

Spis treści

1 Steganografia	1
1.1 Zastosowania steganografii	1
1.2 Steganografia cyfrowa	2
1.3 Steganografia z wykorzystaniem obrazów cyfrowych	3
1.3.1 Techniki przestrzenne	3
1.3.2 Techniki częstotliwościowe	5
1.3.3 Podsumowanie	9
2 Systemy mrówkowe i mrowiskowe	11
2.1 Systemy wieloagentowe	12
2.2 Zastosowania systemów mrówkowych	14
2.3 Zasada działania	16
2.4 Rodzaje systemów mrówkowych	17
2.4.1 Model feromon stały	18
2.4.2 Model feromon średni	18
2.4.3 Model feromon cykliczny	19
2.4.4 System mrowiskowy	20
2.4.5 System mrówkowy Max-Min	21
2.5 Podsumowanie	22
3 Zastosowanie systemów mrówkowych w steganografii	23
3.1 Zastosowanie metod heurystycznych w steganografii	24
3.1.1 Systemy mrówkowe w steganografii	24
3.2 Zaproponowana metoda	26
3.2.1 Założenia	27

3.2.2	Zastosowanie optymalizacji mrówkowej	28
3.3	Sposób reprezentacji problemu oraz interpretacja śladu feromonowego	29
3.3.1	Metoda oparta o wierzchołki	29
3.3.2	Metoda oparta o krawędzie	32
4	Aplikacja steganograficzna wykorzystująca systemy mrówkowe	37
4.1	Specyfikacja zewnętrzna	38
4.1.1	Opis parametrów programu	38
4.1.2	Przykłady użycia	42
4.2	Specyfikacja wewnętrzna	44
4.2.1	Moduł systemu mrówkowego	44
4.2.2	Moduł przetwarzania obrazów	45
4.2.3	Moduł steganograficzny	46
4.3	Walidacja i testowanie	46
4.3.1	Rozwiązywanie problemu komiwojażera	47
5	Wyniki eksperymentów	49
5.1	Metoda badawcza	49
5.1.1	Obrazy	49
5.1.2	Dane	50
5.1.3	Badane parametry	51
5.2	Miary jakości	52
5.3	Wyniki eksperymentów	54
5.3.1	Metoda oparta o wierzchołki	54
5.3.2	Metoda oparta o krawędzie	59
5.4	Ocena subiektywna	67
5.5	Porównanie wyników z innymi metodami	67

Rozdział 1

Steganografia

Steganografia jest dziedziną nauki poświęconą ukrywaniu informacji w jawnych kanałach komunikacji. Nazwa nauki wywodzi się z języka greckiego, i może być tłumaczona jako „ukryte pismo” (*steganós* - ukryty, *graphia* - pismo).

W celu podkreślenia cech oraz charakteru metod steganograficznych często przytaczane jest również pojęcie kryptografii. Obiektem zainteresowań kryptografii jest uniemożliwienie zrozumienia treści wiadomości przez osoby postronne, które nie powinny mieć do niej dostępu. Współcześnie jest to osiągane poprzez stosowanie klucza dzielnego przez osoby zaufane (kryptografia symetryczna) lub par kluczy publicznych i prywatnych (kryptografia asymetryczna). Dzięki ich zastosowaniu, osoba postronna pomimo dostępu do szyfrogramu nie jest w stanie wydobyć tekstu jawnego.

W odróżnieniu od kryptografii, celem technik steganograficznych jest umożliwienie uczestnikom komunikacji przesyłania informacji bez ujawniania faktu istnienia samego przekazu.

1.1 Zastosowania steganografii

Pierwszych przykładów zastosowania steganografii można doszukiwać się już w czasach starożytnych[43]. Herodotus, w swoim dziele *Dzieje* opisuje historię greckiego polityka Histajosa, który w celu przekazania poufnej informacji wytatuował ją na skalpie zaufanego niewolnika. Gdy jego włosy odrosły, został on wysłany w

celu doręczenia listu oraz ukrytej wiadomości, ujawnionej dopiero po jego ostrzyżeniu. Do innych przykładów steganografii można zaliczyć również ukrywanie wiadomości w zapisach nutowych, stosowanie atramentów sympatycznych lub technikę mikrokropek[43], która przeżyła swój renesans w czasach zimnej i drugiej wojny światowej.

Warto również podkreślić, że kolejnym z zastosowań steganografii są znaki wodne oraz symbole pozwalające na identyfikację źródła informacji. Za równo w przypadku multimediów objętych prawami autorskimi jak i poufnymi dokumentów, ich producent lub organizacja strzegąca ich tajności może umieścić ukryte informacje pozwalające wskazać źródło wycieku[37]. Podobną technikę stosują producenci drukarek - seria oraz model drukarki może być odzwierciedlona w drukowanym dokumencie poprzez układ niewidocznych gołym okiem żółtych kropek. Takie działania mają na celu ułatwienia walki z przestępcością polegającą na fałszowaniu dokumentów i banknotów[50].

1.2 Steganografia cyfrowa

Wraz z wzrostem wykorzystania komputerów do celach multimedialnych oraz rozpoznanieniu szerokopasmowego internetu coraz popularniejsza i bardziej opłacalna staje się steganografia cyfrowa. Jej ideą jest wykorzystanie jako medium nośnego różnych rodzajów plików komputerowych lub protokołów komunikacji cyfrowej. Przykładem wykorzystania protokołów do celów steganograficznych może być ukrywanie danych w polach kontrolnych ramek TCP/IP, kontrolowanie opóźnień między poszczególnymi pakietami lub nawet umyślne powodowanie utrat wybranych pakietów[46].

Znacznie prostszą, lecz bardzo rozwiniętą techniką jest wykorzystanie plików multimedialnych takich jak zdjęcia, pliki muzyczne i filmy. Do ich szczególnej atrakcyjności jako medium służące do ukrywania informacji przyczynia się między innymi ich wszechobecność, duże rozmiary oraz wysoka nadmiarowość[34, 45]. Ostatni aspekt w kontekście steganografii ma szczególne znaczenie, gdyż oznacza że modyfikacja pewnej części informacji bitowej zawartej w pliku ma niski wpływ na jego końcową treść. Przykładowo, zmiana wartości jednego z kanałów konkretnego piksela będzie miało mało zauważalny wpływ na końcowy obraz nawet przez

uważnego obserwatora. Podobne prawidłowości można również dostrzec w plikach muzycznych - manipulacja zawartością częstotliwości składowych będących poza granicą percepcji, czyli poniżej 20Hz i powyżej 20kHz również będzie trudna w detekcji przez subiektywnego odbiorcę[61]. Innym przykładem ukrywania informacji w muzyce jest tzw. *backmasking*, polegający na ukrywaniu wiadomości możliwych w odbiorze tylko i wyłącznie po odtworzeniu utworu od tyłu. Jednym z pierwszych zespołów przyczyniających się do wzrostu popularności powyższych eksperymentów jest *The Beatles*.

1.3 Steganografia z wykorzystaniem obrazów cyfrowych

Mimo tego, że jako medium steganograficzne można wykorzystać każdy plik binarny, szczególnie dużo uwagi zostało poświęcone cyfrowym obrazom i zdjęciom. Pomimo pozornej prostoty powyższego zadania powstało wiele wyrafinowanych metod i technik, różniących się za równo pod kątem założeń jak i rezultatów. Pierwszym parametrem mogącym służyć do podziału zaproponowanych metod jest dziedzina w której obraz zostaje poddany analizie.

1.3.1 Techniki przestrzenne

W technikach przestrzennych, obraz jest traktowany jako zbiór punktów (pikseli) umieszczonych w dwuwymiarowym układzie współrzędnych. Zaletą tych metod jest ich intuicyjność oraz przystępność, lecz są to również metody bardziej podatne na ataki polegające na wykryciu lub zniszczeniu ukrytej wiadomości[53].

Najbardziej powszechną techniką przestrzenną jest metoda *Least Significant Bit (LSB)*. Jej zastosowanie sprowadza się do zastąpienia najmniej znaczących bitów obrazu będącego nośnikiem informacji bitami wiadomości ukrywanej. W zależności od spadku jakości który uznajemy za akceptowalny, możemy wykorzystać n najmniej znaczących bitów każdego z kanałów *RGB*. Pewnym uszczegółowieniem *LSB* jest metoda *4LSB*. Zakłada ona wykorzystanie dokładnie 4 bitów z każdego bajtu obrazu maskującego, co przekłada się na wykorzystanie 50% pojemności no-

śnika. Kosztem tak znacznej pojemności jest znaczący spadek jakości obrazu oraz ułatwiona steganoanaliza.

W celu zmniejszenia wykrywalności manipulacji obrazu przy jednoczesnym zachowaniu względnie dużej pojemności steganogramu, zaproponowano technikę *Variable Least Significant Bit (VLSB)*[27]. W przeciwnieństwie do *LSB* podczas ukrywania danych wykorzystywana jest różna ilość bitów obrazu w zależności od położenia piksela. Nośnik zostaje podzielony na zadaną ilość sekcji, a następnie dla każdej z nich wyznaczana jest ilość bitów które zostaną zastąpione tekstem jawnym. Twórcy metody zaproponowali algorytm *Decreasing Distance Decreasing Bits Algorithm (DDDBA)*, który na podstawie odległości sektora względem piksela referencyjnego, którym najczęściej jest środkowy piksel obrazu, wyznacza proporcjonalną ilość ukrywanych bitów. Wynikiem działania algorytmu *VLSB* jest obraz, którego środkowa część jest mniej znieksztalcona, co obniża subiektywne odczucie spadku jakości i pozwala na ukrycie większej ilości danych[27].

Dalszym udoskonaleniem, za równo pod kątem bezpieczeństwa ukrywanej informacji jak i utrudnienia wykrywalności ukrytego przekazu jest metoda o nazwie *Varying Index Varying Bits Substitution (VIVBS)* zaproponowana przez Sahib Khan, N. Ahmad i M. Wahid[26]. Podobnie jak w metodzie *VLSB*, ilość wykorzystanych bitów obrazu nośnego jest zmienna. W przeciwnieństwie do wariantu *VLSB* opartego o algorytm *DDDBA*, ilość bitów ukrywanych w danym pikselu nie jest wyznaczana podczas działania algorytmu, lecz jest zależna od dodatkowego klucza będącego parametrem jego działania. Klucz przyjmuje postać tablicy przypisującej indeksowi każdego z pikseli ilość bitów które należy zastąpić bitami tekstu jawnego. Ponieważ ilość możliwych kombinacji rozmieszczenia bitów informacji w obrazie jest znacząca, odczytanie przekazu poprzez wykorzystanie przeszukiwania wyczerpującego poprzez osobę postronną nieposiadającą klucza będzie praktycznie niemożliwe. Główną wadą, powyżej metody jest jej również największa zaleta - klucz definiujący rozmieszczenie informacji w pikselach obrazu. Każdorazowe kodowanie informacji wymaga utworzenia klucza, od którego będzie również zależeć wpływ procesu na jakość obrazu - arbitralny wybór wysokiej ilości wykorzystanych bitów w nieodpowiednich sekcjach obrazu może przykuć uwagę osób postronnych i zdradzić fakt istnienia ukrytego przekazu. Dodatkową wadę metody jest również rozmiar klucza - w minimalnym przypadku, w którym wykorzystujemy 0 lub 1 bit

każdego piksela klucz ma rozmiar $w * h$ bitów, gdzie w i h to odpowiednio szerokość i wysokość obrazu. Wraz z wzrostem ilości wykorzystywanych bitów, rozmiar klucza również będzie się powiększał[26].

W celu poprawy subiektywnej oceny jakości obrazów oraz zmniejszenia ryzyka przekazu w roku 2003, Da-Chun Wu oraz W. Tsai zaproponowali metodę *Value Pixel Differencing (VPD)*. Jednym z jej założeń jest uzależnienie ilości wykorzystanych bitów obrazu nośnego od różnicy pomiędzy poziomami intensywności kolejnych pikseli[62]. W metodzie *VPD* piksele są odczytywane parami sekwencyjnie, zakreślając ciągły, łamany kształt. Dla każdej napotkanej pary pikseli obliczana jest ich różnica jasności, a następnie na jej podstawie wyznaczana jest ilość bitów które zostaną podmienione na treść ukrywanej wartości. Ilość ukrytych bitów jest proporcjonalna do zmiany wartości pikseli. W ten sposób możliwe jest osiągnięcie obrazów mniej podatnych na steganoanalizę przy jednoczesnym zachowaniu znacznej pojemności[62].

1.3.2 Techniki częstotliwościowe

Alternatywnym podejściem do steganografii wykorzystującej cyfrowe obrazy, są techniki oparte o częstotliwościową reprezentację obrazów. Metody te polegają na transformacji obrazu w postaci bitmapy do macierzy współczynników określających amplitudę lub natężenie fal o konkretnych częstotliwościach występujących w obrazie[9, 49].

Analiza obrazów w dziedzinie częstotliwości daje alternatywną perspektywę na treść obrazu i pozawala na rozpoznawanie, ekstrakcję i manipulację poszczególnych cech które mają odmienny wpływ na jego percepcję. Pasma niskich częstotliwości przekładają się na percepcję ogólnych kształtów i barw obiektów oraz ich kompozycję i wzajemne umiejscowienie. Pasma wysokie pozwalają na rozróżnianie ostrzych krawędzi obrazów, rozpoznawanie detali oraz reprezentują wszelkie złożone tekstury[3].

Metody częstotliwościowe zyskały również na popularności w pokrewnej dziedzinie do steganografii, jaką jest oznaczanie (ang. *watermarking*) produktów cyfrowych. Za równo ukrywając dane poufne, jak i cyfrowe podpisy chroniące praw autorskich, celem metod działających w dziedzinie częstotliwościowej jest nie tylko

zachowanie możliwie najwyższej jakości medium w którym zawarto dodatkowe informacje, lecz również uodpornienie ukrytego przekazu na jego manipulację i zniszczenie przez kompresję[57].

Jedną z transformat pozwalających na wyznaczenie częstotliwościowej reprezentacji obrazu jest *Dyskretna transformata kosinusowa (DCT)*[9]. Polega ona na podziale obrazu na bloki, najczęściej rozmiaru 8x8 pikseli. Następnie wyznaczana jest macierz współczynników $T_{i,j}$ za pomocą poniższego wzoru 1.1. N oznacza rozmiar bloku, w powszechnych zastosowaniach wynosi 8.

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos \left[\frac{(2j+1)i\pi}{2N} \right] & \text{if } i > 0 \end{cases} \quad (1.1)$$

W kontekście złożoności obliczeniowej istotny jest fakt, że macierz transformacji T można wyznaczyć jednokrotnie i ponownie wykorzystać dla każdego z bloków obrazu. W tym przypadku stosowane jest równanie 1.2, w którym wyznacza się macierz współczynników D korzystając z macierzy transformacji T oraz bloku wartości pikseli M przeskalowanych do zakresu $[-128, 127]$.

$$D = TMT' \quad (1.2)$$

Jednym z zastosowań dyskretnej transformacji kosinusowej jest stratna kompresja danych, przykładowo w formacie JPEG. Po wyznaczeniu współczynników odpowiadających kolejnym częstotliwościom następuje proces kwantyzacji - macierz współczynników zostaje skalarnie przemnożona przez macierz kwantyzacji a następnie zaokrąglana. Zadaniem macierzy kwantyzacji jest przeskalowanie współczynników odpowiadających zakresom częstotliwości w taki sposób, aby pasma których zmiany mają najmniejszy wpływ na subiektywną percepcję przyjęły wartości bliskie zeru. Ostatnim krokiem kompresji jest zaokrąglenie uzyskanych współczynników. Poprzez odrzucenie miejsc po przecinku wszystkich współczynników końcowy obraz charakteryzuje się dużo mniejszym rozmiarem. W celu rekonstrukcji wykonuje się transformację odwrotną (*IDCT*).

Przykładem algorytmu steganograficznego korzystającego z *DCT* jest praca „*Reversible Data Hiding in JPEG Images*”[22, 29]. Wyznaczone współczynniki częstotliwości służą za nośnik tekstu jawnego - współczynnikom z eksperymental-

nie dobranych pasm częstotliwości zostają podmienione najmniej znaczące bity, podobnie jak w przestrzennych metodach *LSB*. W celu osiągnięcia dużo większej odporności steganogramu na kompresję, Zhiqiang Zhu, N. Zheng, Tong Qiao oraz Ming Xu zaproponowali metodę opartą o manipulację znaków współczynników, w odróżnieniu do manipulacji ich wartości[66].

Inną transformacją wykorzystywaną w steganografii jest *Dyskretna transformata falkowa* (ang. *Discrete wavelet transform - DWT*) oraz jej odpowiednik pozwalający na bezstratną transformację odwrotną - *Calkowitoliczbową transformata falkową* (ang. *Integer wavelet transform - IWT*)[63].

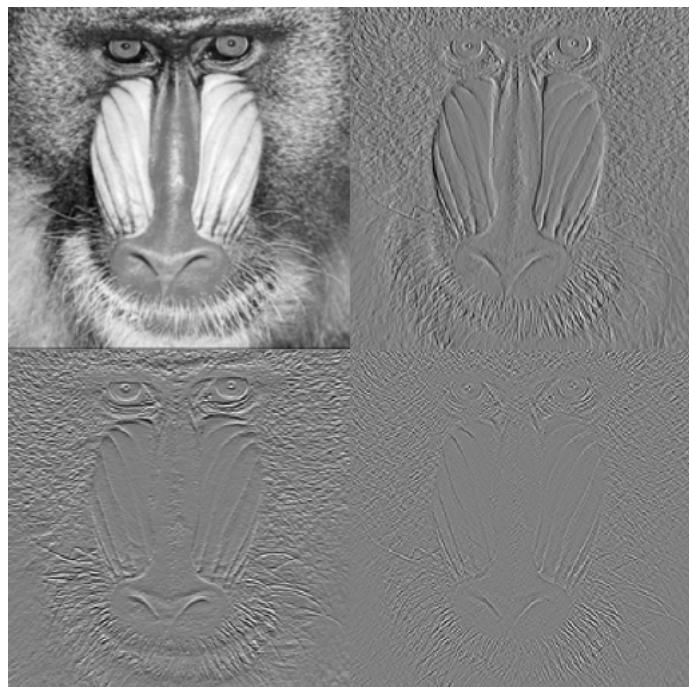
Jedną z możliwości transformaty falkowej jest wyodrębnienie części obrazu w której skład wchodzą osobno wysokie i niskie pasma częstotliwości. Obraz zostaje kolejno przekształcany wierszami i kolumnami przez filtr dolnoprzepustowy (wykonujący operację uśredniania) i górnoprzepustowy (obliczający różnicę).

Po pierwszej iteracji oryginalny obraz zostaje podzielony na sekcje:

- LL - filter dolnoprzepustowy zaaplikowany do wierszy i kolumn,
- LH - filter dolnoprzepustowy zaaplikowany do wierszy, górnoprzepustowy dla kolumn,
- HL - filter górnoprzepustowy zaaplikowany do wierszy, dolnoprzepustowy dla kolumn,
- HH - filter górnoprzepustowy zaaplikowany do wierszy i kolumn.

Przykład działania *IWT* przedstawia rysunek 1.1.

W artykule „*Lossless Data Hiding Using Integer Wavelet Transform and Threshold Embedding Technique*” opisano metodę ukrywania bitów wiadomości w najmniej znaczących bitach współczynników odpowiadających pasmom wysokiej częstotliwości. Wyniki eksperymentalne, względem innych metod, wykazały znaczący wzrost pojemności obrazu przy zachowaniu tego samego współczynnika szczytowego stosunku sygnału do szumu[63].



Rysunek 1.1: Wynik działania *IWT* na przykładowym obrazie. Kolejno sekcje LL, HL, LH, HH. Do wykonania rysunku wykorzystano narzędzie Image Processing Online Demonstration, <http://bigwww.epfl.ch/demo/ip/demos/wavelets/>

1.3.3 Podsumowanie

Na podstawie przytoczonej literatury oraz opisanych metod, można zauważyc pewną tendencję sugerującą istotność doboru regionów obrazu w których są ukrywane dane. W opisanych pracach korzystających z transformat częstotliwościowych, dokonywano wyboru pomiędzy ukrywaniem danych w niskich-średnich[23, 36, 29] lub wysokich pasmach częstotliwości[63, 35].

Główna motywacją autorów decydujących się na dobór niskich bądź średnich pasm w celach ukrywania informacji jest uzyskanie obrazu o większej odporności na manipulacje i zniekształcenia. W pracach opisujących metody wykorzystujące wysokie pasma częstotliwości, wybór jest argumentowany niższą percepcyjną wykrywalnością przez osobę postronną.

Mechanizm ludzkiej percepji obrazów jest niezmiernie złożony a nauki z nią związane pozostają dziedzinami w których pozostaje jeszcze wiele do odkrycia i wyjaśnienia. Na postrzeganie i rozróżnianie obrazów wpływa nie tylko udział poszczególnych składowych częstotliwości, lecz również stosunek kontrastu pasm oraz sposoby uprzedniego przetwarzania obrazu[42]. Niemniej jednak, eksperymentalne badania przeprowadzane na grupie uczestników sugerują większe znaczenie niższych i średnich pasm częstotliwości przy percepji jakości oraz zniekształceń obrazów. Dowodem tego mogą być powszechnie stosowane tablice kwantyzacji wykorzystywane w formatach stratnej kompresji, które faworyzują pasma o średniej lub niższej częstotliwości[9]. Oznacza to, że istnieje większy potencjał w manipulacjach obrazem w zakresie pasm wysokich, przy jednoczesnej zachowaniu jak najwyższej subiektywnej jakości obrazu. Istnieją prace związane z steganografią popierające powyższą tezę. Należą do nich już przytoczona w kontekście metody *PVD* „*A steganographic method for images by pixel-value differencing*”[62], oraz praca „*Edge-based image steganography*” oparta na wykrywaniu krawędzi obrazu[24].

Rozdział 2

Systemy mrówkowe i mrowiskowe

Systemy mrówkowe (ang. *Ant System - AS*) oraz systemy mrowiskowe (ang. *Ant Colony System - ACS*) są metaheurystykami wykorzystywanyymi do rozwiązywania trudnych problemów optymalizacyjnych. Ich fundamenty wywodzą się z pracy z 1991 roku, zaproponowanej przez M. Dorigo, V. Maniezzo i A. Colorni[16]. Jej autorzy przedstawili ideę oraz zastosowania algorytmu wzorującego się na zachowaniu mrówek poszukujących pożywienia. Od tego czasu przedstawiono wiele wariacji oraz ulepszeń algorytmu mrówkowego, a wiele z nich jest obecnie używanych do rozwiązywania problemów w bardzo szerokim zakresie dziedzin.

Ogólna zasada działania systemów mrówkowych opiera się na obserwacji zachowania prawdziwych mrówek eksplorujących otoczenie w celu odnalezienia pożywienia. Początkowo, każda mrówka porusza się w chaotyczny i losowy sposób, przeszukując najbliższe otoczenie mrowiska. W przypadku, w którym mrówka odnajduje pożywienie, zaczyna ona drogę powrotną do mrowiska. Podczas jej przebywania, mrówka niosąca pokarm nanosi na ścieżkę ślad feromonowy. Ma to na celu umożliwienie powrotu do obszaru w którym pokarm został znaleziony. Zatrudnia to mrówka która odniosła zdobytą żywność, jak i pozostałe mrówki w okolicy podczas swojej losowej wędrówki, zaczynają faworyzować trasy oznaczone śladem feromonowym. Jeśli na jego końcu ponownie zostanie znalezione pożywienie, na drogę powrotną zostanie nałożona kolejna jego warstwa. Końcowym efektem tego zjawiska jest efekt pozytywnego sprzężenia zwrotnego, gdyż trasy obierane przez mrówki jednocześnie stają się dla nich bardziej atrakcyjne.

Istotnym aspektem w zjawisku odkładania śladu feromonowego jest jego wyparowywanie pod upływem czasu. Jest to kluczowa właściwość, przyczyniająca się do zbieżności tras obieranych przez mrówki do optymalnej (najkrótszej) drogi prowadzącej do pożywienia. Taki stanu rzeczy może być wyjaśniony poprzez analizę ruchu większej ilości mrówek w pewnym okresie czasu. Ponieważ przebycie dłuższej ścieżki wymaga więcej czasu, średnio mniej mrówek będzie się nią poruszać w stosunku do jednostki odległości. To bezpośrednio przekłada się na mniejszą ilość odłożonego śladu oraz jego szybsze odparowanie. Pomysłodawcy systemów mrówkowych podsumowują ich ogólne właściwości wyszczególniając następujące cechy[16]:

- pozytywne sprzężenie zwrotne pozwalające na szybkie odkrywanie dobrych rozwiązań,
- rozproszony charakter obliczeń zapobiega przedwczesnej zbieżności do lokalnego minimum,
- zachłanne postępowanie każdej mrówki przyczynia się do znajdowania akceptowalnych rozwiązań w bardzo krótkim czasie.

2.1 Systemy wieloagentowe

Ponieważ systemy mrówkowe oraz mrowiskowe można zaliczyć do grupy systemów wieloagentowych (ang. *Multi Agent System - MAS*), istotne jest zrozumienie celów, trudności, zalet oraz ograniczeń tej klasy rozwiązań. Fundamentem systemów wieloagentowych jest pojedynczy agent wchodzący w interakcję z starannie zaprojektowanym środowiskiem. Pomimo braku ścisłej definicji agenta która byłaby w stanie objąć wszystkie istotne przykłady oraz zastosowania systemów, agent musi spełniać następujące kryteria[4].

- Zdolność percepji otoczenia. Sposób postrzegania oraz zakres odbieranych informacji zależy od rozwiązywanego problemu i jest proporcjonalny do poziomu złożoności rozwiązywanego problemu.

- Autonomia. Każdy agent samoistnie dąży do realizacji swoich celów, nie wymaga interakcji z innymi agentami ani zewnętrznej ingerencji człowieka w działanie systemu.
- Responsywność i proaktywność. Agent pod wpływem bodźców odbieranych z środowiska podejmuje decyzje pozwalające przybliżające go do realizacji celu.
- Komunikacja i zachowanie społeczne. Interakcja pomiędzy osobnikami jest kluczowym elementem systemów wieloagentowych. Pomimo że osobniki są w stanie działać autonomicznie, komunikacja pozwala na dzielnie się wiedzą i zdobytym doświadczeniem, co przekłada się na szybsze dążenie do lepszych rozwiązań systemu jako całości.
- Lokalność celu. Agent nie jest w pełni świadomy stanu całego systemu, ani ostatecznego celu jego działania. Przeciwnie, agentom przydzielane jest realizacja lokalnych celów, które są dużo prostsze do osiągnięcia niż globalny cel działania całego systemu. Poprzez interakcję oraz mnogość agentów osiąganie lokalnych celów przekłada się na odkrywanie coraz to lepszych rozwiązań globalnego problemu.

Ze względu na powyższą charakterystykę oraz ogólną koncepcję systemów wieloagentowych, posiadają one wiele zalet których pozbawione są scentralizowane metody rozwiązywania problemów. Autonomia agentów pozwalała na zastosowanie metodyk programowania równoległego i lepszego wykorzystania dostępnych zasobów procesora. Zapewniają również lepszą skalowalność pod kątem rozmiaru problemu - dla większych danych wejściowych możliwe jest uruchomienie większej liczby agentów zaangażowanych w rozwiązanie zadanego problemu. Lokalność i niezależność agentów pozwala na wykorzystanie rozproszonych systemów komputerowych, na przykład klastrów - to z kolei przekłada się na większą niezawodność systemów, gdyż błąd działania pojedynczego agenta nie powoduje awarii całego systemu. Rozproszone systemy wieloagentowe mają również swoje wady, należą do nich narzuć komunikacji agentów który może utrudniać równoległe wykonywanie operacji, oraz brak gwarancji osiągnięcia globalnego celu[17, 4].

W związku z szerokim wachlarzem zalet, systemy wieloagentowe znajdują zastosowanie w rozległym spektrum dziedzin i aplikacji. Ich przeznaczenia można zaobserwować poczynając od modelowania problemów zbyt złożonych do klasycznej analizy, rozwiązywania zadań wyznaczania drogi, za równo w logistycznych łańcuchach zaopatrzeń jak i w trasowaniu pakietów w sieciach IP, oraz zarządzania i monitorowania systemami takimi jak sieci energetyczne lub platformy chmurowe[17, 41].

Na podstawie omówionych cech systemów wieloagentowych, możemy wysnuć wiele paralel pomiędzy nimi i systemami mrówkowymi. Pojedynczymi agentami są mrówki, które postrzegają środowisko w określony i charakterystyczny sposób dla natury problemu. Ich percepcja ogranicza się do postrzegania ich położenia, dostępnych ścieżek oraz śladu feromonowego z nimi związanym. Każda mrówka jest w pełni autonomiczna, gdyż nie wymaga dodatkowej interakcji do wykonywania działań prowadzących do realizacji lokalnego celu. Mrówki przejawiają zachowania społeczne oraz komunikują się za pomocą nanoszonego śladu feromonowego na ścieżki prowadzące do pożywienia. Dzięki tym cechom, agenci realizujący lokalne cele, czyli znalezienie pożywienia, przyczyniają się do wspólnego osiągnięcia celu globalnego, jakim jest wyznaczenie najkrótszej drogi do niego prowadzącej.

2.2 Zastosowania systemów mrówkowych

Ponieważ metaheurystyka systemu mrówkowego jest oparta o zachowanie mrówek poszukujących najkrótszej drogi do pożywienia, oczywiste zdają się być próby zaplikowania jej do problemu komiwojażera, znanego w anglojęzycznej literaturze jako *Travelling Salesman Problem - TSP*. Zadaniem postawionym przed algorytmem poszukującym rozwiązania *TSP* jest znalezienie najkrótszej drogi łączącej wszystkie n miast w taki sposób, że każde miasto zostanie odwiedzone jednokrotnie. Problem komiwojażera jest problemem *NP trudnym*, a asymptotyczna złożoność algorytmu przeszukiwania wyczerpującego wynosi $O(n!)$. *TSP* zawdzięcza swoją popularność prostocie jego opisu, będącej w opozycji do trudności realizacji jego rozwiązania. Jego pierwsza formalna definicja została przedstawiona przez matematyka K. Mengera w roku 1930[32]. Od tego czasu naukowcy z wielu dziedzin starali się zaproponować algorytmy i heurystyki pozwalające na znalezienie

dobrych rozwiązań w czasie wielomianowym. Pomimo braku sukcesu w odkryciu algorytmu pozwalającego na znalezienie optymalnego rozwiązania w czasie wielomianowym, poczyniono istotny postęp w tworzeniu algorytmów skupionych na odkrywaniu akceptowalnych rozwiązań w krótszym czasie.

Jednym z najważniejszych przełomów w badaniach *TSP*, był algorytm zaproponowany przez N. Christofides w roku 1976. Odkryte rozwiązanie gwarantuje znalezienie drogi nie dłuższej od optymalnej o 50% w czasie $O(n^3)$ [11]. Od tego czasu, powstawało wiele alternatywnych rozwiązań, lecz żadne z nich nie zdołało obniżyć górnej granicy długości drogi w znaczący sposób.

Alternatywnym podejściem do problemu komiwojażera oraz innych problemów z klasy *NP trudnych*, które zyskało na popularności jest stosowanie metaheurystyk, czyli ogólnych schematów i metod przeznaczonych do rozwiązywania szerokiej gamy problemów algorytmicznych. Metaheurystyki są najczęściej inspirowane systemami występującymi w naturze, i dają dobre rezultaty w problemach optymalizacyjnych problemów o charakterze losowym i dynamicznym[5]. Do najszerzej stosowanych należą symulowane wyżarzanie (ang. *Simulated Annealing*) [28], algorytmy genetyczne (ang. *Genetic Algorithms*) [19], metody optymalizacji cząsteczkowej (ang. *Particle swarm optimization*) [44] oraz opisywane w tym rozdziale systemy mrówkowe i mrowiskowe. Wszystkie z wymienionych metod były wykorzystywane w rozwiązywaniu problemu komiwojażera[47, 32].

Pomimo niesprzecznej wagi i istotności *TSP*, systemy mrówkowe i mrowiskowe znalazły zastosowanie w wielu innych problemach. Kluczowym etapem decydującym o możliwości i efektywności rozwiązywania zadanego problemu przez system mrówkowy lub mrowiskowy jest wyznaczenie jego odpowiedniej reprezentacji grafowej[16]. Dodatkową zaletą wynikającą z zastosowania tego rodzaju metaheurystyki jest możliwość rozwiązywania problemów dynamicznych, w których warunki ulegają zmianom w trakcie pracy algorytmu. Do problemów rozwiązywanych przez systemy mrówkowe możemy zaliczyć między innymi:

- kwadratowe zagadnienie przydziału (ang. *quadratic assignment problem*) [31, 20],
- harmonogramowanie (ang. *scheduling*) [12, 33],
- marszrutyzacja (ang. *vehicle routing problem*) [8],

- trasowanie w sieciach (ang. *routing*)[10, 6].

2.3 Zasada działania

Ponieważ zachowanie mrówek poszukujących najkrótszej ścieżki do pożywienia najłatwiej i najbardziej intuicyjnie jest analizować w odniesieniu do problemu komiwojażera, zdecydowano się stosować słownictwo oparte o problem poruszania się po mapie złożonej z miast połączonych drogami o znanej długości. Ważne jest jednak mieć na uwadze fakt, że jest to tylko przykład dydaktyczny, a działanie algorytmu można równie trafnie opisać posługując się pojęciami grafu, wierzchołków i krawędzi je łączących.

Pierwszym etapem algorytmu jest umieszczenie mrówek na mapie oraz inicjalizacja struktury reprezentującej ślad feromonowy. Mrówki są przydzielane do miast w losowy sposób, a ślad jest inicjowany wartością τ_0 . Każda z mrówek inicjalizuje strukturę służącą do przechowywania informacji o uprzednio odwiedzonych miastach. Zostaje do niej swoje miasto początkowe. Następnie każda mrówka wybiera drogę prowadzącą z bieżącego miasta i do kolejnego miasta j , pod warunkiem że miasto j nie zostało już odwiedzone. Prawdopodobieństwo każdego przejścia w kroku t jest określone funkcją $P_{ij}(t)$. Po wykonaniu pojedynczego kroku przez każdą mrówkę, następuje proces nakładania śladu feromonowego. Natężenie nakładanego śladu jest określone za pomocą funkcji $\Delta\tau_{ij}(t, t + 1)$. Proces wyboru kolejnego miasta i nanoszenia śladu feromonowego jest powtarzany aż do momentu w którym mrówki odwiedziły już wszystkie miasta, lub został osiągnięty warunek końcowy. Po zakończeniu iteracji, która w przypadku zadań polegających na odwiedzeniu wszystkich miast nazywana jest cyklem, następuje ponowne naniesienie śladu feromonowego. Algorytm jest wykonywany do momentu realizacji ustalonej liczby iteracji lub braku poprawy rozwiązania[16].

Ogólna postać algorytmu może zostać podsumowana w następujący sposób:

1. Umieść mrówki na wierzchołkach grafu.
2. Każda mrówka dokonuje wyboru kolejnego nieodwiedzonego miasta zgodnie z ustaloną funkcją prawdopodobieństwa $P_{ij}(t)$.

3. Aktualizuj ślad feromonowy za pomocą funkcji $\Delta\tau_{ij}(t, t+1)$.
4. Powtórz kroki 2-3 do momentu odwiedzenia wszyskich wierzchołków przez każdą mrówkę
5. Aktualizuj ślad feromonowy za pomocą funkcji $\Delta\tau_{ij}(t, t+n)$.
6. Powtórz kroki 1-5.

2.4 Rodzaje systemów mrówkowych

Pomysłodawcy systemów mrówkowych, M. Dorgio, V. Maniezzo i A. Colorni, zaproponowali trzy wariacje systemu. Każda z nich różni się pod kątem sposobu aktualizacji śladu feromonowego. Są to modele *Ant Density*, *Ant Quantity* oraz *Ant cycle*. W dalszych częściach pracy stosowane są odpowiednio nazwy modelu o feromonie stałym, średnim oraz cyklicznym.

Cechą wspólną powyższych modeli jest metoda wyboru krawędzi w kroku t . Prawdopodobieństwo, że mrówka znajdująca się w wierzchołku i wybierze krawędź prowadzącą do wierzchołka j jest określona wzorem 2.1.

$$P_{ij}(t) = \begin{cases} 0 & j \notin J \\ \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j \in J} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} & j \in J \end{cases} \quad (2.1)$$

gdzie

- J jest zbiorem wierzchołków połączonych krawędzią z wierzchołkiem i , które nie zostały jeszcze odwiedzone przez mrówkę podejmującą decyzję,
- $\tau_{ij}(t)$ jest natężeniem śladu feromonowego krawędzi pomiędzy wierzchołkami i i j w kroku t ,
- η_{ij} jest współczynnikiem widoczności wierzchołka j z perspektywy wierzchołka i . Jego wartość jest obliczana jako odwrotność długości krawędzi $\eta_{ij} = \frac{1}{d_{ij}}$,
- α i β są współczynnikami pozwalającymi kontrolować istotność śladu feromonowego względem widoczności.

Ogólną zasadę śladu feromonowego aktualizacji wyraża wzór 2.2. Współczynnik ρ jest odpowiedzialny za wyparowywanie śladu, co chroni przez nieskończoną jego akumulację. Istotne jest, aby jego wartość była dodatnia, lecz mniejsza niż jeden $0 < \rho < 1$. Wartość $1 - \rho$ mianuje się współczynnikiem wyparowania.

$$\Delta\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (2.2)$$

To co rozróżnia opisywane metody, jest reguła wyznaczania przyrostu śladu feromonowego $\Delta\tau_{ij}(t, t+1)$.

2.4.1 Model feromon stały

W przypadku modelu o feromonie stałym, przyrost jest stałą wartością Q dla krawędzi którą mrówka zdecyduje się przebyć w drodze pomiędzy wierzchołkami i i j . Po zakończeniu kroku lub cyklu, wartości te są sumowane dla każdej z m mrówek - jeśli krawędź została przemierzona przez więcej niż jedną mrówkę, przyrost śladu feromonowego będzie proporcjonalnie większy i jest wyrażony wzorem 2.3.

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \begin{cases} Q & (i, j) \in V_k \\ 0 & w przeciwnym wypadku \end{cases} \quad (2.3)$$

gdzie V_k jest krawędzią wybraną przez k mrówkę w kroku t .

2.4.2 Model feromon średni

W modelu o feromonie średnim (*Ant Quantity*), przyrost śladu feromonowego jest odwrotnie proporcjonalny do długości krawędzi łączącej wierzchołki. Powoduje to dodatkowe faworyzowanie widoczności przy wyborze krawędzi, gdyż krótsze krawędzie będą stawać się bardziej atrakcyjne dla innych mrówek.

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \begin{cases} \frac{Q}{d_{ij}} & (i, j) \in V_k \\ 0 & w przeciwnym wypadku \end{cases} \quad (2.4)$$

2.4.3 Model feromon cykliczny

Ostatnim zaproponowanym wariantem systemu mrówkowego jest model feromonu cyklicznego (*Ant Cycle*). Różni się on znaczco od poprzednich, gdyż feromon jest aktualizowany jednokrotnie w każdej iteracji algorytmu, a nie po każdym kroku. Wyznaczenie nowego śladu następuje po n krokach, gdzie n jest długością cykli pokonywanych przez mrówki. Przyrost śladu feromonowego $\Delta\tau_{ij}(t, t + n)$ jest obliczany dla każdej krawędzi trasy pokonanej przez każdą z m mrówek, i jest odwrotnie proporcjonalny do długości całej trasy. Uzasadnieniem takiego postępowania jest intuicja mówiąca, że znajomość globalnej oceny trasy (jej długości) pozwoli lepiej ocenić każde z należących do niej kroków. Regułę obliczania przyrostu śladu wyraża wzór 2.5. L_k oznacza zbiór krawędzi należących do trasy pokonanej przez mrówkę k .

$$\Delta\tau_{ij}(t, t + n) = \sum_{k=1}^m \begin{cases} \frac{Q}{\|L^k\|} & (i, j) \in L_k \\ 0 & w \text{ przeciwnym wypadku} \end{cases} \quad (2.5)$$

Twórcy powyższych algorytmów z powodzeniem przeprowadzili eksperymenty, których celem było zbadanie ich efektywności w rozwiązywaniu problemu komiwojażera. W przypadku każdego z modeli osiągnięto zadowalające wyniki, lecz najszybszą zbieżność do optymalnych rozwiązań zaobserwowano w modelu z feromonem cyklicznym. Podczas eksperymentów, autorzy uzyskali ówcześnie znane najlepsze rozwiązanie *TSP* dla zbioru Oliver30, lecz nie odnieśli sukcesu w problemach większego rozmiaru - Eilon50 i Eilon75. Jednakże, w artykule wyraźnie podkreślono, że osiągnięcie najlepszych wyników w problemie komiwojażera nie było głównym celem pracy, a problem ten został wybrany jedynie w celach dydaktycznych i porównawczych. Zwrócono jednocześnie uwagę na istotność procesu doboru parametrów algorytmu, takich jak ilość mrówek i współczynnik wyparowania śladu feromonowego. Zaobserwowano, że dla większości rozwiązywanych problemów osiągano najlepsze rezultaty gdy ilość mrówek m jest zbliżona do ilości wierzchołków n grafu reprezentującego problem. Dla wariantu algorytmu z feromonem cyklicznym, najlepsze rezultaty osiągano dla współczynnika odparowania $(1 - \rho)$ bliskiemu 0.5, i wartości α i β będących w stosunku $\frac{\beta}{\alpha}$ zbliżonym do zakresu [2.5, 5] [16].

2.4.4 System mrowiskowy

Znaczącym krokiem naprzód w dziedzinie algorytmów mrówkowych, jest system mrowiskowy zaproponowany przez M. Dorigo, jednego z twórców systemów mrówkowych, oraz L. Gambardella w roku 1997[13]. Tym razem głównym celem pracy było opracowanie zmodyfikowanej wersji algorytmu która mogłaby konkurować z najlepszymi znanyymi algorytmami służącymi do rozwiązywania problemu komiwojażera o dużym rozmiarze wejściowym.

Bazując na doświadczeniu i wynikach eksperymentów przeprowadzonych za pomocą klasycznych systemów mrówkowych, zaproponowano wprowadzenie trzech istotnych zmian w zakresie działania algorytmu.

- Rozszerzenie reguły wyboru krawędzi łączącej miasta i i j o dwa tryby działania - eksplorację i eksploatację. Wybór pomiędzy trybem działania jest czyniony na podstawie wartości zmiennej losowej q przyjmującej wartości z zakresu $[0, 1]$. Jeśli jej wartość jest mniejsza bądź równa wartości parametru algorytmu q_0 , mrówka wybierze krawędź maksymalizującą wartość iloczynu widoczności i natężenia śladu feromonowego - jest to strategia eksploatacji. W przeciwnym razie, kolejne miasto s zostanie wybrana zgodnie z funkcją prawdopodobieństwa zbliżoną do zaproponowanej w systemie mrówkowym. Wzór opisujący wybór kolejnego miasta określa wzór 2.6.
- Wprowadzanie globalnej reguły aktualizacji śladu feromonowego, która jest aplikowana po zakończeniu każdego cyklu algorytmu. Jest to rozszerzenie wariantu systemu mrówkowego typu *Ant cycle*, z tą różnicą że aktualizacji podlegają jedynie krawędzie należące do najlepszej znalezionej trasy. Wpływ na istotność przyrostu śladu jest zależny od osobnego współczynnika odprowadzania α oraz długości najdłuższej trasy L_{gb} . Zależność tą wyraża wzór 2.7 i 2.8.
- Usprawnienie lokalnej reguły aktualizacji śladu feromonowego. Według eksperymentów, globalna reguła nie jest wystarczająca w zapewnianiu akceptowalnych rezultatów. Autorzy testowali różne metody wyznaczania wartości przyrostu śladu feromonowego, w tym metody czerpiące inspiracje z wzmacnianych metod uczenia maszynowego *Q-learning*[60]. Równie dobrze

rezultaty osiągnięto stosując prostszą obliczeniowo regułę opisaną wzorem 2.9 oraz 2.10. Wartość ρ jest współczynnikiem wyparowania niezależnym od α , a V_k oznacza krawędź wybraną przez mrówkę k .

$$s = \begin{cases} argmax_{u \in J} \{\tau_{ij} \eta_{ij}^\beta\} & q \leq q_0 \\ \frac{\tau_{ij} \eta_{ij}^\beta}{\sum_{j \in J} \tau_{ij} \eta_{ij}^\beta} & w \text{ przeciwnym wypadku} \end{cases} \quad (2.6)$$

$$\tau_{ij}(t+n) = (1 - \alpha) \cdot \tau_{ij}(t) + \alpha \cdot \Delta\tau_{ij}(t, t+n) \quad (2.7)$$

$$\Delta\tau_{ij}(t, t+n) = \begin{cases} \frac{1}{L_{gb}} & (i, j) \in L_{gb} \\ 0 & w \text{ przeciwnym wypadku} \end{cases} \quad (2.8)$$

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t, t+1) \quad (2.9)$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \begin{cases} \tau_0 & (i, j) \in V_k \\ 0 & w \text{ przeciwnym wypadku} \end{cases} \quad (2.10)$$

Autorom udało się dowieść, że zaproponowany system mrowiskowy jest w stanie osiągać równie dobre, lub w znacznej części badanych problemów lepsze wyniki niż inne heurystyki. W problemach o rozmiarze 75 i 100 miast, uzyskano lepsze rezultaty niż za pomocą algorytmów genetycznych, elastycznych sieci oraz symulowanego wyżarzania. Dodatkowo, autorzy zgłobili temat i osiągalne rezultaty wykorzystania heurystyki systemu mrowiskowego jako generatora tras wejściowych do algorytmów lokalnej wyczerpującej optymalizacji, takich jak β -opt.

2.4.5 System mrówkowy Max-Min

Alternatywnym ulepszeniem klasycznego algorytmu mrówkowego, jest system mrówkowy max-min (*Max-Min Ant System - MMAS*)[55]. Podobnie jak w przypadku pracy *Ant colony system: a cooperative learning approach to the traveling salesman problem*[13], głównym celem autorów było zaproponowanie wariantu algorytmu mrówkowego który gwarantuje lepsze wyniki dla grafów większych rozmiarów.

Autorzy zaproponowali następujące usprawnienia względem systemu mrówkowego opisanego przez M. Dorgio w 1991 roku.

- Po każdej iteracji algorytmu, jedynie mrówka która przebyła najkrótszą trasę nanosi ślad feromonowy. Istnieją dwie wariacje tej zasady, według pierwszej wybierana jest najkrótsza trasa w bieżącej iteracji (ang. *iteration best*), a w drugiej wybierana jest najlepsza trasa znaleziona w całkowitym czasie działanie algorytmu (ang. *global best*). Wzór opisujący regułę aktualizacji śladu feromonowego jest analogiczny do wzoru wyznaczającego przyrost podczas globalnej aktualizacji w systemie mrowiskowym 2.8.
- W celu uniknięcia stagnacji algorytmu, polegającej na wybieraniu przez wszystkie mrówki tej samej trasy, wartości śladu feromonowego są ograniczone do wartości będących parametrami działania algorytmu $[\tau_{min}, \tau_{max}]$.
- Ślad feromonowy jest inicjalizowany wartością τ_{max} . Uzasadnieniem takiego wyboru jest skłonienie mrówek do śmiajszej eksploracji nieznanych rozwiązań na początku działania algorytmu. Takie działanie początkowo zmniejsza znaczenie widoczności miast podczas wyboru krawędzi.

2.5 Podsumowanie

Pomimo obszerności powyższego porównania istniejących systemów mrówkowych i mrowiskowych, nie jest ono w żadnym stopniu wyczerpujące. Powstało wiele innych rozwiązań charakteryzujących się obiecującymi rezultatami. Należą do nich przykładowo systemy elitystyczne (ang. *Elitist Ant System - EAS*)[14] oraz systemy rankingowe (ang. *Rank Ant System - ASRank*)[7]. Jak można ocenić, dziedzina systemów mrówkowych jest wciąż aktualna w rozwiązywaniu problemów, których rozwiązania są nieosiągalne przy stosowaniu algorytmów wyczerpujących i zachłannych[15].

Rozdział 3

Zastosowanie systemów mrówkowych w steganografii

Jak przedstawiono w poprzednich rozdziałach, zagadnienia steganografii oraz systemów mrówkowych są złożone, nawet gdy są analizowane w izolacji. Obie dziedziny mogą zostać powiązane, jeśli spojrzymy na nie jako parę problemu optymalizacji oraz metaheurystykę mogącą posłużyć do jego rozwiązania.

Wykorzystanie systemów steganograficznych jest bezpośrednio związane z jednoczesną realizacją celów stojących w opozycji. Idealny system steganograficzny, niezależnie od stosowanego medium nośnego, cechuje się jednocześnie dużą pojemnością, odpornością na zakłócenia i zniekształcenia nośnika i niską wykrywalnością istnienia przekazu. Intuicyjnie, lecz również na podstawie wszelkich obserwacji, można zauważać że jednoczesna maksymalizacja ukrywanego przekazu powoduje wzrost cech niepożądanych, takich jak wzrost podatności na ataki steganograficzne. Oznacza to, że kluczowym aspektem wszystkich metod steganograficznych jest zachowanie balansu pomiędzy realizacją tych celów. W zastosowaniach o krytycznej poufności istotniejsze będzie zapewnienie niższej wykrywalności, nawet jeśli się odbędzie kosztem stosunku wiadomości do sygnału. W zależności od wybranej metody, kontrolowanie tego balansu może się to sprowadzać do regulacji parametrów algorytmu lub wyboru nośnika o odpowiednio dużym rozmiarze.

3.1 Zastosowanie metod heurystycznych w steganografii

Trudności związane z optymalizacją procesu nie oznaczają, że należy zaniechać poszukiwań metod pozwalających na zachowanie zadowalającego poziomu każdego z parametrów procesu. W rzeczywistości, zagadnienie wykorzystania metaheurystyk oraz metod uczenia maszynowego do celów steganografii jest aktywną dziedziną odnoszącą ciągłe sukcesy[18, 65, 25, 51, 29]. Jednym z aspektów przemawiających na korzyść stosowania metod heurystycznych do celów steganograficznych jest ich stochastyczny charakter. Podstawowym problemem najprostszych technik steganograficznych, takich jak *LSB*, jest sekwencyjność wyboru fragmentów nośnika informacji. W przypadku ukrywania informacji w obrazach, najoczywistszym działaniem jest kodowanie danych w kolejnych pikselach obrazu. Takie podejście jest wyjątkowo podatne na najprostsze formy ataków, za równo steganograficznych jak i statystycznych. W celu uniknięcia powyższej podatności, możliwe jest stosowanie dodatkowych kluczów instruujących, które i w jakiej kolejności segmenty nośnika należy analizować[2]. Wykorzystanie stochastycznych procesów które zachodzą w znacznej części metaheurystyk, automatycznie rozwiązuje problem przewidywalności umiejscowienia informacji. Gwarantuje również możliwość powtarzanego odtworzenia jego działania poprzez ustalenie ziarna generatora liczb losowych. W takim przypadku, jego wartość stanowi swoisty klucz zwiększający bezpieczeństwo ukrytych danych. Dodatkowym argumentem przemawiającym za słusznością wykorzystania klucza w metodach steganograficznych, jest słynna zasada Kerckhoffsa[21], będąca fundamentem współczesnej kryptografii. Jej treść głosi, że bezpieczeństwo systemu powinno zostać zachowane, nawet jeśli osoba próbująca odkryć poufne informacje zna wszystkie szczegóły jego działania. Gwarantem bezpieczeństwa musi być prywatny klucz. Zasada ta stoi w opozycji do systemów opierających się na niejawności (*ang. Security by obscurity*).

3.1.1 Systemy mrówkowe w steganografii

W związku z zaletami metod heurystycznych, oraz nadziejęmi na znalezienie sposobu optymalizacji przeciwnych cech steganogramów, próby wykorzystania

systemów mrówkowych i mrowiskowych do ukrywania danych w obrazach były niejednokrotnie podejmowane[48, 64, 25]. Najważniejszym aspektem charakteryzującym każdą z opisanych metod jest sposób reprezentacji problemu. W celu zastosowania metaheurystyki systemu mrówkowego, konieczne jest przedstawienie danych wejściowych w postaci grafu. Wybór dotyczący zasady jego budowy ma fundamentalny wpływ na uzyskiwane rezultaty oraz efektywność algorytmu. Kolejnym kluczowym zagadnieniem jest interpretacja rezultatów pracy wirtualnych mrówek. W tej kwestii istnieją conajmniej dwie obierane ścieżki przez eksperymentatorów. Za wynik działania algorytmu można uznać najkrótszą bądź najpopularniejszą ścieżkę obieraną przez mrówki - takie podejście oznacza pozyskanie dyskretnej listy wykorzystanych krawędzi. Alternatywnie, jako rezultat można uznać utworzony ślad feromonowy. Korzystając z tego podejścia, utrzymujemy rozmyty zbiór krawędzi reprezentujących najlepszą ścieżkę - krawędzie częściej należące do krótszych rozwiązań problemu komiwojażera będą bardziej do niego należeć niż ścieżki rzadko obierane. Analiza rozmytego zbioru rozwiązań pozostawia szerszą możliwość interpretacji wyników. Dodatkowo, charakterystyka uzyskiwanego śladu feromonowego jest bardziej podatna na zmiany rodzaju systemu mrówkowego oraz jego parametry.

Systemy mrowiskowe mogą zostać wykorzystane za równo w steganografii przy pomocy obrazów cyfrowych w dziedzinie przestrzennej [64, 25], jak i częstotliwościowej [48]. W artykule „*High capacity and optimized image steganography technique based on ant colony optimization algorithm*”, zaproponowano metodę opartą o przytoczoną metodę całkowitoliczbowej transformaty falkowej (*IWT*). Twórcy opisali algorytm wykorzystujący system mrowiskowy do ukrycia sekretnej informacji w współczynnikach dziedziny transformaty. Przeprowadzone eksperymenty wykazały wysoką skuteczność metody[48].

Pomimo że pozostałe przestudiowane prace oparte są o analizę obrazu w technice przestrzennej, sposób reprezentacji problemu i interpretacji wyników jest zdecydowanie odmienny. W pracy „*Ant Colony Optimization To Enhance Image Steganography*”[64], obrazy zostały podzielone na bloki o rozmiarach 2×2 lub 5×5 . Każdy blok jest interpretowany jako graf, w którym wierzchołkami są piksele, a długościami krawędzi są odwrotności błędu średniokwadratowego spowodowanego przez ukrycie w danym pikselu jednego bitu informacji. Uzyskana przez

system mrówkowy ścieżka wskazuje, w których pikselach należy umieścić informację aby uzyskać najmniejszy wpływ na różnicę między obrazem nośnym a steganogramem[64].

Przykładem pracy wykorzystującej wartości śladu feromonowego naniesionego przez mrówki jest „*Ant Colony Optimization (ACO) based Data Hiding in Image Complex Region*”[25]. W powyższej pracy, mrówki poruszają się po grafie zbudowanym na podstawie bitmapy nośnej. Jego wierzchołkami są piksele, a długościami krawędzi różnice intensywności pikseli przez nie łączone. Dążeniem stosowania takiej reprezentacji jest wykrycie krawędzi oraz złożonych obszarów obrazu, pozwalających na ukrycie większej ilości informacji przy jednoczesnej niższej wykrywalności ingerencji w obraz. Po ukończeniu pracy systemu mrówkowego, ustalana jest wartość graniczna feromonu, dla której piksele uznawane są za należące do złożonego segmentu. W ten sposób, obraz zostaje podzielony na dwa zbiory pikseli. W tych związanych z wartością feromonu przekraczającą wartość graniczną zostaje stosowana technika *LSB*. Pozostałe piksele pozostają niezmienione. Za pomocą wyników eksperymentów udaje się dowieźć, jest to skuteczna metoda. Jej dodatkową zaletą jest możliwość parametryzacji i zmiany sposobu wyznaczania granicznej wartości feromonu, co pozwala na kontrolę pojemności steganogramu kosztem jakości[25].

3.2 Zaproponowana metoda

Przytoczone i opisane metody udowadniają użyteczność i skuteczność wykorzystania systemów mrówkowych w celach steganograficznych. Jednocześnie, przegląd dostępnej literatury sugeruje że nie jest to temat wyczerpany, a jego dalsza eksploracja jest możliwa i niesie nadzieję na odkrycie metod jeszcze lepszych pod kątem pojemności steganogramu i niższej postrzegalności przekazu. Poniżej przytoczono główne założenia i idee dwóch badanych metod wykorzystujących systemy mrówkowe i mrowiskowe do ukrywania informacji w obrazach w celu zapewnienia możliwie najwyższej jakości steganogramu i jego pojemności.

3.2.1 Założenia

Pierwszą decyzją podjętą podczas projektowania rozwiązania problemu było zdecydowanie się na wykorzystanie sekcji obrazów cechujących się większą złożonością, takich jak krawędzie i zróżnicowane tekstury. Istniejąca literatura z dziedziny steganografii sugeruje wyższą podatność na manipulację w takich właśnie obszarach przy zachowaniu jednaczesnej niskiej postrzegalności istnienia przekazu. Przykładem powszechnie przyjętej techniki opartej o powyższą tezę jest opisana w rozdziale 1 metoda *Value Pixel Differencing (VPD)*[62], polecająca na uzależnieniu ilości wykorzystanych bitów od miary różnicy sąsiadujących pikseli. Poparcia powyższej tezy można również szukać w przykładach metod opartych o dziedzinę częstotliwościową obrazu[63].

Na podstawie uzyskanych sekcji obrazu umiejscowionych na spektrum stopnia złożoności, można dokonać procesu ukrywania danych w sposób faworyzujący złożone grupy pikseli. Trafną analogią do wykorzystanej metody jest technika *Variable Significant Bit (VLS)*, będąca rozwinięciem *Least Significant Bit (LSB)* o zmienność ukrywanej informacji w każdym z pikseli. W zaimplementowanym rozwiązaniu, stopień przynależności do złożonego obszaru determinuje ilość bitów obrazu nośnego które zostaną zastąpione bitami ukrywanej informacji. Motywacją reprezentacji obszarów złożonych obrazu jako zbiór rozmyty, odróżniającą ją od dyskretnej metody opisanej w pracy „*Ant Colony Optimization (ACO) based Data Hiding in Image Complex Region*”[25], jest jej większy potencjał na uzyskanie dużej pojemności steganogramu. Decydując się na binarny podział obrazu, rozwiązanie jest ograniczone do wykorzystania dwóch różnych ilości ukrywanych bitów. W szczególnym przypadku opisanym w przytoczonej pracy ilość wykorzystanych bitów w obszarach nie złożonych wynosiła zero. Wadą takiego podejścia, jest niemożliwość odróżnienia i przez to efektywnego wykorzystania obszarów o średniej złożoności. Umiejscowienie złożoności na ciągłej skali, pozwala skuteczniej podejść do problemu wyznaczania ilości bitów ukrytych w konkretnym pikselu obrazu. W takim podejściu, nie tylko jesteśmy w stanie wykorzystać większą liczbę pikseli, lecz również uzyskany steganogram może cechować się niższą wykrywalnością, gdyż obszary w których ukryto dane nie są oddzielone ostrą krawędzią od pozostałych sekcji obrazu.

3.2.2 Zastosowanie optymalizacji mrowiskowej

Etapem procesu, w którym zdecydowano się wykorzystać system mrówkowy jest rozwiązanie problemu oceny złożoności obszarów obrazu. W kontekście opisywanej metody, jest to kluczowe zagadnienie całego procesu, gdyż od niego zależy czy informacje zostaną ukryte w sposób utrudniający ich wykrycie i maksymalizujący pojemność steganogramu. Zaproponowane podejście polega na wykorzystaniu pewnej transformacji obrazu nośnego do postaci grafowej, przeprowadzeniu zadanej ilości iteracji systemu mrówkowego, odczytaniu i odpowiedniej interpretacji pozostawionego śladu feromonowego. Etap interpretacji śladu feromonowego jest ściśle związane z wybraną metodą transformacji obrazu do postaci grafowej, lecz jego ostatecznym celem jest uzyskanie macierzy maskującej o wymiarach odpowiadających wymiarom obrazu nośnego. Wartości współczynników uzyskanej macierzy posłużą do wyznaczenia ilości bitów każdego z pikseli które zostaną zamienione na bity ukrywanej wiadomości. Działanie zaproponowanego algorytmu można podsumować w następujących krokach:

1. Wyznacz grafową reprezentację obrazu nośnego.
2. Wykonaj n iteracji systemu mrówkowego o zadanych parametrach.
3. Odczytaj ślad uzyskany feromonowy.
4. Na podstawie śladu utwórz macierz maskującą K_{xy} o wymiarach obrazu nośnego.
5. W każdym z pikseli zastąp k_{xy} najmniej znaczących bitów bitami informacji.

Proces wydobywania wiadomości zakodowanej w obrazie jest analogiczny do procesu jej ukrywania. W celu odczytania wiadomości należy ponownie wygenerować macierz maskującą i sekwencyjnie odczytać dyktowane przez nią ilości bitów obrazu.

Zastosowanie systemu mrówkowego niesie z sobą dodatkową zaletę w dziedzinie bezpieczeństwa. Uzyskany ślad feromonowy, a co za tym idzie macierz maskująca, jest zależny od i bardzo wrażliwy na zmiany wartości parametrów systemu mrówkowego. Dzięki temu, w hipotetycznym przypadku przejęcia steganogramu przez

osobę postronną i stwierdzeniu samego faktu istnienia ukrytej wiadomości, odczytanie jej treści będzie znaczaco utrudnione. Oznacza to, że parametry systemu mrówkowego stanowią formę klucza.

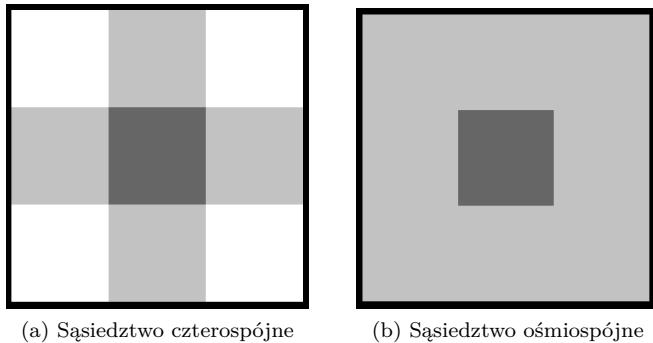
3.3 Sposób reprezentacji problemu oraz interpretacja śladu feromonowego

Jak podkreślono w rozdziale 2 oraz powyższym podsumowaniu zaproponowanej metody, zadanie przekształcenia bitmapy do postaci grafu jest etapem niezbędnym do wykorzystania systemu mrówkowego w celu optymalizacji procesu steganograficznego. Parametry uzyskanego grafu, takie jak ilość wierzchołków i krawędzi będzie miał bezpośredni wpływ na wydajność oraz skuteczność działania całego algorytmu.

Omawiając zagadnienie grafowej reprezentacji obrazu, błędem byłoby pominięcie tematu interpretacji uzyskanego śladu feromonowego. Pomimo pozornej odrębności tych dwóch etapów, są one ściśle z sobą związane. Przyczyną takiego stanu rzeczy jest fakt, że uzyskany ślad feromonowy jest przypisaniem każdej krawędzi grafu pewnej wartości liczbowej. Oznacza to, że interpretacja wartości liczbowej dla każdej krawędzi nie może być oderwana od metody, na podstawie której krawędź została utworzona, a co za tym idzie, również przypisano jej pewne znaczenie. W związku z powyższym, zaproponowane metody budowania grafów będą omówione równolegle z sposobami przekształcenia uzyskanego śladu feromonowego na macierz maskującą.

3.3.1 Metoda oparta o wierzchołki

W pierwszej opisanej metodzie graf (V, E) jest budowany począwszy od wierzchołków. Każdemu pikselowi obrazu wejściowego przypisywany jest jeden wierzchołek tworzonego grafu, co oznacza że zbiór wierzchołków V jest równoliczny zbiorowi pikseli obrazu. Krawędzie grafu E prowadzone są pomiędzy sąsiadującymi z sobą pikselami. Sąsiedztwo może być rozumiane jako czterospójne bądź ośmiospójne, oba rodzaje przedstawia rysunek 3.1.



Rysunek 3.1: Rodzaje sąsiedztwa pikseli

Długość krawędzi E_{ij} łączącej piksele reprezentowane przez wierzchołki V_i oraz V_j wyznaczana jest na podstawie różnicy wartości łącznej intensywności wszystkich kanałów RGB , opisanej wzorem 3.1. Taki sposób doboru długości krawędzi pozwoli „zachęcić” mrówki do poruszania się, a co za tym idzie odkładania śladu feromonowego, po ścieżkach łączących piksele podobne, jeśli ich długość będzie odwrotnie proporcjonalna do różnicy intensywności, lub zróżnicowane jeśli długość będzie proporcjonalna do różnicy.

$$\Delta p_{ij} = \frac{(p_i^r - p_j^r)^2 + (p_i^g - p_j^g)^2 + (p_i^b - p_j^b)^2}{255^2 * 3} \quad (3.1)$$

Tak utworzony graf, nie jest grafem pełnym. Oznacza to, że reprezentowany przez niego problem nie jest równoważny z klasycznym problemem komiwojażera. Kolejną trudnością wystającą w przypadku chęci sprowadzenia tego zadania do TSP jest ilość wierzchołków grafu. Poszukiwanie cykli Hamiltona dla grafu o $w \times h$ wierzchołkach jest conajmniej nieefektywne, gdyż wartość iloczynu szerokości w oraz wysokości h nawet dla małych obrazów osiąga bardzo duże wartości.

W związku z tym, w celu zastosowania powyższej metody wprowadzono pewne zmiany w sposobie uruchamiania i działania systemu mrówkowego. Najważniejsza z nich polega na zrezygnowaniu z końcańcia przez mrówki pełnych cykli. Zamiast tego, ilość wykonywanych kroków w każdej iteracji algorytmu jest kolejnym parametrem algorytmu. W celu usprawnienia działania algorytmu dla obrazów o dużych rozmiarach, wprowadzono możliwość generowania grafu dla obrazu przeskalowanego oraz ponownego przeskalowania macierzy maskującej do oryginalnych

rozmiarów $w \times h$.

Wymusiło to również adaptację części reguł aktualizacji śladu feromonowego. Ponieważ uzyskany graf nie jest pełny, istnieje ryzyko przedwczesnego utknięcia mrówki w pozycji z której nie może wykonać kolejnego kroku, gdyż wszystkie sąsiadujące wierzchołki zostały już odwiedzone. Trasa mrówki która nie zdołała wykonać zadanej ilości kroków będzie nieproporcjonalnie krótsza od tras pozostałych mrówek, co przełoży się na jej niesłuszne faworyzowanie. Powyższym niebezpieczeństwem są dotknięte systemy, które w regule aktualizacji śladu wykorzystują długość całej trasy. Należy do nich cykliczny model systemu mrówkowego (*Ant-cycle*), system mrowiskowy oraz system max-min. W związku z powyższym, zdecydowano się uwzględnić możliwość wystąpienia niepełnych ścieżek i do równań wprowadzono czynnik skalujący dystans trasy do zadanej ilości kroków. Przykładowo, wzór modelu cyklicznego opisanego wzorem 2.5 ma postać 3.2. $|L^k|$ oznacza ilość kroków trasy L^k , a N docelową ilość kroków zadaną podczas uruchamiania algorytmu.

$$\Delta\tau_{ij}(t, t + n) = \sum_{k=1}^m \begin{cases} \frac{Q}{||L^k|| * \frac{|L^k|}{N}} & (i, j) \in L_k \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad (3.2)$$

Ponieważ macierz maskująca musi posiadać wymiary identyczne z obrazem wejściowym, współczynnik na pozycji x, y musi odpowiadać pikselowi p_{xy} . W celu wyznaczenia wartości elementów macierzy, które decydują o ilości bitów które zostaną zastąpione bitami ukrywanej wiadomości, obliczając wartość współczynnika K_{xy} rozpatrywano wszystkie krawędzie wychodzące z wierzchołka grafu któremu odpowiada piksel p_{xy} .

Dylematem, który powstał podczas opracowywania powyższej metody jest decyzja dotycząca zależności pomiędzy różnicami sąsiadujących pikseli Δp_{ij} a długością krawędzi grafu. Początkowe eksperymenty wykazały wyższą skuteczność metody w której długości krawędzi są proporcjonalne do różnicy pomiędzy sąsiadującymi pikselami. Ponieważ oznacza to intensywniejsze i częstsze odkładanie feromonu na krawędziach łączących podobne piksele, musiało to zostać uwzględnione podczas tworzenia macierzy maskującej. Ilość bitów ukrywanych w danym pikselu jest zatem odwrotnie proporcjonalna do natężenia śladu feromonowego.



Rysunek 3.2: Wizualizacja grafu tworzonego metodą opartą o wierzchołki

W związku z powyższym, wartość K_{xy} określa wzór 3.3. A_i oznacza zbiór pikseli sąsiadujących z pikselem i .

$$K_{xy} = 255 * \left(1 - \frac{\sum_{j \in A_i} \tau_{ij}}{|A_i|}\right) \quad (3.3)$$

Rysunek 3.2 przedstawia obraz wejściowy na który nałożono wartości długości krawędzi związanych z konkretnymi pikselami.

3.3.2 Metoda oparta o krawędzie

Drugim zaproponowanym podejściem jest budowanie grafu zaczynając od krawędzi. Pierwszym krokiem jego budowy jest podział obrazu wejściowego na ustaloną liczbę segmentów, znacznie mniejszą niż ilość pikseli bitmapy. Każdy segment obrazu jest reprezentowany przez jedną krawędź grafu. Jej długość jest uzależniona od wariancji wszystkich pikseli należących do segmentu przez nią opisaną. Podobnie jak w przypadku konstrukcji grafu na podstawie wierzchołków, długość krawędzi może być wprost lub odwrotnie proporcjonalna wartości wybranej miary, w tym przypadku wariancji. Uzasadnieniem wyboru wariancji jako miary opisującej każdy z segmentów, jest jej zdolność wyrażenia stopnia odchyлеń pikseli do niego należących. Segmente znajdujące się w złożonych obszarach obrazu będą posiadać

wyzszą wariancję.

Następnie, z uzyskanych krawędzi tworzony jest graf pełny. Aby było to możliwe, konieczne jest spełnienie warunku 3.4. Z tego względu, zaimplementowany algorytm jako parametr wejściowy przyjmuje jedynie docelową ilość węzłów $|V|$, a ilość krawędzi $|E|$, a co za tym idzie segmentów obrazu, jest wyznaczana automatycznie.

$$|E| = \frac{|V| * (|V| - 1)}{2} \quad (3.4)$$

Kolejno, graf zostaje wykorzystany przez system mrówkowy. Zadanie, które jest postawione przed wirtualnymi mrówkami można interpretować następująco: spośród $|E|$ wszystkich krawędzi, wskaż $|V|$ które pozwolą na ukrycie informacji w najbardziej złożonych obszarach obrazu. Jest to zadanie równoważne z problemem komiwojażera, przez co nie muszą być wprowadzane żadne modyfikacje algorytmu tak jak to miało miejsce w metodzie opartej o przekształcenie pikseli w wierzchołki grafu.

Uzyskany ślad feromonowy przypisuje wartość liczbową każdej z krawędzi. Aby na jego podstawie uzyskać macierz maskującą, należy nanieść na piksele należące do segmentu i wartość feromonu związaną z krawędzią E_i . Ponieważ każdy piksel jest przypisany do dokładnie jednego segmentu, wyznaczenie wartości macierzy maskującej nie jest problematyczne.

Implementacja powyższej metody jest nierozerwalnie związana z podziałem obrazu na zadaną ilość segmentów. Wybór techniki nie jest wyborem oczywistym, gdyż istnieje bardzo wiele sposobów na który można takiego podziału dokonać. W związku z powyższym, zdecydowano się wykorzystać trzy różne metody, a następnie porównać ich wyniki. Do wykorzystanych metod należą poniżej opisane.

1. Podział obrazu na pewną ilość nie nachodzących na siebie prostokątów w osi x i y . Konieczny jest wybór takiej liczby podziałów S_x i S_y w osiach x oraz y , aby ich iloczyn był równy $|E|$. W przeciwnym razie, zbudowanie grafu będzie niemożliwe.

Zaletą tej metody jest jej prostota i intuicyjność, lecz ma również kilka wad. W przypadku w którym wymiary bitmapy w i h nie są całkowicie podzielne przez S_x o S_y , konieczne jest zwiększenie segmentów znajdujących się na



Rysunek 3.3: Wizualizacja grafu tworzonego metodą opartą o podział obrazu na nie nachodzące prostokąty

końcach utworzonych wierszy i kolumn. Inną wadą są jednolite krawędzie podziałów - co może przełożyć się na wyraźną różnicę jakości obrazu pomiędzy sąsiadującymi segmentami.

Rysunek 3.3 przedstawia wizualizację podziału obrazu powyższą metodą.

2. Segmentację obrazu można również przeprowadzić wykorzystując algorytmy grupujące. Jednym z nich jest *k-means*. Jego zastosowanie pozwala na podzielenie danych wejściowych na zadaną ilość k grup elementów w sposób minimalizujący odległości pomiędzy elementami należącymi do tej samej grupy[30]. Algorytm pozwala na grupowanie obserwacji wielowymiarowych, lecz w tym przypadku zdecydowano się opisać każdy z pikseli tylko jednym atrybutem - wariancją jego ośmiu najbliższych sąsiadów. Ponieważ grupowanie nie bierze pod uwagę przestrzennego położenia pikseli, piksele należące do tej samej grupy niekoniecznie muszą do siebie przylegać.

Na rysunku 3.4 przedstawiono przykład podziału obrazu na zadaną ilość grup.

3. Ostatnią wykorzystaną metodą segmentacji obrazu jest podział na superpiksele (ang. *superpixels*). Superpiksele są grupami pikseli które cechują zbliżone



Rysunek 3.4: Wizualizacja grafu tworzonego metodą opartą o podział obrazu przy użyciu algorytmu k-średnich

wartości składowych w przestrzeni barw. Zastosowany algorytm *SLIC* polega z przestrzeni *CIELAB*[1]. Popularność superpixeli w dziedzinie przetwarzania obrazów stale rośnie, gdyż pozwalają na uchwycenie najważniejszych cech obrazu przy znacznej redukcji jego reprezentacji. To z kolei przekłada się na krótszy czas pracy algorytmów przetwarzania wizji. Ich zastosowania można doszukiwać się w rozwiązywaniu problemów identyfikacji i wyodrębniania obiektów. W przeciwieństwie do algorytmu *k-means*, uzyskane grupy pikseli są spójne.

Rysunek 3.5 przedstawia wizualizację podziału obrazu powyższą metodą.



Rysunek 3.5: Wizualizacja grafu tworzonego metodą opartą o podział obrazu na superpiksele

Rozdział 4

Aplikacja steganograficzna wykorzystująca systemy mrówkowe

Do implementacji aplikacji umożliwiającej ukrywanie danych w obrazach cyfrowych za pomocą przedstawionych metod wykorzystano język *Rust*. *Rust* jest kompilowanym językiem niskiego poziomu, którego kod wynikowy nie jest uruchamiany za pomocą maszyny wirtualnej. Nie posiada również mechanizmu odśmiecania pamięci (ang. *garbage collection*) ani automatycznego zliczania referencji. Przeciwnie do języków *C* oraz *C++* obowiązek zarządzania pamięcią i gwarantowania bezpieczeństwa jej użycia przeniesiony z dewelopera na rygorystyczny kompilator.

Głównymi celami realizowanymi przez język *Rust* jest umożliwienie pisania aplikacji wymagających dużej wydajności, współprzejności oraz bezpieczeństwa przy wykorzystaniu nowoczesnych praktyk programistycznych. Kolejnym atutem tego języka jest wbudowany manager pakietów *Cargo* umożliwiający bezproblemowe korzystanie z bogatego wachlarza otwartoźródłowych bibliotek.

Rust jest językiem wieloparadygmatowym, lecz wiele cech będących fundamentem jego popularności oraz bezpieczeństwa czerpie z języków funkcyjnych. Umożliwia pisanie wydajnego i deklaratywnego kodu dzięki rozbudowanym iteratorom (analogicznych do przetwarzania strumieniowego z języka *Java*), oraz algebraicznym typom danych. Ważną decyzją podjętą podczas projektowania języka było

zdecydowanie się na wykorzystanie monad *Option* oraz *Result*, co pozwoliło na rezygnację z mechanizmu wyjątków, zapewnienie bezwarunkowego *null safety* oraz umożliwienie kompilatorowi wychwycenia znacznej części najczęściej popełnianych błędów.

Ekspresywny system typów pozwala na pisanie rozszerzalnego i generycznego kodu. Język *Rust* pozbawiony jest mechanizmu dziedziczenia, które nadużywane może być źródłem nieoczekiwanych zachowań oraz błędów związanych z dzieleniem stanu pomiędzy poszczególnymi klasami bazowymi oraz konfliktami nazw. Alternatywnym rozwiązańiem pozwalającym na ponowne wykorzystanie zachowań są cechy (ang. *trait*). Cechy umożliwiają jedynie współdzielenie interfejsu oraz domyślnych implementacji funkcji związanych z typem danych, lecz nie pozwalają na współdzielenie stanu. Są mechanizmem zbliżonym do klas typów (ang. *typeclasses*) znanych z języka *Haskell*.

4.1 Specyfikacja zewnętrzna

Ponieważ funkcjonalność projektu była bardziej znaczącym aspektem niż jego interfejs, program zdecydowano się zaimplementować jako aplikację konsolową. Pomimo wady w postaci mniejszej przystępności dla niedoświadczonego użytkownika, takie rozwiązanie ma również swoje zalety. Umożliwia automatyczne uruchamianie aplikacji dla różnych zestawów danych za pomocą zewnętrznych skryptów i programów, co okazało się bardzo wartościowe podczas ewaluacji rezultatów dla dużej liczby różnych parametrów uruchomieniowych.

4.1.1 Opis parametrów programu

Program działa w trzech trybach, a do każdego z nich są przypisane pewne argumenty. Istnieją też parametry nie związane z wybraną pod-komendą. Należą do nich głównie parametry systemu mrówkowego. Szczegółowy opis każdego z nich zawiera lista:

- **seed** - ziarno generatora liczb losowych, umożliwia powtarzalność wyników.
Domyślnie przyjmuje wartość 42.

- **ants** - liczba mrówek w mrowisku. Jeśli nie zostanie podana, system mrowiskowy zostanie zainicjowany z ilością mrówek równą liczbie wierzchołków grafu.
- **steps** - maksymalna ilość kroków wykonywanych przez mrówki w jednym cyklu. Znajduje zastosowanie jedynie w metodzie budowy grafu opartej o wierzchołki. Domyślnie ilość kroków mrówek jest równa ilości wierzchołków grafu.
- **dispatcher** - rodzaj strategii wyboru krawędzi przez mrówki. Odpowiada metodom opisanyem w sekcji 4.2.1. Dostępne wartości to **basic**, **biased** oraz **colony**. Każda z wartości może następować lista parametrów wybranej strategii. Jeśli lista parametrów nie zostanie podana, zastosowane będą automatycznie dobrane wartości dla wskazanego problemu.
- **updater** - rodzaj strategii nanoszenia śladu feromonowego odpowiadających różnym systemom mrówkowym. Dostępne wartości to **const**, **avg**, **cycle**, **colony** oraz **maxmin**. Podobnie jak w przypadku argumentu **dispatcher**, możliwe jest podanie dodatkowych parametrów do każdej strategii po oddzieleniu ich znakiem :.
- **converter** - strategia przekształcenia bitmapy w graf, które zostały opisane w 3.3. Dostępne wartości to **spatial**, **window**, **kmeans** oraz **superpixels**. Każda z nich może zostać poprzedzona znakami i: aby jednocześnie odwrócić długości wygenerowanego grafu i luminancję każdego z pikseli macierzy maskujących.
- **cycles** - określa ilość iteracji systemu mrówkowego. Jeśli nie jest podana, musi zostać podany argument **stop-after**.
- **stop-after** - określa po ilu iteracjach bez polepszenia rezultatów przeszukiwanie powinno się zatrzymać.
- **mask-width** - pozwala na przeskalowanie obrazu na którego podstawie tworzony jest graf.

- **target-capacity** - umożliwia określenie docelowej pojemności steganogramu. Wykorzystywane podczas eksperymentów i pomiarów degradacji jakości obrazu. Przyjmuje wartości liczbowe z jednostką B, kB lub MB.
- **quiet** - jeśli zostanie podany ten argument, program nie będzie drukował rezultatów na strumień wyjścia.
- **verbose-files** - decyduje czy generowane artefakty (steganogram, macierz konwersji na graf i macierz maskującą) będą posiadały w nazwie znacznik czasu oraz argumenty uruchomienia.

Po podaniu głównych argumentów, użytkownik musi wybrać tryb pracy programu. Należą do nich:

1. **embed** - tryb ukrywania informacji. Wymaga podania argumentów:
 - **image** - ścieżka do obrazu nośnego,
 - **data** - ścieżka do pliku tekstowego zawierającego ukrywaną wiadomość.Rezultatem działania programu w tym trybie są 4 obrazy, które znajdują się w tym samym folderze co obraz nośny. Każdy z wygenerowanych obrazów jest opatrzony przyrostkiem definiującym jego znaczenie.
 - **steg** - utworzony steganogram,
 - **pher** - macierz maskująca w postaci bitmapy,
 - **pher_scaled** - macierz maskująca przeskalowana w taki sposób, aby pomieściła zadaną ilość danych,
 - **conv** - przedstawia proces konwersji bitmapy na graf. Każdy piksel posiada luminancję związaną z odległością na grafie.
2. **extract** - tryb wydobywania informacji z steganogramu. Wymaga podania argumentów:
 - **image** - ścieżka do obrazu nośnego,
 - **steg** - ścieżka do steganogramu.

Odzyskana informacja zostanie wyświetlona na strumieniu wyjściowym procesu.

3. **tsp** - tryb rozwiązywania problemu komiwojażera. Możliwe jest uruchomienie go z następującymi argumentami:

- **n-cities** - powoduje wygenerowanie losowego grafu o zadanej ilości wierzchołków,
- **graph** - ścieżka do pliku *.csv* zawierającego listę współrzędnych opisujących wierzchołki grafu.

Rezultatem pracy tego trybu jest długość najkrótszej ścieżki oraz kolejność odwiedzanych wierzchołków.

Ponadto, użytkownik może się zapoznać z całym wachlarzem opcji uruchomieniowych 4.1 poprzez uruchomienie programu z flagą *-help*.

Listing 4.1: Pomoc programu

```

1 USAGE:
2     stegano-ants [FLAGS] [OPTIONS] --dispatcher <dispatcher> --updater <updater> <SUBCOMMAND>
3
4 FLAGS:
5     -h, --help                  Prints help information
6     -q, --quiet
7     --verbose-files            verbose filenames of output files
8     -V, --version               Prints version information
9
10 OPTIONS:
11    -a, --ants <ants>          amount of ants, by default number of nodes
12    --converter <converter>   converter type in format <type>:<args> [default: i:spatial]
13
14    -c, --cycles <cycles>      number of training cycles
15    -d, --dispatcher <dispatcher> dispatcher definition in format <type>:<args>
16    --mask-width <mask-width> dimension of the pheromone mask, directly affects graph size, height is calculated
17    automatically
18
19    --seed <seed>              rng seed [default: 42]
20    -s, --steps <steps>        number of ant steps in single cycle, by default number of nodes
21
22    --stop-after <stop-after> train until number of cycles does not provide improvement
23
24    --target-capacity <target-capacity> target capacity
25    -u, --updater <updater>   updater type in format <type>:<args>
26
27 SUBCOMMANDS:
28    embed
29    extract
30    help      Prints this message or the help of the given subcommand(s)
31    tsp
32

```

4.1.2 Przykłady użycia

Poniżej zamieszczono przykładowe uruchomienia programu.

- Ukrycie danych w wskazanym pliku przy użyciu systemu mrowiskowego oraz wierzchołkowej metody obrazu.

```

1   stegano-ants \
2       --cycles=10 \
3       --ants=1000 \
4       --steps=100 \
5       --dispatcher=colony \
6       --updater=colony \
7       embed \
8       --data=lorem.txt \
9       --image=photo.bmp

```

- Wydobycie danych z wskazanego steganogramu przy użyciu systemu mrowiskowego oraz wierzchołkowej metody obrazu.

```

1   stegano-ants \
2       --cycles=10 \
3       --ants=1000 \
4       --steps=100 \
5       --dispatcher=colony \
6       --updater=colony \
7       extract \
8       --steg=photo_pher.bmp \
9       --image=photo.bmp

```

- Ukrycie danych w wskazanym pliku przy użyciu systemu max-min z ustalonymi parametrami $\alpha = 1$ i $\beta = 2$ oraz metodą podziału na 50 super pikseli.

```

1   stegano-ants \
2       --cycles=10 \
3       --ants=1000 \

```

```

4      —steps=100 \
5      —dispatcher=biased:1,2 \
6      —updater=maxmin \
7      —converter=superpixels:50 \
8      embed \
9      —data=lorem.txt \
10     —image=photo.bmp

```

4. Ukrycie danych w wskazanym pliku przy użyciu systemu max-min z ustalonymi parametrami $\alpha = 1$ i $\beta = 2$ oraz metodą podziału na 50 superpixeli.

```

1  stegano-ants \
2      —cycles=10 \
3      —ants=1000 \
4      —steps=100 \
5      —dispatcher=biased:1,2 \
6      —updater=maxmin \
7      —converter=superpixels:50 \
8      embed \
9      —data=lorem.txt \
10     —image=photo.bmp

```

5. Poszukiwanie rozwiązania problemu komiwojażera opisanego plikiem *.csv* za pomocą systemu o feromonie cyklicznym. Program zatrzyma działanie po 10 iteracjach bez poprawy rezultatu.

```

1  stegano-ants \
2      —stop-after=10 \
3      —dispatcher=biased \
4      —updater=cycle \
5      tsp \
6      —graph=tsp.csv

```

4.2 Specyfikacja wewnętrzna

Kod źródłowy aplikacji został podzielony w taki sposób, że odzwierciedla poruszane domeny. Powstały trzy główne moduły aplikacji: moduł systemu mrówkowego (*ant_colony*), moduł przetwarzania obrazów (*images*) i moduł steganograficzny (*steganography*).

4.2.1 Moduł systemu mrówkowego

W powyższym module zawarto implementację generycznego systemu mrówkowego, który jest parametryzowany strategiami wyboru krawędzi oraz nanoszenia śladu feromonowego. W ten sposób możliwe było efektywne zaimplementowanie wszystkich testowanych modeli systemów: z feromonem stałym, średnim, cyklicznym, systemu mrowiskowego oraz systemu max-min.

Strategie wyboru krawędzi określone są cechą (ang. *trait*) **AntDispatcher**. Jest to interfejs wymagający od każdej implementującej go struktury zapewnienia funkcji wybierającej kolejną krawędź dla wskazanej mrówki, grafu oraz śladu feromonowego. Cecha posiada również domyślną implementację funkcji odpowiedzialnej za umieszczenie zadanej liczby mrówek na wierzchołkach grafu w sposób losowy.

Strukturami implementującymi cechę **AntDispatcher** są:

- *BasicAntDispatcher* implementująca wzór 2.1 dla $\alpha = \beta = 1$,
- *BiasedAntDispatcher* implementująca wzór 2.1, jest wykorzystywana w wszystkich systemach z wyjątkiem systemu mrowiskowego,
- *ColonyAntDispatcher* implementująca zasadę wyboru krawędzi w systemie mrowiskowym, będącą opisaną wzorem 2.6.

Kolejną kluczową cechą należącą do tego modułu jest **PheromoneUpdater**, opisująca strategie odkładania śladu feromonowego na skutek ruchu mrówek. Interfejs wymaga implementacji funkcji inicjalizacji struktury feromonu, jego aktualizacji po wykonaniu kroku przez wszystkie mrówki oraz po wykonaniu całego cyklu. Dwie ostatnie funkcje mają domyślną postać funkcji identycznościowej.

Powyższa cecha jest implementowana przez następujące struktury:

- *ConstantPheromoneUpdater* nanoszący stała wartość na każdą odwiedzoną krawędź grafu, jest opisany wzorem 2.3,
- *AveragePheromoneUpdater* odpowiada modelowi o feromonie średnim, opisanym wzorem 2.4,
- *CyclicalPheromoneUpdater* aktualizujący ślad dla krawędzi w proporcji do długości trasy do której należy (wzór 2.5),
- *ColonyPheromoneUpdater* nanoszący ślad w każdym kroku algorytmu oraz po zakończeniu cyklu dla krawędzi należących do najkrótszej ścieżki (wzór 2.5),
- *MaxMinPheromoneUpdater* implementujący logikę aktualizacji śladu feromonowego w systemie max-min (wzór 2.8 oraz 2.10).

Kolejnym ważnym elementem powyższego modułu jest struktura śladu feromonowego **Pheromone**. Jej implementacja jest oparta o tablicę asocjacyjną (ang. *HashMap*) przypisującą każdego identyfikatorowi krawędzi wartość zmienoprzecinkową. Zawiera metody umożliwiające skalowanie, inkrementację i normalizację wartości mapy.

Ostatnim istotnym elementem modułu systemu mrówkowego jest struktura reprezentująca graf. Jego implementacja również wykorzystuje tablicę asocjacyjną, której kluczami są identyfikatory wierzchołków a wartościami struktury wierzchołków zawierające listę sąsiadztwa.

4.2.2 Moduł przetwarzania obrazów

Wszelkie operacje związane z odczytem i manipulacją obrazów zostały zawarte w module **images**. Można w nim wyszczególnić strukturę **Image** oraz związane z nią funkcje pozwalające na wczytywanie obrazu z dysku, manipulację rozmiarem oraz przekształcenie do struktury **PixelMap** będącej wygodną abstrakcją nad bitmapami *RGB*. Struktura **PixelMap** pozwala na iterację kolejnych pikseli, iterację pikseli sąsiadujących z wskazanymi współrzędnymi, dowolne przekształcenie każdego piksela za pomocą przekazanej funkcji oraz wyznaczenie negatywu obrazu.

Najbardziej złożoną częścią tego modułu jest cecha **ImageGraphConverter** oraz implementujące je struktury. Jej zadaniem przekształcenie dowolnej instancji bitmapy do grafu oraz przekształcenie śladu feromonowego w macierz maskującą, która jest reprezentowana przezinstancję struktury **PixelMap**.

Do struktur implementujących powyższą cechę należą:

- *SpatialEdgeChangeConverter* implementująca metodę opisaną w sekcji 3.3.1,
- *WindowToEdgeConverter* odpowiedzialną za segmentację obrazu na nienachodzące na siebie prostokąty, opisaną w sekcji 3.3.2,
- *KMeansConverter* dzielącą obraz na grupy pikseli o zbliżonej wariancji sąsiadujących pikseli,
- *SuperPixelConverter* segmentująca obraz na zadaną ilość superpikseli według algorytmu *SLIC*.

4.2.3 Moduł steganograficzny

Ostatnim modułem aplikacji jest moduł odpowiedzialny za ukrywanie informacji w wskazanej bitmapie na podstawie macierzy maskującej. Za równo bitmapa jak i macierz maskująca jest reprezentowana przez struktury **PixelMap**. Strukturą implementującą powyższą funkcjonalność jest **MaskImageEmbedder**. W trakcie ukrywania informacji wykorzystywany jest iterator bitów pozwalający na odczyt zadanej ilości kolejnych bitów wiadomości.

W module zawarto również implementację podstawowych miar jakości obrazów, takich jak błąd średniokwadratowy i szczytowy stosunek sygnału do szumu.

4.3 Walidacja i testowanie

W celu zapewnienia pożdanego działania oraz powtarzalności rezultatów aplikacja została w znacznym stopniu pokryta testami jednostkowymi. Umożliwiło to natychmiastowe wykrycie nieoczekiwanych zmian w działaniu programu oraz umożliwiło dynamiczny rozwój kolejnych modułów aplikacji.

Kolejnym istotnym zagadnieniem związanym z wykorzystanymi metodami zapewnienia jakości oprogramowania są testy oparte o właściwości (ang. *property based testing*). Jest to technika polegająca na testowaniu kodu dla bardzo dużej ilości generowanych losowo danych wejściowych i sprawdzaniu czy pewne fundamentalne założenia dotyczące danych wyjściowych są wciąż spełnione. Przykładowo, test oparty o właściwość funkcji sortującej tablicę zawierającą wartości liczbowe może sprawdzać czy każdy kolejny element posortowanej tablicy jest większy bądź równy poprzedniemu. W ten sposób jesteśmy w stanie zweryfikować działanie kodu bez określania konkretnych zestawów danych.

W aplikacji wykorzystano testy oparte o właściwości do zapewnienia odwracalności procesu ukrywania i wydobywania danych z obrazu (złożenie funkcji ukrywającej i funkcji wydobywającej powinno być funkcją identycznościową z względem na argument ukrywanych danych) oraz zapewniania unikalności generowanych identyfikatorów krawędzi łączących wskazane wierzchołki grafu za pomocą metody zaproponowanej przez M. Szudzik[56].

4.3.1 Rozwiązywanie problemu komiwojażera

Aby zweryfikować poprawność implementacji modułu systemu mrówkowego postanowiono sprawdzić jego efektywność w rozwiązywaniu problemu komiwojażera. Krok ten również umożliwił implementację pomocniczych heurystyk służących do automatycznego doboru parametrów strategii wyboru krawędzi przez mrówki oraz odkładania śladu feromonowego. Uzyskane zależności wyprowadzono na postawie eksperymentów wskazówek znajdujących się w pracach opisujących poszczególne systemy mrówkowe[16, 13, 55].

W tabeli 4.1 zamieszczono uzyskane rezultaty znanych z literatury problemów o trzydziestu (*oliver30*) i stu miastach (*kroa100*), wraz z powszechnie znanimi najlepszymi rozwiązaniami. Parametry poszczególnych wariantów systemów mrówkowych były konfigurowane za pomocą opracowanych heurystyk. Przyjęto również że ilość mrówek jest równa ilości wierzchołków grafu.

Rodzaj systemu	<i>oliver30</i>		<i>kroa100</i>	
	rozwiążanie	iteracja	rozwiążanie	iteracja
<i>Ant Density</i>	446.87012	18	25606.107	30
<i>Ant Quantity</i>	426.54388	185	23150.693	162
<i>Ant Cycle</i>	429.2246	119	22826.52	82
<i>Ant Colony</i>	435.7622	178	23401.607	121
<i>Max-min Ant System</i>	424.69177	84	22075.67	221
Wynik referencyjny	423.741	-	21282	-

Tablica 4.1: Uzyskane rozwiązania problemów *oliver30* oraz *kroa100*.

Rozdział 5

Wyniki eksperymentów

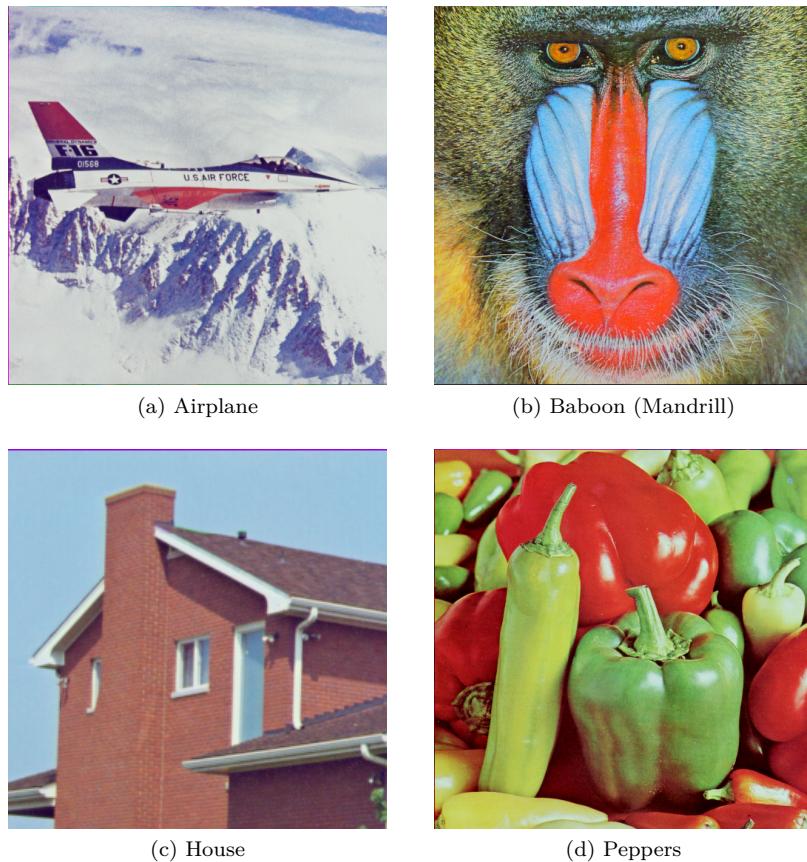
Aby zbadać skuteczność i efektywność metod zaproponowanych w rozdziale 3 oraz podrozdziale 3.3, postanowiono przeprowadzić pewne eksperymenty. Ich celem była walidacja fundamentalnych założeń, takich jak słuszność doboru złożonych sekcji obrazów, sprawdzenie zasadności sposobów wyznaczania reprezentacji grafowej problemu i interpretacji śladu feromonowego, numeryczna ocena degradacji jakości steganogramu oraz subiektywna opinia dotycząca postrzegalności tych zmian. Numeryczna ewaluacja degradacji jakości obrazu pozwoli odnieść uzyskane wyniki do istniejących rozwiązań oraz stwierdzić, czy zaproponowane metody mają swoje zastosowanie.

5.1 Metoda badawcza

Weryfikacja polegała na przeprowadzeniu procesu ukrywania danych w bitmapach powszechnie wykorzystywanych w dziedzinie przetwarzania obrazów.

5.1.1 Obrazy

Pomimo istnienia alternatywnych zbiorów składających się z syntetycznie spoprawionych obrazów mających na celu wyszczególnienie pewnych cech[58], zdecydowano się na wykorzystanie prawdziwych zdjęć, gdyż bardziej odpowiadają one praktycznym zastosowaniom steganografii. W ich doborze kierowano się głównie ilością odniesień w tematycznie powiązanych pracach oraz bieżącymi tendencjami



Rysunek 5.1: Obrazy wykorzystane w eksperymetach.

w kwestii wykorzystywania publicznych zbiorów obrazów[38, 39]. Ostatecznie, zdecydowano się skupić na obrazach *Mandrill* (*a.k.a Baboon*), *Airplane* (*F-16*) (*a.k.a Jet*), *House* oraz *Peppers*, które są udostępniane przez *Uniwersytet Południowej Kalifornii (USC)*[40]. Przedstawione są na rysunku 5.1.

5.1.2 Dane

Dane które ukrywano w obrazach miały postać tekstu w formacie *ASCII*, lecz po drobnych adaptacjach metody ukrywania danych w obrazie możliwe byłoby ukrywanie dowolnych danych w postaci binarnej. W eksperymetach wykorzystano automatycznie wygenerowany tekst *Lorem ipsum* o rozmiarze 625kB. Cechą zaproponowanej metody oraz zaimplementowanego programu jest możliwość

skalowania wygenerowanej macierzy maskującej w taki sposób, aby można było umieścić zadaną ilość bitów tekstu. Pozwoli to na zbadanie degradacji jakości w zależności od objętości ukrywanego tekstu, oraz ułatwi porównanie rezultatów z innymi metodami.

5.1.3 Badane parametry

Podczas przeprowadzanych eksperymentów, badano wpływ poszczególnych metod każdego z etapów procesu oraz ich parametrów. Porównywane wartości parametrów dotyczą poniższych procesów.

1. Konstrukcji grafu oraz wizualizacji śladu feromonowego.
 - Metoda oparta o wierzchołki. Jedynym parametrem jest opcjonalny parametr skalujący s_0 obraz wejściowy podczas budowania grafu oraz budowania macierzy maskującej. Jego wartości znajdują się w zakresie $[0, 1]$. Jego domyślna wartość wynosi 1, i oznacza budowanie grafu o liczbie wierzchołków równej $w \times h$.
 - Metoda oparta o krawędzie. Do jej parametrów należy algorytm segmentacji oraz docelowa liczba segmentów N_s związana z ilością krawędzi grafu. Podczas badań wykorzystano algorytmy prostego podziału na prostokąty, algorytm k-średnich oraz algorytm *SLIC* służący do konstrukcji super pikseli.
2. Wyznaczenie śladu feromonowego przez różne rodzaje systemów mrówkowych. Do parametrów wspólnych dla każdego z wykorzystanych systemów należy:
 - ilość mrówek A , która domyślnie jest równa ilości wierzchołków grafu V ,
 - liczba wykonanych cykli C
 - liczba kroków wykonywanych przez mrówki w każdej iteracji algorytmu S , dla grafów skonstruowanych na podstawie segmentacji obrazu wynosi jest ona równa ilości wierzchołków V

- preferencja względem śladu feromonowego α
- preferencja względem widoczności wierzchołka β
- początkowe natężenie śladu feromonowego τ_0
- współczynnik opisujący prędkość wyparowania śladu feromonowego ρ

Do porównanych rodzajów systemów należą poniższe odmiany. Z niektórymi z nich związane są dodatkowe parametry oraz domyślne wartości powyższych atrybutów.

- model feromon stały,
- model feromon średni,
- model feromon cykliczny,
- system mrowiskowy, który wprowadza prawdopodobieństwo eksploatacji najkrótszej krawędzi q_0 oraz ustalona parametr $\alpha = 1$,
- system max-min, który wprowadza ograniczenia wartości śladu feromonowego $[\tau_{min}, \tau_{max}]$. Ich wartości wyznaczane są za pomocą estymaty dotyczącej długości poszukiwanego cyklu. Wartość początkowa feromonu wynosi $\tau_0 = \tau_{max}$.

5.2 Miary jakości

W celu umożliwienia porównywania jakości steganogramów, zdecydowano się skorzystać z metryk służących do porównawczej analizy obrazów. Zastosowane metryki można podzielić na dwie kategorie, obiektywne i subiektywne. Metryki obiektywne służą do wyznaczenia pewnej wartości charakteryzującej różnicę pomiędzy dwoma sygnałami. W przypadku metryk subiektywnych, ich zadaniem jest również wyznaczenie wartości numerycznej, lecz nacisk kładziony jest na korelację wartości z postrzegalnością zmian przez ludzki układ wzrokowy. Do zastosowanych metryk należą:

- Błąd średniokwadratowy (ang. *Mean Square Error, MSE*).

Jest jedną z najprostszych metryk służących do pomiaru różnic między obrazami. Jej wartości należą do zbioru nieujemnych liczb rzeczywistych, a wartości bliższe zeru stanowią o mniejszym spadku jakości. Do jej zalet należy prostota implementacji i możliwość optymalnej implementacji. Jedną z jej wad jest niska korelacja z postrzeganiem różnic między obrazami przez ludzi oraz nieuwzględnianie informacji o relacji natężenia szumu do wartości sygnału. Jej wartość wyraża wzór 5.1, w którym \bar{X} oznacza wartość średnią.

$$MSE = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \quad (5.1)$$

- Szczytowy stosunek sygnału do szumu (ang. *Peak Signal Noise Ratio, PSNR*).

Jest udoskonaleniem błędu średniokwadratowego, gdyż metryka ta uzależnia swoją wartość od maksymalnej wartości przyjmowanej przez sygnał. Oznacza to, że taka sama wartość *PSNR* odpowiada różnicom będącym w proporcjonalnym do ilości informacji. Przykładowo, w przypadku *MSE* ta sama wartość będzie przekładać się na różną postrzegalność błędu w obrazie korzystającym z 8 i 24 bitów na kanał. Szczytowy stosunek sygnału do szumu rozwiązuje powyższy problem. W związku z dużym zakresem przyjmowanych wartości metryka korzysta z skali logarytmicznej. Wartościami typowymi przy analizie obrazów korzystających z 8 bitów na jeden kanał jest zakres [30dB, 50dB], przy czym wyższa wartość oznacza mniejszą degradację. Wartość metryki można wyznaczyć za pomocą wzoru 5.2.

$$PSNR = 10 \cdot \log_{10} \frac{MAX^2}{MSE} \quad (5.2)$$

- Indeks podobieństwa strukturalnego (ang. *Structural Similarity Index, SSIM*).

Zdecydowanie bardziej złożoną metryką jest *SSIM*. Jej celem jest uchwycenie złożoności i właściwości postrzegania ludzkiego systemu wzrokowego. Opiera się na założeniu mówiącym o istotności struktury obrazu w odniesieniu do postrzeganych różnic luminancji oraz kontrastu[59, 52]. Jej wartość jest zwykle normalizowana do zakresu [0, 1], gdzie wartość 1 oznacza identyczność obrazów.

5.3 Wyniki eksperymentów

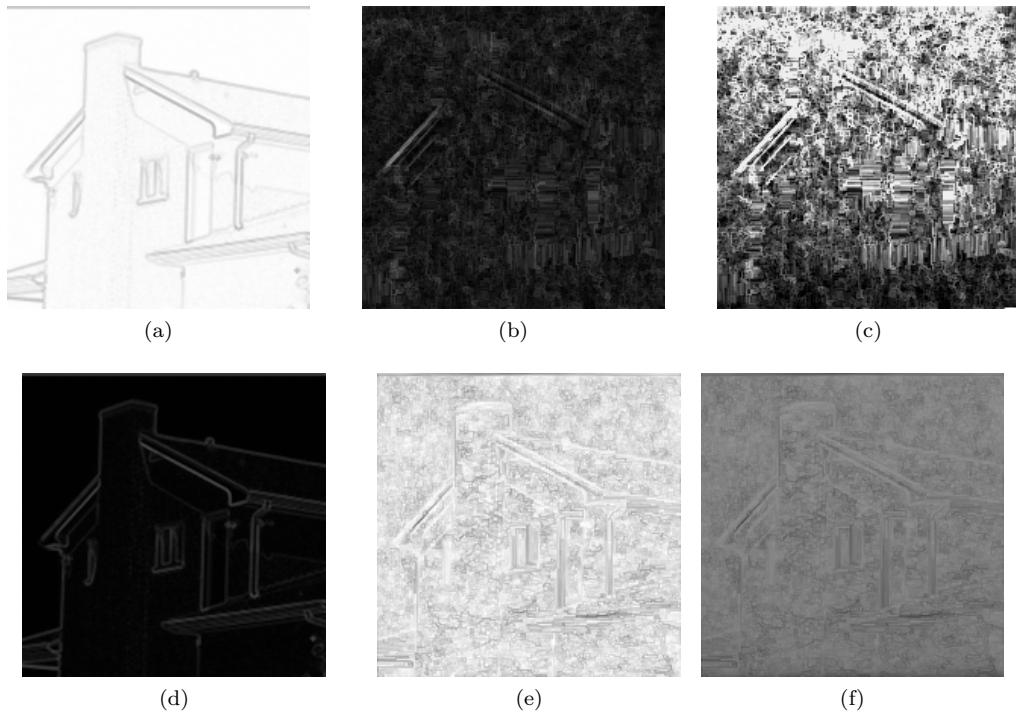
Eksperymenty rozpoczęto od analizy wyników uzyskanych metodą opartą o budowanie grafu na podstawie wierzchołków.

5.3.1 Metoda oparta o wierzchołki

Eksperymenty rozpoczęto od porównania wpływu proporcjonalności długości krawędzi grafu od różnicy pomiędzy pikselami. Jak wspomniano w rozdziale 3.3, jeśli długości krawędzi są proporcjonalne do różnicy pikseli, system mrówkowy będzie dążył do odkładania śladu feromonowego w obszarach cechujących mniejszą złożoność. Aby uzyskać zamierzony rezultat, czyli macierz maskującą wskazującą obszary bardziej złożone, konieczne będzie odwrócenie jej wartości. Alternatywnie, jeśli długości pomiędzy pikselami będą odwrotnie proporcjonalne do dzielącej je odległości, system mrówkowy, a co za tym idzie ślad feromonowy, będzie preferował obszary bardziej złożone.

Wizualizację procesu konwersji obrazu na graf oraz uzyskane macierze maskujące przedstawia rysunek 5.2. W pierwszej kolumnie umieszczone obrazy wyjaśniające wynik konwersji bitmapy na graf, w których do poszczególnych pikseli obrazu wejściowego przypisana jest luminancja odwrotnie proporcjonalna do długości krawędzi z nim związanych (obszary „atrakcyjniesze” dla mrówek cechujące się jaśniejszym kolorem). Kolumna druga przedstawia macierze maskujące, które będą decydować o ilości bitów które zostaną zamienione na bity ukrywanej wiadomości. Im jaśniejszy jest dany piksel, tym więcej jego bitów zostanie zamienione. W trzeciej kolumnie przedstawiono macierze maskujące przetransformowane w taki sposób, aby odpowiadające im pojemność ukrywanej wiadomości była taka sama.

Na podstawie obrazów 5.3 oraz 5.2e można stwierdzić, że obie metody poprawnie wykrywają krawędzie i obszary złożone, lecz po porównaniu macierzy maskujących o tej samej pojemności należy stwierdzić lepszy rezultat procesu w którym długości krawędzi grafu są proporcjonalne do różnicy pomiędzy pikselami, a macierz maskująca zostaje odwrócona. Przeskalowana macierz maskująca 5.2f gwarantuje bardziej równomierne rozłożenie informacji przy jednoczesnym faworyzowaniu krawędzi o obszarów złożonych.



Rysunek 5.2: Porównania wizualizacji konwersji oraz macierzy maskujących. W pierwszym wierszu ((a), (b), (c)) przedstawiono obrazy związane z procesem o długości krawędzi odwrotnie proporcjonalnej do różnicy pikseli. W drugim wierszu ((d), (e), (f)) długości są proporcjonalne do różnicy pikseli. Pierwsza kolumna przedstawia wizualizacje procesu konwersji bitmapy na graf, druga przestawia uzyskane macierze maskujące, trzecia zawiera macierze maskujące przeskalowane w taki sposób, aby odpowiadała im taka sama pojemność ukrywanej informacji.

Kolejnym parametrem który postanowiono zbadać, jest rodzaj systemu mrówkowego oraz związane z nim parametry. Ponieważ problem rozwiązywany przez mrówki nie jest równoważny z problemem komiwojażera, konieczne jest również ustalenie parametru S określającego ilość kroków wykonywanych przez każdą mrówkę w każdej iteracji algorytmu.

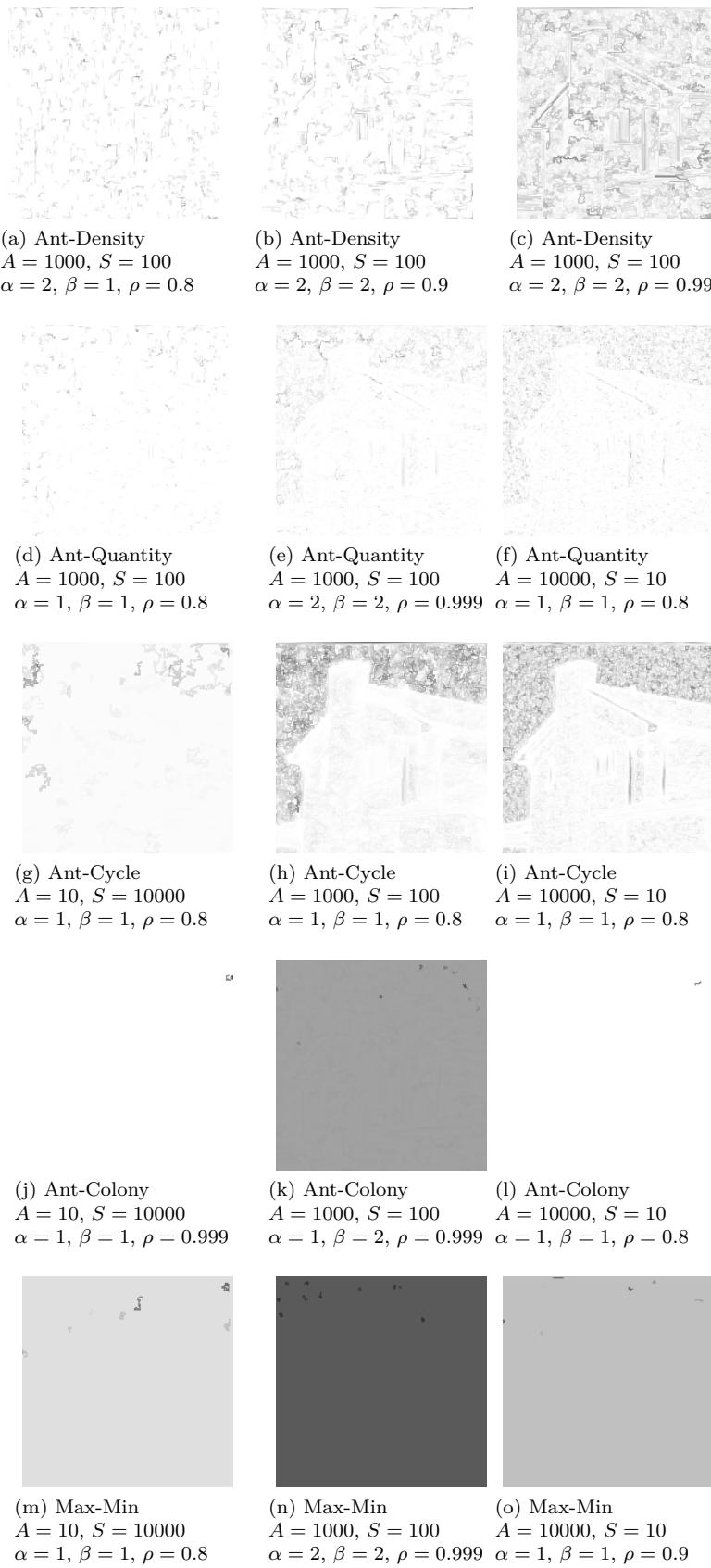
Wygenerowane macierze maskujące przedstawia rysunek 5.3. Na ich podstawie można wyciągnąć kilka interesujących wniosków. System mrowiskowy (*Ant-Colony*) oraz system *Max-min* nie odpowiadają charakterystyce zadанego zadania, które jest znaczco różne od klasycznego problemu *TSP*. Przyczyn można doszukiwać się w silnej faworyzacji najlepszego rozwiązania, czyli najkrótszej obranej ścieżki kosztem pozostałych ścieżek. Można wnioskować, że w przedstawionym problemie znalezienie najkrótszej ścieżki łączącej $S + 1$ pikseli spośród $w \times h$ pikseli nie przyczynia się do ogólnego wyznaczenia obszarów obrazu.

Na dodatkową uwagę zasługuje system z cykliczną regułą aktualizacji śladu feromonowego, gdyż za jego pomocą udało się wyraźnie wyodrębnić jednorodny obszar nieba oraz pewne homogeniczne elementy budynku. Kolejną obserwacją płynącą z powyższego eksperymentu jest istotność doboru ilości mrówek A oraz współczynnika związanego z odparowaniem śladu ρ . Macierze lepiej odzwierciedlające faktyczne segmenty obrazów powstały w przypadku konfiguracji charakteryzujących się większą ilością mrówek (na przykład 5.3i) lub wyższym współczynnikiem ρ (na przykład 5.3a), przekładającym się na wolniejsze tempo odparowania.

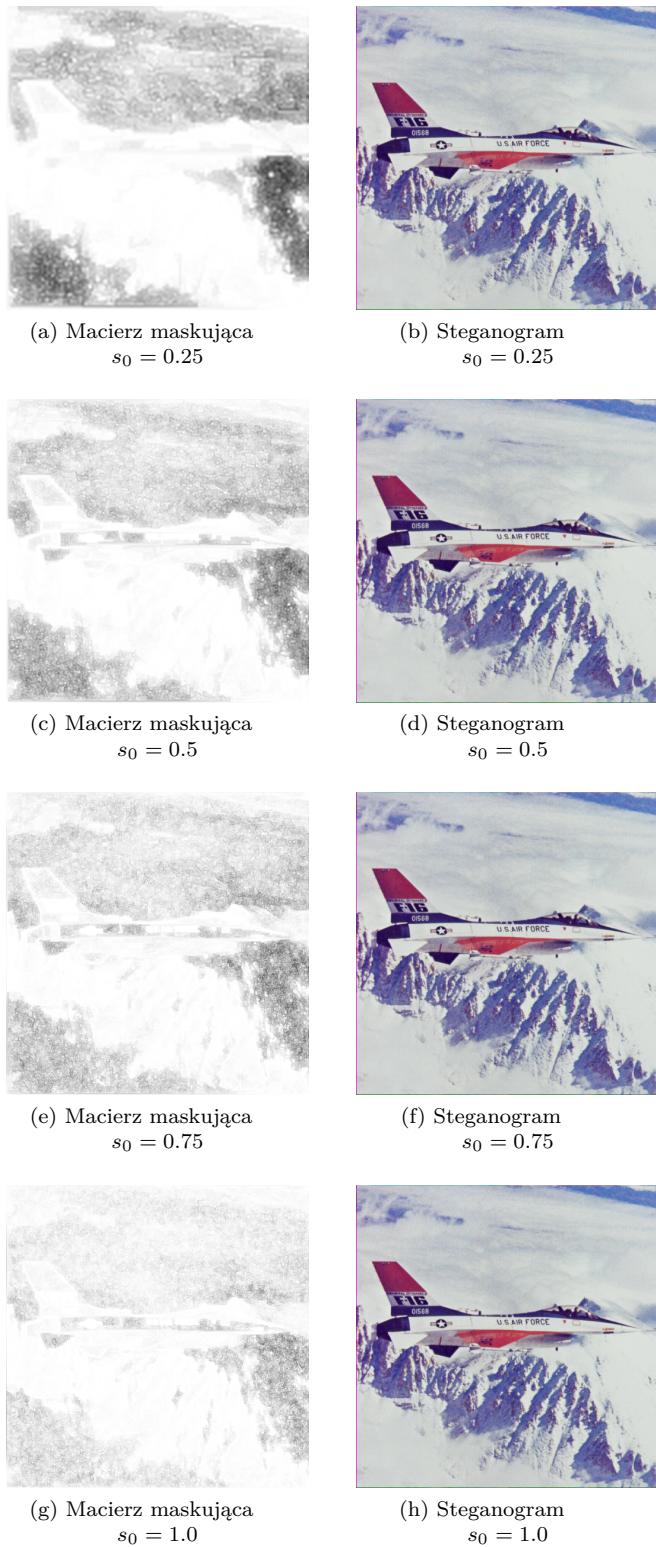
Ponieważ pozostałe obrazy z zbioru *USC* mają rozmiary 512×512 , zdecydowano się również zbadać wpływ skalowania obrazu podczas tworzenia grafu oraz macierzy maskującej. Wykorzystano system oparty o model cykliczny o parametrach $A = 10000$, $S = 100$, $\alpha = \beta = 1$, $\rho = 0.8$, i starano się umieścić $250kB$ danych. Na rysunku 5.4 zamieszczono macierze maskujące powstałe przy $s_0 \in \{0.25, 0.5, 0.75, 1.0\}$. Tabela 5.1 zawiera porównanie miar jakości steganogramów.

Na postawie uzyskanych wyników można stwierdzić że tworzenie grafu na podstawie skalowanego obrazu nie ma znaczącego wpływu na miary jakości steganogramu. Jest to pożądana cecha, ponieważ operacje na mniejszych grafach zajmują zdecydowanie mniej czasu.

Ostatecznie, postanowiono zbadać wyniki uzyskane na pozostałych obrazach



Rysunek 5.3: Macierze maskujące wygenerowane przy pomocy różnych systemów mrówkowych.



Rysunek 5.4: Macierze maskujące oraz steganogramy wygenerowane dla różnych wartości skali s_0 .

Współczynnik skalający s_0	<i>MSE</i>	<i>PSNR</i>	<i>SSIM</i>
0.25	6.93092	39.7568dB	0.952935
0.5	6.67252	39.9218dB	0.951723
0.75	6.62924	39.9501dB	0.95122
1.0	6.61124	39.9619dB	0.948736

Tablica 5.1: Miary jakości steganogramów w zależności od współczynnika s_0 .

Obraz	objętość danych	<i>MSE</i>	<i>PSNR</i>	<i>SSIM</i>
Baboon 512×512	100 000B (12.71%)	0.538851	50.85dB	0.998768
Baboon 512×512	250 000B (31.78%)	6.60594	39.9654dB	0.990015
Baboon 512×512	500 000B (63.57%)	272.732	23.8074dB	0.777025
Airplane 512×512	100 000B (12.71%)	0.53689	50.8659dB	0.995503
Airplane 512×512	250 000B (31.78%)	6.67252	39.9218dB	0.951723
Airplane 512×512	500 000B (63.57%)	311.43	23.2311dB	0.516503
Peppers 512×512	100 000B (12.71%)	0.536934	50.8655dB	0.996048
Peppers 512×512	250 000B (31.78%)	6.71828	39.8922dB	0.958181
Peppers 512×512	500 000B (63.57%)	292.99	23.4962dB	0.502009

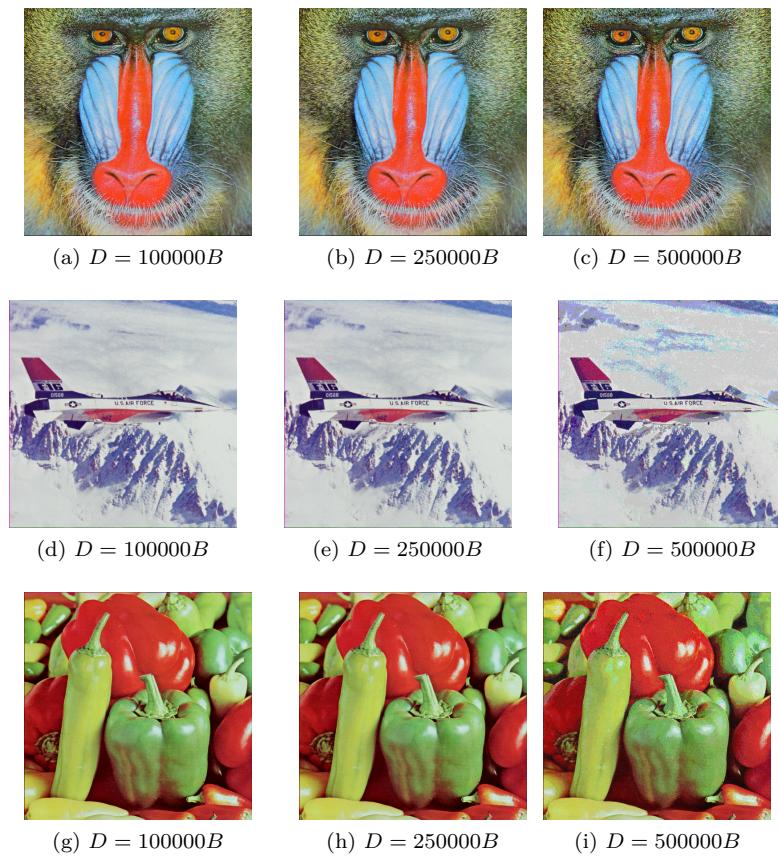
Tablica 5.2: Wyniki uzyskane metodą opartą o wierzchołki.

testowych. Uzyskane rezultaty zawiera tabela 5.2 a steganogramy rysunek 5.5. Parametry algorytmu zachowano z poprzedniego eksperymentu.

5.3.2 Metoda oparta o krawędzie

Następnym krokiem eksperymentów było zbadanie metod opartych o budowanie grafu z wyznaczonych krawędzi. W przypadku tej metody, obraz jest dzielony na zadaną ilość segmentów, które odpowiadają krawędziom tworzonego grafu pełnego. W związku z koniecznością budowy grafu pełnego, istotnym ograniczeniem metody jest konieczność takiego doboru ilości segmentów aby spełniała one równanie wyrażające ilość krawędzi E w grafie pełnym o V wierzchołkach $E = \frac{V \times (V-1)}{2}$. Problem komiwojażera osadzony na tak utworzonym grafie można interpretować jako problem znalezienia n segmentów obrazu tworzących najkrótszy cykl. Długości krawędzi tak powstałego grafu są zależne od różnicy wariancji łączących przez nie segmenty.

Podobnie jak w przypadku metody opisanej w sekcji powyżej, długości kra-



Rysunek 5.5: Porównanie uzyskanych steganogramów przy zadanej objętości ukrytej informacji.

węzły grafu mogą być proporcjonalne lub odwrotnie proporcjonalne do wyznaczonej miary odległości. Z tego względu, postanowiono rozpatrzyć ten parametr jako pierwszy.

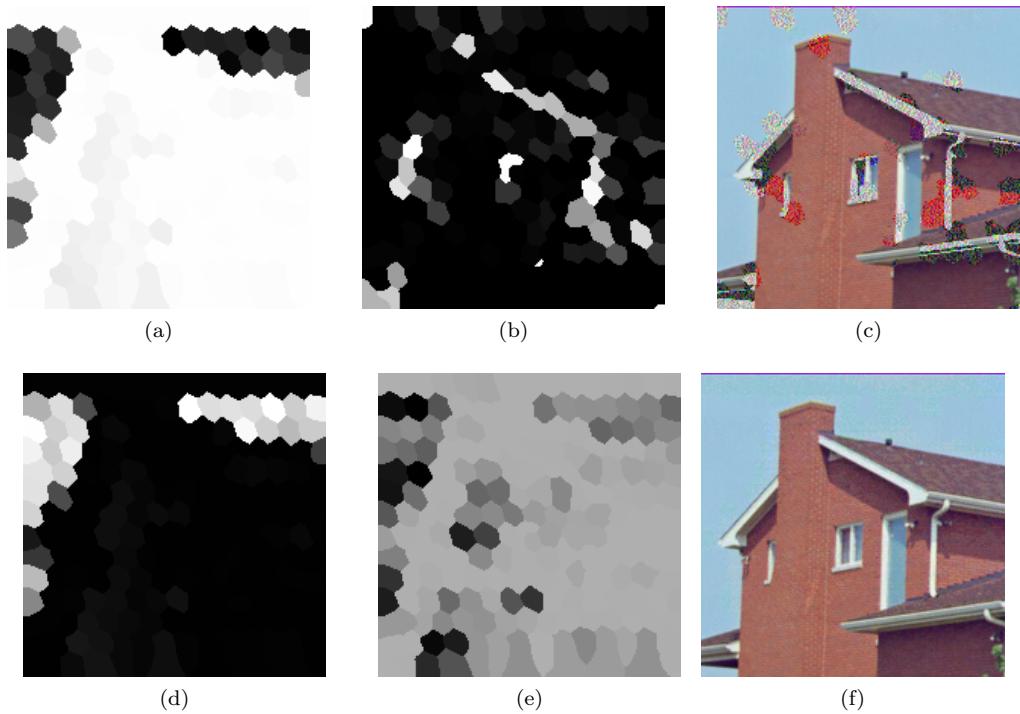
Rysunek 5.6 przedstawia wizualizację utworzonego grafu oraz uzyskane ślady feromonowe. Porównywane obrazy zostały podzielone metodą superpixeli na 190 segmentów ($V = 20$). Na podstawie uzyskanych steganogramów możemy zauważać kołym okiem mankament wariantu w którym długość grafu jest odwrotnie proporcjonalna do różnicy wariancji segmentów (obrazy 5.6a, 5.6b, 5.6c).

W przypadku rozwiązywania problemu *TSP* ostatecznym wynikiem jest wybór jedynie V spośród wszystkich krawędzi których jest $\frac{V \times (V-1)}{2}$. Oznacza to, że gdy system mrówkowy osiągnie zbieżność ślad feromonowy na wszystkich krawędziach poza tymi należącymi do najkrótszej ścieżki będzie bliski zeru. W takim przypadku, do ukrycia zadanej informacji zostanie wykorzystana jedynie znikoma część obrazu nośnego, co negatywnie wpływa na pojemność steganogramu. Wykres 5.7 przedstawia stosunek liczby V do ilości krawędzi grafu w zależności od liczby jego wierzchołków.

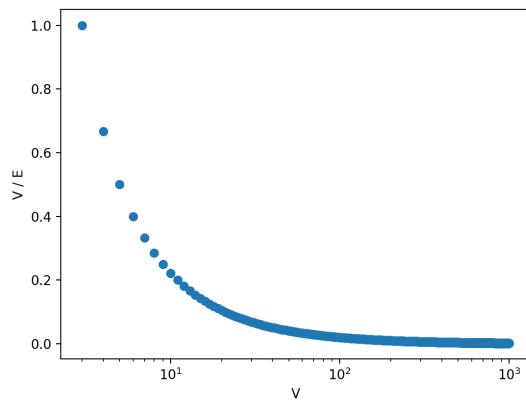
Powyższy problem można rozwiązać na conajmniej dwa sposoby. Pierwszym jest odpowiedni dobór parametrów systemu mrówkowego, który będzie wymuszał na mrówkach częstszą eksplorację (na przykład wybór dużej wartości α). Dzięki temu, większa ilość krawędzi będzie odwiedzana, lecz wnosi to również ryzyko otrzymywania nieoptimalnych cykli. Alternatywą jest odwrócenie problemu reprezentowanego przez graf. Jeśli długości krawędzi będą proporcjonalne do różnic wariancji łączonych segmentów, mrówki będą dążyć do znalezienia V krawędzi w których nie należy umieszczać informacji, gdyż nie są wystarczająco złożone. W ten sposób, uzyskany ślad feromonowy wskaże $V - \frac{V \times (V-1)}{2}$ segmentów w których powinno zostać ukryte najwięcej informacji.

Ponieważ rozwiązywany problem był analogiczny do problemu komiwojażera, w trakcie eksperymentów wykorzystano optymalne parametry dla każdego z rodzajów systemu wyznaczone podczas testowania poprawności działania systemu mrówkowego. Ich wartości oraz metodę ich wyznaczenia można znaleźć w rozdziale 4.

Ważnym aspektem metody wykorzystującą segmentację obrazu jest dobór docelowej ilości grup pikseli. Wartość zbyt niska może powodować powstanie seg-



Rysunek 5.6: Porównania wizualizacji konwersji, macierzy maskujących oraz uzyskanych steganogramów. W pierwszym wierszu ((a), (b), (c)) przedstawiono obrazy związane z procesem o długości krawędzi odwrotnie proporcjonalnej do różnicy wariancji pikseli należących do segmentów. W drugim wierszu ((d), (e), (f)) długości są proporcjonalne do różnicy wariancji. Pierwsza kolumna przedstawia wizualizacje procesu konwersji bitmapy na graf, druga przestawia uzyskane macierze maskujące, trzecia zawiera uzyskane steganogramy zawierające $50kB$ informacji.



Rysunek 5.7: Wykres przedstawia stosunek liczby krawędzi należących do cyklu Hamiltona grafu pełnego o V wierzchołkach i E krawędziach w zależności od liczby wierzchołków.

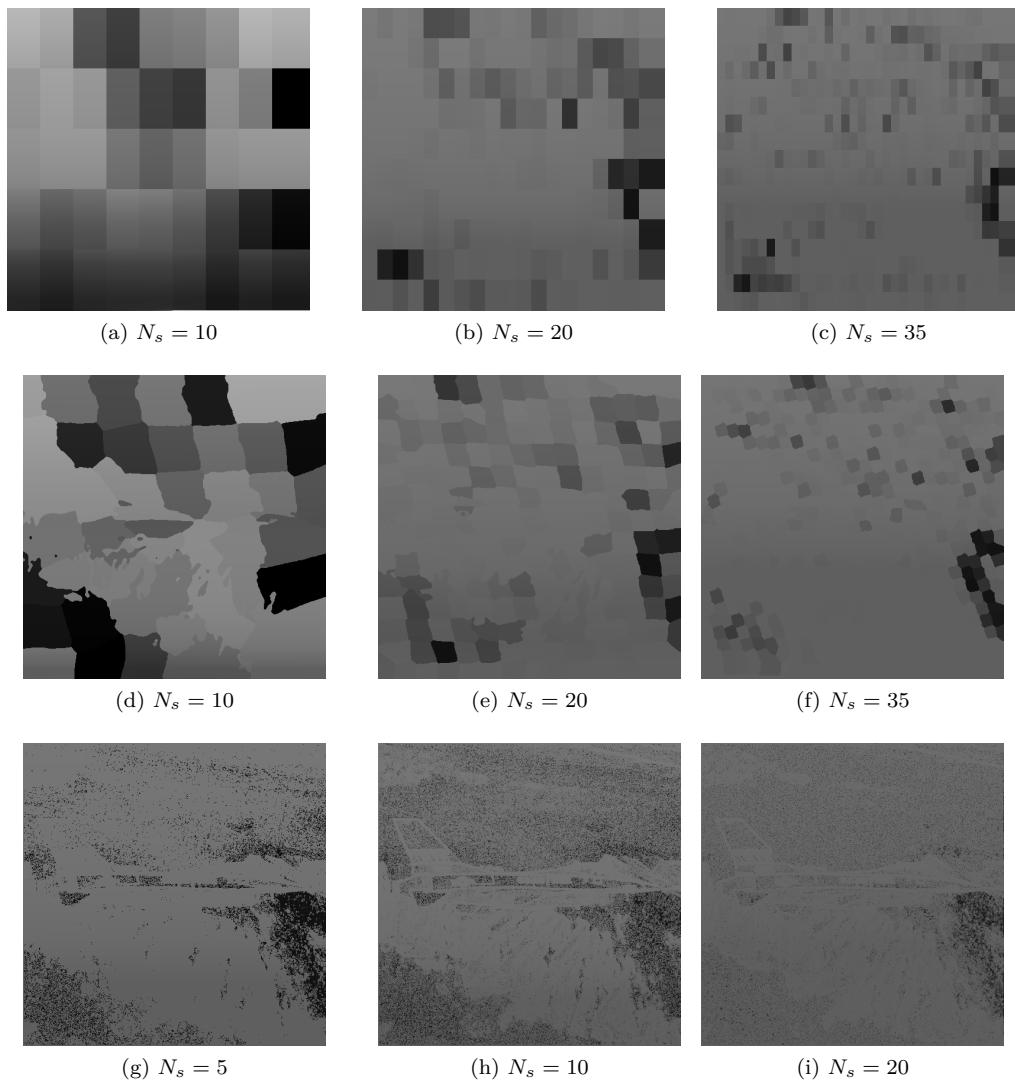
mentów łączących obszary jednorodne i złożone, co przełoży się na zakodowanie nieoptimalnej ilości bitów w wszystkich pikselach segmentu. Zbyt wysoka ilość, oprócz większej złożoności obliczeniowej, może przełożyć się na stworzenie segmentów zbyt małych, których wariancja nie będzie wiernie oddawać złożoności całego obszaru.

Rysunek 5.8 przedstawia segmentację każdego z obrazów nośnych na zadane ilości segmentów z naniesionym śladem feromonowym. W tabeli 5.4 zawarto wartości miar uzyskanych steganogramów dla zadanej pojemności $25kB$ oraz obrazu nośnego o wymiarach 512×512 .

Każda z metod podziału obrazu osiągnięto zbliżone rezultaty, lecz warto zwrócić uwagę na fakt, że metody podziału na prostokąty oraz superpiksele wymagają przekroczenia pewnej wartości granicznej powyżej której osiągane są dobre wyniki.

W związku z porównywalnymi wynikami poszczególnych metod podziału obrazu, dalej zdecydowano się korzystać z podziału na 20 superpikseli.

Ostatnim krokiem badań metody opartej o wierzchołki, było porównanie uzyskanych wyników na pozostałych obrazach testowych. Uzyskane rezultaty zawiera tabela 5.4 a steganogramy rysunek 5.9.



Rysunek 5.8:

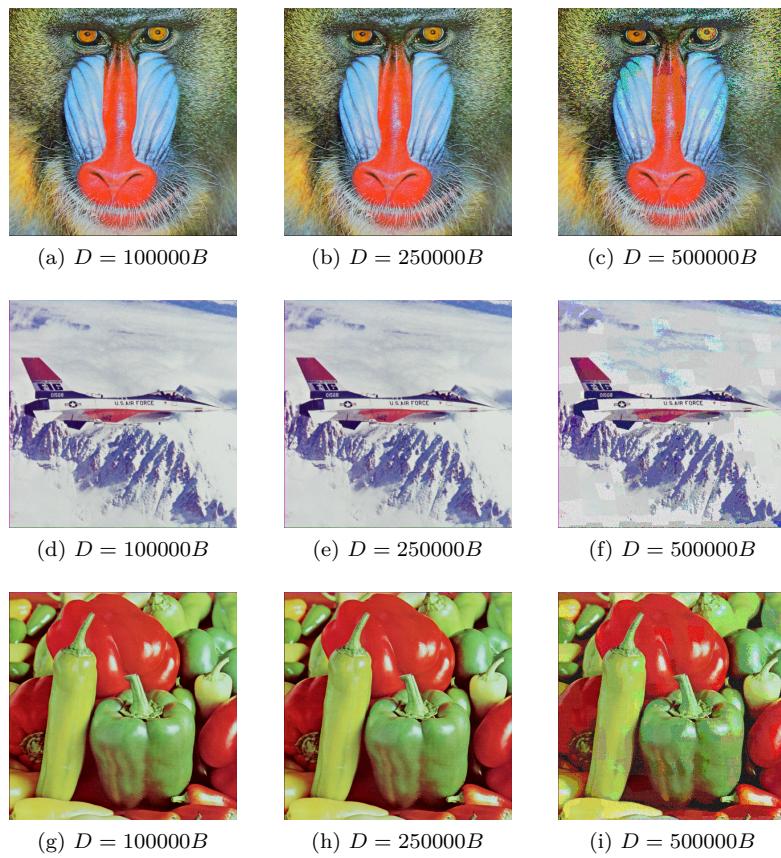
Macierze maskujące uzyskane przy podziale obrazu na nienachodzące prostokąty ((a), (b), (c)), super piksele ((d), (e), (f)) i zbiory wyznaczone metodą k-średnich ((g), (h), (i)).

Metoda segmentacji	liczba segmentów	<i>MSE</i>	<i>PSNR</i>	<i>SSIM</i>
Podział na prostokąty	10 ($V = 5$)	1764.78	15.6978dB	0.186649
	21 ($V = 7$)	1764.78	15.6978dB	0.186649
	45 ($V = 10$)	27.2669	33.8084dB	0.891967
	190 ($V = 20$)	7.25700	39.5572dB	0.948116
	595 ($V = 35$)	6.85841	39.8025dB	0.947396
	1225 ($V = 50$)	6.85694	39.8034dB	0.947764
Super piksele	10 ($V = 5$)	1764.78	15.6978dB	0.186649
	21 ($V = 7$)	21.7769	34.7848dB	0.903446
	45 ($V = 10$)	19.2949	35.3103dB	0.913704
	190 ($V = 20$)	7.07101	39.6699dB	0.948129
	595 ($V = 35$)	6.90553	39.7728dB	0.946333
	1225 ($V = 50$)	6.77511	39.8556dB	0.945946
Grupowanie k-średnich	10 ($V = 5$)	7.89710	39.1901dB	0.940743
	21 ($V = 7$)	7.63390	39.3373dB	0.94536
	45 ($V = 10$)	7.30898	39.5262dB	0.948244
	190 ($V = 20$)	6.90061	39.7759dB	0.948481
	595 ($V = 35$)	6.76695	39.8608dB	0.949062
	1225 ($V = 50$)	-	-	-

Tablica 5.3: Miary jakości w zależności od metody segmentacji obrazu.

Obraz	objętość danych	<i>MSE</i>	<i>PSNR</i>	<i>SSIM</i>
Baboon 512×512	100 000B (12.71%)	0.893791	48.6524dB	0.99839
Baboon 512×512	250 000B (31.78%)	7.52079	39.4021dB	0.987705
Baboon 512×512	500 000B (63.57%)	663.829	19.9442dB	0.723623
Airplane 512×512	100 000B (12.71%)	0.589252	50.4617dB	0.995266
Airplane 512×512	250 000B (31.78%)	7.07101	39.6699dB	0.948129
Airplane 512×512	500 000B (63.57%)	362.857	22.5674dB	0.510537
Peppers 512×512	100 000B (12.71%)	0.586621	50.4812dB	0.995526
Peppers 512×512	250 000B (31.78%)	7.21685	39.5813dB	0.951955
Peppers 512×512	500 000B (63.57%)	411.621	22.0198dB	0.47874

Tablica 5.4: Wyniki uzyskane metodą opartą o wierzchołki.



Rysunek 5.9: Porównanie uzyskanych steganogramów przy zadanej objętości ukrytej informacji.

5.4 Ocena subiektywna

Uzyskane steganogramy spełniają postawione przed nimi oczekiwania. Dla obrazów w których wykorzystano $\sim 32\%$ dostępnej objętości zmiany są praktycznie niezauważalne gołym okiem. Artefakty stały się dopiero wyraźnie widoczne gdy spróbowano ukryć $500kB$, czyli blisko 64% objętości. W przypadku metody opartej o podział obrazy na grupy pikseli były one bardziej widoczne, gdyż występowały krawędzie pomiędzy superpikselami lub prostokątami dzielącymi obraz. Metoda budowania grafu na podstawie wierzchołków gwarantowała płynniejsze przejścia pomiędzy obszarami w których wykorzystano większą ilość bitów pikseli.

Warto również podkreślić zgodność subiektywnych odczuć z wartościami przedstawionych miar jakości. Dodatkowo, przytoczony zakres szczytowego stosunku sygnału do szumu, $30dB - 50dB$, pokrywa się z doznaniami empirycznymi. Dla akceptowalnych steganogramów wykorzystujących $\sim 32\%$ pojemności, wartość $PSNR$ wynosiła blisko $40dB$. W przypadku wykorzystania $\sim 64\%$ bitów obrazu, wartość $PSNR$ spadła do nieakceptownego poziomu bliskiemu $20dB$.

Kolejną interesującą obserwacją płynie z porównania efektów ukrywania informacji w obrazie *Baboon* i *Airplane*. Nawet w przypadku ukrycia blisko $500kB$ danych, były one ewidentnie bardziej widoczne w obrazie *Airplane*. Może to być tłumaczone dużo większym zróżnicowaniem obrazu *Baboon*, oraz większą ilością szczegółów i tekstur.

5.5 Porównanie wyników z innymi metodami

Ostatnim krokiem związanym z badaniem efektywności zaproponowanych metod było porównanie uzyskanych wyników do powiązanych prac z dziedziny. Udostępnione wyniki badań pozwoliły na utworzenia porównującego zestawienia z metodą *Least Significant Bit*[54], *Pixel Value Differencing*[54], *Particle Swarm Optimization-Integer Wavelet Transform*[35] oraz metodą opartą o wykrywanie złożonego regionu obrazu za pomocą systemu mrowiskowego[25], do której odniesienie można znaleźć w rozdziale 3.

W związku z tym, że w powyższych artykułach przedstawione wyniki zostały podane dla różnych objętości ukrywanych informacji, zdecydowano się umieścić

Obraz	pojemność			Least Significant Bit			Metoda oparta o wierzchołki			Metoda oparta o krawędzi		
	bpp	%	B	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Airplane	2.67	11.12%	87491B	-	51.65dB	0.9970	0.447	51.66dB	0.9963	0.448	51.65dB	0.9963
Baboon	2.67	11.12%	87491B	-	51.65dB	0.9997	0.449	51.64dB	0.9990	0.816	49.05dB	0.9985
Peppers	2.67	11.12%	87491B	-	51.62dB	0.9998	0.448	51.65dB	0.9968	0.447	51.66dB	0.9967

Tablica 5.5: Porównanie miar jakości z uzyskanymi metodą *LSB* w pracy „*LSB Substitution and PVD performance analysis for image steganography*”

Obraz	pojemność			Pixel Value Differencing			Metoda oparta o wierzchołki			Metoda oparta o krawędzi		
	bpp	%	B	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Airplane	4.6699	19.45%	153023B	-	40.61dB	0.9751	1.596	46.13dB	0.9879	1.645	46.00dB	0.9865
Baboon	5.3627	22.34%	175725B	-	37.83dB	0.9945	2.012	45.13dB	0.9960	2.647	43.94dB	0.9953
Peppers	4.7177	19.65%	154590B	-	40.93dB	0.9980	1.638	46.21dB	0.9892	1.684	45.90dB	0.9878

Tablica 5.6: Porównanie miar jakości z uzyskanymi metodą *PVD* w pracy „*LSB Substitution and PVD performance analysis for image steganography*”

poszczególne porównania w osobnych tabelach. Format zapisu wyników był niejednolity, część prac wyrażała uzyskaną pojemność w postaci udziału procentowego bitów obrazu nośnego, ilości bitów w przeliczeniu na piksel (*bpp*) lub bezwzględnej ilości bajtów. Ponieważ obrazy na których wykonywano eksperymenty są znane, możliwe było ustalanie przytoczonych wartości.

Zestawienia z poszczególnymi pracami zawierają tabele 5.5, 5.6, 5.7 oraz 5.8.

Obraz	pojemność	<i>PSO-IWT</i>			Metoda oparta o wierzchołki			Metoda oparta o krawędzi		
		B	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR
Airplane	73728B	-	41.13dB	-	0.378	52.39dB	0.9971	0.378	52.39dB	0.9968
Baboon	73728B	-	41.38dB	-	0.379	52.38dB	0.9993	0.378	52.38dB	0.9993

Tablica 5.7: Porównanie miar jakości z uzyskanymi metodą *PSO-IWT* w pracy „*A Novel Image Steganographic Method based on Integer Wavelet Transformation and Particle Swarm Optimization*”

Obraz	pojemność		PSO-IWT			Metoda oparta o wierzchołki			Metoda oparta o krawędzie		
	%	B	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Baboon	2.8473%	22396B	0.8409	48.88dB	-	0.1151	57.55dB	0.9999	0.1146	57.57dB	0.9998
Pepper	2.8661%	22544B	1.0605	47.88dB	-	0.1151	57.56dB	0.9994	0.1150	57.56dB	0.9991

Tablica 5.8: Porównanie miar jakości z uzyskanymi metodą ACO w pracy „Ant Colony Optimization (ACO) based Data Hiding in Image Complex Region”

Bibliografia

- [1] R. Achanta, A. Shaji, K. Smith, Aurélien Lucchi, P. Fua, S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:2274–2282, 2012.
- [2] M. A. F. Al-Husainy, Diaa M. Uliyan. A secret-key image steganography technique using random chain codes. *International Journal of Technology*, 10:731, 2019.
- [3] American Psychological Association. Perception of high and low spatial frequency information in pigeons and people, 2015.
- [4] P. Balaji, D. Srinivasan. An introduction to multi-agent systems. 2010.
- [5] L. Bianchi, M. Dorigo, L. Gambardella, W. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8:239–287, 2008.
- [6] E. Bonabeau, F. Hénaux, S. Guerin, D. Snyers, P. Kuntz, G. Theraulaz. Routing in telecommunications networks with smart ant-like agents. 1998.
- [7] B. Bullnheimer, R. Hartl, C. Strauss. A new rank based version of the ant system: A computational study. 1997.
- [8] B. Bullnheimer, R. Hartl, C. Strauss. Applying the ant system to the vehicle routing problem. 1999.
- [9] K. Cabeen, P. Gent. image compression and the discrete cosine transform. College of Redwoods.

- [10] G. D. Caro. Antnet: A mobile agents approach to adaptive routing. 1999.
- [11] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. 1976.
- [12] Alberto Colomni, Marco Dorigo, Vittorio Maniezzo, Marco Trubian. Ant system for job-shop scheduling. *STATISTICS AND COMPUTER SCIENCE*, 34, 01 1994.
- [13] M. Dorigo, L. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.*, 1:53–66, 1997.
- [14] M. Dorigo, V. Maniezzo, A. Colomni. Ant system: optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 26 1:29–41, 1996.
- [15] M. Dorigo, T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications, and advances. 2003.
- [16] Marco Dorigo, Vittorio Maniezzo, Alberto Colomni. Ant system: An autocatalytic optimizing process technical report 91-016. 02 1999.
- [17] A. Dorri, Salil S. Kanhere, Raja Jurdak. Multi-agent systems: A survey. *IEEE Access*, 6:28573–28593, 2018.
- [18] Nameer N. El-Emam, Rasheed Abdul Shaheed AL-Zubidy. New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm. *J. Syst. Softw.*, 86:1465–1481, 2013.
- [19] A. Fraser. Simulation of genetic systems by automatic digital computers i. introduction. *Australian Journal of Biological Sciences*, 10:484–491, 1957.
- [20] L. Gambardella, É. Taillard, M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50:167–176, 1999.

- [21] P. Guillot. Auguste kerckhoffs et la cryptographie militaire. 2013.
- [22] Fangjun Huang, Xiaochao Qu, H. J. Kim, J. Huang. Reversible data hiding in jpeg images. *IEEE Transactions on Circuits and Systems for Video Technology*, 26:1610–1621, 2016.
- [23] J. Huang, Y. Shi. Embedding image watermarks in dc components. *IEEE Trans. Circuits Syst. Video Technol.*, 10:974–979, 2000.
- [24] S. Islam, Mangat Rai Modi, P. Gupta. Edge-based image steganography. *EURASIP Journal on Information Security*, 2014:1–14, 2014.
- [25] Sahib Khan. Ant colony optimization (aco) based data hiding in image complex region. *International Journal of Electrical and Computer Engineering*, 8:379–389, 2018.
- [26] Sahib Khan, N. Ahmad, M. Wahid. Varying index varying bits substitution algorithm for the implementation of vlsb steganography. *Journal of the Chinese Institute of Engineers*, 39:101 – 109, 2016.
- [27] Sahib Khan, M. Yousaf, Jamal Akram. Implementation of variable least significant bits stegnography using dddb algorithm. 2011.
- [28] S. Kirkpatrick, C. D. Gelatt, M. Vecchi. Optimization by simulated annealing. *Science*, 220:671 – 680, 1983.
- [29] X. Li, J. Wang. A steganographic method based upon jpeg and particle swarm optimization algorithm. *Inf. Sci.*, 177:3099–3109, 2007.
- [30] James B. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.
- [31] V. Maniezzo, A. Colorni. The ant system applied to the quadratic assignment problem. *IEEE Trans. Knowl. Data Eng.*, 11:769–778, 1999.
- [32] A. Mazidi, Elham Damghanijazi. Meta-heuristic approaches for solving travelling salesman problem. *International Journal of Advanced Research in Computer Science*, 8:18–23, 2017.

- [33] D. Merkle, M. Middendorf, H. Schmeck. Ant colony optimization for resource-constrained project scheduling. *IEEE Trans. Evol. Comput.*, 6:333–346, 2002.
- [34] Scott Mitchell. H.264 encoded digital video protection using temporal redundancy lsb steganography. 2018.
- [35] P. Muhuri, Z. Ashraf, Swati Goel. A novel image steganographic method based on integer wavelet transformation and particle swarm optimization. *Appl. Soft Comput.*, 92:106257, 2020.
- [36] Sunil Muttoo, Sushil Kumar. Data hiding in jpeg images. *International Journal of Information Technology (IJIT)*, 1, 07 2009.
- [37] Hebah H. O. Nasereddin. Digital watermarking a technology overview. 2011.
- [38] Nature. A note on the lena image. *Nature Nanotechnology*, 13, 2018.
- [39] Journal of Modern Optics. On alternatives to lenna. *Journal of Modern Optics*, 64(12):1119–1120, 2017.
- [40] University of Southern California. Usc database. <http://sipi.usc.edu/database/database.php?volume=misc#top>.
- [41] M. Oprea. Applications of multi-agent systems. *IFIP Congress Tutorials*, 2004.
- [42] Sabrina Perfetto, John D. Wilder, Dirk B. Walther. Effects of spatial frequency filtering choices on the perception of filtered images. *Vision*, 4, 2020.
- [43] F. Petitcolas, R. Anderson, M. Kuhn. Information hiding-a survey. 1999.
- [44] R. Poli, J. Kennedy, T. Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1:33–57, 2007.
- [45] M. Pope, M. Warkentin, E. Bekkering, M. B. Schmidt. Digital steganography - an introduction to techniques and tools. *Commun. Assoc. Inf. Syst.*, 30:22, 2012.
- [46] Roshan Poudél. *Covert Channel and Data Hiding in TCP/IP*. 11 2019.

- [47] S. Prabakaran, T. S. Kumar, J. Ramana, K. Reddy. A survey on approaches to solve travelling salesman problem. *Eurasian Journal of Analytical Chemistry*, 13:292–299, 2019.
- [48] A. Priya. High capacity and optimized image steganography technique based on ant colony optimization algorithm. 2018.
- [49] J. Reichel, G. Menegaz, M. Nadenau, M. Kunt. Integer wavelet transform for embedded lossy to lossless image compression. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 10 3:383–92, 2001.
- [50] T. Richter, S. Escher, D. Schönfeld, Thorsten Strufe. Forensic analysis and anonymisation of printed documents. *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, 2018.
- [51] A. Saleema, T. Amarunnishad. A new steganography algorithm using hybrid fuzzy neural networks. *Procedia Technology*, 24:1566–1574, 2016.
- [52] Umme Sara, Morium Akter, Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computational Chemistry*, 7:8–18, 2019.
- [53] N. Sharma, Usha Batra. An inclusive study and analysis of steganographic methodologies for data security. 2020.
- [54] Serdar Solak, Umut ALTINIand Ik. Lsb substitution and pvd performance analysis for image steganography. 2018.
- [55] T. Stützle, H. Hoos. Max-min ant system. *Future Gener. Comput. Syst.*, 16:889–914, 2000.
- [56] Matthew Szudzik. An elegant pairing function. 2006.
- [57] Hai Tao, Li Chongmin, J. Zain, A. Abdalla. Robust image watermarking theories and techniques: A review. *Journal of Applied Research and Technology*, 12:122–138, 2014.

- [58] J. Uhlmann. A canonical image set for examining and comparing image processing algorithms. *ArXiv*, abs/1805.00116, 2018.
- [59] Zhou Wang, A. Bovik, H. Sheikh, Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.
- [60] Chris Watkins. Learning from delayed rewards. 1989.
- [61] David Wheeler, Daryl Johnson, Bo Yuan, P. Lutz. Audio steganography using high frequency noise introduction. 2012.
- [62] Da-Chun Wu, W. Tsai. A steganographic method for images by pixel-value differencing. *Pattern Recognit. Lett.*, 24:1613–1626, 2003.
- [63] G. Xuan, Y. Shi, Chengyun Yang, Yizhan Zhen, D. Zou, P. Chai. Lossless data hiding using integer wavelet transform and threshold embedding technique. *2005 IEEE International Conference on Multimedia and Expo*, strony 1520–1523, 2005.
- [64] Eman Talib Zghaer, Assist. Prof. Dr Soukaena H.Hashem. Ant colony optimization to enhance image steganography. *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, 6(2278-6856), 2017.
- [65] Dong Zhao, L. Liu, F. Yu, A. Heidari, Mingjing Wang, D. Oliva, Khan Muhammad, Huiling Chen. Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Syst. Appl.*, 167:114122, 2021.
- [66] Zhiqiang Zhu, N. Zheng, Tong Qiao, Ming Xu. Robust steganography by modifying sign of dct coefficients. *IEEE Access*, 7:168613–168628, 2019.

Spis rysunków

1.1	Wynik działania <i>IWT</i> na przykładowym obrazie. Kolejno sekcje LL, HL, LH, HH. Do wykonania rysunku wykorzystano narzędzie Image Processing Online Demonstration, http://bigwww.epfl.ch/demo/ip/demos/wavelets/	8
3.1	Rodzaje sąsiedztwa	30
3.2	Wizualizacja grafu tworzonego metodą opartą o wierzchołki	32
3.3	Wizualizacja grafu tworzonego metodą opartą o podział obrazu na nie nachodzące prostokąty	34
3.4	Wizualizacja grafu tworzonego metodą opartą o podział obrazu przy użyciu algorytmu k-średnich	35
3.5	Wizualizacja grafu tworzonego metodą opartą o podział obrazu na super piksele	36
5.1	Porównanie obrazów	50
5.2	Porównania wizualizacji konwersji oraz macierzy maskujących.	55
5.3	Porównania wizualizacji konwersji oraz macierzy maskujących.	57
5.4	Porównania wizualizacji konwersji oraz macierzy maskujących.	58
5.5	Porównanie rezultatów	60
5.6	Porównania wizualizacji konwersji oraz macierzy maskujących.	62
5.7	Wykres przedstawia stosunek liczby krawędzi należących do cyklu Hamiltona grafu pełnego o V wierzchołkach i E krawędziach w za- leżności od liczby wierzchołków.	63
5.8	Macierze maskujące.	64
5.9	Porównanie rezultatów	66

Spis tablic

4.1	Uzyskane rozwiązania problemów <i>oliver30</i> oraz <i>kroa100</i>	48
5.1	Miary jakości steganogramów w zależności od współczynnika s_0 . . .	59
5.2	Wyniki uzyskane metodą opartą o wierzchołki.	59
5.3	Miary jakości w zależności od metody segmentacji obrazu.	65
5.4	Wyniki uzyskane metodą opartą o wierzchołki.	65
5.5	Porównanie miar jakości z uzyskanymi metodą <i>LSB</i> w pracy „ <i>LSB Substitution and PVD performance analysis for image steganography</i> ”	68
5.6	Porównanie miar jakości z uzyskanymi metodą <i>PVD</i> w pracy „ <i>LSB Substitution and PVD performance analysis for image steganography</i> ”	68
5.7	Porównanie miar jakości z uzyskanymi metodą <i>PSO-IWT</i> w pracy „ <i>A Novel Image Steganographic Method based on Integer Wavelet Transformation and Particle Swarm Optimization</i> ”	68
5.8	Porównanie miar jakości z uzyskanymi metodą <i>ACO</i> w pracy „ <i>Ant Colony Optimization (ACO) based Data Hiding in Image Complex Region</i> ”	69