

**Universitatea Tehnică „Gheorghe Asachi” din Iași
Universitatea din Zaragoza, Spania**



OBIECTIVE DE LOGICĂ TEMPORALĂ ÎN PLANIFICAREA DE NIVEL ÎNALT A TRAIECTORIILOR: CADRU CONCEPTUAL STRUCTURAT PE SISTEME CU EVENIMENTE DISCRETE

Sofia HUȘTIU

Conducător de doctorat:

Prof. Dr. Ing. Octavian-Cezar PĂSTRĂVANU (Universitatea Tehnică „Gheorghe Asachi” din Iași)

Prof. Dr. Ing. Cristian Florentin MAHULEA (Universitatea din Zaragoza, Spania)

IAȘI, 2025

Cuprins

Listă de figuri	v
Listă de tabele	vii
1 Introducere	1
1.1 Context	1
1.2 Descrierea problemei și contribuții	2
1.3 Structura tezei	2
2 Sisteme cu evenimente discrete pentru soluții de planificare a traseelor în sisteme multi-agent	5
2.1 Ipotezele problemei	5
2.1.1 Spațiul de lucru al sistemului multi-robot	5
2.2 Discretizare în celule	6
2.3 Reprezentări ale agenților prin evenimente discrete	6
2.3.1 Rețea Petri	6
2.3.2 Rețea Petri Temporizată	7
2.4 Misiunea roboților	7
2.4.1 Logica Temporală Liniară	8
2.4.2 Logica Temporală cu Interval Metric	8
2.5 Criterii de comparație pentru strategiile de planificare ale sistemele multi-agent	8
3 Metodă de decompoziție a acțiunilor pentru sistemele multi-agent	9
3.1 Decompoziția acțiunilor	9
3.2 Soluția pentru descompunerea în acțiuni	10
3.3 Evaluarea numerică	11
4 Planificarea traiectoriilor cu specificații LTL și optimizarea traiectoriilor pentru sisteme multi-robot	13
4.1 Conceptul unui strat intermediar	14
4.2 Modelul Rețelei Petri Compuse	14

4.2.1	Soluția bazată pe probleme de optimizare	14
4.2.2	Exemplu numeric	15
4.3	Reconfigurarea traiectoriei, considerând execuția paralelă a mișcărilor. . .	15
4.3.1	Factori determinanți	15
4.3.2	Algoritmul pentru mișcarea paralelă a roboților	15
4.3.3	Rezultate numerice	16
5	Planificarea traiectoriilor cu specificație MITL pentru sisteme multi-robot	17
5.1	Flux de lucru pentru modelare	17
5.2	Mecanism de coordonare pentru multiple modele de tip Rețea Petri Temporizată Compusă	18
5.3	Model-checking approach in numerical evaluation	19
6	Planificarea traiectoriei cu specificație LTL bazată pe o abordare ierarhică pentru un sistem multi-robot.	21
6.1	Paradigma Nets-within-Nets	21
6.2	Formularea problemei	22
6.3	Nets-within-Nets adaptate pentru planificarea traiectoriei	23
6.3.1	Rețea de tip obiect	23
6.3.2	Rețea Petri de Nivel Înalt pentru o echipă robotică	24
6.3.3	Funcția de sincronizare	25
6.4	Evaluare numerică	25
7	Rutine și aplicații software dezvoltate	27
7.1	Implementare	27
7.1.1	Implementarea în MATLAB	27
7.1.2	Implementarea în Renew	28
7.2	Rezultatele simulărilor	28
7.2.1	Vopsirea acoperișului serelor de către o echipă de drone	29
7.2.2	Asistarea sistemelor robotice multi-agent în domeniul sănătății . . .	30
7.3	Validare experimentală a modelului Rețea Petri Temporizată Compusă . . .	30
8	Observații finale	33
8.1	Contribuții	33
8.2	Direcții viitoare de cercetare	34
	Bibliografie	35

Listă de figuri

2.1	Exemplu de celule etichetate ca regiuni de interes	6
3.1	Independent trajectories of the two agents	10
4.1	Diagrama algoritmului global	13
4.2	Parte a modelului Petri net compus, luând în considerare observația activă, respectiv inactivă b_3 , asociată cu regiunea y_3	14
4.3	Scenariul pentru rerutarea traiectoriilor roboților într-un mediu cu 4 roboți și 8 regiuni de interes (ROI).	16
5.1	General Framework of <i>Composed Time Petri net</i> model	18
5.2	Un exemplu de conversie a unui automat temporizat într-o rețea Petri Temporalizată (stânga) pentru formula MITL $\varphi = \Diamond_{\leq \tau} b_1$, care încorporează topologii de ceas (dreapta).	18
5.3	Vedere parțială din <i>Rețeaua Petri Temporalizată Compusă</i> , pentru propoziția atomică b_1 inclusă în specificația MITL $\varphi = \Diamond_{\leq \tau} b_1$	19
5.4	Mecanism de sincronizare pentru modele individuale de tip Rețea Petri Temporalizată Compusă	20
6.1	Exemplu demonstrând paradigma Nets-within-Nets: (i) Rețeaua Petri pentru specificație, (ii) Rețeaua de sistem, (iii) Rețeaua Petri pentru roboți	23
6.2	(a) RobotOPN care modelează trei roboți. Doi dintre roboți, r_1 și r_2 , se pot mișca liber în mediu, în timp ce robotul r_3 nu poate intra în zona suprapusă între y_2 și y_3 (modelul ar fi același, dar fără elementele roșii); (b) SpecOPN: calea marcată corespunde celei mai scurte soluții din 100 de simulări, conform lungimii traiectoriei echipei de roboți.	25
7.1	Diagrama implementării din MATLAB în RMTTool	28
7.2	Decompunerea în celule pentru acoperișul serii, cu precizia $\varepsilon = 4$	29
7.3	Exemplu de scenariu din spital cu trei etaje și 12 camere pentru un sistem multi-robot.	30

7.4	Configurația spațiului de lucru experimental: diagramă din vedere de sus (partea stângă) și vedere laterală reală (partea dreaptă)	31
-----	---	----

Listă de tabele

3.1	Valorile medii pentru costurile maxime, respectiv costurile totale.	11
4.1	Numerical data comparison between the sequential approach [1] and the parallel (current) approach	16
5.1	Evaluare numerică pentru modelul de tip Rețea Petri Temporizată Compusă	20
7.1	Evaluarea rezultatelor simulărilor	29
7.2	Rezultatele comparației pentru echipa eterogenă de roboți în abordarea propusă	30

Capitolul 1

Introducere

Planificarea traiectoriilor unor echipe de roboți mobili a evoluat de la problema clasică de navigație, unde un singur robot trebuie să ajungă la o destinație dorită evitând obstacole, către scenarii complexe utilizând o echipă cu echipe de roboți. În aceste sisteme multi-robot, traseele sunt proiectate pentru a îndeplini o misiune comună, incluzând sincronizarea și secvențierea acțiunilor, cum ar fi vizitarea sau evitarea unor regiuni de interes. În medii cunoscute sau necunoscute, roboții trebuie să planifice trasee optime, adaptându-se la obstacole dinamice, precum alți roboți, esențial în aplicații industriale și explorare autonomă.

1.1 Context

Planificarea traiectoriilor și alocarea acțiunilor în sistemele multi-robotice sunt esențiale pentru optimizarea performanței și succesul misiunilor complexe, precum operațiunile de salvare, transportul autonom și producția industrială. Aceste sisteme utilizează metode avansate de planificare atât în spații 2D, cât și 3D, implicând roboți mobili și drone pentru o coordonare eficientă și reducerea intervenției umane.

Pentru exprimarea și îndeplinirea eficientă a cerințelor misiunilor, sunt utilizate formalizări precum Logica Temporală Liniară (LTL) și Logica Temporală cu Interval Metric (MITL). Aceste expresii de nivel înalt permit definirea acțiunilor complexe, incluzând restricții spațiale și temporale, fiind aplicate în strategii de planificare centralizate sau distribuite. Reprezentările bazate pe Sisteme cu Evenimente Discrete (SED), cum ar fi rețelele Petri, sunt folosite pentru modelarea mișcărilor roboților și gestionarea traiectoriilor în medii statice sau dinamice.

Această teză propune o strategie de planificare combinată care combină modul echipei de roboți cu modelul misiunii într-un singură reprezentare (cadru), reducând timpul de calcul și îmbunătățind eficiența pentru echipe eterogene de roboți. Prin integrarea rețelelor Petri și a automatelor Büchi asociate cu misiunea, soluția permite o execuție optimizată a acțiunilor

și abordarea provocărilor legate de creșterea complexității, coordonarea echipelor și evitarea coliziunilor în medii dinamice.

1.2 Descrierea problemei și contribuții

Scopul acestei teze poate fi rezumat astfel:

Dat fiind un sistem multi-agent care trebuie să asigure o misiune complexă sub formalismul logicii temporale, se propun soluții bazate pe reprezentări ale Sistemelor Discrete de Evenimente pentru alocarea acțiunilor și generarea traiectoriilor fără coliziuni.

Contribuțiile tezei vizează provocările din domeniul roboticii, concentrându-se pe metode avansate de planificare la nivel înalt. Propunerile aduc îmbunătățiri semnificative în autonomie, scalabilitate și flexibilitate prin:

- Descompunerea sarcinilor, permițând împărțirea unei misiuni complexe în acțiuni independente, reducând complexitatea și sporind modularitatea. În literatură nu există o metodă automată de realizare a acestei decompozări, iar soluția propusă adresează acest aspect.
- Un model inovator bazat pe rețele Petri, care combină mișcarea echipei robotice cu cerințele misiunii. Spre deosebire de abordările secvențiale existente, acest model permite accesul la starea globală a roboților. Soluția planificării este returnată de probleme de optimizare și verificării formale.
- Un cadru bazat pe paradigma Nets-within-Nets, care gestionează planificarea mișcării pentru echipe de roboți eterogeni. Această metodă folosește o structură ierarhică a rețelelor Petri pentru a sincroniza datele locale și globale, evitând problema exploziei stărilor întâlnită în abordările discrete.

Toate aceste contribuții urmăresc optimizarea alocării sarcinilor și planificării mișcărilor pentru sisteme multi-robotice, inclusiv în scenarii unde roboții au capacități spațiale diferite.

1.3 Structura tezei

Structura acestei lucrări este organizată în mai multe capitole, fiecare adresând componente cheie ale cercetării. Exemple ilustrative însoțesc formalismul teoretic, abordând problema planificării mișcării, așa cum este definită în enunțul problemei. Soluțiile propuse sunt prezentate și în articole științifice, după cum urmează. Așadar, lucrarea detaliază următoarea structură:

Capitolul 2 introduce conceptele fundamentale utilizate, inclusiv ipotezele problemei și notările robotului, mediului și ideile matematice esențiale. Sunt definite reprezentări

ale evenimentelor discrete, modelând mișcarea echipei de roboți, precum: *Rețele Petri* (PN) și *Rețele Petri Temporale* (TPN). De asemenea, sunt definite reprezentări sub logica temporală pentru misiunea de nivel înalt, cum ar fi *Logica Temporală Liniară* (LTL) și *Logica Temporală a Intervalelor Metice* (MITL). La final, sunt descrise metrice și metode folosite pentru evaluarea comparativă a soluțiilor propuse.

Capitolul 3 prezintă o abordare de planificare, incluzând o tehnică de descompunere a misiunii pentru un sistem multi-robot. Metoda descompune o misiune globală LTL, returnând acțiuni individuale ce sunt atribuite ulterior roboților, îmbunătățind comportamentul autonom și reducând sincronizările. Soluția este evaluată prin simulări în medii 3D, având ca model propus un algoritm de partajare a spațiului 3D. Acest capitol este bazat pe articolul:

- **Sofia Hustiu**, Ioana Hustiu, Marius Kloetzer, and Cristian Mahulea. [LTL task decomposition for 3D high-level path planning](#). In *Journal of Control Engineering and Applied Informatics*, 23(3), pp.76-87, 2021.

Capitolul 4 propune un cadru nou numit *Rețea Petri Compusă*, unde beneficiile a două reprezentări sunt combinate printr-un strat intermediar. Modelul combinat descrie comportamentul echipei de roboți în raport cu spațiul de lucru și misiunea globală. Strategia de planificare se bazează pe rezolvarea problemelor de optimizare, iar această metodă permite și o execuție paralelă a mișcărilor roboților. Acest capitol este bazat pe articolele:

- **Sofia Hustiu**, Cristian Mahulea, Marius Kloetzer, and Jean-Jacques Lesage. [On multi-robot path planning based on Petri net models and LTL specifications](#). In *IEEE Transactions on Automatic Control*, vol. 69, no. 9, pp. 6373-6380, 2024.
- **Sofia Hustiu**, Cristian Mahulea, and Marius Kloetzer. [Parallel motion execution and path rerouting for a team of mobile robots](#). In *IFAC-PapersOnLine*, 55(28), 73–78. In *16th IFAC Workshop on Discrete Event Systems WODES*, 2022.

Capitolul 5 extinde modelul *Rețea Petri Compusă*, adăugând constrângeri temporale, rezultând un cadru *Rețea Petri Compusă Temporizată*, care integrează atât constrângerile spațiale, cât și cele temporale referitoare la acțiunile roboților. Acest cadru este flexibil, potrivit atât pentru echipe omogene de roboți, cât și pentru echipe heterogene. Acest capitol este bazat pe articolele:

- **Sofia Hustiu**, Dimos V. Dimarogonas, Cristian Mahulea, and Marius Kloetzer. [Multi-robot Motion Planning under MITL Specifications based on Time Petri Nets](#). In *2023 European Control Conference (ECC)* (pp. 1-8). *IEEE*, 2023.
- **Sofia Hustiu**, Alexandru-Florian Brasoveanu, and Andrei-Iulian Iancu. [Integration of MITL for Cobots Workflow in a Manipulating Application](#). In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8, 2024.

Capitolul 6 explorează un model bazat pe paradigma *Nets within Nets*, utilizând o structură ierarhică de rețele Petri. Această abordare permite includerea stărilor locale și globale ale fiecărui robot, asigurând sincronizarea necesară îndeplinirii misiunii globale, chiar și cu roboți cu capabilități spațiale diferite. Acest capitol este bazat pe articolul:

- **Sofia Hustiu**, Eva Robillard, Joaquín Ezpeleta, Cristian Mahulea, and Marius Kloetzer. [Multi-robot Motion Planning based on Nets-within-Nets Modeling and Simulation](#). *Under review*. Available in [Online]: <https://arxiv.org/abs/2304.08772>, 2023.

Capitolul 7 examinează rezultatele obținute din simulări numerice și aplicații reale. Evaluarea cadrelor propuse este comparată cu alte modele de Sisteme cu Evenimente Discrete din literatura de specialitate, luând în considerare timpul de execuție și lungimea traiectoriilor roboților. Rezultatele sunt prezentate în trei articole publicate, dintre care două dintre ele apar în contribuțiile din Capitolul 5 (al doilea articol) și 6.1:

- **Sofia Hustiu**, Marius Kloetzer, Eva Robillard, Alejandro López-Martínez, and Cristian Mahulea. [Whitening of greenhouse's roof using drones and Petri net models](#). In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2022.

Capitolul 2

Sisteme cu evenimente discrete pentru soluții de planificare a traseelor în sisteme multi-agent

Acest capitol introduce conceptele esențiale pentru planificarea traseului unui sistem multi-agent, destinat unei misiuni specifice. Sunt prezentate noțiuni de bază despre un sistem multi-robot într-un spațiu dat, urmate de două modele formale în sisteme de evenimente discrete pentru mișcarea echipei robotice, luând în considerare restricțiile spațiale ale mediului. În final, sunt abordate metode de descriere a misiunii, ce impun restricții spațiale și temporale, facilitând secvențierea și sincronizarea acțiunilor, ilustrare cu exemple practice.

2.1 Ipotezele problemei

Această secțiune introduce noțiuni fundamentale esențiale a înțelege modelarea unei echipe robotice și misiunea atribuită acesteia.

2.1.1 Spațiul de lucru al sistemului multi-robot

Să considerăm un set de roboți mobili, adesea denumiți *agenți* pentru a extinde aplicabilitatea metodei propuse în diferite domenii, notat cu $\mathcal{R} = \{r_1, r_2, \dots, r_{|\mathcal{R}|}\}$. Acest set reprezintă roboți care se presupun a fi punctiformi și omnidirecționali, evoluând în spații 2D sau 3D (de exemplu, modelând UAV-uri, cunoscute și sub denumirea de drone). Acești roboți operează într-un mediu cunoscut E , ce conține mai multe regiuni de interes (ROIs) definite prin setul $\mathcal{Y} = \{y_1, y_2, \dots, y_{|\mathcal{Y}|}\}$. Să notăm cu $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$ setul de celule reprezentate de politoape disjuncte pe baza unei tehnici de descompunere în celule [2]. Aceste metode ajută la modelarea discretă a spațiului de lucru pentru roboți, fiind potrivite mediilor 2D și 3D.

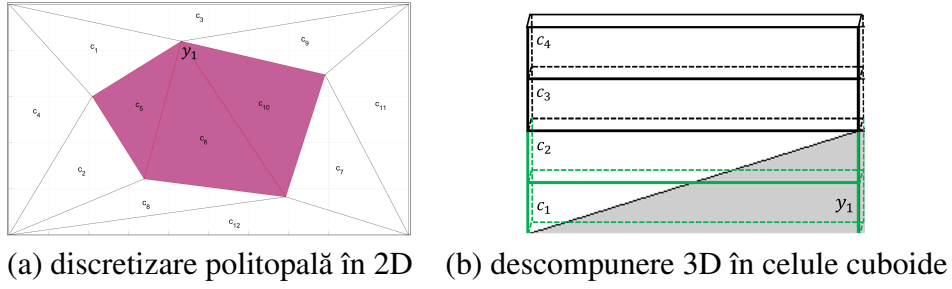


Fig. 2.1. Exemplu de celule etichetate ca regiuni de interes

2.2 Discretizare în celule

Înainte de a introduce o funcția de etichetare, să definim propoziția atomică $b_i \in \mathcal{B}$ fiind o variabilă cu valori logice (Adevărat sau Fals). Aceasta va fi denumită *observație*, care poate fi *activă* sau *inactivă*. Fiecare propoziție atomică este asociată unei regiuni de interes, relația fiind unu-la-unu. Astfel, funcția de etichetare se definește ca $h : \mathcal{C} \rightarrow 2^{\mathcal{B}}$, cu $\mathcal{C}$ setul celulelor și $2^{\mathcal{B}}$ puterea mulțimii $\mathcal{B}$. Notăția $\emptyset \in 2^{\mathcal{B}}$ reprezintă eticheta pentru spațiul liber.

Example 2.2.1 Pentru a ilustra conceptele discutate, să examinăm un exemplu simplu din Figura 2.1. Partea stângă arată o discretizare a mediului în 12 celule, iar partea dreaptă prezintă o decompoziție 3D a unui cuboid în 4 celule. În urma discretizării, trei celule sunt etichetate cu b_1 pentru regiunea de interes y_1 (magenta). În decompoziția 3D, celulele mixte sunt etichetate cu b_1 , iar altele sunt libere, exemplificate prin $h(c_1) = \emptyset$. ■

2.3 Reprezentări ale agenților prin evenimente discrete

Această subsecțiune introduce conceptele de bază pentru modelarea mișcării echipei de roboți în reprezentări SED, care analizează schimbările de stare declanșate de evenimente în timp discret.

2.3.1 Rețea Petri

Definition 2.3.1 [2] Un sistem Petri Net pentru Mișcarea Roboților (RMPN) este caracterizat prin tupla $\mathcal{Q} = \langle \mathcal{N}, m_0, \mathcal{B}, h \rangle$, unde:

- $\mathcal{N} = \langle P, T, Post, Pre \rangle$ este o rețea Petri compusă din:
 - Un set de poziții P (fiecare poziție reprezentând o celulă din setul $\mathcal{C}$);
 - Un set de tranziții T , fiecare semnificând mișcarea unui robot între celule adiacente;

- $Post \in \{0, 1\}^{|P| \times |T|}$, matricea post-incidentei, care detaliază conexiunile dintre tranziții și locuri. În mod specific, $Post[p, t] = 1$ dacă tranziția $t \in T$ este conectată la locul $p \in P$, iar $Post[p, t] = 0$ altfel;
- $Pre \in \{0, 1\}^{|P| \times |T|}$, matricea pre-incidentei, care descrie conexiunile dintre locuri și tranziții. În mod specific, $Pre[p, t] = 1$ dacă locul $p \in P$ este conectat la tranziția $t \in T$, iar $Pre[p, t] = 0$ altfel;
- m_0 este vectorul de marcaj inițială, unde $m_0[p]$ denotă numărul de roboți localizați inițial în celula $c \in \mathcal{C}$;
- $\mathcal{B} \cup \{\emptyset\}$ reprezintă setul de simboluri de ieșire, reprezentate prin propoziții atomice, asociate cu setul de regiuni de interes $\mathcal{Y}$;
- $h : P \rightarrow 2^{\mathcal{B}}$ este funcția de observație. Dacă poziția p_i conține cel puțin un token (indică prezența unui robot în celula c_i), atunci regiunea/ile de interes asociate cu propozițiile atomice din setul $\mathcal{B}$ sunt considerate ca vizitate.

2.3.2 Rețea Petri Temporizată

Definiția formală a modelului *Rețea Petri Temporizată* extinde notația RMPN, bazându-se pe concepte teoretice din [3, 4] și înglobează constrângerile de timp ale echipei de roboți:

Definition 2.3.2 *Un model Rețea Petri Temporizată este definit prin tupla $\mathcal{TPN} = \langle \mathcal{Q}, I \rangle$, unde:*

- $\mathcal{Q} = \langle \mathcal{N}, m_0, \mathcal{B}, h \rangle$ este modelul RMPN definit anterior (Definiția 2.3.1);
- $I : T \rightarrow [\mathbb{Q}_+ \rightarrow \mathbb{Q}_+ \cup \{\infty\}]$ atribuie fiecărei tranziții un interval static de timp. Intervalul de timp pentru o tranziție este reprezentat de tupla $I(t) = [\alpha, \beta]$, unde $0 \leq \alpha < \infty$ este timpul minim de pornire, iar $0 \leq \beta \leq \infty$ este timpul maxim de pornire, având condiția ca $\alpha \leq \beta$ dacă $\beta \neq \infty$, sau $\alpha < \beta$ dacă $\beta = \infty$.

2.4 Misiunea roboților

Planificarea traseului pentru roboți a evoluat de la un singur robot la scenarii multi-agent, în care aceștia colaborează pentru a vizita regiuni secvențial sau sincron, evitând obstacole. Misiunile sunt exprimate prin operatori Booleani și coordonarea agenților se face prin constrângeri spațiale și temporale.

2.4.1 Logica Temporală Liniară

Expresiile complexe care implică acțiuni secvențiale sau sincronizate sunt extinse cu *operatori temporali* precum $\mathcal{U}$, $\diamond$, $\square$ și $\bigcirc$. Acesta formează Logica Temporală Liniară (LTL) [5, 6], definită pe un set de propoziții atomice $\mathcal{B}$.

Specificațiile LTL pot fi modelate ca un automat Büchi [7]. În această teză, se consideră o traducere a misiunii LTL într-un automat Büchi nedeterminist [8].

2.4.2 Logica Temporală cu Interval Metric

Spre deosebire de Logica Temporală Liniară, care impune constrângeri spațiale și temporale pentru sincronizarea și secvențierea regiunilor de interes ale sistemului multi-agent, Logica Temporală cu Interval Metric (MITL) exprimă constrângeri legate de timp, indicând intervalul în care o regiune trebuie atinsă.

De exemplu formula MITL $\varphi = \square_{[0,20]} \neg b_1$ specifică faptul că regiunea y_1 este întotdeauna evitată în primele 20 de unități de timp. O astfel de formulă se poate modela ca un automat temporizat [9].

2.5 Criterii de comparație pentru strategiile de planificare ale sistemele multi-agent

Această secțiune prezintă metrici utilizate pentru compararea metodelor propuse cu alte soluții de planificare SED. Metricile includ:

- **Dimensiunea modelului:** numărul de locuri și tranziții în reprezentarea rețelei Petri sau a Sistemului de Tranziții.
- **Timpul de rulare:** timpul necesar pentru a găsi o soluție care să asigure misiunea LTL sau MITL.
- **Lungimea traiectoriei:** numărul total de tranziții activate în modelul robotic, reprezentând calea totală parcursă.

Sunt introduse două metode de comparație:

- **Metoda FB** [7]: utilizează o abordare secvențială cu automatonul Büchi și optimizare MILP, producând o soluție suboptimă.
- **Metoda TS** [10]: modelează mișcarea fiecărui robot individual prin Sistemul de Tranziții, rezolvând traiectoriile printr-un algoritm de căutare pe graf.

Capitolul 3

Metodă de decompozitie a acțiunilor pentru sistemele multi-agent

Acest capitol propune o metodă pentru alocare ale acțiunilor într-un sistem multi-robotic, care include planificarea traseului și evitarea coliziunilor. Metoda implică decompoziția acțiunilor dintr-o specificație globală LTL co-sigură, unde sarcinile sunt atribuite independent fiecărui robot. Rezultatele sunt evaluate într-un spațiu 3D, împărțit în celule printr-o tehnică personală de decompoziție 3D. Simulările includ mișcarea UAV-urilor care asigură misiunea globală LTL. Contribuția principală este soluția pentru decompoziția acțiunilor, obținând traiectorii independente pentru fiecare robot, fără necesitatea comunicării sau sincronizării între aceștia, oferind o soluție scalabilă și eficientă pentru misiuni complexe.

3.1 Decompoziția acțiunilor

Această secțiune se concentrează pe decompoziția automată a misiunii LTL φ într-un set de sub-formule, denumite *acțiuni* (sau sarcini), care îndeplinesc două cerințe esențiale: *independența* și *completitudinea*. *Independența* înseamnă că sarcinile sunt non-conflictuale și pot fi realizate fără sincronizare între roboți. Astfel, o sarcină φ_i nu trebuie să interfereze cu alta φ_j , unde $j \neq i$. *Completitudinea* asigură că misiunea globală φ este îndeplinită atunci când toate sarcinile sunt finalizate.

Metoda propusă convertește formula LTL φ într-un automat Büchi B , folosind o hartă de observație h . Automatului i se aplică o simplificare pentru a permite activarea tranzițiilor de către un singur robot, iar apoi sunt generate traiectorii fără bucle. Decompoziția continuă prin validarea traiectoriilor și asigurarea independenței sarcinilor, rezultând acțiuni ce garantează îndeplinirea completă a misiunii globale.

3.2 Soluția pentru descompunerea în acțiuni

Deoarece B a fost simplificat înainte de a calcula traiectoriile valide, fiecare φ_i poate fi executat de un singur robot, asigurându-se că nu există suprapunere între regiunile de interes. Proprietățile de independență și completitudine garantează că, formula LTL originală este satisfăcută, eliminând necesitatea sincronizării, doar dacă toate sarcinile sunt realizate independent.

Acțiunile φ_i sunt atribuite celor $|\mathcal{R}|$ roboți prin construirea unei matrici de cost W și calcularea alocării optime a sarcinilor pentru a minimiza funcția de cost. Soluția este returnată de problema de optimizare:

$$\begin{aligned} \min & N \cdot \lambda + \sum_{r=1}^{|\mathcal{R}|} Z(r, :) \cdot W(:, r) \\ \text{s.t. } & \sum_{r=1}^{|\mathcal{R}|} Z(i, r) = 1, \forall i = 1, \dots, |\rho| - 1 \\ & Z(r, :) \cdot W(:, r) \leq \lambda, \forall r = 1, \dots, |\mathcal{R}| \\ & Z \in \{0, 1\}^{|\mathcal{R}| \times |\rho| - 1}, \lambda \in \mathbb{R}_{\geq 0} \end{aligned} \quad (3.1)$$

Example 3.2.1 Să considerăm matricea de cost $W = \begin{bmatrix} 11 & 3 \\ 4 & 12 \end{bmatrix}$ pentru Figura 3.1. Rezolvând MILP (3.1), sarcina $\varphi_1 = b_1$ este atribuită robotului r_1 (roșu), iar $\varphi_2 = b_3 \vee b_4$ robotului r_2 (albastru). Robotul r_1 se deplasează în regiunea y_1 (setul D) și evită y_2 (setul I), iar robotul r_2 se deplasează într-o celulă cu b_3 sau b_4 , evitând b_2 . Secvențele rezultate sunt Seq_{r_1} cu 4 celule și Seq_{r_2} cu 3 celule, evidențiindu-se prin numărul de mișcări ale roboților.

■

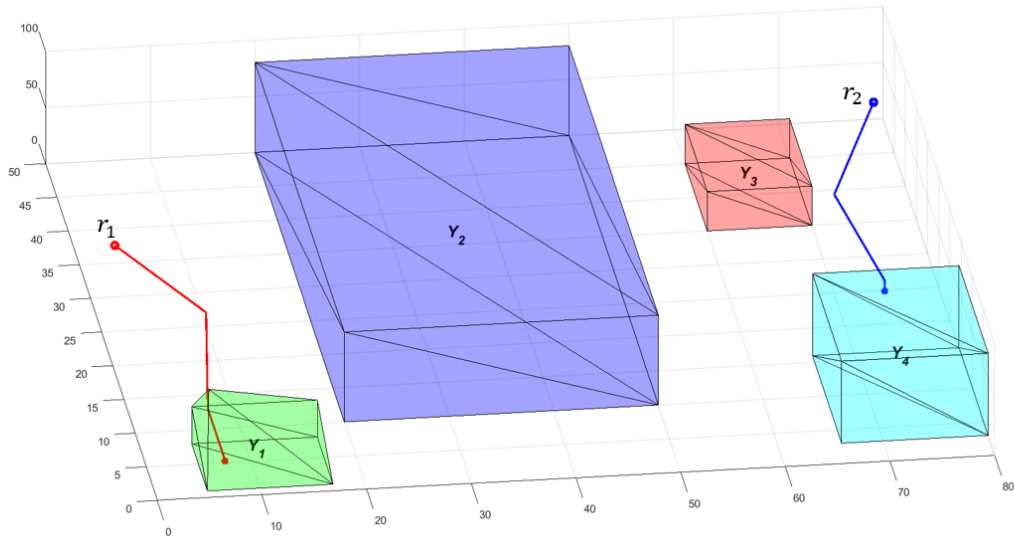


Fig. 3.1. Independent trajectories of the two agents

3.3 Evaluarea numerică

Se consideră un mediu 3D cu șase regiuni de interes, $\mathcal{Y} = \{y_1, y_2, y_3, y_4, y_5, y_6\}$, unde y_2 și y_3 se intersectează. Formula LTL impune ca echipa de roboți să viziteze regiunile y_1, y_4, y_5, y_6 și intersecția dintre y_2 și y_3 .

$$\varphi = \Diamond b_1 \wedge \Diamond (b_2 \wedge b_3) \wedge \Diamond b_4 \wedge \Diamond b_5 \wedge \Diamond b_6 \quad (3.2)$$

Pentru cele două tipuri de partiții, *Grid* și *OctTree* s-a utilizat o precizie de $\varepsilon = 16$. Metoda *OctTree* generează un model cu 1849 de noduri și 13081 de tranziții calculat în 6,7 secunde, considerând celule cu dimensiuni diferite dar cu același raport de lungime-lățime-înălțime cu spațiul de lucru. Metoda *Grid* produce un model cu 4096 de noduri și 27136 de tranziții, în 19 secunde, considerând celulele egale între ele.

O contribuție semnificativă a metodei de decompoziție LTL este algoritmul implementat. Pentru a ilustra acest lucru, costul îndeplinirii misiunii este calculat pentru o singură dronă care efectuează întreaga misiune, comparativ cu distribuirea sarcinilor între roboți. Pentru partiția *OctTree*, drona ar trebui să traverseze în medie 34,6 celule, iar pentru *Grid*, 68,74 celule pentru simulări cu 100 de experimente. Comparând aceste valori, se poate concluziona că metoda de decompoziție *OctTree* reduce numărul de celule traversate și permite execuția paralelă a sarcinilor independente, timpul de finalizare fiind reflectat de costul maxim 27.52 față de 47.10. Acest lucru se poate observa în Tabelul 3.1.

Table. 3.1. Valorile medii pentru costurile maxime, respectiv costurile totale.

Costul mediu	<i>OctTree</i>	<i>Grid</i>
Costul maxim	11.31	19.13
Costul total	27.52	47.10

Capitolul 4

Planificarea traiectoriilor cu specificații LTL și optimizarea traiectoriilor pentru sisteme multi-robot

Acest capitol prezintă un cadru bazat pe rețele Petri pentru echipe robotice, care modelează atât mișcarea roboților, cât și îndeplinirea unei misiuni LTL globale. Se definește un model de *Rețea Petri Compusă* pentru echipe omogene ce ating sau evită regiuni de interes. Exemplele ilustrează teoria propusă.

În a doua parte, se propune o abordare eficientă de planificare a mișcărilor, care permite mișcarea paralelă a roboților, menținând îndeplinirea misiunii, chiar și când traseele sunt redirectionate. Metoda se inspiră din algoritmul Bancherului, evitând blocajele. Simulările validează soluția, comparând-o cu o metodă existentă pentru un scenariu de spațiu îngust.

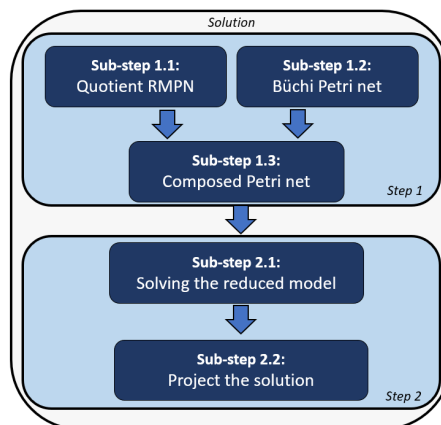


Fig. 4.1. Diagrama algoritmului global

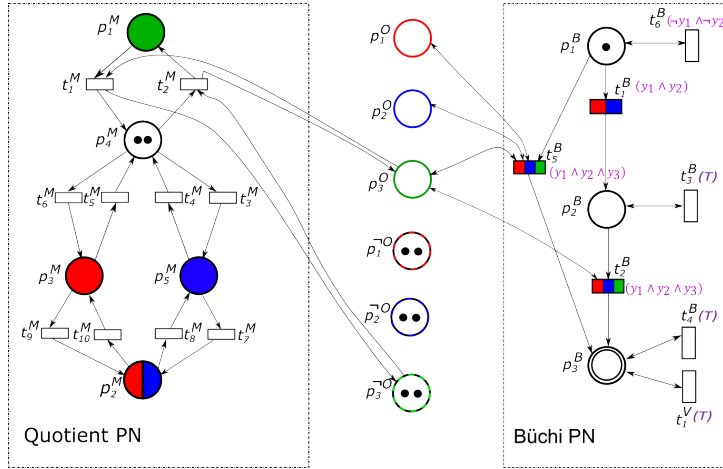


Fig. 4.2. Parte a modelului Petri net compus, luând în considerare observația activă, respectiv inactivă b_3 , asociată cu regiunea y_3

4.1 Conceptul unui strat intermediar

Ideea principală a combinării celor două modele este integrarea avantajelor (i) modelului de rețea Petri pentru o echipă de agenți, care permite ajustarea topologiei pe măsură ce numărul agenților variază, și (ii) modelului de automat pentru o misiune de nivel înalt LTL. Cadrul propus construiește un singur model de rețea Petri prin utilizarea unui strat intermediar de poziții.

4.2 Modelul Rețelei Petri Compuse

Etapele construirii și a utilizării acestui model se observă în Figura 4.1. De asemenea, Figura 4.2 prezintă o reprezentare parțială a rezultatului de compunere a rețele compuse, considerându-se regiunea de interes y_3 .

4.2.1 Soluția bazată pe probleme de optimizare

Soluția, bazată pe traiectoriile roboților, este obținută din două probleme de optimizare MILP: prima rezolvă modelul redus Quotient RMPN, iar a doua proiectează soluția în modelul complet RMPN. Se prezintă un exemplu numeric, comparând cadrul propus cu articolul [7]. Algoritmul de planificare calculează mișcările roboților pentru a îndeplini misiunea φ , cautându-se un marcaj final în rețea.

4.2.2 Exemplu numeric

Validarea cadrului *Rețea Petri Compusă* este demonstrată prin simulări numerice. Modelul este implementat în RMTool - MATLAB [11], iar MILP este rezolvat cu CPLEX Optimizer [12], rulând pe un laptop cu i7 (2.20GHz, 8GB RAM).

Discuție. Simulările arată că metoda de compunere a RMPN și automatului Büchi reduce timpul de calcul și dimensiunea modelului comparativ cu **FB**. Modelul rezultat este scalabil cu numărul de roboți. În schimb, modelul **FB** include pozițiile din RMPN $\mathcal{Q}$, iar abordările **TS** [10] depind exponențial de dimensiunea echipei.

4.3 Reconfigurarea traiectoriei, considerând execuția paralelă a mișcărilor.

Alocarea eficientă a sarcinilor și planificarea mișcărilor îmbunătățesc productivitatea echipei, iar în medii constrânse, cum sunt pasajele înguste, realocarea dinamică previne coliziunile și întârzierile [13, 14]. Soluția propusă permite mișcări paralele și traiectorii fără coliziuni pentru o echipă omogenă de roboți, optimizând execuția prin reducerea stărilor de așteptare și utilizând rerutarea traiectoriilor printr-un MILP. Algoritmul, integrat în RMTool [11], sprijină mișcarea eficientă a roboților și îndeplinirea misiunii fără observații noi.

4.3.1 Factori determinanți

Mișcarea fără coliziuni și blocaje a roboților este abordată prin **Algoritmul Bancherului**, care previne blocajele prin simularea alocării maxime a resurselor. Deși algoritmul este eficient, aplicarea sa în planificarea mișcărilor roboților mobili este limitat la câteva studii [15, 16]. Abordarea actuală îmbunătățește soluția din [1]. Pe parcursul acestei strategii de planificare, se va utiliza terminologia definită în [1].

4.3.2 Algoritmul pentru mișcarea paralelă a roboților

Ideea principală este de a modifica mișcările inițiale din traiectoriile returnate în [1], încurajând roboții să se miște paralel când este posibil. Aceasta este o problemă de alocare a resurselor, având ca scop prevenirea blocajelor și coliziunilor. Finalizarea proceselor corespunde echipelor care ajung la celulele țintă. Strategia se bazează pe conceptul de *stare sigură* din algoritmul Bancherului adaptat contextului actual. O precondiție este returnarea ordinii de acces a resurselor. De asemenea, dacă roboții sunt blocați din cauza mișcării în paralel, atunci o recalculare a traiectoriilor va avea loc pe baza rezolvării unei probleme de optimizare. În Figure 4.3 se poate observa un scenariu unde roboții erau anterior blocați în

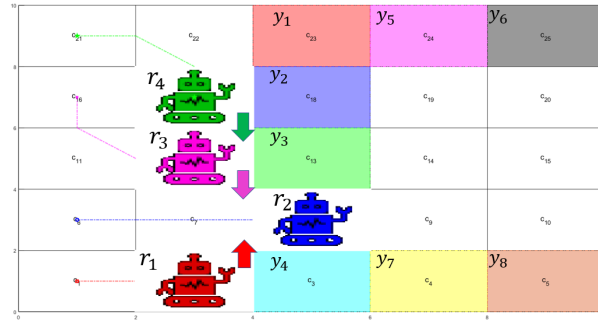


Fig. 4.3. Scenariul pentru rerutarea traiectoriilor roboților într-un mediu cu 4 roboți și 8 regiuni de interes (ROI).

mișcarea lor, iar recalcularea traseului este dată de problema de optimizare. În această figură, roboții trebuie să ocolească regiunile $y_1, \dots, y_4$ și să se oprească în regiunile $y_5, \dots, y_8$.

4.3.3 Rezultate numerice

Algoritmul a fost implementat și integrat în toolbox-ul open-source RMTTool - MATLAB [11].

Tabela 4.1 prezintă rezultate pentru scenarii cu resurse partajate și complexitate crescută. Timpul de rulare se referă la rezolvarea optimizării pentru fiecare abordare: MILP-urile din [1] calculează traiectoriile folosind un model global, iar MILP-ul actual gestionează rerutarea, când este necesar. Rerutarea nu este aplicabilă (NA) în [1], fiind o contribuție nouă a lucrării actuale.

Table. 4.1. Numerical data comparison between the sequential approach [1] and the parallel (current) approach

Environment scenario	Numerical performances	Planning approach [2]	Case I	Case I or Case II (here $N = \lfloor \mathcal{R} /2 \rfloor$)
Grid-based 5×5 , with $ \mathcal{R} = 4$ robots	Step count	24	11	10
	Count of rerouting actions	NA	0	2 times
	Execution time [sec]	0.1	0.1	$0.1 + (\mu = 7 \cdot 10^{-3}, \text{std} = 1 \cdot 10^{-2})$
	Order of robots to reach their final cell	r_3, r_1, r_4, r_2	r_2, r_1, r_3, r_4	r_2, r_1, r_3, r_4
	Assigned final regions for tuple (r_1, r_2, r_3, r_4)	(y_5, y_7, y_6, y_8)	(y_5, y_7, y_6, y_8)	(y_6, y_5, y_8, y_7)
Grid-based 10×20 , with $ \mathcal{R} = 10$ robots	Step count	155	39	43
	Count of rerouting actions	NA	0	6 times
	Execution time [sec]	3.2	3.2	$3.2 + (\mu = 1 \cdot 10^{-2}, \text{std} = 1 \cdot 10^{-2})$
Grid-based 20×20 , with $ \mathcal{R} = 20$ robots	Step count	352	78	7
	Count of rerouting actions	NA	2 times	14 times
	Execution time [sec]	12	12.0	$12 + (\mu = 9 \cdot 10^{-3}, \text{std} = 3 \cdot 10^{-3})$

Capitolul 5

Planificarea traiectoriilor cu specificație MITL pentru sisteme multi-robot

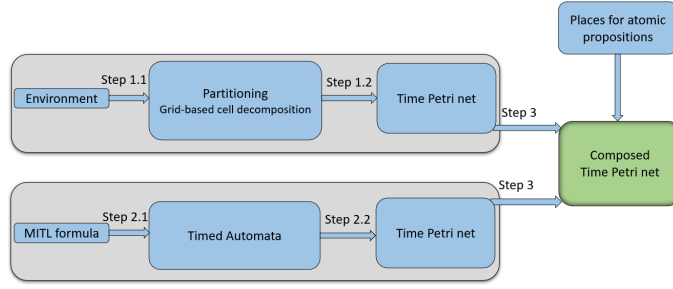
Acest capitol extinde cadrul Petri net propus anterior, adăugând constrângeri de timp pentru comportamentul echipei de roboți și misiunea MITL. Modelul *Rețea Petri Temporizată Compusă* include (i) o strategie de planificare bazată pe verificare a modelului în timp real și (ii) adaptarea modelului pentru sisteme robotice eterogene, unde fiecare robot îndeplinește o misiune MITL individuală. Se prezintă un mecanism de sincronizare între misiuni MITL, folosind o topologie fixă de Rețea Petri Temporizată. Exemple și simulări sunt incluse, iar Capitolul 7 validează modelul pentru aplicații de fabricație.

5.1 Flux de lucru pentru modelare

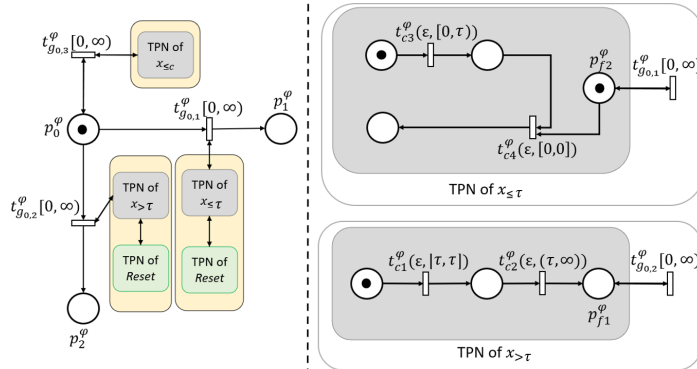
Problem 1 *Se consideră că mișcarea unei echipe de roboți este reprezentată printr-un model Rețea Petri Temporizată. Având o specificație globală MITL asupra setului de propoziții atomice, se generează automat traiectoriile roboților pentru a satisface specificația, respectând atât constrângerile spațiale, cât și cele temporale.*

Cadrul *Rețea Petri Temporizată Compusă* oferă o strategie de planificare pentru sisteme multi-agent, operând sub constrângeri de timp. Acesta poate gestiona scenarii în care întreaga echipă de roboți trebuie să îndeplinească o misiune globală MITL sau sub-grupuri de agenți cu capacități similare trebuie să finalizeze misiuni MITL. Figura 5.1 prezintă soluția propusă, unde sunt combinate două modele Petri cu timp: modelul mediului (pasul 1) și modelul specificației MITL (pasul 2). Deși structura acestui model este similară cu cea prezentată în capitolul precedent, avantajul său major este reducerea utilizării modelelor temporale pentru planificarea mișcării unui sistem robotic.

Pasul a doilea necesită o translație dintre automat temporizat într-o rețea Petri Temporizată pentru specificația MITL. Mai multe studii au analizat aceste asocieri dintre cele două modele, relația lor cu bisimilaritatea temporală și acceptarea limbajelor temporale. O

Fig. 5.1. General Framework of *Composed Time Petri net* model

abordare prezentată în [17], traduce un automat temporizat având constrângeri de ceas într-o rețea Petri Temporizată, asigurând bisimilaritatea temporală între cele două. Fiecare stare sin automat corespunde unei poziții, iar tranzițiile reprezintă constrângeri de ceas și resetări. Prin această metodă se garantează condițiile temporale din automat în noul model Petri. Topologiile de rețea utilizate pentru restrangerile temporale se pot observa în Figura 5.2. Un exemplu pentru rețeaua Petri Temporizată Compusă se poate vizualiza în Figura 5.3.

Fig. 5.2. Un exemplu de conversie a unui automat temporizat într-o rețea Petri Temporizată (stânga) pentru formula MITL $\varphi = \Diamond_{\leq \tau} b_1$, care încorporează topologii de ceas (dreapta).

5.2 Mecanism de coordonare pentru multiple modele de tip Rețea Petri Temporizată Compusă

Principalul avantaj al modelului de tip *Rețea Petri Temporizată Compusă* este combinarea specificațiilor MITL cu capacitățile roboților. Modelul permite:

1. **Misiuni comune** – Un grup de roboți îndeplinește o misiune MITL colectivă.
2. **Misiuni individuale** – Fiecare robot are o misiune separată, necesitând coordonare.

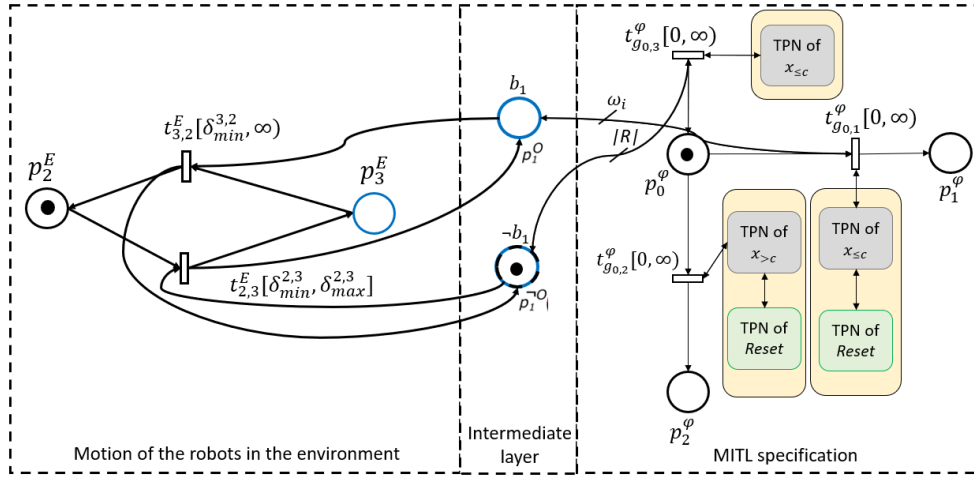


Fig. 5.3. Vedere parțială din *Reteaua Petri Temporizată Compusă*, pentru propoziția atomică b_1 inclusă în specificația MITL $\varphi = \Diamond_{\leq \tau} b_1$.

Adaptabilitatea modelului acoperă:

- **Propoziții atomice:** Restricții temporale pentru acțiuni sau regiuni de interes.
- **Model robotic:** Capacități de mișcare și acțiuni ale roboților în regiuni specifice.
- **Coordonare:** Sincronizarea roboților prin locuri și tranziții comune.

Sincronizarea aliniază mișcările roboților cu restricțiile temporale, ajustând modelul pentru a satisface cerințele misiunii.

Example 5.2.1 Considerăm formulele MITL $\varphi_1 = \Diamond_{\tau_1} b_1$ și $\varphi_2 = \Diamond_{\tau_2} b_2$, unde robotul r_1 trebuie să ajungă în regiunea y_1 în timpul τ_1 , iar r_2 în regiunea y_2 în timpul τ_2 .

Figura 5.4 arată sincronizarea între roboți. Presupunem $\tau_1 = \tau_2$, de exemplu, 5 unități de timp. Misiunile pot fi îndeplinite dacă r_1 finalizează în 2 secunde și r_2 în 3, sincronizându-se în timpul minim dintre τ_1 și τ_2 .

Pentru sincronizare, locurile finale $p_1^{\varphi_1}$ și $p_1^{\varphi_2}$ primesc token-uri. Locuri de așteptare ($p_4^{\varphi_1}, p_4^{\varphi_2}$) și tranziția $t_{g0,0}^{\varphi}$ sunt adăugate pentru sincronizarea misiunilor, iar token-urile sunt direcționate către locurile ce modelează îndeplinirea misiunilor, cu activare imediată. ■

5.3 Model-checking approach in numerical evaluation

Pentru planificare se utilizează ROMEO [18], care verifică accesibilitatea unui marcaj și oferă tranzițiile necesare.

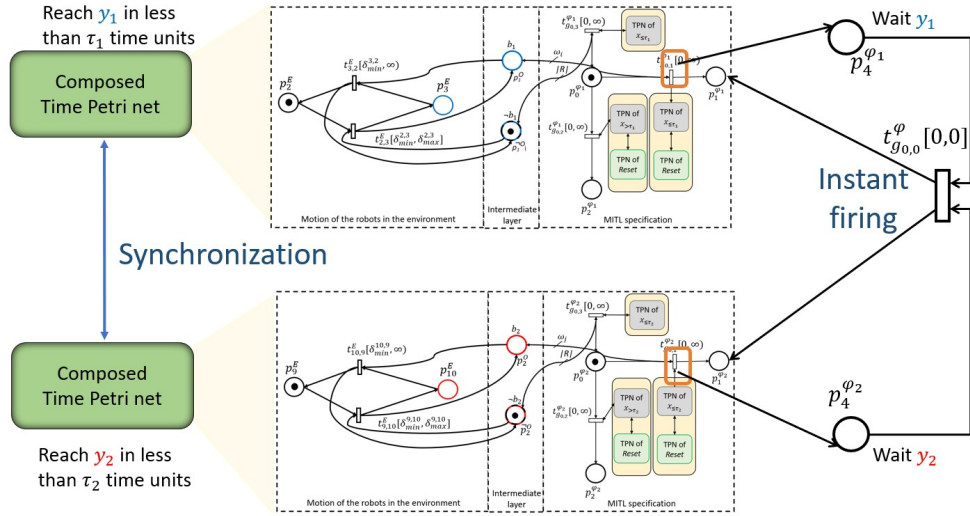


Fig. 5.4. Mecanism de sincronizare pentru modele individuale de tip Rețea Petri Temporizată Compusă

Studiul din [19] analizează un sistem cu brațe robotice împărțite în două subgrupuri: r_{pp} , care transportă barele, și r_c , care le conectează între ele. Lucrarea curentă adaugă flexibilitate în numărul roboților și integrează constrângeri de timp în planificarea mișcării, presupunând datele cunoscute dinainte. În această teză se alocă o misiune MITL pentru fiecare astfel de grup robotic. Tabelul 5.1 afișează rezultatele considerând mărimea modelului compus temporal, numărul de roboți și timpul de rezolvare printr-o metodă model-checking.

Table. 5.1. Evaluare numerică pentru modelul de tip Rețea Petri Temporizată Compusă

Misiunea MITL	Dimensiunea modelului	Numărul de roboți	Timpul de rezolvare [sec]
$\varphi_{r_{pp}}$	$ P^C = 50, T^C = 65$	2	2.2
		3	81.3
φ_{r_c}	$ P^C = 42, T^C = 58$	2	1.8
		3	49

Capitolul 6

Planificarea traiectoriei cu specificație LTL bazată pe o abordare ierarhică pentru un sistem multi-robot.

Acest capitol abordează problema planificării mișcării pentru echipe heterogene de roboți care respectă o specificație globală. Se propune un nou cadru *Rețea Petri de Nivel Înalt pentru o echipă robotică*, care utilizează paradigma *Nets-within-Nets* (NwN). Cadrul include două caracteristici: comportamentul roboților și misiunea echipei sunt reprezentate prin *rețea de tip obiect*, iar o *rețea de tip sistem* coordonează starea globală a sistemului robotic prin *Global Enabling Function* (GEF). Soluția este obținută prin simularea HLrtPN în software specializat. Exemplele, bazate pe misiuni LTL, demonstrează fezabilitatea computațională a cadrului propus.

Contribuțiile acestui capitol sunt:

- Introducerea modelului *Rețea Petri de Nivel Înalt pentru o echipă robotică* pentru planificarea mișcării echipelor de roboți heterogene, cu o funcție de sincronizare numită *Funcție de Activare Globală*.
- Implementarea în Renew [20] și prezentarea exemplului pe web [21], cu exemple ilustrative bazate pe formule LTL.
- Evaluarea soluției prin simulare numerică în Renew și compararea cu diferite reprezentări DES, inclusiv un scenariu real dintr-un spital (detaliat în Capitolul 7).

6.1 Paradigma Nets-within-Nets

Recent, planificarea mișcării în domeniul roboticii a devenit o provocare, mai ales pentru sisteme multi-robot eterogene, utilizate în agricultură [22] sau cartografiere [23]. Această lucrare propune strategii de planificare care păstrează avantajele cadrelor bazate pe evenimente

discrete. Capitolul 2.1 evidențiază rețelele Petri care simplifică modelarea echipelor robotice omogene prin asocierea unui jeton fiecărui robot. Pentru roboți eterogeni, utilizarea rețelelor Petri din Definiția 2.3.1 poate fi limitativă, sugerând alternative precum rețele Petri colorate [24] sau coordonarea mai multor rețele.

Soluția prezentată se bazează pe paradigma *Nets-within-Nets* [25], care permite token-urilor să transfere informații despre stările altor procese. Fiecare jeton este vizualizat ca o *rețea de tip obiect*, iar relația dintre aceste rețele este capturată într-o *rețea de tip sistem*, oferind o vedere globală a sistemului [26]. Această metodologie orientată pe obiecte permite reprezentări de nivel înalt și exprimarea acțiunilor sincrone și asincrone [27].

6.2 Formularea problemei

Problema: Lucrarea abordează alocarea sarcinilor și planificarea pentru un sistem multi-robot eterogen într-un mediu cunoscut, cu regiuni de interes. Echipa trebuie să îndeplinească o misiune globală, definită ca o specificație co-safe LTL, cu constrângeri spațiale de vizitare/-evitare a regiunilor și constrângeri temporale de secvențiere și sincronizare.

Se oferă o explicație intuitivă pentru o mai bună înțelegere a conceptului. Modelul NwN, numit *Rețea Petri de Nivel Înalt pentru o echipă robotică*, include: (i) rețele de obiecte ce modelează roboți (*RobotOPNs*) și o rețea de obiecte pentru misiune (*SpecOPN*); (ii) o rețea de sistem unde fiecare jeton corespunde unei rețele de obiecte. Rețeaua de sistem controlează evoluția, sincronizând tranzițiile dintre RobotOPNs și SpecOPN. Mișcarea roboților între regiuni actualizează valorile propozițiilor atomice $\mathcal{B}$. Funcția de sincronizare asigură coerența sistemului.

Rețeaua propusă este ilustrată în Figura 6.1. În mod specific:

- **Figura 6.1-ii** prezintă rețeaua de sistem cu două locuri: Rb , conținând trei jetoane, fiecare corespunzând unui RobotOPN, și Ms , cu un token pentru SpecOPN. Aceasta include trei tranziții: t_1, t_2 și t_3 , reprezentând mișcarea sincronizată a unuia, a doi sau a trei roboți.
- **Figura 6.1-i** arată SpecOPN pentru formula $\varphi = \Diamond b_3$, considerând misiunea îndeplinită când p_2^S conține un jeton (adică un robot ajunge în regiunea y_3). Acest lucru este posibil prin tranziția t_1^S , activată când un robot intră în y_3 .
- **Figura 6.1-iii** ilustrează un RobotOPN o^1 , identic pentru ceilalți roboți dacă sunt similari. Fiecare loc din RobotOPN corespunde unei celule din $\mathcal{C}$, iar roboții pornesc dintr-un spațiu inițial. De exemplu, dacă robotul r_1 activează tranziția $t_1^{o^1}$, schimbă valoarea propozițiilor atomice (b_3 devine *True*).

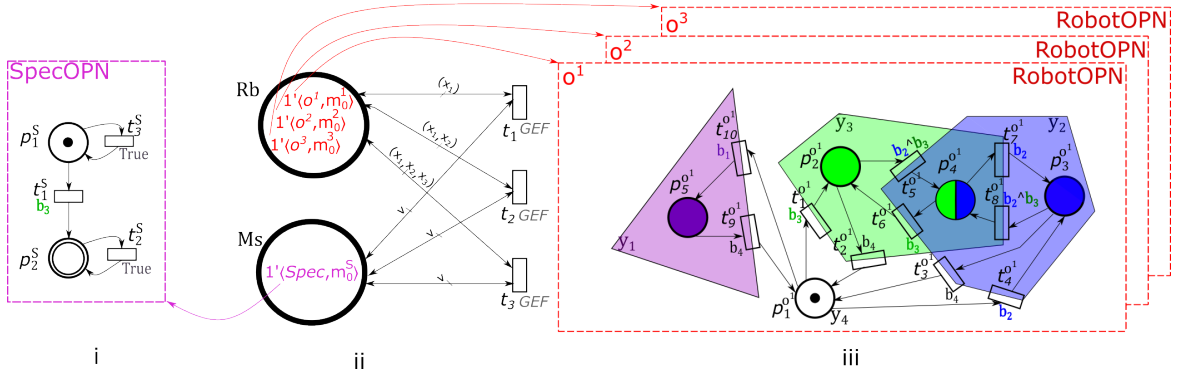


Fig. 6.1. Exemplu demonstrând paradigma Nets-within-Nets: (i) Rețeaua Petri pentru specificație, (ii) Rețeaua de sistem, (iii) Rețeaua Petri pentru roboți

Funcția de sincronizare asigură faptul că dacă r_1 se mișcă și activează tranziția t_1 , atunci misiunea va fi îndeplinită prin activarea tranziției t_1^S în sincron cu $t_1^{o^1}$. Astfel, jetonul ajunge în p_2^S , iar misiunea este considerată îndeplinită.

6.3 Nets-within-Nets adaptate pentru planificarea traiectoriei

Cadru propus *Rețea Petri de Nivel Înalt pentru o echipă robotică* este definit formal mai jos.

6.3.1 Rețea de tip obiect

Dinamica echipei de roboți este modelată prin rețele de tip obiect *RobotOPN* pentru fiecare tip de robot și *SpecOPN* pentru specificația ce conține cerințele misiunii globale. Subsecțiunile următoare definesc aceste rețele.

Rețea de tip obiect pentru specificație

Definition 6.3.1 (SpecOPN) O Rețea de tip obiect pentru specificație este reprezentată printr-un tuplu $Spec = \langle P, P_f, T, F, \lambda \rangle$, unde: P și T sunt mulțimile disjuncte de poziții și tranziții, $P_f \subseteq P$ este mulțimea pozițiilor finale, iar $F \subseteq (P \times T) \cup (T \times P)$ este mulțimea arcelor. Funcția de etichetare a tranzițiilor $\lambda_{\wedge}(t) \equiv t_{\wedge}$ atribuie fiecărei tranziții $t \in T$ o formulă booleană definită prin utilizarea propozițiilor atomice $\mathcal{B}$ și a negațiilor acestora.

Rețea de tip obiect pentru robot

Definition 6.3.2 (RobotOPN) *Data fiind o mulțime de celule $\mathcal{C}$ care modelează spațiul de lucru al echipei de roboți, Rețeaua Petri pentru obiecte robotice este modelul unui robot, exprimat prin tuplul $o = \langle P, T, F, h, \lambda, \gamma \rangle$:*

- P este mulțimea finită de poziții, bijectivă cu mulțimea $\mathcal{C}$. Fiecare poziție este asociată cu un element $c_i \in \mathcal{C}$ în care robotul poate intra;
- T este mulțimea finită de tranziții. O tranziție $t_{ij} \in T$ este adăugată între două poziții $p_i, p_j \in P$ doar dacă robotul poate să se deplaseze din orice poziție din celula c_i în celula c_j din spațiul de lucru partitionat E ;
- $F \subseteq (P \times T) \cup (T \times P)$ este mulțimea arcelor. Dacă t_{ij} este tranziția care modelează mișcarea de la p_i la p_j , atunci $(p_i, t_{ij}) \in F$ și $(t_{ij}, p_j) \in F$;
- $h_{\wedge}$ este funcția de etichetare a locurilor $p \in P$, definită în secțiunea anterioară și care asociază fiecărei poziții o formulă booleană peste mulțimea de propoziții $\mathcal{B}$;
- $\lambda_{\wedge}$ este funcția de etichetare booleană a tranzițiilor $t \in T$, astfel încât $\lambda_{\wedge}(t_i) = h(t_i^{\bullet})_{\wedge}$;
- $\gamma: P \rightarrow \mathcal{C}$ este funcția de asociere. Dacă locul $p_i \in P$ este asociat cu $c_i \in \mathcal{C}$, atunci $\gamma(p_i) = c_i$. ■

6.3.2 Rețea Petri de Nivel Înalt pentru o echipă robotică

Această secțiune prezintă modelul propus, denumit *Rețea Petri de Nivel Înalt pentru o echipă robotică*, care permite vizualizarea globală prin sincronizarea între rețeaua de tip sistem și rețelele de tip obiecte definite anterior.

Definition 6.3.3 *O Rețea Petri de Nivel Înalt pentru o echipă robotică este un tuplu $\mathcal{N} = \langle \bar{P}, \bar{T}, \mathcal{O}, \mathcal{S}, \text{Vars}, \bar{F}, W, \mu_{cap} \rangle$, unde:*

- $\bar{P} = \{Rb, Ms\}$ este mulțimea de poziții;
- $\bar{T} = \{t_1, t_2, \dots, t_s\}$ este mulțimea de tranziții;
- $\mathcal{O} = \{\langle o^1, m_0^1 \rangle, \langle o^2, m_0^2 \rangle, \dots, \langle o^n, m_0^n \rangle\}$ este mulțimea de n modele RobotOPN, câte unul pentru fiecare robot;
- $\mathcal{S} = \langle \text{Spec}, m_0^S \rangle$ este sistemul SpecOPN;
- $\text{Vars} = \{v, x_1, x_2, \dots, x_n\}$ este mulțimea de variabile;
- $\bar{F}$ este mulțimea arcelor: $\bar{F} = \bigcup_{t \in \bar{T}, p \in \bar{P}} \{(p, t), (t, p)\}$;

Capitolul 7

Rutine și aplicații software dezvoltate

Acest capitol prezintă simulări și aplicații care evaluează rezultatele metodelor propuse în teza folosind rețelele Petri. Scopul este de a demonstra versatilitatea cadrelor propuse prin analiza strategiilor de planificare pentru echipe robotice: UAV-uri identice, roboți mobili heterogeni și cobots.

7.1 Implementare

Una dintre contribuțiile acestei teze este procesul de implementare, prin care mai multe metode au fost puse online ca surse deschise pentru cercetători. Accesul la aceste metode reprezintă un punct de plecare pentru îmbunătățiri în domeniul planificării traiectoriilor robotice. Această secțiune descrie procesul de implementare, având în vedere validarea metodelor formale prezentate în capitolele anterioare. Implementarea este detaliată în trei faze: (a) implementarea în MATLAB, axată pe formalismul *Composed Petri net* (Capitolul 4.2) și algoritmul de rerutare a traiectoriilor (Capitolul 4.3) disponibil în RMTool [1], (b) implementarea Renew [28] pentru modelul *Rețea Petri de Nivel Înalt pentru o echipă robotică* sub paradigma Nets-within-Nets (Capitolul 6.1), descrisă în [21], cu funcțiile GitHub folosite pentru validare.

7.1.1 Implementarea în MATLAB

Implementarea a fost realizată în RMTool, un toolbox open-source din MATLAB [1], disponibil [aici]([29]). Acesta facilitează implementarea și validarea metodelor propuse pe baza Sistemelor cu Evenimente Discrete, fiind reproductibil și adaptabil nevoilor utilizatorului.

Metodele implementate în MATLAB sunt ușor de validat și comparat cu altele din literatură, datorită integrării evaluării algoritmilor în același spațiu de lucru. Figura 7.1 prezintă interfața grafică a RMTool, care include metode de planificare a traiectoriilor implementate prin scripturi MATLAB. Secțiunea subliniază două metode din teză: (i)

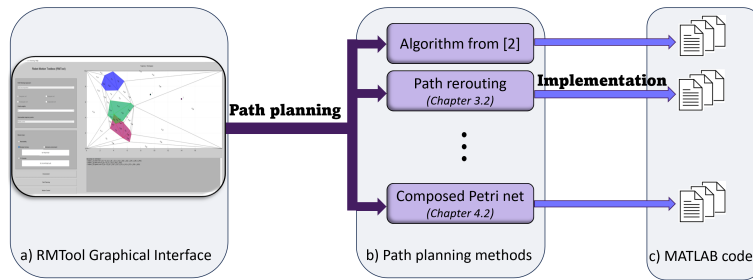


Fig. 7.1. Diagrama implementării din MATLAB în RMTool

algoritmul de rerutare a traiectoriilor și (ii) cadrul *Rețea Petri Compusă*, comparat cu alți algoritmi din literatură din RMTool.

7.1.2 Implementarea în Renew

Renew (Reference Net Workshop) este un simulator bazat pe Java pentru modelarea și simularea rețelelor Petri, având un suport puternic pentru paradigma Nets-within-Nets. Aceasta permite modelarea ierarhică, oferind design modular și reprezentări complexe ale sistemelor. Instrumentul susține și caracteristici orientate pe obiecte, permițând utilizatorilor să definească și să manipuleze clase și obiecte în cadrul acestei paradigme [20].

Proiectele GitHub menționate aici [21]s conțin mai multe tipuri de fișiere:

- *.rnw* pentru fișiere Renew ce modelează *RobotOPN*, *SpecOPN* și rețeaua sistemului,
- *.hoa* și *.pnml* pentru a reprezenta misiunea LTL în *SpecOPN*,
- *.java* pentru scripturi Java.

Principalele scripturi Java sunt *Eval*, care gestionează sincronizarea între rețeaua sistemului și rețelele obiect, și *Perf_eval*, care analizează rezultatele, precum pașii minimi și maximi pentru echipa robotică și mișcările realizate în *SpecOPN*.

7.2 Rezultatele simulărilor

În această secțiune sunt prezentate câteva studii de caz, care includ scenarii precum: (i) vopsirea acoperișului serelor de către o echipă de drone, aplicat în agricultură, și (ii) un scenariu futurist de spital, unde o echipă de roboți mobili heterogeni trebuie să se sincronizeze pentru a îndeplini o misiune. În ambele cazuri, strategia de planificare se bazează pe formalismul Petri net.

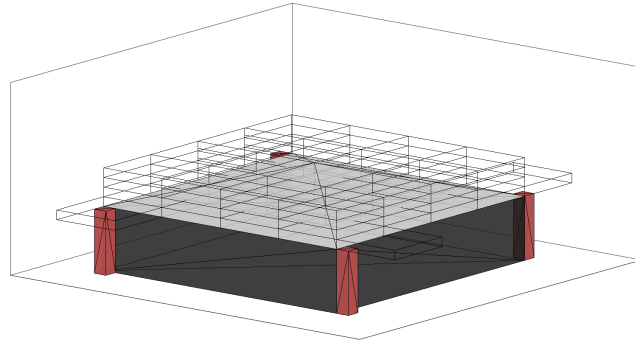
Fig. 7.2. Decompunerea în celule pentru acoperișul serii, cu precizia $\varepsilon = 4$.

Table. 7.1. Evaluarea rezultatelor simulărilor

Scenariu	Timpul de rezolvare pentru partiționarea spațiului[sec]	No. ROI	Timpul de rezolvare pentru o instanță de MILP [sec]	No. tururilor
$r = 8$ UAVs, precizie $\varepsilon = 4$ PN model with $ P = 60, T = 288$	0.24	32	0.06	2
$r = 4$ UAVs, precizie $\varepsilon = 4$ PN model with $ P = 60, T = 288$	0.24	32	0.06	3
$r = 4$ UAVs, precizie $\varepsilon = 8$ PN model with $ P = 404, T = 2328$	2.1	112	0.17	4
$r = 4$ UAVs, precizie $\varepsilon = 16$ PN model with $ P = 2980, T = 18856$	29.66	512	13.26	6

7.2.1 Vopsirea acoperișului serelor de către o echipă de drone

În acest scenariu, considerăm un mediu 3D cunoscut, care include o seră și mai multe stații de încărcare pentru o echipă de UAV-uri identice. Fiecare dronă are o autonomie limitată de zbor, care este resetată la stațiile de încărcare. Misiunea echipei este de a vopsi acoperișul serii, ceea ce presupune vizitarea și vopsirea mai multor regiuni de interes de pe suprafața acestuia.

Figura 7.2 prezintă decompoziția celulară în zona acoperișului, iar Tabelul 7.1 conține rezultatele bazate pe decompoziția în celule și rezolvarea automată a copririi pe baza unei probleme de optimizare ce ține cont de faptul că nu toate dronele zboară în același timp din cauza faptului că o parte din ele se vor încarca în stațiile de încărcare. Pentru a facilita vizualizarea rezultatelor, o animație video poate fi accesată la acest link.

7.2.2 Asistarea sistemelor robotice multi-agent în domeniul sănătății

Spitalul are trei etaje și include 12 camere, $\mathcal{Y} = \{y_1, \dots, y_{12}\}$, printre care săli de examinare (y_7, y_{11}), săli de operație (y_8, y_{12}), săli de terapie (y_9, y_{10}) și camere RMN (y_1, y_3, y_4), monitorizabile doar din y_2, y_5, y_6 . Mai multe camere sunt accesate simultan: pacienții sunt examinați inițial, iar dacă este necesar un RMN, camerele aferente trebuie accesate sincron pentru monitorizare. Ulterior, sălile de operație și terapie pot fi vizitate pentru aprovizionare și curățare. Misiunea LTL co-safe este definită în (7.1), corelată cu propozițiile atomice ale fiecărei regiuni din $\mathcal{Y}$ (ex.: b_1 pentru y_1).

$$\varphi = \Diamond(b_1 \wedge b_2) \wedge \Diamond(b_3 \wedge b_6) \wedge \Diamond(b_4 \wedge b_5) \wedge \neg(b_1 \vee b_3 \vee b_4) \mathcal{U}(b_7 \vee b_{11}) \wedge \Diamond(b_8 \vee b_9 \vee b_{10} \vee b_{12}) \quad (7.1)$$

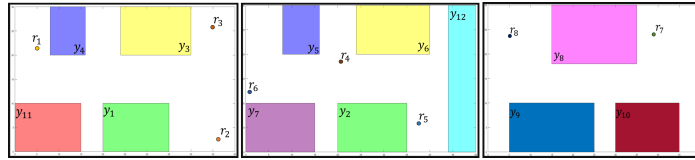


Fig. 7.3. Exemplu de scenariu din spital cu trei etaje și 12 camere pentru un sistem multi-robot.

Analiza pentru echipe de 2-8 roboți (Tabel 7.2) arată că **(a) dimensiunea modelului** influențează **(b) timpul de execuție**. Simulările confirmă planificarea mișcării, iar **(c) lungimea traiectoriei** poate fi optimizată, deoarece soluțiile actuale sunt aleatorii.

No. roboți	Tipul de roboți				No. simulărilor	(a) Dimensiunea modelului	(b) Timpul de rezolvare [s]	(c) Lungimea traiectoriilor
	r_p	r_m	r_{sc}	r_a				
2	1			1	1000	$(P , T) = (42, 294)$	0.39	27
3	1	1	1		1000	$(P , T) = (37, 293)$	0.24	29
4	2	1	1		1000	$(P , T) = (44, 305)$	1.1	25
5	2	2	1		1000	$(P , T) = (48, 311)$	1.89	15
6	2	2	2		1000	$(P , T) = (54, 321)$	10.4	14
7	2	2	2	1	250	$(P , T) = (69, 337)$	107.15	20
8	3	2	2	1	245	$(P , T) = (76, 349)$	228.91	16

Table. 7.2. Rezultatele comparației pentru echipa eterogenă de roboți în abordarea propusă

7.3 Validare experimentală a modelului Rețea Petri Temporizată Compusă

În Figura 7.4 se folosește modelul *Rețea Petri Temporizată Compusă*. Se aplică o metodă de decompoziție a spațiului, împărțind mediul de lucru într-un model Petri Temporizat,

capturând spațiul liber și regiunile de interes. Pentru fiecare robot se specifică câte o misiune. Cei doi roboți trebuie să lucreze împreună pentru a construi o structură rigidă. Rezultatele experimentale pot fi evaluate accesând link-ul [30].

Ecuția 7.2 descrie misiunea robotului r_1 , care începe prin atingerea poziției $Idle_{r_1}$ în timpul τ_1 , apoi ridică un obiect $PickPiece$ în τ_2 unități de timp, și îl plasează în timpul τ_3 . Robotul r_2 primește două misiuni (7.3): $\varphi_{r_2}^1$ presupune ridicarea unui instrument $PickTool$ în τ_2 , după ce ajunge la poziția $Idle_{r_2}$ în τ_1 , și întoarcerea la $Idle$ în τ_3 ; iar $\varphi_{r_2}^2$ presupune utilizarea instrumentului $UseTool$ în $\tau_4 = \tau_1 + \tau_2$. Cerințele de timp sunt interdependente.

Specificațiile MITL pentru r_1 , respectiv r_2 sunt notate cu φ_{r_1} , respectiv $\varphi_{r_2}^1$ și $\varphi_{r_2}^2$ și sunt date după cum urmează:

$$\varphi_{r_1} = \Diamond_{\tau_1} Idle_{r_1} \wedge (Idle_{r_1} \rightarrow \Diamond_{\tau_2} PickPiece) \wedge (PickPiece \rightarrow \Diamond_{\tau_3} PlacePiece) \quad (7.2)$$

$$\begin{aligned} \varphi_{r_2}^1 &= \Diamond_{\tau_1} Idle_{r_2} \wedge (Idle_{r_2} \rightarrow \Diamond_{\tau_2} PickTool) \wedge (PickTool \rightarrow \Diamond_{\tau_3} Idle_{r_2}) \\ \varphi_{r_2}^2 &= \Diamond_{\tau_4} UseTool \wedge (UseTool \rightarrow \Diamond_{\tau_3} Idle_{r_2}) \end{aligned} \quad (7.3)$$

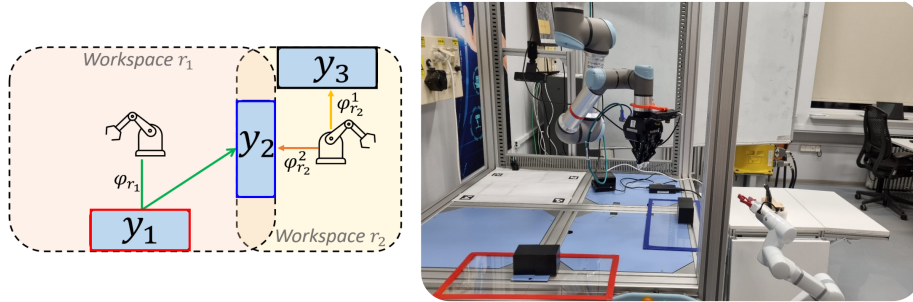


Fig. 7.4. Configurația spațiului de lucru experimental: diagramă din vedere de sus (partea stângă) și vedere laterală reală (partea dreaptă)

Capitolul 8

Observații finale

Această teză analizează strategii de planificare pentru sisteme multi-robot utilizând logica temporală și Sistemele cu Evenimente Discrete (SED), facilitând modelarea și integrarea constrângerilor spațiale și temporale (LTL și MITL).

8.1 Contribuții

În **Capitolul 3**, sunt propuse două strategii de alocare a sarcinilor pentru echipe multi-robot. Prima metodă descompune o misiune LTL în sarcini independente, iar a doua metodă asigură mișcarea paralelă a roboților.

Capitolul 4 introduce un cadru inovator care combină modelul redus *quotient* al spațiului de lucru cu cel al misiunii. **Capitolul 5** extinde cadrul cu restricții temporale.

În **Capitolul 6**, este propus un model bazat pe *Nets-within-Nets* (NwN) pentru echipe robotice eterogene, permițând coordonarea diferitelor tipuri de roboți printr-o rețea de obiecte (*RobotOPN* și *SpecOPN*).

Contribuțiile acestei teze au fost diseminate astfel:

1. Patru articole în reviste:

- (a) **Sofia Hustiu**, Cristian Mahulea, Marius Kloetzer, and Jean-Jacques Lesage. [On multi-robot path planning based on Petri net models and LTL specifications](#). In *IEEE Transactions on Automatic Control*, vol. 69, no. 9, pp. 6373-6380, 2024.
- (b) **Sofia Hustiu**, Ioana Hustiu, Marius Kloetzer, and Cristian Mahulea. [LTL task decomposition for 3D high-level path planning](#). In *Journal of Control Engineering and Applied Informatics*, 23(3), pp.76-87, 2021.
- (c) **Sofia Hustiu**, Eva Robillard, Joaquín Ezpeleta, Cristian Mahulea, and Marius Kloetzer. [Multi-robot Motion Planning based on Nets-within-Nets Modeling and Simulation](#). *Under review*. Available in [Online]: <https://arxiv.org/abs/2304.08772>, 2023.

- (d) **Sofia Hustiu**. [Prerequisites to Design a Collision Free Trajectory in a 3D Dynamic Environment for an UAV](#). In *Bulletin of the Polytechnic Institute of Iași. Electrical Engineering, Power Engineering, Electronics Section*, 67(2), pp.65-78, 2021.

2. Șase lucrări de conferință:

- (a) **Sofia Hustiu**, Alexandru-Florian Brasoveanu, and Andrei-Iulian Iancu. [Integration of MITL for Cobots Workflow in a Manipulating Application](#). In *2024 IEEE 29th Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 1–8, 2024.
- (b) **Sofia Hustiu**, Dimos V. Dimarogonas, Cristian Mahulea, and Marius Kloetzer. [Multi-robot Motion Planning under MITL Specifications based on Time Petri Nets](#). In *2023 European Control Conf. (ECC) (pp. 1-8)*. IEEE, 2023.
- (c) **Sofia Hustiu**, Cristian Mahulea, and Marius Kloetzer. [Parallel motion execution and path rerouting for a team of mobile robots](#). In *IFAC-PapersOnLine*, 55(28), 73–78. In *16th IFAC Workshop on Discrete Event Systems WODES*, 2022.
- (d) **Sofia Hustiu**, Marius Kloetzer, Eva Robillard, Alejandro López-Martínez, and Cristian Mahulea. [Whitening of greenhouse's roof using drones and Petri net models](#). In *2022 IEEE 27th Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, 2022.
- (e) **Sofia Hustiu**, Marius Kloetzer, and Cristian Mahulea. [Mission assignment and 3D path planning for a team of UAVs](#). In *2021 25th Int. Conf. on System Theory, Control and Computing (ICSTCC)* (pp. 401-406). IEEE, 2021.
- (f) **Sofia Hustiu**, Marius Kloetzer, and Adrian Burlacu. [Collision Free Path Planning for Unmanned Aerial Vehicles in Environments with Dynamic Obstacles](#). In *2020 24th Int. Conf. on System Theory, Control and Computing (ICSTCC)* (pp. 520-525). IEEE, 2020.

8.2 Direcții viitoare de cercetare

Limitările cadrelor propuse sunt: (i) implementarea open-source, (ii) reducerea complexității pentru echipe mari, (iii) extinderea modelului Nets-within-Nets pentru planificare cu constrângeri de timp și (iv) planificarea în medii necunoscute. Primele două provocări pot fi abordate prin extinderea RMTTool, integrând medii incerte și suport MITL pentru o planificare controlată. O direcție viitoare pentru Nets-within-Nets este combinarea planificării de nivel înalt cu algoritmi de control pentru mișcări realiste ale roboților. Planificarea în medii necunoscute este importantă pentru scenarii de căutare și salvare, unde se pot explora abordări probabilistice și Logica Temporală Probabilistică.

Bibliografie

- [1] C. Mahulea, M. Kloetzer, and J.-J. Lesage, “Multi-robot path planning with boolean specifications and collision avoidance,” *IFAC-PapersOnLine*, vol. 53, no. 4, pp. 101–108, 2020.
- [2] C. Mahulea, M. Kloetzer, and R. González, *Path Planning of Cooperative Mobile Robots Using Discrete Event Models*. John Wiley & Sons, 2020.
- [3] P. Merlin and D. Farber, “Recoverability of communication protocols-implications of a theoretical study,” *IEEE transactions on Communications*, vol. 24, no. 9, pp. 1036–1043, 1976.
- [4] L. Popova-Zeugmann and L. Popova-Zeugmann, *Time Petri nets*. Springer, 2013.
- [5] E. M. M. Clarke, D. Peled, and O. Grumberg, *Model checking*. MIT Press, 1999.
- [6] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT Press, 2008.
- [7] M. Kloetzer and C. Mahulea, “Path planning for robotic teams based on LTL specifications and Petri net models,” *Discrete Event Dynamic Systems*, vol. 30, no. 1, pp. 55–79, 2020.
- [8] P. Wolper, M. Vardi, and A. Sistla, “Reasoning about infinite computation paths,” in *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science* (E. N. et al., ed.), (Tucson, AZ), pp. 185–194, 1983.
- [9] R. Alur and D. Dill, “The theory of timed automata,” in *Workshop/School/Symposium of the REX Project (Research and Education in Concurrent Systems)*, pp. 45–73, Springer, 1991.
- [10] X. C. Ding, M. Kloetzer, Y. Chen, and C. Belta, “Automatic deployment of robotic teams,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 75–86, 2011.
- [11] R. González, C. Mahulea, and M. Kloetzer, “A Matlab-based interactive simulator for mobile robotics,” in *IEEE CASE’2015: Int. Conf. on Autom. Science and Engineering*, 2015.
- [12] IBM, “IBM ILOG CPLEX Optimization Studio. Software.” <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>, 2016.
- [13] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot task allocation: A review of the state-of-the-art,” *Cooperative robots and sensor networks 2015*, pp. 31–51, 2015.

- [14] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, “Multiple mobile robot task and motion planning: A survey,” *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–35, 2023.
- [15] L. Kalinovic, T. Petrovic, S. Bogdan, and V. Bobanac, “Modified Banker’s algorithm for scheduling in multi-AGV systems,” in *International Conference on Automation Science and Engineering*, pp. 351–356, IEEE, 2011.
- [16] V. Bobanac and S. Bogdan, “Routing and scheduling in multi-AGV systems based on dynamic banker algorithm,” in *16th Mediterranean Conference on Control and Automation*, pp. 1168–1173, IEEE, 2008.
- [17] B. Bérard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux, “Comparison of the expressiveness of timed automata and time Petri nets,” in *International conference on formal modeling and analysis of timed systems*, pp. 211–225, Springer, 2005.
- [18] G. Gardey, D. Lime, M. Magnin, and O. Roux, “Romeo: A tool for analyzing time Petri nets,” in *International Conference on Computer Aided Verification*, pp. 418–423, Springer, 2005.
- [19] M. Momeni, J. Relefors, A. Khatry, L. Pettersson, A. V. Papadopoulos, and T. Nolte, “Automated fabrication of reinforcement cages using a robotized production cell,” *Automation in Construction*, vol. 133, p. 103990, 2022.
- [20] O. Kummer, F. Wienberg, M. Duvigneau, M. Köhler, D. Moldt, and H. Rölke, “Renew—the reference net workshop,” in *Tool Demonstrations, 21st International Conference on Application and Theory of Petri Nets, Computer Science Department, Aarhus University, Aarhus, Denmark*, pp. 87–89, 2000.
- [21] “Nets-within-Nets for Motion Planning with Renew.” <https://eva-robillard.github.io/>, 2024. Accessed: 24-11-2024.
- [22] C. Ju and H. I. Son, “A hybrid systems-based hierarchical control architecture for heterogeneous field robot teams,” *IEEE Transactions on Cybernetics*, 2021.
- [23] J. J. Roldán, P. Garcia-Aunon, M. Garzón, J. De León, J. Del Cerro, and A. Barrientos, “Heterogeneous multi-robot system for mapping environmental variables of greenhouses,” *Sensors*, vol. 16, no. 7, 2016.
- [24] M. Allison and M. Spradling, “Modeling sub-team formations for heterogeneous multi-robot systems using colored Petri-net semantics,” in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 237–243, IEEE, 2022.
- [25] K. Jensen and G. Rozenberg, *High-level Petri nets: theory and application*. Springer Science & Business Media, 2012.
- [26] R. Valk, “Object Petri nets,” in *Advanced Course on Petri Nets*, pp. 819–848, Springer, 2003.
- [27] M. Köhler, D. Moldt, and H. Rölke, “Modelling mobility and mobile agents using Nets within nets,” in *ICATPN 2003: Applications and Theory of Petri Nets, Eindhoven, The Netherlands, June 23–27*, pp. 121–139, Springer, 2003.

- [28] L. Cabac, M. Haustermann, and D. Mosteller, “Renew 2.5—towards a comprehensive integrated development environment for Petri net-based applications,” in *Application and Theory of Petri Nets and Concurrency: 37th International Conference, Petri NETS 2016, Toruń, Poland, June 19-24, 2016. Proceedings 37*, pp. 101–112, Springer, 2016.
- [29] C. Mahulea, “Robotmotiontoolbox-rmtool-under-matlab.” <https://github.com/cmahulea/RobotMotionToolbox-RMTool-under-MATLAB>, 2024. Available at: <https://github.com/cmahulea/RobotMotionToolbox-RMTool-under-MATLAB>.
- [30] H. Sofia, I. Andrei-Iulian, and B. Florian-Alexandru, “Integration of MITL for cobots workflow in manipulating application.” [Available at] https://youtu.be/4IQ0_5Uv9bQ.