działanie układu napędowego robota, lub układu zasilania (np. akumulatory Li-Po),
3. Realizacja eksperymentów wirtualnych z losowym generowaniem konfiguracji danej misji oraz połączeń robotów i zadań.
4. Wprowadzenie wag kryteriów oceny mających wpływ na pozycję konkurencyjną robota i atrakcyjność zadania. W pracy wszystkie kryteria miały tę samą wagę, niezależnie od aktualnej konfiguracji robotów i zadań. Przykładowo można by wprowadzić do modelu zadań parametry określające ważność, czy też pilność zadania.
5. Identyfikacja oraz wprowadzenie innych kryteriów mających wpływ na pozycje konkurencyjne robotów i atrakcyjność zadań.
6. Wprowadzenie innego sposobu obliczania wielokryterialnej oceny przystosowania robotów do zadań, np. wartość  $mcfv^{r \rightarrow z}$  mogłaby być odległością euklidesową pomiędzy:
  - punktem wynikającym z przecięcia prostej prostopadłej do prostej przechodzącej przez początek prostokątnego układu współrzędnych, którego osie reprezentują pozycje konkurencyjne robotów i atrakcyjności zadania,
  - a punktem należącym do tego układu o współrzędnych  $(rcp^{r \rightarrow z}, za^{z \rightarrow r})$ .

Sterowanie kątem nachylenia prostej przechodzącej przez początek układu współrzędnych  $(rcp^{r \rightarrow z}, za^{z \rightarrow r})$  pozwoliłoby na różnicowanie wpływu pozycji konkurencyjnej robota dla zadania i atrakcyjności zadania dla robota na wartość oceny  $mcfv^{r \rightarrow z}$ .

7. Rozszerzenie modelu systemu o elementy pozwalające na tworzenie koalicji przez roboty, co przydałoby się w trakcie realizacji zadań, których nie może podjąć się żaden robot z osobna ze względu na niewystarczające zasoby inspekcyjne, ale mogłaby się podjąć grupa robotów, których zbiór wszystkich zasobów inspekcyjnych byłby wystarczający dla realizacji zadania,
8. Rozbudowanie sterownika behawioralnego o inne zachowania, np. pozwalające robotowi na unikanie miejsc, w których dopiero co przebywał (*avoid-past*, (Balch i Arkin, 1993)) lub zachowań pozwalających na generowanie gładkich ścieżek ruchu robota w obecności przeszkód (Borenstein i Koren, 1991; Ferreira *i in.*, 2008).



# Bibliografia

- Arbib M. A. (1981): Perceptual structures and distributed motor control. W: Handbook of Physiology - The Nervous System II, V. B. Brooks (Red.). American Physiological Society: pp. 1449–1465.
- Arbib M. A. i House D. H. (1987): Depth and detours: an essay on visually guided behavior. W: Vision, brain, and cooperative computation, M. A. Arbib i A. R. Hanson (Red.). MIT Press. Cambridge, MA, USA: pp. 129–163.
- Arkin R. (1987): *Motor schema based navigation for a mobile robot: An approach to programming by behavior*. Proceedings of the IEEE International Conference on Robotics and Automation. Tom 4: pp. 264–271.
- Arkin R. C. (1989): *Motor schema – based mobile robot navigation*. The International Journal of Robotics Research 8(4): pp. 92–112.
- Arkin R. C. (1998): *Behavior-based robotics*. MIT Press. Cambridge, Mass.
- Arkin R. C. i Balch T. (1997): *AuRA: Principles and practice in review*. Journal of Experimental and Theoretical Artificial Intelligence 9: pp. 175–189.
- Balch T. i Arkin R. (1993): *Avoiding the past: A simple but effective strategy for reactive navigation*. IEEE International Conference on Robotics and Automation (ICRA '93): pp. 678–685.
- Balch T. i Arkin R. C. (1998): *Behavior-based formation control for multi-robot teams*. IEEE Transactions on Robotics and Automation 14(6): pp. 926–939.
- Batalin M. A. i Sukhatme G. S. (2002): *Spreading out: A local approach to multi-robot coverage*. Proc. of 6th International Symposium on Distributed Autonomous Robotic Systems: pp. 373–382.
- Bigaj P., Trojnacki M. T. i Bartoszek J. (2010): Robot IBIS - realizacja ruchu w trybie teleoperacji i semiautonomii. W: Problemy Robotyki, K. Tchoń i C. Zieliński (Red.). Tom 175 z : *Prace Naukowe Elektronika*. Oficyna Wydawnicza Politechniki Warszawskiej: pp. 135–148.
- Bond A. i Gasser L. (1988): *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers.
- Borenstein J. i Koren Y. (1991): *The vector field histogram - fast obstacle avoidance for mobile robots*. IEEE Journal of Robotics and Automation 7: pp. 278–288.
- Brooks R. A. (1986): *A robust layered control system for a mobile robot*. IEEE Journal of Robotics and Automation 2(1): pp. 14–23.

- Brooks R. A. (1991): *New approaches to robotics*. Science 253: pp. 1227–1232.
- Burkhard H.-D., Duhaut D., Fujita M., Lima P., Murphy R. i Rojas R. (2002): *The road to robocup 2050*. IEEE Robotics and Automation Magazine : pp31–38.
- Cao Y. U., Fukunaga A. S. i Kahng A. B. (1997): *Cooperative mobile robotics: antecedents and directions*. Autonomous Robots 4(1): pp. 7–27.
- Chojecki R., Olszewski M. i Marcinkiewicz P. (2004): Miniaturowe roboty mobilne instytutu automatyki i robotyki politechniki warszawskiej. W: Postępy Robotyki. Przemysłowe i medyczne systemy robotyczne, K. Tchoń i C. Zieliński (Red.). Oficyna Wydawnicza Politechniki Warszawskiej.
- Czupryniak R. i Trojnacki M. (2010): Taktyczny robot miotany. W: Problemy Robotyki, K. Tchoń i C. Zieliński (Red.). Tom 175 z : *Prace Naukowe Elektronika*. Oficyna Wydawnicza Politechniki Warszawskiej: pp. 149–160.
- Dahl T. S., Mataric M. J. i Sukhatme G. S. (2009): *Multi-robot task allocation through vacancy chain scheduling*. Robotics and Autonomous Systems 57(6-7): pp. 674–687.
- desJardins M. E., Durfee E. H., Ortiz C. L. i Wolverton M. J. (1999): *A survey of research in distributed, continual planning*. AI Magazine 20: pp. 13–22.
- Dias M. B. (2004): *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. Praca doktorska. Robotics Institute, Carnegie Mellon University.
- Dudek G., Jenkin M. i Wilkes D. (1996): *A taxonomy for multi-agent robotics*. Autonomous Robots 3: pp. 375–397.
- Ferreira A., Pereira F. G., Vassallo R. F., Bastos Filho T. F. i Filho S. (2008): *An approach to avoid obstacles in mobile robot navigation: the tangential escape*. Controle Sociedade Brasileira de Automatica 19: pp. 395 – 405.
- Fierro R., Chaimowicz L. i Kumar V. (2006): Multi-robot cooperation. W: Autonomous mobile robots: sensing, control, decision making and applications, S. S. Ge i F. L. Lewis (Red.). CRC Press: pp. 417–459.
- Gasser L. (2001): *Multi-Agent Systems and Applications*. Tom Volume 2086/2006 z *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. Rozdz. Perspectives on Organizations in Multi-agent Systems, : pp1–16.
- Gerkey B. P. (2003): *On multi-robot task allocation*. Praca doktorska. University of Southern California, Computer Science Department. Los Angeles, CA, USA. Adviser-Mataric, Maja J.
- Gerkey B. P. i Mataric M. J. (2004): *Are (explicit) multi-robot coordination and multi-agent coordination really so different*. Proceedings of the AAI Spring Symposium on Bridging the Multi-Agent and Multi-Robotic Research Gap: pp. 1–3.
- Gerkey B. P. i Mataric M. J. (2002): *Sold!: Auction methods for multi-robot coordination*. IEEE Transactions on Robotics and Automation 18(5): pp. 758–768.

- Gerkey B. P. i Mataric M. J. (2003a): *A framework for studying multi-robot task allocation*. W: *Multi-Robot Systems: From Swarms to Intelligent Automata*, Volume II, A. Schultz i in. (Red.). Kluwer Academic Publishers. Kluwer Academic Publishers. the Netherlands: pp. 15–26.
- Gerkey B. P. i Mataric M. J. (2003b): *Multi-robot task allocation: Analyzing the complexity and optimality of key architectures*. Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA '03). Taipei, Taiwan: pp. 3862–3868.
- Giergiel J., Kurc K. i Giergiel M. (2010): *Mechatroniczne projektowanie robotów inspekcyjnych*. Oficyna Wydawnicza Politechniki Rzeszowskiej.
- Gnatowski M. (2005): *Wykorzystanie systemów wieloagentowych we współdziałaniu robotów mobilnych*. Praca doktorska. Polska Akademia Nauk - IPPT.
- Horling B. i Lesser V. (2005): *A survey of multi-agent organizational paradigms*. The Knowledge Engineering Review 19(4): pp. 281–316.
- Januszka M. (2008): *Mechanical carrier of an autonomous mobile robot for inspecting technical objects*. Problemy Eksploatacji 3: pp. 31–38.
- Januszka M., Adamczyk M. i Moczulski W. (2008): Nieholonomiczny autonomiczny robot mobilny do inspekcji obiektów technicznych. W: *Problemy Robotyki*, Tom 1, K. Tchoń i C. Zieliński (Red.). Tom 166 z *Prace Naukowe Politechniki Warszawskiej. Elektronika*. Oficyna Wydawnicza Politechniki Warszawskiej. Warszawa: pp. 143–152.
- Johns K. i Taylor T. (2008): *Professional Microsoft Robotics Developer Studio*. Wrox.
- Jones C. i Mataric M. (2006): Behavior-based coordination in multi-robot teams. W: *Autonomous mobile robots: sensing, control, decision making and applications*, S. S. Ge i F. L. Lewis (Red.). CRC Press: pp. 549–569.
- Kalra N. (2007): *A Market-Based Framework for Tightly-Coupled Planned Coordination in Multirobot Teams*. Praca doktorska. Carnegie Mellon University.
- Kalra N., Ferguson D. i Stentz A. (2005): *Hoplites: A market-based framework for planned tight coordination in multirobot teams*. IEEE International Conference on Robotics and Automation (ICRA), 2005.: pp. 1170–1177.
- Kasprzyk J. (2006): *Programowanie sterowników przemysłowych*. WNT.
- Kościelny J. M. (2001): *Diagnostyka zautomatyzowanych procesów przemysłowych*. EXIT.
- Korbicz J., Kościelny J. M., Kowalczyk Z. i Cholewa W. (Red.) (2002): *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania*. PWN.
- Koren Y. i Borenstein J. (1991): *Potential field methods and their inherent limitations for mobile robot navigation*. In Proc. IEEE Int. Conf. Robotics and Automation: pp. 1398–1404.
- Laue T. i Röfer T. (2004): *A behavior architecture for autonomous mobile robots based on potential fields*. 8th International Workshop on RoboCup 2004 , Lecture Notes in Artificial Intelligence (LNAI). Springer: pp. 122–133.

- Lisiński M. (2004): *Metody planowania strategicznego*. Polskie Wydawnictwo Ekonomiczne.
- MacKenzie D. C., Arkin R. i Cameron J. M. (1997): *Multiagent mission specification and execution*. *Autonomous Robots* 4(1): pp. 29–52.
- Matarić M. J. (2007): *The Robotics Primer*. MIT Press.
- Moczulski W., Adamczyk M., Januszka M., Panfil W., Przystałka P. i Wyleźoł M. (2008): *Multi-robot group for inspecting large area objects*. *Problemy Eksploatacji* 3: pp. 19–29.
- Moczulski W., Adamczyk M., Januszka M., Panfil W., Przystałka P. i Wyleźoł M. (2009): *Team of specialized mobile robots for group inspection of large-area technical objects*. W: *Robot Motion and Control 2009*, K. Kozłowski (Red.). Tom 396 z *LNCIS*. Springer: pp. 411–420.
- Moczulski W., Adamczyk M., Przystałka P. i Timofiejczuk A. (2007): *Recent Advances in Mechatronics*. Springer Berlin Heidelberg. Rozdz. *Mobile robot for inspecting ventilation ducts*.
- Moczulski W., Skarka W., Adamczyk M., Januszka M., Giesko T., Mężyk J., Mizak W., Pająk D., Panfil W., Przystałka P., Targosz M., Wiglenda R. i Wyleźoł M. (2010): *Projekt grupy wielozadaniowych robotów mobilnych wykorzystujących zaawansowane technologie*. Artykuł przyjęty na XI Kongres Budowy i Eksploatacji Urządzeń Technicznych, Radom 2011.
- Moczulski W., Skarka W., Adamczyk M., Januszka M., Pająk D., Panfil W., Przystałka P., Targosz M., Wiglenda R. i Wyleźoł M. (2011): *Rozwiązania konstrukcyjne wielozadaniowych robotów mobilnych wykorzystujących zaawansowane technologie*. Artykuł przyjęty na XI Kongres Budowy i Eksploatacji Urządzeń Technicznych, Radom 2011.
- Nuzhet A. i Burchan B. (2006): *Mixed-integer linear programming solution to multi-robot task allocation*. *Technical report*. Department of Computer Science and Engineering, Washington University.
- Panfil W. i Moczulski W. (2010a): *Incorporating deliberative and reactive components into the control system of the inspection robots group*. *Mechatronics 2010*. Zurich.
- Panfil W. i Moczulski W. (2010b): *System sterowania grupą robotów inspekcyjnych – opis badań wstępnych*. W: *Problemy Robotyki*, K. Tchoń i C. Zieliński (Red.). Tom 175 z : *Prace Naukowe Elektronika*. Oficyna Wydawnicza Politechniki Warszawskiej: pp. 161–170.
- Panfil W., Przystałka P. i Adamczyk M. (2007): *Recent Advances in Mechatronics*. Springer Berlin Heidelberg. Rozdz. *Behavior-based control system of a mobile robot for the visual inspection of ventilation ducts*, : pp62–66.
- Parker L. E. (1994): *Heterogeneous multi-robot cooperation*. Praca doktorska. MIT, Cambridge. Cambridge, MA, USA.

- Parker L. E. (1998): *Alliance: An architecture for fault tolerant multi-robot cooperation*. IEEE Transactions on Robotics and Automation 14(2): pp. 220–240.
- Rosenblatt J. (1997): *Damn: A distributed architecture for mobile navigation*. Journal of Experimental and Theoretical Artificial Intelligence 9(1): pp. 339 – 360.
- Sariel S. (2007): *An Integrated Planning, Scheduling and Execution Framework for Multi-Robot Cooperation and Coordination*. Praca doktorska. Istanbul Technical University, Institute of Science and Technology.
- Sariel S. i Balch T. (2005): *Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments*. Integrating Planning into Scheduling: Papers from the 2005 AAAI Workshop: pp. 27–33.
- Scerri P., Modi P. J., Shen W.-M. i Tambe M. (2003): *Are multiagent algorithms relevant for real hardware? A case study of distributed constraint algorithms*. SAC. Tom 110: pp. 38–44.
- Smith R. G. (1980): *The contract net protocol: High-level communication and control in a distributed problem solver*. IEEE Transactions on Computers 29(12): pp. 1104–1113.
- Stabryła A. (2000): *Zarządzanie strategiczne w teorii i praktyce firmy*. Wydawnictwo Naukowe PWN.
- Turek W. (2009): *Agentowy system wielomodelowy do zarządzania grupą robotów mobilnych*. Praca doktorska. Akademia Górniczo - Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki.
- Vail D. i Veloso M. M. (2003): *Multi-robot dynamic role assignment and coordination through shared potential fields*. W: Multi-Robot Systems, A. Schultz, L. Parker i F. Schneider (Red.). Kluwer.
- Werger B. B. i Mataric M. J. (2000): *Broadcast of local eligibility for multi-target observation*. W: Distributed Autonomous Robotic Systems, L. E. Parker, G. Bekey, i J. Barhen (Red.). Tom 4. Springer-Verlag: pp. 347–356.
- Werger B. B. i Mataric M. J. (2001): *From insect to internet: Situated control for networked robot teams*. Annals of Mathematics and Artificial Intelligence 31(1-4): pp. 173–197.
- Wiglenda R. (2009): *Oprogramowanie dla generowania optymalnej ścieżki robotów mobilnych współpracujące ze środowiskiem microsoft robotics developer studio*. Technical report. Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska.
- Wiglenda R. (2010): *Software for modelling and monitoring mission of a group of robots*. Pr. magisterska. Politechnika Śląska.
- Zaborowski W. (2008): *Zastosowanie systemów agentowych do wspomagania zarządzania zespołami robotów mobilnych*. Praca doktorska. Akademia Górniczo - Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki.

- Zieliński C., Szykiewicz W., Winiarski T. i Staniak M. (2006): *Rubik's cube puzzle as a benchmark for service robots*. 12th IEEE International Conference on Methods and Models in Automation and Robotics, MMAR 2006: pp. 579–584.
- Zlot R. M., Stentz A. T., Dias M. B. i Thayer S. (2002): *Multi-robot exploration controlled by a market economy*. W: IEEE International Conference on Robotics and Automation, IEEE (Red.). Tom 3: pp. 3016 – 3023.

# System sterowania grupą inspekcyjnych robotów mobilnych bazujący na zachowaniach

Praca doktorska - Streszczenie

Autor: mgr inż. Wawrzyniec Panfil

Promotor: prof. dr hab. Wojciech A. Moczulski

Politechnika Śląska w Gliwicach, Wydział Mechaniczny Technologiczny

Celem niniejszej pracy było opracowanie systemu sterowania grupą robotów mobilnych wykonujących zadania inspekcyjne. Podstawowym problemem naukowym podjętym w pracy było określenie systemu sterowania grupą robotów w taki sposób, aby możliwa była współpraca tych robotów celem realizacji postawionych zadań.

Na podstawie przeglądu istniejących rozwiązań w zakresie budowy systemów sterowania grup robotów mobilnych stwierdzono, że na efektywne działanie tych systemów znaczący wpływ ma właściwa alokacja zadań wśród członków tej grupy. Zidentyfikowane zostały dwa istotne problemy, których rozwiązanie pozwoliłoby na efektywniejsze wykonywanie zadań (misji) przez grupy robotów. Wiele spośród znanych systemów sterowania jako podstawę przydzielania zadań stosuje metody aukcyjne, w których przedmiotem aukcji są zadania. Wydaje się, że dla misji o znacznej nadmiarowości zadań korzystniejszym będzie prowadzenie aukcji, dla których przedmiotem ofert będą roboty. Drugim problemem, który zidentyfikowano w istniejących rozwiązaniach, jest jednostronna ocena przystosowania robota do zadania. Stwierdzono, że jednoczesna ocena przystosowania robota do zadania oraz ocena atrakcyjności zadania dla robota wpłynie na wzrost efektywności działania grupy robotów wykonujących zadania inspekcyjne.

W celu wykazania słuszności sformułowanych tez zaproponowano metodykę budowy systemów sterowania grup robotów oraz w oparciu o nią opracowano opis formalny systemu, za pomocą którego możliwe jest przydzielanie zadań w grupie robotów stosujących różne metody wyznaczania oceny przystosowania robotów do zadań oraz przydziału zadań. Zaproponowana została wielokryterialna metoda oceny przystosowania robotów do zadań, której składowymi są pozycja konkurencyjna robota dla zadania wśród innych robotów oraz atrakcyjność zadania dla robota wśród innych zadań. Ponadto zaproponowane zostały metody przydzielania zadań stosujące wielokryterialną metodę oceny przystosowania robotów do zadań. Opracowano również behawioralny sterownik służący sterowaniu ruchem robotów mobilnych.

Weryfikację systemu, a także zaproponowanych metod alokacji zadań, przeprowadzono na drodze eksperymentu symulacyjnego, w którym obiektem badań była grupa wirtualnych robotów będących odpowiednikami robotów inspekcyjnych będących na wyposażeniu Katedry Podstaw Konstrukcji Maszyn Politechniki Śląskiej. Analiza zebranych wyników badań potwierdziła słuszność postawionych w pracy tez.

**Słowa kluczowe:** systemy wielorobotowe, aukcjonowanie, przydzielanie zadań, sterowanie behawioralne.

# Behavior-based control system of the inspection robots group

PhD thesis - Summary

Author: Wawrzyniec Panfil, MSc, Eng.

Supervisor: Prof. Wojciech A. Moczulski, PhD, DSc

Silesian University of Technology at Gliwice, Faculty of Mechanical Engineering

The goal of this PhD thesis was to elaborate a control system of a mobile robots group intended for carrying out inspection missions. The main research problem was to define such a control system in order to facilitate a cooperation of the robots resulting in realization of the committed inspection tasks.

Taking into account the current state-of-the-art in the area of the multi-robot control systems it was stated that a significant influence on the effective operation of these systems has an appropriate allocation of tasks among members of the multi-robot group. There were identified two fundamental problems – resolving them will allow to realize the tasks (missions) by the robots more effectively. Many of the well-known control systems use auctions for tasks allocation, where a subject of an auction is a task to be allocated. It seems that in the case of missions characterized by much larger number of tasks than number of robots it will be better if robots (instead of tasks) are subjects of auctions. The second identified problem concerns the one-sided robot-to-task fitness evaluation. It was stated that simultaneous assessment of the robot-to-task fitness and task attractiveness for robot will affect positively for the overall effectiveness of the multi-robot system performance.

To prove the correctness of the mentioned hypotheses there was proposed the methodology of building the control systems of multi-robot groups and basing on that there was elaborated the formal description of the system. This system allows to assign tasks to robots using various methods for evaluation of fitness between robots and tasks, and using some tasks allocation methods. There was proposed the method for multi-criteria analysis, which is composed of two assessments, i.e. robot's concurrency position for task among other robots and task's attractiveness for robot among other tasks. Furthermore, there were proposed methods for tasks allocation applying the mentioned multi-criteria analysis method. There was also elaborated a behavior-based controller for robots' motion control.

The verification of both the elaborated system and the proposed tasks' allocation methods was carried out with the help of simulated experiments. The object under test was a group of inspection mobile robots being a virtual counterpart of the real mobile-robot group owned by the Department of Fundamentals of Machinery Design, Silesian University of Technology. The analysis of the achieved verification results proved the correctness of the formulated theses.

**Keywords:** multi-robot systems, auctions, tasks allocation, behavior-based control.