Spis treści

[1. Wstęp 3](#_Toc85394407)

[2. Problemy obliczeniowe 4](#_Toc85394408)

[2.1. Problem spełnialności (SAT) 4](#_Toc85394409)

[2.2. Problem komiwojażera (TSP) 5](#_Toc85394410)

[2.3. Programowanie nieliniowe (NLP) 6](#_Toc85394411)

[2.4. Problem planowania projektu z wieloma wymaganymi umiejętnościami i ograniczonymi zasobami (MSRCPSP) 7](#_Toc85394412)

[3. Elementu problemu 9](#_Toc85394413)

[3.1. Model 9](#_Toc85394414)

[3.2. Rozmiar przestrzeni przeszukiwania 9](#_Toc85394415)

[3.3. Sąsiedztwo 9](#_Toc85394416)

[3.4. Cel 9](#_Toc85394417)

[3.5. Funkcja oceny 9](#_Toc85394418)

[3.6. Ograniczenia 9](#_Toc85394419)

[4. Klasyczne metody rozwiązywania problemów 10](#_Toc85394420)

[4.1. Przeszukiwanie lokalne 10](#_Toc85394421)

[4.2. Algorytmy zachłanne 10](#_Toc85394422)

[4.3. Symulowane wyżarzanie 10](#_Toc85394423)

[5. Algorytm genetyczny 11](#_Toc85394424)

[5.1. Selekcja 11](#_Toc85394425)

[5.2. Krzyżowanie 11](#_Toc85394426)

[5.3. Mutacja 11](#_Toc85394427)

[6. Badania 12](#_Toc85394428)

[6.1. Opis sposoby realizacji projektu i metodyki badań 12](#_Toc85394429)

[6.2. Otrzymane wyniki 12](#_Toc85394430)

[7. Podsumowanie 13](#_Toc85394431)

[8. Bibliografia 14](#_Toc85394432)

# Wstęp

# Problemy obliczeniowe

W życiu napotyka się na wiele problemów które są trudne do rozwiązania, zarówno dla człowieka, jak i dla komputerów. Powodów takiego stanu rzeczy może być kilka, między innymi:

Liczba potencjalnych rozwiązań dla danego problemu jest tak duża, że nie jest możliwe przeszukanie wszystkich rozwiązań w zadowalającym nas czasie. Czasem potrafiło by to zająć miliony lat dla niektórych problemów, ale nie raz nawet parę minut może być zbyt długim oczekiwaniem – przykładowo przy wyznaczaniu nowej trasy dla nawigacji samochodowej.

* Problem jest tak skomplikowany, że modele które używamy są zbyt uproszczone aby dać sensowny rezultat – przykładowo ciężko jest przewidzieć pogodę na rok do przodu, gdyż jest tak wiele zmiennych, że jest to praktycznie niemożliwe do zrobienia przy zachowaniu jakiejkolwiek dokładności i idącej za tym użyteczności dla takiej prognozy.
* Ograniczenia które są nałożone na rozwiązania są tak skomplikowane, że problematyczne jest w ogóle stworzenie takiego które było by prawidłowe i nie łamało żadnych zasad – przykładowo tylko jeden klucz może odszyfrować poprawnie zaszyfrowane dane.

W poniższym rozdziale zostaną przedstawione trzy pomocnicze problemy i jeden główny, który jest tematem tej pracy magisterskiej.

## Problem spełnialności (SAT)

Jednym z podstawowych zagadnień rachunku zdań w logice jest problem spełnialności. Polega on na znalezieniu takich wartości zmiennych, które mogą przyjmować tylko wartości prawda lub fałsz, dla których dana formuła logiczna będzie spełniona. Przykładowy fragment takiej formuły może wyglądać następująco:

Problem ten, tak samo jak kolejne przedstawione w tej pracy, był pierwszym który udowodniono, że należy do grupy problemów NP-zupełnych (Cook, 1971). Oznacza to, że:

* Najlepsze rozwiązanie nie może zostać znalezione w czasie wielomianowym,
* Sprawdzenie czy dane rozwiązanie jest poprawne jest możliwe w czasie wielomianowym.

Tak długo jak formuła nie jest skomplikowana, to możemy sprawdzić po prostu każdą możliwą opcję. Jednak jak łatwo zauważyć liczba kombinacji rośnie wykładniczo – dla każdej zmiennej są możliwe dwa stany, więc potencjalnych rozwiązań jest tyle ile dwa do potęgi. Rośnie to na tyle szybko, że już dla 50 takich zmiennych, liczba możliwych stanów jest większa niż biliard. Już taka ilość jest bardzo trudna do sprawdzenia nawet w przypadku pomocy komputerów, a dla ludzi bez nich dosłownie niemożliwa.

Jednym z najczęściej spotykanych obecnie zastosowań dla silników rozwiązujących ten problem są zagadnienia związane z projektowaniem układów cyfrowych. Dzięki temu można chociażby znaleźć bardziej optymalne rozłożenia komponentów w układach scalonych (Nam, Sakallah i Rutenbar, 2002), oraz przeprowadzić formalną weryfikację mikroprocesorów potokowych (Bryant, German, Velev i Murray, 1999).

## Problem komiwojażera (TSP)

W problemie komiwojażera (skrót jego nazwy to TSP) mamy zdefiniowaną listę miast i odległości pomiędzy każda parą z nich. Polega on na znalezieniu najkrótszej możliwej takiej trasy, która odwiedza każde miasto dokładnie raz, a na sam koniec wraca do miejsca początkowego. Jest to szczególny przypadek problemu marszrutyzacji, który pozwala na odwiedzenie każdego miasta więcej niż raz. Przykładowe rozwiązanie może wyglądać następująco:

Obraz zawierający tekst, zewnętrzne, ruch uliczny, jasne

Opis wygenerowany automatycznie

Optymalnych rozwiązań dla problemu jest zawsze więcej niż jeden. Nawet jeżeli istnieje tylko jeden optymalny cykl dla takiego grafu, to zawsze możemy wygenerować nowe rozwiązanie zaczynając podróż z innego miasta, a także można je odwrócić. Przykładowo, jeżeli przedstawimy rozwiązanie jako listę odwiedzanych miast, to jeżeli pierwsza była by optymalnym rozwiązaniem, to wszystkie następujące po niej także takie będą:

* 1-3-5-2-4
* 3-5-2-4-1
* 5-2-4-1-3, itd.

Występują w nim pewnie uproszczenia w stosunku do prawdziwego życia. Jednym z nich jest fakt, że w rzeczywistości nie każde miasto musi mieć pomiędzy sobą bezpośrednią drogę. Dodatkowo nie zawsze koszt pokonania takiej ścieżki jest taki sam w obie strony. Może on także zależeć od godziny w jakiej dana podróż się odbywa.

Problem ten został pierwszy raz zdefiniowany w latach 30 i jest jednym z najczęściej używanym punktów odniesienia dla nowo powstających metod optymalizacji. Ze względu na jego popularność, mimo tego, że jest on NP-kompletny, istnieje wiele algorytmów i heurystyk które pozwoliły wygenerować najlepsze możliwe rozwiązania nawet dla przypadków składających się z dziesiątek tysięcy miast. Często używanym zbiorem instancji problemu o rożnym poziomie trudności jest TSPLIB, a największym całkowicie rozwiązanym przypadkiem jest problem składający się z 85 900 miast (Rego, Gamboa, Glover i Osterman, 2011).

## Programowanie nieliniowe (NLP)

Kolejna klasą problemów, są te związane z programowaniem nieliniowym. Polegają one na zalezieniu minimum, maksimum, lub punktów zerowych dla danej funkcji. Dodatkowo na przestrzeń rozwiązań mogą zostać nałożone ograniczenia, w postaci równości, lub nierówności, które muszą zostać spełnione, aby dane rozwiązanie było poprawne.

Zmienne mogą przyjmować dowolne wartości z zakresu liczb rzeczywistych. W związku z tym istnieje nieskończoność potencjalnych rozwiązań – ze względu na to, że zawsze możemy podać taką wartość, której jeszcze nie rozwiązywaliśmy. Nie każde z tych rozwiązań za to musi być optymalne, lub w ogóle poprawne. W szczególności możemy dla danego problemu nie wiedzieć, czy w ogóle istnieje takie rozwiązanie które spełnia wszystkie podane ograniczenia i dana metoda może próbować znaleźć jakiekolwiek które będzie poprawne.

Przykładowym problemem może być znalezienie maksimum dla następującej funkcji (Keane, 1996), z następującymi ograniczeniami:

Możemy zwizualizować tą funkcję przykładowo dla (Michalewicz i Fogel, 2004). Rozwiązaniom niespełniającym ograniczeń nadano zerową wartość:



Jak widać powierzchnia tego wykresu jest bardzo nieregularna, zwłaszcza im bliżej początków osi. Dodatkowo możemy zaobserwować tutaj nagły spadek do zera, co oznacza niepoprawne rozwiązania. Jest on bardzo blisko fragmentu gdzie znajdują się najlepsze rozwiązania według tego wykresu, co pokazuje, że granica pomiędzy globalnym maksimum, a nieprawidłowym rozwiązaniem może być bardzo cienka. W związku z tym niektóre metody rozwiązań koncentrują się właśnie na tych krawędziach, uważając je za rejony z największym potencjałem. Oczywiście wszystko to zależy od konkretnego przypadku dla danego równania.

## Problem planowania projektu z wieloma wymaganymi umiejętnościami i ograniczonymi zasobami (MSRCPSP)

Problem planowania projektów z wieloma wymaganymi umiejętnościami i ograniczonymi zasobami został zdefiniowany przez naukowców związanych z Politechniką Wrocławską, Ich celem było stworzenie problemu, która miał by jednoczesny balans prostoty implementacji i wierności rzeczywistej sytuacji. Na nim skupiłem swoje badania, których założenia i wyniki zostaną przedstawione w nadchodzących rozdziałach.

Celem klasycznego problemu planowania projektów było przydzielenie zasobów do zadań w taki sposób aby zminimalizować czas i/lub koszt wykonania projektu. Jednakże pojawiają się w nim także ograniczenia:

* Niektóre zadania, mogą do rozpoczęcia pracy nad nimi wymagać zakończenia innych zadań.
* Dany zasób może zostać przydzielony tylko do jednego zadania naraz i musi je skończyć w całości, bez możliwości dzielenia pracy pomiędzy dwoma rozpoczętymi zadaniami.

Problem ten został rozszerzony przez dodanie umiejętności. Każdy zasób posiada pewien zbiór umiejętności na określonym poziomie, a każde zadanie wymaga jedną umiejętność na określonym poziomie. W związku z tym zadanie może mieć przydzielony dany zasób, tylko jeżeli posiada on daną umiejętność na równym lub wyższym poziomie niż to potrzebne.

Został on opracowany razem ze współpracy z inżynierami z firmy Volvo. Dzięki temu jego twórcy dostarczają oparte na rzeczywistości, ale zanonimizowane, zestawy danych odpowiadające rzeczywistym projektom. Jest ich łącznie 42 - 6 łatwiejszych problemów i 36 pełnowymiarowych, które posiadają do:

* 200 zadań.
* 150 relacji pomiędzy zadaniami.
* 15 umiejętności na różnych poziomach.
* 40 zasobów.

# Elementu problemu

Aby użyć jakikolwiek algorytm lub heurystykę do rozwiązania danego problemu musimy najpierw zdefiniować pewne podstawowe koncepty dla danego problemu:

* Model dla problemu
* Cel który chcemy osiągnąć
* Funkcję oceny rozwiązania
* Sąsiedztwo rozwiązań
* Potencjalne ograniczenia nałożone na rozwiązania

Zostaną one przestawione w poniższych podrozdziałach na przykładzie omówionych w poprzednim rozdziale przypadków.

## Model i wielkość przestrzeni poszukiwań

Pierwsza rzeczą którą trzeba zdefiniować w celu rozwiązania danego problemu obliczeniowego jest zdefiniowanie jego modelu. Jest to określenie sposobu w jaki możemy przedstawić alternatywne rozwiązania i umożliwić nam ich modyfikację. Nie istnieje jeden najlepszy model dla danego problemu. Jak udowodnili (Fogel i Ghozeil, 1997) wśród takich przedstawień problemu, które są dla siebie bijekcjami, żadna z nich nie daje przewagi w rezultatach nad innymi. W związku z tym najczęściej wybierane są takie, które są najbardziej intuicyjne dla danego problemu, co pozwala na ich łatwiejsze zrozumienie.

Przypadek SAT jest najprostszy do zamodelowania. Jego rozwiązaniem jest ciąg binarnych wartości, reprezentujących po kolei stany jakie przypisujemy poszczególnym zmiennym. W związku z tym możliwych rozwiązań jest dokładnie . Dla 100 zmiennych jest to wartość rzędu .

W przypadku TSP najczęściej spotykanym modelem jest permutacja liczb naturalnych od do . Każda z liczb jest przypisana do konkretnego miasta i ich kolejność w danym rozwiązaniu jest także kolejnością w której zostaną odwiedzone. W związku z tym, że w podstawowym problemie komiwojażera odległości pomiędzy miastami są symetryczne, to nie ma znaczenia dla danego rozwiązania czy lista miast zostanie przeprocesowana od lewej do prawej, czy od prawej do lewej. Dodatkowo także nie ma znaczenia od którego miasta zaczniemy taką podróż. W związku z tymi dwoma obserwacjami wielkość przestrzeni poszukiwań wynosi . Dla porównania z poprzednim problemem, dla 100 miast wartość ta jest rzędu

Dla NLP teoretyczna przestrzeń przeszukiwań jest nieograniczona, ponieważ każda zmienna może przyjąć dowolną wartość ze zbioru liczb rzeczywistych. W związku z tym potrzeba każdą ze zmiennych odpowiednio poddać dyskretyzacji, aby było możliwe zastosowanie komputerów w celach obliczeniowych. W związku z tym możemy albo dokonać tego procesu samemu, na przykład dzieląc daną przestrzeń na określoną liczbę punktów w stałej odległości, lub skorzystać z precyzji jakie dają nam liczby zmiennoprzecinkowe na danych platformach obliczeniowych. W przypadku gdy użylibyśmy standardowych liczb zmiennoprzecinkowych o podwójnej precyzji (IEEE Standard for Floating-Point Arithmetic, 2008) takich rozwiązań było by , co dla równań z 100 zmiennymi dało by rząd wielkości .

## Sąsiedztwo

## Cel

## Funkcja oceny

## Ograniczenia

# Klasyczne metody rozwiązywania problemów

## Przeszukiwanie lokalne

## Algorytmy zachłanne

## Symulowane wyżarzanie

# Algorytm genetyczny

## Selekcja

## Krzyżowanie

## Mutacja

# Badania

## Opis sposoby realizacji projektu i metodyki badań

## Otrzymane wyniki

# Podsumowanie

# Bibliografia

Bryant, R. E., German, S., Velev, M. N. i Murray, N. V. (1999). *Microprocessor Verification Using Efficient Decision Procedures for a Logic of Equality with Uninterpreted Functions.* Springer Berlin Heidelberg. doi:10.1007/3-540-48754-9\_1

Cook, S. (1971). *The complexity of theorem proving procedures.* Proceedings of the Third Annual ACM Symposium on Theory of Computing. doi:10.1145/800157.805047

Fogel, D. i Ghozeil, A. (1997). A note on representations and variation operators. *IEEE Transactions on Evolutionary Computation, 1*(2), strony 159-161. doi:10.1109/4235.687882

IEEE Standard for Floating-Point Arithmetic. (2008). 1-70. doi:10.1109/IEEESTD.2008.4610935

Keane, A. (1996). A brief comparison of some evolutionary optimization methods. W *Modern Heuristic Search Methods.* John Wiley.

Michalewicz, Z. i Fogel, D. B. (2004). *How to Solve It: Modern Heuristics.* Springer. doi:10.1007/978-3-662-07807-5

Nam, G.-J., Sakallah, K. i Rutenbar, R. (2002). A new FPGA detailed routing approach via search-based Boolean satisfiability. IEEE. doi:10.1109/TCAD.2002.1004311

Rego, C., Gamboa, D., Glover, F. i Osterman, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research, 211*(3), 427-441. doi:10.1016/j.ejor.2010.09.010