AUTOMATYKA • 2009 • Tom 13 • Zeszyt 2

Łukasz Wrona*

Gwarantowanie bezpieczeństwa w systemie z połączeniami awaryjnymi

1. Wprowadzenie

Załóżmy, że w pewnym budynku wystąpiło niebezpieczeństwo ataku terrorystycznego. Grupa mobilnych, zautomatyzowanych agentów, ma za zadanie odnalezienie intruza lub stwierdzenie, że alarm był fałszywy. Ponieważ zadanie musi zostać wykonane niezawodnie, agenci winni przyjąć strategię gwarantującą odnalezienie intruza niezależnie od przyjętej przez niego strategii, posiadanej przez niego wiedzy, a także prędkości z jaką może się przemieszczać. Oznacza to, że jeśli w dowolnym momencie poszukiwania pojawi się ścieżka, na której nie stoi żaden agent, między już przeszukanym obszarem a potencjalną lokacją intruza, obszar ten będzie trzeba przeszukać ponownie. Przyjęcie tych założeń powoduje, że przedstawiony problem staje się równoważny innym problemom bezpieczeństwa systemów, takim jak opanowanie rozprzestrzeniającego się pożaru, usunięcia zagrożenia bakteriologicznego z danego obszaru czy usunięcie wirusa rozprzestrzeniającego się w sieci komputerowej.

Do zamodelowania tego problemu wykorzystane zostanie zagadnienie przeszukiwania grafu i wyznaczania jego liczby przeszukiwawczej. Liczba przeszukiwacza grafu określa minimalną ilość agentów niezbędnych do zagwarantowania przechwycenia uciekiniera przy przyjętych powyżej założeniach. By tego dokonać, agenci stosują strategię przeszukiwawczą. Wyznaczenie liczby przeszukiwawczej grafu jest problemem trudnym obliczeniowo dla dowolnych grafów, jednak jest możliwe w czasie wielomianowym dla drzew.

1.1. Systemy z połączeniami awaryjnymi

Załóżmy, że rozważany system posiada topologię drzewa. Oznacza to, że nawet pomiędzy kluczowymi elementami systemu istnieje tylko jedno połączenie. Ze względu na zawodność systemów, takie rozwiązanie jest często niewskazane.

Jeśli w systemie o topologii drzewa wprowadzimy połączenia awaryjne między kluczowymi punktami – przykładowo dodatkowa nitka gazociągu poprowadzona tak, by omi-

* Katedra Algorytmów i Modelowania Systemów, Politechnika Gdańska

nąć wybrany region i zagwarantować dostarczenie gazu do wyznaczonego odbiorcy – możemy uzyskać system o topologii kaktusa. W tak zaprojektowanym systemie między wybranymi elementami systemu – wierzchołkami istnieje więcej niż jedna ścieżka, jednak żadna krawędź nie należy do więcej niż jednego cyklu.

W tej pracy rozważamy problem wyznaczenia liczby przeszukiwawczej, a także strategii agentów zapewniających bezpieczeństwo, w systemie o topologii kaktusa.

1.2. Zakres pracy

W rozdziale 2 przedstawione są podstawowe terminy związane z gwarantowaniem bezpieczeństwa poprzez przeszukiwanie grafów. System z połączeniami awaryjnymi o topologii kaktusa powstaje z systemu o topologii drzewa i zachowuje pewne własności struktury drzewiastej względem gwarantowania bezpieczeństwa. Własności te w odniesieniu do drzew zostały podsumowane w podrozdziale 2.1.

Wybrane dotychczasowe obserwacje w zakresie gwarantowania bezpieczeństwa w kaktusach zostały przedstawione w podrozdziale 2.2. Aby zastosować metodologię analogiczną do gwarantowania bezpieczeństwa w drzewach, konieczne jest przeanalizwanie wpływu istnienia powstałych w wyniku dodania połączeń awaryjnych cykli na liczbę i strategię przeszukiwawczą drzewa. By tego dokonać, proponujemy ustalenie liczby przeszukiwawczej dowolnego grafu, zawierającego cykl taki, że usunięcie krawędzi należących do cyklu rozspaja graf na kaktusy o znanej liczbie przeszukiwawczej nie większej niż k. Liczba przeszukiwawcza takiego grafu zależy nie tylko od k, ilości odgałęzień i ich rozmieszczenia, ale także od ich dodatkowych własności. Konieczne jest więc zidentyfikowanie ukorzenionych kaktusów, w których te własności są obecne. W podrozdziale 3.1 zaprezentowano jedną z ważniejszych własności, oznaczaną jako (**), oraz zdefiniowano jej występowanie dla kaktusów posiadających rdzeń lub aleję.

Zgodnie z wynikami przedstawionymi w podrozdziale 2.2 liczba przeszukiwawcza kaktusa skonstruowanego w sposób omówiony powyżej należy do zbioru $\{k, k+1, k+2\}$. W podrozdziale 3.2 zmniejszono moc tego zbioru dla kaktusów podkubicznych do dwóch, dla każdej ilości odgałęzień z liczbą przeszukiwawczą nie większą niż k.

2. Dotychczasowe badania

Problem przeszukiwania grafów został po raz pierwszy zaproponowany w [1]. Grupa mobilnych agentów wykonuje **strategię przeszukiwawczą**, czyli sekwencję ruchów gwarantujących przechwycenie uciekiniera. Strategia przeszukiwawcza składa się z dowolnej kombinacji następujących ruchów:

– umieszczenie agenta w dowolnym wierzchołku grafu;
– usunięcie agenta z dowolnego wierzchołka grafu;
– przemieszczenie agenta wzdłuż krawędzi grafu.

Każdy z tych ruchów może zostać wykonany w stałym czasie.

Inicjalnie wszystkie krawędzie grafu są **skażone**. Krawędź $e=\{u,v\}$ może zostać **wyczyszczona**, w dwóch przypadkach. W każdym z nich pewien agent przemieszcza się wzdłuż *e,* od wierzchołka *u* do *v*. Krawędź *e* zostaje wyczyszczona, jeśli albo w wierzchołku *u* znajduje się agent, albo wszystkie krawędzie incydentne z *u,* oprócz *e*, są czyste. Wyczyszczenie *e* oznacza, iż agenci upewnili się, iż uciekinier nie znajduje się w *e*.

Ścieżkę nazywamy **strzeżoną**, jeśli w przynajmniej jednym z jej wewnętrznych wierzchołków przebywa agent. Jeśli w dowolnym momencie czysta krawędź *e* należy do niestrzeżonej ścieżki zawierającej krawędź skażoną, to *e* zostaje **ponownie skażona** (*recontaminated*).

Minimalna ilość agentów potrzebnych do przeszukania grafu *G* jest nazywana jego **(krawędziową) liczbą przeszukiwawczą** i będzie w tej pracy oznaczana jako ***s*(*G*)**. Jeśli *k* agentów wystarcza do przeszukania grafu *G*, to *k* agentów wystarcza do przeszukania grafu *G* bez dopuszczania do ponownego skażenia krawędzi [2]. Ta właściwość nazywana jest **monotonicznością.**

W dalszej części pracy używana będzie poszerzona definicja odgałęzienia, zaproponowana w [3]. Niech *v* będzie dowolnym przegubem w *G*. Niech *H* oznacza dowolną składową spójności powstałą po usunięciu *v* z *G*. **Odgałęzieniem** grafu *G* w wierzchołku *v* nazywamy podgraf powstały jako suma *H, v* oraz wszystkich krawędzi *G*, incydentnych zarówno z *v* jak i z jednym z wierzchołków w *H.* **Odgałęzieniem cyklu *C*** będziemy nazywali dowolne odgałęzienie w wierzchołku należącym do *C*, nie zawierające *C*.

Lemat 2.1. [1] *Niech G' będzie dowolnym podgrafem grafu G. Wówczas* $s(G') \leq s(G)$.

Więcej informacji na temat problemów gonitwy i ucieczki można znaleźć w [4]. Aktualny przegląd wyników z dziedziny przeszukiwania grafów znajduje się w [5].

2.1. Przeszukiwanie drzew

Wyznaczanie liczby przeszukiwawczej dowolnego grafu jest NP-trudne [6], jednak jej wyznaczenie jest możliwe w czasie wielomianowym dla drzew [6]. U podstaw algorytmu leży lemat Parsona [1]:

Lemat 2.2. [1] *Dla dowolnego drzewa T i liczby całkowitej k, wierzchołkowa liczba przeszukiwawcza* $s(T) = k + 1$ *wtedy i tylko wtedy, gdy w drzewie T istnieje wierzchołek v, w którym istnieją co najmniej trzy różne odgałęzienia o liczbie przeszukiwawczej równej k.*

Ścieżkę wierzchołków, o których mowa w lemacie 2.2, nazywamy aleją. Bardziej formalnie aleja drzewa *T* to ścieżka $v_1, v_2, \ldots, v_r$, składająca się z przynajmniej dwóch wierzchołków, taka że w v_1 i v_r istnieje dokładnie jedno odgałęzienie z liczbą przeszukiwawczą równą $s(T)$, a we wszystkich pozostałych istnieją dokładnie dwa takie odgałęzienia. Rdzeniem drzewa posiadającego liczbę przeszukiwawczą równą $s(T)$ nazywamy taki wierzchołek, w którym wszystkie odgałęzienia posiadają liczbę przeszukiwawczą mniejszą od $s(T)$. Rdzeń możemy traktować jako aleję o długości 0.

Lemat 2.3. [6] *Każde drzewo T posiada albo rdzeń, albo unikatową aleję.*

Warto zauważyć, że rdzeń drzewa nie musi być unikatowy. W minimalnym drzewie (w sensie definicji zaproponowanej w [1]) o danej liczbie przeszukiwawczej, każdy wierzchołek nie będący liściem jest rdzeniem.

Na podstawie lematu 2.3 wszystkie ukorzenione drzewa możemy podzielić na cztery kategorie, w zależności od położenia korzenia:

1) Typ H – Korzeń jest rdzeniem drzewa T.
2) Typ E – Korzeń znajduje się w jednym z odgałęzień w wierzchołkach krańcowych alei (v_1 lub v_r).
3) Typ I – Korzeń jest jednym z wewnętrznych wierzchołków alei ($v_2, \ldots, v_{r-1}$).
4) Typ M – Korzeń znajduje się w odgałęzieniu w którymś z wewnętrznych wierzchołków alei, ale nie na samej alei.

Dla drzewa typu M, M-odgałęzieniem nazywamy odgałęzienie alei zawierające korzeń drzewa. M-odgałęzienie zawsze posiada mniejszą liczbę przeszukiwawczą niż całe drzewo.

Na podstawie lematu 2.3 i powyższego podziału w [6] zaproponowano algorytm obliczający liczbę przeszukiwawczą drzewa w czasie liniowym. Algorytm działa w oparciu o struktury *info(T)* = [*typ, s, M-info*], gdzie *typ*, odpowiada jednemu z czterech typów wymienionych powyżej, s jest liczbą przeszukiwawczą drzewa T, a *M-info* jest polem pustym dla drzew typu H, E i I a dla drzew typu M zawiera strukturę *info* M-odgałęzienia drzewa T. *info*(T) jest wyznaczane w czasie $O(s(T))$ na podstawie struktur *info* dowolnych dwóch niepustych poddrzew T, posiadających dokładnie jeden wspólny wierzchołek w T. Zapamiętując konstrukcję struktur *info*, można wygenerować strategię przeszukiwawczą dla T w czasie $O(n \log n)$. Wynik ten został poprawiony w [7], gdzie zaproponowano algorytm wyznaczający strategię przeszukiwawczą drzewa w czasie liniowym.

2.2. Przeszukiwanie kaktusów

Niech G będzie dowolnym kaktusem z liczbą przeszukiwawczą k oraz wyróżnionym wierzchołkiem zwanym dalej korzeniem. Mówimy, że G jest **typu (*)**, jeśli istnieje strategia przeszukiwawcza dla G wykorzystująca k agentów taka, że przez cały czas jej trwania jeden z agentów znajduje się w korzeniu. Kaktus może być typu (*) tylko wtedy, gdy korzeń nie jest liściem [3].

Niech graf G' powstaje z grafu G poprzez dodanie do niego wierzchołka stopnia 1 sąsiadującego z korzeniem. Mówimy, że G jest **typu (**)**, gdy nie istnieje taka strategia przeszukiwacza dla grafu G, wykorzystująca k agentów, że jeśli zostanie ona zrealizowana w grafie G', to dodana krawędź pozostanie jedyną skażoną krawędzią w G'.

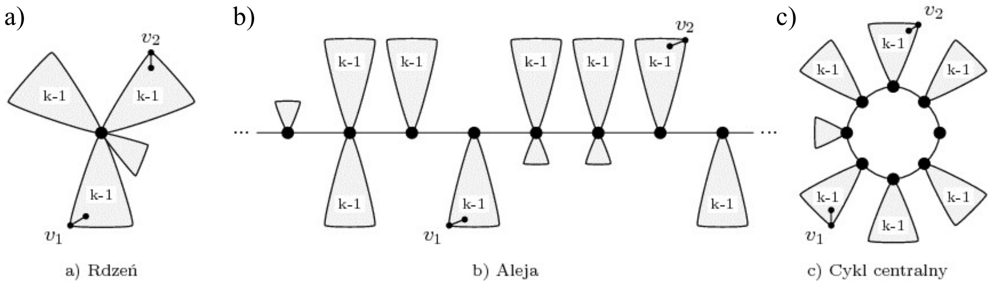
Lemat 2.4. [3] *Jeśli w danym kaktusie G istnieje cykl, którego wszystkie odgałęzienia posiadają liczbę przeszukiwawczą co najwyżej k, to* $k \le s(G) \le k + 2$.

Poniższy lemat nie został dosłownie przytoczony, jednak jest szczególnym przypadkiem jednego z lematów przedstawionych w [3].

Lemat 2.5. *Jeśli w danym grafie G istnieje cykl C taki, że w przynajmniej trzech różnych wierzchołkach istnieją odgałęzienia, których liczba przeszukiwacza wynosi k, to* $s(G) \geq k + 1$.

3. Przeszukiwanie grafów z cyklem centralnym

Zgodnie z obserwacjami przedstawionymi w podrozdziale 2.2, możliwe jest, by kaktus posiadający liczbę przeszukiwawczą k nie miał rdzenia ani alei, jeśli zawiera on cykl spełniający określone warunki. Taki cykl będziemy nazywać **cyklem centralnym**. W każdym kaktusie istnieje albo rdzeń, albo unikatowa aleja, albo cykl centralny.



Rys. 1. Czyszczenie kaktusów za pomocą k+1 agentów
Objaśnienia w tekście

Lemat 3.1. *Niech dany kaktus G posiada liczbę przeszukiwawczą* $s(G) = k$. *Dla dowolnych dwóch liści* v_1 *i* v_2, *incydentnych odpowiednio z krawędziami* e_1 *i* e_2, *możliwe jest przeszukanie kaktusa G za pomocą* k+1 *agentów tak, by* e_1 *została wyczyszczona jako pierwsza a* e_2 *jako ostania.*

Dowód: Łatwo zauważyć, iż analogiczny lemat zakładający wykorzystanie k+2 agentów jest prawdziwy. Pierwszy agent czyści e_1 i zatrzymuje się w wierzchołku sąsiadujący z v_1, drugi zostaje umieszczony w wierzchołku sąsiadującym z v_2. k agentów czyści pozostałą część grafu, a następnie drugi agent przemieszcza się do v_2 czyszcząc e_2.

Jeśli kaktus G posiada rdzeń (rys. 1a), to odgałęzienie w rdzeniu zawierające wierzchołek v_1 posiada liczbę przeszukiwawczą co najwyżej k–1. Można je więc wyczyścić, rozpoczynając od e_1, a kończąc na krawędzi incydentnej z rdzeniem. Następnie można wyczyścić wszystkie odgałęzienia niezawierające e_2. Odgałęzienie w rdzeniu zawierające e_2 posiada liczbę przeszukiwawczą co najwyżej k–1, więc może teraz zostać wyczyszczone tak, by e_2 została wyczyszczona jako ostatnia.

Jeśli G posiada aleję (rys. 1b) można postąpić analogicznie. Najpierw można wyczyścić odgałęzienie w wierzchołku alei zawierające v_1, jako pierwszą krawędź czyszcząc e_1, a kończąc w wierzchołku alei. W wierzchołku tym istnieją co najwyżej dwa odgałęzienia z liczbą przeszukiwawczą k. Jeden agent pozostaje w miejscu, a pozostali czyszczą odgałęzienia niezawierające v_2. Agenci następnie przemieszczają się wzdłuż alei, czyszcząc wszystkie odgałęzienia, aż dotrą do wierzchołka, w którym odgałęzienie zawiera v_2. Liczba przeszukiwawcza tego odgałęzienia jest mniejsza od k. Jeden z agentów pozostaje w tym wierzchołku, a pozostałych k agentów czyści odgałęzienia niezawierające v_2. W końcu czyszczone jest odgałęzienie zawierające v_2, zaczynając od krawędzi incydentnej z aleją, a kończąc na e_2.

Jeśli G nie posiada rdzenia ani alei, to musi posiadać cykl centralny (rys. 1c). Odgałęzienia cyklu centralnego C, posiadają liczbę przeszukiwawczą co najwyżej k–1. W pierwszej kolejności agenci czyszczą odgałęzienie zawierające v_1, kończąc na krawędzi incydentnej z wierzchołkiem cyklu C. Następnie agent a_1 pozostaje w miejscu, a_2 przemieszcza się wzdłuż C, aż dotrze do wierzchołka, w którym odgałęzienie zawiera v_2, a pozostałych k–1 czyści wszystkie odgałęzienia w wierzchołkach odwiedzanych przez a_2. Następnie, w kierunku przeciwnym do a_2, po C przemieszcza się a_1 i wszystkie pozostałe odgałęzienia czyszczone są w sposób analogiczny. W końcu agenci czyszczą odgałęzienie zawierającą v_2, zaczynając od krawędzi incydentnej z cyklem centralnym, a kończąc na e_2. □

3.1. Grafy typu (**)

Lemat 3.2. *Wyczyszczenie grafu G tak, by dana krawędź e została wyczyszczona jako pierwsza, możliwe jest wtedy i tylko wtedy, gdy możliwe jest wyczyszczenie G tak, by e została wyczyszczona jako ostatnia.*

Dowód: Ponieważ rozważamy jedynie strategie monotoniczne, wystarczy zauważyć, że jeśli wszystkie ruchy strategii czyszczącej zostaną wykonane w odwrotnej kolejności, przy czym jeśli w danym ruchu agent przemieszczał się z wierzchołka v_1 do v_2, to przemieści się z v_2 do v_1, to również wykonana zostanie strategia czyszcząca. □

W [3] zdefiniowana została klasa grafów (**). Warto zauważyć, że jeśli korzeń kaktusa G jest liściem, to klasa grafów (**) jest tożsama z klasą grafów, dla których istnieje strategia przeszukiwacza, wykorzystująca $s(G)$ agentów, w której krawędź incydentna z korzeniem jest czyszczona jako ostatnia. Zgodnie z lematem 3.2 nie istnieje również strategia, w której krawędź ta jest czyszczona pierwsza.

Lemat 3.3. *Niech G będzie dowolnym kaktusem, zawierającym przynajmniej jeden wierzchołek stopnia 1. Niech r będzie wyróżnionym korzeniem stopnia* 1. *Jeśli G jest typu* (**)*, to w każdej strategii przeszukiwawczej wykorzystującej* $s(G)$ *agentów, krawędź e incydentna z korzeniem musi zostać wyczyszczona pomiędzy pierwszym a ostatnim momentem, w którym w G znajdzie się* $s(G)$ *agentów.*

Dowód: Zgodnie z dowodem lematu 3.2 wystarczy pokazać, że e nie może zostać wyczyszczona, zanim w G znajdzie się $s(G)$ agentów. Załóżmy, że istnieje strategia przeszukiwawcza P, w której tak się dzieje. Można skonstruować strategię P', w której najpierw czyszczona jest krawędź e, po czym jeden z agentów pozostaje w wierzchołku incydentnym z korzeniem. Pozostałych k–1 agentów kontynuuje strategię P, a od chwili, w której krawędź e została wyczyszczona w P, wszyscy agenci kontynuują strategię P. P' zapewnia wyczyszczenie grafu, więc G nie jest (**). □

Lemat 3.4. *Niech G będzie dowolnym kaktusem, zawierającym przynajmniej jeden wierzchołek stopnia 1. Niech r bedzie wyróżnionym korzeniem stopnia 1. Jeśli G zawiera* *rdzeń, to nie jest typu* (**). *Jeśli G zawiera aleję, to jeśli G jest typu E to nie jest* (**), *a w przeciwnym wypadku G jest typu* (**).

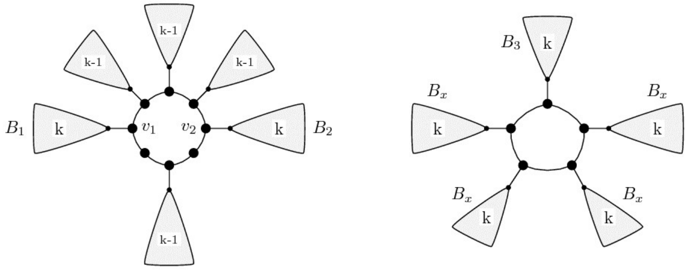
Dowód: Jeśli G posiada rdzeń, lub posiada aleję i jest typu E, to liczba przeszukiwawcza odgałęzienia w rdzeniu lub krańcowym wierzchołku alei, zawierającego r, posiada liczbę przeszukiwawczą mniejszą od $s(G)$. Zgodnie z lematem 3.1 możemy przeszukać je, zaczynając od dowolnej krawędzi i kończąc w rdzeniu lub w krańcowym wierzchołku alei. Wszystkie odgałęzienia w rdzeniu posiadają liczbę przeszukiwawczą mniejszą od $s(G)$. Krańcowy wierzchołek alei posiada dokładnie jedno odgałęzienie z liczbą przeszukiwawczą równą $s(G)$, ale może ono zostać wyczyszczone poprzez przemieszczanie jednego agenta wzdłuż alei i czyszczenie wszystkich odgałęzień w wierzchołkach alei, niezawierających kolejnego wierzchołka alei, przez pozostałych $s(G)–1$ agentów.

W przeciwnym razie, ponieważ korzeń jest stopnia 1, G nie może być typu H ani I. G jest więc typu M. Niech v_i oznacza wierzchołek alei, w którym ukorzeniony jest kaktus zawierający r. Zgodnie z definicją alei w v_i istnieją dokładnie dwa odgałęzienia, których liczba przeszukiwawcza wynosi $s(G)$. W chwili, gdy po raz pierwszy w jednej z nich zostanie rozmieszczonych $s(G)$ agentów, w grafie pojawi się niestrzeżona ścieżka między krawędzią incydentną z r a skażonymi krawędziami w drugim z odgałęzień. Ponieważ rozważamy jedynie strategie bez ponownego skażenia krawędzi, nie jest możliwe wyczyszczenie G, rozpoczynając od krawędzi incydentnej z r. □

3.2. Kaktusy podkubiczne z cyklem centralnym

Graf kubiczny, to dowolny graf regularny stopnia 3. Grafem podkubicznym nazywamy dowolny podgraf grafu kubicznego. W tym podrozdziale rozważamy kaktusy podkubiczne. Należy zauważyć, iż dla tej klasy grafów w każdym wierzchołku cyklu jest co najwyżej jedno odgałęzienie. Dla kaktusów podkubicznych żadne odgałęzienie nie jest typu (*), ponieważ żaden wierzchołek nie należy do dwóch cykli.

W dalszej części k będzie oznaczało największą z pośród liczb przeszukiwawczych odgałęzień cyklu centralnego. $t_1 < t_2 < \ldots < t_r$ oznaczają momenty realizacji strategii przeszukiwawczej, w których k agentów po raz pierwszy znalazło się w odgałęzieniu cyklu w wierzchołkach, odpowiednio, $v_1, v_2, \ldots, v_r$, poza wierzchołkiem należącym do cyklu. Odgałęzienia te są oznaczone jako $B_1, B_2, \ldots, B_r$ i posiadają liczbę przeszukiwawczą k.



Rys. 2. Cykle z dwoma i z pięcioma odgałęzieniami z liczbą przeszukiwawczą k. Odgałęzienia oznaczone jako B_x to dowolna kombinacja odgałęzień B_1, B_2, B_4 i B_5

Lemat 3.5. *Dany jest kaktus podkubiczny G z cyklem centralnym. Jeśli istnieją co najwyżej 2 odgałęzienia cyklu centralnego z liczbą przeszukiwawczą równą k, to* $s(G) \leq k + 1$.

Dowód: Niech agent a_1 zostanie umieszczony w wierzchołku v_1. Pozostałych k agentów czyści B_1. Następnie agent a_2 przemieszcza się wzdłuż cyklu centralnego, aż dotrze do wierzchołka v, w którym znajduje się odgałęzienie z liczbą przeszukiwawczą k, a pozostałych k–1 agentów czyści odgałęzienia w wierzchołkach odwiedzanych przez a_2. Jeśli $v = v_1$, to graf został wyczyszczony. Jeśli nie, to $v = v_2$. a_2 pozostaje w v_2, a a_1 przemieszcza się wzdłuż cyklu centralnego w kierunku przeciwnym do a_2. Pozostałych k–1 agentów czyści wszystkie odgałęzienia w wierzchołkach odwiedzonych przez a_1. Gdy a_1 dotrze do v_2 to pozostaje w nim, a pozostałych k agentów czyści B_2. □

Lemat 3.6. *Dany jest kaktus podkubiczny G z cyklem centralnym. Jeśli istnieje nie mniej niż* 5 *odgałęzień cyklu centralnego z liczbą przeszukiwawczą równą k, to* $s(G) = k+2$.

Dowód: Rozważmy dowolną strategię czyszczącą G, bez ponownego skażenia krawędzi. W chwili t_3 w odgałęzieniach B_4 i B_5 znajduje się przynajmniej po jednej skażonej krawędzi, a w B_1 i B_2 przynajmniej po jednej czystej krawędzi. W G istnieją jednak dwie wierzchołkowo rozłączne ścieżki między $B_1 \cup B_2$ a $B_4 \cup B_5$. W chwili t_3 jeden agent nie może więc zapewnić by nie doszło do ponownego skażenia. Mamy więc $s(G) = k+1$ i zgodnie z lematem 2.4 $s(G) = k+2$. □

Lemat 3.7. *Dany jest kaktus podkubiczny G z cyklem centralnym. Niech istnieją co najmniej* 3 *odgałęzienia cyklu centralnego z liczbą przeszukiwawczą k, a* $s(G) < k+2$. *Niech* t_e *oznacza chwilę, w której wyczyszczona zostaje pierwsza z krawędzi należących do cyklu centralnego, e. Wówczas* $t_e > t_1$.

Dowód: Dowód analogiczny do dowodu lematu 3.6. W G istnieją dwie wierzchołkowo rozłączne ścieżki między $B_2 \cup B_3$ a pierwszą wyczyszczoną krawędzią. □

Tabela 1 przedstawia możliwe wartości liczby przeszukiwawczej dowolnego kaktusa podkubicznego G, w którym istnieje cykl którego odgałęzienia posiadają liczbę przeszukiwawczą co najwyżej k, w zależności od ilości tych odgałęzień. Zawartość pól tabeli wynika bezpośrednio z lematów 2.4, 2.5, 3.5 i 3.6.

Tabela 1
Możliwe wartości $s(G)$, w zależności od ilości odgałęzień cyklu, posiadających liczbę przeszukiwawczą k

Ilość odgałęzień	k	k+1	k+2
1	Możliwe	Możliwe	Nigdy
2	Możliwe	Możliwe	Nigdy
3	Nigdy	Możliwe	Możliwe
4	Nigdy	Możliwe	Możliwe
5+	Nigdy	Nigdy	Zawsze

Literatura

[1] Parsons T.D., *Pursuit-evasion in a graph.*Lecture Notes in Mathematics, 642, 1976, 426.
[2] LaPaugh A., *Recontamination does not help to search a graph.* J. ACM, 40, 1993, 224.
[3] Wrona L., *Scanning Networks with Cactus Topology.* Proceedings of the 1st International Conference on Information Technology, 2008, 301.
[4] Wrona L., *Metody przechwytywania poruszających się obiektów przez mobilnych agentów.* Praca magisterska, Politechnika Gdańska, Wydział ETI, 2006.
[5] Fomin V.F., Thilikos M.D., *An annotated bibliography on guaranteed graph searching*. Theor. Comput. Sci., 399(3), 2008, 236.
[6] Megiddo N., Hakimi S.L., Garey M.R., Johnson D.S., Papadimitriou C.H., *The Complexity of Searching a Graph.* J. ACM., 35, 1988, 18.
[7] Peng S.-L., Ho C.-W., Ko M.-T., Tang C.Y., *Edge and node searching problems on trees.* Theor. Comput. Sci., 240(2), 2000, 429.