

Autoreferat

1 Imię i nazwisko

Dominik Pająk

2 Posiadane dyplomy i stopnie naukowe

- 13.06.2014 Stopień naukowy: doktor (informatyka)
Tytuł rozprawy (ang.): *Algorithms for Deterministic Parallel Graph Exploration.*
Tytuł rozprawy (pl.): *Deterministyczne, współbieżne algorytmy eksploracji grafów.*
Promotorzy: prof. Ralf Klasing, dr hab. Adrian Kosowski
Instytucja: Université de Bordeaux
Rozprawa uzyskała nagrodę **Prix de thèse 2015**
- 22.06.2011 Tytuł zawodowy: magister (informatyka)
Tytuł rozprawy: *Progowy algorytm alertu w radiowych sieciach ad-hoc.*
Promotor: dr hab. Marek Klonowski
Instytucja: Politechnika Wroclawska

3 Dotychczasowe zatrudnienie w jednostkach naukowych

03.2020 - obecnie	Adiunkt naukowy	Politechnika Wroclawska	Polska
08.2017 - 04.2019	Postdoc	Massachusetts Institute of Technology	USA
04.2016 - 07.2017	Postdoc/Adiunkt naukowy	Politechnika Wroclawska	Polska
11.2014 - 10.2015	Postdoc	University of Cambridge	Anglia
08.2014 - 10.2014	Postdoc	Friedrich-Schiller-Universität Jena	Niemcy
10.2011 - 07.2014	Doktorant	Inria/Université de Bordeaux	Francja

3.1 Zatrudnienie na stanowiskach badawczych w przemyśle

04.2019 - 03.2020 ML/AI Researcher Tooploox Wrocław, Polska

4 Omówienie osiągnięć, o których mowa w art. 219 ust. 1 pkt. 2 Ustawy.

4.1 Tytuł

Wybrane problemy algorytmiczne w bardziej realistycznych modelach obliczeń rozproszonych.

Prace składające się na dzieło habilitacyjne

- [H1] Anna R. Dornhaus, Nancy A. Lynch, Frederik Mallmann-Trenn, Dominik Pajak, Tsvetomira Radeva. Self-stabilizing task allocation in spite of noise. *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, USA, July 15-17, 2020*, strony 201–211. ACM, 2020. (Ranga A).
- [H2] Frederik Mallmann-Trenn, Yannic Maus, Dominik Pajak. Noidy communixatipn: On the convergence of the averaging population protocol. *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, wolumen 132 serii *LIPICs*, strony 148:1–148:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. (Ranga A).
- [H3] Seth Gilbert, Nancy A. Lynch, Calvin Newport, Dominik Pajak. On simple back-off in unreliable radio networks. *Theor. Comput. Sci.*, 806:489–508, 2020. (100 Punktów na liście MNiSW, wersja konferencyjna zdobyła nagrodę Best Paper na konferencji OPODIS 2018).
- [H4] Marek Klonowski, Dominik Pajak. Electing a leader in wireless networks quickly despite jamming. *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, strony 304–312. ACM, 2015. (Ranga A).
- [H5] Artur Menc, Dominik Pajak, Przemyslaw Uznanski. Time and space optimality of rotor-router graph exploration. *Inf. Process. Lett.*, 127:17–20, 2017. (70 Punktów na liście MNiSW).
- [H6] Marek Klonowski, Dominik Pajak. Brief announcement: Broadcast in radio networks, time vs. energy tradeoffs. *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, strony 115–117. ACM, 2018. (Ranga A*).
- [H7] Andrej Ivaskovic, Adrian Kosowski, Dominik Pajak, Thomas Sauerwald. Multiple random walks on paths and grids. *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, wolumen 66 serii *LIPICs*, strony 44:1–44:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. (Ranga A).

Autorzy w powyższych publikacjach są podani w kolejności alfabetycznej. Wkład w poszczególne prace został szczegółowo opisany w odrębnych załącznikach.

4.2 Wstęp

W wielu praktycznych problemach niezbędne jest, aby pracę lub obliczenia, konieczne do rozwiązania problemu, wykonywała więcej niż jedna jednostka. Na przykład, wyszukiwarki internetowe odświeżają indeks sieci stron WWW poprzez wielokrotne przeszukiwanie sieci, przy użyciu wielu agentów (robotów internetowych). Jedna jednostka jest niewystarczająca ze względu na ograniczenie czasowe (w styczniu 2021 sieć WWW zawierała ponad 1.5 miliarda witryn², a indeks stron należy aktualizować każdego dnia). Innym przykładem jest sieć czujników monitorujących pewien obszar, gdzie z powodu fizycznych ograniczeń pojedynczy sensor nie może objąć całego obszaru swoim zasięgiem. Używanie wielu urządzeń rozwiązuje te problemy, ale powoduje powstanie nowych. Jak rozdzielić pracę równo pomiędzy jednostki? Jak zorganizować wymianę informacji między jednostkami? Jak zaprojektować algorytmy tak, aby proste, ograniczone jednostki mogły je wykonać? Znajdowanie odpowiedzi na takie pytania jest jednym z głównych celów dziedziny obliczeń rozproszonych. Obecnie istnieje już rozbudowana, formalna teoria obliczeń rozproszonych (np. Lyn96). Większość z tej teorii, rozwijanej od lat 70, skupia się na mocno zoptymalizowanych czasowo algorytmach, działających w modelach nie zakładających żadnych błędów. Jednak współcześnie teoria algorytmów rozproszonych, motywowana praktycznymi wyzwaniem, przed którymi stoją systemy, coraz częściej zakłada istnienie dodatkowych utrudnień dla algorytmu (takich jak ograniczone zasoby lub złośliwe działania, których modelem jest *adwersarz*). Takie dodatkowe utrudnienia wymagają od projektujących algorytmy większej ostrożności, oraz innego podejścia przy konstrukcji i analizie algorytmów w porównaniu z klasycznymi podejściami. W wymienionych pracach rozważamy kilka takich utrudnień pomimo których, jak udowodnimy, możliwe jest zaprojektowanie efektywnego algorytmu rozproszonego.

¹Rangi konferencji według the Computing Research and Education (CORE) Conference Portal: <http://portal.core.edu.au/conf-ranks/>

²Zródło: <https://www.internetlivestats.com>

4.3 Rozważane modele obliczeń rozproszonych

W pracach [H1–H7] analizujemy, wraz ze współautorami, klasyczne problemy obliczeń rozproszonych, takie jak rozgłaszanie (znane w anglojęzycznej literaturze jako *broadcast*), wybór lidera, czy eksploracja grafów. Problemy te są fundamentalne, to znaczy są wykorzystane jako podprocedury bardziej złożonych algorytmów. Efektywne, elastyczne i odporne rozwiązanie podstawowych problemów jest warunkiem koniecznym do konstrukcji bardziej rozbudowanych protokołów rozproszonych w konkretnych modelach. Myślą przewodnią całego cyklu prac jest konstruowanie odpornych algorytmów rozproszonych, czyli takich, które są w stanie działać efektywnie pomimo ograniczonych zasobów, czy niepewności w komunikacji. Konkretnie definicje tych ograniczeń i niepewności są motywowane faktycznymi zjawiskami występującymi w systemach rozproszonych. W kilku przypadkach udaje nam się pokazać, że odporność da się uzyskać bez negatywnego wpływu na złożoność czasową, co jest szczególnie pożądaną własnością.

Prace składające się na dzieło habilitacyjne zawierają:

1. konstrukcja i analiza algorytmów rozproszonych odpornych na losowe zaburzenia ([H1–H3]),
2. konstrukcja i analiza algorytmów rozproszonych odpornych na działania adwersarza ([H3, H4]),
3. konstrukcja i analiza algorytmów rozproszonych odpornych na ograniczenia zasobów ([H5, H6]),
4. analizę wielu równoległe działających błędów losowych w grafach ([H7]),
5. konstrukcja i analiza algorytmów w modelu sieci bezprzewodowych ([H3, H4, H6]),
6. analiza algorytmów oraz ograniczeń dolnych dla problemu eksploracji grafów ([H5, H7]).

We wszystkich pracach rozważamy model czasowy, w którym czas podzielony jest na dyskretne *rundy czasowe*. Zakładamy ponadto model synchroniczny, w którym wszystkie jednostki rozważanego modelu rozproszonego wykonują akcje w kolejnych rundach równocześnie i czas trwania każdej rundy jest taki sam dla wszystkich jednostek.

4.3.1 Model sieci radiowych

W dziedzinie obliczeń rozproszonych jest wiele modeli reprezentujących sieć urządzeń komunikujących się między sobą. Przykładami są modele LOCAL, CONGEST [Pel00], czy model sieci radiowych. Cechą wspólną tych wszystkich modeli jest lokalność obliczeń. Każda jednostka (wierzchołek sieci) wykonuje niezależne obliczenia i komunikuje się jedynie z sąsiadami w sieci. Podobne są również problemy rozważane w tych modelach. W wielu problemach punktem wyjścia jest problem łamania symetrii, czego przykładem jest wybór lidera i problemy komunikacyjne, jak na przykład rozgłaszanie.

W pojedynczej rundzie, wierzchołki mają możliwość komunikowania się jedynie z sąsiadami w sieci, więc we wszystkich problemach wymagających zebrania lub rozpropagowanie pewnej informacji po całej sieci, wymagane jest wykonanie wielu rund komunikacyjnych. Reguły, zgodnie z którymi przebiega komunikacja jest podstawową cechą rozróżniającą modele rozproszone w sieciach. W niektórych komunikacja jest ograniczona do wiadomości rozmiaru $O(\log n)$ (w sieciach o rozmiarze n), a w niektórych dostarczenie wiadomości zależy od energii, z jaką wykonywana jest transmisja (jak w sieciach SINR [GK00]).

Klasyczny model sieci radiowych [CK85] powstał jako reprezentacja zbioru urządzeń komunikujących się bezprzewodowo, w którym wiadomości nadawane w tym samym czasie do tego samego odbiorcy kolidują ze sobą. W tych sieciach mamy dany zbiór urządzeń komunikujących się w celu wykonania pewnego zadania (jak wybór lidera, czy rozgłaszanie). Sieć jest modelowana jako graf, w którym wierzchołki reprezentują stacje, a krawędzie łączą wierzchołki, które mogą się komunikować. Komunikacja między stacjami jest ograniczona na cztery sposoby:

1. **Ograniczenie odległości.** Nie każde dwie stacje mogą się bezpośrednio komunikować (czyli sieć komunikacyjna nie musi być kliką).
2. **Ograniczenie czasowe.** Aby transmisja była skutecznie odebrana, w sąsiedztwie odbiorcy musi nadawać dokładnie jeden nadający.
3. **Ograniczenie energetyczne.** Używanie radia do komunikacji wymaga energii, której każda stacja ma ograniczony zasób.

4. **Błędy komunikacyjne.** Transmisja może być nieudana z innych powodów (błąd sprzętowy, celowe złośliwe działanie, zmiana w topologii sieci).

Konstrukcje algorytmów odpornych na błędy komunikacyjne [CHHZ17, NR18, CHHZ19] i ograniczoną energię [CDH⁺18, CKP19, CDHP20] w sieciach radiowych jest w ostatnich latach obszarem bardzo intensywnych badań. Każde z ograniczeń 1-4 ma również istotne praktyczne motywacje.

W dużych sieciach bezprzewodowych, niektóre stacje są ulokowane zbyt daleko od siebie, aby mogły się bezpośrednio komunikować, więc takie sieci można (w pewnym uproszczeniu) modelować za pomocą grafów.

Kolizje między transmisjami występującymi blisko siebie w tym samym czasie, są problemem w powszechnie używanych sieciach (np. standardy IEEE 802.11 dla sieci WLAN). Bardzo duża liczba protokołów mających na celu unikanie kolizji, maksymalizujących przepustowość, gwarantujących dostarczenie wiadomości, została zaproponowana dla sieci bezprzewodowych w celu uzyskania efektywnej warstwy MAC (*medium-access control*).

Ograniczenia energetyczne są kluczowe na przykład w nowoczesnych sieciach Internetu Rzeczy (IoT), gdzie urządzenia są małe, często bez stałego połączenia z zasilaniem, więc oszczędne nadawanie może wydłużyć czas działania urządzeń w takiej sieci. Ponadto, duża liczba nadawań utrudnia ukrywanie informacji.

Odporność na różne błędy (wynikające z przypadku, czy celowego działania) jest kluczową cechą praktycznych systemów. W sieciach działających długo, urządzenia komunikacyjne mogą ulec awarii, całe sektory sieci mogą przestać działać ze względu na brak zasilania. Różne celowe ataki mające na celu zagłuszenia komunikacji (np. *denial of service*) również występują i wdrożone algorytmy powinny być na nie odporne. Przy projektowaniu algorytmów rozproszonych, mając na uwadze praktyczne zastosowanie, należy analizować je nie tylko pod kątem efektywności czasowej, ale również odporności na różne zagrożenia i utrudnienia. Podstawowe błędy i ataki analizowane w literaturze dotyczącej obliczeń rozproszonych to:

- *Błędy bizantyjskie* [LSP82], gdzie niektóre ze stacji występujących w sieci zostają przejęte przez adwersarza. Taka stacja może wysyłać dowolne wiadomości i celowo działać na niekorzyść algorytmu.
- *Crash fault* [FLP85], gdzie stacja może przestać działać. Taka stacja przestaje odpowiadać na wiadomości. Moment takiego wyłączenia jest zwykle pod kontrolą adwersarza.
- *Link failure* [LL90], gdzie niektóre z połączeń komunikacyjnych mogą w pewnym momencie przestać transmitować wiadomości.
- *Jammer* [ARS08] adwersarz, który może blokować kanał komunikacyjny w niektórych rundach.
- *Dual graph model* [KLN⁺10], gdzie niektóre z połączeń są kontrolowane przez adwersarza, który decyduje, czy w danej rundzie przenoszą wiadomości.

W sieciach radiowych rozważamy, między innymi, następujące problemy:

- *Globalne rozgłaszanie.* Na początku algorytmu, pojedyncza stacja posiada wiadomość, a celem jest rozpropagowanie tej wiadomości do całej sieci. Zwykle w literaturze pojęcie problem rozgłaszania (ang. *broadcast*) jest rozumiane jako problem globalnego rozgłaszania [Pe100].
- *Lokalne rozgłaszanie* [GHLN12]. Zakłada, że początkowo pewien podzbiór stacji ma wiadomości do przekazania. Polega na dostarczeniu co najmniej jednej wiadomości do każdej stacji, która ma sąsiada z wiadomością.
- *Wybór lidera.* Polega na ustanowieniu dokładnie jednej stacji jako lidera. Aby uznać problem wyboru lidera za rozwiązany, każda stacja musi znać tożsamość lidera.

W pracy [H3] zajmujemy się analizą algorytmu rozgłaszania w sieciach, w których występują ograniczenia 1,2,4, a utrudnienie polega na dynamicznych zmianach w topologii sieci. Rozważamy zarówno problem lokalnego rozgłaszania, jak i globalnego (warto zauważyć, że algorytm lokalnego rozgłaszania, może być używany jako podprocedura do rozwiązania globalnego rozgłaszania). Model dynamicznych zmian w sieci powstał jako modyfikacja modelu *dual graph*, w którym pewien podzbiór krawędzi jest

pod kontrolą adwersarza. Takie krawędzie, mogły być w danej rundzie, zależnie od decyzji adwersarza, *aktywne* (czyli mogły transmitować wiadomości, ale również powodować konflikty) lub *nieaktywne* (takie krawędzie były tymczasowo usuwane z grafu komunikacyjnego). Modyfikacja modelu dual graph w pracy [H3] polegała na ograniczeniu siły adwersarza, który nie mógł zmieniać stanu krawędzi w każdej rundzie, ale raz na τ rund modyfikował *rozkład prawdopodobieństwa*, z którego losowane były połączenia aktywne podczas poszczególnych rund.

W pracy [H6] zajmujemy się problemem rozgłaszania przy ograniczeniach 1, 2, 3. Zakładamy, że każda stacja zużywa jedną jednostkę energii na każdą próbę transmisji, a liczba jednostek energii jest ograniczona. Konstruujemy dwa algorytmy rozwiązujące rozgłaszanie przy ograniczonej energii. W pierwszym algorytmie, liczba jednostek energii dostępnej dla każdej stacji jest parametrem, a algorytm działa przy dowolnej jego wartości. Drugi algorytm działa w niemal optymalnym czasie i przy ograniczonym, sublogarytmicznym koszcie energetycznym.

W modelu w pracy [H4] występują ograniczenia 2, 4, a utrudnienie polega na występowaniu adwersarza typu jammer, blokującego niektóre transmisje. Przy takim adwersarzu, stacje mają za zadanie wybrać spośród siebie lidera. Trudność zadania w tym modelu, polega między innymi na tym, że adwersarz może blokować większość ze wszystkich rund. Dodatkowo stacje nie mają dostępu do wartości żadnych globalnych parametrów (jak na przykład, rozmiaru sieci czy parametrów określających możliwości adwersarza). Prezentujemy kilka algorytmów, a jeden z nich działa, przy pewnym ograniczeniu na parametry adwersarza, w czasie asymptotycznie identycznym do czasu wyboru lidera w modelu bez adwersarza.

4.3.2 Agenci w sieciach

W przeciwieństwie do modelu sieci radiowych, gdzie obliczenia są wykonywane przez wierzchołki, w modelu mobilnych agentów w sieciach, jednostki przeprowadzające obliczenia mogą odwiedzać wierzchołki i przechodzić przez krawędzie w grafie. Taki model jest reprezentacją na przykład robotów internetowych (*crawlers*), zbierających informacje o sieci stron w Internecie, czy fizycznych robotów (zakładając dyskretyzację środowiska, w którym zostały umieszczone). Jest to bardzo szeroki obszar badań, zawierający dużą liczbę modeli, których założenia zależą zwykle od właściwości agentów. Rozważane w literaturze modele zawierają, między innymi, zagadnienia takie jak:

- *Dostępność pamięci*. Pamięć w modelach agentowych może być umieszczona na wierzchołku lub na agencie. Pamięć na wierzchołku definiujemy jako pewną liczbę bitów na każdym wierzchołku, które agent może czytać i modyfikować przy każdej wizycie w tym wierzchołku. Pamięć na agencie jest zdefiniowana jako pamięć, którą agent może czytać i modyfikować w każdej rundzie.
- *Dostęp do losowości*. Błądzenie losowe jest użytecznym procesem przy rozwiązywaniu różnych problemów dla jednostek mobilnych, ze względu na prostotę implementacji i rozbudowaną teorię ściśle opisującą wiele kluczowych parametrów. W wariantach deterministycznych algorytmów dla agentów są czasem konstruowane jako derandomizacja błędzeń losowych [AKL⁺79, YWB03].
- *Komunikacja między agentami*. W przypadku wielu agentów, istotna jest pewna forma koordynacji, do czego niezbędna jest wymiana informacji. Najczęściej rozważa się modele z *komunikacją lokalną*, wyłącznie między agentami zlokalizowanymi w tej samej pozycji lub z *komunikacją globalną*, gdzie dowolna para agentów może się komunikować niezależnie od pozycji [DDK⁺15].

Dwa najczęściej rozważane problemy w tej dziedzinie:

- *Eksploracja*, gdzie agenci mają za zadanie odwiedzić co najmniej raz, całe, początkowo nieznanne środowisko. Formalnie odpowiada to odwiedzeniu wszystkich wierzchołków grafu.
- *Gathering*, gdzie wszyscy agenci muszą zebrać się w jednym wierzchołku grafu.

W przypadku, gdy wiele agentów jest umieszczonych w grafie, w celu wykonania pewnego zadania, zakładamy, że ci agenci wykonują zadanie równolegle. Czas algorytmu jest w modelu synchronicznym najczęściej definiowany jako liczba rund, gdzie podczas każdej rundy każdy z agentów może przejść jedną krawędź i odwiedzić jeden wierzchołek. Przy konstruowaniu algorytmów eksploracji przy użyciu wielu agentów oczekujemy, że czas działania będzie istotnie niższy niż przy eksploracji pojedynczym agentem.

Dwa możliwe podejścia do zdefiniowania modelu agentów mogących zrealizować eksplorację dowolnych grafów, jest użycie pamięci lub użycie losowości, gdyż bezpamięciowy agent deterministyczny, bez pamięci na wierzchołku, nie ma możliwości użycia różnych krawędzi wychodzących podczas dwóch wizyt w dowolnym wierzchołku, co jest koniecznym warunkiem eksploracji nawet prostych grafów.

W pracy [H5] pokazujemy granice dolne na czas eksploracji pojedynczym agentem w modelu deterministycznym z pamięcią na wierzchołku. Ta granica pokazuje, że w modelu z wyłącznie pamięcią na wierzchołku (agent bezstanowy) nie jest możliwe skonstruowanie algorytmu szybszego, niż prosty algorytm round-robin, w tym modelu znany jako rotor-router i rozważany jako algorytm eksploracji grafów między innymi w [YWB03]. W ten sposób pokazujemy, że prosty, oszczędny pamięciowo algorytm, który jest również znany ze swojej odporności na zaburzenia [BGH⁺17], jest czasowo optymalny w klasie algorytmów, w których agenci mogą korzystać jedynie z pamięci na wierzchołkach.

W pracy [H7] rozważamy model zrandomizowany przy wielu agentach bezpamięciowych, wykonujących niezależne błądzenia losowe. Jest to popularna metoda eksploracji grafów ze względu na brak koniecznej pamięci i odporność na adversarialne zmiany w topologii grafu [AKL18]. W pracy [H7] pokazujemy dokładne ograniczenia na czas eksploracji wielu błędzeń dla ścieżki i wielowymiarowych krat.

4.3.3 Kolonie owadów społecznych (*social insect colonies*)

Niektóre gatunki owadów (jak mrówki, termity, pszczoły) rozwiązują wiele problemów jako społeczność, w sposób rozproszony, bez centralnej kontroli zarządzającej działaniami poszczególnych jednostek. Z tego powodu techniki i narzędzia z teorii obliczeń rozproszonych mogą służyć wyjaśnieniu powodów pewnych zachowań tych owadów, ustaleniu teoretycznych granic efektywności takich społeczności, czy nawet znalezieniu dokładnego algorytmu używanego przez jednostki. Zwykle w tych modelach nie wymagamy znalezienia rozwiązania globalnie optymalnego, tylko takiego, które jest możliwe do osiągnięcia poprzez lokalne optymalizacje poszczególnych jednostek określanych jako robotnicy (*workers*). Konkretnie sformułowanie założeń modelu zależy zwykle od zachowań obserwowanych na pewnych gatunkach owadów. Przykładowo w problemie szacowania rozmiaru przyjmuje się, że owady szacują liczebność na podstawie częstości spotkań [MSL17], co jest inspirowane zachowaniem mrówek *Temnothorax albipennis* [Pra05].

Alokacja zadań w koloniach owadów społecznych jest procesem przydziału robotników do różnych zadań (np. zbieranie pożywienia, karmienie młodych, obrona gniazda) w sposób optymalny, czyli maksymalizujący szansę na sukces całej społeczności. Każdy robotnik jest w stanie wykonywać każde z zadań, ale w każdej chwili może wykonywać tylko jedno. Robotnik w dowolnym momencie, pod wpływem różnych czynników, może zmienić swój przydział do zadania. Każde zadanie ma ustalone zapotrzebowanie, czyli liczbę robotników, która powinna to zadanie wykonywać. Robotnicy w takich koloniach nie mają bezpośredniej świadomości tego zapotrzebowania i nie są w stanie dokładnie ocenić liczby robotników pracujących w danej chwili nad danym zadaniem [GM99]. Ponadto, w koloniach owadów nie ma żadnej centralnej jednostki koordynującej działania robotników [Gor96]. Zatem alokacja musi być wykonywana przez każdą z mrówek niezależnie, bazując jedynie na lokalnych obserwacjach, informacji zwrotnej od środowiska i ograniczonej komunikacji z innymi robotnikami. Ta informacja zwrotna (bodziec) polega na wycuciu, na przykład zbyt wysokiej temperatury w gnieździe, spostrzeżeniu światła wpadającego przez otwór w ścianie gniazda lub wycuciu feromonów produkowanych przez głodne larwy. Pomimo używania ograniczonej komunikacji, jedynie lokalnych obserwacji obarczonych niepewnością, wiele gatunków owadów znakomicie radzi sobie z alokacją zadań. W pracy [H1] zajmujemy się odpowiedzią na pytania, jak mrówki mogą przeprowadzać alokację zadań w modelu z niepewnymi obserwacjami.

4.3.4 Protokoły populacyjne

Kolejnym modelem obliczeń rozproszonych, rozważanym w pracach składających się na dzieło habilitacyjne, są protokoły populacyjne, wprowadzone po raz pierwszy w pracy [AAD⁺06]. W tym modelu, agenci są prostymi jednostkami, zwykle o bardzo małej pamięci, które wchodzi między sobą w interakcje. W pierwszej pracy o protokołach populacyjnych [AAD⁺06] rozważane były protokoły z agentami posiadającymi jedynie stałą liczbę bitów pamięci, jednak kolejne prace rozważały również rozmiar pamięci jako funkcję rosnącą wraz z liczbą agentów n . Interakcje przebiegają w kolejnych rundach między parami jednostek wybieranymi losowo. Jednostki podczas interakcji komunikują się ze sobą i mogą zmienić swój stan pamięci. Najczęściej rozważane problemy w tym modelu to:

- *Wybór lidera*, gdzie, podobnie jak w sieciach radiowych celem jest oznaczenie dokładnie jednego agenta jako lidera i pozostałych jako nie-liderów.
- *Majority*, gdzie początkowo każda ze stacji ma pewną wartość w pamięci i celem jest ustalenie, która z wartości pojawia się najczęściej w populacji.

Ten model znalazł interesujące zastosowanie między innymi w analizie reakcji chemicznych [CCDS17], gdzie jednostki reprezentują cząsteczki w dobrze wymieszanym roztworze, a interakcje reprezentują reakcje między cząsteczkami.

W pracy [H2] rozważamy model, w którym interakcje między jednostkami w populacji podlegają losowemu zaburzeniu. Początkowo, każdy z agentów posiada w pamięci pewną wartość i celem procesu jest obliczenie średniej ze wszystkich wartości (lub liczby całkowitej będącej jak najbliższej średniej). Ten problem znajduje zastosowanie na przykład przy obliczaniu średniej wartości temperatury przez zbiór sensorów, czy uzgodnieniu oceny, na przykład jakości potencjalnego nowego miejsca na gniazdo, w modelu owadów społecznych. Definicja rozważanego przez nas algorytmu jest prosta: *Agenci przy spotkaniu wymieniają się swoimi wartościami i obliczają średnią, którą zapisują w pamięci*. Dodatkowym utrudnieniem są losowe błędy, które mogą przekłamać wartości podczas komunikacji między agentami. W pracy [H2] pokazujemy, pod jakimi warunkami wartości pamiętane przez agentów zbiegają do szukanej średniej i jak zależy to od parametrów losowego szumu w komunikacji.

4.4 Wyniki

W tej sekcji omówimy najważniejsze wyniki z prac wchodzących w skład dzieła habilitacyjnego.

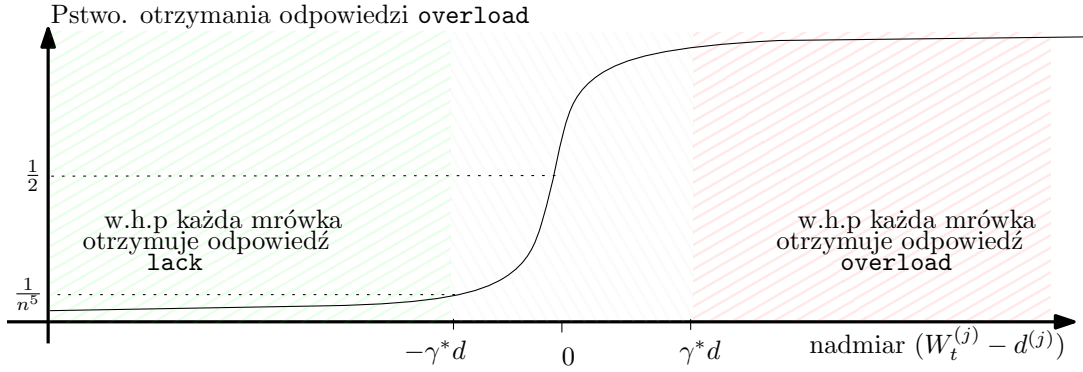
4.4.1 Praca H1

Problem alokacji zadań w koloniach owadów, rozważany w pracy [H1], definiujemy w następujący sposób. Mamy dany zbiór n robotników i k zadań. Każde z zadań j , ma ustaloną całkowitoliczbową wartość zapotrzebowania $d^{(j)}$. Czas jest podzielony na dyskretne jednostki, w każdej z których, każdy z robotników może pracować jednym, dowolnym zadaniem. Zakładamy, że jeden robotnik zaspokaja jedną jednostkę zapotrzebowania pracując nad zadaniem w danej rundzie. Informacja zwrotna jest funkcją binarną zależną od różnicy między zapotrzebowaniem $d^{(j)}$, a liczbą robotników wykonujących zadanie w danej chwili $W_t^{(j)}$. Wartość funkcji informacji zwrotnej jest przekazywana do robotników. Zakładamy, że w każdej rundzie, każdy z robotników może otrzymać wartość informacji zwrotnej dla jednego, wybranego zadania. Celem całej kolonii jest aby wraz z t , wektor $(W_t^{(1)}, W_t^{(2)}, \dots, W_t^{(k)})$ dążył do wektora zapotrzebowania $(d^{(1)}, d^{(2)}, \dots, d^{(k)})$. Zakładamy, że sytuacją niepożądaną jest zarówno zbyt mała jak i zbyt duża, w porównaniu do zapotrzebowania, liczba robotników pracujących nad zadaniem.

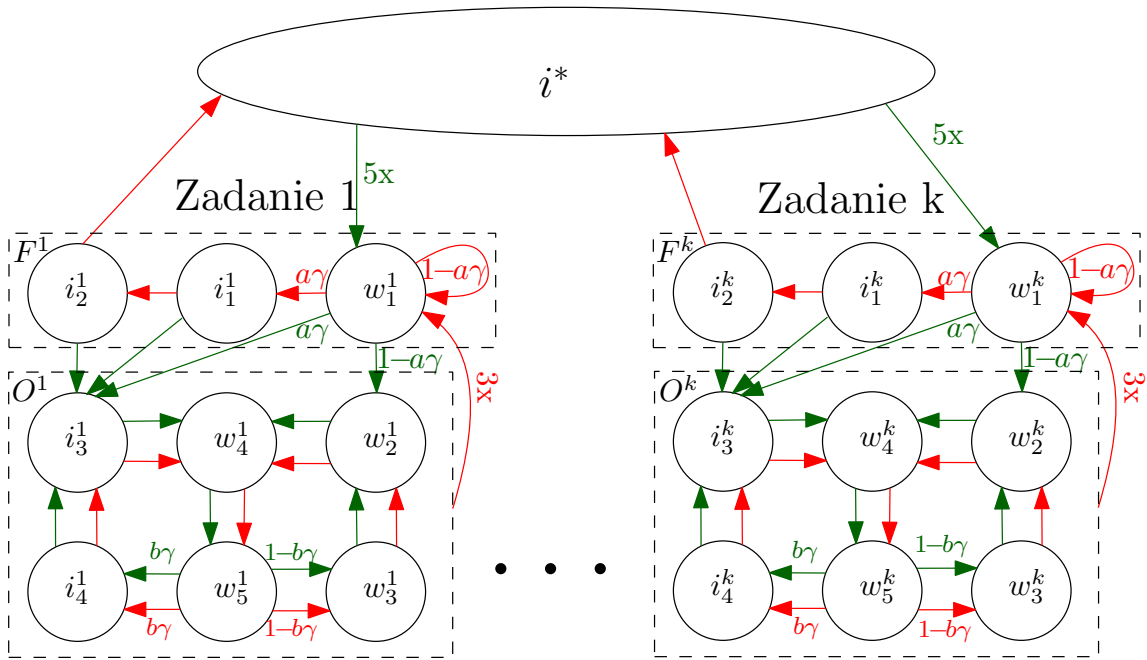
Taki sam problem był rozważany w pracy [CDLN14], gdzie autorzy przyjęli, że ta informacja zwrotna (odpowieź) jest dokładna, czyli w każdej rundzie t zwraca 0 (**lack**) lub 1 (**overload**), w zależności od tego, czy $d^{(j)} - W_t^{(j)}$ jest dodatnie, czy nie. Taka informacja odpowiada dokładnemu stwierdzeniu, czy liczba robotników pracujących nad zadaniem jest za duża, czy za mała. Autorzy w [CDLN14] zaprezentowali algorytm alokacji zadań działający szybko (w czasie proporcjonalnym do $\log n$) i w którym alokacja robotników do zadań zbiegała do dowolnego wektora zapotrzebowań $(d^{(1)}, d^{(2)}, \dots, d^{(k)})$.

Celem pracy [H1] jest rozważenie bardziej realistycznej funkcji informacji zwrotnej. Przyjeliśmy, że taka funkcja powinna zwracać dokładne wartości, gdy liczba robotników $W_t^{(j)}$ mocno odbiega od zapotrzebowania $d^{(j)}$, i mocno nieprzewidywalne (losowe) wartości, gdy $W_t^{(j)}$ jest blisko $d^{(j)}$. Zaproponowaliśmy taką losową funkcję zwracającą 0 (**lack**) lub 1 (**overload**) z prawdopodobieństwem zależnym od funkcji *sigmoid* z różnicy $W_t^{(j)} - d^{(j)}$ (patrz Rysunek 1).

Następnie zaproponowaliśmy miarę jakości algorytmu działającego przy niepewnej informacji zwrotnej. Ze względu na niepewność, nie można już wymagać, żeby zapotrzebowanie było dokładnie zaspokojone, zatem przyjęliśmy, że alokacja $W_t^{(j)}$ jest ε -dokładna, gdy odbiega $|W_t^{(j)} - d^{(j)}| \leq \varepsilon d^{(j)}$ dla pewnej, małej wartości $\varepsilon > 0$. Zdefiniowaliśmy, że algorytm działający w czasie T , jest ε -dokładny, jeżeli alokacja jest ε -dokładna dla *wszystkich zadań* w $T - o(T)$ rundach. Uznaliśmy, że w realistycznych scenariuszach niewielka rozbieżność między zapotrzebowaniem, a alokacją nie wpływa negatywnie na jakość wykonania danego zadania. Natomiast rundy, w których rozbieżność jest duża muszą być nieliczne. Głównym wynikiem w pracy jest 15γ -dokładny algorytm alokacji, dla parametru γ zależnego od



Rysunek 1: Wykres prawdopodobieństwa zwrócenia przez funkcję informacyjnej zwrotnej wartości **overload**, w zależności od $W_t^{(j)} - d^{(j)}$. Gdy nadmiar robotników w zadaniu jest w zielonym (czerwonym) regionie, wszystkie mrówki, które otrzymują odpowiedź od tego zadania, otrzymają odpowiedź **lack** (**overload**). Gdy nadmiar jest w szarym regionie, im bliżej $W_t^{(j)}$ jest do $d^{(j)}$, tym bardziej informacja zwrotna jest nieprzewidywalna.



Rysunek 2: Ilustracja maszyny stanów w algorytmie [H1] Section 4]. Czerwone strzałki oznaczają przejścia przy otrzymaniu informacji **overload**, a zielone są trawersowane przez robotników po otrzymaniu odpowiedzi **lack**. Wartości ponad strzałkami (np. $a\gamma$ i $1 - b\gamma$) oznaczają prawdopodobieństwa przejść. Oznaczenia $3x$ (i $5x$) mówią, że aby robotnik wykonał dane przejście między stanami, odpowiadającą informację zwrotną musi otrzymać 3 (lub 5) razy z rzędu. Przejścia z $3x$ mają wyższy priorytet (czyli są wybierane niezależnie od innych, po otrzymaniu danej informacji zwrotnej 3 razy z rzędu). Stany oznaczone literą i oznaczają stany w których robotnicy są przypisani do tego zadania i obserwują informację zwrotną od tego zadania, ale nie pracują nad nim. Stany oznaczone literą w oznaczają stany w których robotnicy pracują nad danym zadaniem. Robotnicy w stanie i^* nie pracują nad żadnym zadaniem ale każdy z nich obserwuje informację zwrotną od jednego, losowego zadania i może dołączyć, jeżeli pięciokrotnie otrzyma odpowiedź **lack**.

dokładności funkcji informacji zwrotnej [H1, Theorem 3.2]. Dodatkowo pokazujemy ograniczenie dolne mówiące, że żaden algorytm używający mniej niż $\lfloor \ln(1/(16\lambda)) \rfloor$ bitów pamięci na robotnika, nie może być $2\lambda\gamma$ -dokładny [H1, Theorem 3.5].

Nasz algorytm bazuje na następującej obserwacji: w przypadku, gdy alokacja robotników do zadań jest dokładna, to wtedy informacja zwrotna jest mocno nieprzewidywalna. Natomiast, gdy alokacja przewyższa prawidłową o pewną frakcję wynoszącą w przybliżeniu $(1 + \gamma)$, to informacja zwrotna z dużym prawdopodobieństwem zwróci tę samą wartość do wszystkich robotników obserwujących to zadanie. Podobnie w przypadku niedoboru robotników w zadaniu o frakcję $(1 - \gamma)$. Bazując na powyższej obserwacji, nasz algorytm uzyskuje *kontrolowane oscylacje* w liczbie robotników w zadaniu w celu uzyskania stabilnej alokacji. Oczywiście oscylacje nie mogą być zbyt duże, aby alokacja była bliska docelowej. Uzyskanie takiego efektu nie jest proste, gdyż każdy robotnik podejmuje niezależne decyzje bazując jedynie na pewnej małej pamięci i informacji zwrotnej od jednego, obserwowanego zadania.

Nasz projekt algorytmu jest maszyną stanów składającą się ze stanu i^* (robotnicy bez zadania) i k maszyn (po jednej dla każdego zadania). Maszyna dla zadania i składa się z dwóch komponentów: F^i służy do odfiltrowania odpowiedniej liczby robotników, a O^i zapewnia utrzymanie odpowiedniej liczby robotników pracujących w zadaniu (patrz Rysunek 2). Każdy z robotników jest niezależną maszyną i pamięta w swojej pamięci swój stan.

4.4.2 Praca H2

W pracy [H2] analizujemy dynamikę prostego protokołu populacyjnego polegającego na uśrednianiu. Początkowo mamy n agentów i każdy agent ma w pamięci pojedynczą wartość. Przy każdym spotkaniu, dwaj agenci wysyłają do siebie swoje aktualne wartości. Komunikacja jest obciążona zaburzeniami, więc jeżeli agent przekazuje wartość v , to do odbiorcy dociera $v + N$, gdzie N jest zmienną losową wybraną z pewnego rozkładu \mathcal{N} o zerowej wartości oczekiwanej. Agent następnie oblicza średnią między wartością w pamięci, a wartością otrzymaną i nową wartość zapisuje w pamięci. Warto zauważyć, że ze względu na zaburzenia, nowe wartości obliczone przez obu agentów po spotkaniu, niekoniecznie są takie same.

Rozważamy dwa warianty spotkań między agentami:

1. W *modelu sekwencyjnym*, w każdej rundzie spotyka się jedną, losowo wybraną, parę agentów.
2. W *modelu współbieżnym*, w każdej rundzie wybierane jest losowe doskonałe skojarzenie między agentami i następuje równoczesne spotkanie między $n/2$ parami agentów, zgodnie ze skojarzeniem.

Wartości agentów w rundzie t oznaczamy przez $X_1^{(t)}, X_2^{(t)}, \dots, X_n^{(t)}$, a bieżącą średnią w rundzie t , oznaczamy przez $\lambda^{(t)} = \sum_{i=1}^n X_i^{(t)} / n$. Celem całego procesu jest obliczenie przez wszystkich agentów wartości możliwie jak najbliższej początkowej średniej $\lambda^{(0)}$. Rozważamy dwie miary zbieżności:

- suma kwadratów odległości od początkowej średniej: $TSS(t) = \sum_i \left(X_i^{(t)} - \lambda^{(0)} \right)^2$,
- suma kwadratów odległości od bieżącej średniej $\bar{\phi}(t) = \sum_i \left(X_i^{(t)} - \lambda^{(t)} \right)^2$.

Praca [H2] zawiera następujące wyniki:

1. Przy pewnych słabych założeniach na rozkład zmiennej N , udowadniamy ograniczenia na wartość $\bar{\phi}(t)$, co pokazuje jak proces zbiega do bieżącej średniej $\lambda^{(t)}$. Nasze ograniczenia są optymalne z dokładnością do multiplikatywnej stałej. Udowadniamy, że $\bar{\phi}(t)$ asymptotycznie zbiega do iloczynu n i drugiego momentu $\mathbf{E}[N^2]$ [H2, Theorem 1].
2. Pokazujemy, że jest możliwa zbieżność do aktualnej średniej, przy ograniczeniu na liczbę rund. Udowadniamy, że przy $O(n^2)$ rundach (w modelu sekwencyjnym) wartość $TSS(t)$ można ograniczyć podobnie jak $\bar{\phi}(t)$. Natomiast przy większej liczbie rund $\Omega(n^2)$, wartość $TSS(t)$ ze stałym prawdopodobieństwem istotnie odbiega od $\bar{\phi}(t)$ [H2, Corollary 2].
3. Rozważmy następnie wariant całkowitoliczbowy problemu. Zakładamy, że wtedy każdy z agentów może zapamiętywać jedynie liczby całkowite. Jeżeli przy uśrednianiu, agent otrzymuje liczbę niecałkowitą, to dokonuje on losowego zaokrąglenia do jednej z dwóch sąsiednich liczb całkowitych. W tym wariantcie otrzymujemy dokładnie analogiczne gwarancje zbieżności, jak w punktach 1. i 2. [H2, Corollary 3].

4. W modelu współbieżnym również udowadniamy gwarancje zbieżności takie, jak w punktach 1. i 2. z jedyną różnicą polegającą na przemnożeniu liczby rund w tych wynikach przez czynnik $2/n$. [H2, Corollary 4].
5. Ostatnim wynikiem jest eksperymentalna analiza przypadku, w którym agenci mogą pamiętać jedynie wartości z pewnego przedziału (np. jedynie wartości od 1 do 10). Uzyskujemy wtedy istotnie szybszą rozbieżność między średnią początkową, a średnią bieżącą. To sugeruje, że w tym modelu, wynik taki jak w punkcie 2. nie zachodzi. [H2, Section 5].

4.4.3 Praca H3

W pracy [H3] rozważamy wariant modelu *dual graph* w sieciach radiowych. W rozpatrywanym modelu zakładamy, że topologia sieci jest pod częściową kontrolą adwersarza. W modelu *dual graph* zakłada się, że sieć komunikacyjna jest reprezentowana przez parę grafów $G = (V, E)$, $G' = (V, E')$. Oba grafy mają ten sam zbiór wierzchołków V , natomiast zbiór krawędzi E' jest nadzbiorem E . Krawędzie z E reprezentują pewne, niezawodne kanały komunikacji, które nie ulegną awarii w żadnym momencie działania algorytmu. Oznacza to, że każda krawędź z E jest aktywna w każdej rundzie. Aktywne krawędzie mogą transmitować wiadomości, ale również powodować kolizje. Natomiast zbiór $E' \setminus E$ zawiera niepewne krawędzie. Każda krawędź z $E' \setminus E$ może w każdej rundzie być aktywna lub nieaktywna. W klasycznym wariacie, adwersarz decyduje w każdej rundzie, które z niepewnych krawędzi będą aktywne, więc topologia sieci komunikacyjnej może być istotnie inna z rundy na rundę. Zakładamy, że graf G jest spójny, aby możliwe było poprawne rozwiązanie problemu globalnego rozgłaszania nawet w przypadku, gdy adwersarz wyłączy na stałe wszystkie krawędzie z $E' \setminus E$.

Model komunikacyjny zakłada brak detekcji kolizji. Oznacza to, że nadawanie jest skuteczne tylko, gdy odbiorca sąsiaduje z dokładnie jedną transmitującą stacją. W pozostałych przypadkach (zero lub więcej niż jeden transmitujący sąsiad), odbiorca otrzymuje zawsze taką samą odpowiedź od kanału komunikacyjnego.

Istniejące prace [KLN⁺10, GLN13, LN15] pokazują, że efektywne rozwiązanie standardowego problemu rozgłaszania nie jest możliwe w modelu *dual graph* bez dodatkowych silnych założeń. Natomiast w prawdziwych sieciach radiowych, które podlegają dynamicznym zmianom topologii reprezentowanym w modelu *dual graph*, proste strategie typu *backoff* działają na tyle efektywnie, że są stosowane w praktyce. Pojawiający się paradoks, w którym problemy rozwiązywane prostymi metodami w praktyce, są niemożliwe w abstrakcyjnych modelach zasugerował nam, że abstrakcyjny model adwersarza w modelu *dual graph* jest zbyt silny i nie jest adekwatną reprezentacją dla zmian topologii w rzeczywistych sieciach. W pracy [H3] proponujemy słabszy, i przez to bardziej realistyczny model adwersarza, a następnie konstruujemy dla takiego modelu efektywne algorytmy dla problemu rozgłaszania.

Zaproponowany adwersarz opiera się na następujących dwóch założeniach:

1. Zmiany, które występują często (np. pomiędzy kolejnymi rundami) są z natury losowe. Takie zmiany odpowiadają na przykład ruchom użytkowników, czy losowym fluktuacjom w propagacji sygnału.
2. Zmiany rzadkie (takie, które zdarzają się raz na pewną liczbę rund) mogą być adwersarialne. Takie zmiany odpowiadają na przykład rekonfiguracji sieci, kataklizmom, awariom zasilania, czy innym gwałtownym zdarzeniom mogącym zaburzyć działanie sieci.

Zgodnie powyższymi założeniami zaproponowaliśmy model adwersarza z parametrem τ , który raz na τ rund może zredefiniować rozkład prawdopodobieństwa, według którego w kolejnych rundach są losowane krawędzie ze zbioru $E' \setminus E$. Wylosowane krawędzie są dodawane do sieci jako aktywne, a pozostałe pozostają nieaktywne. Jeżeli rozkład prawdopodobieństwa nie jest zmieniony, to w kolejnych rundach zmiany w topologii sieci są losowe. Natomiast raz na τ rund, adwersarz może wyznaczyć zupełnie nowy rozkład prawdopodobieństwa na podzbiorach $E' \setminus E$.

W takim modelu udowadniamy, że czas $\mathcal{O}(\Delta^{1/\tau} \tau^2 \log n / \log \Delta)$ jest wystarczający do rozwiązania z dużym prawdopodobieństwem³ problemu lokalnego rozgłaszania w sieciach rozmiaru n , o maksymalnym stopniu wierzchołka Δ . Nasz algorytm bazuje na klasycznym podejściu *exponential backoff*, gdzie rundy są pogrupowane w cykle o pewnej długości t i w i -tej rundzie cyklu każda ze stacji, która ma wiadomość do przekazania, transmituje z prawdopodobieństwem p^i , dla pewnej wartości $p < 1$. W pracy [H3]

³Z dużym prawdopodobieństwem oznacza z prawdopodobieństwem co najmniej $1 - 1/n$.

znajdujemy odpowiednie wartości parametrów p i t (w zależności od Δ i τ) i dowodzimy, że otrzymany algorytm wykazuje się odpornością na wyżej zdefiniowanego adwersarza [H3, Theorem 6]. Pokazaliśmy również, że czas $\Omega(\Delta^{1/\tau}\tau^2/\log\Delta)$ jest wymagany, jeżeli $\tau = \mathcal{O}(\log\Delta/\log\log\Delta)$ [H3, Theorem 8]. Korzystając z wyników dla rozgłaszania lokalnego skonstruowaliśmy również ograniczenie górne i dolne na czas algorytmu globalnego rozgłaszania [H3, Theorem 9, Theorem 10].

W pracy [H3] analizujemy również inny wariant zmiennej topologii: (l, τ) -adwersarza. Taki adwersarz, ma możliwość w każdej rundzie zmiany stanu nie więcej niż l -krawędzi ze zbioru $E' \setminus E$ w bezpośrednim otoczeniu każdego wierzchołka, Natomiast raz na τ rund adwersarz ma możliwość zmiany stanu wszystkich krawędzi z $E' \setminus E$. Dla takiego adwersarza pokazaliśmy, że dla $l = \mathcal{O}(\Delta^{1/\tau})$ istnieje algorytm lokalnego rozgłaszania o czasie działania $\mathcal{O}(\Delta^{1/\tau}\tau\log(1/\varepsilon))$ z prawdopodobieństwem $1 - \varepsilon$ [H3, Theorem 11].

4.4.4 Praca H4

W pracy [H4] rozważamy problem wyboru lidera w modelu single-hop. W tym modelu, sieć komunikacyjna jest kliką, zatem możliwa jest komunikacja między dowolnymi dwiema stacjami, ale jednocześnie nadawanie dowolnych dwóch stacji powoduje kolizje. W tej pracy rozważamy model komunikacyjny z detekcją kolizji, gdzie cisza i kolizja są rozróżnialne przez stacje. Wspólny kanał komunikacyjny może w każdej rundzie być w tym modelu w jednym z trzech stanów: Null (żadna stacja nie nadaje), Single (dokładnie jedna stacja nadaje) i Collision (co najmniej dwie stacje nadają).

Dodatkowymi utrudnieniami, które sprawiały, że model jest bardziej realistyczny, jest brak znajomości globalnych parametrów (takich jak rozmiar sieci n) i występowanie $(T, 1 - \varepsilon)$ -adwersarza, definiowanego w następujący sposób. Taki adwersarz ma możliwość blokowania rund czasowych, gdzie zablokowana runda jest w stanie nierozróżnialnym od stanu Collision niezależnie od liczby stacji transmitujących. Liczba rund blokowanych przez adwersarza w każdym oknie czasowym długości $w \geq T$, jest ograniczona przez $(1 - \varepsilon)w$. Taki adwersarz jest rozważany w literaturze pod nazwą *jammer* [ARS08, BKL⁺13].

Pomimo, że łamanie symetrii w tym modelu jest możliwe jedynie przy pomocy rund czasowych, w których występuje Single, a adwersarz ma możliwość zablokowania zdecydowanej większości rund, w [H4] pokazujemy, że jest możliwy efektywny wybór lidera. Dla dowolnego stałego ε i dla $T = \mathcal{O}(\log n)$ skonstruowaliśmy algorytm działający z dużym prawdopodobieństwem w czasie $\mathcal{O}(\log n)$ [H4, Theorem 2.6]. Co ciekawe, czas $\Omega(\log n)$ dla algorytmów działających z dużym prawdopodobieństwem jest wymagany nawet bez adwersarza i nawet przy znanym n .

Dla dowolnych $\varepsilon > 0, T$ znaleźliśmy algorytm działający w czasie $\mathcal{O}\left(\frac{\log(1/\varepsilon)}{\varepsilon^3} \log n\right)$ jeżeli $T \leq \frac{\log n}{\varepsilon^3 \log(1/\varepsilon)}$ i $\mathcal{O}\left(\max\left\{\log\log\left(\frac{T}{\varepsilon \log n}\right), \log(1/\varepsilon) \log\log(1/\varepsilon)\right\} \cdot T\right)$ w przeciwnym przypadku [H4, Theorem 2.9]. Dodatkowo pokazaliśmy, że czas $\Omega(\max\{\log n/\varepsilon, T\})$ jest ograniczeniem dolnym na czas działania każdego algorytmu działającego w tym modelu z dużym prawdopodobieństwem sukcesu [H4, Lemma 2.7].

W algorytmie wyboru lidera w tym modelu wystarczy uzyskać jedną rundę w stanie Single, aby dokonać wyboru lidera. Wtedy liderem zostaje stacja, której udaje się nadać, przy założeniu, że stacja nadająca może jednocześnie słuchać i jest świadoma swojej ewentualnej udanej transmisji (to założenie nazywamy Full-Duplex albo strong-CD). Podejściem, jakie zastosowaliśmy w pracy, była aproksymacja łącznej liczby stacji n . Łatwo zauważyć, że (bez adwersarza) jeżeli wszystkie n stacje nadaje niezależnie z prawdopodobieństwem p , to dla $p \ll 1/n$ z dużym prawdopodobieństwem kanał będzie w stanie Null. Natomiast, jeżeli stacje nadają z $p \gg 1/n$, to z dużym prawdopodobieństwem stan kanału będzie Collision. Nadawanie z prawdopodobieństwem $1/n$ skutkuje wynikiem Single ze stałym prawdopodobieństwem.

Bazując na tych obserwacjach możemy zwiększać lub zmniejszać naszą aproksymację, w zależności od odpowiedzi od kanału, tak aby zbliżyć się do poprawnej wartości n . Kluczowym przy konstrukcji naszych algorytmów jest to, że każda runda, w której interweniuje adwersarz, jest odbierana przez stacje jako Collision, zatem Null jest zawsze wiarygodny. Korzystając z tego, w naszym algorytmie aproksymującym n , redukujemy istotność odpowiedzi Collision, aby adwersarz (który może spowodować Collision w bardzo dużej frakcji rund czasowych) nie miał możliwości zwiększania naszej aproksymacji w nieskończoność. Odpowiednia wartość tej redukcji zależy od siły adwersarza (parametru ε). W ten sposób konstruujemy pierwszy algorytm w pracy [H4]. Kolejny algorytm pozbywa się założenia strong-CD i działa w modelu weak-CD, gdzie stacja nie może jednocześnie transmitować i odbierać transmisji. Ostatni algorytm usuwa

założenie o znajomości ε przez aproksymację tego parametru. Praca [H4] uzyskuje przy tych samych założeniach lepszą złożoność czasową od istniejących rozwiązań [ARS08, RSSZ11, ARS⁺14].

4.4.5 Praca H5

W pracy [H5] pokazujemy dwie granice dolne na czas deterministycznej eksploracji grafów w modelu z pamięcią wyłącznie na wierzchołkach. Rozważane grafy są anonimowe (brak identyfikatorów na wierzchołkach), nieskierowane i mają poetykietowane porty. Ostatnia własność modelu polega na tym, że w każdym wierzchołku o stopniu d , krawędzie wychodzące z tego wierzchołka są ponumerowane $\{1, 2, \dots, d\}$. Dodatkowo, w tej pracy zakładamy, że agent wchodząc do wierzchołka, nie jest świadomy portu którym wszedł. W takim modelu (deterministyczny, bez pamięci na agencie) łatwo zauważyć, że w każdym wierzchołku pamięć $\log_2 d$ jest konieczna, żeby algorytm odwiedził wszystkie wierzchołki dowolnego grafu. Mając taką liczbę bitów pamięci, możliwe jest zaimplementowanie algorytmu rotor-router⁴, którego czas eksploracji grafów o średnicy D i liczbie krawędzi m jest $\mathcal{O}(mD)$ [YWB03].

W pracy [H5] rozważamy model, w którym agent ma nieograniczoną pamięć na każdym z wierzchołków. Pokazujemy, że w tym modelu dowolny algorytm potrzebuje w najgorszym przypadku $(n-1)^2$ rund, żeby wyeksplorować ścieżkę długości n . Dokładnie taka sama liczba rund jest potrzebna w najgorszym przypadku dla algorytmu rotor-router, więc pokazujemy, że dodatkowa pamięć nie daje w tym przypadku żadnego przyspieszenia. Drugim wynikiem pracy jest dowód, że dla dowolnego algorytmu z nieograniczoną pamięcią na wierzchołku istnieje graf w którym ten algorytm potrzebuje $\Omega(mD)$ rund, aby ukończyć eksplorację. Ten wynik pokazuje, że w modelu z pamięcią wyłącznie na wierzchołkach, algorytm, który używa minimalnej możliwej pamięci, jest asymptotycznie optymalny czasowo.

4.4.6 Praca H6

W pracy [H6] zajmujemy się problemem rozgłaszania w sieciach radiowych przy ograniczonej liczbie nadawców. Zakładamy, że każda ze stacji w sieci ma prawo nadać maksymalnie $\mathcal{O}(\varphi)$ -krotnie podczas całego algorytmu (taki wariant jest też znany pod nazwą *k-shot broadcast*⁵ [KP16]).

W tym problemie, zakładamy model bez detekcji kolizji, więc aby odbiorca uzyskał jakąkolwiek informację, w jego otoczeniu musi nadawać dokładnie jedna stacja. Typowe rozwiązania problemu rozgłaszania w tym modelu opierają się na podejściach typu *uniform*, gdzie każda ze stacji, która ma wiadomość do nadania, nadaje niezależnie z pewnym prawdopodobieństwem (np. algorytmy w pracy [H3] są tego typu). Analiza takich algorytmów jest ułatwiona ze względu na niezależność transmisji między rundami. Niestety tutaj podejścia tego typu nie działają z dwóch powodów. Po pierwsze prawdopodobieństwo skutecznej transmisji w algorytmach uniform jest zawsze co najwyżej pewną stałą (około $1/e$), więc aby uzyskać duże prawdopodobieństwo sukcesu należałoby transmitować $\Omega(\log n)$ razy, więc z niezerowym prawdopodobieństwem pewna stacja mogłaby nadawać $\Omega(\log n)$. Natomiast w pracy [H6] ograniczenie na energię jest sublogarytmiczne (a nawet stałe). Ponadto wymagamy, żeby ograniczenie na energię było zawsze zachowane (a nie tylko z dużym prawdopodobieństwem). Po drugie, optymalne prawdopodobieństwo nadawania zależy od liczby stacji, które mają wiadomość do nadania w pewnym otoczeniu, a taka wartość nie jest znana.

Aby poradzić sobie z drugim problemem, w naszym algorytmie, gdy chcemy rozwiązać konflikt między pewną grupą stacji, dzielimy je losowo na niejednorodne podgrupy. Pokazujemy, że niezależnie od początkowego rozmiaru grupy, znajdziemy niepustą podgrupę o określonym rozmiarze co najwyżej $\mathcal{O}(n^{1/\varphi}/\varphi \cdot \log n)$. Nie wiemy, która podgrupa ma ten rozmiar, ale wiemy, że taka istnieje. Następnie, aby poradzić sobie z pierwszym problemem zaproponowaliśmy w naszym algorytmie podprocedurę rozwiązywania konfliktów w podgrupie, bazującą na klasycznym, dobrze przebadanym zagadnieniu *Balls into Bins* [RS98], znanym również pod nazwą *Random Mapping* [FO89]. W tej podprocedurze, jesteśmy w stanie zagwarantować duże prawdopodobieństwo skutecznej transmisji. Dodatkowo każda stacja w podprocedurze wykonuje dokładnie jedno nadawanie. Aby podprocedura działała z dużym prawdopodobieństwem, jest wymagane ograniczenie $\mathcal{O}(n^{1/\varphi}/\varphi \cdot \log n)$ na liczbę stacji uczestniczących w niej jednocześnie, ale to jest już zagwarantowane dzięki podziałowi na podgrupy. Te dwie procedury są głównymi

⁴Algorytm rotor-router polega na wybieraniu przez agenta portów przy kolejnych wizytach w danym wierzchołku, zgodnie z regułą round-robin, czyli po kolei porty $1, 2, \dots, d, 1, 2, \dots, d, \dots$. Aby zaimplementować taką procedurę na każdym wierzchołku wystarczy $\log_2 d$ bitów, gdyż wystarczy pamiętać pozycję w sekwencji.

⁵Oznaczenie k odpowiada oznaczeniu φ w pracy [H6].

składnikami naszego pierwszego algorytmu działającego w czasie $\mathcal{O}(D \cdot n^{1/\varphi} \cdot \varphi)$ w.h.p. i używającego mniej niż 2φ transmisji dla każdej stacji w grafach rozmiaru n i średnicy D [H6, Theorem 2.2].

Drugim wynikiem jest algorytm rozwiązujący globalne rozgłaszanie w czasie $\mathcal{O}((D + \log n) \log n)$ i używający $\mathcal{O}(\log n / \log \log n)$ jednostek energii [H6, Theorem 2.3]. Złożoność czasowa naszego drugiego algorytmu jest taka sama jak klasycznego algorytmu Decay [BGI92], natomiast złożoność energetyczna jest niższa. W pracach [ABLP91, KM98] autorzy pokazali, że w sieciach radiowych nie istnieje algorytm działający szybciej niż $\Omega(D \log(n/D) + \log^2 n)$. Dodatkowo pokazujemy ograniczenie dolne mówiące, że każdy algorytm o złożoności polilogarytmicznej ze względu na n (nawet w grafach o stałej średnicy) potrzebuje $\Omega(\log n / \log \log n)$ jednostek energii [H6, Theorem 3.1]. Pokazaliśmy tym samym, że nasz algorytm, o asymptotycznie optymalnej złożoności czasowej (dla większości wartości parametrów n i D) ma złożoność energetyczną, której nie da się poprawić bez istotnego wydłużenia czasu działania algorytmu.

4.4.7 Praca H7

W pracy [H7] zajmujemy się czasem eksploracji (*cover time*) dla wielu błędzeń losowych na ścieżce i wielowymiarowych kratkach. Praca [AAK⁺11] jako pierwsza pokazała, że wiele błędzeń losowych działających współbieżnie eksploruje niektóre grafy szybciej, niż pojedyncze błędzenie. Czas eksploracji pojedynczego błędzenia jest zdefiniowany jako wartość oczekiwana zmiennej losowej będącej liczbą rund, po której wszystkie wierzchołki zostaną odwiedzone przez błędzenie. Punkt startowy jest brany jako ten, który powoduje największą oczekiwaną wartość tej zmiennej. Dla wielu błędzeń, czas eksploracji jest zdefiniowany jako wartość oczekiwana zmiennej będącej liczbą rund, po której każdy wierzchołek jest odwiedzony przez dowolne z błędzeń. Zakładamy, że wszystkie błędzenia startują z tego samego wierzchołka (wybranego tak, aby zmaksymalizować czas eksploracji). Wartość, którą badamy w pracy [H7] jest *przyspieszenie*, czyli iloraz czasu eksploracji jednego błędzenia i k błędzeń na takim samym grafie.

W pracy [AAK⁺11] autorzy postawili hipotezę że $\Omega(\log k)$ i $\mathcal{O}(k)$ są ograniczeniami na przyspieszenie, które zachodzą dla każdego grafu. W literaturze jest wiele wyników, które pokazują, że przyspieszenie dla różnych klas grafów zawiera się w tych ograniczeniach, jednak obie te hipotezy dla ogólnych grafów pozostają otwarte.

Pierwszym wynikiem w pracy [H7] jest dokładna analiza czasu eksploracji dla ścieżki [H7, Main result 1]. Inspiracją był fascynujący przykład w pracy [ER09], gdzie autorzy pokazali, że dla ścieżki długości 3 i dla dwóch agentów przyspieszenie wynosi 2.25. W pracy [H7] pokazujemy, że dla dowolnie długiej ścieżki i dla dwóch agentów, efekt synergii wciąż zachodzi i przyspieszenie jest większe niż 2 [H7, Theorem 4.9]. Jednak dla $k \geq 3$ pokazujemy, że przyspieszenie jest już mniejsze niż k [H7, Theorem 4.3]. Oczywiście nie obala to hipotezy, że przyspieszenie jest zawsze $\mathcal{O}(k)$, gdyż przyspieszenie powyżej k zachodzi tylko dla $k = 2$ i dodatkowo bardzo nieznacznie przekracza 2, jednak jest to ciekawe, że takie efekty zachodzą nawet w grafach o dowolnej liczbie wierzchołków.

Drugim wynikiem [H7] jest asymptotyczna analiza czasów eksploracji dla wielowymiarowych krat. Dla krat o wymiarze $d \geq 3$ i n wierzchołkach pokazaliśmy, że czas wynosi $\Theta(n \log n / k)$ dla $k \in [1, n^{1-2/d} \log n]$, natomiast dla $k \in [n^{1-2/d} \log n, n]$, czas eksploracji wynosi $\mathcal{O}(n^{2/d} / \log(k / (n^{1-2/d} \log n)))$ [H7, Main result 3]. To pokazuje, że przyspieszenie jest liniowe dla $k \leq n^{1-2/d} \log n$ i logarytmiczne dla większej liczby błędzeń. Analogiczne wyniki uzyskaliśmy również dla $d = 2$ [H7, Main result 2]. Taki progowy fenomen, gdzie przyspieszenie zmienia się wraz ze zmianą k był już obserwowany w literaturze dla innych grafów [ES11], jak również dla modelu derandomizowanych błędzeń losowych (rotor-router) [KP14].

4.5 Podsumowanie

W pracach składających się na dzieło habilitacyjne otrzymaliśmy następujące najważniejsze wyniki:

1. W pracy [H1] skonstruowaliśmy algorytm efektywnej alokacji zadań w koloniach owadów społecznych w modelu z niepewną informacją zwrotną.
2. W pracy [H2] przebadaliśmy zbieżność procesu usredniania w modelu z losowym szumem w komunikacji.
3. W pracy [H3] pokazaliśmy, że prosty algorytm typu back-off jest skutecznym narzędziem do komunikacji w modelu z realistycznym adwersarzem modyfikującym topologię sieci.

4. W pracy [H4] skonstruowaliśmy algorytmy wyboru lidera w sieci single-hop z detekcją kolizji działające pomimo bardzo silnego adwersarza typu jammer.
5. W pracy [H5] pokazaliśmy, że algorytm eksploracji używający minimalnej możliwej (logarytmicznej) pamięci na wierzchołkach nie może zostać przyspieszony nawet przez dodanie nieograniczonej pamięci na wierzchołkach.
6. W pracy [H6] pokazaliśmy, jak zmodyfikować popularny algorytm komunikacyjny tak, aby bez straty na złożoności czasowej, zużywał mniej energii.
7. W pracy [H7] przeanalizowaliśmy czas eksploracji krat wielowymiarowych przez wiele błędzeń losowych.

W przedstawionych pracach analizowaliśmy algorytmy rozproszone z punktu widzenia *odporności* na zaburzenia, działania adwersarza i ograniczenia w zasobach. Każde z problemów, które analizowaliśmy, były już rozważane w literaturze w modelach bez proponowanych przez nas ograniczeń. Rozszerzenie analizy problemów rozproszonych o modele z takimi utrudnieniami zwiększa ich niezawodność (w przypadku modeli z zaburzeniami i adwersarzem) i oszczędność (w przypadku modeli z ograniczonymi zasobami). Obie te cechy mają duże znaczenie z punktu widzenia wdrożenia systemu w praktyce. W przypadku modelu kolonii mrówek, włączenie niepewności do obserwacji zwiększa realizm naszego modelu.

5 Informacja o wykazywaniu się istotną aktywnością naukową albo artystyczną realizowaną w więcej niż jednej uczelni, instytucji naukowej lub instytucji kultury, w szczególności zagranicznej.

- **08.2017–04.2019 MIT, Stany Zjednoczone** W latach 2017 - 2019 byłem na dwuletnim stażu podoktorskim na MIT w grupie prof. Nancy Lynch. Brałem udział w projekcie dotyczącym sieci radiowych ze stochastycznym adwersarzem, czego efektem jest praca [H3]. Ponadto uczestniczyłem w projekcie dotyczącym algorytmów dla koloni owadów, czego efektem jest praca [H1]. Dodatkowo, podczas gdy MIT odwiedził dr Yannic Maus, razem z dr Frederikiem Mallmann-Trenn rozpoczęliśmy współpracę naukową na temat protokołów populacyjnych, co zaowocowało pracą [H2]. Podczas wizyt na Politechnice Wrocławskiej, prowadziliśmy razem z dr hab. Markiem Klonowskim badania nad efektywnym energetycznie algorytmem rozgłaszania. Wyniki tych badań zostały opublikowane w pracy [H6].
- **11.2014–10.2015 University of Cambridge, Wielka Brytania** W latach 2014-2015, podczas rocznego stażu na uniwersytecie w Cambridge zajmowałem się głównie analizą wielu błędzeń losowych w zespole dr Thomasa Sauerwalda. Efektem tych badań jest praca [H7]. W tym samym czasie, podczas wizyt na Politechnice Wrocławskiej, pracowaliśmy razem z dr hab. Markiem Klonowskim nad rozwiązaniem problemu wyboru lidera w modelu z adwersarzem. Rezultaty zostały opublikowane w [H4].

6 Informacja o osiągnięciach dydaktycznych.

- W 2018 roku byłem autorem tematów projektów studenckich do przedmiotu *Distributed algorithms*, prowadzonego przez prof. Nancy Lynch na MIT.
- W 2015 roku prowadziłem ćwiczenia do przedmiotu *Advanced algorithms*, którego wykładowcą był dr Thomas Sauerwald na uniwersytecie w Cambridge.
- Byłem współpromotorem pracy magisterskiej Artura Menca na Politechnice Wrocławskiej, ocenionej na ocenę celującą.

7 Omówienie pozostałych osiągnięć naukowych.

7.1 Nagrody

- **Best Paper Award OPODIS 2018** za artykuł *On Simple Back-Off in Unreliable Radio Networks*, którego współautorami byli Seth Gilbert, Nancy Lynch i Calvin Newport.
- **Prix de thèse 2015**. Moja praca doktorska zdobyła specjalną nagrodę jury w konkursie Prix de thèse 2015 na uniwersytecie w Bordeaux⁶
- **Best Paper Award ICALP 2013** za artykuł *Fast Collaborative Graph Exploration*, którego współautorami byli Dariusz Dereniowski, Yann Disser, Adrian Kosowski i Przemysław Uznański.

7.2 Pozostałe publikacje

Moje pozostałe osiągnięcia naukowe, opublikowane po obronie doktoratu to: osiem publikacji konferencyjnych [CGG⁺14, CKPP15, BKK⁺16, BCG⁺16, KKP16, CGK⁺16b, CGK⁺16a, GKP17], trzy artykuły czasopismowe [DKP15, BCG⁺19, BKK⁺20], pięć artykułów czasopismowych zawierających rozszerzone wersje artykułów wchodzących w skład mojej pracy doktorskiej [DDK⁺15, DKPU16, CDG⁺17, KKPS17, KP19] oraz dwie publikacje konferencyjne złożone do recenzji.

7.2.1 Problemy dla mobilnych robotów

W cyklu prac [CGG⁺14, CKPP15, BCG⁺16, CGK⁺16b, CGK⁺16a, BCG⁺19] rozważamy różne problemy (m.in. ewakuacji i przeszukiwania) dla robotów mobilnych w środowiskach ciągłych.

Ewakuacja W pracy [CGG⁺14] rozważamy k mobilnych robotów umieszczonych na okrągłym dysku o jednostkowym promieniu. Wszystkie roboty mają za zadanie, w jak najkrótszym czasie opuścić dysk przez, początkowo nieznaną, punkt wyjściowy umieszczony na brzegu dysku. Zakładamy, że wszystkie roboty mają identyczną prędkość i rozpoczynają algorytm na środku dysku. Rozważamy dwa warianty komunikacji między robotami:

- Przy komunikacji *lokalnej* roboty mogą się poinformować o pozycji wyjścia jedynie przy spotkaniu.
- Przy komunikacji *bezprowadowej* roboty mogą się komunikować w dowolnym momencie.

W [CGG⁺14] udowadniamy liczne ograniczenia na optymalny czas ewakuacji (patrz Tabela 1).

Model komunikacji	Ograniczenie	$k = 2$	$k = 3$	$k \geq 4$
Lokalna	Górne	~ 5.74	~ 5.09	$3 + \frac{2\pi}{k}$
	Dolne	~ 5.199	~ 4.519	$3 + \frac{2\pi}{k} - O(k^{-2})$
Bezprowadowa	Górne	~ 4.83	~ 4.22	$3 + \frac{\pi}{k} + O(k^{-4/3})$
	Dolne	~ 4.83	~ 4.159	$3 + \frac{\pi}{k}$

Tabela 1: Ograniczenia na czas ewakuacji dla $k \geq 2$ robotów

W pracy [BCG⁺19] (i w jej wersji konferencyjnej [BCG⁺16]) rozważamy problem ewakuacji dla dwóch robotów o różnych prędkościach na nieskończonej linii. Zakładamy, że szybszy robot ma prędkość 1, a wolniejszy ma prędkość $v \leq 1$, a punkt wyjściowy jest położony w odległości d od punktu startu (wartość d nie jest znana dla robotów).

W modelu z komunikacją lokalną dla $v > \frac{1}{3}$, udowadniamy ograniczenie dolne $9d - o(d)$ na czas ewakuacji. Dla $v > \frac{1}{3}$, czas ewakuacji $9d$ jest również ograniczeniem górnym [CGGM15]. Dla $v \leq \frac{1}{3}$ pokazujemy strategię o czasie działania $\frac{1+3v}{v-v^2}d$ i udowadniamy, że jest optymalna.

⁶Jest konkurs, w którym corocznie wyłanianych jest czterech laureatów z różnych dziedzin.

W modelu z komunikacją bezprzewodową, konstruujemy algorytm o czasie ewakuacji $\frac{2+v+\sqrt{v^2+8v}}{2v}d$. Nasz algorytm w modelu z komunikacją bezprzewodową osiąga krótszy czas ewakuacji niż dowolny algorytm w modelu z komunikacją lokalną dla $v > \sqrt{17} - 4 \approx 0.123$, co pokazuje, że komunikacja bezprzewodowa jest przydatna w pewnych zakresach prędkości. Co ciekawe, pokazujemy również ograniczenie dolne, które mówi, że dla $v \leq \sqrt{17} - 4$ nasz algorytm dla komunikacji lokalnej jest również optymalny dla komunikacji bezprzewodowej. To pokazuje, że dla pewnego zakresu prędkości, komunikacja bezprzewodowa nie pozwala na szybszą ewakuację.

Patrowanie W problemie patrolowania odcinka robotami, analizowanym w pracach [CGK⁺16b, CGK⁺16a], każdy z robotów może poruszać się w jednym z dwóch trybów: *chodzenie* i *patrowanie*. W tych trybach robot porusza się z różnymi maksymalnymi prędkościami. Celem problemu jest znalezienie takiego nieskończonego harmonogramu poruszania (i zmiany trybów) dla robotów, aby zminimalizować *czas nieobecności*. Ta wartość jest zdefiniowana jako maksymalny (po wszystkich punktach) czas pomiędzy kolejnymi odwiedzinami punktu przez robota w trybie patrolowania.

Pierwszym wynikiem prac [CGK⁺16b, CGK⁺16a] jest scentralizowana konstrukcja harmonogramu osiągająca optymalny czas nieobecności. Następnie pokazujemy, w jaki sposób rozproszony system robotów (bez centralnej kontroli) może zbiegać do optymalnego harmonogramu. W [CGK⁺16b] rozważamy taką zbieżność dla dwóch robotów, a w [CGK⁺16a] dla trzech i czterech robotów.

Interesującym zastosowaniem zdefiniowanego problemu patrolowania jest następujący problem transportowy, który nazwaliśmy problemem regularnych dostaw. Zakładamy, że jeden koniec odcinka jest źródłem produktów, które należy regularnie dostarczać na drugi koniec odcinka. Ponadto, robot niosący produkt, porusza się wolniej niż bez produktu. W pracach [CGK⁺16b, CGK⁺16a] pokazujemy, że algorytmy patrolowania mogą być wykorzystane jako algorytmy dla problemu regularnych dostaw.

Komunikacja na linii W pracy [CKPP15] rozważamy problemy komunikacyjne dla grupy robotów poruszających się ze stałymi, różnymi prędkościami na linii. Każdy z robotów posiada pewną informację i zakładamy, że dwa roboty przy spotkaniu mogą wymienić się informacjami (początkowymi, i tymi które zdobyły przy innych spotkaniach). Dodatkowo, przy spotkaniu następuje wymiana prędkości (zgodnie z regułami zderzenia sprężystego dla ciał o równych masach). Liczba takich kolizji jest skończona [Sev93], więc możemy zastanawiać się na przykład jaki będzie zasięg informacji początkowo posiadanej przez robota r_i .

Pokazujemy warunki konieczne i wystarczające na stan początkowy robotów, żeby informacja od robota r_i przedostała się do robota r_j (dla dowolnych r_i i r_j). Dodatkowo udowadniamy warunki konieczne i wystarczające na rozwiązanie podstawowych problemów komunikacyjnych (gossiping, convergecast, rozgłaszanie). Te warunki zależą wyłącznie od początkowej kolejności robotów i ich prędkości, a nie zależą od początkowych pozycji.

7.2.2 Poziom symetrii (*level of symmetry*)

Anonimowy graf z poetykietowanymi portami jest grafem nieskierowanym, w którym wierzchołki nie mają etykiet, ale w każdym wierzchołku o stopniu d , krawędzie wychodzące z tego wierzchołka są poetykietowane etykietami ze zbioru $\{1, 2, \dots, d\}$ (oba końce tego samego wierzchołka mogą mieć inne etykiety). *Widok* z wierzchołka jest nieskończonym drzewem wszystkich spacerów rozpoczynających się w tym wierzchołku. Widok z wierzchołka v ograniczony do pewnego poziomu t jest całą wiedzą, jaka może być zdobyta przez wierzchołek v w t rund rozproszonego algorytmu, na przykład w modelu komunikacyjnym LOCAL. Poziom symetrii jest minimalną wartością δ , taką, że jeżeli widoki z dwóch wierzchołków v i w ograniczone do poziomu δ są identyczne, to całe nieskończone drzewa widoków z v i w są identyczne. Zatem poziom symetrii ma istotne znaczenie dla wszystkich problemów wymagających deterministycznego łamania symetrii (jak wybór lidera), gdyż jest to ograniczenie na liczbę rund, do której jest możliwe rozróżnienie wierzchołków przy pomocy widoków [FP15]. W pracy [Hen14] autor udowadnia granicę dolną $\mathcal{O}(D \log(n/D))$ na poziom symetrii, dla dowolnych grafów o średnicy D i liczbie wierzchołków n .

W pracy [DKP15] konstruujemy rodzinę zbiorów, dla której $\Omega(D \log(n/D))$ jest ograniczeniem dolnym na poziom symetrii, pokazujemy zatem optymalność wyniku z pracy [Hen14].

W pracy [KKP16] pokazujemy, że losowe etykietowanie portów ma poziom symetrii $\mathcal{O}(\min\{D, \log n\})$ w.h.p., ponadto pokazujemy, jak deterministycznie w czasie wielomianowym skonstruować takie poetykietowanie.

7.2.3 Aproksymacja rozmiaru (*size approximation*)

W pracy [BKK⁺20] (wersja konferencyjna: [BKK⁺16]) zajmujemy się problemem aproksymacji rozmiaru w sieciach radiowych Ad-hoc. W tych pracach zakładamy sieć single-hop, gdzie każde dwie stacje mogą się komunikować (sieć komunikacyjna jest kliką). Modelem komunikacyjnym w tych pracach jest *Beeping Model*, gdzie wspólny kanał jest w każdej rundzie komunikacyjnej w jednym z dwóch stanów:

- Silence, żadna stacja nie nadaje w tej rundzie,
- Beep, co najmniej jedna stacja nadaje w tej rundzie.

W tym modelu nie jest możliwe przesłanie żadnej treści w wiadomości, a cały protokół jest oparty jedynie na tych dwóch stanach kanału Silence i Beep. W takim prostym modelu komunikacyjnym jest możliwe rozwiązywanie nawet skomplikowanych problemów [GN15] i ma on również interesujące powiązanie z obliczeniami w sieciach biologicznych [AAB⁺11].

W pracach [BKK⁺16, BKK⁺20] pokazujemy algorytm, w którym każda stacja oblicza $(1+\varepsilon)$ -aproksymację łącznej liczby stacji w sieci n w czasie $O(\log \log n + \log f/\varepsilon^2)$ z prawdopodobieństwem co najmniej $1-1/f$. Nasz algorytm należy do klasy algorytmów *uniform*, w której w każdej rundzie każda stacja transmituje niezależnie z identycznym prawdopodobieństwem p (to prawdopodobieństwo może się zmieniać z rundy na rundę w zależności od stanu kanału). Pokazujemy również ograniczenie dolne $\Omega(\log \log n + \log f/\varepsilon^2)$ na czas aproksymacji rozmiaru dowolnym algorytmem typu uniform.

7.2.4 Ukrywanie w sieciach

W pracy [GKP17] zakładamy, że pewna liczba mobilnych agentów pracuje nad pewnym zadaniem w grafie. Wierzchołki, w których pracują agenci jest informacją wrażliwą i nie chcemy żeby adwersarz zdobył jakąkolwiek informację na ich temat. Adwersarz może sprawdzić pozycje agentów, ale na krótko przed takim sprawdzeniem agenci są o tym informowani i mają możliwość zmiany swojej pozycji. Chcemy zaprojektować taki algorytm przemieszczania, aby po możliwie małej liczbie rund, końcowa pozycja agentów ujawniała mało informacji (w sensie informacji wzajemnej) o początkowej pozycji. W pracy [GKP17] prezentujemy algorytm i ograniczenia dolne na liczbę rund potrzebną do ukrycia w zależności od znajomości przez agentów topologii grafu oraz dostępności pamięci i losowości.

Bibliografia

- [AAB⁺11] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, Ziv Bar-Joseph. A biological solution to a fundamental distributed computing problem. *science*, 331(6014):183–185, 2011.
- [AAD⁺06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.
- [AAK⁺11] Noga Alon, Chen Avin, Michal Koucký, Gady Kozma, Zvi Lotker, Mark R. Tuttle. Many random walks are faster than one. *Comb. Probab. Comput.*, 20(4):481–502, 2011.
- [ABLP91] Noga Alon, Amotz Bar-Noy, Nathan Linial, David Peleg. A lower bound for radio broadcast. *J. Comput. Syst. Sci.*, 43(2):290–298, 1991.
- [AKL⁺79] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, strony 218–223. IEEE Computer Society, 1979.
- [AKL18] Chen Avin, Michal Koucký, Zvi Lotker. Cover time and mixing time of random walks on dynamic graphs. *Random Structures & Algorithms*, 52(4):576–596, 2018.

- [ARS08] Baruch Awerbuch, Andréa W. Richa, Christian Scheideler. A jamming-resistant MAC protocol for single-hop wireless networks. *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, Toronto, Canada, August 18-21, 2008*, strony 45–54. ACM, 2008.
- [ARS⁺14] Baruch Awerbuch, Andréa W. Richa, Christian Scheideler, Stefan Schmid, Jin Zhang. Principles of robust medium access and an application to leader election. *ACM Trans. Algorithms*, 10(4):24:1–24:26, 2014.
- [BCG⁺16] Evangelos Bampas, Jurek Czyzowicz, Leszek Gasieniec, David Ilcinkas, Ralf Klasing, Tomasz Kociumaka, Dominik Pajak. Linear search by a pair of distinct-speed robots. *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, wolumen 9988 serii *Lecture Notes in Computer Science*, strony 195–211, 2016.
- [BCG⁺19] Evangelos Bampas, Jurek Czyzowicz, Leszek Gasieniec, David Ilcinkas, Ralf Klasing, Tomasz Kociumaka, Dominik Pajak. Linear search by a pair of distinct-speed robots. *Algorithmica*, 81(1):317–342, 2019.
- [BGH⁺17] Evangelos Bampas, Leszek Gasieniec, Nicolas Hanusse, David Ilcinkas, Ralf Klasing, Adrian Kosowski, Tomasz Radzik. Robustness of the rotor-router mechanism. *Algorithmica*, 78(3):869–895, 2017.
- [BGI92] Reuven Bar-Yehuda, Oded Goldreich, Alon Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *J. Comput. Syst. Sci.*, 45(1):104–126, 1992.
- [BKK⁺16] Philipp Brandes, Marcin Kardas, Marek Klonowski, Dominik Pajak, Roger Wattenhofer. Approximating the size of a radio network in beeping model. *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, wolumen 9988 serii *Lecture Notes in Computer Science*, strony 358–373, 2016.
- [BKK⁺20] Philipp Brandes, Marcin Kardas, Marek Klonowski, Dominik Pajak, Roger Wattenhofer. Fast size approximation of a radio network in beeping model. *Theor. Comput. Sci.*, 810:15–25, 2020.
- [BKL⁺13] Emrah Bayraktaroglu, Christopher King, Xin Liu, Guevara Noubir, Rajmohan Rajaraman, Bishal Thapa. Performance of IEEE 802.11 under jamming. *Mobile Networks and Applications*, 18(5):678–696, 2013.
- [CCDS17] Ho-Lin Chen, Rachel Cummings, David Doty, David Soloveichik. Speed faults in computation by chemical reaction networks. *Distributed Computing*, 30(5):373–390, 2017.
- [CDG⁺17] Jurek Czyzowicz, Dariusz Dereniowski, Leszek Gasieniec, Ralf Klasing, Adrian Kosowski, Dominik Pajak. Collision-free network exploration. *J. Comput. Syst. Sci.*, 86:70–81, 2017.
- [CDH⁺18] Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, Qizheng He, Wenzheng Li, Seth Pettie. The energy complexity of broadcast. *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, strony 95–104. ACM, 2018.
- [CDHP20] Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, Seth Pettie. The energy complexity of BFS in radio networks. *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, strony 273–282. ACM, 2020.
- [CDLN14] Alejandro Cornejo, Anna R. Dornhaus, Nancy A. Lynch, Radhika Nagpal. Task allocation in ant colonies. *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, wolumen 8784 serii *Lecture Notes in Computer Science*, strony 46–60. Springer, 2014.

- [CGG⁺14] Jurek Czyzowicz, Leszek Gasieniec, Thomas Gorry, Evangelos Kranakis, Russell Martin, Dominik Pajak. Evacuating robots via unknown exit in a disk. *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, wolumen 8784 serii *Lecture Notes in Computer Science*, strony 122–136. Springer, 2014.
- [CGGM15] Marek Chrobak, Leszek Gasieniec, Thomas Gorry, Russell Martin. Group search on the line. *SOFSEM 2015: Theory and Practice of Computer Science - 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24-29, 2015. Proceedings*, wolumen 8939 serii *Lecture Notes in Computer Science*, strony 164–176. Springer, 2015.
- [CGK⁺16a] Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Fraser MacQuarrie, Dominik Pajak. Distributed patrolling with two-speed robots (and an application to transportation). *Operations Research and Enterprise Systems - 5th International Conference, ICORES 2016, Rome, Italy, February 23-25, 2016, Revised Selected Papers*, wolumen 695 serii *Communications in Computer and Information Science*, strony 71–95, 2016.
- [CGK⁺16b] Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Fraser MacQuarrie, Dominik Pajak. Fence patrolling with two-speed robots. *Proceedings of 5th the International Conference on Operations Research and Enterprise Systems (ICORES 2016), Rome, Italy, February 23-25, 2016*, strony 229–241. SciTePress, 2016.
- [CHHZ17] Keren Censor-Hillel, Bernhard Haeupler, D. Ellis Hershkowitz, Goran Zuzic. Broadcasting in noisy radio networks. *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, strony 33–42. ACM, 2017.
- [CHHZ19] Keren Censor-Hillel, Bernhard Haeupler, D. Ellis Hershkowitz, Goran Zuzic. Erasure correction for noisy radio networks. *33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary*, wolumen 146 serii *LIPICs*, strony 10:1–10:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [CK85] Imrich Chlamtac, Shay Kutten. On broadcasting in radio networks-problem analysis and protocol design. *IEEE Trans. Commun.*, 33(12):1240–1246, 1985.
- [CKP19] Yi-Jun Chang, Tsvi Kopelowitz, Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model. *SIAM J. Comput.*, 48(1):122–143, 2019.
- [CKPP15] Jurek Czyzowicz, Evangelos Kranakis, Eduardo Pacheco, Dominik Pajak. Information spreading by mobile particles on a line. *Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings*, wolumen 9439 serii *Lecture Notes in Computer Science*, strony 285–298. Springer, 2015.
- [DDK⁺15] Dariusz Dereniowski, Yann Disser, Adrian Kosowski, Dominik Pajak, Przemyslaw Uznanski. Fast collaborative graph exploration. *Inf. Comput.*, 243:37–49, 2015.
- [DKP15] Dariusz Dereniowski, Adrian Kosowski, Dominik Pajak. Distinguishing views in symmetric networks: A tight lower bound. *Theor. Comput. Sci.*, 582:27–34, 2015.
- [DKPU16] Dariusz Dereniowski, Adrian Kosowski, Dominik Pajak, Przemyslaw Uznanski. Bounds on the cover time of parallel rotor walks. *J. Comput. Syst. Sci.*, 82(5):802–816, 2016.
- [ER09] Klim Efremenko, Omer Reingold. How well do random walks parallelize? *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, wolumen 5687 serii *Lecture Notes in Computer Science*, strony 476–489. Springer, 2009.
- [ES11] Robert Elsässer, Thomas Sauerwald. Tight bounds for the cover time of multiple random walks. *Theor. Comput. Sci.*, 412(24):2623–2641, 2011.

- [FLP85] Michael J. Fischer, Nancy A. Lynch, Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.
- [FO89] Philippe Flajolet, Andrew M Odlyzko. Random mapping statistics. *Workshop on the Theory and Application of Cryptographic Techniques*, strony 329–354. Springer, 1989.
- [FP15] Emanuele G. Fusco, Andrzej Pelc. Knowledge, level of symmetry, and time of leader election. *Distributed Comput.*, 28(4):221–232, 2015.
- [GHLN12] Mohsen Ghaffari, Bernhard Haeupler, Nancy A. Lynch, Calvin C. Newport. Bounds on contention management in radio networks. *Distributed Computing - 26th International Symposium, DISC 2012, Salvador, Brazil, October 16-18, 2012. Proceedings*, wolumen 7611 serii *Lecture Notes in Computer Science*, strony 223–237. Springer, 2012.
- [GK00] Piyush Gupta, Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.
- [GKP17] Karol Gotfryd, Marek Klonowski, Dominik Pajak. On location hiding in distributed systems. *Structural Information and Communication Complexity - 24th International Colloquium, SIROCCO 2017, Porquerolles, France, June 19-22, 2017, Revised Selected Papers*, wolumen 10641 serii *Lecture Notes in Computer Science*, strony 174–192. Springer, 2017.
- [GLN13] Mohsen Ghaffari, Nancy A. Lynch, Calvin C. Newport. The cost of radio network broadcast for different models of unreliable links. *Proceedings of the 32nd Annual ACM Symposium on Principles of Distributed Computing*, strony 345–354, Montreal, Canada, July 2013.
- [GM99] Deborah M. Gordon, Natasha J. Mehdiabadi. Encounter rate and task allocation in harvester ants. *Behavioral Ecology and Sociobiology*, 45(5):370–377, Apr 1999.
- [GN15] Seth Gilbert, Calvin C. Newport. The computational power of beeps. *International Symposium on Distributed Computing*, strony 31–46. Springer, 2015.
- [Gor96] Deborah M. Gordon. The organization of work in social insect colonies. *Nature*, 380(14 March):121–124, 1996.
- [Hen14] Julien M. Hendrickx. Views in a graph: To which depth must equality be checked? *IEEE Trans. Parallel Distributed Syst.*, 25(7):1907–1912, 2014.
- [KKP16] Ralf Klasing, Adrian Kosowski, Dominik Pajak. Setting ports in an anonymous network: How to reduce the level of symmetry? *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, wolumen 9988 serii *Lecture Notes in Computer Science*, strony 35–48, 2016.
- [KKPS17] Ralf Klasing, Adrian Kosowski, Dominik Pajak, Thomas Sauerwald. The multi-agent rotor-router on the ring: a deterministic alternative to parallel random walks. *Distributed Comput.*, 30(2):127–148, 2017.
- [KLN⁺10] Fabian Kuhn, Nancy A. Lynch, Calvin C. Newport, Rotem Oshman, Andréa W. Richa. Broadcasting in unreliable radio networks. *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010*, strony 336–345. ACM, 2010.
- [KM98] Eyal Kushilevitz, Yishay Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM J. Comput.*, 27(3):702–712, 1998.
- [KP14] Adrian Kosowski, Dominik Pajak. Does adding more agents make a difference? A case study of cover time for the rotor-router. *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, wolumen 8573 serii *Lecture Notes in Computer Science*, strony 544–555. Springer, 2014.

- [KP16] Erez Kantor, David Peleg. Efficient k-shot broadcasting in radio networks. *Discret. Appl. Math.*, 202:79–94, 2016.
- [KP19] Adrian Kosowski, Dominik Pajek. Does adding more agents make a difference? A case study of cover time for the rotor-router. *J. Comput. Syst. Sci.*, 106:80–93, 2019.
- [LL90] Leslie Lamport, Nancy A. Lynch. Distributed computing: Models and methods. *Formal models and semantics*, strony 1157–1199. Elsevier, 1990.
- [LN15] Nancy A. Lynch, Calvin C. Newport. A (truly) local broadcast layer for unreliable radio networks. *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, 2015.
- [LSP82] Leslie Lamport, Robert Shostak, Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.
- [Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
- [MSL17] Cameron Musco, Hsin-Hao Su, Nancy A. Lynch. Ant-inspired density estimation via random walks. *Proc. Natl. Acad. Sci. USA*, 114(40):10534–10541, 2017.
- [NR18] Calvin C. Newport, Peter Robinson. Fault-tolerant consensus with an abstract MAC layer. *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, wolumen 121 serii *LIPICs*, strony 38:1–38:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [Pel00] David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.
- [Pra05] Stephen C. Pratt. Quorum sensing by encounter rates in the ant *temnothorax albipennis*. *Behavioral Ecology*, 16(2):488–496, 2005.
- [RS98] Martin Raab, Angelika Steger. "Balls into Bins" - A simple and tight analysis. *Randomization and Approximation Techniques in Computer Science, Second International Workshop, RANDOM'98, Barcelona, Spain, October 8-10, 1998, Proceedings*, wolumen 1518 serii *Lecture Notes in Computer Science*, strony 159–170. Springer, 1998.
- [RSSZ11] Andréa W. Richa, Christian Scheideler, Stefan Schmid, Jin Zhang. Self-stabilizing leader election for single-hop wireless networks despite jamming. *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2011, Paris, France, May 16-20, 2011*, strona 15. ACM, 2011.
- [Sev93] Mikhail B. Sevryuk. Estimate of the number of collisions of n elastic particles on a line. *Theoretical and Mathematical Physics*, 96(1):818–826, 1993.
- [YWB03] Vladimir Yanovski, Israel A. Wagner, Alfred M. Bruckstein. A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37(3):165–186, 2003.