

**Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej**

Projekt dyplomowy

*Optymalizacja trajektorii dronów*

*Drone trajectory optimization*

Autor: *Maciej Morgalla*

Kierunek studiów: *Automatyka i Robotyka*

Opiekun pracy: *dr inż. Piotr Kadłuczka*

Kraków, 2021

Spis treści

[1. Wstęp 3](#_Toc90463178)

[1.1. Wprowadzenie 3](#_Toc90463179)

[1.2. Cel i zakres pracy 3](#_Toc90463180)

[2. Zagadnienia związane z modelem matematycznym i algorytmem 4](#_Toc90463181)

[2.1. Sposoby użycia dronów 4](#_Toc90463182)

[2.2. Problemy optymalizacyjne w sterowaniu dronami 4](#_Toc90463183)

[2.3. Stosowane algorytmy 5](#_Toc90463184)

[3. Formalizacja modelu matematycznego 6](#_Toc90463185)

[3.1. Podstawowe parametry symulacji 6](#_Toc90463186)

[3.2. Wyznaczanie kolejnych pozycji dronów 7](#_Toc90463187)

[3.3. Kolizje 7](#_Toc90463188)

[3.4. Funkcja celu 7](#_Toc90463189)

[4. Algorytm przeszukiwania z zabronieniami 7](#_Toc90463190)

[4.1. Schemat algorytmu 8](#_Toc90463191)

[4.2. Pamięć krótko- i długoterminowa 10](#_Toc90463192)

[5. Opis aplikacji 11](#_Toc90463193)

[5.1. Środowiska programistyczne 11](#_Toc90463194)

[5.2. Weryfikacja prawidłowego działania modelu i algorytmu 11](#_Toc90463195)

[6. Scenariusze testowe 11](#_Toc90463196)

[6.1. Optymalna liczba dronów 11](#_Toc90463197)

[6.2. Prędkość maksymalna dronów 11](#_Toc90463198)

[6.3. Wyrównywanie czasu lotu 11](#_Toc90463199)

[6.4. Maksymalizacja oddalenia dronów 11](#_Toc90463200)

[7. Podsumowanie 11](#_Toc90463201)

[8. Bibliografia 11](#_Toc90463202)

# Wstęp

## Wprowadzenie

## Cel i zakres pracy

# Zagadnienia związane z modelem matematycznym i algorytmem

## Sposoby użycia dronów

## Problemy optymalizacyjne w sterowaniu dronami

Sterowanie dronami nie jest prostym zadaniem. Jak powszechnie wiadomo ich lot jest możliwy dzięki zamontowanymi na nich wirnikom. Ich ilość zależy głównie od producenta, jednakże najczęściej drony posiadają cztery wirniki. Tutaj niestety pojawia się pewien problem, O ile lot tylko w osi *z* (wysokości) nie wymaga szczególnych działań, to już lot w osiach *x* oraz *y* wymaga, aby wirniki miały różne prędkości obrotowe. To zaś może powodować różnicę pomiędzy zadaną trajektorią drona, a rzeczywistą trajektorią [2.5]. Dlatego konieczne jest aby zachowywać określony minimalny dystans pomiędzy dronami.

Dodatkowym czynnikiem przemawiającym za zachowywaniem dystansu jest skutek pracy wirników. Wytwarzanie siły jest możliwe tylko dzięki szybkiemu „wypychaniu” powietrza, dającemu potrzebny efekt odrzutu. Może to powodować nietypowe (turbulentne) ruchy powietrza zarówno powyżej jak i poniżej drona. Gdyby inny dron znalazłby się w takiej strefie, jego dalsza trajektoria mogłaby zostać w pewnym stopniu zaburzona, a w najgorszym przypadku kontrola nad nim byłaby niemożliwa i stałby się zagrożeniem dla innych dronów, a nawet mógłby się rozbić.

Kolejną przeszkodą jest ograniczona pojemność akumulatora. Im dłużej wirniki obracają się tym bardziej zapasy energii kurczą się, tym samym ograniczając maksymalny czas jaki może spędzić w powietrzu. W dodatku po przekroczeniu pewnej granicy poziomu energii maksymalna osiągalna prędkość obrotowa wirników może być nieosiągalna. Wymusza to wymianę bądź ładowanie akumulatora w odpowiednim czasie, a także ograniczenie czasu użycia drona.

Jeszcze innym problemem jest udźwig. Jest to szczególnie palący problem w użyciu dronów do celów transportowych. Technologia ta jest wprowadzana m. in. przez firmę Amazon [2.6]. Wymaga jednak użycia dronów, które są w stanie wytworzyć dużą siłę potrzebną do lotu. Poza tym należy także wziąć pod uwagę dodatkowy opór powietrza spowodowany zwiększeniem się powierzchni, co ponownie pochłania część wytworzonej siły.

## Stosowane algorytmy

Problem przydziału jest powszechnie spotykanym problemem optymalizacyjnym. Pojawia się on przy m.in. masowej produkcji, logistyce, a nawet zwykłym planowaniu dnia. W wielu przypadkach człowiek potrafi samodzielnie rozwiązać taki problem, szczególnie gdy nie jest zbyt skomplikowany. Są też sytuacje, w których złożoność jest na tyle trudna, że poza najtęższymi umysłami na Ziemi, nikt nie jest w stanie szybko podać rozwiązania. Wtedy z pomocą przychodzą właśnie algorytmy.

Wiele z tych problemów można przedstawić za pomocą tabel, grafów czy też funkcji liniowych. Popularnymi algorytmami i sposobami do ich rozwiązywania są w zależności od problemu: algorytm Dijkstry, algorytm węgierski, metoda Symplex, programowanie liniowe, algorytm aproksymacyjny itp.. [2.1]

Istnieją też tak bardzo skomplikowane problemy, że konieczne jest użycie algorytmów heurystycznych. Stosowane są szczególnie w przypadku problemów, których rozwiązanie wymagałoby ogromnej złożoności obliczeniowej, czy też dużej ilości czasu. Ich właściwością jest bowiem skrócenie czasu potrzebnego na obliczenia, możliwym kosztem jakości rozwiązania. Wśród algorytmów heurystycznych możemy znaleźć takie algorytmy jak: algorytm przeszukiwania z zabronieniami (Tabu Search - TS), algorytmy genetyczne oraz algorytm symulowanego wyżarzania. [2.2]

Przykładami użycia jednego z algorytmów heurystycznych – algorytmu przeszukiwania z zabronieniami – są następujące problemy:

* planowanie tras [2.3],
* niepermutacyjny problem przypływowy z kryterium sumacyjnym [2.4],
* uogólniony problem przydziału [4.2].

# Formalizacja modelu matematycznego

## Podstawowe parametry symulacji

Symulowanie lotu dronów wymaga określenia pewnych parametrów. Podstawą jest wyznaczenie przestrzeni symulacji. Tak jak w rzeczywistości jest ona trójwymiarowa. Jej długość, szerokość oraz wysokość jest określana przez operatora symulacji. Zakłada się, że przestrzeń ta jest wypełniona równomiernie powietrzem atmosferycznym o uśrednionych proporcjach poszczególnych jego składników. Ponadto ciśnienie atmosferyczne oraz gęstość powietrza uznaje się za jednakowe niezależnie od wysokości. Dodatkowo uznaje się, że panująca temperatura nie wpływa w żaden sposób na powietrze oraz pracę dronów.

Dodatkowym czynnikiem wprowadzonym do symulacji jest poruszanie się powietrza czyli wiatr. Określana jest jego prędkość względem osi *x* oraz *y* (względem osi *z* zawsze wynosi 0), która w trakcie trwania symulacji jest stała. Co więcej zakłada się, że ruch cząsteczek jest zawsze prostoliniowy, a nigdy nie jest turbulentny. Wyjątkiem jest przelot drona. Na potrzeby wzoru na opór aerodynamiczny zakłada się, że zaraz za dronem powietrze znajduje się w ruchu turbulentnym. To zaburzenie jednak trwa na tyle krótko, iż nie wpływa na inne drony (kluczowa jest tutaj strefa bezpieczeństwa drona, o której więcej informacji w rozdziale 3.3).

Pozycję drona w przestrzeni symulacji określają współrzędne (*x*,*y*,*z*). Jest to dokładna pozycja jego środka ciężkości. Symulacja polega na określaniu kolejnych pozycji co podany okres czasu (okres próbkowania). Czas ten także jest wyznaczany przez operatora symulacji w zależności od oczekiwanej dokładności. Należy tutaj wspomnieć, iż w pewnym zakresie obliczeń rozwiązuje się równanie różniczkowe metodą Eulera. Zbieżność uzyskanego wyniku z wynikiem rzeczywistym jest tym lepsza im mniejszy jest okres próbkowania.

Z perspektywy operatora wyznacza się również siłę oraz maksymalną prędkość drona. Siła to nic innego jak maksymalna siła jaką są w stanie wytworzyć wirniki „wypychając” powietrze. Maksymalna prędkość to swoiste ograniczenie prędkości jaką mogą rozwinąć drony, które służy do testów właściwości symulacji.

## Wyznaczanie kolejnych pozycji dronów

## Kolizje

Każdy dron posiada własną strefę bezpieczeństwa wyznaczoną względem wszystkich osi (np. . Pojawienie się innego drona w tej strefie uznaje się zawsze jako kolizję, niezależnie od tego czy fizyczna kolizja rzeczywiście mogła nastąpić. Należy bowiem brać pod uwagę także inne niebezpieczeństwa wynikające z bliskości dronów opisane w rozdziale 2.2.

Funkcja sprawdzająca czy dochodzi do kolizji otrzymuje pełną listę zawierającą kolejne pozycje dronów w każdej próbce symulacji. Następnie wszystkie pozycje w danym kroku sprawdza się pod kątem odległości we wszystkich trzech wymiarach. Celem uniknięcia powtarzania porównań między tymi samymi pozycjami, wpierw sprawdza się ewentualne kolizje pomiędzy pierwszym dronem i wszystkimi pozostałymi, później zaś pomiędzy drugim, a tymi których numer porządkowy jest wyższy itd.. Jeśli odległości we wszystkich wymiarach będą mniejsze od wyznaczonych dla strefy, to funkcja oznacza takie zbliżenie jako kolizję.

## Funkcja celu

Bardzo ważnym pod względem optymalizacji symulacji jest określająca ją funkcja celu. Jej wartość uzyskuje się z następującego równania:

– czas potrzebny na zajęcie punktów docelowych przez wszystkie drony

*k* – liczba kolizji pomiędzy dronami

*d* – średnie oddalenie dronów podczas symulacji [m]

*a,b,c –* mnożniki poszczególnych zmiennych

Każde z wymienionych kryteriów ma swój własny mnożnik. Jest on wyznaczany odgórnie w zależności od potrzeb symulacji i testów. Najważniejszym z punktu widzenia realizmu jest kryterium kolizji, które powinno praktycznie uniemożliwiać zaliczenie symulacji do jednych z najlepszych. Dlatego mnożnik tego kryterium musi być w porównaniu z pozostałymi dużo większy.

Kryterium czasu trwania zmiany pozycji wszystkich dronów jest szczególnie ważne w kontekście strat energii oraz oszczędności czasu. Co prawda w symulacji zakłada się, że energia akumulatorów dronów pozostaje niezmienna, ale w przypadku użycia symulacji do rzeczywistego wykorzystania kryterium to staje się niezwykle ważne. Jego mnożnik należy więc dostosować w zależności od preferencji.

Można bowiem wyżej cenić kryterium oddalenia dronów od siebie. Jest to szczególnie pożądane jeśli chodzi o estetykę, gdyż donioślej wygląda lot dronów daleko od siebie niż stłoczenie na bardzo małej przestrzeni. Mnożnik tego kryterium musi więc być odpowiednio dostosowany względem mnożnika kryterium czasu trwania.

Ideą istnienia funkcji celu jest jej minimalizacja. Zadanie znalezienia takiego przydziału dronów do punktów docelowych, aby wynikająca z cech symulacji funkcja celu była jak najniższa, należy do zastosowanego algorytmu.

# Algorytm przeszukiwania z zabronieniami

## Schemat algorytmu

Przydział dronów do poszczególnych punktów końcowych w celu optymalizacji parametrów symulacji w danym zakresie nie jest banalny. Dokonanie zmiany celu lotu pomiędzy zaledwie dwoma dronami może całkowicie zmienić wyniki symulacji, ze względu na możliwość zajścia kolizji. W dodatku przy wzroście liczby dronów zwiększa się również liczba możliwych kombinacji, a tym samym pracochłonność przeanalizowania każdej z nich. Dlatego konieczne jest zastosowanie algorytmu heurystycznego w celu zrównoważenia czasu oraz jakości otrzymanego rozwiązania. Na potrzeby niniejszego projektu wybrano algorytm przeszukiwania z zabronieniami – Tabu Search (*TS*).

Algorytm Tabu Search ma swoje korzenie na przełomie lat 60-tych i 70-tych. Znaną dzisiaj formę algorytmu zawdzięczamy profesorowi Fredowi Gloverowi, który sformułował ją w latach 80-tych [4.1]. Idea działania opiera się na przeszukiwaniu określonego sąsiedztwa w poszukiwaniu najlepszego rozwiązania. W rozpatrywanym przypadku będzie to wymiana punktów docelowych pomiędzy dwoma dronami. Następnie algorytm wykonuje kolejne „ruchy” czyli zmiany sąsiedztwa na najlepsze, które mogło zostać osiągnięte z poprzedniego sąsiedztwa. Najlepszy wynik osiągnięty podczas działania algorytmu oraz odpowiadający mu przydział są finalnym rozwiązaniem problemu.

**Algorytm Tabu Search** (na podstawie pseudokodu z artykułu [4.2])

**Krok 1**:

Wyznacz rozwiązanie początkowe *xpocz* i podstaw: *x* := *xpocz*, *xts* := *xpocz*, *Qts* := Q(*xpocz*), *CA*=*False*

**Krok 2**: dla *k*=1 do *K* wykonaj: (*K*- warunek stopu):

*x*(*i\*, j\**) = arg min;

*x*(*i’,j’*) = arg min ;

podstaw: *x* := *x*(*i\*, j\**);

Jeżeli < *Qts*  to *xts* := *x, Qts* := Q(*x*);

Jeżeli < *Qts*  to *xts* := *x*(*i’,j’*)*, Qts* := Q(*x*(*i’,j’*)), *CA*=*True* ;

Dla każdego (*i*,*j*) podstaw: ;

Jeżeli *CA*=*False*, to *KLT*(*i\**,*j\**) = *T*, inaczej *KLT*(*i’*,*j’*) = *T;*

Jeżeli *CA*=*False*, to *DLT(i\**,*j\**):= *DLT(i\**,*j\**)+1, inaczej *DLT(i’*,*j’*):= *DLT(i’*,*j’*)+1,*CA*=*False.*

Opis oznaczeń w pseudokodzie:

*CA* - zmienna określająca czy zastosowano kryterium aspiracji,

α – wartość współczynnika kary,

*KLT* - lista tabu zawierająca liczbę tur, w których określony krok jest zabroniony,

*DLT* – lista tabu zawierająca liczbę poszczególnych dokonanych kroków algorytmu,

*T* – liczba tur, w których dany krok znajduje się na liście *KLT*.

## Pamięć krótko- i długoterminowa

W zależności od potrzeb możliwe jest zastosowanie algorytmu przeszukiwania z zabronieniami na kilka sposobów. W tym przypadku użyte zostaną algorytmy z pamięcią krótkoterminową oraz algorytm z pamięcią zarówno krótko- jak i długoterminową.

Pamięć krótkoterminowa (*KLT*) to lista tabu, która posiada po jednej pozycji dla każdego możliwego ruchu algorytmu. Domyślnie zapełniona jest zerami. Po każdym kroku algorytmu, najpierw od wszystkich niezerowych wartości odejmuje się jeden, a następnie na pozycji odpowiadającej wykonanemu ruchowi pojawia się wartość *T,* czyli określony w liczbie kroków okres znajdowania się ruchu na liście tabu. Algorytm sprawdza dla każdego ruchu czy nie znajduje się na *KLT* (czyli czy *KLT*(*i*,*j*)>0). Jeśli *KLT*(*i*,*j*)=0, to gdy ruch okaże się najlepszy, może być wykonany. Jeśli jednak *KLT*(*i*,*j*)>0, to ruch może zostać wykonany tylko jeżeli spełni warunek aspiracji.

Warunek aspiracji to swoisty wytrych, który umożliwia algorytmowi wykonanie niedozwolonego ruchu. Jeśli jego wynik okaże się lepszy nie tylko od najlepszego wyniku „dozwolonego” ruchu w tym kroku algorytmu, ale także od najlepszego wyniku w perspektywie wszystkich dotychczasowych kroków algorytmu, to kryterium aspiracji zostaje spełnione.

Główną różnicą pomiędzy zastosowanymi dwoma typami algorytmu jest zastosowanie bądź pominięcie pamięci długoterminowej (*DLT*). Podobnie jak lista *KLT* zawiera po jednej pozycji dla każdego ruchu. Jednakże, jej przeznaczeniem jest zliczanie liczby wykonanych poszczególnych ruchów. Informacja ta jest wykorzystywana przy obliczaniu wyniku dla rozpatrywanego przydziału, bowiem do podstawowego wyniku dodaje się iloczyn wartości na odpowiedniej pozycji *DLT* i stałej wartości współczynnika kary, podzielonego przez ilość wykonanych iteracji algorytmu. Ma to na celu zapobieganiu wykonywaniu tych samych ruchów zbyt wiele razy. [4.2]

## 

# Opis aplikacji

## Środowiska programistyczne

## Weryfikacja prawidłowego działania modelu i algorytmu

# Scenariusze testowe

## Optymalna liczba dronów

## Prędkość maksymalna dronów

## Wyrównywanie czasu lotu

## Maksymalizacja oddalenia dronów

# Podsumowanie

# Bibliografia

Rozdział 4:

[4.1] F.Glover, E. Taillard, D. de Werra: *A user's guide to tabu search.* Annals of Operations Research 41: 3-28, 1993r.

(użyte w rozdziale 2)[4.2] K. Wala: *Algorytm Tabu w optymalizacji uogólnionego problemu przydziału.* Automatyka Tom 11 Zeszyt 1-2: 309-316, 2007r.

Rozdział 2:

[2.1] S. Dasgupta, C. Papadimitriou, U. Vazirani: *Algorytmy* Wydawnictwo Naukowe PWN 2012r.

[2.2] *Algorytmy heurystyczne*, [Online]: <http://algorytmy.ency.pl/artykul/algorytmy_heurystyczne>

[2.3] J. Pempera *Równoległy algorytm tabu z elementami inspirowanymi naturą dla problemu planowania tras.* Automatyka Tom 15 Zeszyt 2: 399-408, 2011r.

[2.4] J. Grabowski, J. Pempera *Hybrydowy algorytm tabu dla niepermutacyjnego problemu przepływowego z kryterium sumacyjnym.* Automatyka Tom 13 Zeszyt 2: 289-296, 2009r

[2.5] R. Kowalik, T. Łusiak, A. Novak *A mathematical model for controlling a quadrotor UAV.* Transactions on Aerospace Research Rocznik 2021 Tom 3 (264): 58-70, 2021r.

[2.6] D. Tobór, J. Barcik, P. Czech *Legal aspects of air transport safety and the use of drones*. Scientific Journal of Silesian University of Technology. Series Transport Tom 97: 167-179, 2017r.