MathUp

Konferencja Zastosowań Matematyki

Edycja VII

Matematyka – od zastosowań oczywistych do zdumiewających - MathUp 2025

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Centrum Nauczania Matematyki i Fizyki
Politechnika Łódzka

Politechnika Łódzka
WEEIA
CENTRUM NAUCZANIA MATEMATYKI I FIZYKI POLITECHNIKI ŁÓDZKIEJ

MathUp
Konferencja Zastosowań Matematyki

SPIS TREŚCI

MathUp
Konferencja Zastosowań Matematyki

CZY WIEMY JUŻ WSZYSTKO?

Gertruda GWÓŹDŹ-ŁUKAWSKA, Elżbieta GALEWSKA,
Elżbieta KOTLICKA-DWURZNIK, Monika POTYRAŁA
Politechnika Łódzka, Łódź

Wstęp

Z pewnością wielu ludzi – zarówno tych, którzy zakończyli już edukację, jak i tych, którzy są właśnie uczniami szkół podstawowych lub średnich – uważa, że matematyka to nauka w pewnym sensie zamknięta. I, jeśli weźmiemy pod uwagę materiał omawiany w szkole (nie mówimy tu o szkole wyższej), to faktycznie zakres materiału jest bardzo ograniczony i nie jest specjalnie nowy w sensie odkryć matematycznych. Czasami nauczyciele sięgają po ilustracje i ciekawostki nawiązujące do współczesnej matematyki lub jej zastosowań, ale nie jest to zjawisko częste. Zatem, gdyby w nauce decydowała demokracja, to można by uznać, że matematyka jest „dziedziną wymarłą".
Tymczasem wszyscy ci, którzy bądź zajmują się matematyką (naukowcy, studenci, pasjonaci), jak i ci, którzy po prostu są ciekawi wszelkich nowinek (niekoniecznie ograniczający się do jednej dziedziny wiedzy) powiedzą z całą mocą, że królowa nauk żyje i wciąż zdumiewa coraz to nowymi odkryciami, o czym można przeczytać już w [1].

Niniejsza monografia daje uważnemu Czytelnikowi okazję do zapoznania się z szerokim wachlarzem zastosowań matematyki, począwszy od tych najszerzej znanych, jak własności trójkąta, po te wybiegające w przyszłość, takie jak algorytmy umożliwiające przetwarzanie coraz większych zbiorów danych.

Zastosowania oczywiste

Być może wiele osób myśląc o figurach geometrycznych wyobraża sobie koło, kwadrat lub trójkąt. I właśnie trójkątom poświęcony jest jeden rozdział, gdyż te powszechnie znane i używane już od przedszkola kształty okazują się – wciąż – mieć wiele zastosowań. Powiązanie długości boków trójkąta i miar jego kątów wewnętrznych od setek lat jest przydatne do wyznaczania siatek triangulacyjnych (tj. zespołów trójkątów połączonych w taki sposób, że mają one wspólne, przyległe boki), które były podstawą kartografii. Wykorzystanie twierdzenia sinusów w geometrii sferycznej pozwala zachować precyzję pomiarów na powierzchni kuli ziemskiej. Trójkąty wykorzystywano również w działaniach bojowych (w dalmierzach optycznych stosowanych jeszcze w XX wieku), a także przy określaniu położenia ciał na sferze niebieskiej. Jednym z najbardziej powszechnych zastosowań trójkątów jest triangulacja umożliwiająca lokalizację nawet tam, gdzie system GPS nie dociera.

Barwy i matematyka? Czemu nie. Jeśli odwołamy się do oznaczeń barw zgodnie z modelem RGB, połączenie to wydaje się zupełnie naturalne i oczywiste.
Jednak, by opisać świat barw matematycznie, potrzebujemy nie tylko liczb lecz przede wszystkim rachunku wektorowego. Wszelkie kolory modelu RGB to kombinacje liniowe trzech bazowych kolorów: czerwonego (255,0,0), zielonego (0,255,0) i niebieskiego (0,0,255). Mnożenie koloru (wektora) przez liczbę zmienia jego nasycenie.

W modelu RGB kolory powstają poprzez sumowanie intensywności świateł. Ale można inaczej – analizując światło odbite od przedmiotów (kolory uzyskuje się poprzez pochłanianie – odejmowanie określonych długości fal światła przez barwniki). Cyan (C) pochłania światło czerwone, magenta (M) – zielone, Yellow (Y) – niebieskie. To znany wszystkim model CMYK, K – key to kolor czarny.

Zastosowania nieoczywiste

Jak perspektywa łączy się z grą Dobble? Kluczowym pojęciem jest płaszczyzna projekcyjna (rzutowa), której najważniejszą cechą jest to, że nie ma na niej żadnych prostych równoległych. Dwa dowolne punkty na tej płaszczyźnie łączy dokładnie jedna prosta. Zamieniając słowa punkty na karty, a prosta na symbol, otrzymujemy podstawowy warunek spełniony w Dobble: dwie dowolne karty łączy dokładnie jeden symbol. Talia składa się z 55 kart, a na każdej karcie jest 8 różnych symboli. Płaszczyzna projekcyjna opisująca grę Dobble ma rząd równy 7. Posługując się algorytmem tworzenia własnych kart Dobble zaznaczamy najpierw na płaszczyźnie 7×7 wszystkie proste, które wyglądają jak proste równoległe, dokładamy punkt w nieskończoności (punkt niewłaściwy) i łączymy wszystkie proste z tym właśnie punktem. Następnie wykonujemy taką samą procedurę dla prostych o dowolnym nachyleniu i na koniec łączymy wszystkie punkty niewłaściwe jedną prostą. Działanie tego algorytmu jest podobne do perspektywicznego przedstawienia dwóch równoległych pasów drogi przecinających się w punkcie (niewłaściwym) na horyzoncie. Pozostaje pytanie, czy można za pomocą takiego schematu postępowania stworzyć talię Dubble o dowolnej liczbie kart? Odpowiedź jest negatywna. Algorytm jest słuszny tylko w przypadku, gdy rząd odpowiedniej płaszczyzny projekcyjnej jest liczbą pierwszą, co nie oznacza, że gry opartej na płaszczyźnie innego rzędu nie da się stworzyć.

Być może fakt, że matematyka pomaga ubezpieczycielom nie popaść w ruinę, nie jest zaskakujący. Nie jest jednak oczywiste, jak dobrze wszystko przekalkulować by chronić się przed ryzykiem. Np. powodzie. Zdarzają się lokalnie, nieprzewidywalnie… wystarczy niż genueński. Jak zatem będąc ubezpieczycielem trafnie ustalić stawkę?
Idealnym narzędziem są jednorodne łańcuchy Markowa. Przy wykorzystaniu przełącznikowego modelu Sparre Andersena możliwe staje się zróżnicowanie wysokości pobieranej składki oraz zmiana rozkładów szkód w zależności od stanu rzeczywistości przełącznika.
Najistotniejsze jest jednak wyznaczenie momentu ruiny – okresu, w którym nadwyżka złożona z kapitału początkowego i składek spada poniżej zera (po wypłaceniu szkody) oraz określenie prawdopodobieństwa niewypłacalności ubezpieczyciela w określonym horyzoncie czasowym. Gdy mowa o pierwszym okresie rozliczeniowym – wystarczają definicje i podstawowe własności prawdopodobieństwa. By ustalić prawdopodobieństwo niewypłacalności w kolejnych okresach, a w szczególności jaki kapitał początkowy jest niezbędny, by prawdopodobieństwo ruiny nie przekroczyło 0,5%, pracuje dodatkowe narzędzie – tzw. operator ryzyka.

Poruszając temat ryzyka, nie sposób nie wspomnieć o faktycznym zagrożeniu jakim są przestępcy. I choć ściganiem przestępstw zajmuje się kryminalistyka, to okazuje się, że bez matematyki jej skuteczność byłaby zdecydowanie mniejsza. Wszystko zaczyna się od pojęcia metryki (czyli odległości), która w realnym życiu nie jest liczona jako długość najkrótszego odcinka łączącego dwa punkty. Kolejnym pojęciem jest prawdopodobieństwo – za pomocą wzoru Rossmo określa się prawdopodobieństwo zamieszkania sprawcy w konkretnym miejscu. Umieszczenie wyliczonych prawdopodobieństw (dla różnych potencjalnych miejsc pobytu złoczyńcy) w postaci graficznej pozwala na utworzenie mapy cieplnej, która służy profilerom do profilowania geograficznego. Znacznie zmniejsza to obszar poszukiwań i ułatwia pracę organom ścigania.
Jednak nie są to jedyne zagadnienia matematyczne stosowane w kryminalistyce; wykorzystuje ona bowiem rachunek macierzowy, a nawet teorię równań różniczkowych i szeregów. Dokładając

jeszcze regresję liniową, możemy całkiem skutecznie prognozować datę kolejnego ataku seryjnego przestępcy.
Na szczęście, przed niektórymi atakami możemy się w pewnym stopniu zabezpieczyć, w czym pomóc nam może kryptografia.

Liczby pierwsze od wieków fascynują matematyków – to liczby, które dzielą się tylko przez 1 i samą siebie, a mimo swojej prostoty kryją w sobie ogromną moc. Już w starożytności Eratostenes opracował słynne sito, pozwalające szybko wyszukiwać liczby pierwsze. Z czasem pojawiły się również metody sprawdzania, czy dana liczba jest pierwsza, jak choćby test Fermata. Dziś liczby te nie są już tylko ciekawostką, ale stanowią fundament współczesnej kryptografii, czyli nauki o szyfrowaniu danych. To właśnie one odpowiadają np. za bezpieczeństwo algorytmu RSA, używanego codziennie przy logowaniu do banku czy wysyłaniu wiadomości przez Internet. Szyfrowanie RSA opiera się na tym, że łatwo jest pomnożyć dwie duże liczby pierwsze, ale odwrotny proces, czyli faktoryzacja ich iloczynu, jest niezwykle trudny dla klasycznych komputerów. Nawet najlepsze metody faktoryzacji, takie jak GNFS, potrzebują tysięcy lat, aby rozłożyć naprawdę duże liczby. Dlatego klucze RSA, które liczą ponad 600 cyfr, wciąż pozostają bezpieczne. Jednak rozwój komputerów kwantowych, w połączeniu z algorytmem faktoryzacji Shora, w przyszłości może to zmienić.

Algorytmy asocjacyjne to kolejne zagadnienie z pogranicza matematyki i informatyki, z którym mamy styczność na co dzień. To właśnie dzięki nim sklepy internetowe potrafią podpowiedzieć nam, że „klienci, którzy kupili chleb, często kupują też mleko”. Takie zależności, odkrywane w ogromnych bazach danych, nazywamy regułami asocjacyjnymi. Ich zadaniem jest znalezienie wzorców, które powtarzają się wystarczająco często, aby można było z nich wyciągnąć wnioski i na ich podstawie podejmować skuteczne decyzje marketingowe. Do odkrywania reguł asocjacyjnych służą specjalne algorytmy – od klasycznego Apriori, po bardziej zaawansowane, jak FP-Growth czy EClaT. Dzięki nim możliwa jest analiza nawet milionów transakcji w rozsądnym czasie. W algorytmach transakcje reprezentowane są przez wektory binarne (wektory składające się tylko z 0 i 1), zaś zbiory transakcji przez tabele, macierze i ,,drzewa”. Reguły asocjacyjne, choć najczęściej kojarzone są z marketingiem i rekomendacjami produktów, mają też szersze zastosowania – od medycyny, przez bioinformatykę, aż po wspomnianą już kryptografię.

Każdy, kto próbował nagrać rozmowę w zatłoczonej kawiarni, wie, jak trudno później zrozumieć wypowiedziane słowa. Szumy tła, stukot filiżanek, rozmowy innych gości — wszystko to miesza się z naszym głosem, tworząc akustyczny chaos. Czy da się z tego bałaganu wyłowić czysty sygnał? Odpowiedź brzmi: tak — dzięki matematyce, a konkretnie dzięki transformatom Fouriera.
Transformata Fouriera to narzędzie, które pozwala spojrzeć na dźwięk nie tylko jako na ciąg wartości w czasie, ale jako zbiór składowych częstotliwościowych. Dzięki temu możemy łatwo zidentyfikować, które częstotliwości są pożądane (np. głos rozmówcy), a które to zakłócenia. We współczesnej analizie sygnałów stosuje się bardziej efektywne algorytmy FFT i STFT oparte na zmodyfikowanych transformatach Fouriera. Odgrywają one kluczową rolę w cyfrowym przetwarzaniu dźwięku, zwłaszcza w redukcji szumów.

Zastosowania zdumiewające

Co ma wspólnego DNA z matematyką? DNA to biologia. Gdzie tu miejsce na matematyczne rachunki? Odpowiedzią jest kodowanie informacji. Wszystkie cyfrowe informacje zapisywane są w kodzie binarnym. Jak zatem wykorzystać znajomość biologicznego zapisu informacji w DNA do przechowywania?

W DNA kodowanie następuje przy wykorzystaniu czterech zasad azotowych: adeniny, guaniny, cytozyny i tyminy. Naturalne zatem wydaje się kodowanie czwórkami (0,1,2,3), bądź kodowanie binarne parami (00,01,10,11). Okazuje się jednak, że zdecydowanie bardziej efektywne jest kodowanie trójkowe (kodowane są trzy zasady w zależności od poprzedniej - bazowej). Analizując entropię Shannona, można oszacować, że kodowanie trójkowe pozwala zapisać w jednym nukleotydzie aż 1,59 bita. To bardzo dobry wynik, gdyż dla DNA wielkość ta wynosi 2 bity. Niedościgniona w DNA pozostaje jednak gęstość zapisu – niewiarygodnie niedościgniona. Kto podejmie wyzwanie?

Podejmowanie decyzji jest nieodzownym elementem naszego codziennego życia. Wybór tej najlepszej często bywa niełatwym zadaniem. Przekonał się też o tym pewien góral, który, aby dostać się do składziku oscypków, musiał odśnieżyć podwórze po obfitej, nocnej zawierusze. Chciał, by ilość śniegu do odgarnięcia była możliwie najmniejsza. Jaką zatem ścieżkę powinien wybrać? Pomocnym narzędziem matematycznym w rozwiązaniu tego i podobnych problemów optymalizacyjnych jest rachunek wariacyjny. Polega on na analizie wszystkich potencjalnych rozwiązań rozważanego zadania i zastosowaniu kryteriów prowadzących do wyboru rozwiązania optymalnego. Wykorzystuje zaawansowany aparat matematyki wyższej z zakresu analizy matematycznej. Kluczowym wnioskiem jest równanie Eulera-Lagrange'a, które spełnia minimalizowany funkcjonał odpowiadający kryterium najlepszego wyboru.
Przykładem problemu rozwiązywalnego metodami rachunku wariacyjnego jest znalezienie minimalnej powierzchni powstałej przez obrót pewnej krzywej wokół osi układu współrzędnych. Szukana powierzchnia to katenoida wyznaczona przez krzywą łańcuchową. Okazuje się, że dzięki tej samej krzywej łańcuchowej możliwe jest projektowanie optymalnych struktur architektonicznych, takich jak łuki czy kopuły. A stosowane dawniej modelowanie eksperymentalne zastępują w obecnych czasach specjalistyczne programy komputerowe wykorzystujące właśnie rachunek wariacyjny.

Podsumowanie

Powyższe pogrupowanie opisanych w monografii zastosowań jest jedynie sugestią autorów (aktualną w dniu pisania rozdziału, a być może w dniu, w którym Czytelnik zacznie rozważać opisane zagadnienia, rozwój matematyki sprawi, że zastosowania opisane tutaj przestaną już być zdumiewające lub nawet nieoczywiste). Bez względu jednak na zaproponowany podział, każdy z nas zapewne może zachwycić się zupełnie innym zastosowaniem matematyki w dziedzinie, która jego ciekawi najbardziej. Zachęcamy wszystkich, by sami odnajdywali te drobne powiązania różnych nauk i wierzymy, że otworzy to nasze oczy na piękny i szalenie interesujący świat, w którym zawsze jest coś do odkrycia.

Literatura

[1] red. Gwóźdź-Łukawska G., Potyrała M., MathUp Jak matematyka pomaga w lepszym zrozumieniu rzeczywistości – MathUp2024
[dostęp https://mathup.p.lodz.pl/shared/documents/Monografia_MathUp_2024.pdf]

MathUp
Konferencja Zastosowań Matematyki

TRIANGULACJA – MAGIA TRÓJKĄTÓW W PRAKTYCE

Maksym MALICHENKO
Uniwersytet Ekonomiczny w Krakowie, Kraków

Wstęp

Celem niniejszego artykułu jest przegląd technicznych aspektów metody triangulacji oraz wskazanie jej praktycznych zastosowań w nauce i technologii. Na podstawie prezentacji konferencyjnej przedstawione zostaną przykłady wykorzystania triangulacji w geodezji, optyce, astronomii oraz nowoczesnych systemach lokalizacji opartych o Wi-Fi. W oparciu o rysunki oraz schematy zilustrowane zostaną kluczowe idee i procesy stanowiące punkt wyjścia do głębszej analizy matematycznej i geometrycznej.

Trygonometria i jej praktyczne zastosowania

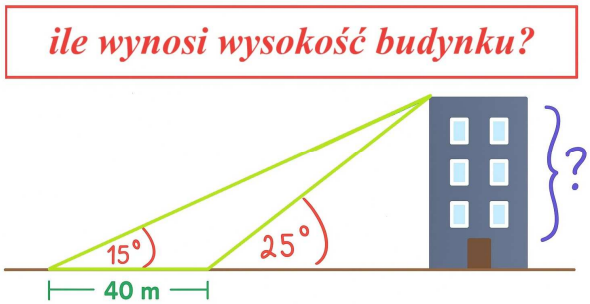
Podstawą metody triangulacji jest trygonometria – dział matematyki badający zależności między bokami i kątami trójkątów. Kluczowym narzędziem jest tu twierdzenie sinusów:

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma},$$

gdzie
α, β, γ – miary kątów w trójkącie,
a, b, c – długości boków leżących naprzeciw odpowiednich kątów.

Jeżeli znane są miary dwóch kątów oraz długość jednego boku, wówczas możliwe jest wyznaczenie miary trzeciego kąta na podstawie sumy miar kątów w trójkącie, a następnie – przy użyciu twierdzenia sinusów – obliczenie długości pozostałych boków. Twierdzenie sinusów pozwala na proporcjonalne powiązanie długości boków z sinusami ich przeciwległych kątów. Znając długość jednego boku oraz miarę odpowiadającego mu kąta, a także miarę jednego z pozostałych kątów, można bezpośrednio obliczyć długość nieznanego boku. Analogicznie, w przypadku znajomości długości dwóch boków oraz miary kąta leżącego naprzeciw jednego z nich, twierdzenie sinusów umożliwia wyznaczenie miary drugiego kąta, a następnie – poprzez sumę kątów – trzeciego. Na tej podstawie można obliczyć długość trzeciego boku.

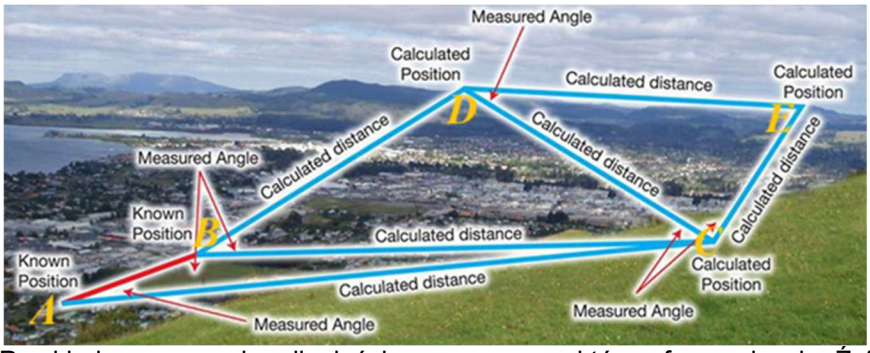
Jednym z najprostszych i najdawniejszych zastosowań twierdzenia sinusów jest mierzenie wysokości obiektów.



Rys. 1 Przykład stosowania twierdzenia sinusów do zmierzenia wysokości budynku.

Czym jest triangulacja

Triangulacja to proces wyznaczania pozycji punktu w przestrzeni za pomocą pomiarów kątów w stosunku do znanych punktów odniesienia. Poniższa ilustracja pokazuje w jaki sposób możemy znaleźć poszczególne odległości bez potrzeby ich faktycznego mierzenia.
W praktyce oznacza to wykorzystanie geometrycznej niezawodności trójkąta jako struktury stabilnej i zamkniętej. Ponieważ możemy stosować znalezione odległości jako nowe punkty odniesienia, proces może być powtarzany wielokrotnie w celu stworzenia całej siatki punktów referencyjnych – co prowadzi nas do kolejnego paragrafu.



Rys. 2 Przykład wyznaczania odległości za pomocą punktów referencyjnych. Źródło: [6]

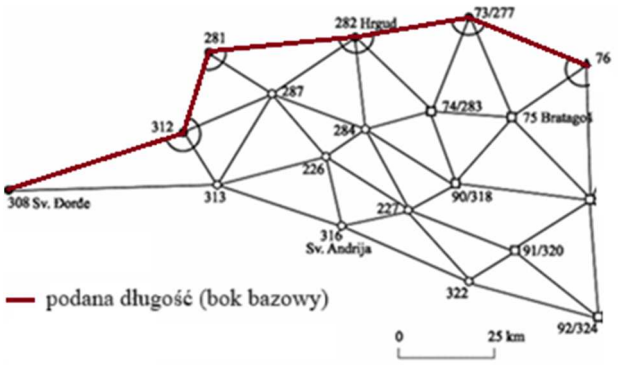
Znając długość jedynie odcinka AB, możemy wyznaczyć długość każdego innego odcinka w konstrukcji powyżej. Dla przykładu, aby obliczyć długość odcinka CD, postępujemy według następującego schematu:

1. Mierzymy długość odcinka AB.
2. W trójkącie ABC dokonujemy pomiaru dwóch kątów wewnętrznych (na podstawie których możemy wyznaczyć trzeci kąt, korzystając z faktu, że suma kątów wewnętrznych trójkąta wynosi 180°).
3. Obliczamy wartości sinusów wszystkich kątów trójkąta ABC.
4. Zastosowawszy twierdzenie sinusów, obliczamy długość odcinka BC (znana jest długość AB oraz wartości sinusów wszystkich kątów trójkąta ABC).
5. W trójkącie BCD mierzymy dwa kąty wewnętrzne i wyznaczamy trzeci.
6. Obliczamy wartości sinusów kątów trójkąta BCD.
7. Korzystając z twierdzenia sinusów, obliczamy długość odcinka CD (dysponujemy długością BC oraz wartościami sinusów wszystkich kątów trójkąta BCD).

MathUp
Konferencja Zastosowań Matematyki

Sieci triangulacyjne do budowania map

Rozbudowane siatki triangulacyjne stanowiły podstawę kartografii przez kilkaset lat. Poniższa ilustracja przedstawia przykładową sieć zbudowaną z trójkątów w której wierzchołkami są punkty triangulacyjne (miasta, miasteczka lub wsie), a boki reprezentują odległości pomiędzy tymi punktami. W takiej sieci znane są długość jednego boku bazowego oraz miary kątów przy wierzchołkach. Bokiem bazowym nazywamy odcinek którego długość została zmierzona, a który jednocześnie jest bokiem jednego z trójkątów w siatce triangulacyjnej. Proces propagacji informacji pomiarowych przez kolejne trójkąty pozwala rozbudowywać siatkę minimalizując błędy pomiarowe. Kluczowym aspektem budowy takich sieci jest zapewnienie redundancji (czyli wprowadzenie nadmiarowych pomiarów, na przykład zmierzenie kilku odcinków, które są bokami trójkątów sieci triangulacyjnej) oraz utworzenie zamkniętych figur kontrolnych (fragmentów siatki triangulacyjnej, w których kolejne trójkąty łączą się w pętlę, a pomiary pozwalają wyznaczyć wszystkie boki i kąty co najmniej dwiema różnymi metodami obliczeniowymi).
Takie figury kontrolne umożliwiają weryfikację dokładności oraz wykrywanie błędów systematycznych – powtarzalnych odchyleń wyników w jednym kierunku, spowodowanych np. niewłaściwą kalibracją instrumentów, błędami obserwatora lub niekorzystnymi warunkami pomiaru. Współczesna geodezja częściowo zastępuje triangulację metodami GNSS (system satelitarny umożliwiający dokładne pomiary lokalizacji i nawigacji), jednak zasady pozostają analogiczne.



Rys. 3 Przykładowa sieć triangulacyjna (wycinek sieci triangulacyjnej Czarnogóry z roku 1946). Źródło: [7]

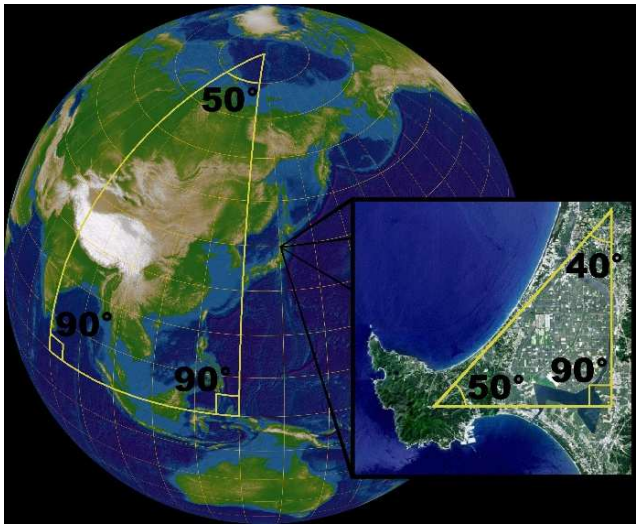
Numery przedstawione na ilustracji powyżej identyfikują punkty triangulacyjne w sieci.

Uwagi do stosowania triangulacji na krzywiźnie Ziemi

Na płaszczyźnie euklidesowej suma miar kątów wewnętrznych w trójkącie zawsze wynosi 180°, ale na powierzchni kuli (geometria sferyczna) jest mniejsza od 540° i większa od 180°. Zatem dla zachowania dokładności pomiarów dużych odległości należy stosować sferyczne twierdzenie sinusów:

$$\frac{\sin a}{\sin \alpha} = \frac{\sin b}{\sin \beta} = \frac{\sin c}{\sin \gamma},$$

gdzie
a, b, c są długościami odcinków sferycznych,
α, β, γ są miarami kątów umieszczonymi naprzeciw boków odpowiednio a, b, c.

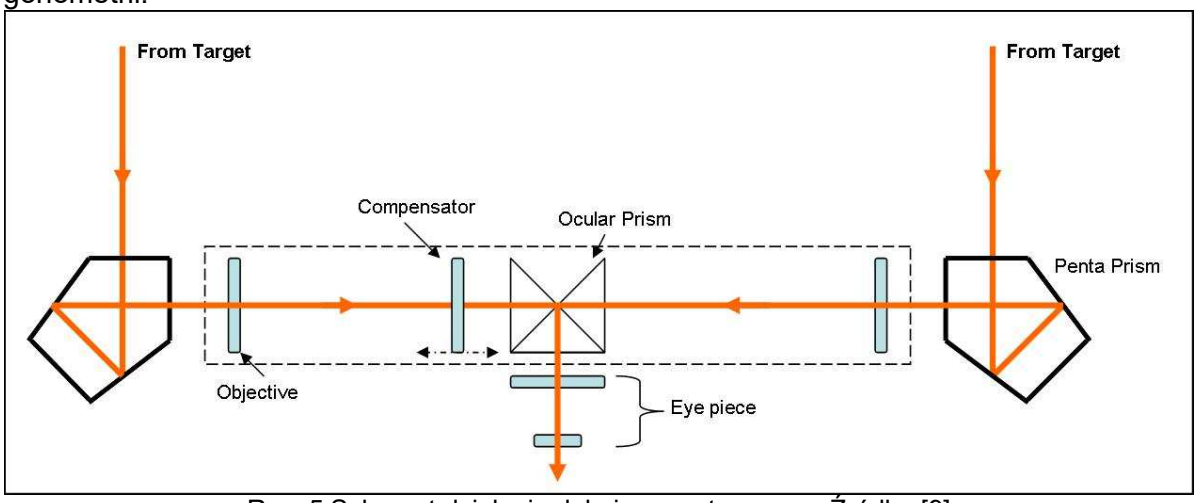


Rys. 4 Przykład różnicy pomiędzy geometrią sferyczna a euklidesową. Źródło: [8]

Zastosowanie w dalmierzach optycznych

Dalmierze optyczne są urządzeniami przeznaczonymi do pomiaru odległości bez konieczności fizycznego pokonywania tej odległości. Konstrukcyjnie składają się z wydłużonej tuby, w której na obu końcach umieszczone są soczewki skierowane do przodu, natomiast okular operatora znajduje się w części środkowej. Zasada działania opiera się na triangulacji: przy znanej bazie (odległości pomiędzy soczewkami) i zmierzonym kącie możliwe jest obliczenie odległości do obserwowanego obiektu. Tego rodzaju urządzenia stanowiły standardowe wyposażenie artylerii polowej w XX wieku oraz były powszechnie stosowane w instrumentach topograficznych.

Schemat poniżej przedstawia zasadę działania dalmierza optycznego. Promienie świetlne z celu wpadają jednocześnie do dwóch pryzmatów pentagonalnych, które zmieniają kierunek biegu wiązki i kierują je do układu optycznego wewnątrz tubusu. Następnie światło przechodzi przez obiektywy i kompensator, którego zadaniem jest zestrojenie obrazów obu wiązek. Dalej promienie trafiają na pryzmat okularowy, gdzie obrazy są nakładane, a obserwator przez okular widzi je jako jeden. Przesunięcie kątowe obrazów, wynikające z różnicy kierunków padania światła na oba wejścia dalmierza, umożliwia obliczenie odległości do celu na podstawie znanej długości bazy i trygonometrii.



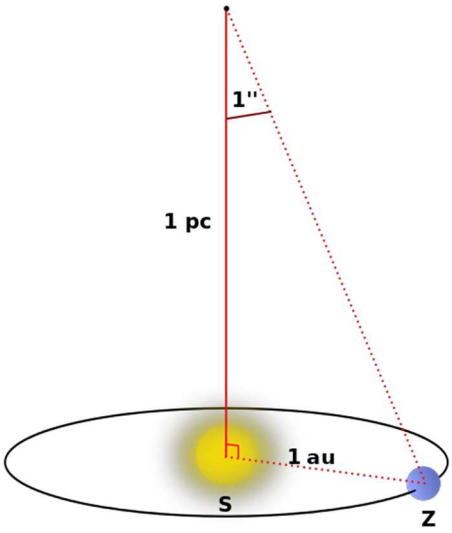
Rys. 5 Schemat działania dalmierza optycznego. Źródło: [9]

Zastosowanie w astronomii (paralaksa)

W astronomii paralaksa odnosi się do zmiany pozornego położenia ciała niebieskiego, gdy obserwator zmienia swoje miejsce obserwacji. Zjawisko to wykorzystuje fakt, że Ziemia porusza się po orbicie wokół Słońca, co pozwala na uzyskanie dwóch różnych punktów widzenia na te same obiekty w przestrzeni. Dwa punkty obserwacji – oddalone o średnią odległość Ziemi od Słońca (czyli o około 150 milionów kilometrów) – tworzą podstawę trójkąta, który jest wykorzystywany do obliczenia odległości do obiektu.
Podstawą triangulacji w przypadku paralaksy jest fakt, że kąt, o jaki przesuwa się gwiazda na tle bardziej odległych obiektów, jest zależny od jej odległości. Ten kąt, określany jako kąt paralaksy, jest bardzo mały, dlatego zwykle wyrażany jest w sekundach kątowych (jednostkach miary kąta, gdzie jedna sekunda to 1/3600 stopnia).

Aby obliczyć odległość do gwiazdy za pomocą paralaksy, mierzymy ten kąt w dwóch różnych punktach na orbicie Ziemi. Kiedy kąt paralaksy jest znany, możemy obliczyć odległość do gwiazdy, dzieląc jednostkową odległość zdefiniowaną jako jeden parsek przez wartość zmierzonego kąta.

W praktyce, jeśli kąt paralaksy wynosi jedną sekundę kątową, oznacza to, że odległość do gwiazdy wynosi jeden parsek, czyli około 3,2616 lat świetlnych. Jeśli kąt jest dwa razy mniejszy, to odległość jest dwa razy większa – i analogicznie dla innych wartości.

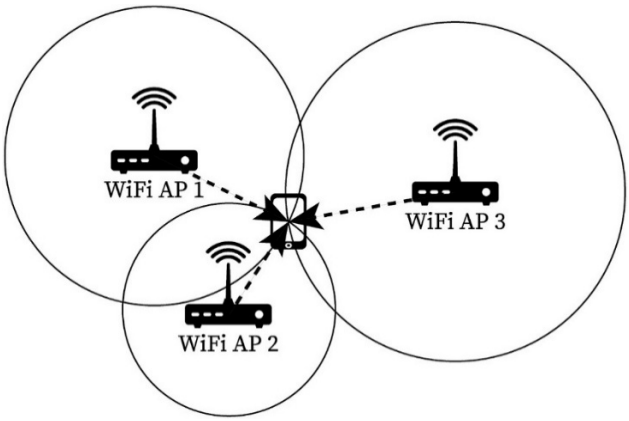


Rys. 6 Graficzna ilustracja parseka. Źródło: [10]

Triangulacja Wi-Fi (trilateracja)

W systemach lokalizacji wewnętrznej, gdzie sygnał GPS nie dociera, wykorzystuje się triangulację oraz trilaterację z sygnałów Wi-Fi. Schemat pokazuje, jak urządzenie odbiera sygnały z trzech routerów, dla których znane są współrzędne.
W metodzie trilateracji obliczamy odległości od każdego z nadajników (np. z siły sygnału RSSI) i wyznaczamy punkt przecięcia okręgów. Przy dostępie do danych o kątach nadejścia (AoA), możliwa jest dokładniejsza triangulacja.

Takie podejście znajduje zastosowanie m.in. w handlu detalicznym, muzeach, logistyce i systemach inteligentnych budynków.



Rys. 7 Przegląd trilateracji w przestrzeni dwuwymiarowej. Źródło: [11]

Podsumowanie

Triangulacja, oparta na prostych zasadach trygonometrii, od wieków służy ludziom do poznawania i opisywania otaczającej rzeczywistości. Od antycznych pomiarów wysokości, przez nowożytne sieci geodezyjne i instrumenty wojskowe, aż po współczesne systemy nawigacji satelitarnej i lokalizacji wewnętrznej – metoda ta pozostaje jednym z najbardziej uniwersalnych narzędzi w nauce i technice. Stabilność geometryczna trójkąta sprawia, że triangulacja nie traci na aktualności, lecz znajduje coraz to nowe zastosowania w świecie zdominowanym przez technologie cyfrowe i systemy bezprzewodowe. Jest to przykład rozwiązania, które dzięki swej prostocie i elegancji przekracza granice epok, łącząc tradycję matematyki z wyzwaniami współczesności.

Literatura

[1] Paralaksa, [w:] Encyklopedia PWN [online], Wydawnictwo Naukowe PWN [dostęp: 25.04.2025]
[2] O'Connor John J., Robertson E. F., Abu Arrayhan Muhammad ibn Ahmad al-Biruni, MacTutor History of Mathematics Archive, University of St Andrews
[3] Yoder P.R., Mounting optics in optical instruments, 2nd Ed., Society of Photo-Optical Instrumentation Engineers, United States, p. 239, 2008
[4] Coxeter H. S. M., Non-Euclidean Geometry, chapters 5, 6, & 7: Elliptic geometry in 1, 2, & 3 dimensions, University of Toronto Press, reissued 1998 by Mathematical Association of America, 1942
[5] Youssef M., Youssef A., Rieger C., Shankar U., Agrawala A., PinPoint, Proceedings of the 4th international conference on Mobile systems, applications and services., pp. 165-176, 10.01.2006
[6] Kumar, S. P., Mahajan S. K., Economical Applications of GPS in Road Projects in India. Procedia-Social and Behavioral Sciences, 96, 2800-2810, 2013
[7] Delčev S., Gučević J., Ogrizović V., Kuhar M., First-order trigonometric network in the former Yugoslavia. Acta Geodaetica et Geophysica, 50(2), 219-241, 2015
[8] Lars H., Rohwedder, Sarregouset, OgaPeninsulaAkiJpLandsat.jpg and Orthographic Projection Japan.jpg, https://commons.wikimedia.org/w/index.php?curid=2717502
[9] Jatosado - Own work, https://commons.wikimedia.org/w/index.php?curid=16586920
[10] Doctore, https://commons.wikimedia.org/w/index.php?curid=26690786
[11] Feng X., Nguyen K., Luo Z., WiFi round-trip time (RTT) fingerprinting: an analysis of the properties and the performance in non-line-of-sight environments. Journal of Location Based Services, 17(4), 307-339. https://doi.org/10.1080/17489725.2023.2239748, 2023

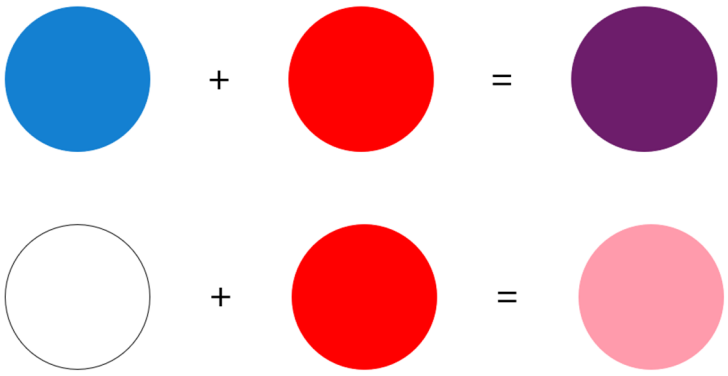
MathUp
Konferencja Zastosowań Matematyki

RGB: Matematyka barw – od kombinacji liniowych do ekranu

Gabriela SMEJDA
Politechnika Łódzka, Łódź

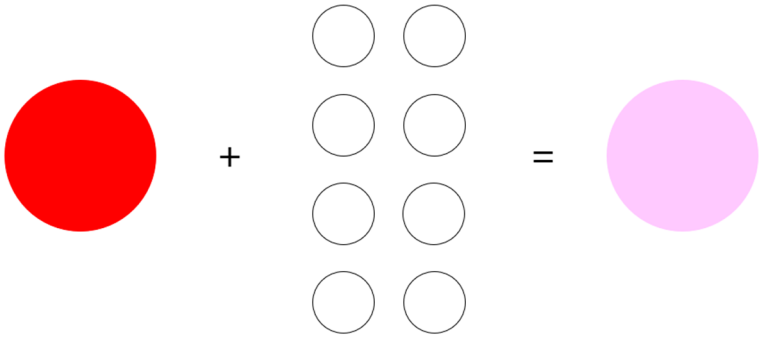
Wstęp

Z mieszaniem kolorów mamy do czynienia już od najmłodszych lat. W tradycyjnym podejściu – na przykład podczas malowania farbami – za barwy podstawowe uznaje się czerwony, żółty i niebieski. Wiadomo, że niebieski w połączeniu z czerwonym daje fioletowy, a dodanie czerwonego do białego pozwala uzyskać kolor różowy.



Rys. 1 Mieszanie kolorów.

Oczywiście efekt końcowy zależy od proporcji, w jakich łączymy poszczególne kolory. Możemy kupić jedną puszkę czerwonej farby i osiem puszek farby białej. Również otrzymamy wtedy kolor różowy, ale jego odcień będzie znacznie jaśniejszy niż w przypadku zmieszania barw w proporcjach jeden do jednego.



Rys. 2 Rola proporcji w mieszaniu kolorów.

Warto zastanowić się, dlaczego za barwy podstawowe uznaje się czerwony, żółty i niebieski oraz czy możliwe jest wybranie innych kolorów podstawowych (tzn. takich, za pomocą których można

uzyskać wszystkie inne kolory). Przykładowo, jeśli kupimy puszki farby czerwonej, niebieskiej i fioletowej, nie będziemy w stanie uzyskać koloru żółtego. Fioletowy z kolei okaże się niepotrzebny, ponieważ można uzyskać go poprzez zmieszanie czerwonego i niebieskiego. Zagadnienie wyboru kolorów podstawowych związane jest z ważnym pojęciem liniowej niezależności wektorów.

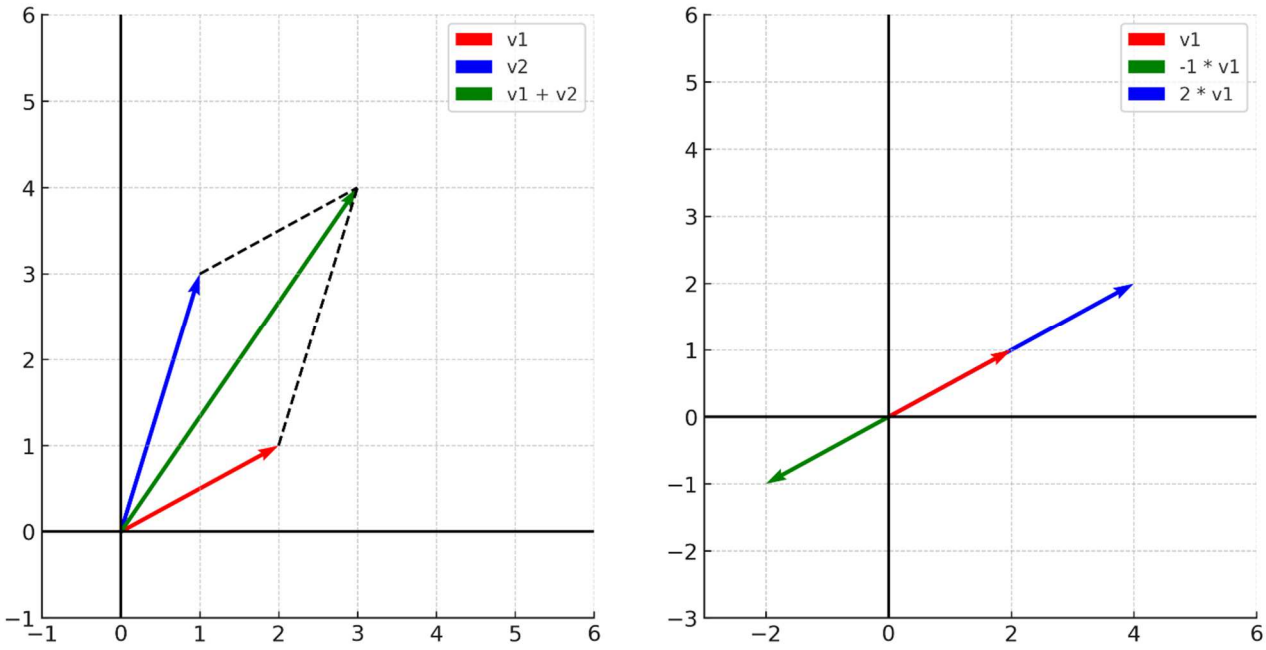
W tym artykule pokażę w jaki sposób mieszanie kolorów można opisać za pomocą narzędzi algebry liniowej, takich jak wektory i kombinacje liniowe. Omówię też najważniejsze modele uzyskiwania barw i wyjaśnię problem wyboru kolorów podstawowych.

Mieszanie kolorów w matematycznym języku

Jak zatem problem mieszania barw przedstawić w języku algebry liniowej?

- Dodawanie wektorów odpowiada za mieszanie kolorów w równych proporcjach, np. czerwony i niebieski daje fioletowy.
- Przemnożenie wektora przez liczbę odpowiada za dobranie proporcji, w jakiej kolory mają zostać zmieszane, np. kupienie ośmiu puszek białej farby zamiast jednej zwiększa „wpływ" tego koloru na wynik.

Działania te zostały zilustrowane na poniższym rysunku.

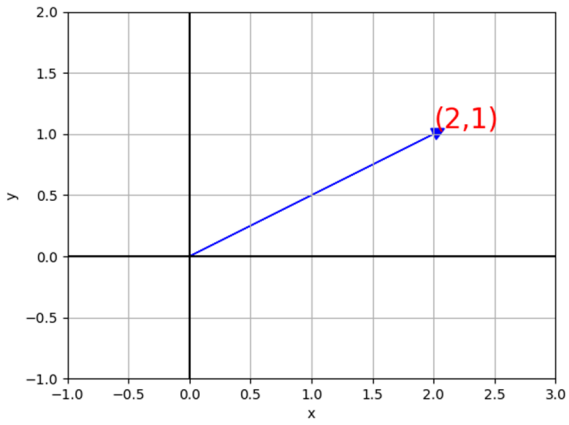


Rys. 3 Działania na wektorach.

Oczywiście taka interpretacja ma swoje ograniczenia. Trudno zinterpretować przemnożenie koloru przez ułamek lub liczbę ujemną, a są to przecież naturalne działania na wektorach. Problematyczne staje się też odejmowanie kolorów, które trudno jest uzasadnić intuicyjnie. Z tego względu konieczne będzie przypisanie kolorom konkretnych wartości liczbowych, na przykład w tytułowym modelu RGB.

Kolory jako wektory

Wektor na płaszczyźnie o początku w punkcie $(0,0)$ można utożsamić z parą współrzędnych odpowiadających jego końcowi, na przykład wektor na Rys. 4. z parą $(2,1) \in \mathbf{R}^2$.



Rys. 4 Wektor o początku w punkcie $(0,0)$ i końcu w punkcie $(2,1)$.

Możemy również rozważać wektory w przestrzeni trójwymiarowej $\mathbf{R}^3$. Jej elementy mają trzy współrzędne, np. $(2,1,5)$.
Pojęcie wektora możemy rozszerzyć na n-wymiarową przestrzeń. Mamy wtedy n współrzędnych. Definiujemy działania dodawania wektorów i mnożenia wektora przez liczbę następująco:

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$$
$$a(x_1, \ldots, x_n) = (ax_1, \ldots, ax_n)$$

dla $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathbf{R}^n$, $a \in \mathbf{R}$.

W dalszej części ważne będzie pojęcie kombinacji liniowej wektorów. Rozważmy najpierw dwa wektory.

Definicja. Kombinacją liniową dwóch wektorów $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$ o współczynnikach $a, b \in \mathbf{R}$ nazywamy wektor

$$\mathbf{u} = a\mathbf{v} + b\mathbf{w} \in \mathbf{R}^n.$$

W świecie kolorów dobrym przykładem kombinacji liniowej jest wspomniany we wstępie problem wskazania proporcji farb. Wektory odpowiadają tutaj kolorom, a współczynniki informują o liczbach puszek farby, czyli mamy

$$jasnoróżowy = 1\, czerwony + 8\, biały$$

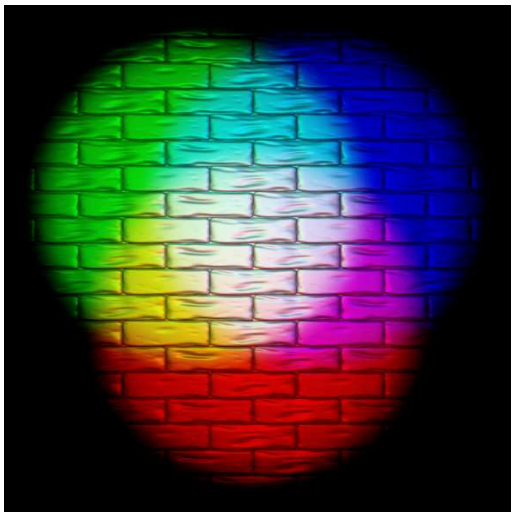
Definicję kombinacji liniowej można rozszerzyć również na m wektorów.

Definicja. Kombinacją liniową wektorów $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbf{R}^n$ o współczynnikach $a_1, \ldots, a_m \in \mathbf{R}$ nazywamy wektor

$$\mathbf{u} = a_1\mathbf{v}_1 + \cdots + a_m\mathbf{v}_m \in \mathbf{R}^n.$$

Model RGB

Model RGB to addytywny model mieszania barw, w którym kolorami podstawowymi są: czerwony, zielony i niebieski. Wykorzystywany jest w urządzeniach emitujących światło, takich jak telewizory, monitory czy projektory. W modelu tym światła o określonych długościach fal nakładają się na siebie, a ich intensywności się sumują.
Najłatwiej wyobrazić sobie to jako czarną ścianę, na którą padają trzy wiązki światła: czerwona, zielona i niebieska. W miejscach, gdzie nie dociera żadne światło, ściana pozostaje czarna. Tam, gdzie nakładają się dwa kolory, powstają barwy pośrednie – na przykład czerwony z zielonym daje żółty. W centralnym obszarze, gdzie łączą się wszystkie trzy światła, otrzymujemy kolor biały. Efekt ten można interpretować jako wynik coraz większego natężenia światła – im więcej barw się nakłada, tym jaśniejszy jest efekt wizualny.



Rys. 5 Kolory w modelu RGB.

Naturalną wątpliwością jest kwestia kolorów podstawowych. W tradycyjnym malarstwie przyjmuje się, że barwy podstawowe to czerwony, żółty i niebieski. Dlaczego więc w modelu RGB podstawową barwą jest zielony – kolor, który przecież powstaje z połączenia żółtej i niebieskiej farby? Odpowiedź jest prosta: malarstwo i RGB to dwa różne modele barw. Malarstwo opiera się na modelu subtraktywnym, w którym kolory uzyskuje się przez odejmowanie (pochłanianie) światła. Tymczasem RGB to model addytywny, oparty na dodawaniu światła. Do tej różnicy jeszcze wrócimy – teraz wykorzystamy matematyczne pojęcia wektorów i kombinacji liniowych do opisu działania modelu RGB.

Matematyczny opis modelu RGB

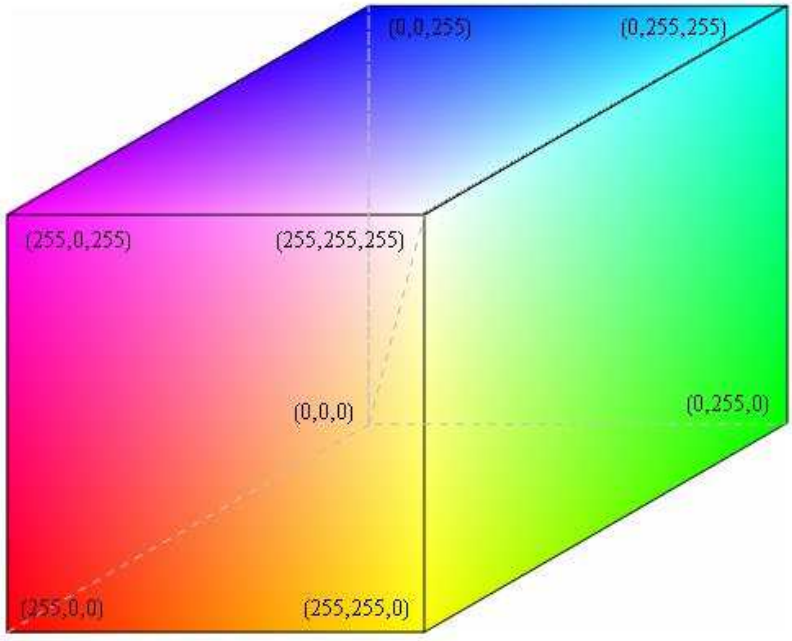
W modelu RGB przypisujemy trzem kolorom podstawowym określone wektory. Są to odpowiednio:

- czerwony $\mathbf{r} = (255,0,0)$,
- zielony $\mathbf{g} = (0,255,0)$,
- niebieski $\mathbf{b} = (0,0,255)$.

Wszystkie kolory w modelu RGB możemy interpretować jako kombinacje liniowe kolorów podstawowych

$$a_1\mathbf{r} + a_2\mathbf{g} + a_3\mathbf{b}$$

dla $a_1, a_2, a_3 \in [0,1]$.
Użycie współczynników z przedziału $[0,1]$ sprawia, że każda ze współrzędnych dowolnego koloru jest z zakresu od 0 do 255.
Kolor biały w tym modelu to $(255,255,255)$. Jest to suma trzech kolorów podstawowych. Z kolei kolor czarny to $(0,0,0)$. Wszystkie kolory w modelu RGB można przedstawić na sześcianie.



Rys. 6 Sześcian kolorów RGB.

Każda współrzędna określa zatem udział jednego z trzech kolorów podstawowych: czerwonego, zielonego i niebieskiego. Na przykład kolor żółty ma współrzędne $(255, 255, 0)$. Oznacza to, że żółty powstaje ze zmieszania w równych proporcjach koloru czerwonego i zielonego, ale bez udziału koloru niebieskiego.

Zauważmy, że przemnożenie koloru przez liczbę określa jego nasycenie.

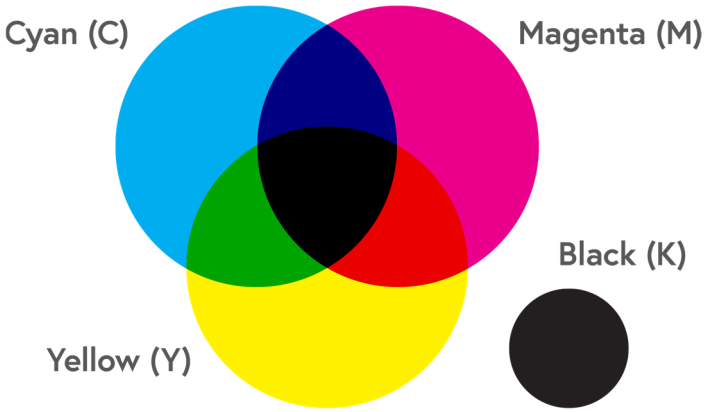
Rys. 7 Wartości $a\mathbf{r}$ dla $a = \frac{10}{10}, \frac{9}{10}, \ldots, \frac{1}{10}, \frac{0}{10}$.

Przyjrzyjmy się teraz przekątnej podstawy sześcianu. Znajdują się tam kolory, które powstały poprzez połączenie koloru zielonego i czerwonego w określonych proporcjach. Te kolory są zatem kombinacjami liniowymi zielonego i czerwonego.

Rys. 8 Kombinacje liniowe $a\mathbf{g} + b\mathbf{r}$ dla $a = \frac{10}{10}, \frac{9}{10}, \ldots, \frac{1}{10}, \frac{0}{10}$ oraz $b = 1 - a$.

Modele addytywne i subtraktywne

Model RGB jest przykładem modelu addytywnego, w którym kolory powstają poprzez sumowanie intensywności świateł. Warto wspomnieć o jego przeciwieństwie — modelu subtraktywnym, w którym mamy do czynienia ze światłem odbitym od przedmiotów. Znalazł on zastosowanie między innymi w malarstwie i poligrafii.
W modelu subtraktywnym kolory uzyskuje się poprzez pochłanianie (odejmowanie) określonych długości fali światła przez barwniki. W praktyce oznacza to, że zamiast dodawać światło do ciemnego tła (jak w RGB), zaczynamy od jasnego podłoża — na przykład białej kartki — a nanoszone pigmenty stopniowo ograniczają ilość światła, które zostaje odbite. Tak działa stosowany w druku model CMYK, w którym kolorami podstawowymi są cyjan, magenta i żółty. Teoretycznie, zmieszanie tych trzech barw w równych proporcjach daje kolor czarny. W praktyce jednak często dodaje się osobny czarny tusz, oznaczony literą K (od ang. *Key*), ze względu na jego niższy koszt i lepszą jakość druku.



Rys. 9 Kolory w modelu CMYK.

Malowanie również opiera się na modelu subtraktywnym, ale warto zauważyć, że tradycyjnie uznawane kolory podstawowe (czerwony, żółty, niebieski) nie są optymalne z punktu widzenia fizyki światła. Fotoreceptory w ludzkim oku są najbardziej czułe na światło czerwone, zielone i niebieskie — stąd wybór barw w modelu RGB. W modelu CMY (lub CMYK) za kolory podstawowe przyjmuje się cyjan, magentę i żółty, ponieważ umożliwiają one najpełniejsze sterowanie pochłanianiem światła:

- żółty pochłania niebieskie światło (odbija czerwone i zielone),
- magenta pochłania zielone (odbija czerwone i niebieskie),
- cyjan pochłania czerwone (odbija zielone i niebieskie).

Dzięki temu możliwe jest precyzyjne sterowanie barwą odbitego światła i uzyskiwanie szerokiej gamy kolorów z ograniczonej liczby pigmentów.

Kolory podstawowe w modelu RGB

Jak wspomniałam wcześniej, kolory podstawowe w modelu RGB zostały dobrane tak, aby możliwie najlepiej odpowiadały czułości trzech typów fotoreceptorów w ludzkim oku — odpowiedzialnych za odbieranie światła czerwonego, zielonego i niebieskiego. Nie oznacza to jednak, że tylko te trzy barwy mogą być używane jako podstawowe. Teoretycznie istnieje wiele trójek kolorów, które mogą pełnić taką funkcję — pod warunkiem, że pozwalają uzyskać odpowiednio szeroki zakres kolorów poprzez ich mieszanie.

Warto jednak wiedzieć, kiedy dany zestaw kolorów nie może być traktowany jako podstawowy. Przykładowo, jeśli w przypadku malowania wybierzemy jako „podstawowe" kolory niebieski, czerwony i fioletowy, szybko okaże się, że nie da się z ich połączenia uzyskać barw takich jak zielony czy żółty. Dzieje się tak, ponieważ fioletowy sam jest mieszaniną czerwieni i niebieskiego, a więc nie wnosi nowej informacji. W tym przypadku jest możliwe zapisanie trzeciego koloru za pomocą dwóch pozostałych — dzięki matematycznym narzędziom zobaczymy, że jest to kluczowa własność do sprawdzenia.
Wracając do modelu RGB, możemy rozważyć kolory

Rys. 10 Kolejno kolory: $(255,0,0), (0,255,0), (128,128,0)$.

gdzie pierwszy to czerwony $\mathbf{r} = (255, 0, 0)$, drugi to zielony $\mathbf{g} = (0, 255, 0)$, a trzeci jest kombinacją liniową tych dwóch kolorów: $\frac{1}{2}\mathbf{g} + \frac{1}{2}\mathbf{r} \approx (128, 128, 0)$. Zauważmy, że nie pozwalają one wygenerować dowolnego koloru. Nie uzyskamy w ten sposób na przykład koloru niebieskiego – w kombinacjach liniowych trzecią współrzędną będzie zawsze zero.

Okazuje się, że w modelu RGB umieszczonym w trójwymiarowej przestrzeni sprawdzanie, czy trzy kolory są podstawowe, sprowadza się do sprawdzania, czy odpowiadające im wektory są liniowo niezależne.

Definicja. Układ wektorów $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbf{R}^n$ nazywamy liniowo zależnym, jeżeli istnieją takie współczynniki $a_1, \ldots, a_m \in \mathbf{R}$, nie wszystkie równe zero, że

$$a_1\mathbf{v}_1 + \cdots + a_m\mathbf{v}_m = 0_{\mathbf{R}^n}.$$

Układ wektorów $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbf{R}^n$ nazywamy liniowo niezależnym, jeśli nie jest liniowo zależny.

Wygodnie myśleć o liniowej zależności przy pomocy poniższego twierdzenia. Zostało ono już tak naprawdę wykorzystane w rozważanych wcześniej przykładach.

Twierdzenie. Niech $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbf{R}^n$. Następujące warunki są równoważne:

- układ wektorów $\mathbf{v}_1, \ldots, \mathbf{v}_m$ jest liniowo zależny,
- jeden z wektorów $\mathbf{v}_1, \ldots, \mathbf{v}_m$ jest kombinacją liniową pozostałych wektorów.

Przedstawimy teraz kilka przykładów ilustrujących pojęcia liniowej zależności i niezależności wektorów.

Przykład. W przestrzeni $\mathbf{R}^2$ układ wektorów $(1,1), (2,2)$ jest liniowo zależny, ponieważ

$$(2,2) = 2(1,1).$$

Przykład. W przestrzeni $\mathbf{R}^2$ układ wektorów $(1,-1),(1,1),(2,1)$ jest liniowo zależny, ponieważ

$$(1,-1) = -3(1,1) + 2(2,1).$$

Poniższy przykład to matematyczne uzasadnienie tego, że czerwony, zielony i niebieski to kolory podstawowe w modelu RGB.

Przykład. W przestrzeni $\mathbf{R}^3$ układ wektorów $\mathbf{r}, \mathbf{g}, \mathbf{b}$, gdzie $\mathbf{r} = (255,0,0)$, $\mathbf{g} = (0,255,0)$, $\mathbf{b} = (0,0,255)$ jest liniowo niezależny. Istotnie, jeżeli

$$a_1(255,0,0) + a_2(0,255,0) + a_3(0,0,255) = (0,0,0),$$

to $a_1 = a_2 = a_3 = 0$.

Rozważymy teraz inną przykładową trójkę kolorów podstawowych w modelu RGB. Zwrócimy uwagę na założenia dotyczące współczynników w kombinacji liniowej. Przypomnijmy, że w klasycznym modelu RGB należą one do przedziału $[0,1]$.
Mianowicie weźmy trzy kolory $\mathbf{v_1} = (255,\ 84,\ 0)$, $\mathbf{v_2} = (85,\ 85,\ 0)$, $\mathbf{v_3} = (50,\ 80,\ 50)$.

Rys. 11 Kolejno kolory: $(255,84,0), (85,85,0), (50,80,50)$.

Wtedy na przykład kombinacja liniowa $\frac{1}{3}\mathbf{v_1} + \mathbf{v_2} + \frac{3}{2}\mathbf{v_3}$ to kolor:

$$\frac{1}{3}(255,84,0) + (85,85,0) + \frac{3}{2}(50,80,50) = (245,233,75).$$

Rys. 12 Kolor $(245,233,75)$.

Pokażemy, że powyższe trzy wektory $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{v_3}$ są liniowo niezależne, a więc odpowiadające im kolory są kolorami podstawowymi w modelu RGB. Niech $\alpha, \beta, \gamma \in \mathbf{R}$. Mamy

$$\alpha\mathbf{v_1} + \beta\mathbf{v_2} + \gamma\mathbf{v_3} = 0_{\mathbf{R}^3} \Leftrightarrow \alpha(255,84,0) + \beta(85,85,0) + \gamma(50,80,50) = (0,0,0) \Leftrightarrow$$

$$\begin{cases} 255\alpha + 85\beta + 50\gamma = 0 \\ 84\alpha + 85\beta + 80\gamma = 0 \\ 50\gamma = 0 \end{cases}.$$

Z trzeciego równania $\gamma = 0$ i podstawiając do dwóch pierwszych równań otrzymujemy

$$\begin{cases} 255\alpha + 85\beta = 0 \\ 84\alpha + 85\beta = 0 \end{cases}.$$

Odejmując drugie równanie od pierwszego, dostajemy $171\alpha = 0$, a stąd $\alpha = 0$. Po podstawieniu mamy również $\beta = 0$. Skoro $\alpha = \beta = \gamma = 0$, to układ wektorów $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ jest liniowo niezależny. Przy założeniu, że można stosować kombinacje liniowe z ujemnymi współczynnikami, istnieje zatem możliwość wygenerowania z nich wszystkich kolorów z RGB.

Zauważmy, że gdyby współczynniki należały do przedziału $[0,1]$, to nie dałoby się uzyskać koloru o składowej niebieskiej większej niż 50. Na przykład czysty niebieski $(0, 0, 255)$ wymagałby przy „nowych" kolorach podstawowych współczynnika 5.1 przy wektorze $\mathbf{v_3}$.

Literatura

[1] https://mimuw.edu.pl/~amecel/gal1_mecel.pdf
[2] https://science.howstuffworks.com/primary-colors.htm
[3] https://en.wikipedia.org/wiki/RGB_color_model
[4] https://en.wikipedia.org/wiki/CMYK_color_model
[5] https://www.whydomath.org/node/wavlets/imagebasics.html
[6] https://medium.com/swlh/how-to-understand-linear-independence-linear-algebra-8bab1d918509
[7] https://mixam.com/support/cmykchart

MathUp
Konferencja Zastosowań Matematyki

CO MA PERSPEKTYWA DO GRY W DOBBLE, CZYLI KILKA SŁÓW O PŁASZCZYZNACH PROJEKCYJNYCH

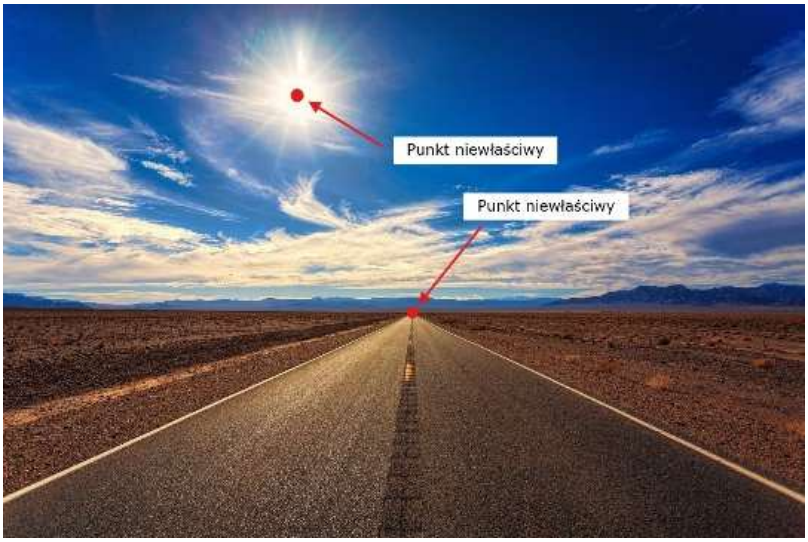
Karolina FISIAK
Politechnika Łódzka, Łódź

Wstęp

Dobble to gra karciana składająca się z talii 55 kart z 8 symbolami na każdej karcie. Karty te mają taką własność, że na dwóch dowolnych z nich znajduje się dokładnie jeden wspólny symbol. Można pomyśleć, że stworzenie talii kart spełniającej taki warunek nie jest trudne, ale wielkość talii znacznie utrudnia to zadanie. Można więc dojść do wniosku, że autorzy gry korzystali z jakiegoś algorytmu przy jej tworzeniu. W artykule tym omówimy, jakim matematycznym zagadnieniem da się opisać karty w tej grze, a także wyjaśnimy algorytm, który pozwoli podobną grę stworzyć. Przedstawianym pojęciem będą płaszczyzny rzutowe (inaczej projekcyjne).

Płaszczyzna rzutowa (projekcyjna)

Zanim podamy warunki, które musi spełniać płaszczyzna, żeby można ją było nazwać płaszczyzną rzutową, opiszemy jej najważniejszą cechę, aby łatwiej było zrozumieć główną ideę, która stoi za tym pojęciem. Najważniejszą cechą płaszczyzny rzutowej jest to, że nie ma na niej żadnych prostych równoległych, to znaczy takich, które się nie przecinają. Prawdopodobnie każdy z nas miał z taką płaszczyzną do czynienia – pomyślmy o sytuacji takiej jak na zdjęciu poniżej (Rys. 1). Wiemy, że w rzeczywistości pasy po bokach drogi albo promienie słoneczne nigdzie się nie przetną, ale dzięki perspektywie, jaką zastosowano, wydaje się jakby gdzieś w nieskończoności istniały takie punkty, nazywane punktami niewłaściwymi lub punktami w nieskończoności (*ang. vanishing point*), w których proste te się przetną. W przykładzie z drogą będzie to zaznaczony punkt na horyzoncie, a w przykładzie z promieniami słonecznymi – słońce.

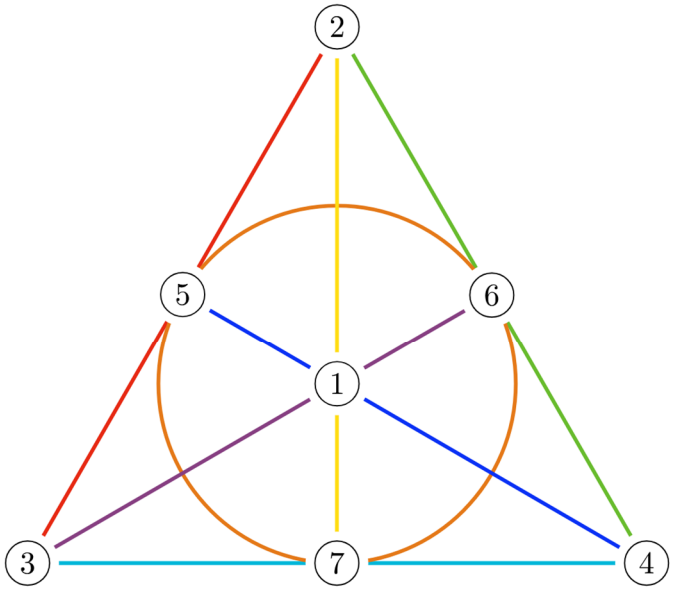


Rys. 1 Źródło: https://pixabay.com/photos/death-valley-desert-highway-road-3133502/.

Płaszczyzną rzutową nazywamy trójkę (P, L, I) składającą się ze zbioru punktów P, zbioru prostych L oraz symetrycznej relacji I określonej na zbiorze $P \cup L$ spełniającej następujące warunki:

1. dwa dowolne punkty łączy dokładnie jedna prosta;
2. dwie dowolne proste mają dokładnie jeden wspólny punkt;
3. istnieją cztery punkty takie, że nie istnieje linia łącząca więcej niż dwa z nich.

Przykładem płaszczyzny projekcyjnej jest płaszczyzna Fano (Rys. 2), czyli taka, która składa się z 7 punktów i 7 prostych. Prostą byłaby np. zielona linia łącząca punkty 2, 4 i 6 albo pomarańczowa linia z punktami 5, 6 oraz 7. Mimo, że intuicyjnie nie postrzegamy pomarańczowego okręgu jako prostą, to należy pamiętać, że schemat jest tylko wizualizacją abstrakcyjnego tworu, więc nawet jeśli pomarańczowy okrąg umyka intuicji prostej, to możemy go jako prostą traktować.



$$P = \Big\{\{2,3,5\},\{2,4,6\},\{3,4,7\},\{5,6,7\},\{1,2,7\},\{1,4,5\},\{1,3,6\}\Big\}$$

Rys. 2 Przykład płaszczyzny rzutowej – płaszczyzna Fano.

Dobble jako płaszczyzna rzutowa

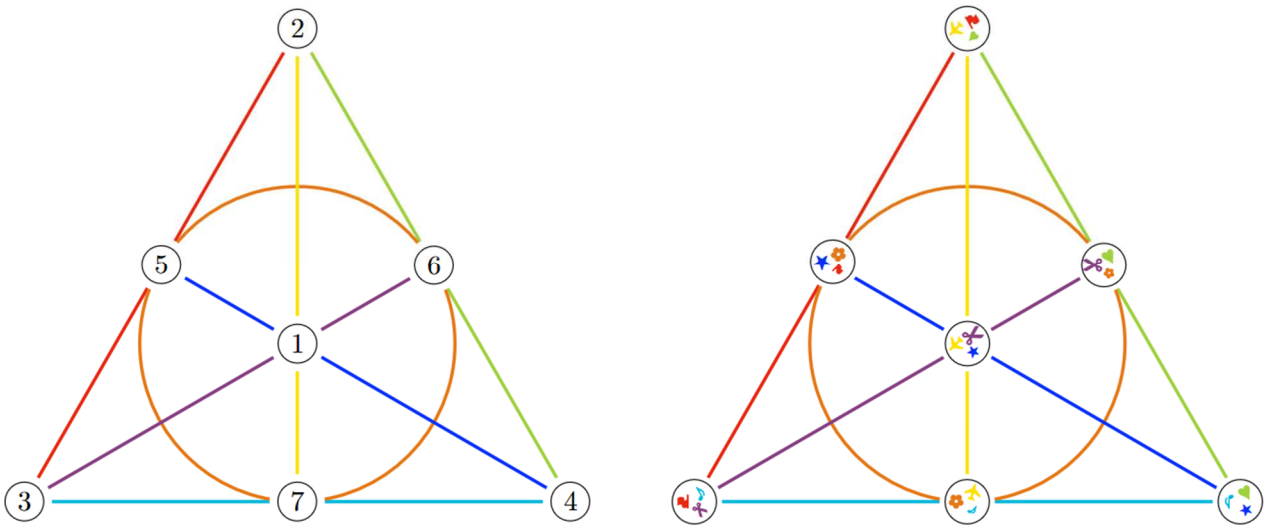
Jak więc płaszczyzny projekcyjne łączą się z grą Dobble? Jeśli zamienimy w warunku 1. definicji płaszczyzny rzutowej słowa *punkty* na *karty*, a *prosta* na *symbol*, to z warunku

dwa dowolne PUNKTY łączy dokładnie jedna PROSTA

otrzymamy

dwie dowolne KARTY łączy dokładnie jeden SYMBOL,

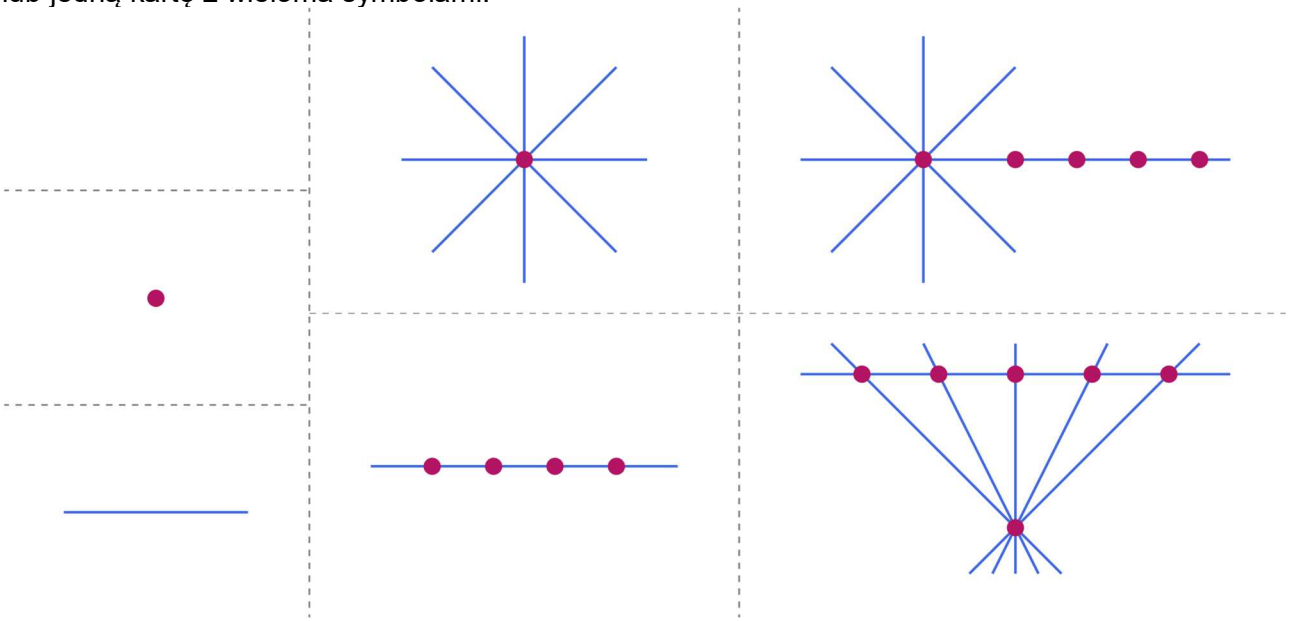
czyli dokładnie ten warunek, który jest spełniony w Dobble. Wówczas przykładowa talia składająca się z 7 kart i 7 symboli wyglądałaby następująco:



Rys. 3 Zilustrowanie talii składającej się z 7 kart.

Przykładowo, zielona prosta zamieniłaby się na symbol zielonego serca, a punkt 6 na kartę składającą się z symboli: zielone serce, pomarańczowy kwiatek i fioletowe nożyczki.

Dlaczego jednak warunki 1. – 3. są takie, a nie inne? Na pierwszy rzut oka możemy zauważyć symetrię między warunkami 1. i 2., o czym będzie mowa w dalszej części artykułu. Warunek 3. jest nam potrzebny, żeby uniknąć sytuacji zdegenerowanych, pokazanych na Rysunku 4. W tworzeniu talii kart gwarantuje nam to wykluczenie sytuacji, gdy za grę uważać będziemy np. jedną kartę bez żadnych symboli; symbol bez żadnych kart; talię kart, na których widnieje tylko jeden symbol lub jedną kartę z wieloma symbolami.



Rys. 4 Płaszczyzny zdegenerowane.

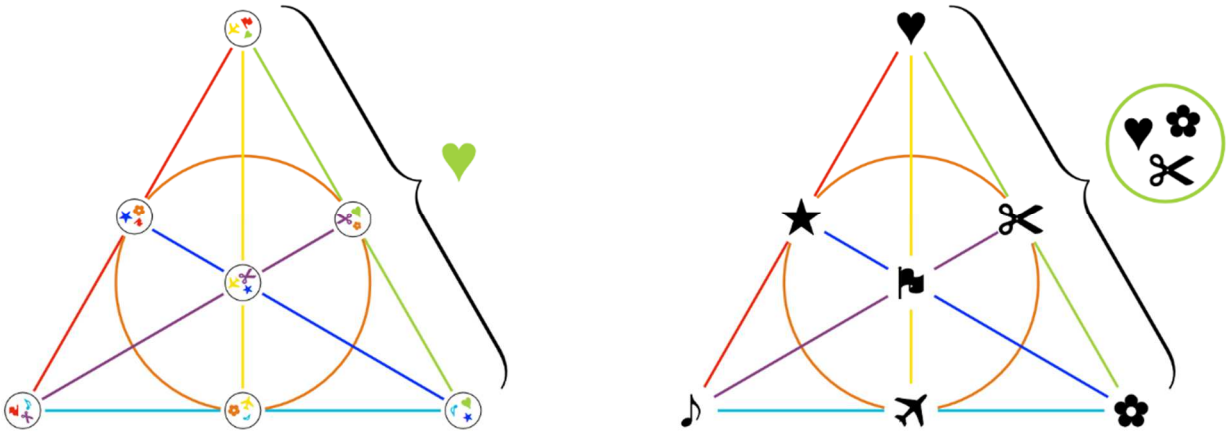
Rząd płaszczyzny i dualność prostych

Aby odróżnić od siebie różne płaszczyzny wprowadzamy pojęcie rzędu płaszczyzny. Pozwala ono na pewną klasyfikację płaszczyzn. Z definicji płaszczyzny rzutowej wynika, że liczba punktów $n+1$ na każdej prostej jest równa liczbie prostych przechodzących przez dowolny punkt. Rzędem płaszczyzny nazwiemy liczbę n. Jeśli rząd płaszczyzny oznaczymy przez n, to zachodzi:

	W ogólności	Płaszczyzna Fano
Rząd płaszczyzny	n	2
Liczba punktów na każdej prostej	$n+1$	3
Liczba prostych przechodzących przez dowolny punkt		
Liczba wszystkich prostych	n^2+n+1	7
Liczba wszystkich punktów		

Patrząc na tabelę, możemy zauważyć podobieństwo prostych i punktów – liczba punktów na dowolnej prostej jest równa liczbie prostych przechodzących przez dowolny punkt oraz liczba wszystkich punktów jest równa liczbie wszystkich prostych. Okazuje się, że proste i punkty są na płaszczyznach rzutowych obiektami dualnymi, tzn. możemy je traktować zamiennie, a dowolne twierdzenie wyrażone dla prostych będzie prawdziwe także dla punktów i na odwrót. Dzięki ich dualności możemy wcześniejszy schemat dotyczący gry zmodyfikować.

Otrzymamy wówczas:



Rys. 5 Zilustrowanie dualności punktów i prostych.

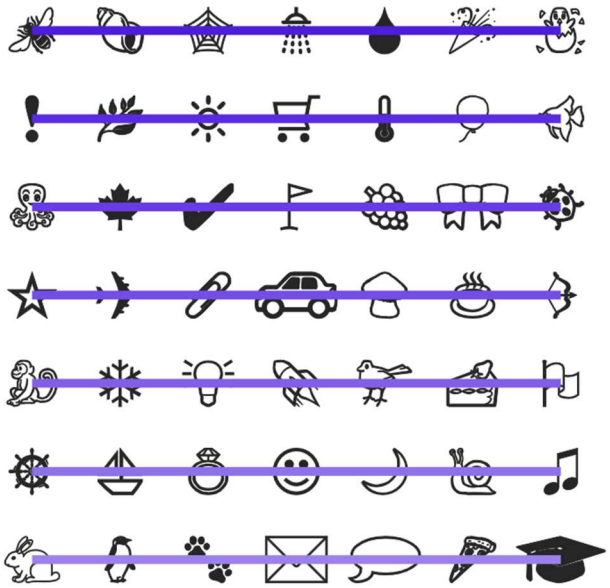
Zatem karty zamieniły się na symbole, a symbole na karty. W ten sposób zielony odcinek wcześniej symbolizujący zielone serce, symbolizuje teraz zieloną kartę składającą się z serca, nożyczek i kwiatka, a punkt dopowiadający wcześniej karcie z symbolami serca, nożyczek i kwiatka, odpowiada obecnie symbolowi nożyczek.

W dalszej części artykułu będziemy korzystać z wizualnej reprezentacji płaszczyzny przedstawionej po prawej stronie Rysunku 5, gdyż jest bardziej przejrzysta od tej pokazanej po lewej.

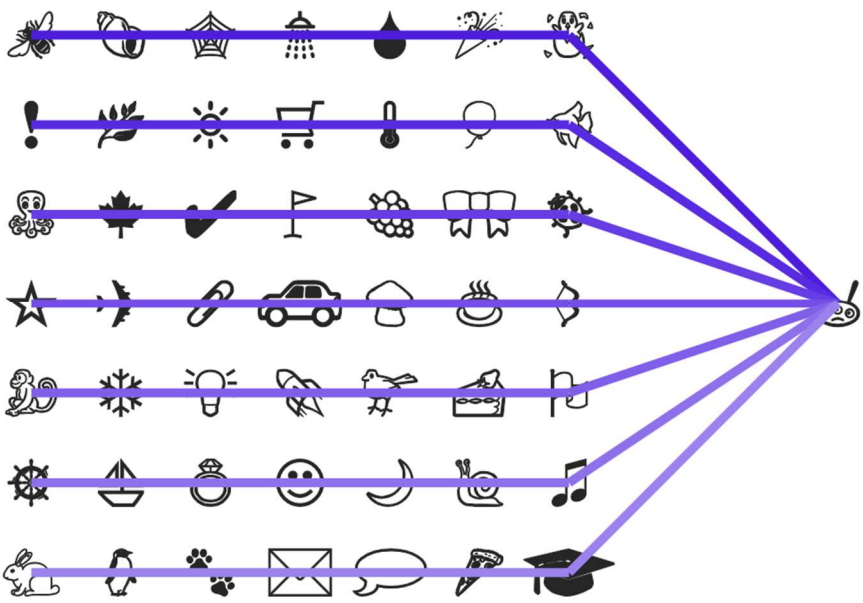
Algorytm tworzenia własnych kart Dobble

Przejdziemy teraz do opisania algorytmu, który pozwoli nam na stworzenie talii kart mających taką własność jaką ma Dobble. Skoro wiemy, że na każdej karcie jest po 8 symboli, to zgodnie ze wzorem z tabeli $n+1=8$. Wiemy zatem, że płaszczyzna, która opisuje grę Dobble ma rząd równy $n=7$. Algorytm opiszemy w kilku krokach.

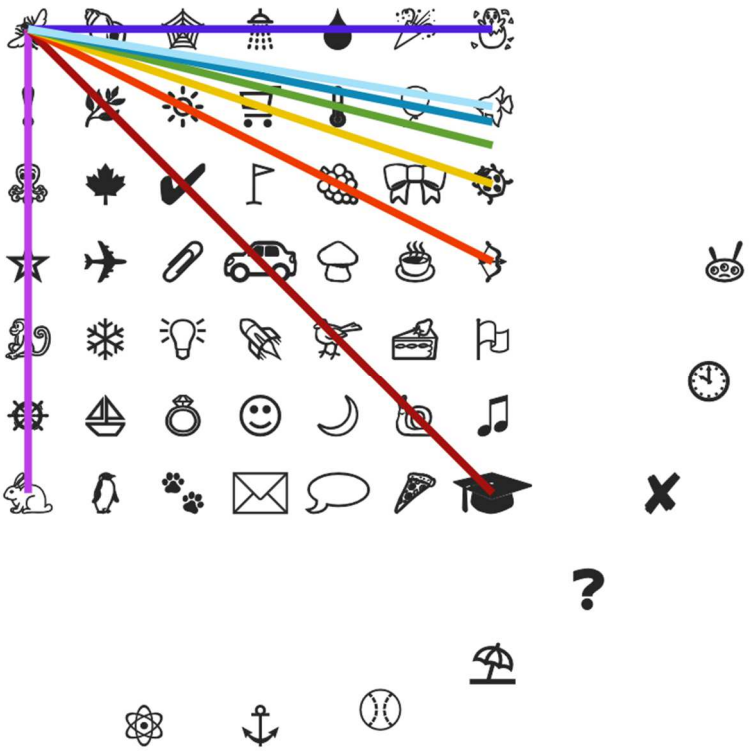
Rys. 6 Krok 1: Narysowanie płaszczyzny 7×7.



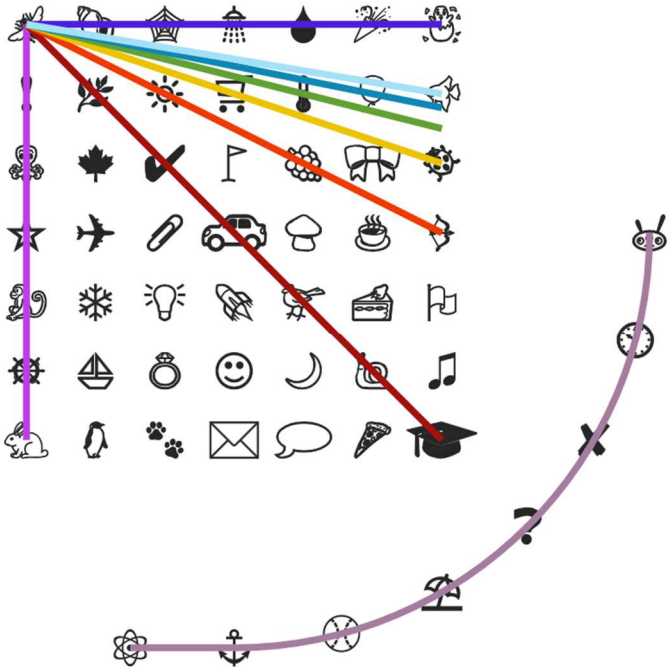
Rys. 7 Krok 2: Zaznaczenie wszystkich prostych, które wyglądają jak proste równoległe.



Rys. 8 Krok 3: Dołożenie punktu w nieskończoności (punktu niewłaściwego, punktu na ,,horyzoncie'') i połączenie wszystkich prostych z tym właśnie punktem.

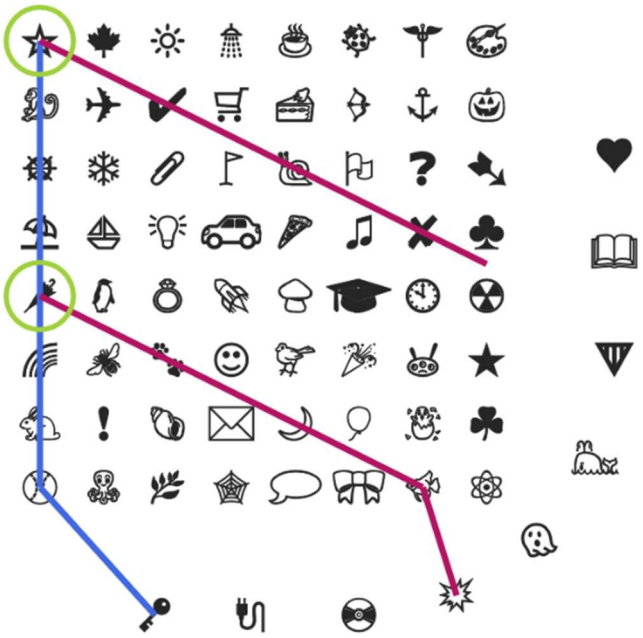


Rys. 9 Krok 4: Wykonanie tej procedury dla prostych o dowolnym nachyleniu.



Rys. 10 Krok 5: Połączenie punktów niewłaściwych jedną prostą („horyzontem”).

Można by zadać pytanie: *Czy postępując w ten sposób możemy stworzyć karty do gry, której podstawą byłaby płaszczyzna rzędu 8?* Odpowiedź na to pytanie brzmi: *Nie*. Zauważmy, że postępując według uprzednio podanego schematu znaleźlibyśmy dwie proste (karty), które miałyby dwa punkty (symbole) wspólne, co nie spełnia najważniejszego warunku gry, którą chcemy stworzyć. Nie oznacza to oczywiście od razu, że gry opartej na płaszczyźnie rzędu 8 stworzyć się nie da – pokazuje to jedynie, że przedstawiony algorytm nie jest bez wad. W rzeczywistości, jest on dobry, gdy rząd płaszczyzny jest liczbą pierwszą.



Rys. 11 Przypadek, gdy podany algorytm nie zadziała – płaszczyzna rzędu 8.

Na początku może się to wydawać rozczarowujące, ale kiedy zorientujemy się, że sami matematycy nie są pewni, jakie płaszczyzny istnieją, okaże się, że algorytm ten jest całkiem przydatny, a jego istnienie, nawet jeśli tylko dla liczb pierwszych, jest pewnym osiągnięciem.

To jak to jest z tymi płaszczyznami? Co właściwie wiadomo, a czego nie? Wiemy na pewno, że jeśli liczba n jest potęgą liczby pierwszej (np. $2, 3, 8, 9, 16$), to płaszczyzna rzutowa rzędu n istnieje. Istnieje też hipoteza, że zachodzi implikacja odwrotna, czyli jeśli płaszczyzna ma rząd równy n, to n jest potęgą liczby pierwszej. Jest to na razie jednak tylko hipoteza. Pokazano (był to dowód przeprowadzony z pomocą komputera), że nie istnieje płaszczyzna, która miałaby rząd równy 10, natomiast kwestia istnienia płaszczyzny rzędu 12 pozostaje otwarta.

Literatura

[1] Krych M., Co to jest: Geometria rzutowa?, Delta 4/2019, p.5, 2019
[2] Czerwiński W., Dobble, Delta 6/2018, p.7, 2018
[3] Weisstein E. W., Projective Plane, *MathWorld*--A Wolfram Web Resource, https://mathworld.wolfram.com/ProjectivePlane.html, [dostęp: 12.05.2025]
[4] Wikipedia contributors, Projective plane, Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Projective_plane&oldid=1287537678, [dostęp: 12.05.2025]

MathUp
Konferencja Zastosowań Matematyki

JAK NIE POPAŚĆ W RUINĘ? — CZYLI JAK MATEMATYKA POMAGA W MODELOWANIU WYPŁACALNOŚCI UBEZPIECZYCIELI.

Jakub ŁOMPIEŚ
Politechnika Łódzka, Łódź

Wstęp

Dyrektywa Wypłacalność II (z ang. Solvency II), która weszła w życie w krajach Unii Europejskiej w 2016 roku, uwzględnia nowoczesne praktyki zarządzania ryzykiem w zakładach ubezpieczeń. Jej celem jest zapewnienie stabilności finansowej ubezpieczycieli, ochrony klientów oraz spójności prawnej we wszystkich krajach członkowskich.

Jednym z fundamentów tej dyrektywy jest modelowanie wypłacalności zakładów ubezpieczeniowych w taki sposób, aby prawdopodobieństwo wypłacalności w ciągu roku nie było mniejsze niż 99,5%. Wymagania Solvency II promują stosowanie klasycznych modeli matematycznych służących do oceny ryzyka, takich jak model Craméra–Lundberga czy model Sparre Andersena.

Na tle tych klasycznych podejść coraz większym zainteresowaniem cieszą się modele przełącznikowe, w których parametry procesu mogą zmieniać się w czasie w zależności od aktualnego stanu rzeczywistości – na przykład stanu normalnego, przejściowego czy ekstremalnego. Dzięki takiej modyfikacji możliwe jest różnicowanie wysokości składki ubezpieczeniowej oraz uwzględnianie zmiennych rozkładów szkód pod względem ich intensywności i częstotliwości. Pozwala to na dokładniejszą ocenę kapitału wymaganego do pokrycia strat oraz lepsze modelowanie ryzyka związanego ze zjawiskami ekstremalnymi. Z tego względu modele przełącznikowe stanowią cenne rozszerzenie klasycznych metod oceny ryzyka.

W dalszej części pracy, w sekcjach 2 i 3, wprowadzimy potrzebne oznaczenia oraz narzędzie umożliwiające modelowanie stanu rzeczywistości – jednorodny łańcuch Markowa. Następnie, w rozdziale 4, przedstawimy przełącznikowy model Sparre Andersena, a w sekcji 5 omówimy metodę służącą do wyznaczania kolejnych prawdopodobieństw niewypłacalności ubezpieczyciela w skończonym horyzoncie.

Oznaczenia

Na wstępie należy zaznaczyć, że niniejsza praca opiera się w głównej mierze na wynikach oraz metodologii przedstawionej w [1].

Niech $(\boldsymbol{\Omega}, \boldsymbol{\mathcal{F}}, \mathbb{P})$ będzie przestrzenią probabilistyczną, na której zdefiniowane są wszystkie zmienne losowe i ich rodziny występujące w tej pracy. Przyjmujemy, że $\mathbb{N} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots\}$ oraz $\mathbb{N}_{\mathbf{0}} = \{\mathbf{0}\} \cup \mathbb{N}$.

W przełącznikowym modelu Sparre Andersena będziemy stosować następujące oznaczenia:

- $u \geq 0$ – początkowa nadwyżka ubezpieczyciela,
- $U_n(u)$ – nadwyżka po n-tym okresie, gdy początkowo wynosiła u,

- T_k – czas oczekiwania na k-tą szkodę; może być losowy (model ciągły) lub deterministyczny, tzn. $T_k = 1$ dla każdego k (model dyskretny),
- X_k – wielkość k-tej szkody (model ciągły) lub całkowita wielkość zniszczeń do końca k-tego okresu rozliczeniowego (model dyskretny),
- I_k – stan rzeczywistości w momencie, gdy wystąpiła k-ta szkoda.

Ponadto zakładamy, że zmienne losowe X_k i T_k są dodatnie z prawdopodobieństwem 1 dla każdego $k \in \mathbb{N}_0$.

Łańcuchy Markowa

Do modelowania zmian stanu rzeczywistości będziemy korzystać z jednorodnego łańcucha Markowa $(I_k)_{k\in\mathbb{N}_0}$ o skończonej przestrzeni stanów $\mathcal{S} = \{1,2,3,\dots,s\}$, $s \in \mathbb{N}$.

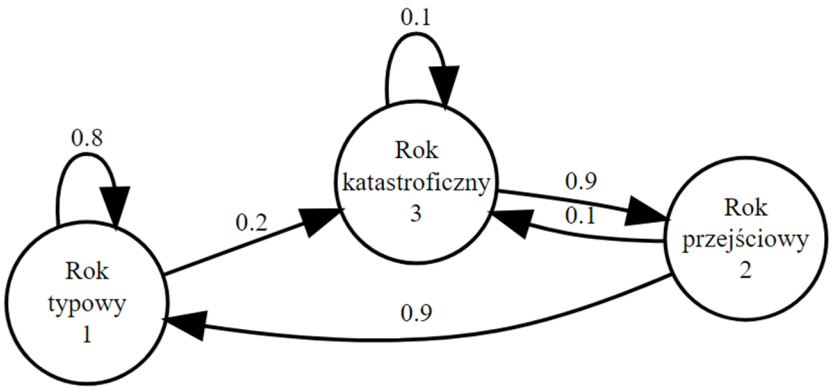
Definicja 1. Jeżeli prawdopodobieństwa warunkowe

$$\boldsymbol{p_{ij} = \mathbb{P}(I_{k+1} = j \mid I_k = i) \quad i,j \in \mathcal{S}}$$

nie zależą od $k \in \mathbb{N}_0$, to proces $(I_k)_{k\in\mathbb{N}_0}$ będziemy nazywać *jednorodnym łańcuchem Markowa* o przestrzeni stanów $\mathcal{S}$ i macierzy przejść

$$\boldsymbol{P = [p_{ij}]_{i,j\in\mathcal{S}}}.$$

Przykład 1. Załóżmy, że modelujemy ubezpieczenia osiedla zlokalizowanego na terenach zalewowych w dolinie pewnej rzeki. Z danych historycznych wiemy, że powódź występuje średnio raz na 5 lat, powodzie rok po roku zdarzają się średnio raz na 10 lat oraz powrót powodzi po roku przejściowym występuje średnio raz na 10 lat. Jest to adekwatna sytuacja na wykorzystanie łańcuchów Markowa. Przechodzimy teraz do konstrukcji jednorodnego łańcucha Markowa o trzech stanach, gdzie:



Rys. 1 Graf przedstawiający zaproponowany łańcuch Markowa o macierzy przejścia P.
Brak krawędzi pomiędzy wierzchołkami oznacza, że prawdopodobieństwo przejścia pomiędzy odpowiadającymi stanami jest równe 0.

- stan 1 reprezentuje rok typowy,
- stan 2 odpowiada za rok przejściowy (bezpośrednio po powodzi),
- stan 3 oznacza rok katastroficzny (czyli taki, w którym wystąpiła powódź).

Ponadto zakładamy następujące zasady przejść między stanami:

- po roku typowym nie może wystąpić rok przejściowy,
- po roku katastroficznym nie może nastąpić bezpośrednio rok typowy,
- rok przejściowy może przejść albo w rok typowy, albo znów w katastroficzny.

Na podstawie powyższych informacji możemy skonstruować macierz przejść

$$\boldsymbol{P} = \begin{bmatrix} \mathbf{0.8} & \mathbf{0} & \mathbf{0.2} \\ \mathbf{0.9} & \mathbf{0} & \mathbf{0.1} \\ \mathbf{0} & \mathbf{0.9} & \mathbf{0.1} \end{bmatrix}.$$

Co więcej, tak opisany łańcuch Markowa można przedstawić za pomocą grafu widocznego na Rys. 1.

Przełącznikowy model Sparre Andersena

Przy wcześniej wprowadzonych oznaczeniach oraz zakładając, że mechanizm przełącznikowy opisany jest za pomocą jednorodnego łańcucha Markowa, przechodzimy do definicji przełącznikowego modelu Sparre Andersena.

Definicja 2. Model opisany za pomocą procesu $(\boldsymbol{X_k}, \boldsymbol{T_k}, \boldsymbol{I_k})_{\boldsymbol{k}\in\mathbb{N}_0}$ będziemy nazywać *przełącznikowym modelem Sparre Andersena*. Wówczas proces ryzyka ma postać

$$\boldsymbol{U_n(u)} = \boldsymbol{u} - \sum_{\boldsymbol{k=1}}^{\boldsymbol{n}} (\boldsymbol{X_k} - \boldsymbol{c(I_{k-1})T_k}), \quad \boldsymbol{n} \in \mathbb{N}, \boldsymbol{u} \geq \mathbf{0},$$

gdzie $c(I_{k-1})$ oznacza wielkość składki w k-tym okresie, gdy łańcuch Markowa w poprzednim okresie znajdował się w stanie I_{k-1}.

Zwróćmy uwagę, że powyższa definicja modelu umożliwia na zróżnicowanie wysokości pobieranej składki oraz zmianę rozkładów szkód w zależności od stanu rzeczywistości przełącznika.

Ponadto konstrukcja ta pozwala na płynne przechodzenie pomiędzy modelem ciągłym (ubezpieczenia krótkoterminowe, np. turystyczne), a dyskretnym (ubezpieczenia długoterminowe, np. nieruchomości), w zależności od potrzeb. Przyjmując, że $\boldsymbol{\gamma}$ jest stałą składką pobieraną w ustalonej jednostce czasu (czyli $\boldsymbol{c(I_{k-1})} = \boldsymbol{\gamma}$ oraz $\boldsymbol{T_k} = \mathbf{1}$ dla każdego $\boldsymbol{k}$) uzyskujemy model dyskretny

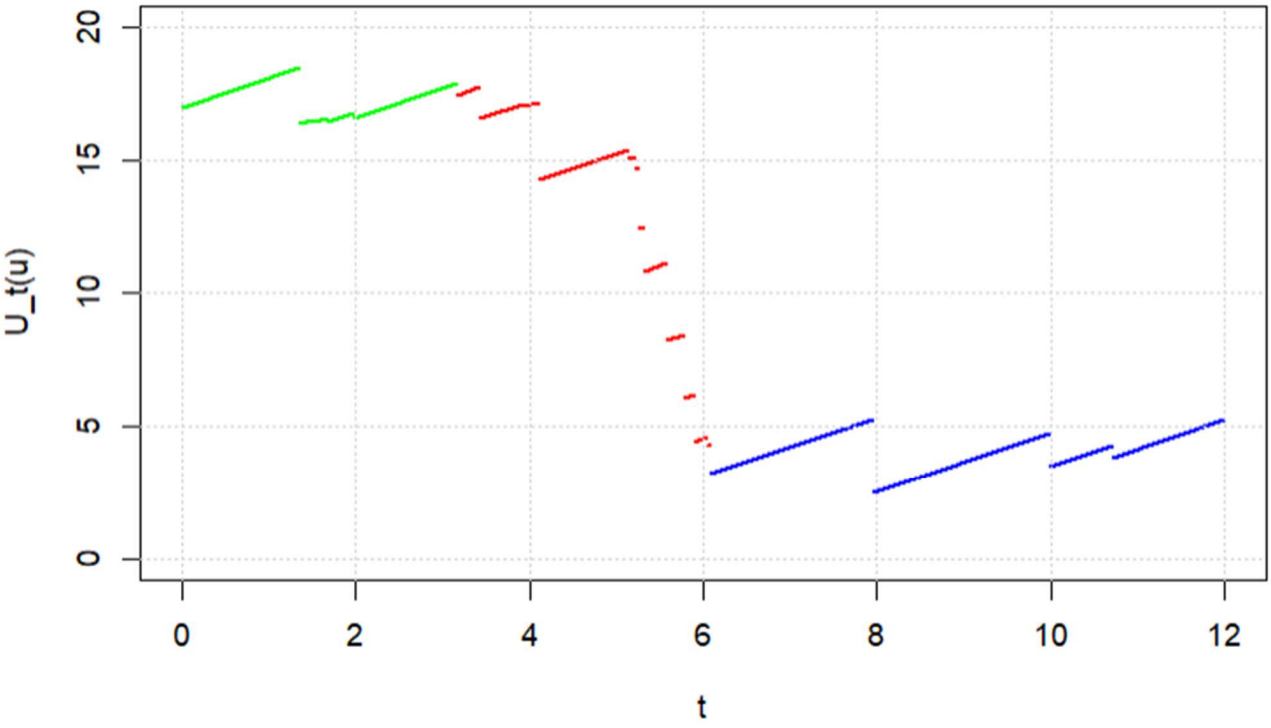
$$\boldsymbol{U_n(u)} = \boldsymbol{u} + \boldsymbol{\gamma n} - \sum_{\boldsymbol{k=1}}^{\boldsymbol{n}} \boldsymbol{X_k}.$$

Natomiast gdy T_k ma rozkład wykładniczy, przechodzimy do modelu ciągłego

$$\boldsymbol{U_t(u)} = \boldsymbol{u} + \boldsymbol{\gamma t} - \sum_{\boldsymbol{k=1}}^{\boldsymbol{N(t)}} \boldsymbol{X_k},$$

gdzie $(N(t))_{t\geq 0}$ to proces zliczający szkody, tzw. proces Poissona [1, Rozdział 2.3].

Zauważmy, że oba szczególne przypadki modelu przełącznikowego wciąż pozwalają na różnicowanie rozkładów strat. Na Rys. 2 przedstawiono wpływ, jaki może mieć zmiana stanu rzeczywistości przełącznika na wysokość oraz częstotliwość kolejnych szkód.



Rys. 2 Wykres przykładowej nadwyżki $U_t(u)$ w ciągłym modelu ryzyka,
w stanach: typowym, katastroficznym i przejściowym.

Moment niewypłacalności ubezpieczyciela lub inaczej moment ruiny definiujemy jako

$$\boldsymbol{\tau(u) = \inf\{n \in \mathbb{N}: U_n(u) < 0\}, \quad u \geq 0.}$$

Wartość $\tau(u)$ oznacza pierwszy moment n, w którym nadwyżka $U_n(u)$ spada poniżej zera. Jest to numer pierwszej szkody (w modelu ciągłym) lub numer pierwszego okresu rozliczeniowego (w modelu dyskretnym), w którym dochodzi do ruiny. Właśnie wtedy, po raz pierwszy od rozpoczęcia działalności, skumulowane szkody przekraczają kapitał początkowy ubezpieczyciela powiększony o zgromadzone składki.

Moment ruiny pozwala zdefiniować prawdopodobieństwo niewypłacalności ubezpieczyciela, czyli tzw. prawdopodobieństwo ruiny, zarówno w skończonym, jak i nieskończonym horyzoncie czasowym.

Definicja 3. Prawdopodobieństwo warunkowe zdarzenia $\boldsymbol{\tau(u) \leq n}$, przy warunku, że w chwili początkowej łańcuch Markowa był w stanie $\boldsymbol{i}$, nazywamy *prawdopodobieństwem ruiny w horyzoncie* $\boldsymbol{n}$ i oznaczamy $\boldsymbol{\Psi_n^i(u)}$, przy konwencji $\boldsymbol{\Psi_0^i(u) = 0}$, $\boldsymbol{i \in S}$.

Definicja 4. Prawdopodobieństwo warunkowe zdarzenia $\boldsymbol{\tau(u) < \infty}$, przy warunku, że w chwili początkowej łańcuch Markowa był w stanie $\boldsymbol{i}$, nazywamy *prawdopodobieństwem ruiny w nieskończonym horyzoncie* i oznaczamy $\boldsymbol{\Psi^i(u)}$, $\boldsymbol{i \in S}$.

Zwróćmy uwagę, że powyższe prawdopodobieństwa tworzą $\boldsymbol{s}$-wymiarowe wektory $\boldsymbol{\Psi_n(u) = (\Psi_n^1(u), \ldots, \Psi_n^s(u))}$ oraz $\boldsymbol{\Psi(u) = (\Psi^1(u), \ldots, \Psi^s(u))}$, gdzie $\boldsymbol{s}$ jest liczbą wszystkich stanów przełącznika.

W tym miejscu można postawić pytanie: dlaczego w prawdopodobieństwie ruiny uwzględniamy stan łańcucha Markowa w chwili początkowej? Aby na nie odpowiedzieć, będziemy kontynuować rozważania dotyczące problemu modelowania ubezpieczeń osiedla zlokalizowanego na terenach zalewowych pewnej rzeki.

Przykład 2. W Przykładzie 1 skonstruowaliśmy przełącznik $(\boldsymbol{I_k})_{\boldsymbol{k\in\mathbb{N}_0}}$ o trzech stanach: typowym, przejściowym i katastroficznym, wraz z odpowiadającą mu macierzą przejść

$$\boldsymbol{P}=\begin{bmatrix}\mathbf{0.8} & \mathbf{0} & \mathbf{0.2}\\ \mathbf{0.9} & \mathbf{0} & \mathbf{0.1}\\ \mathbf{0} & \mathbf{0.9} & \mathbf{0.1}\end{bmatrix}.$$

Ponieważ przedmiotem ubezpieczenia są nieruchomości, skorzystamy z modelu dyskretnego. Z danych historycznych wynika, że szkody mają następujące dystrybuanty warunkowe (gdy $p_{ij}=0$, dystrybuanta nie jest obserwowalna):

- $F^{11}(x)=F^{13}(x)=1-e^{-x}$,
- $F^{21}(x)=F^{23}(x)=1-e^{-0.5x}$,
- $F^{32}(x)=F^{33}(x)=1-e^{-0.2x}$,

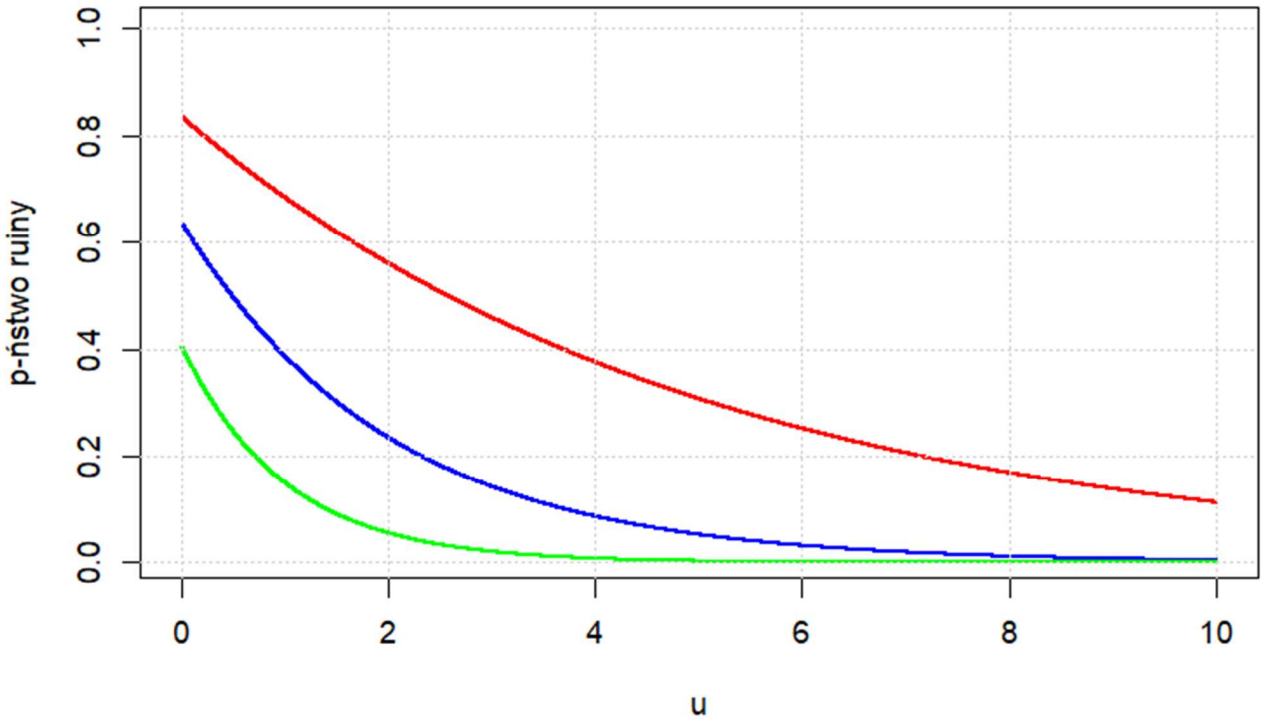
dla $x>0$. Ponadto wysokość pobieranej składki pod koniec jednego okresu rozliczeniowego wynosi $\gamma=0.9$. Sprawdźmy, jak kształtują się prawdopodobieństwa ruiny na najbliży okres sprawozdawczy. W tym celu musimy wyznaczyć Ψ_1. Przechodzimy do obliczeń:

$$\begin{aligned}
&\boldsymbol{\Psi_1^1(u)=\mathbb{P}(\tau(u)\le 1\mid I_0=1)=\mathbb{P}(X_1>u+0.9\mid I_0=1)}\\
&=\frac{\mathbb{P}(\boldsymbol{X_1>u+0.9, I_0=1})}{\mathbb{P}(\boldsymbol{I_0=1})}\\
&=\sum_{\boldsymbol{j=1}}^{\boldsymbol{3}}\frac{\mathbb{P}(\boldsymbol{X_1>u+0.9, I_0=1, I_1=j})}{\mathbb{P}(\boldsymbol{I_0=1})}\\
&=\sum_{\boldsymbol{j=1}}^{\boldsymbol{3}}\frac{\mathbb{P}(\boldsymbol{I_0=1, I_1=j})}{\mathbb{P}(\boldsymbol{I_0=1})}\cdot\frac{\mathbb{P}(\boldsymbol{X_1>u+0.9, I_0=1, I_1=j})}{\mathbb{P}(\boldsymbol{I_0=1, I_1=j})}\\
&=\sum_{\boldsymbol{j=1}}^{\boldsymbol{3}}\boldsymbol{p_{1j}\,\mathbb{P}(X_1>u+0.9\mid I_0=1, I_1=j)}\\
&=\mathbf{0.8}\left(\mathbf{1}-\boldsymbol{F^{11}(u+0.9)}\right)+\mathbf{0.2}\left(\mathbf{1}-\boldsymbol{F^{13}(u+0.9)}\right)=\boldsymbol{e^{-(u+0.9)}}.
\end{aligned}$$

Zwróćmy uwagę, że wszystkie wcześniej wspomniane dystrybuanty pochodzą z rozkładu wykładniczego, tylko z różnymi parametrami. Zatem, analogicznie otrzymujemy

$$\boldsymbol{\Psi_1^2(u)=e^{-0.5(u+0.9)}}\quad\text{oraz}\quad\boldsymbol{\Psi_1^3(u)=e^{-0.2(u+0.9)}}.$$

Jak można zauważyć (patrz Rys. 3) prawdopodobieństwa ruiny różnią się w zależności od stanu przełącznika. Wynika to z faktu, że w roku katastroficznym wysokość poniesionych strat jest znacznie większa niż np. w roku typowym. Dlatego też w modelowaniu tego typu ubezpieczeń należy uwzględniać wszystkie możliwe scenariusze.



Rys. 3 Wykres wyznaczonych prawdopodobieństw ruiny do końca pierwszego okresu sprawozdawczego Ψ_1^1, Ψ_1^2 i Ψ_1^3.

Wyznaczanie prawdopodobieństw ruiny w skończonym horyzoncie

W poprzednim przykładzie udało się wyznaczyć prawdopodobieństwo niewypłacalności ubezpieczyciela do końca pierwszego okresu rozliczeniowego na podstawie definicji oraz podstawowych własności rachunku prawdopodobieństwa. Niestety, analogiczne wyznaczanie kolejnych prawdopodobieństw w ten sam sposób byłoby bardzo pracochłonne. Okazuje się jednak, że istnieje bardziej efektywna metoda.

W niniejszej sekcji przedstawimy podejście umożliwiające wyznaczanie kolejnych prawdopodobieństw ruiny w skończonym horyzoncie czasowym.

W tym celu wprowadzimy operator, który odegra kluczową rolę w dalszych obliczeniach. Przez $\boldsymbol{F^{ij}}$ (odpowiednio $\boldsymbol{G^{ij}}$) będziemy oznaczać prawostronnie ciągłą dystrybuantę warunkową zmiennej losowej $\boldsymbol{X_1}$ (odpowiednio $\boldsymbol{T_1}$) pod warunkiem, że w chwili początkowej łańcuch Markowa był w stanie $\boldsymbol{i}$, a w momencie wystąpienia pierwszej szkody w stanie $\boldsymbol{j}$. Dystrybuantę warunkową dla wektora losowego $(\boldsymbol{X_1}, \boldsymbol{T_1})$, pod warunkiem analogicznym jak w przypadku zmiennej losowej $\boldsymbol{X_1}$, będziemy oznaczać przez $\boldsymbol{H^{ij}}$. Ponadto składka do momentu wystąpienia pierwszej szkody, pod warunkiem, że łańcuch Markowa w momencie początkowym był w stanie $\boldsymbol{i}$ wynosi $\boldsymbol{c(i)}$. Niech $\boldsymbol{\mathcal{R}}$ będzie zbiorem funkcji mierzalnych $\boldsymbol{\rho}: [\boldsymbol{0}, \infty) \to [\boldsymbol{0}, \boldsymbol{1}]$. Wtedy $\boldsymbol{\mathcal{R}^s} = \{\boldsymbol{\rho} = (\boldsymbol{\rho_1}, \ldots, \boldsymbol{\rho_s}): \boldsymbol{\rho_i} \in \boldsymbol{\mathcal{R}}\}$.

Definicja 5. Operator $\boldsymbol{L}: \boldsymbol{\mathcal{R}^s} \to \boldsymbol{\mathcal{R}^s}$ nazywamy *operatorem ryzyka*, jeżeli $\boldsymbol{L\rho(u)} = (\boldsymbol{L_1\rho(u)}, \ldots, \boldsymbol{L_s\rho(u)})$, gdzie

$$\boldsymbol{L_i\rho(u)} = \sum_{j=1}^{s} \boldsymbol{p_{ij}} \int_0^\infty \int_0^{u+c(i)t} \boldsymbol{\rho_j}(\boldsymbol{u} + \boldsymbol{c(i)t} - \boldsymbol{x}) \mathbf{d}\boldsymbol{F^{ij}(x)} \mathbf{d}\boldsymbol{G^{ij}(t)} +$$
$$\sum_{j=1}^{s} \boldsymbol{p_{ij}} \int_0^\infty \int_{u+c(i)t}^\infty \mathbf{d}\, \boldsymbol{F^{ij}(x)} \mathbf{d}\boldsymbol{G^{ij}(t)}, \quad \boldsymbol{i} \in \boldsymbol{S},$$

dla dowolnych $\rho \in \mathcal{R}^s$ i $u \geq 0$.

Uwaga. W Definicji 5 przyjmujemy następujące oznaczenia: $\int_0^{u+c(i)t} = \int_{[0,u+c(i)t]}$ oraz $\int_{u+c(i)t}^{\infty} = \int_{(u+c(i)t,\infty)}$. Są one szczególnie istotne w przypadku, gdy rozkłady wielkości szkód są rozkładami typu dyskretnego lub mieszanego.

Operator ryzyka $\boldsymbol{L}$ jest monotoniczny [1, Lemat 2.47], przekształca funkcje nierosnące w funkcje nierosnące [1, Lemat 2.46] oraz przy odpowiednich założeniach jest kontrakcją w pewnej zupełnej przestrzeni metrycznej [1, Twierdzenie 2.54].

Przechodzimy teraz do kluczowej części tego rozdziału. Zaprezentujemy twierdzenie, które daje nam możliwość wyznaczenia prawdopodobieństw w skończonym horyzoncie poprzez kolejne iteracje operatora $\boldsymbol{L}$. Zdefiniujmy zmienne $\boldsymbol{Z_k} = \boldsymbol{X_k} - \boldsymbol{c(I_{k-1})T_k}$, $\boldsymbol{k} \in \mathbb{N}$, które oznaczają wkład $\boldsymbol{k}$-tej szkody do procesu ryzyka $(\boldsymbol{U_n(u)})_{\boldsymbol{n}\in\mathbb{N}}$, $\boldsymbol{u} \geq \boldsymbol{0}$.

Twierdzenie 1. Jeśli dla dowolnych $\boldsymbol{i,j} \in \boldsymbol{S}$, $\boldsymbol{k} \in \mathbb{N}\backslash\{\boldsymbol{1}\}$, $\boldsymbol{t,x} > \boldsymbol{0}$:

1. warunkowy rozkład zmiennych $Z_2, \ldots, Z_k$ przy warunku $I_0 = i$, $I_1 = j$, $T_1 = t$ i $X_1 = x$ jest taki sam, jak rozkład zmiennych $Z_1, \ldots, Z_{k-1}$ przy warunku $I_0 = j$,
2. $H^{ij}(x,t) = F^{ij}(x)G^{ij}(t)$,

to

$$\boldsymbol{\Psi_n(u)} = \boldsymbol{L^n\Psi_0(u)}, \quad \boldsymbol{n} \in \mathbb{N}, \boldsymbol{u} \geq \boldsymbol{0}.$$

Uwaga. Zapis $\boldsymbol{L^n}$ interpretujemy jako $\boldsymbol{n}$-krotne złożenie operatora $\boldsymbol{L}$.

Na zakończenie, ponownie wracamy do problemu ubezpieczeń nieruchomości zlokalizowanych na terenach zalewowych. Tym razem celem jest wyznaczenie prawdopodobieństwa niewypłacalności na dwa najbliższe okresy sprawozdawcze.

Przykład 3. Przypomnijmy, że skonstruowaliśmy przełącznik o trzech stanach: typowym, przejściowym i katastroficznym, wraz z odpowiadającą mu macierzą przejść $\boldsymbol{P}$.

W przypadku ubezpieczania nieruchomości stosujemy model dyskretny, w którym składka pobierana na koniec jednego okresu rozliczeniowego wynosi $\gamma = 0.9$, a warunkowe dystrybuanty szkód mają następującą postać (dla $x > 0$):

- $F^{11}(x) = F^{13}(x) = 1 - e^{-x}$,
- $F^{21}(x) = F^{23}(x) = 1 - e^{-0.5x}$,
- $F^{32}(x) = F^{33}(x) = 1 - e^{-0.2x}$.

Prawdopodobieństwo niewypłacalności do końca pierwszego okresu sprawozdawczego zostało już wyznaczone i wynosi

$$\boldsymbol{\Psi_1(u) = (e^{-(u+0.9)}, e^{-0.5(u+0.9)}, e^{-0.2(u+0.9)}), \quad u \geq 0.}$$

Naszym celem jest teraz wyznaczenie Ψ_2. Zauważmy, że w modelu dyskretnym mamy $T_k = 1$ dla każdego k, a więc $G^{ij}(t) = \mathbb{1}_{[1,\infty)}(t)$ dla wszystkich $i, j \in \{1,2,3\}$. Zatem poszczególne współrzędne operatora ryzyka w modelu dyskretnym możemy zapisać jako

$$\boldsymbol{L_i\rho(u) = \sum_{j=1}^{3} p_{ij}\left(\int_0^{u+0.9} \rho_j(u+0.9-x)\mathrm{d}F^{ij}(x) + \int_{u+0.9}^{\infty} \mathrm{d}\,F^{ij}(x)\right), \quad i \in \{1,2,3\},}$$

dla dowolnych $\rho \in \mathcal{R}^s$ i $u \geq 0$. Wówczas z Twierdzenia 1 mamy

$$\boldsymbol{\Psi_2(u) = L^2\Psi_0(u) = L(L\Psi_0(u)) = L\Psi_1(u).}$$

Przystępujemy teraz do wyznaczenia Ψ_2^1.

$$\begin{aligned}
\boldsymbol{\Psi_2^1(u) = L_1\Psi_1(u)} &= \boldsymbol{\sum_{j=1}^{3} p_{1j}\left(\int_0^{u+0.9} \Psi_1^j(u+0.9-x)\mathrm{d}F^{1j}(x) + \int_{u+0.9}^{\infty} \mathrm{d}\,F^{1j}(x)\right)} \\
&= \boldsymbol{0.8\left(\int_0^{u+0.9} e^{-(u+1.8-x)}\, e^{-x}\mathrm{d}x + 1 - F^{11}(u+0.9)\right) +} \\
&\boldsymbol{0.2\left(\int_0^{u+0.9} e^{-0.2(u+1.8-x)}\, e^{-x}\mathrm{d}x + 1 - F^{13}(u+0.9)\right) =} \\
&\boldsymbol{0.8e^{-(u+1.8)}\int_0^{u+0.9} \mathrm{d}\,x + 0.2e^{-0.2(u+1.8)}\int_0^{u+0.9} e^{-0.8x}\,\mathrm{d}x + e^{-(u+0.9)} =} \\
&= \boldsymbol{0.8e^{-(u+1.8)}(u+0.9) + 0.25e^{-0.2(u+1.8)}(1 - e^{-0.8(u+0.9)}) + e^{-(u+0.9)}.}
\end{aligned}$$

Analogicznie otrzymujemy

$$\boldsymbol{\Psi_2^2(u) = 0.9e^{-(u+1.8)}(e^{0.5(u+0.9)} - 1) + 6^{-1}e^{-0.2(u+1.8)}(1 - e^{-0.3(u+0.9)}) + e^{-0.5(u+0.9)}}$$

oraz

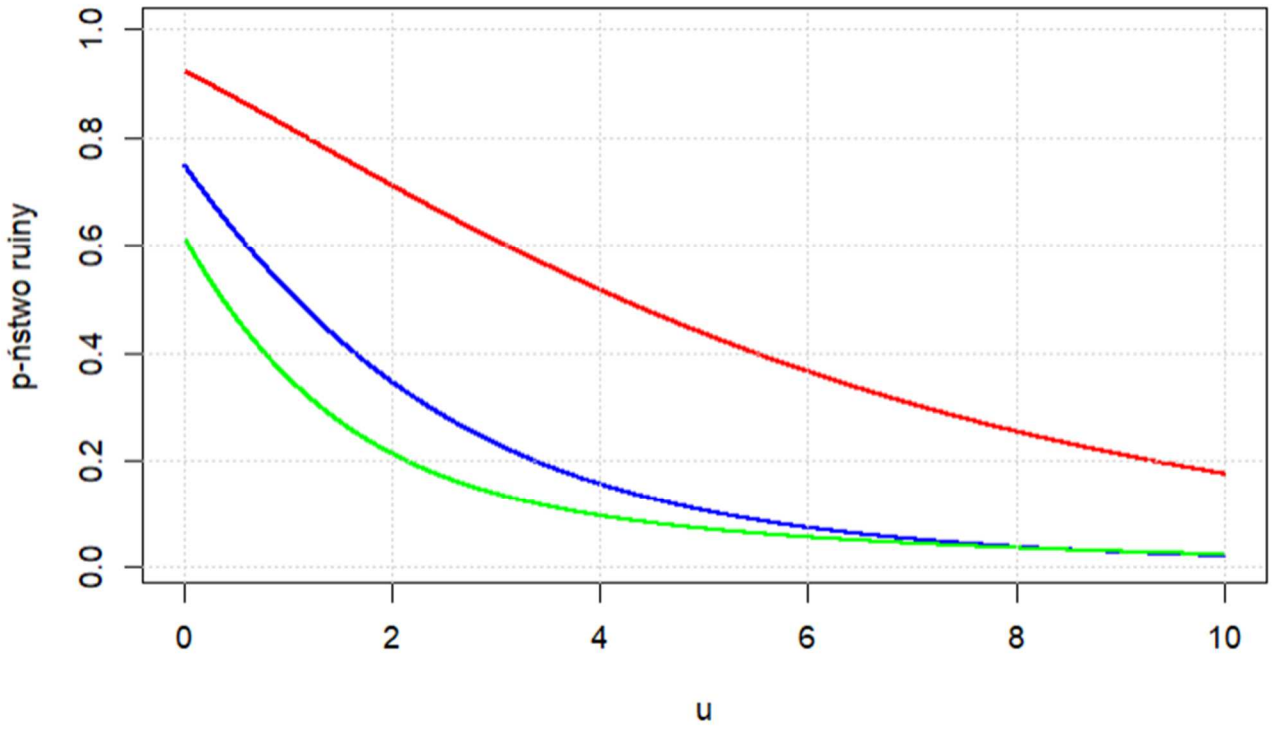
$$\boldsymbol{\Psi_2^3(u) = 0.6e^{-0.5(u+1.8)}(e^{0.3(u+0.9)} - 1) + 0.02e^{-0.2(u+1.8)}(u+0.9) + e^{-0.2(u+0.9)}.}$$

Gdybyśmy chcieli dobrać kapitał początkowy $\boldsymbol{u}$ w taki sposób, aby prawdopodobieństwo niewypłacalności w ciągu dwóch lat nie przekraczało $\boldsymbol{0.5\%}$, należałoby rozwiązać nierówność

$$\boldsymbol{\max\{\Psi_2^1(u), \Psi_2^2(u), \Psi_2^3(u)\} \leq 0.005.}$$

Zastosowanie funkcji maksimum zapewnia, że każde z tych prawdopodobieństw spełnia wymagania regulacyjne, niezależnie od początkowego stanu przełącznika — czy mamy rok powodzi, po powodzi, czy bez powodzi.

Ze względu na nieliniowy charakter powyższej nierówności, do jej rozwiązania można zastosować np. metodę bisekcji. W rezultacie otrzymujemy przybliżone rozwiązanie w postaci $\boldsymbol{u \geq 28.75}$. Zatem wystarczy przyjąć kapitał początkowy w wysokości $\boldsymbol{u = 28.75}$, aby mieć pewność, że prawdopodobieństwo niewypłacalności w ciągu dwóch lat nie przekroczy $\boldsymbol{0.5\%}$.



Rys. 4 Wykres wyznaczonych prawdopodobieństw ruiny do końca drugiego okresu sprawozdawczego Ψ_2^1, Ψ_2^2 i Ψ_2^3 dla analizowanego przykładu.

Literatura

[1] Gajek L., Podstawy Ubezpieczeń Majątkowych. Modele i Metody Aktuarialne, PWN, Warszawa, 2022

MathUp
Konferencja Zastosowań Matematyki

LICZBY NA TROPIE – GDY MATEMATYKA STAJE SIĘ DETEKTYWEM

Monika LOLO, Kinga KUJAWIAK
Politechnika Łódzka, Łódź

Wstęp

Współczesna matematyka znajduje coraz więcej zastosowań, nie tylko w naukach ścisłych, technice czy ekonomii, ale również w obszarach, które jeszcze niedawno wydawały się całkowicie zdominowane przez doświadczenie i intuicję. Jednym z takich zaskakujących obszarów zastosowań jest kryminalistyka, w której narzędzia matematyczne stają się wsparciem w analizach śledczych. Dzięki narzędziom możliwe jest nie tylko tworzenie geograficznych profili sprawców przestępstw, ale także prognozowanie, gdzie i kiedy może dojść do kolejnych ataków.
Matematyczne podejście do analizy przestępczości skupia się głównie na wykorzystaniu danych oraz ich interpretacji za pomocą odpowiednio dobranych modeli. Zastosowanie matematyki w badaniach nad przestępczością opiera się na modelach, które pozwalają analizować zachowania seryjnych sprawców na podstawie dostępnych danych. Modele takie jak formuła Rossmo umożliwiają określenie najbardziej prawdopodobnego miejsca zamieszkania przestępcy na podstawie lokalizacji dokonanych przez niego zbrodni, natomiast model GM (1,1), pozwala prognozować kolejne ataki w czasie. Połączenie tych modeli z metodami tradycyjnymi znacząco zwiększa szanse skutecznego śledztwa, a jednocześnie pokazuje, że matematyka może być narzędziem o dużym potencjale praktycznym również w walce z przestępczością.

Model Rossmo – geograficzne profilowanie przestępcy

Formuła Rossmo to model geograficznego profilowania przestępców, który pozwala przewidywać prawdopodobne miejsce zamieszkania sprawcy na podstawie lokalizacji jego przestępstw. Model ten opracował w 1996 r. kanadyjski kryminolog, były policjant Kim Rossmo. Jego głównym założeniem jest to, że seryjni przestępcy nie działają losowo a ich wybór miejsca zbrodni podlega konkretnym schematom i ograniczeniom. Unikają terenów zbyt bliskich swojemu miejscu zamieszkania, aby nie przyciągać uwagi, ale jednocześnie nie oddalają się zbyt daleko, by zminimalizować ryzyko i nakład logistyczny. Geograficzne profilowanie według Rossmo znajduje zastosowanie przede wszystkim w sprawach dotyczących przestępstw seryjnych, takich jak morderstwa czy włamania, pomagając organom ścigania zawęzić obszar poszukiwań podejrzanego.

Główne założenia modelu Rossmo

- Model uwzględnia schematy działania sprawców, przypisując różnym punktom na mapie prawdopodobieństwo tego, że sprawca tam mieszka lub przebywa.
- Sprawca nie popełnia przestępstw blisko swojego miejsca zamieszkania, aby nie wzbudzać podejrzeń.

- Wzrost odległości od miejsca zamieszkania zmniejsza prawdopodobieństwo wyboru lokalizacji przestępstwa.
- Seryjny morderca popełni kolejną zbrodnię w czasie możliwie najdalszym od poprzednich.

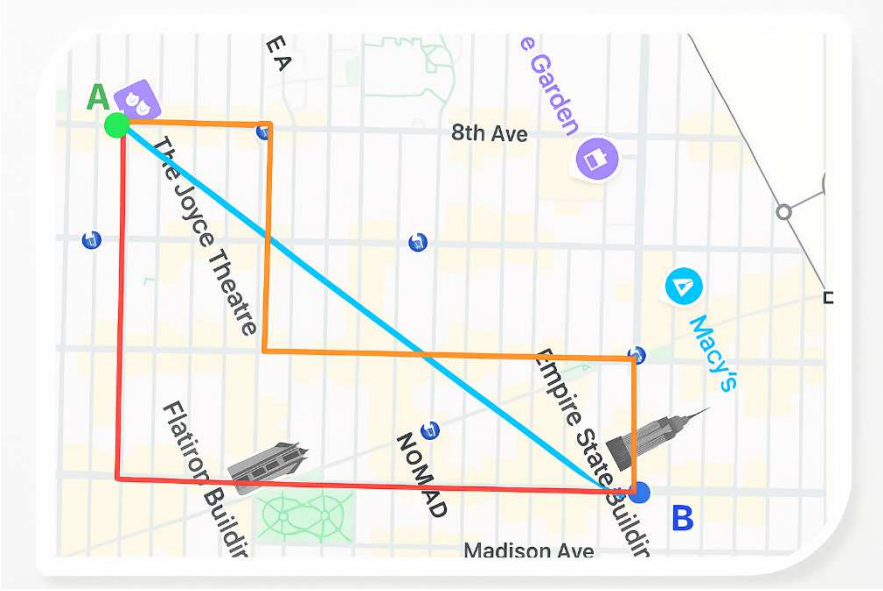
Miasto jako siatka – wybór metryki

W analizie geograficznej ważne jest przyjęcie odpowiedniej miary odległości. W środowisku miejskim, gdzie ulice tworzą prostokątne siatki jak np. na Manhattanie, bardziej trafna niż klasyczna odległość euklidesowa wyrażona jako odległość odcinka łączącego punkty, okazuje się metryka taksówkowa, zwana także Manhattan. Na obszarach miejskich, w których działa większość seryjnych morderców, budynki są rozmieszczone na planie prostokątów, a proste ulice przecinają się pod kątami prostymi. W mieście takim możemy poruszać się jedynie w kierunkach wschód-zachód oraz północ-południe.

Odległość punktów $A = (x_a, y_a)$ i $B = (x_b, y_b)$ w metryce Manhatan wyraża sie wzorem

$$d(A,B) = |x_b - x_a| + |y_b - y_a| \, .$$

Kolorami: pomarańczowym i czerwonym zaznaczone są różne drogi z punktu A do B, a sumy długosci odcinków je tworzących to odległość d(A, B) w metryce Manhattan.
Dla porównania przedstawiona jest droga niebieska (niemożliwa do przejścia w mieście), a długość niebieskiego odcinka jest odległością punktów A i B w metryce euklidesowej.



Rys. 1 Odległości między punktami A i B w metryce Manhattan i metryce euklidesowej.

Ogólna postać wzoru Rossmo

Wzór Rossmo to narzędzie, które pomaga policji ustalić, gdzie może mieszkać seryjny przestępca. Wzór ten przypisuje każdemu punktowi (X_i, Y_j) na mapie wartość $p(X_i, Y_j)$, która oznacza względne prawdopodobieństwo zamieszkania sprawcy w punkcie (X_i, Y_j). Indeksy i oraz j oznaczają numery wiersza i kolumny sektora na siatce mapy.

Wartości $p(X_i, Y_j)$ mają charakter względny, co oznacza, że nie przedstawiają rzeczywistego prawdopodobieństwa. Na przykład, $p(X_i, Y_j) = 25$ nie oznacza, że istnieje 25% szans na znalezienie sprawcy w tym miejscu. Prawdopodobieństwa te to jedynie wartości porównawcze, które pozwalają wskazać obszary o wyższym lub niższym prawdopodobieństwie w stosunku do innych punktów na mapie.

Wzór Rossmo można zapisać następująco:

$$p(X_i, Y_j) = \alpha \sum_{k=1}^{N} \left(\frac{\lambda}{\left(|X_i - x_k| + |Y_j - y_k|\right)^{\xi}} + \frac{(1-\lambda)\left(\beta^{\eta-\xi}\right)}{\left(2\beta - \left(|X_i - x_k| + |Y_j - y_k|\right)\right)^{\eta}} \right)$$

gdzie:

- (X_i, Y_j) – analizowany punkt (możliwe miejsce zamieszkania sprawcy),
- $p(X_i, Y_j)$ – prawdopodobieństwo, że sprawca mieszka w punkcie (X_i, Y_j),
- N – liczba miejsc przestępstw,
- (x_k, y_k) – współrzędne k-tego miejsca przestępstwa,
- λ – współczynnik określający, jak bardzo odległość wpływa na obniżenie prawdopodobieństwa w strefie buforowej (blisko miejsca zamieszkania sprawcy); w praktyce często dla uproszczenia przyjmuje się λ =1,
- ξ – wykładnik dla wpływu odległości (w strefie działania sprawcy); typowe wartości: 1,2 – 2,0; im większa wartość, tym gwałtownej maleje prawdopodobieństwo wraz ze wzrostem odległości od punktu przestępstwa (poza strefą buforową),
- β – parametr wyznaczający promień koła wokół prawdopodobnego miejsca zamieszkania sprawcy, w którym rzadko popełnia on przestępstwa; dobierany empirycznie do lonkretnego przypadku; typowe wartości: 1 – 2 km,
- η – wykładnik dla wpływu poza strefą; typowe wartości: 1,0 – 1,5; im wyższa η, tym szybciej maleje prawdopodobieństwo wraz ze wzrostem odległości punktu od lokalizacji
- α – stała normalizująca, dostosowuje sumę prawdopodobieństw tak, aby można było je porównywać względnie; nie ma stałej wartości, jej wybór zależy od liczby punktów przestępstw i ich rozmieszczenia.

Uwaga: W praktyce, wartości parametrów są dostosowywane do konkretnego przypadku w zależności od:

- typu przestępstw (np. włamania, zabójstwa),
- charakterystyki geograficznej terenu (miejski/wiejski),
- danych statystycznych i empirycznych pochodzących z danego regionu.

Wzór Rossmo składa się z dwóch głównych wyrażeń, które analizują różne strefy wokół miejsc przestępstw: poza strefą buforową i wewnątrz strefy buforowej.

W pierwszym wyrażeniu

$$\frac{\lambda}{\left(|X_i - x_k| + |Y_j - y_k|\right)^{\xi}}$$

najważniejszą rolę odgrywa mianownik, który odpowiada za analizę prawdopodobieństwa poza strefą buforową. Im dalej dany punkt (X_i,Y_j) znajduje się od miejsca przestępstwa, tym mniejsza wartość tego wyrażenia, a wraz z nią mniejsze prawdopodobieństwo, że sprawca mieszka w tej okolicy.

Drugie wyrażenie

$$\frac{(1-\lambda)\left(\beta^{\eta-\xi}\right)}{\left(2\beta-\left(|X_i-x_k|+|Y_j-y_k|\right)\right)^{\eta}}$$

analizuje obszar **blisko miejsca przestępstwa**, czyli **wewnątrz strefy buforowej**. Im dalej dany punkt znajduje się poza obszarem buforowym, to wartość w mianowniku rośnie, co powoduje, że cały ułamek maleje, czyli prawdopodobieństwo wyznaczone przez wzór Rossmo dla tego punktu staje się znacznie mniejsze.

Oba te wyrażenia pomagają zrównoważyć odległość między zamieszkaniem zbyt blisko lub zbyt daleko od miejsca przestępstwa.
Po dodaniu wszystkich składników powstaje mapa prawdopodobieństwa, która pozwala wyznaczyć tzw. *Hot Zone*, czyli obszar o najwyższym prawdopodobieństwie, w którym może znajdować się miejsce zamieszkania sprawcy.

Tworzenie profilu geograficznego – przykład

Załóżmy, że popełniono pięć przestępstw w różnych częściach miasta. Na podstawie dowodów okolicznościowych śledczy dochodzi do wniosku, że zostały one popełnione przez dokładnie tę samą osobę. Policja chce przewidzieć, gdzie sprawca może mieszkać.

Dla każdego miejsca przestępstwa, wyznaczono ze wzoru Rossmo wartości prawdopodobieństw odnośnie wszystkich punktów siatki analizy, które pokrywają badany obszar. Obliczenia uwzględniają zarówno pierwsze wyrażenie, opisujące wpływ odległości od miejsca przestępstwa, jak i drugie wyrażenie, eliminujące wpływ punktów znajdujących się zbyt blisko, czyli w tzw. strefie buforowej.

$\boldsymbol{k}$	$(\boldsymbol{x_k}, \boldsymbol{y_k})$
1	$(751, 147)$
2	$(399, 550)$
3	$(884, 251)$
4	$(118, 719)$
5	$(835, 710)$

Tab. 1 Lokalizacje miejsc popełnienia przestępstw.

Obliczono prawdopodobieństwo dla $\text{punktu } (X_i, Y_j) = (150, 400)$, przyjmując:

- $\lambda = 0{,}5$

- $\beta = 200$
- $\xi = 1{,}2$
- $\eta = 2$

Podstawiając powyższe parametry do wzoru Rossmo, otrzymano następującą wartość:

$$p(150{,}400) = \alpha \sum_{k=1}^{5} \left(\frac{0{,}5}{(|150 - x_k| + |400 - y_k|)^{1{,}2}} + \frac{(1 - 0{,}5)(200^{0{,}8})}{\left(2 \cdot 200 - (|150 - x_k| + |400 - y_k|)\right)^2} \right).$$

Następnie dla każdej z pięciu lokalizacji przestępstw $(\mathrm{k}=1,2,3,4,5)$ obliczono sumy dwóch wyrażeń, otrzymując kolejne składniki we wzorze równe:

- 0,000199, dla przestępstwa 1 o współrzędnych (751, 147),
- 0,00060, dla przestępstwa 2 o współrzędnych (399, 550),
- 0,00019, dla przestępstwa 3 o współrzędnych (884, 251),
- 0,00072, dla przestępstwa 4 o współrzędnych (118, 719),
- 0,00016, dla przestępstwa 5 o współrzędnych (835, 710).

Zatem

$$p(150{,}400) = \alpha(0{,}00019 + 0{,}00060 + 0{,}00019 + 0{,}00072 + 0{,}00016) = \alpha\ 0{,}00186.$$

W celu normalizacji wybrano $\alpha = 10000$.

Na podstawie obliczeń, prawdopodobieństwo, że sprawca mieszka w punkcie (150, 400) wynosi 18,60 jednostek. Należy przy tym pamiętać, że jest to wartość względna.

W kolejnym etapie analizy, dokonano porównania uzyskanej wartości prawdopodobieństwa z innymi punktami na mapie miasta. W ramach przeprowadzonego porównania wybrano punkty (100,100) oraz (950, 950). Wartości prawdopodobieństwa dla tych punktów zestawione są w tabeli 2.

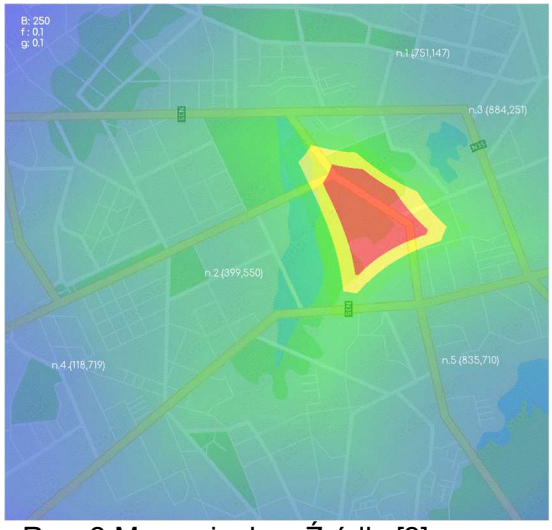
(X_i, Y_j)	$p(X_i, Y_j)$
$(150, 400)$	18,60
$(100, 100)$	10,90
$(950, 950)$	14,27

Tab. 2 Współrzędne punktów (X_i, Y_j) na mapie oraz odpowiadające im wartości prawdopodobieństwa wyznaczone na podstawie wzoru Rossmo.

Największe prawdopodobieństwo jest dla punktu o współrzędnych (150, 400), co oznacza, że sprawca najprawdopodobniej mieszka w tym rejonie.

Mapa cieplna jako narzędzie geograficznego profilowania

Jednym z kluczowych elementów praktycznego zastosowania modelu Rossmo w analizie śledczej jest możliwość przedstawienia wyników w formie wizualizacji przestrzennej, tzw. mapy cieplnej. Mapa ta ilustruje wartości funkcji $p(X_i, Y_j)$, przypisane kolejnym punktom siatki analizowanego obszaru, umożliwiając graficzne wskazanie obszarów o najwyższym ryzyku.
Po naniesieniu na mapę pięciu miejsc przestępstw i zastosowaniu modelu Rossmo, uzyskujemy mapę cieplną, która wskazuje prawdopodobne miejsca zamieszkania sprawcy. Kolory na mapie pokazują, gdzie warto prowadzić śledztwo, a które obszary są mniej istotne. Najbardziej intensywne barwy, w tym przypadku czerwone, wskazują miejsca, w których z największym prawdopodobieństwem może znajdować się sprawca. Jaśniejsze obszary oznaczają rejony o mniejszym znaczeniu. Takie przedstawienie danych pozwala skupić działania śledczych na najbardziej istotnych fragmentach miasta, zamiast prowadzić poszukiwania na jego całym obszarze.



Rys. 2 Mapa cieplna. Źródło [2].

Analiza mapy cieplnej umożliwia wskazanie obszaru o najwyższym prawdopodobieństwie, w którym może znajdować się sprawca.
W przedstawionym przykładzie, największe wartości prawdopodobieństwa uzyskano dla punktu o współrzędnych (150, 400). Wynik ten wskazuje, że to właśnie ten rejon powinien zostać objęty szczególną uwagą przez organy ścigania jako potencjalne miejsce zamieszkania sprawcy.

Teoria szarych systemów – Model GM(1,1)

Współczesna kryminalistyka coraz częściej korzysta z zaawansowanych narzędzi matematycznych w celu zwiększenia skuteczności działań operacyjnych. Jednym z takich narzędzi jest model GM(1,1), należący do tzw. teorii szarych systemów (Grey Systems Theory – GST), opracowanej przez profesora uniwersytetu Huazhong, Deng Julonga. Systemy te nazywane są „szarymi", ponieważ nasza wiedza o ich strukturze, granicach i interakcjach z otoczeniem jest ograniczona

i niepewna. Zgodnie z ogólną teorią systemów, rzadko mamy do czynienia z tzw. „białymi skrzynkami" – systemami, które są nam w pełni znane. Częściej spotykamy się z systemami szarymi, o których posiadamy jedynie częściową wiedzę, lub z systemami czarnymi, w których możemy obserwować jedynie wejścia i wyjścia, tj. dane, bodźce lub zasoby dostarczane do systemu (wejścia) oraz rezultaty jego działania w postaci generowanych odpowiedzi lub wyników (wyjścia), bez znajomości jego wewnętrznej struktury [3]. Można więc stwierdzić, że system ten jest szary – nie jest całkowicie nieznany, jak system czarny, ale też nie w pełni znany, jak system biały.
Zatem skrót GM oznacza „Grey Model", co odnosi się do charakteru analizowanego systemu – częściowo znanego, czyli szarego. Liczby w nawiasach określają strukturę modelu: pierwsza cyfra „1" wskazuje, że model dotyczy jednej zmiennej zależnej, natomiast druga jedynka oznacza, że model oparty jest na równaniu różniczkowym rzędu pierwszego. W efekcie GM(1,1) to jednowymiarowy model szary pierwszego rzędu, który pozwala na modelowanie i prognozowanie zachowania systemów przy ograniczonej ilości danych.

Zastosowanie modelu GM(1,1) w analizie kryminalnej

W przypadku przestępstw seryjnych często obserwuje się pewną regularność i cykliczność w odstępach czasowych między kolejnymi atakami. Zjawisko to wiąże się z psychologicznymi mechanizmami leżącymi u podstaw działań sprawcy, takimi jak narastająca potrzeba napięcia emocjonalnego, dążenie do jego rozładowania, a także subiektywne poczucie ulgi po dokonaniu czynu. Te wewnętrzne motywy prowadzą do powstawania powtarzalnych wzorców zachowań, które znajdują odzwierciedlenie w czasie i sposobie popełniania przestępstw.

Budowa modelu

Model GM(1,1) opiera się na założeniu, że przekształcony (skumulowany) szereg danych spełnia równanie różniczkowe pierwszego rzędu. Budowę modelu GM(1,1) rozpoczyna się od przekształcenia oryginalnych danych, co ma na celu ułatwienie modelowania dynamicznego trendu zjawiska.
Dla danego szeregu oryginalnego (sekwencji pierwotnej):

$$T^{(0)} = \{T^{(0)}(1), T^{(0)}(2), \dots, T^{(0)}(n)\}$$

tworzy się nowy szereg $T^{(1)}$ jako skumulowaną sekwencję przy użyciu operacji AGO (Accumulated Generating Operation). Polega ona na sukcesywnym sumowaniu kolejnych wartości z oryginalnego szeregu

$$T^{(1)}(k)\alpha = \sum_{k=1}^{N} T^{(0)}(i), \quad k = 1,2,\dots,n\,.$$

Następnie przyjmuje się, że skumulowany szereg $T^{(1)}$ spełnia równanie różniczkowe postaci

$$\frac{dT^{(1)}}{dt} + aT^{(1)} = b$$

gdzie:

a – współczynnik wzrostu/zaniku, określa tempo zmian badanej wielkości (np. liczby zdarzeń w czasie) oraz odzwierciedla, czy zjawisko ma tendencję do przyspieszania czy hamowania.

b – współczynnik wejściowy; wpływa na ogólny poziom modelowanej zmiennej, niezależnie od dynamiki sterowanej przez a.

Powyższe równanie opisuje dynamikę systemu w sposób ciągły i deterministyczny, przy czym jego rozwiązanie jest funkcją wykładniczą. Równanie różniczkowe przekształca się do postaci dyskretnej (sekwencji pośredniej) za pomocą średnich arytmetycznych

$$Z^{(1)}(k) = \frac{1}{2}\left(T^{(1)}(k-1) + T^{(1)}(k)\right), \quad k = 2, \dots, n \ ,$$

co prowadzi do liniowego modelu regresyjnego

$$T^{(0)}(k) + aZ^{(1)}(k) = b$$

lub inaczej

$$B \cdot \hat{a} = Y,$$

gdzie

$$Y = \begin{bmatrix} T^{(0)}(2) \\ T^{(0)}(3) \\ \vdots \\ T^{(0)}(n) \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{1}{2}\left(T^{(1)}(1) + T^{(1)}(2)\right) & 1 \\ -\frac{1}{2}\left(T^{(1)}(2) + T^{(1)}(3)\right) & 1 \\ \vdots & \\ -\frac{1}{2}\left(T^{(1)}(n-1) + T^{(1)}(n)\right) & 1 \end{bmatrix}, \quad \hat{a} = \begin{bmatrix} a \\ b \end{bmatrix}.$$

Współczynniki a i b wyznacza się metodą najmniejszych kwadratów

$$a = (B^{\mathrm{T}} \cdot B)^{-1} \cdot B^{\mathrm{T}} \cdot Y,$$

gdzie

B^{T} oznacza transpozycję macierzy B,

$(B^{\mathrm{T}} \cdot B)^{-1}$ oznacza macierz odwrotną iloczynu B^{T} i B.

Jeżeli spojrzeć na rozwiązanie w czasie ciągłym równania różniczkowego $\frac{dT^{(1)}}{dt} + aT^{(1)} = b$, to przyjmuje ono postać

$$T^{(1)}(\mathrm{t}) = Ce^{-at} + \frac{b}{a} \ .$$

Wartość stałej C wyznacza się z warunku początkowego. Przyjmując, że w chwili $t = 1$ rozwiązanie ma wartość $T^{(1)}(1)$, otrzymuje się

$$C = \left(T^{(1)}(1) - \frac{b}{a}\right) \cdot e^{a}.$$

Po podstawieniu tej wartości oraz zapisaniu czasu w kolejnych krokach całkowitych i wyznaczeniu parametrów modelu można zapisać jego rozwiązanie ogólne, które dla kolejnych wartości k przyjmuje postać

$$T^{(1)}(k+1) = \left(T^{(0)}(1) - \frac{b}{a}\right) \cdot e^{-a \cdot k} + \frac{b}{a}.$$

Analiza modelu na studium przypadku

Zakładamy, że znane są nam daty czterech kolejnych przestępstw dokonanych przez jednego sprawcę - każde oznaczone numerem porządkowym oraz odpowiadającym mu miesiącem (liczonym od stycznia 2020 roku), a naszym celem jest oszacowanie prawdopodobnego czasu wystąpienia kolejnego ataku. Dane zawiera poniższa tabela

Nr	Data przestępstwa	Miesiąc (w liczbach)
1	styczeń 2020	1
2	czerwiec 2020	6
3	grudzień 2020	12
4	czerwiec 2021	18

Tab. 3 Daty dokonania przestępstw (w miesiącach).

Na podstawie powyższych danych tworzy się pierwotną sekwencję danych czasowych, oznaczoną jako $T^{(0)}$zawierającą numery miesięcy, w których doszło do kolejnych ataków

$$T^{(0)} = [1,6,12,18].$$

Zgodnie z procedurą budowy modelu GM(1,1), kolejnym krokiem jest utworzenie sekwencji skumulowanej ($T^{(1)}$).

$$T^{(1)}(1) = T^{(0)}(1) = 1$$

$$T^{(1)}(2) = T^{(0)}(1) + T^{(0)}(2) = 1 + 6 = 7$$

$$T^{(1)}(3) = T^{(0)}(2) + T^{(0)}(3) = 7 + 12 = 19$$

$$T^{(1)}(4) = T^{(0)}(3) + T^{(0)}(4) = 19 + 18 = 37$$

$$T^{(1)} = [1,7,19,37]$$

Ostatecznie uzyskujemy szereg skumulowany. Następnie obliczamy tzw. sekwencję pośrednią $Z(k)$, wykorzystywaną do szacowania wartości parametrów modelu. Każda jej wartość stanowi średnią arytmetyczną z dwóch sąsiednich elementów sekwencji skumulowanej:

$$Z^{(1)}(k) = \frac{1}{2}\left(T^{(1)}(k-1) + T^{(1)}(k)\right), \quad k = 2,3,4.$$

Obliczenia przebiegają następująco

$$Z^{(1)}(2) = \frac{1+7}{2} = 4, \qquad Z^{(1)}(3) = \frac{7+19}{2} = 13, \; Z^{(1)}(4) = \frac{19+37}{2} = 28.$$

Ostatecznie sekwencja pośrednia przyjmuje postać:

$$Z^{(1)} = [4,13,28].$$

Model wykorzystuje sekwencję $T^{(0)}$, przedstawioną jako dane źródłowe.

W celu wyznaczenia parametrów a oraz b modelu GM(1,1), zastosowano, wcześniej już wspomnianą, metodę najmniejszych kwadratów. W analizowanym przypadku wektor obserwacji Y i macierz B przyjmują postać

$$Y = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} \quad B = \begin{bmatrix} -4 & 1 \\ -13 & 1 \\ -28 & 1 \end{bmatrix}.$$

Zatem

$$B^T \cdot B = \begin{bmatrix} -4 & -13 & -28 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} -4 & 1 \\ -13 & 1 \\ -28 & 1 \end{bmatrix} = \begin{bmatrix} 969 & -45 \\ -45 & 3 \end{bmatrix}.$$

Macierz odwrotna do powyższej jest postaci

$$(B^{\mathrm{T}} \cdot B)^{-1} = \begin{bmatrix} \frac{1}{294} & \frac{5}{98} \\ \frac{5}{98} & \frac{323}{294} \end{bmatrix}.$$

Iloczyn macierzy transponowanej i wektora obserwacji

$$B^T \cdot Y = \begin{bmatrix} -4 & -13 & -28 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix} = \begin{bmatrix} -684 \\ 36 \end{bmatrix}.$$

Ostatecznie

$$\hat{a} = (B^{\mathrm{T}} \cdot B)^{-1} \cdot B^{\mathrm{T}} \cdot Y = \begin{bmatrix} \frac{1}{294} & \frac{5}{98} \\ \frac{5}{98} & \frac{323}{294} \end{bmatrix} \cdot \begin{bmatrix} -684 \\ 36 \end{bmatrix} = \begin{bmatrix} -\frac{24}{49} \\ \frac{228}{49} \end{bmatrix},$$

co po uproszczeniu daje następujące wartości parametrów modelu

$$a = \frac{-24}{49} \approx -0{,}49 \quad , \quad b = \frac{228}{49} \approx 4{,}65.$$

Wartość parametru a odpowiada za tempo zmian skumulowanej sekwencji i związana jest z dynamiką zjawiska (tu cykl przestępstw), natomiast b reprezentuje tzw. wpływ zewnętrzny – wartość, do której dąży system w dłuższej perspektywie.
Następnie przechodzimy do obliczenia wartości prognozowanej dla piątego, szukanego czasu przestępstwa ($k = 5$), w którym wykorzystujemy dane historyczne do wyznaczenia trendu

$$T^{(1)}(5) = \left(1 - \frac{4{,}65}{-0{,}49}\right) \cdot e^{-(-0{,}49)\cdot 4} + \frac{4{,}65}{-0{,}49},$$

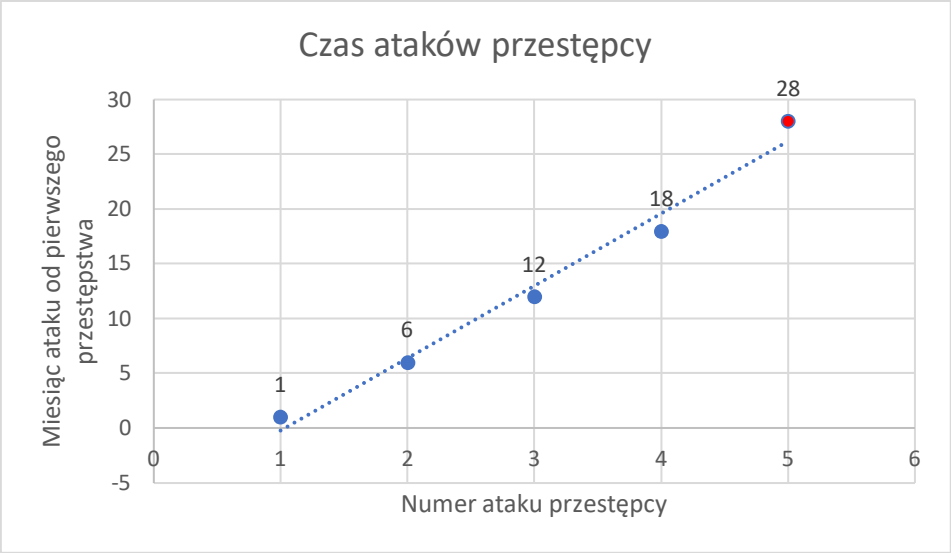
$$T^{(1)}(5) = (1 + 9{,}497) \cdot e^{1{,}96} - 9{,}49 \approx 10{,}497 \cdot 7{,}1 - 9{,}497 \approx 74{,}53 - 9{,}497 = 65{,}03.$$

Na koniec wyznaczamy różnicę między wartościami funkcji w kolejnych punktach czasu

$$T^{(0)}(5) = T^{(1)}(5) - T^{(1)}(4) = 65{,}03 - 37 = 28{,}03,$$

$$T^{(0)}(5) = 28{,}03 \approx 28.$$

Wynik ten interpretowany jest jako przewidywany moment kolejnego przestępstwa, który odpowiada 28–emu miesiącowi od początku obserwacji. Obserwacje rozpoczęto w styczniu 2020 roku, zatem przewidywany miesiąc wystąpienia kolejnej zbrodni przypada na kwiecień 2022 roku.



Rys. 2 Wykres trendu ataków (czasowych).

Niebieskie punkty oznaczają rzeczywiste czasy ataków, czerwony punkt to przewidywany czas następnego ataku (28 miesiąc), a przerywana linia to trend wyznaczony na podstawie danych rzeczywistych.

Podsumowanie

Matematyczne narzędzia - formuła Rossmo i model GM(1,1), mogą ułatwić śledczym zarówno zawężenie obszaru potencjalnego pobytu sprawcy, jak i przewidywanie momentu kolejnego ataku. Dzięki temu możliwe jest skuteczniejsze i szybsze wykorzystanie danych przestrzennych oraz czasowych w działaniach operacyjnych. Wiadomo, że to czasami właśnie czas i precyzja w poszukiwaniu przestępcy są najważniejsze. Wykorzystanie tych technik pokazuje, że matematyka może stanowić realne wsparcie dla organów ścigania, zwiększając efektywność prowadzonych analiz i interwencji. Zaprezentowane metody wskazują, że matematyka nie jest tylko narzędziem teoretycznym. Może praktycznie służyć operacyjnej praktyce i jest niezbędnym elementem nowoczesnych metod walki z przestępczością.

Literatura

[1] Syed F. S., The Role of Geographic Profiling in Criminal Investigations, Malaysian Journal of Medical Sciences, vol. 20, no. 1, pp. 125–145, 2013
[2] Samlan Y., Rossmo-Plotter, Repozytorium GitHub, 2019
[3] Cempel C., Teoria szarych systemów – nowa metodologia analizy i oceny złożonych systemów. Przegląd możliwości, Zeszyty Naukowe Politechniki Poznańskiej. Organizacja i Zarządzanie, 2014

[4] Syed F. S., Li D., Zhang X., Gu Z., Mathematical Modelling in Criminology, Malaysian Journal of Mathematical Sciences, vol. 7, no. 1, pp. 125–145, 2013

MathUp
Konferencja Zastosowań Matematyki

JAK LICZBY PIERWSZE CHRONIĄ NASZE DANE – I CZY KOMPUTERY KWANTOWE TO ZMIENIĄ?

Kinga HISZPAŃSKA, Alicja SZOSTAK
Politechnika Gdańska, Gdańsk

Wstęp

Liczby pierwsze pojawiają się już na wczesnym etapie poznawania nauk matematycznych i stąd znane nawet od szkoły podstawowej. Liczba pierwsza to taka, która dzieli się przez jeden i samą siebie. Własność ta wydaje się zupełnie nieznacząca w świecie zaawansowanej nauki, jednakże liczby pierwsze odgrywają fundamentalną rolę w wielu dziedzinach matematyki i informatyki.

Szczególną rolę odgrywają w teorii liczb oraz kryptografii, gdzie – jak sam tytuł sugeruje – chronią nasze dane i przyczyniają się do rozwoju komputerów kwantowych. Właśnie te zastosowania oraz droga od poszukiwania liczb pierwszych aż do rozwoju najnowszych technologii zostaną szerzej omówione w niniejszym artykule.

Geneza

Wraz z odkrywaniem kolejnych zastosowań liczb pierwszych, ludzie zaczęli im poświęcać coraz więcej uwagi. Wielu matematyków organizowało konkursy na odnalezienie jak największej liczby pierwszej. Ich głównymi motywacjami były: sława, chęć odnalezienia nowych liczb doskonałych, czy pragnienie głębszego poznania struktur liczbowych.

Jedną z postaci, które przyczyniły się do rozwoju tej dziedziny, był Marin Mersenne – francuski duchowny katolicki i uczony (teolog, filozof, matematyk i teoretyk muzyki). Jego wkład w badania nad liczbami pierwszymi jest na tyle znaczący, że znamy 28 liczb Mersenne'a, z których osiem należy do największych znanych liczb pierwszych. Są one postaci:

$$2^p - 1,$$

gdzie p jest liczbą pierwszą.
Największa liczba posiada wykładnik 51 i jest złożona z 41 milionów cyfr.

Jak się jednak później okazało powyższy wzór nie jest spełniany przez $p = 61,\ 87,\ 107$.

Nie tylko liczby nazwano imieniem Mersenne'a; istnieje także algorytm Mersenne Twister, który służy do generowania liczb pseudolosowych. Długość okresu tego generatora wynosi $2^{19937} - 1$, czyli taka ilość liczb pseudolosowych zostanie wygenerowanych, zanim ciąg zacznie się powtarzać. Liczba 19937 jest wykładnikiem jednej z liczb Mersenne'a, zatem twórcy zdecydowali się użyć nazwiska mnicha do nazwy algorytmu.

Jak pokazuje powyższy przypadek, matematycy szukali nie tylko liczb pierwszych, ale i sposobów na ich wyznaczenie. Jednym z takich sposobów jest metoda sita. Metody sita to metody służące do oszacowania liczebności różnych zbiorów, zdefiniowanych przy użyciu własności multiplikatywnych. Sito Eratostenesa to najprostsza i najczęściej stosowana metoda w praktyce dla umiarkowanie dużych liczb. Oto instrukcja jego działania:

1. Zapisz wszystkie liczby naturalne od 2 do n .
2. Zaczynając od 2 skreśl wszystkie jej wielokrotności.
3. Znajdź kolejną nieprzekreśloną liczbę i skreśl jej wielokrotności.
4. Powtarzaj krok 3, aż dojdziesz do liczby większej od $\sqrt{n}$.
5. Wszystkie nieprzekreślone liczby są liczbami pierwszymi.

Jak widać jest to bardzo łatwy sposób na szukanie liczb pierwszych.
Na przykład, dla $n = 12$ mamy

2	3	4	5	6	7	8	9	10	11	12

Skreślamy wszystkie wielokrotności 2 i zostają liczby:

2	3	5	7	9	11

Kolejną nieprzekreśloną liczbą jest 3, więc skreślmy teraz jej wielokrotności:

2	3	5	7	11

Następną liczbą jest 5, ale 5 jest większe od $\sqrt{12}$, zatem 2, 3, 5, 7 i 11 są liczbami pierwszymi.

Kolejną równie przystępną metodą jest test pierwszości Fermata, nazwany od Pierre'a Fermata – francuskiego prawnika i matematyka żyjącego w XVII wieku. W przeciwieństwie do wyżej opisanego sita, które sprawdza wiele liczb naraz, test ten służy do oceny pojedynczych liczb i stwierdzenia, czy dana liczba może być pierwsza. Test pierwszości Fermata polega na tym, że dla liczby całkowitej p > 2, wybiera się losowo liczbę a taką, że 1 < a < p . Następnie sprawdza się, czy zostało spełnione małe twierdzenie Fermata:

$$a^{p-1} - 1 \equiv 0 \,(mod\, p).$$

Trzeba pamiętać, że a i p muszą być względnie pierwsze, czyli największy wspólny dzielnik tych liczb wynosi 1. Gdy otrzymana wartość ($a^{p-1}\, mod\, p$) jest różna od 1, to sprawdzana liczba z pewnością nie jest pierwsza. Jeśli jednak wynik wynosi 1 to liczba może być pierwsza, jednakże trzeba uważać na liczby Carmichaela, które spełniają małe twierdzenie Fermata, lecz są złożonymi liczbami naturalnymi na przykład 561. Przykładowo sprawdźmy p = 7 i wybierzmy a = 3 (7 i 3 są względnie pierwsze). Zgodnie z małym twierdzeniem Fermata:

$$3^{7-1} - 1 \equiv 0 \,(mod\, 7)$$
$$729 \equiv 1 \,(mod\, 7).$$

Otrzymujemy 1, zatem test pierwszości Fermata nie wyklucza faktu, że liczba 7 może być pierwsza.

Mimo swej prostoty wymienione metody są bardzo czasochłonne przy dużych liczbach. Dlatego wraz z rozwojem komputerów powstały algorytmy, które umożliwiają szybkie sprawdzanie pierwszości. Na przykład za pomocą algorytmu szybkiego potęgowania, czyli metody umożliwiającej szybkie obliczanie potęg o dużym wykładniku naturalnym, test pierwszości Fermata można wykonać w czasie:

$$O(k \cdot \log^3 n),$$

gdzie k to liczba prób z różnymi wartościami a.

Kryptografia

Ze względów politycznych i wojskowych rozwinęła się dziedzina nauki zajmująca się zabezpieczaniem informacji poprzez ich szyfrowanie, tak aby były one dostępne wyłącznie dla uprawnionych odbiorców. Dziedzina ta nosi nazwę kryptografii i posiada zastosowanie między innymi w bankowości, komunikatorach internetowych, sieci Wi-Fi, czy też dokumentacji wojskowej oraz politycznej.

Pierwsze znane próby szyfrowania sięgają około 1900 roku p.n.e. i pochodzą ze starożytnego Egiptu. Zastosowano wówczas nietypowe hieroglify o symbolicznym charakterze, które miały ukrywać prawdziwe znaczenie tekstu.

Z biegiem czasu, wraz z rosnącą wartością i wrażliwością informacji, ludzie zaczęli opracowywać coraz bardziej zaawansowane metody ich ochrony – od prostych szyfrów po złożone systemy kryptograficzne. Liczby pierwsze odegrały fundamentalną rolę w rozwoju kryptografii i do dziś stanowią podstawę wielu systemów bezpieczeństwa w informatyce. W szczególności wykorzystywane są w tzw. funkcjach haszujących oraz w algorytmach szyfrowania operujących na liczbach w systemie modulo. Dzięki swojej unikalnej strukturze liczby pierwsze sprawiają, że operacje te są trudne do odwrócenia, co znacznie zwiększa poziom bezpieczeństwa danych.

Zastosowanie liczb pierwszych umożliwia tworzenie funkcji, które są łatwe do wykonania w jedną stronę, ale bardzo trudne do odwrócenia bez znajomości odpowiednich kluczy – a właśnie na tym opiera się nowoczesna kryptografia. Jednym z najbardziej znanych i powszechnie używanych asymetrycznych algorytmów szyfrowania jest RSA (od nazwisk jego twórców: Rivest, Shamir, Adleman). W metodach asymetrycznych zarówno nadawca, jak i odbiorca dysponują oddzielną parą kluczy. Para ta składa się z klucza publicznego, który jest jawny i dostępny publicznie, oraz klucza prywatnego, który musi być przechowywany w tajemnicy. Nadawca używa klucza publicznego odbiorcy do zaszyfrowania wiadomości, a odbiorca odszyfrowuje ją za pomocą swojego prywatnego klucza. Asymetria polega na tym, że dane zaszyfrowane kluczem publicznym mogą zostać odszyfrowane wyłącznie przy użyciu odpowiadającego mu klucza prywatnego.

Przykład szyfrowania danych za pomocą algorytmu RSA:

Stosujemy oznaczenia:

- p, q – tajne liczby pierwsze (para (p, q) tworzy klucz prywatny),
- s – wykładnik szyfrowania (para $(p \cdot q, s)$ tworzy klucz publiczny),
- dla ułatwienia $r = p \cdot q$ (wtedy klucz publiczny to para (r, s)),
- C – zaszyfrowana wiadomość.

Niech $p = 23$, $q = 17$, $s = 5$. Za pomocą algorytmu RSA zaszyfrujemy wiadomość ”X”.

Najpierw obliczmy r : $r = 23 \cdot 17 = 391$

Teraz mamy dwa klucze: prywatny $= (23, 17)$ i publiczny$= (391, 5)$. W alfabecie łacińskim litera "X" jest na 24. pozycji, zatem: $L = 24$.

Szyfrowanie odbywa się według wzoru: $C \equiv L^s (mod\, r)$
$C \equiv 24^5 (mod\, 391) = 300$

Jak wynika z obliczeń, zakodowana wiadomość to "300".

Odszyfrowanie przebiega według wzoru: $L \equiv C^{\,q}\ (mod\, r),$
$L \equiv 300^{17} (mod\, 24).$

W alfabecie łacińskim dwudziestą czwartą literą jest 'X', zatem odszyfrowaliśmy wiadomość.

Ogólne sito ciała liczbowego

Metoda, która obecnie pozwala na najszybszą faktoryzację dużych liczb, to Ogólne Sito Ciała Liczbowego (*GNFS - Great Number Field Sieve*), może być więc zastosowane przy próbie złamania klucza RSA. Metoda GNFS jest zoptymalizowaną wariacją algorytmu sita liczbowego (*NFS - Number Field Sieve*). W ogólności GNFS polega na szukaniu odpowiednich gładkich liczb, czyli takich, których wszystkie dzielniki pierwsze są mniejsze lub równe od określonego progu ustalonego przez daną bazę liczb pierwszych.

Algorytm GNFS poddający faktoryzacji liczbę obejmuje wykonanie następujących kroków opisanych skrótowo poniżej:

1. *Wybór liczby oraz odpowiadającego jej wielomianu.*

Wybrana liczba powinna być naturalna, a odpowiadający jej wielomian spełniać warunek:

$$f(m) \equiv 0\ (mod\, n)$$

Najłatwiej go spełnić, gdy:

$$f(m) = a_d m^d + a_{(d-1)} m^{(d-1)} + \ \ldots\ + a_0 = n.$$

gdzie a , d to odpowiednie współczynniki.

2. *Definicja baz liczbowych*

Zdefiniujmy teraz bazy liczbowe R oraz A . Baza R składa się ze skończonej liczby liczb pierwszych, zaś baza A zawiera pary liczb $(r,\ p)$, określone zależnością:

$$f(r) \equiv 0\ (\, mod\, p\,)$$

Liczby p są liczbami pierwszymi. Wyróżnia się również bazę Q , złożoną z par $(s,\ q)$, które nie zostały uwzględnione w bazie A .

3. *Zastosowanie sita w znalezieniu par liczb.*

Oznaczmy przez θ zmienną reprezentującą pierwiastek z wybranego w pierwszym punkcie wielomianu. Po zdefiniowaniu baz należy przystąpić do znalezienia metodą sita par liczb (a, b), dla których następujące wyrażenia:

$$x = a + bm$$
$$y = a + b\theta$$

są gładkie odpowiednio dla baz liczbowych R oraz A. Oznacza to, że ich dzielniki występują w danej bazie.

4. *Konstrukcja macierzy.*

Odnalezione pary (a, b) pozwalają na skonstruowanie macierzy X. Każdy wiersz macierzy reprezentuje daną parę (a, b). Wartości w poszczególnych kolumnach zależą od znaku przyjmowanego przez wyrażenie, czynników pierwszych gładkich wyrażeń opisanych w poprzednim punkcie, a dokładniej od potęg, w jakich te czynniki występują przy faktoryzacji oraz wartości przyjmowanych przez określone ilorazy. Dokładna postać wiersza macierzy X dla danej pary (a, b) wygląda następująco:

$$\left[\begin{cases} 0, a+bm \geq 0 \\ 1, \text{else} \end{cases} \right. \quad \begin{matrix} e_1 \pmod 2 & e_2 \pmod 2 & \cdots & e_k \pmod 2 \\ f_1 \pmod 2 & f_2 \pmod 2 & \cdots & f_l \pmod 2 \end{matrix}$$
$$\begin{cases} 0, \left(\frac{a+bs_1}{q_1}\right) = 1 \\ 1, \text{else} \end{cases} \quad \begin{cases} 0, \left(\frac{a+bs_2}{q_2}\right) = 1 \\ 1, \text{else} \end{cases} \quad \cdots \quad \left. \begin{cases} 0, \left(\frac{a+bs_u}{q_u}\right) = 1 \\ 1, \text{else} \end{cases} \right]$$

Rys. 1 Postać pojedynczego wiersza macierzy X [2].

Po zdefiniowaniu macierzy X, wykorzystuje się ją do rozwiązania równania postaci:

$$X^T \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_y \end{bmatrix} \equiv 0 \mod 2$$

Elementy macierzy A_1, A_2, ... przyjmują wartość 0 albo 1, w zależności od tego czy wyrażenie dla danej pary $(a, b)_i$ jest równe liczbie będącej kwadratem całkowitym. Takie pary można wydzielić do nowej bazy V.

5. *Rozwiązanie równania*

Dla wybranych par (a, b) znajdujących się w bazie V oblicza się dwa iloczyny dla wyrażeń $x = a + bm$ oraz $y = a + b\theta$. Następnie należy wyznaczyć pierwiastki z tych iloczynów oraz przeprowadzić przekształcenie (mapowanie) wyrażenia z θ. Wówczas następuje weryfikacja, czy uzyskane pierwiastki spełniają poniższą zależność:

$$y^2 \equiv x^2 (mod\, n).$$

Szukane czynniki pierwsze można znaleźć wykorzystując algorytm Euklidesa do wyznaczania największego wspólnego dzielnika:

$$NWD(n, y + x) = P_1,$$
$$NWD(n, y - x) = P_2.$$

Stąd: $n = P_1 \cdot P_2$.
Jeśli odnalezione liczby nie są czynnikami pierwszymi, należy powtórzyć algorytm dla ponownie wybranej wartości m.

Jak można zauważyć, algorytm ogólnego sita ciała liczbowego ma skomplikowaną procedurę. Aby pomóc w zrozumieniu mechanizmu jego działania, poniżej przedstawiamy przykład zastosowania metody sita liczbowego (NFS), uproszczonej wersji GNSF.

Przykład

Poszukujemy czynników pierwszych liczby $n = 51$. Ustalmy bazę czynników $P = \{2, 5, 7, 11, 13\}$. Celowo nie ma w niej dzielników n – gdyby występowały w bazie, nie byłoby potrzeby stosowania algorytmu. Następnie losowane są wartości z, które składają się z czynników pierwszych występujących w bazie P. Tworzy się wówczas relacje między wartościami z, a liczbami, których reszta po dzieleniu przez n jest równa z. Poniżej znajdują się przykładowe relacje:

1. $z = 1:\quad 2^0 5^0 7^0 11^0 13^0 = 1 \equiv 52 = 2^2 5^0 7^0 11^0 13^1$
2. $z = 4:\quad 2^2 5^0 7^0 11^0 13^0 = 4 \equiv 55 = 2^0 5^1 7^0 11^1 13^0$
3. $z = 5:\quad 2^0 5^1 7^0 11^0 13^0 = 5 \equiv 56 = 2^3 5^0 7^1 11^0 13^0$
4. $z = 13:\quad 2^0 5^0 7^0 11^0 13^1 = 13 \equiv 64 = 2^6 5^0 7^0 11^0 13^0$

Spośród znalezionych relacji trzeba wybrać takie połączenia, dla których zachodzi relacja $a^2 \equiv b^2\, mod(n)$, wartość a jest iloczynem wybranych relacji, a b odpowiada liczbie, która podniesiona do kwadratu jest resztą dzielenia a przez n. W naszym przypadku jest to spełnione dla relacji $3 \cdot 4$:

$$56^2 \cdot 64^2 \equiv 25 \cdot 16 = 20^2\ (mod\ 51).$$

Równanie można sprowadzić do postaci:

$$14^2 \equiv 20^2\ (mod\ 51),$$
$$Z^2 \equiv S^2\ (mod\ 51).$$

Szukamy teraz największego wspólnego dzielnika dla następujących par liczb:

$$NWD(Z - S, n) = NWD(14 - 20, 51) = NWD(-6, 51) = 3,$$
$$NWD(Z + S, n) = NWD(14 + 20, 51) = NWD(34, 51) = 17.$$

W ten sposób poznaliśmy czynniki pierwsze liczby 51.

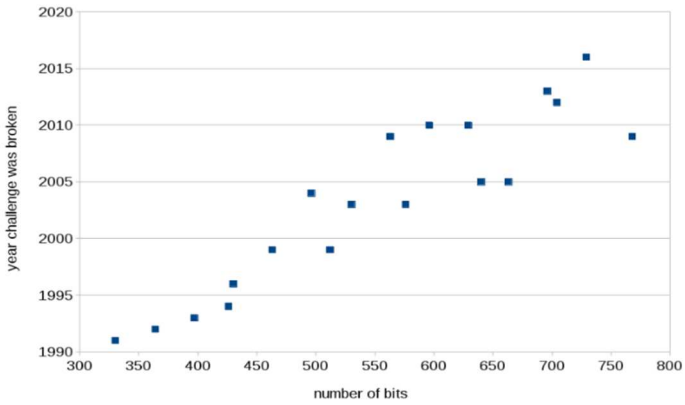
Zależność między liczbą znaków a czasem faktoryzacji dla algorytmu Ogólnego Sita Ciała Liczbowego ma charakter wykładniczy:

$$O\left(e^{((64/9)^{(1/3)} \cdot (\ln n)^{(1/3)} \cdot (\ln \ln n)^{(2/3)})}\right)$$

Oznacza to, że faktoryzacja bardzo dużych liczb pierwszych za pomocą GNFS, pomimo bycia najefektywniejszą metodą, wymaga niewyobrażalnej ilości czasu i mocy obliczeniowej. Poniższy

wykres przedstawia postępy w faktoryzacji liczb o coraz to większej liczbie bitów. Bit oznacza pojedynczą cyfrę w zapisie binarnym liczby, np. liczba 5 w binarnym zapisie to 101 – ma 3 bity.

W 2020 roku dokonano faktoryzacji klucza RSA o 250 liczbach dziesiętnych (czyli składającego się z 829 bitów). Przy użyciu pojedynczego komputera proces ten zająłby 2700 lat. Klucze RSA używane powszechnie w szyfrowaniu danych posiadają aż 2048 bitów, czyli ponad 600 cyfr. Biorąc to pod uwagę, można stwierdzić, że obecnie bezpieczeństwo kluczy RSA nie jest zagrożone klasycznymi metodami faktoryzacji.



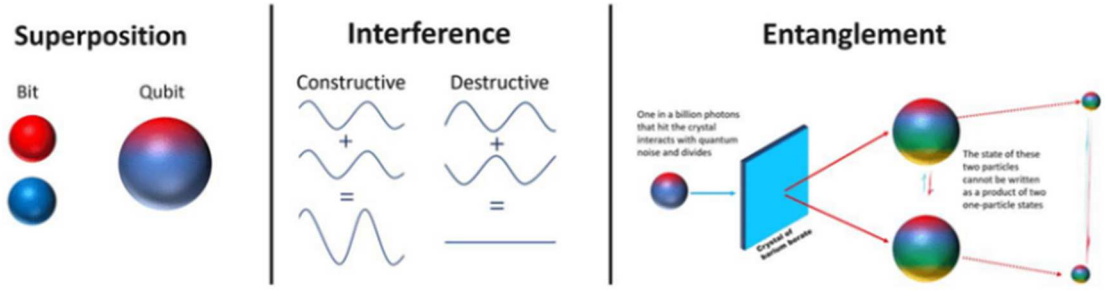
Rys. 2 Wykres przedstawiający lata, dla których po raz pierwszy łamane były szyfry RSA o danej liczbie bitów. [7]

Kryptografia kwantowa

W ostatnich latach coraz częściej słyszy się o postępach związanych z budową komputera kwantowego – urządzenia działającego w inny sposób niż powszechnie używane klasyczne komputery. Otwiera to również nowe możliwości związane z wydajnością i charakterem algorytmów, które można przeprowadzić z pomocą komputerów kwantowych.

W komputerach kwantowych podstawową jednostką informacji jest kubit. Różni się on od bitów dla klasycznych komputerów tym, że podlega zjawiskom opisywanym przez mechanikę kwantową takim jak:

1. *Superpozycja* – oznacza, że kubity mogą być jednocześnie w wielu stanach z różnym prawdopodobieństwem.
2. *Interferencja* – w jej wyniku następuje wzmacnianie lub osłabianie prawdopodobieństw różnych wyników.
3. *Splątanie* – czyli połączenia między kubitami, przez co ich zmiany wpływają wzajemnie na siebie.



Rys. 3 Zjawiska wykorzystywane przez komputer kwantowy. [8]

Dzięki tym właściwościom, komputery kwantowe są w stanie wykonywać operacje w znacznie szybszym czasie niż klasyczne komputery. Przetwarzają one wiele możliwych stanów jednocześnie i szybciej dochodzą do prawidłowego rozwiązania, zwłaszcza tam, gdzie klasyczne maszyny muszą próbować każdą możliwość osobno.

W przypadku faktoryzacji dużych liczb naturalnych, algorytmem kwantowym, który może zostać zastosowany jest *Algorytm Shora*. Opracował go w 1994 Peter Shor. Algorytm ten stał się powszechnie znany jako algorytm kwantowy stanowiący potencjalne zagrożenie dla systemu RSA. Poniżej, w kilku krokach przedstawiamy procedurę jego realizacji.

Algorytm Shora
Faktoryzacji poddana zostanie liczba N, z kolei liczba g to dowolnie wybrana liczba naturalna mniejsza od N, służy jako punkt startowy w realizacji algorytmu. Ważnym punktem procedury jest skorzystanie z poniższej zależności:

$$g^p - 1 \equiv m \cdot N,$$

gdzie g, p, m *oraz* N są dodatnimi liczbami całkowitymi.

Wynika z niej, że jakakolwiek wybrana liczba g podniesiona do odpowiedniej potęgi p pomniejszona o jeden jest wielokrotnością szukanej liczby N. Wyrażenie po lewej stronie równania można zapisać w postaci iloczynu.

$$\left(g^{\frac{p}{2}} + 1\right)\left(g^{\frac{p}{2}} - 1\right) \equiv m \cdot N$$

Wówczas wyrażenia w nawiasie z pewnym prawdopodobieństwem są wielokrotnościami czynników pierwszych liczby N. Prawdopodobieństwo to wynosi dokładnie 37,5%, w pozostałych przypadkach okazuje się, że wartość p jest nieparzysta lub wartości w nawiasie są wielokrotnością N. Oznacza to, że w 99% wystarczy mniej niż 10 prób dobrania parametru g, aby przeprowadzić udaną faktoryzację. Kolejne kroki algorytmu koncentrują się wokół znalezienia odpowiedniej potęgi p.

1. *Część klasyczna:*

Na wejściu podawana jest liczba x, która tworzy liczbę g^x. Zarówno x *jak i* g^x są zapisywane. Później badane jest o ile g^x jest większe od liczby N pomnożonej przez możliwie najmniejsze m, tę różnicę oznacza się jako r. Na wyjściu zapisywana jest wartość x oraz reszta r:

$$|x\rangle \to g^x \to |x, g^x\rangle \to m \cdot N \to |x, +r\rangle.$$

W tym miejscu spotykamy się z charakterystycznym oznaczeniem w mechanice kwantowej: | ⟩. Jest to tzw. ket, służący do opisu stanów kwantowych. $|x\rangle$ można przedstawić jako wektor kolumnowy (macierz o jednej kolumnie), którego elementy odpowiadają różnym możliwym wartościom x:

$$|x\rangle = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Taki sposób zapisu podkreśla, że dzięki superpozycji kubitów weryfikację kolejnych liczb x można przeprowadzić równolegle, co zwiększa szybkość algorytmu.

2. *Część kwantowa:*

Celem tej części algorytmu jest znalezienie rozwiązania wśród zbioru wartości x oraz odpowiadających reszt o postaci:

$$|p, +1\rangle.$$

Czyli szukamy wśród x wartości p, dla której r jest równe 1. Jest to możliwe z wykorzystaniem prostej matematycznej obserwacji. Przy mnożeniu g^p oraz g^x, gdzie x jest dowolne, otrzymuje się nową wielokrotność m_2, ale reszta r nadal odpowiada danej liczbie x.

$$g^{(p+x)} = g^p \cdot g^x = (m \cdot N + 1)(m_1 \cdot N + r) = (mm_1N + rm + m_1) \cdot N + r = m_2 \cdot N + r.$$

Działa to również dla $g^{(2p+x)}$, $g^{(3p+x)}$ itd., a więc jest to powtarzalna własność, którą można wykorzystać wraz z superpozycją stanów.

Kolejnym krokiem jest wybór jednej wartości r, przykładowo $r = 3$, i wydzielenie tych potęg x, dla których występuje r o takiej wartości. Okazuje się, że te potęgi znajdują się w odległości od siebie równej p, występują więc z pewną częstotliwością:

$$f = \frac{1}{p}.$$

Do znalezienia częstotliwości f posłuży kwantowa transformata Fouriera, która jest w stanie wydzielić z superpozycji stanów te, które powtarzają się z zadaną częstotliwością, a pozostałe poddać interferencji destruktywnej.

$$|x\rangle + |x + p\rangle + |x + 2p\rangle + \cdots \rightarrow QFT \rightarrow |1/p\rangle.$$

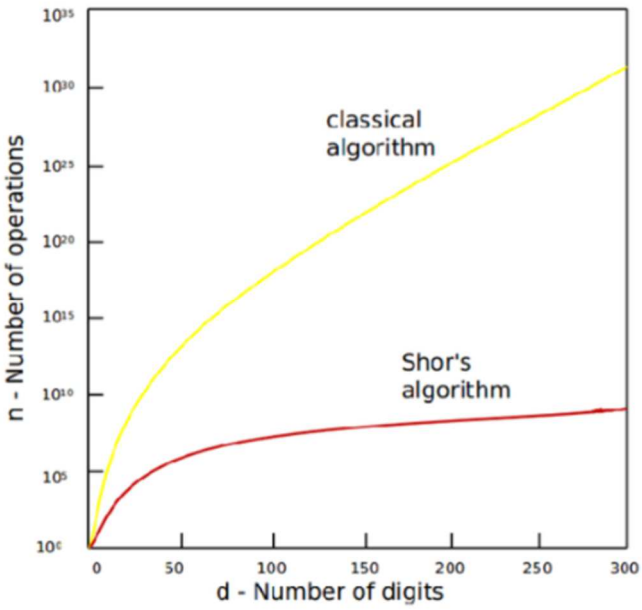
Jeśli uzyskane p jest liczbą parzystą, procedura zakończyła się powodzeniem i wówczas wystarczy wykorzystać algorytm Euklidesa, aby znaleźć największe wspólne dzielniki liczby N oraz wyrażeń występujących w początkowym wzorze.

$$\left(g^{\left(\frac{p}{2}+1\right)}\right)\left(g^{\left(\frac{p}{2}-1\right)}\right) = a \cdot P^1 \cdot b \cdot P^2 = m \cdot N$$

Wykorzystując komputer kwantowy czas faktoryzacji liczby pierwszej N wyniósłby:

$$O((\log N)^3).$$

Dla dużych liczb różnica w czasie obliczeniowym między klasycznym, a kwantowym komputerem staje się ogromna, co pokazuje poniższy wykres zależności liczby operacji od liczby cyfr liczby poddawanej faktoryzacji. Dla liczby 250-cyfrowej algorytm Shora dokona rozkładu na czynniki pierwsze około 10^{20} razy szybciej.



Rys. 4 Zależność liczby operacji od ilości cyfr w liczbie dla klasycznego algorytmu i algorytmu Shora. [9]

Podsumowanie

Komputery kwantowe mają potencjał do łamania klasycznych systemów szyfrowania, takich jak RSA. Szacuje się, że maszyna z miliardem kubitów mogłaby złamać klucz RSA-2048 w zaledwie 8 godzin. Obecnie dostępne komputery kwantowe dysponują jednak jedynie 50–80 kubitami. Do tej pory algorytm Shora został zastosowany do faktoryzacji małych liczb, takich jak 15, 21 i 35, nawet w tym przypadku działanie było dalekie od ideału. Pomimo ograniczeń, rozwój tej technologii stale postępuje, jednak nie stanowi ona realnego zagrożenia dla klasycznej kryptografii w najbliższej przyszłości.

Literatura

[1] Kijko, T., Wroński, M., Comparison of Algorithms for Factorization of Large Numbers Having Several Distinct Prime Factors, Institute of Mathematics and Cryptology, Cybernetics Faculty, Military University of Technology, Warsaw, Poland, 2023
[2] Case, M., A Beginner's Guide to the General Number Field Sieve, Oregon State University, ECE 575, [dostęp: 01.04.2025]
[3] Narkiewicz, W.: Teoria Liczb (Wyd. II poprawione), Warszawa, 1990
[4] How Quantum Computers Break Encryption | Shor's Algorithm Explained https://www.youtube.com/watch?v=lvTqbM5Dq4Q, [dostęp: 01.04.2025]
[5] Czerwiński, W., Algorytm faktoryzacji Shora, https://www.deltami.edu.pl/2017/12/algorytm-faktoryzacji-shora/ [dostęp: 05.2025]
[6] Amico, M., Saleem, Z. H., & Kumph, M.: An Experimental Study of Shor's Factoring Algorithm on IBM Q, The Graduate School and University Center, The City University of New York; Theoretical Research Institute of Pakistan Academy of Sciences; IBM T.J. Watson Research Center, 2020
[7] https://www.esat.kuleuven.be/cosic/blog/another-rsa-challenge-broken/ [dostęp: 01.04.2025]
[8] Giani, A., & Eldredge, Z., Quantum computing opportunities in renewable energy. SN Computer Science, 2(5), 2021
[9] Gupta, K. D., Nag, A., Rahman, M. L., & Mahmud, M. A. P., Utilizing computational complexity to protect crypto-currency against quantum threats: A review. IT Professional, 23(5), 50–55, 2021

MathUp
Konferencja Zastosowań Matematyki

TO TEŻ MOŻE CI SIĘ SPODOBAĆ...". CZYLI O ALGORYTMACH ASOCJACYJNYCH SŁÓW KILKA.

Kacper WIĄCEK

Politechnika Krakowska im. Tadeusza Kościuszki, Kraków

Wstęp

Odkrywanie reguł asocjacyjnych stanowi kluczowy element analizy koszykowej, umożliwiający identyfikację zależności pomiędzy obiektami w dużych zbiorach danych. Dzięki ekstrakcji z bazy danych zbiorów elementów występujących wystarczająco często razem, możemy wyciągnąć wnioski co do prawdopodobieństwa ich dalszego wspólnego występowania. Tworzone na podstawie częstych zbiorów elementów implikacje nazywamy regułami asocjacyjnymi. Wyrażają one empiryczną wartość probabilistyczną np. tego, jaka jest szansa, że klient zakupi określone przedmioty, jeśli wiadomo jakie przedmioty już kupił. Algorytmy służące do wynajdywania reguł asocjacyjnych nazywamy algorytmami asocjacyjnymi. Używane są do efektywnej ekstrakcji reguł asocjacyjnych z tabeli o liczbie sięgającej kilkuset tysięcy wierszy, stanowią więc cenne narzędzie w analizie danych, szczególnie w marketingu, w celu tworzenia trafnych rekomendacji, ale także w medycynie, bioinformatyce czy kryptografii.

Tabela transakcji

Towarami lub przedmiotami (*ang. Items*) nazywać będziemy zbiór binarnych atrybutów:

$$I = \{I_1, I_2, \dots, I_i\},$$

natomiast zbiorem przedmiotów X nazwiemy dowolny niepusty podzbiór zbioru atrybutów.

Mówimy, że n-wymiarowy wektor w jest binarny, jeśli:

$$\forall k \in \{1, \dots, n\}: w[k] \in \{0,1\}.$$

Niech $T = \{t_1, t_2, \dots\}$ będzie zbiorem transakcji. Każdą transakcję t_n możemy przedstawić jako wektor binarny $\tilde{t}_n$, taki że:

$$\begin{cases} I_k \in t_n \Leftrightarrow \tilde{t}_n[k] = 1 \\ I_k \notin t_n \Leftrightarrow \tilde{t}_n[k] = 0 \end{cases}$$

Przykład 1a

Rozważmy następującą tabelę transakcji:

T	mleko	sok	chleb	masło	jajka
$\tilde{t}_1$	1	0	1	0	0
$\tilde{t}_2$	1	1	0	1	1
$\tilde{t}_3$	0	0	1	0	1
$\tilde{t}_4$	1	1	0	0	1
$\tilde{t}_5$	1	1	1	0	0

Tab. 1 Tabela opisująca zbiór transakcji T

W tym przypadku $T = \{t_1, t_2, t_3, t_4, t_5\}$, natomiast $I = \{mleko, sok, chleb, masło, jajka\}$. Każda transakcja jest w tej tabeli wektorem binarnym, a wartości poszczególnych współrzędnych tych wektorów informują o przynależności danego produktu do tej transakcji. Przykładowo: $t_1 = \{mleko, chleb\}$, a $\tilde{t}_1 = [1, 0, 1, 0, 0]$.

Częste zbiory przedmiotów

Wsparcie $\sigma(X)$ zbioru atrybutów X (*ang. support of the itemset*) definiujemy wzorem:

$$\sigma(X) = |\{t_i \in T : X \subseteq t_i\}|.$$

Jest to liczba transakcji, w których zakupiony został zbiór X. Jest to więc po prostu liczba wszystkich wystąpień zbioru X w całej analizowanej tabeli transakcji.

Przykład 1b

W powyższej tabeli: $\sigma(\{mleko, chleb\}) = 2$, ponieważ zbiór $\{mleko, chleb\}$ występuje w niej 2 razy, zaś, $\sigma(\{mleko\}) = 4$, bo singleton (zbiór jednoelementowy) $\{mleko\}$ występuje w tabeli 4 razy.
Wsparciem minimalnym $minS$ nazywamy dowolną, wprowadzoną przez użytkownika, liczbę naturalną.

Powiemy, że zbiór X spełnia warunek częstości (jest częsty), jeśli: $\sigma(X) \geq minS$.

Reguły asocjacyjne

Reguła asocjacyjna ma postać implikacji $X \Rightarrow Y$, gdzie: $X, Y \subseteq I$, $X \cap Y = \emptyset$.
Istotność reguły asocjacyjnej mierzona jest przy pomocy następujących parametrów:

- wsparcie reguły dane wzorem: $s(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$, gdzie N to liczba wszystkich transakcji,
- pewność reguły (*ang. confidence*) dana wzorem: $c(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$.

Przykład 1c

$$s(\{mleko\} \Rightarrow \{chleb\}) = \frac{2}{5} = 0.4$$

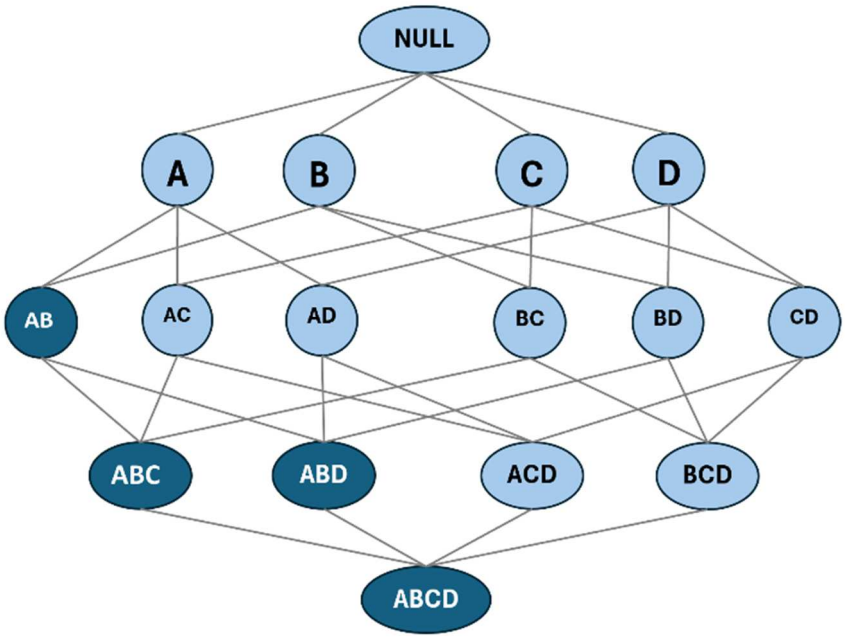
$$c(\{mleko\} \Rightarrow \{chleb\}) = \frac{2}{4} = 0.5$$
$$c(\{chleb\} \Rightarrow \{mleko\}) = \frac{2}{3} = 0.(6).$$

W przeciwieństwie do informacji o częstości zbioru, reguły asocjacyjne są niesymetryczne, pozwalając na wyciągnięcie dokładniejszych wniosków, jak w powyższym przykładzie ilustrującym, że empiryczne prawdopodobieństwo zakupienia przez klienta mleka, zważając na fakt, że kupił już chleb, jest znacząco inne niż prawdopodobieństwo zakupu chleba, jeśli wiadomo, że zostało zakupione mleko.

Algorytm Apriori

W celu stworzenia reguł asocjacyjnych dla dużych baz danych, stosuje się algorytmy asocjacyjne. Warto zaznaczyć, że reguły asocjacyjne są zdefiniowane tylko poprzez częste zbiory przedmiotów, a więc podstawowym zadaniem tych algorytmów jest znalezienie takich zbiorów, które spełniają warunek częstości zdefiniowany przez wartość $minS$ wprowadzoną przez użytkownika. Przykładem takiego algorytmu jest algorytm Apriori, który jest pierwszym takim algorytmem. Stworzony w 1994 przez R. Agrawala i R. Srikanta charakteryzuje się prostym podejściem: na podstawie tabeli k-elementowych zbiorów częstych wraz z ich wsparciem, generuje zbiory $(k+1)$-elementowe, a następnie przeszukuje całą tabelę, w celu zliczenia ich wystąpień, co dla dużych baz danych potrafi być bardzo pamięciochłonne. Proces ten jest iterowany, aż do momentu znalezienia wszystkich częstych zbiorów.

Kluczową cechą algorytmu Apriori (pod względem oszczędzania pamięci) jest pomijanie generowania tych zbiorów, których podzbiory nie spełniają warunku częstości.



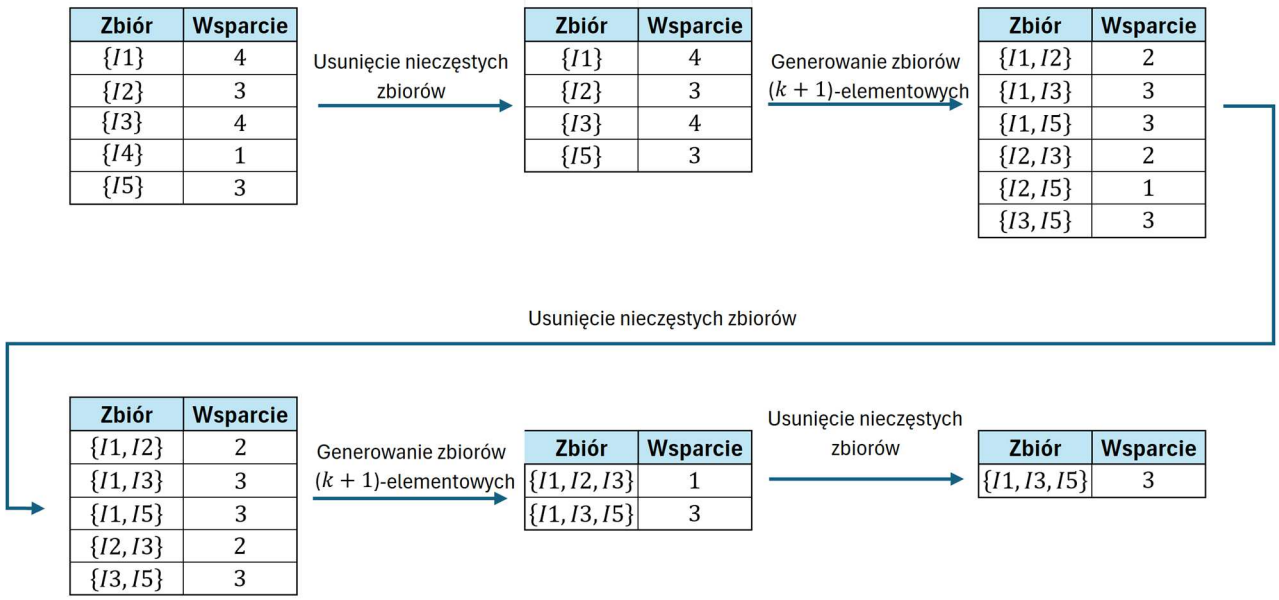
Rys. 1. Przykładowy diagram tworzenia i pomijania zbiorów dla 4-elementowej tabeli transakcji.

Przykładowo, dla powyższego diagramu: Jeśli zbiór *{AB}* został uznany przez algorytm jako niespełniający warunku częstości, to algorytm całkowicie pomija generowanie zbiorów *{ABC}, {ABD}, {ABCD},* ponieważ z definicji nie mogą one występować w tabeli częściej niż zbiór *{AB}*.

Przebieg algorytmu Apriori zobrazowany zostanie na następującej, przykładowej tabeli pięciu transakcji, zawierającej pięć przedmiotów, przyjmując $minS = 2$. W celu zwiększenia czytelności przyjmijmy $T = \{T100, T200, T300, T400, T500\}$ oraz $I = \{I1, I2, I3, I4, I5\}$.

ID Transakcji	Lista przedmiotów
$T100$	$I1, I2$
$T200$	$I2, I3$
$T300$	$I1, I3, I4, I5$
$T400$	$I1, I3, I5$
$T500$	$I1, I2, I3, I5$

Tab. 2



Rys. 2 Schemat działania algorytmu Apriori na przykładzie Tabeli 2.

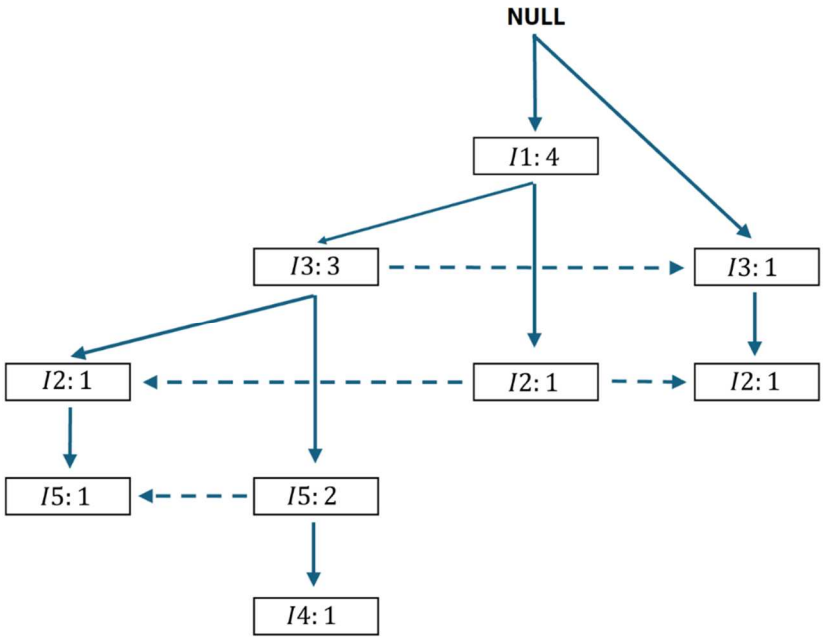
W wyniku usunięcia zbioru $\{I4\}$ *w kroku 2,* w następnym kroku nie będą generowane zbiory $\{I1, I4\}, \{I2, I4\}, \{I3, I4\}, \{I4, I5\}$, a w wyniku dodatkowego usunięcia zbioru $\{I2, I5\}$ w kroku 4, w dalszych krokach nie będą generowane zbiory: $\{I1, I2, I4, \}, \{I1, I2, I5\}, \{I1, I3, I4\}, \{I1, I4, I5\}$, $\{I2, I3, I4\}, \{I2, I4, I5\}, \{I2, I3, I5\}$ $\{I3, I4, I5\}$. Natomiast dzięki usunięciu zbioru $\{I1, I2, I3\}$, algorytm nie generuje już zbiorów 4-elementowych (ponieważ znaleziony został tylko jeden zbiór częsty długości 3). W tym przypadku proces ten zmniejsza liczbę wymaganych skanów całej tabeli z 31 do 13, gdyż

$$5 + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + 1 = 31$$

Jest to liczba wszystkich kombinacji zbioru 5-elementowego, natomiast dzięki wcześniejszemu usunięciu ww. zbiorów, algorytm zlicza wystąpienia jedynie 13 zbiorów.

Algorytm FP-Growth

Kolejnym omawianym algorytmem jest algorytm FP-Growth (*ang. frequent pattern growth*), który kompresuje tabelę transakcji w drzewo reguł (*ang. FP-Tree*) poprzez tworzenie gałęzi będących kolejnymi transakcjami, w których przedmioty posortowane są malejąco względem ich wsparcia (rozumianego jako wsparcie singletonu). Każdy przedmiot przy wpisaniu w drzewo otrzymuje wagę równą 1, natomiast przy wpisywaniu kolejnych transakcji zwiększa ona o 1, dla każdego zawartego w niej przedmiotu. Następnie algorytm przeszukuje poddrzewa zawarte w FP-tree i zwraca tabelę ze ścieżkami tych drzew, zawierającą częste zbiory. W ten sposób algorytm FP-Growth pomija w zupełności proces generowania kandydatów, jak i konieczność wielokrotnego skanowania całej tabeli.



Rys. 3 Drzewo reguł dla Tabeli 2, skonstruowane poprzez wpisanie w nie transakcji, posortowanych malejąco względem wsparcia przedmiotów.

Wpisanie pierwszej posortowanej transakcji w drzewo ogranicza się do stworzenia gałęzi, w której każdy przedmiot wpisany jest początkowo z wagą 1. Kolejne transakcje wpisywane są nakładając je (jeśli to możliwe) na istniejące gałęzie, przy jednoczesnym zwiększeniu wagi każdego z przedmiotu występującego w transakcji.

Przykładowo: pierwsza wpisana transakcja $\{I1, I2\}$ tworzy pierwszą gałąź, druga transakcja $\{I3, I2\}$ również tworzy nową gałąź, natomiast transakcja $\{I1, I3, I5, I4\}$ pozwala „przejść" przez przedmiot $I1$, podnosząc jego wagę o 1, a następnie tworzy ona kolejną gałąź. Po wpisaniu w drzewo pozostałych transakcji, otrzymujemy rezultat jak powyżej.

Następnie tworzona jest baza warunkowa, zawierająca wszystkie możliwe ścieżki dotarcia do danego przedmiotu, wraz z wagą, którą ten przedmiot otrzymuje w wyniku przejścia przez daną ścieżkę.

Przykładowo, ścieżki prowadzące do przedmiotu $I5$, to: $\{I1, I3\}$ z wagą 2, oraz $\{I1, I3, I2\}$ z wagą 1.

Przedmiot	Baza warunkowa
$I4$	$\{I1, I3, I5: 1\}$
$I2$	$\{I1: 1\}, \{I3: 1\}, \{I1, I3: 1\}$
$I5$	$\{I1, I3: 2\}, \{I1, I3, I2: 1\}$
$I3$	$\{I1: 3\}$
$I1$	-

Tab. 3

Następnie wszystkie podzbiory zbiorów z bazy warunkowej umieszczane są w drzewie warunkowym z wagą odpowiadającą sumie wag zbiorów, których są podzbiorami.

Przykładowo, dla przedmiotu $I5$, podzbiór $\{I1, I3\}$ występuje w bazie warunkowej w zbiorach $\{I1, I3\}$ oraz $\{I1, I3, I2\}$, więc wpisany zostanie do drzewa warunkowego z wagą $2 + 1 = 3$.

Przedmiot	Baza warunkowa	Drzewo warunkowe
$I4$	$\{I1, I3, I5: 1\}$	$\{I1: 1\}, \{I3: 1\}, \{I5: 1\}, \{I1, I3: 1\}, \{I1, I5: 1\}, \{I3, I5: 1\},$ $\{I1, I3, I5: 1\}$
$I2$	$\{I1: 1\}, \{I3: 1\}, \{I1, I3: 1\}$	$\{I1: 2\}, \{I3: 2\}, \{I1, I3: 1\}$
$I5$	$\{I1, I3: 2\}, \{I1, I3, I2: 1\}$	$\{I1: 3\}, \{I2: 1\}, \{I3: 3\}, \{I1, I3: 3\}, \{I3, I2: 1\}, \{I1, I2: 1\}$
$I3$	$\{I1: 3\}$	$\{I1: 3\}$
$I1$	-	

Tab. 4

Częste zbiory otrzymywane są w wyniku dołączenia odpowiedniego przedmiotu do zbiorów z drzewa warunkowego, których waga (równoważna wsparciu zbioru) jest większa od wybranego wsparcia minimalnego.

Przedmiot	Drzewo warunkowe	Częste zbiory
$I4$	$\{I1:1\},\{I3:1\},\{I5:1\},\{I1,I3:1\},\{I1,I5:1\},\{I3,I5:1\},\{I1,I3,I5:1\}$	-
$I2$	$\{I1:2\},\{I3:2\},\{I1,I3:1\}$	$\{I1,I2:2\},\{I3,I2:2\}$
$I5$	$\{I1:3\},\{I2:1\},\{I3:3\},\{I1,I3:3\},\{I3,I2:1\},\{I1,I2:1\}$	$\{I1,I5:3\},\{I3,I5:3\},$ $\{I1,I3,I5:3\}$
$I3$	$\{I1:3\}$	$\{I1,I3:3\}$
$I1$	-	-

Tab. 5

W rezultacie otrzymywane są zawsze te same zbiory częste, co w wyniku przeprowadzenia algorytmu Apriori.

Algorytm EClaT

Algorytm EClaT (*ang. Equivalence class transformation*) znajduje częste zbiory przedmiotów poprzez zmianę formatu tabeli na pionowy. Wszystkie transakcje zawierające dany przedmiot są grupowane w jeden rekord, a ich liczba jest wprost z definicji równa wsparciu tego przedmiotu. Częste zbiory $(k+1)$-elementowe generowane są przez przecięcie transakcji odpowiednich podzbiorów. W wyniku wcześniejszego pomijania bądź usuwania nieczęstych zbiorów, algorytm działa na coraz to mniejszej tabeli, co znacząco zwiększa jego efektywność.

ID Transakcji	Lista przedmiotów
$T100$	$I1,I2$
$T200$	$I2,I3$
$T300$	$I1,I3,I4,I5$
$T400$	$I1,I3,I5$
$T500$	$I1,I2,I3,I5$

Tab. 6

Następuje zmiana formatu tabeli na pionowy:

Przedmiot	**Lista ID transakcji**
$I1$	$T100, T300, T400, T500$
$I2$	$T100, T200, T500$
$I3$	$T200, T300, T400, T500$
$I4$	$T300$
$I5$	$T300, T400, T500$

Tab. 7

Zbiory $(k+1)$-elementowe generowane są poprzez przecięcie zbiorów transakcji dla odpowiednich przedmiotów, z pominięciem zbiorów, których liczba transakcji nie spełnia warunku częstości (jak poprzednio przyjmujemy $minS = 2$):

Przedmiot	**Lista ID transakcji**
$\{I1, I2\}$	$T100, T500$
$\{I1, I3\}$	$T300, T400, T500$
$\{I1, I5\}$	$T300, T400, T500$
$\{I2, I3\}$	$T200, T500$
$\{I3, I5\}$	$T300, T400, T500$
$\{I1, I3, I5\}$	$T300, T400, T500$

Tab. 8

Warto odnotować, że zmiana formatu tabeli wraz z wyszukiwaniem przecięć zbiorów w coraz to mniejszych tabelach wpływa pozytywnie na pamięciochłonność, jednak w przypadku tabeli transakcji zawierającej zbiory o stosunkowo dużym wsparciu, proces znajdowania ich przecięć znacząco ogranicza efektywność tego algorytmu.

Podsumowanie

Reguły asocjacyjne są niezwykle przydatne w dziedzinach takich jak marketing, gdyż umożliwiają podejmowanie skutecznych decyzji na podstawie wzorców odkrytych w dużych zbiorach danych. Warto zauważyć jednak, że przykład tworzenia rekomendacji ukazuje też ich ograniczenia. Reguły asocjacyjne dają nam tylko poglądowe informacje, o charakterze czysto ilościowym. W przypadku, gdy kluczowe są odpowiednie cechy przedmiotów, jak na przykład gatunki filmów czy utworów, reguły asocjacyjne mogą wykazać zależności pomiędzy nimi, jednak dla pełniejszego obrazu zachowań konsumentów potrzebne jest wsparcie innych algorytmów, nierzadko opartych o uczenie maszynowe.

Literatura

[1] Agrawal R., Imielinski T., Swami A., Mining Association Rules between sets of items in large databases, IBM Almaden Research Center, 1993
[2] Ogihara M., Zaki MJ., Theoretical Foundations of Association Rules, University of Rochester, 2004
[3] Sawicki A., Reguły asocjacyjne w bioinformatyce – zastosowanie wybranych technik, Politechnika Warszawska, 2015
[4] Chee CH., Jaafar J., Aziz IA., Hasan MH., Yeoh W., Algorithms for frequent itemset mining: a literature review, Artif Intell Rev 24, 2018

MathUp
Konferencja Zastosowań Matematyki

TIME–FREQUENCY FOURIER METHODS FOR LIVE AUDIO NOISE REDUCTION

Eldar MUKHTAROV
Politechnika Łódzka, Łódź

Introduction

In today's digital world, where we rely mostly on video calls, voice assistants, and streaming music, having clear and understandable audio is essential. However, background noise (e.g., conversations or mechanical sounds) often makes it hard to hear the main sound clearly. Hence, the digital signal processing provides a range of methods to tackle this issue, where Fourier analysis is one of the most important methods in the resolution.

The Fourier Transform, in its different forms, is a mathematical tool to decompose a signal from its time-domain representation into its constituent frequencies in the frequency domain [1]. This transformation allows the separation of desired signal components from noise components, which may have distinct spectral characteristics, hence the usage in broad signal processing tasks.

This chapter introduces the theoretical foundations of Fourier Transforms and their practical application in audio denoising. We begin with the continuous Fourier Transform, then its adaptations for digital signals, namely, the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT). Despite their power and broad use, they are best suited for stationary signals, where statistical properties do not change over time. Real-world audio, particularly speech mixed with environmental noise, is often non-stationary. To address this, we introduce the Short-Time Fourier Transform (STFT), which enables time-frequency analysis that involves how the spectral content of a signal evolves [2].

Next, we focus on a common STFT-based audio denoising pipeline. This involves segmenting the noisy audio, applying a window function (e.g., the Hamming window) before STFT, estimating the noise characteristics in the frequency domain, creating and applying a threshold or gain function to suppress noise, and, finally, reconstructing the cleaner audio signal using the Inverse STFT (ISTFT). The effectiveness of the denoising process is evaluated using both subjective listening tests and objective metrics such as Signal-to-Noise Ratio (SNR).

Fundamentals of Fourier Analysis for Audio Signals

The Fourier Transform (FT)

The Fourier Transform, developed by Jean-Baptiste Joseph Fourier, states that any continuous and periodic signal can be represented as a sum of sine and cosine waves of different frequencies and amplitudes. For a non-periodic, continuous-time signal $x(t)$, its Fourier Transform $X(f)$ is given by:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi f t}\, dt$$

And the inverse transform, which converts a signal from the frequency domain back to the time domain, is:

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{j2\pi f t}\, df$$

where j is the imaginary unit, t is time, and f is frequency [3]. $X(f)$ is a complex function representing the magnitude and phase of each frequency component in $x(t)$. For audio signals, the magnitude spectrum $|X(f)|$ reveals the intensity of different frequencies, which is necessary for identifying tonal components, formants in speech, and the spectral signature of noise.

The Discrete Fourier Transform (DFT)

In digital audio processing, signals are discrete in time and finite in duration. The Discrete Fourier Transform (DFT) is the analog of the FT for such signals. For a sequence $x[n]$ of N samples, its DFT $X[k]$ is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi k \frac{n}{N}} \qquad \text{for } k = 0, 1, \ldots, N-1$$

The inverse DFT (IDFT) is:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi k \frac{n}{N}} \qquad \text{for } n = 0, 1, \ldots, N-1$$

Here, n denotes the sample index in the time domain, indicating the position of a sample within the sequence of N points. In contrast, k denotes the frequency bin index, which corresponds to a discrete frequency component. Each bin k represents a sinusoid that completes k cycles over the N-point interval. The DFT thus provides N complex values representing the signal's content at N discrete frequency bins [4]. Although the DFT was a revolution in signal processing, its direct computation requires on the order of N^2 operations, which can be computationally expensive for large N (i.e., long audio segments).

The Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) is not a new transform but a family of highly efficient algorithms for computing the DFT. Most commonly, FFT algorithms reduce the computational complexity from $O(N^2)$ to $O(NlogN)$ [5]. This reduction in computation made real-time spectral analysis and complex audio processing more practical. The FFT produces the same result as the DFT but does so much faster, using Divide-and-conquer algorithm, that made it ubiquitous in digital audio workstations, analysis software, and embedded audio devices.

Time-Frequency Analysis for Non-Stationary Audio

Stationary vs. Non-Stationary Signals

A stationary signal's frequency content is consistent throughout its duration. Some examples might include steady sound of a fan or a long and static beep sound. However, many real-world audio signals, such as speech, music, and most environmental noises (e.g., cafeteria background, traffic sounds), are non-stationary – frequency content changes over time [6]. Analyzing such signals with a single FFT across their entire duration would average out these temporal variations which could lose important information about when specific frequency events occur.

The Short-Time Fourier Transform (STFT)

To analyze non-stationary signals, we need a method that captures how the frequency content changes over time. The Short-Time Fourier Transform (STFT) achieves this by performing DFTs on short, overlapping segments of the signal [2].
This process involves:

1. **Segmentation**: The input signal $x[n]$ is divided into shorter, overlapping frames, to make sure that transitions between frames are smooth enough.
2. **Windowing**: Each frame is multiplied by a window function $w[n]$ (e.g., Hamming, Hann, Blackman). Windowing tapers the frame at its edges, which reduces the spectral leakage that could arise from the abrupt truncation of the signal at the boundaries of that frame. The Hamming window, for example, is commonly used and is defined as:

$$w[n] = 0.54 - 0.46\,cos\left(\frac{2\pi n}{M-1}\right) \qquad \text{for } 0 \le n < M$$

   where M is the window length [7].
3. **DFT Computation**: An FFT is computed for each windowed frame.

The STFT $X(m,k)$ of a signal $x[n]$ is given by:

$$X(m,k) = \sum_{n=0}^{N-1} x[n]w[n-mH]e^{-j2\pi k\frac{n}{N}} \qquad k = 0, 1, \dots, N-1,$$

where m is the frame index, H is the hop size (number of samples between the start of consecutive frames), N is the FFT length, and k is the frequency bin index. The result is a two-dimensional representation of the signal's energy, distributed over time (frame index m) and frequency (bin index k).
The application of STFT involves a trade-off in the choice of window length and hop size: shorter windows give better time resolution but poorer frequency resolution, while longer windows give better frequency resolution at the cost of time resolution [8].

Practical Application of Fourier Transforms in Audio Denoising

One of the applications of STFT in audio processing is noise reduction, where the goal is to suppress unwanted noise and still preserve the essence of the signal (e.g., speech). In this section, the scenario, workflow, and an evaluation of a real-time audio denoising project will be demonstrated and explained.

The Denoising Problem

The demonstration started with capturing a voice in real-time using a microphone. To simulate a noisy environment, pre-recorded cafeteria noise was combined with the clean audio. This composite noisy audio then served as the input for the real-time denoising algorithm. The objective was to attenuate the cafeteria noise while preserving the intelligibility and quality of the original speech.

STFT-Based Noise Reduction Workflow

The denoising process, illustrated in Figure 1, followed these steps:

1. *Transformation to Time-Frequency Domain (STFT)*

The noisy audio signal $y[n] = s[n] + d[n]$ (where $s[n]$ is the clean signal and $d[n]$ is the noise) is transformed into its time-frequency representation $Y(m,k)$ using the STFT with a window function called the Hamming window:

$$Y(m,k) = STFTy[n].$$

2. *Noise Estimation and Threshold Creation*

Noise reduction was done in time-frequency domain with the following steps:

- **Adaptive Noise Estimation**: Because the cafeteria noise changed over time (i.e., it was non-stationary), our algorithm used a method to keep updating the noise estimation adaptively. This was employed as a way to adjust to new sounds in the background as they appear [9].

- **Spectral Attenuation**: After identifying which parts of the signal were likely noisy, a gain function or a threshold was created and applied to attenuate noisy frequency components. The aggressiveness of this attenuation was controlled by the set parameters of the algorithm, which influenced how much of the estimated noise is removed [10, 11].
- **Phase Preservation**: Despite the algorithm modifying the magnitude of the signal to reduce noise, the phase of the original noisy signal was preserved. This helped to keep the natural quality of the original audio. The final clean speech signal $\hat{S}(m,k)$ was then reconstructed using the adjusted magnitude and the original phase.

3. *Transformation back to Time Domain (ISTFT)*

The denoised time-frequency representation $\hat{S}(m,k)$ was then transformed back to the time domain using the Inverse Short-Time Fourier Transform (ISTFT) to obtain the denoised audio signal $\hat{s}[n]$. The ISTFT involves performing an inverse DFT on each frame and then combining these frames using an overlap-add synthesis method to reconstruct the signal [8].

$$\hat{s}[n] = ISTFT\ \hat{S}(m,k)$$

Fig. 1 Workflow Diagram of the Real-Time Audio Denoising Process (created by the author).

Evaluating Denoising Performance

To see how effective the noise reduction algorithm was, both subjective and objective evaluations were employed.
During the live demonstration, the improvement was assessed by listening. Listeners noticed a clear reduction in the background cafeteria noise that made the speech easier to understand.
To support the subjective observations, the Signal-to-Noise Ratio (SNR) was calculated at a specific point during the demo:

- **SNR before denoising**: When the cafeteria noise was added to the original signal, the SNR dropped to 18.34 dB.
- **SNR after denoising**: After applying the noise reduction algorithm, the SNR was 5.80 dB.
- **SNR change**: The difference, −12.54 dB (5.80 dB − 18.34 dB), indicates that the SNR decreased after denoising.

Although a negative change in SNR might seem counterintuitive, it suggests that, from a strict mathematical standpoint, the overall signal quality decreased. This could be due to the algorithm unintentionally modifying parts of the speech signal or introducing small artifacts, even though the loud background noise was reduced successfully. This further underlines the fact that objective methods like SNR do not always match with human perception. While listeners may hear clearer speech, the algorithm's changes can still negatively affect the numerical quality score. Therefore, a

full evaluation of audio quality should consider both quantitative metrics and how the audio actually sounds to people [12].

Additionally, comparing the spectrograms of the noisy and denoised audio, can provide further information visually. Figure 2 illustrates these changes, showing attenuation in frequency bands dominated by noise and preservation of speech formants.

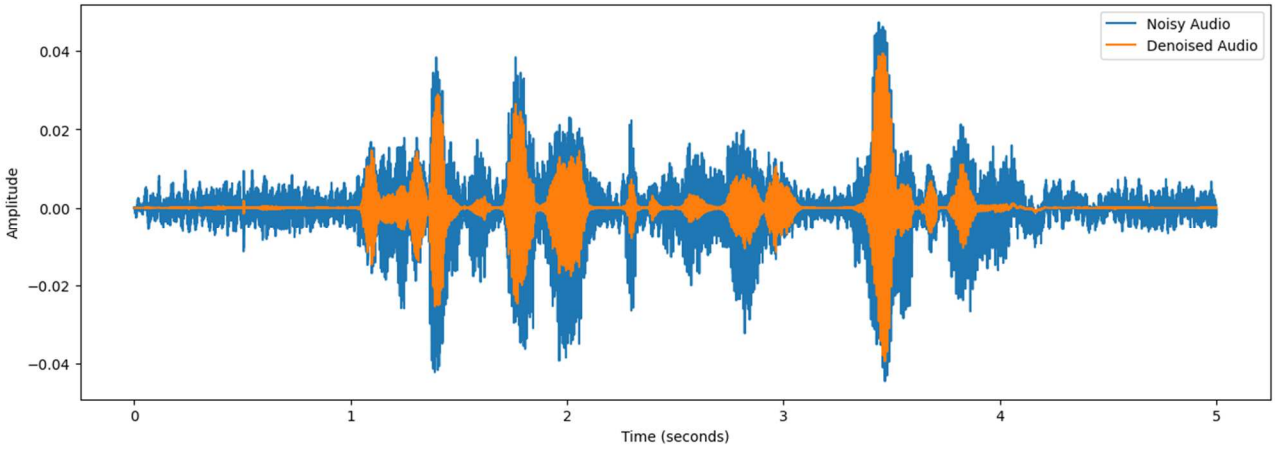


Fig. 2 Comparison of Noisy and Denoised Audio Waveforms (plotted from experimental results by the author).

Conclusion

Fourier Transforms, from the foundational FT and DFT to the efficient FFT and the time-frequency resolving STFT, are crucial in digital audio processing, especially in noise reduction. This chapter described their theoretical background and a practical application in one of its most common applications – noise reduction. In particular, the STFT, which allows the analysis of audio in the time-frequency domain, was fundamental in handling non-stationary noise effectively.

The denoising process, which involved – STFT transformation, noise estimation and threshold creation, and ISTFT reconstruction – was one of the simplistic and efficient approaches. Various algorithms for noise estimation and gain application can be utilized that differ in complexity (e.g., simple spectral subtraction, Wiener filtering, or even machine learning-based approaches). Nonetheless, the Fourier analysis to move between time and frequency domains remains as the core of the project.

As these methods evolve, they show promising results and improvements in our ability to extract clarity from noise, which leads to better communication, entertainment, and information retrieval across a multitude of audio-driven applications.

References

[1] Oppenheim A. V., Schafer R. W., Discrete-Time Signal Processing (3rd ed.), 2009
[2] Allen J. B., Rabiner L. R.,A Unified Approach to Short-Time Fourier Analysis and Synthesis, Proceedings of the IEEE, vol. 65, no. 11, pp. 1558-1564, 1977
[3] Bracewell R. N., The Fourier Transform and Its Applications (3rd ed.), 2007
[4] Smith J. O., Mathematics of the Discrete Fourier Transform (DFT), Identity, pp. 1-247, 2007, [Accessed: May 27, 2025]
[5] Cooley J. W., Tukey J. W., An Algorithm for the Machine Calculation of Complex Fourier Series, 1965
[6] Huang N. E. *et al.*, The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1971, pp. 903-995, 1998
[7] Harris F. J., On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceedings of the IEEE, vol. 66, no. 1, pp. 51-83, 1978
[8] Griffin D. W., Lim J. S., Signal Estimation from Modified Short-Time Fourier Transform, IEEE Trans Acoust, vol. 32, no. 2, pp. 236-243, 1984
[9] Hirsch H. G., Ehrlicher C., Noise estimation techniques for robust speech recognition, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 1, pp. 153-156, 1995
[10] Vaseghi S. V., Advanced Digital Signal Processing and Noise Reduction: Fourth Edition, Advanced Digital Signal Processing and Noise Reduction: Fourth Edition, pp. 1-514, Jan. 2008
[11] Boll S. F., Suppression of Acoustic Noise in Speech Using Spectral Subtraction, IEEE Trans Acoust, vol. 27, no. 2, pp. 113-120, 1979
[12] ITU-T Recommendation P.862, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, www.itu.int, [Accessed: May 27, 2025]

MathUp
Konferencja Zastosowań Matematyki

DNA as the Hard Drive of The Future – and the Math That Makes It Possible.

Adrianna CZECHOWSKA
Politechnika Łódzka, Łódź

Introduction

This article explores the mathematical foundations that enable DNA to function as high-density data storage and explores its theoretical ideal storage capabilities. It begins with DNA structure and binary encoding concepts, explaining key principles such as base conversion theory, ternary encoding, and Shannon entropy, and ends with real-world implementations and future implications. It addresses theoretical questions such as: How do numeral system conversions transform digital data into biological sequences? Why do ternary approaches succeed where quaternary encoding fails?

DNA Structure and Biological Foundations

Deoxyribonucleic acid, known as DNA, is the genetic material of all living organisms. It is made of a double helix that consists of long chains of monomer nucleotides. Nucleotides consist of a sugar backbone, phosphate groups as well as nitrogenous base pairs, Adenine (A) pairs with Thymine (T) and Guanine (G) pairs with Cytosine (C). The structural integrity of DNA relies on covalent phosphodiester bonds that link adjacent nucleotides, creating a stable sugar-phosphate backbone from which the nitrogenous bases extend [1]. The two complementary strands are held together through specific hydrogen bonding patterns - adenine exclusively bonds with thymine, while cytosine pairs only with guanine, ensuring the precise base-pairing rules that maintain molecular stability across biological systems.

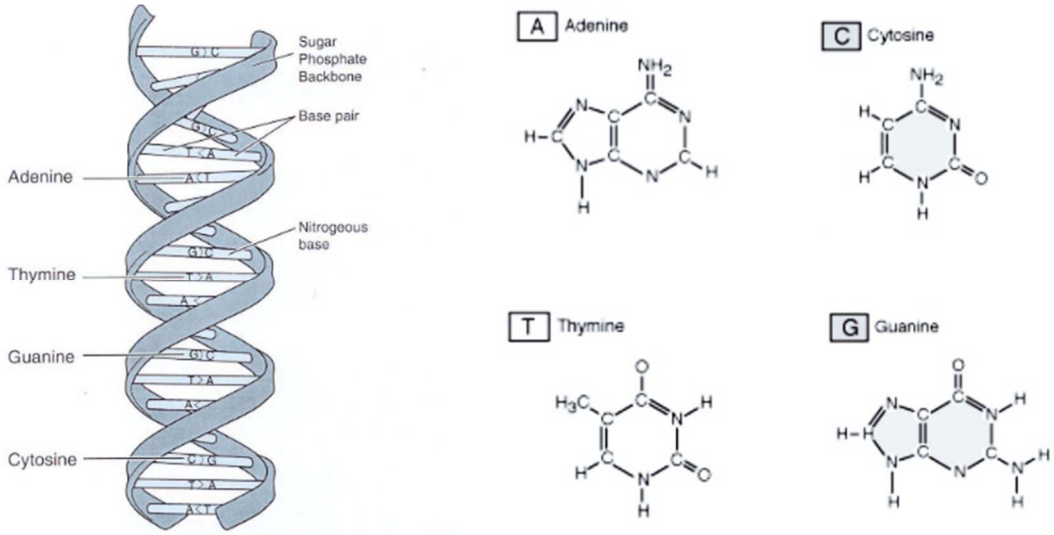
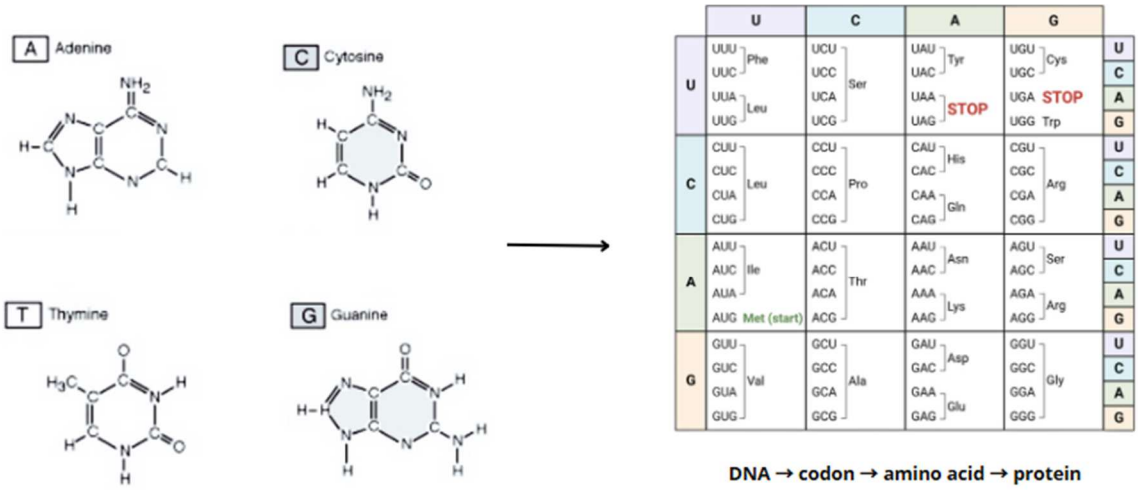


Fig. 1 Structure and Components of DNA [2,3].

These four bases are organized into functional units called codons, which are triplets of nucleotides that serve as the basic coding units of genetic information. As shown in the genetic code table (Figure 2), each three-base combination corresponds to a specific amino acid or stop signal in protein synthesis. For example, the codon UUU codes for the amino acid phenylalanine, while UAG acts as a stop signal that ends protein production. With four bases arranged in groups of three, DNA can create $4^3 = 64$ different combinations, which is enough to encode all 20 standard amino acids plus control signals [4]. These 20 types of amino acid can link in various combinations to create different proteins [5]. The process from DNA to functional proteins follows a clear sequence: DNA → codon → amino acid → protein.



	U	C	A	G	
U	UUU Phe, UUC Phe, UUA Leu, UUG Leu	UCU Ser, UCC Ser, UCA Ser, UCG Ser	UAU Tyr, UAC Tyr, UAA STOP, UAG STOP	UGU Cys, UGC Cys, UGA STOP, UGG Trp	U C A G
C	CUU Leu, CUC Leu, CUA Leu, CUG Leu	CCU Pro, CCC Pro, CCA Pro, CCG Pro	CAU His, CAC His, CAA Gln, CAG Gln	CGU Arg, CGC Arg, CGA Arg, CGG Arg	U C A G
A	AUU Ile, AUC Ile, AUA Ile, AUG Met (start)	ACU Thr, ACC Thr, ACA Thr, ACG Thr	AAU Asn, AAC Asn, AAA Lys, AAG Lys	AGU Ser, AGC Ser, AGA Arg, AGG Arg	U C A G
G	GUU Val, GUC Val, GUA Val, GUG Val	GCU Ala, GCC Ala, GCA Ala, GCG Ala	GAU Asp, GAC Asp, GAA Glu, GAG Glu	GGU Gly, GGC Gly, GGA Gly, GGG Gly	U C A G

Fig. 2 From DNA Bases to Proteins [3,6].

From Biological to Digital Information

All digital information is written in binary code, while biological information is stored in DNA. This fundamental difference creates both an opportunity and a challenge for using DNA as a digital storage medium.

Binary code is a positional numeral system in mathematics that uses base 2, with only two digits: 0 and 1. Each position represents a power of 2 (1, 2, 4, 8...), and binary code is the foundation of all digital systems [7]. All computer data, such as images, text, and videos, is stored as binary code using sequences of 0s and 1s.

In theory, since DNA has 4 bases (A, T, C, G), we could encode binary into base-4 (quaternary), but this might lead to unstable sequences with repeated bases like "AAAA" or "TTTT". The encoding approaches can be categorized into three main types: binary encoding, which maps DNA bases to binary pairs (e.g., 00 = A, 01 = C, 10 = G, 11 = T); quaternary encoding, which treats each DNA base as a base-4 digit (e.g., A = 0, C = 1, G = 2, T = 3); and ternary encoding, which maps each digit to one of three bases different from the previous one [7–9].

Encoding Type	Base	Digits / Symbols	DNA Mapping Logic
Binary Encoding	2	0, 1 (combined as pairs)	DNA bases are mapped to **binary pairs** (e.g. 00 = A, 01 = C, 10 = G, 11 = T)
Quaternary Encoding	4	0, 1, 2, 3	Each DNA base is treated as a **base-4 digit** (e.g. A = 0, C = 1, G = 2, T = 3)
Ternary Encoding	3	0, 1, 2	Each digit is mapped **to one of the 3 bases not equal to previous one** (no repeats)

Table 1 DNA Data Encoding Schemes: Binary, Quaternary, and Ternary.

Ternary encoding is used in practice to make DNA sequences more stable. This is appropriate because it prevents the same base from appearing twice in a row by selecting the next base from the three nucleotides not used previously.

For example, if the last base was A, the choices are C, G, T, with the ternary digits 0, 1, 2 corresponding in that order. If the previous base was C, the available choices become A, G, T, with the ternary digits once again corresponding to them. Since the previous base is always known during decoding, the mapping for each step is clear, so the same nucleotide can represent different digits without causing ambiguity.

The Stability Problem and Ternary Solutions

Ternary encoding used in this context is the first step of the DNA Fountain Method. It is used for converting digital information into stable DNA sequences through a multi-step encoding process that prevents repeated bases while maintaining reasonable storage density.

As illustrated in the encoding pipeline in Figure 3, the transformation begins with a binary file containing the original digital data. This file undergoes base-3 encoding, where the binary information is converted into ternary digits that can be systematically mapped to DNA bases while avoiding consecutive repeats.

The base-3 encoded data is then transformed into DNA-encoded sequences, where each ternary digit selects from the available nucleotides that differ from the previous base. This process creates DNA strands that maintain sequence stability while preserving the original information. However, rather than storing data in single continuous strands, the DNA Fountain method fragments these sequences into shorter, manageable pieces of approximately 25 base pairs each.

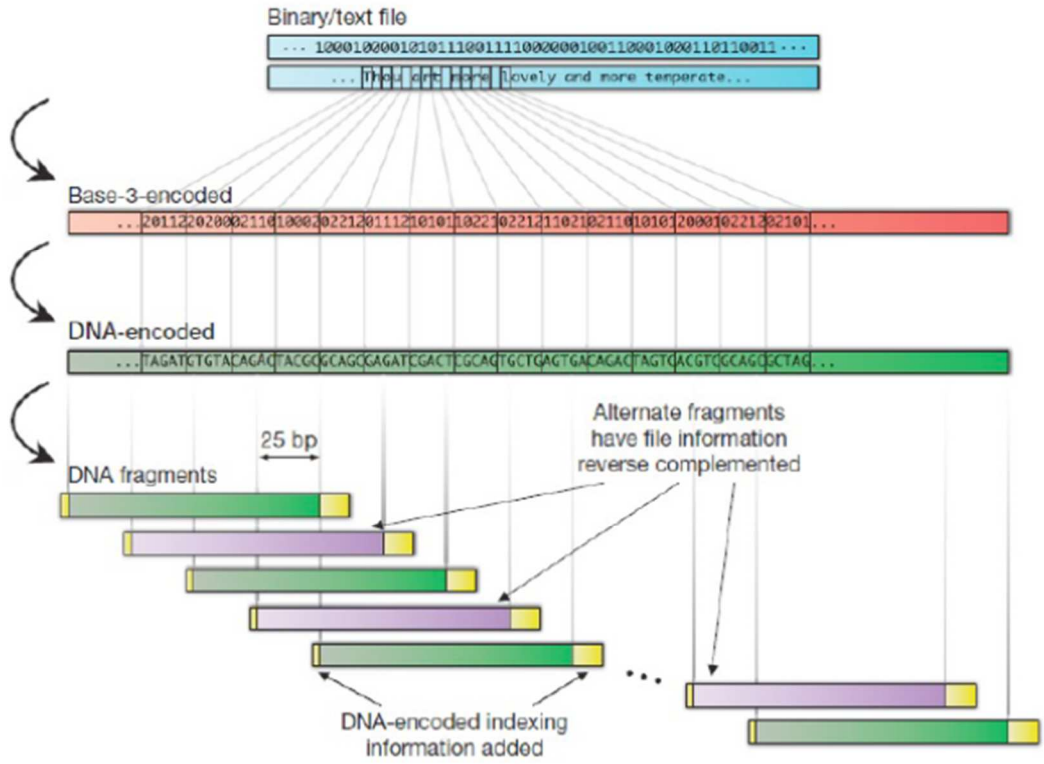


Fig. 3 Workflow of DNA-based data storage – Fountain method [8].

The fragmentation step serves multiple purposes: it prevents synthesis errors that accumulate in longer sequences, enables parallel processing during both writing and reading operations, and provides redundancy for error correction [8]. Each fragment includes DNA-encoded indexing information that allows for proper reconstruction of the original file. Additionally, alternate fragments are reverse complemented, creating additional redundancy that helps recover data even if some fragments are lost or corrupted during storage or retrieval. While this method demonstrates the practical feasibility of DNA storage, determining its theoretical efficiency requires a mathematical analysis of information density limits.

Shannon Entropy Analysis

To investigate the theoretical limits of DNA storage, we must first examine Shannon entropy – a concept from information theory that quantifies the maximum information content possible in any communication system [10]. Shannon entropy is defined by the formula [11]:

$$H = -\Sigma\, p(x)\, log_2\,(p(x)),$$

where:

- H – entropy (in bits),
- $p(x)$ – probability of symbol x,
- $log_2\,(p(x))$ – amount of information (in bits) associated with x,
- Σ – summation symbol indicating the sum runs over all possible symbols,
- x – represents each possible symbol (in DNA: A, T, C, G).

The formula allows to calculate the average information content per symbol based on the probability distribution of each symbol appearing. Thus, it shows entropy.

For DNA storage, using equal distribution of all four bases (A, T, C, G), the maximum entropy is 2 bits per nucleotide. This is calculated as:

$$H = -4 \cdot 0.25 \cdot log_2(0.25) = -4 \cdot 0.25 \cdot (-2) = 2 \text{ bits.}$$

This value represents the theoretical maximum information content.

Ternary encoding, used in Fountain method, which avoids consecutive identical bases for stability, reduces this to approximately 1.59 bits per nucleotide:

$$H = -3 \cdot 0.(3) \cdot log_2(0.(3)) = -3 \cdot 0.(3) \cdot (-1.59) \approx 1.59 \text{ bits.}$$

This illustrates how theory must be adjusted when biological constraints are taken into consideration. Nevertheless, using these entropy calculations, we can now determine physical data storage capacity of DNA.

Theoretical Storage Capacity

A nucleotide, the basic building block of DNA, weighs approximately 330 g/mol [12], which means that 1 gram of DNA contains approximately 0.00303 moles of nucleotides:

$$n = \frac{1g}{330g/mol} \approx 0.00303\, mol.$$

Using Avogadro's number ($6.022 \cdot 10^{23}$), we can calculate the number of nucleotides in one gram:

$$0.00303\ mol \cdot 6.022 \cdot 10^{23} = 1.83 \cdot 10^{21} \text{ nucleotides.}$$

In the DNA Fountain encoding method, each nucleotide carries approximately 1.59 bits of information (as calculated from ternary encoding entropy). Therefore, the theoretical storage capacity can be calculated as:

$$1.83 \cdot 10^{21} \text{ nucleotides} \cdot 1.59 \text{ bits} = 2.90 \cdot 10^{21} \text{ bits.}$$

Converting to exabytes (1 exabyte $= 10^{18}$ bytes $= 8 \cdot 10^{18}$ bits):

$$2.90 \cdot 10^{21} \text{ bits} \div 8 \cdot 10^{18} \text{ bits} \approx 363 \text{ exabytes per gram.}$$

Conclusions and discussion

The theoretical number of bits which can be stored in each nucleotide is 2 bits. The practical limit (Shannon limit) is 1.83 bits, while the DNA Fountain method achieves 1.59 bits per nucleotide, which is approximately 85% of the practical limit. Assuming the theoretical maximum of 2 bits per nucleotide, the storage density of DNA would be about 4455 exabytes per gram. Using the Fountain method, this value decreases to about 363 exabytes per gram.

Experimentally, the DNA Fountain experiment published in Science in 2017 achieved 214 petabytes per gram [13]. This corresponds to only about 0.05% of the theoretical maximum density (455 exabytes per gram) and approximately 0.06% of the Fountain method density (363 exabytes per gram). To put this into perspective, 214 petabytes equal 214 million gigabytes, which can store 71 million hours of HD video, or about 8 thousand years of continuous footage, all contained within a single gram of DNA.

This striking disparity between theoretical potential and current achievements highlights significant opportunities for future technological development. As synthesis accuracy, error correction, and encoding efficiency continue to improve, DNA storage may become the hard drive of the future, offering astounding data storage density and demonstrating how mathematical foundations drive technological innovation.

Literature

[1] The Editors of Encyclopedia Britannica. DNA. Encyclopedia Britannica, https://www.britannica.com/science/DNA [Accessed: May, 2025]
[2] EBSCO. *Gene*. EBSCO Research Starters; n.d., https://www.ebsco.com/research-starters/health-and-medicine/gene [Accessed: May 31, 2025].
[3] Zhang W., Wang M., Cranford S., Ranking of molecular biomarker interaction with targeted DNA nucleobases via full atomistic molecular dynamics, Scientific Reports, 2016
[4] Human Genome Research Institute. Codon. Genome.gov; n.d., https://www.genome.gov/genetics-glossary/Codon [Accessed: May 31, 2025]
[5] U.S. National Library of Medicine. How do genes direct the production of proteins? MedlinePlus; n.d., https://medlineplus.gov/genetics/understanding/howgeneswork/protein/#:~:text=There%20are%2020%20different%20types,by%20the%20sequence%20of%20genes [Accessed: May 31, 2025]
[6] Monash University. Transcription and translation. Student Academic Success; n.d., https://www.monash.edu/student-academic-success/biology/nucleic-acids-and-proteins/transcription-and-translation [Accessed: May 31, 2025]
[7] The Editors of Encyclopedia Britannica. Binary code. Encyclopedia Britannica; n.d., https://www.britannica.com/technology/binary-code [Accessed: May 31, 2025]
[8] Limbachiya D., Dhameliya V., Khakhar M., Gupta, M., On optimal family of codes for archival DNA storage. 2015 International Workshop on Signal Design and Its Applications in Communications (IWSDA), pp 90-94, 2015
[9] Choi Y., Ryu T., Lee A. C., Choi H., Lee H., Park J., Song S.-H., Kim S., Kim H., Park W., Kwon S., High information capacity DNA-based data storage with augmented encoding characters using degenerate bases. Scientific Reports, 2019
[10] Drake G.W., Entropy. Encyclopedia Britannica, 2025, https://www.britannica.com/science/entropy-physics [Accessed: May 31, 2025]
[11] Shannon C. E., A mathematical theory of communication. The Bell System Technical Journal, 27(3), pp 379-423, 1948
[12] Thermo Fisher Scientific. DNA and RNA molecular weights and conversions. Thermo Fisher Scientific; n.d., https://www.thermofisher.com/pl/en/home/references/ambion-tech-support/rna-tools-and-calculators/dna-and-rna-molecular-weights-and-conversions.html [Accessed: May 31, 2025]
[13] Erlich Y., Zielinski D., DNA Fountain enables a robust and efficient storage architecture. Science, 355(6328), pp 950-954, 2017

MathUp
Konferencja Zastosowań Matematyki

JAK tu nie zWARIOWAĆ?

Piotr ŁUKAWSKI[1], Michał SZYC[2], Jakub ZDANOWSKI[1]

[1] Uniwersytet Warszawski, Warszawa
[2] Politechnika Warszawska, Warszawa

Wstęp

W poszukiwaniu najlepszych rozwiązań rzeczywistych problemów niezmiennie odpowiedzi dostarcza nam matematyka – mimo swej abstrakcyjnej natury. Z tego powodu znajdowanie najkrótszej drogi promienia świetlnego, szkolenie sieci neuronowych, czy optymalizacja elementów konstrukcyjnych, mają wspólny aspekt, jakim jest wykorzystanie odpowiedniego narzędzia matematycznego. Mowa jest o rachunku wariacyjnym, który polega na analizowaniu wszystkich możliwych funkcji podejrzanych o bycie rozwiązaniami danego problemu, a następnie wybór tej z nich o optymalnych parametrach. Jak jednak tego dokonać? Jak tu nie zwariować przy znalezieniu wszystkich możliwych rozwiązań i wyłuskaniu wśród nich tego najlepszego? W tym artykule podejmujemy się próby wytłumaczenia aparatu matematycznego stojącego za rachunkiem wariacyjnym jak i również przedstawienia jego możliwości praktycznych.

Wprowadzenie Matematyczne

Ponieważ interesują nas praktyczne zastosowania przedstawianych derywacji matematycznych, to należnym jest, aby wyprowadzenia te były przedstawione z pewnym praktycznym przykładem w tle. W związku z tym, przed sformalizowaniem problemu zagadnienia wariacyjnego, zarysujemy przykładowy problem.

Pewien góral chce dostać się do składziku oscypków, jednak w nocy śnieg zasypał mu podwórze sposób tworząc nieregularne, wysokie śnieżne hałdy. Jak góral powinien przekopywać się przez śnieg, żeby musieć odgarnąć go jak najmniej?

Matematyczny problem z tą sytuacją związany [1], [2] możemy opisać jako problem wyznaczenia ścieżki, która przejdzie od punktu a do punktu b tak, że ilość śniegu potrzebnego do odgarnięcia na tej drodze będzie najmniejsza.

Poszukujemy więc gładkiej krzywej $\gamma(v) = (v, f(v))$ w R^{k+1}, gdzie parametryzacja zadana będzie przez funkcję $f: R \to R^k$ klasy C^n na odcinku $[a, b]$.

Wiemy, że dana trajektoria ma być optymalna dla założonego problemu, to oznacza, że z wszystkich możliwych ścieżek sparametryzowanych przez różne funkcje f szukamy takiej, która będzie minimalizować pewien zadany funkcjonał. Będzie interesować nas klasa funkcjonałów lokalnych zwracających wartości liczbowe, tzn.

$$J: C^n([a, b]; R^k) \ni f \mapsto J[f] \in R.$$

Taki funkcjonał będzie zadany jako funkcjonał całkowy postaci:

$$J[f] = \int_a^b F\big(v, f(v), f'(v), \dots, f^{(n)}(v)\big)dv.$$

Wówczas Fdv będzie przyczynkiem do minimalizowanego funkcjonału. Trzymając cały czas w pamięci przykład górala będziemy całkować przyczynki do odgarnianego śniegu na drodze od a do b i sprawdzać, ile takiego śniegu jest do odgarnięcia na całej ścieżce – ten wynik, to wartość funkcjonału.

W praktyce, dla przypadków fizycznych w trójwymiarowej przestrzeni funkcjonał przybiera formę, w której funkcja F zależy jedynie od v, funkcji f oraz pierwszej pochodnej tej funkcji z powodów, które staną się jasne w dalszym rozumowaniu. Ponadto funkcja F musi być podwójnie różniczkowalana ze względu na swoje argumenty:

$$C^2 \ni F: R \times R^k \times R^k \mapsto R.$$

Tak więc funkcjonał sprowadza się do formy:

$$J[f] = \int_a^b F\big(v, f(v), f'(v)\big)dv.$$

Wracając teraz do przykładu odśnieżania, chcemy znaleźć ścieżkę, dla której ilość odgarniętego śniegu jest najmniejsza, czyli chcemy wyznaczyć minimum funkcjonału J. Funkcjonał jest odwzorowaniem o charakterze ciągłym i różniczkowalnym w sensie różniczkowalności w przestrzeni funkcji gładkich. Aby znaleźć jego ekstrema, musimy znaleźć jego pochodną w sensie funkcyjnym.

W tym celu zwróćmy uwagę na przyrost funkcjonału (ΔJ), czyli różnicę funkcjonału policzonego dla starej ścieżki f oraz dla nowej ściężki ($f + h$), jedynie niewiele różniącej się od poprzedniej:

$$\Delta J[f] = J[f + h] - J[f].$$

Funkcja $h(v)$ nie zmienia znacznie krzywej zadanej przez funkcję $f(v)$; powyższą różnicę nazywamy wariacją funkcji f. Dodatkowo w punktach a i b wariacja funkcji f musi być zerowa, bo punkty początku i końca nie mogą się zmienić. To oznacza, że $h(a) = h(b) = 0$.

Następnie chcemy znaleźć liniową część przyrostu funkcjonału $\Delta J[f]$ oraz sprawdzić, że norma nieliniowej reszty $R(h)$ z tego przyrostu będzie spełniać warunki zanikania przy podzieleniu przez normę z funkcji h w granicy z normą wariacji dążącą do zera, czyli:

$$\Delta J[f] = \delta_f \cdot \|h\| + R(h), \qquad \lim_{\|h\| \to 0} \frac{|R|}{\|h\|} = 0.$$

Wybieramy tutaj normę supremum na przestrzeni funkcji ciągłych na odcinku $[a, b]$ [8]:

$$\|g\|_{C^1([a,b],R^k)} = \sum_{i=0}^{n} \sup_{t \in [a,b]} \left|g^{(i)}(t)\right|_k,$$

gdzie $|\cdot|_k$ oznacza normę euklidesową na przestrzeni R^k.
Wówczas norma w interesującym nas problemie to:

$$\|f\|_{C^1([a,b],R^k)} = \sup_{t\in[a,b]} |f(t)|_k + \sup_{t\in[a,b]} |f'(t)|_k.$$

Teraz rozważmy przyrost funkcjonału z jego definicji:

$$\Delta J[f] = \int_a^b F\big(v, f(v)+h(v), f'(v)+h'(v)\big)\,dv - \int_a^b F\big(v, f(v), f'(v)\big)\,dv.$$

Dalej użyjmy wzoru Taylora, żeby rozwinąć funkcję podcałkową F wokół punktu (v, f, f') w ten sposób, że w pierwszym argumencie nie odsuniemy się wcale, w drugim odsuniemy się o h, a w trzecim o h' (możemy tego dokonać ze względu na własność $F \in C^2$):

$$F(v+0, f+h, f'+h') = F(v,f,f') + 0\cdot\frac{\partial F}{\partial v}\Big|_{(0,h,h')} + h\cdot\frac{\partial F}{\partial f}\Big|_{(0,h,h')} + h'\cdot\frac{\partial F}{\partial f'}\Big|_{(0,h,h')} + r_2(h,h'),$$

gdzie wyrażenie oznacza resztę drugiego rzędu w h, h'.

Wówczas przyrost funkcjonału sprowadza się do postaci:

$$\Delta J[f] = \int_a^b \left(\frac{\partial F}{\partial f}\cdot h + \frac{\partial F}{\partial f'}\cdot h' + r_2(h,h')\right) dv.$$

Wykorzystamy teraz regułę Leibniza o pochodnej iloczynu funkcji:

$$\Delta J[f] = \int_a^b \left(\frac{\partial F}{\partial f}\cdot h + \frac{d}{dv}\left(\frac{\partial F}{\partial f'}\cdot h\right) - \frac{d}{dv}\left(\frac{\partial F}{\partial f'}\right)h + r_2(h,h')\right) dv.$$

Dodatkowo na krańcach odcinka $[a,b]$ wariacja zanikała, więc drugi wyraz pod całką wyzeruje się po scałkowaniu. Całka zachowuje liniowość, więc dostajemy wzór, w którym możemy odnaleźć liniowy przyrost funkcjonału J przy wariacji h:

$$\Delta J[f] = \delta_f\cdot\|h\| + R = \int_a^b \left[\frac{\partial F}{\partial f} - \frac{d}{dv}\left(\frac{\partial F}{\partial f'}\right)\right] h\;dv + \int_a^b r_2(h,h')\;dv.$$

Sprawdźmy teraz, że iloraz normy reszty i normy funkcji h spełnia warunek znikania przy normie wariacji dążącej do zera. Z własności normy C^1 mamy:

$$\forall t \in [a,b]: |h_1(t), \dots, h_k(t), h_1'(t), \dots, h_k(t)|^2 \le 2k\|h\|_{C^1}^2.$$

Wówczas możemy ograniczyć z góry resztę R:

$$|R| \le \sqrt{2k}\int_a^b \left|\frac{r_2\big(h(t),h'(t)\big)}{|h(t),h'(t)|^2}\right| \cdot |h(t),h'(t)|^2\,dt,$$

czyli:

$$\frac{|R|}{|h|_{C^1}} \le \sqrt{2k}\int_a^b \left|\frac{r_2\big(h(t),h'(t)\big)}{|h(t),h'(t)|^2}\right| \cdot \|h\|_{C^1}\,dt \xrightarrow{\|h\|\to 0} 0.$$

W ten sposób pokazaliśmy, że reszta faktycznie spełnia warunki na istnienie pochodnej mocnej funkcjonału J. Wróćmy teraz do równania (14) i przeanalizujmy warunki zerowania pochodnej mocnej:

$$\delta_f \cdot \|h\| = 0 \Leftrightarrow \int_a^b \left[\frac{\partial F}{\partial f} - \frac{d}{dv}\left(\frac{\partial F}{\partial f'}\right)\right] h \, dv = 0.$$

Zatem z dowolności wyboru h warunek zanikania liniowego przyrostu funkcjonału jest opisany równaniem:

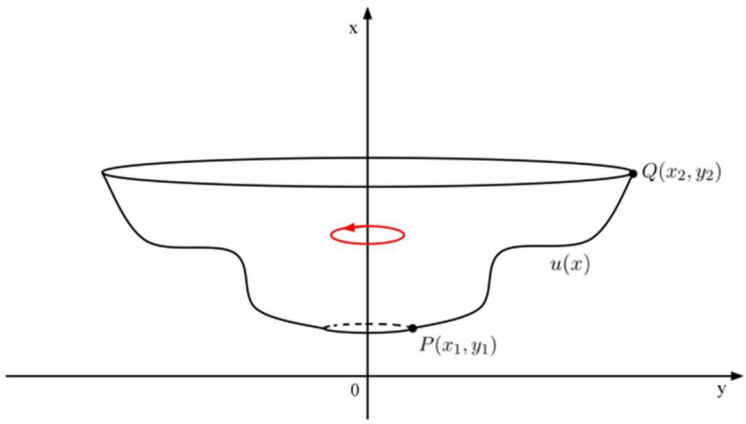
$$\frac{\partial F}{\partial f} = \frac{d}{dv}\left(\frac{\partial F}{\partial f'}\right).$$

Równanie to nazywamy równaniem Eulera – Lagrange'a. Stanowi ono kluczowy wniosek rachunku wariacyjnego.

Przykład z nieoczekiwanymi konsekwencjami

Dysponując wszystkimi wprowadzonymi narzędziami jesteśmy w stanie rozwiązywać całkowicie nowej klasy problemy, polegające na szukaniu ekstremów funkcjonałów. Przykładem takiego zagadnienia jest wyznaczenie minimalnych powierzchni obrotowych.

Mając dane dwa punkty P i Q na półpłaszczyźnie o współrzędnych (x_1, y_1) oraz (x_2, y_2), rozważmy krzywą $u(x)$ łączącą punkty P i Q. Powierzchnia obrotu powstaje przez obrót tej krzywej względem osi x. Naszym celem jest znalezienie krzywej minimalizującej pole powierzchni. Jest to jednoznaczne ze znalezieniem powierzchni minimalnej łączącej dwa pierścienie o promieniach y_1 i y_2. W zależności od położenia punktów P i Q, problem może mieć rozwiązanie ciągłe lub nieciągłe, jednak tutaj rozpatrzymy wyłącznie rozwiązanie ciągłe.



Rys. 1 Punkty P i Q wraz z łączącą je krzywą oraz szukaną powierzchnią obrotową.

Zakładając, że szukane rozwiązanie jest ciągłe klasy C^2, do wyznaczenia krzywej minimalizującej możemy zastosować rachunek wariacyjny.

Niech $y(x) \in C^2([x_1, x_2])$ będzie funkcją minimalizującą funkcjonał pola powierzchni obrotowej. Pole to uzyskujemy sumując pola infinitezymalnie małych pierścieni (cylindrów) o powierzchni elementarnej dS, wyrażonej wzorem:

$$dS = 2\pi y \cdot \sqrt{dx^2 + dy^2}\,.$$

Wówczas całkowite pole powierzchni wynosi:

$$Pole = \int_{x_1}^{x_2} dS = \int_{x_1}^{x_2} 2\pi y \sqrt{1 + (\frac{dy}{dx})^2}\, dx = \int_{x_1}^{x_2} 2\pi y \sqrt{1 + (y'(x))^2}\, dx\,.$$

Funkcjonał pod całką ma postać $F = F(y(x), y'(x)) = y\sqrt{1 + (y'(x))^2}$ i nie zależy jawnie od zmiennej x, tj. $\frac{\partial F}{\partial x} = 0$. W takim przypadku równanie Eulera – Lagrange'a przyjmuje postać:

$$y' \frac{\partial F}{\partial y'} - F = C,$$

gdzie $C \in R$ jest stałą.

Rozpisując to równanie dla naszego funkcjonału, otrzymujemy:

$$y' \frac{\partial (2\pi y\sqrt{1 + (y')^2})}{\partial y'} - 2\pi y \sqrt{1 + y'} = C$$

$$y' \cdot 2\pi y - y'\sqrt{1 + (y')^2} - 2\pi y\sqrt{1 + (y')^2} = C$$

$$\frac{-y}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}} = -\left(\frac{-C}{2\pi}\right) \equiv -a.$$

Przekształcając powyższe równanie tak, aby dx i dy znalazły się po przeciwnych stronach i obustronnie całkując, uzyskujemy:

$$\int \frac{dy}{\sqrt{y^2 - a^2}} = \int \frac{1}{a} dx.$$

Aby rozwiązać całkę po lewej stronie, podstawiamy:

$$y = a\ \cosh t, dy = a\ \sinh t\ dt.$$

Wtedy:

$$\int \frac{a\ \sinh t\ dt}{\sqrt{(a\ \cosh t)^2 - a^2}} = \int \frac{1}{a} dx$$

$$\int \frac{a\ \sinh t\ dt}{a\ \sinh t} = \int \frac{1}{a} dx$$

$$\int dt = \int \frac{1}{a} dx$$

$$t + C = \frac{x}{a} + D.$$

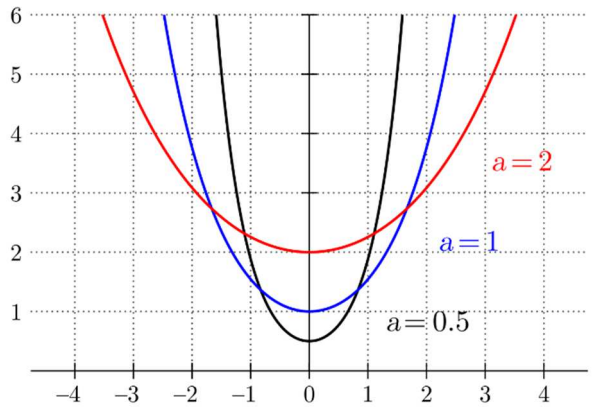
Po podstawieniu odwrotnej funkcji hiperbolicznej oraz wprowadzeniu stałej przesunięcia $b \in R$, mamy:

$$cosh^{-1}\left(\frac{y}{a}\right) = \frac{x - b}{a}.$$

Stąd otrzymujemy równanie funkcji

$$y(x) = a\, cosh\left(\frac{x - b}{a}\right),$$

która nazywana jest krzywą łańcuchową i została przedstawiona na rysunku 2.



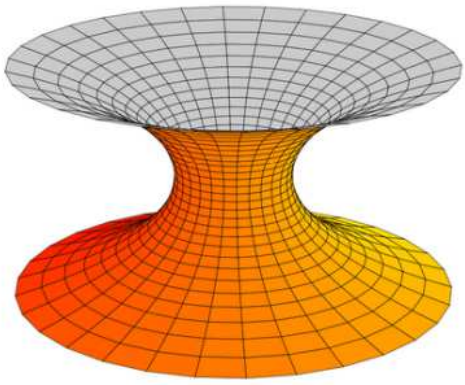
Rys. 2 Wykres funkcji krzywej łańcuchowej dla $b = 0$ i różnych wartości stałej a.

Szukaną powierzchnię o minimalnym polu otrzymujemy przez obrót tej krzywej wokół osi $x \in R$. Powierzchnia ta nazywana jest katenoidą, a jej równanie uzyskujemy, stosując odpowiednią macierz obrotu:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}.$$

Wtedy układ równań, który zadaje powierzchnię katenoidy ma postać:

$$\begin{aligned} x &= x \\ y &= a\cos\theta\, cosh\left(\frac{x - b}{a}\right) \\ z &= a\sin\theta\, cosh\left(\frac{x - b}{a}\right) \end{aligned}$$



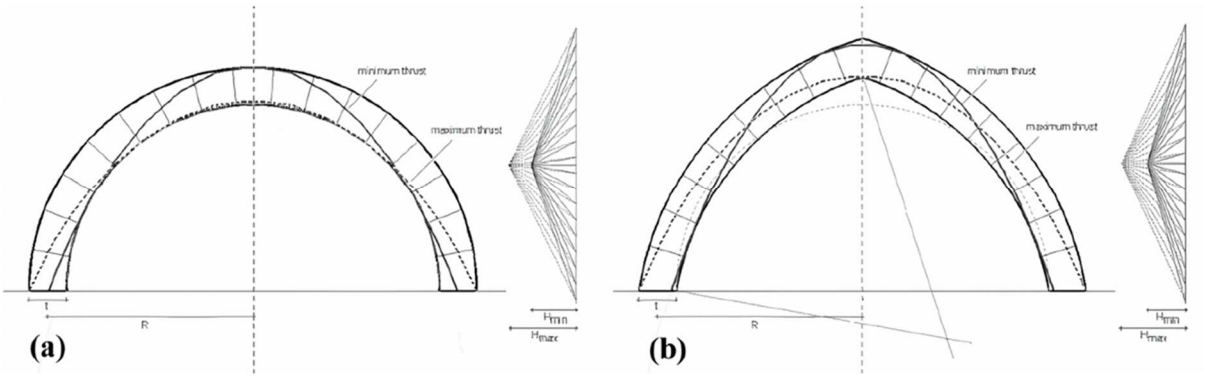
Rys. 3 Ilustracja przedstawiająca powierzchnię katenoidy [9].

Przedstawiony przykład ilustruje, w jaki sposób można zastosować rachunek wariacyjny do rozwiązywania praktycznych problemów. Dzięki równaniu Eulera – Lagrange'a mogliśmy zminimalizować funkcjonał pola powierzchni, co pozwoliło na znalezienie równania krzywej łańcuchowej a dalej również równania katenoidy.

Krzywa którą otrzymaliśmy, nazywana jest krzywą łańcuchową (catenary), ponieważ jest to naturalny kształt doskonale nierozciągliwej i nieskończenie wiotkiej liny o jednostajnie rozłożonej masie. Funkcjonał, który zminimalizowano, jednocześnie minimalizuje energię potencjalną układu, na którą wpływają naciski i naprężenia w konstrukcji. Dzięki zastosowaniu krzywej łańcuchowej możliwe jest projektowanie optymalnych struktur architektonicznych, takich jak łuki czy kopuły powstałe przez obrót łuku względem osi symetrii.

W 1675 roku Robert Hooke odkrył zasadę, którą podsumował maksymą: *„Jak wisi elastyczna lina, tak odwrócony będzie stał sztywny łuk”* [3]. Hooke nie wyprowadził formalnego równania krzywej łańcuchowej, wiedział jednak, że jego intuicja jest poprawna. Odkrycie to umożliwiło optymalizację budowy łuków i kopuł pod kątem rozkładu nacisków, w odróżnieniu od wcześniej stosowanych form półsferycznych.

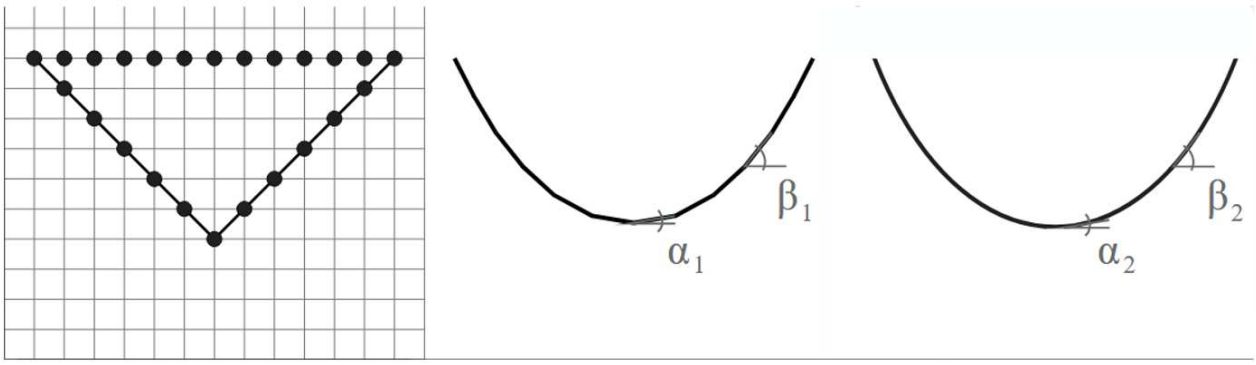
Minimalny nacisk elementu konstrukcyjnego to minimalna siła z jaką ten obiekt oddziałuje na sąsiednie elementy konstrukcji, a maksymalny nacisk, zwany również stanem aktywnym, to maksymalna wartość siły poziomej jaką element jest w stanie przenieść lub zapewnić. Wartość ta może stać się bardzo duża, a zatem maksymalny nacisk elementu będzie ograniczony przez wytrzymałość materiału na zgniatanie, lub stabilność sąsiednich elementów.



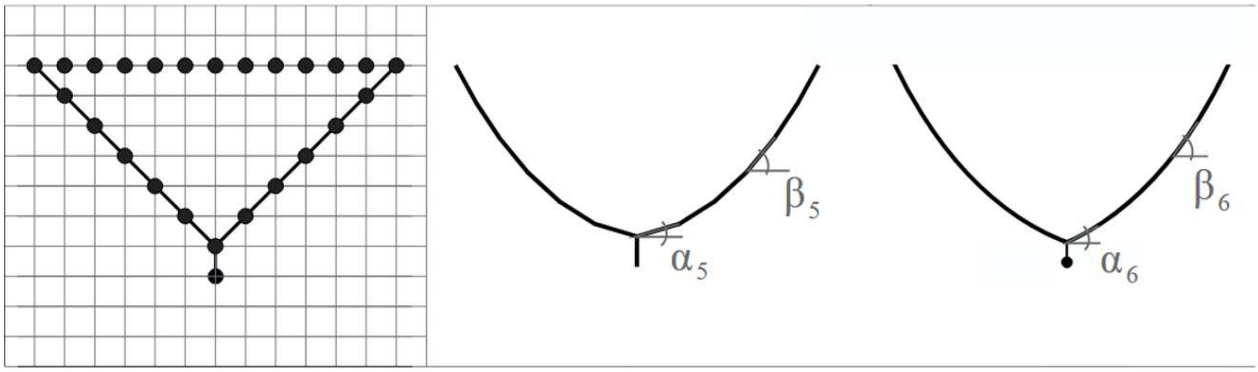
Rys. 4 Ilustracja porównująca (a) łuk półkolisty oraz (b) łuk spiczasty. Źródło: [3]

Łuki przedstawione na Rysunku 4 mają taki sam stosunek grubości do promienia. Ich minimalne i maksymalne naciski, wyrażone jako procent ich ciężaru własnego wynoszą odpowiednio: (a) 16% i 25%, (b) 14% i 23%. Oba łuki mają prawie identyczny ciężar własny, jednak łuk na rysunku (b) wywiera około 15% mniejszy nacisk niż łuk na rysunku (a).

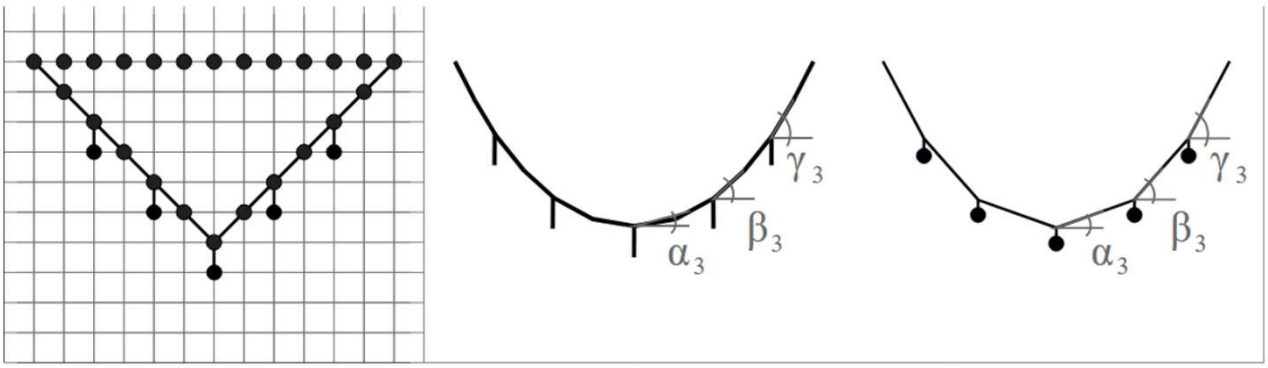
Aby znajdować optymalne kształty, możemy wykorzystywać więc modele fizyczne, składające się z podwieszonych łańcuchów, obciążonych dodatkową masą.



Rys. 5 Otrzymane kształty łańcucha pod ciężarem własnym, od lewej: model wyjściowy, deformacja komputerowa, model fizyczny. Źródło: [3]



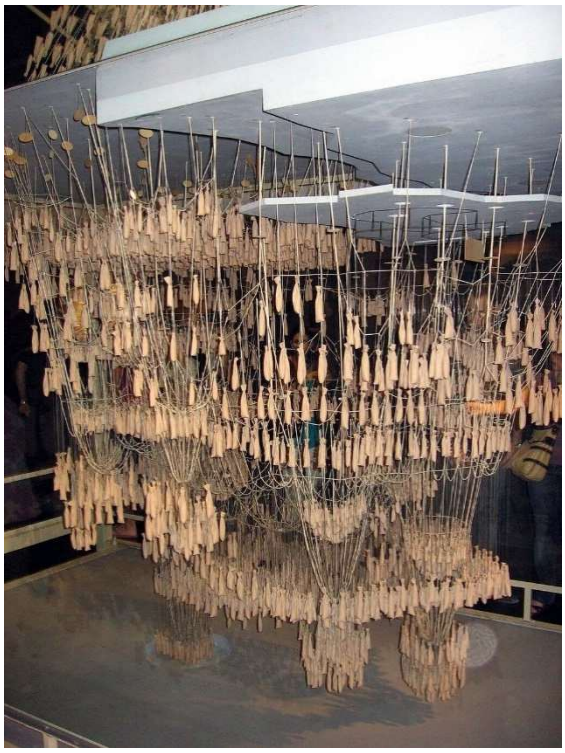
Rys. 6 Otrzymane kształty łańcucha pod ciężarem własnym i jedną siłą skupioną, od lewej: model wyjściowy, deformacja komputerowa, model fizyczny. Źródło: [3]



Rys. 7 Otrzymane kształty łańcucha pod ciężarem własnym i pięcioma siłami skupionymi, od lewej: model wyjściowy, deformacja komputerowa, model fizyczny. Źródło: [3]

Robert Hooke zastosował swoją technikę do wyznaczenia idealnego kształtu dla kopuły katedry św. Pawła w Londynie [4], [5]. W 1748 roku Giovanni Poleni dokonał oceny bezpieczeństwa pękniętej kopuły św. Piotra w Rzymie. Stosując metody inspirowane pracami Hooke'a wykazał, że kopuła była bezpieczna.

Jednym z bardziej rozpoznawalnych architektów wykorzystujących eksperymentalne modelowanie konstrukcji jest Antoni Gaudi [6]. W Sagrada Família Museum znajduje się replika jego modelu niedokończonej kaplicy Colònia Güell [7].



Fot 1. Model Colònia Güell, Sagrada Família Museum

We współczesnym inżynierskim świecie, specjalistyczne programy komputerowe wykorzystują rachunek wariacyjny w ramach zaawansowanych technik numerycznych, takich jak na przykład metoda elementów skończonych (MES) [8]. Dzięki temu nie ma potrzeby tworzenia rzeczywistych, fizycznych modeli. Używając rachunku wariacyjnego, możemy minimalizować energię potencjalną układu dla dowolnych warunków brzegowych. Sposób cyfrowy pozwala również na większą kontrolę nad dokładnością rozwiązania, w stosunku do modelowania eksperymentalnego.

Literatura

[1] Wojtyński W., Matematyczne metody mechaniki, Uniwersytet Warszawski, 2011
[2] Gelfand I.M., Fomin S.V., Calculus Of Variations, Moscow State University, 1963
[3] Block P., DeJong M., Ochsendorf J., As Hangs the Flexible Line: Equilibrium of Masonry Arches, The Nexus Network Journal, Vol. 8, No 2, pp 13-24, October 2006
[4] Inwood S., The Forgotten Genius. San Francisco: MacAdam/Cage Pub. (Published in the UK (2003) as *The man who knew too much: the inventive life of Robert Hooke, 1635-1703*, London, Pan Books), 2003

[5] Gribbin J., Gribbin M., Out of the shadow of a giant: Hooke, Halley and the birth of British science. London: William Collins, 2017
[6] Nasir O., Akhtar W., Iqbal M., Kamal M. A., Analyzing the Architecture of Antonio Gaudí with Reference to Art Nouveau Style: An Inspiration from Nature. Architecture Engineering and Science. 3. 309, 2022
[7] UNESCO World Heritage Centre. (n.d.). Works of Antoni Gaudí
[8] Ajay H., Finite Element Method – What Is It? FEM and FEA Explained, March 2024
[9] Karakoc S., On Minimum Homotopy Areas, PhD Thesis, Tulane University, May 2017

MathUp

Konferencja Zastosowań Matematyki

Pod redakcją: dr inż. Gertruda Gwóźdź-Łukawska
Recenzja naukowa: dr inż. Gertruda Gwóźdź-Łukawska,
dr Elżbieta Galewska,
dr inż. Elżbieta Kotlicka-Dwurznik,
dr Monika Potyrała
Skład i edycja: dr hab. inż. Ewa Korzeniewska, prof. uczelni

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Centrum Nauczania Matematyki i Fizyki
Politechnika Łódzka 2025

ISBN 978-83-938538-5-4