Politechnika Rzeszowska im. Ignacego Łukasiewicza

Projekt Optymalizacja nieliniowa

Krawiec Piotr

07/11/2021

Streszczenie

Celem pracy jest przedstawienie zagadnień optymalizacji z wykorzystaniem metod bezpośredniego przeszukiwania.

Piotr Krawiec L1

Semestr: 2021/2022

Kierunek: III/FS0-DI

Numer indeksu: 164165

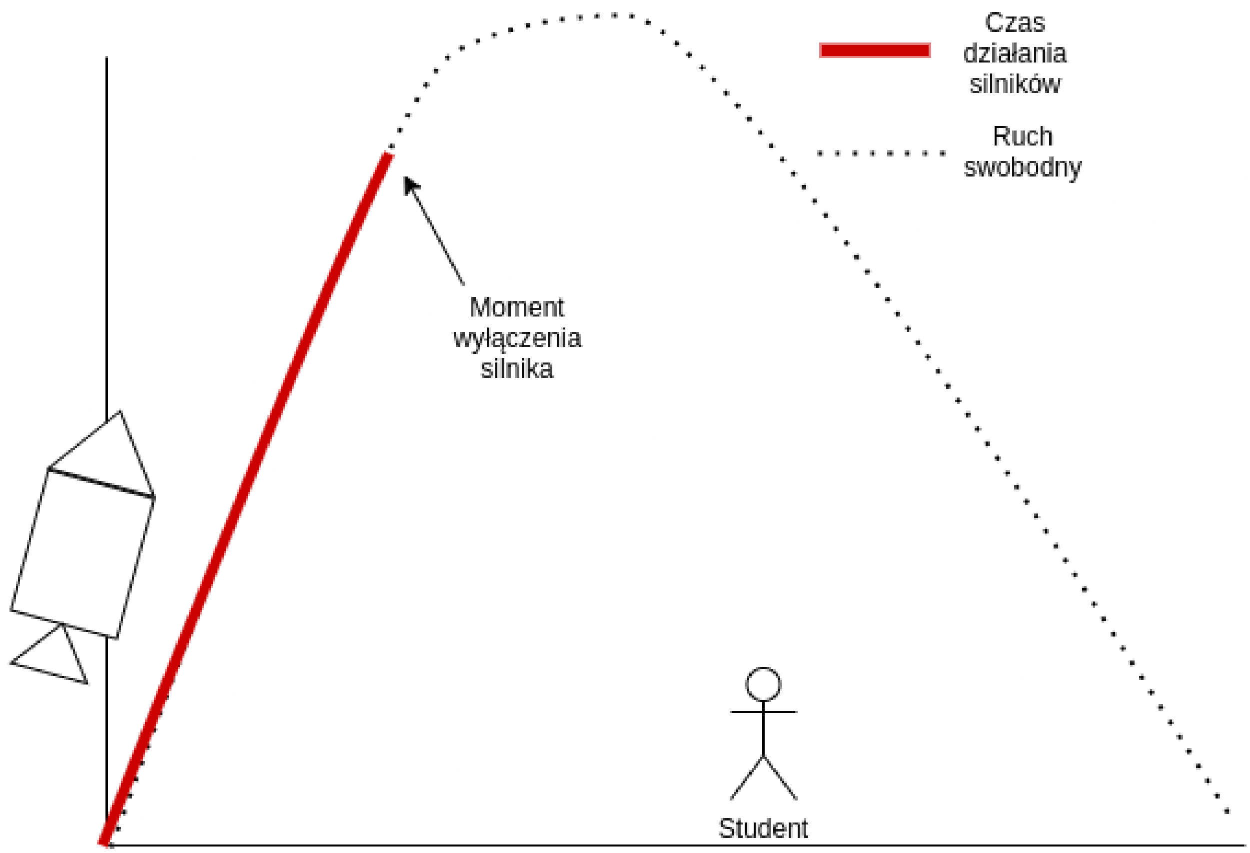
Spis treści

1 Treść zadania

Grupa studentów pracuje nad modelem rakiety na paliwo stałe. Ich zadaniem jest wysłanie ładunku na jak największą wysokość, dostosowując tempo spalania paliwa (czas pracy silnika). Studenci mają do dyskozycji rakietę o nasępujących parametrach:

- $Isp = 150$ - impuls właściwy silnika
- $m_o = 500$ kg - masa rakiety (paliwo + masa ładunku)
- $m_{final} = 100$ kg - masa ładunku

Dla uproszczenia przyjęto Isp jest stałe, niezależne od czasu działania silnika oraz, że rakieta osiągnie maksimum wysokości gdy wysokość na której wyłączy silniki będzie maksymalna dla zadanych parametrów rakiety.



Rysunek 1: Tor lotu rakiety

Równanie[1], które opisuje na jakiej wysokości wyłączą się silniki wygląda następująco:

$$h(t) = g\left[-t \cdot Isp \cdot \frac{ln(\frac{m_o}{m_{final}})}{\left(\frac{m_o}{m_{final}} - 1\right)} + t \cdot Isp - \frac{1}{2} \cdot t^2\right]$$

2 Rozwiązanie

Podane równanie jest jedynie przybliżeniem, gdyż nie uwzględnia tarcia, efektów pojawiających się przy przekroczeniu prędkości dźwięku itp. Dlatego do rozwiązania nie wykorzystam faktu iż jest to równanie

[1] https://web.mit.edu/16.unified/www/SPRING/propulsion/notes/node103.html

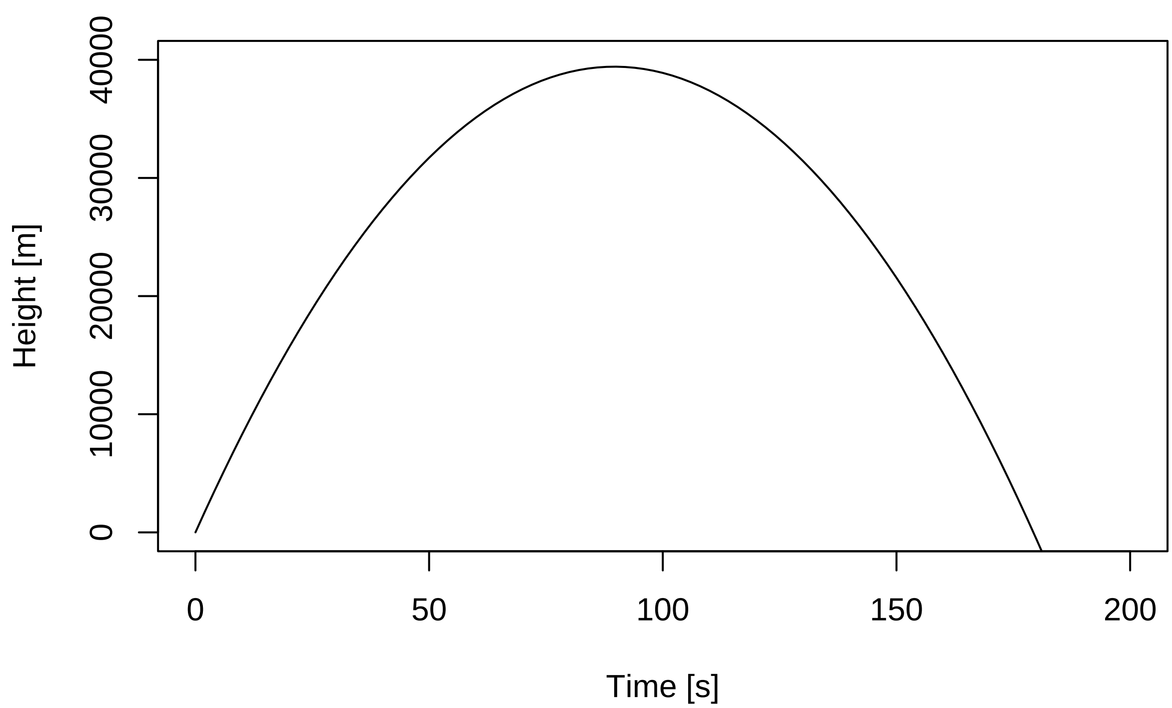
paraboli, a rozwiążę je z pomocą algorytmów Fibonacciego oraz *NAZWA DRUGIEGO ALGORYTMU*

```
# Inicjalizacja parametrów
Isp <- 150
m_o <- 500
m_final <- 100
g <- 9.81

# Deklacacja badanej funkcji
h <- function(t) {
  g*(
    - t * Isp * log(m_o/m_final)/(m_o/m_final - 1)
    + t * Isp
    - 1/2 * t^2
    )
}
```

Dziedziną tej funkcji jest $x \in [0, \inf)$, ponieważ czas nie może być < 0. Wykres tej funkcji prezentuje się następująco:

```
plot(h, xlim = c(0, 200), ylim = c(0, 40000), xlab = "Time [s]", ylab = "Height [m]")
```



Rysunek 2: Wykres funkcji h(t)

Maximum znajduje się w przedziale $[0, 150]$.

2.1 Metoda Fibonacciego

Metoda ta wymaga korzysta z liczb Fibonacciego, aby zoptymalizować algorytm zostaną one obliczone metodą iteracyjną i przechowywane będą w tablicy.

```
# Stworzenie tablicy zawierającej ciąg Fibonacciego
phi <- c(rep(0, 100))
```

```r
phi[1:3] <- c(1,1,1)
for(i in c(3:length(phi))) {phi[i] = phi[i-1] + phi[i-2]}

# Obliczanie K
fib_k <-function(lower, upper, tol) {
  i <- 1
  l <- upper - lower
  while(phi(i) < l / tol) {
    i <- i + 1
  }
  return(i)
}

fibonacci <- function(f,lower, upper, tol){
  k <- fib_k(lower, upper, tol) + 1
  c <- upper - phi(k-1)/phi(k)*(upper - lower)
  d <- lower + upper -c
  i <- 0
  cat("iteracja=", i, "a=", lower, "c=",c,"d=",d,"b=", upper ,
      "beta=", phi(k-1-i)/phi(k-i),"\n",sep=" ")
  while (i <= k-4) {
    if (f(c) < f(d)) {
      upper <- d
      i <- i + 1
    } else {
      lower <- c
      i <- i + 1
    }
    c <- upper - phi(k-1)/phi(k)*(upper - lower)
    d <- lower + upper - c
    cat("iteracja=", i, "a=", lower, "c=",c,"d=",d,"b=", upper ,
        "beta=", phi(k-1-i)/phi(k-i),"\n",sep=" ")
  }
  return((lower + upper)/2)
}
```