

Wykład na Politechnice Krakowskiej  
w dniu 18 stycznia 2012 r.

ZŁOŻONOŚĆ OBLICZENIOWA  
ZADAŃ I ALGORYTMÓW  
W OPTYMALIZACJI DYSKRETNEJ

dr hab. Krzysztof SZKATUŁA, prof. PAN  
Instytut Badań Systemowych PAN  
Uniwersytet Przyrodniczo-Humanistyczny

**Zadanie optymalizacyjne**

Rozważmy zadanie optymalizacyjne (zwane też zadaniem programowania matematycznego) w jednej z dwóch postaci:

$$Z_{OPT}(n) = \max_{X \in \mathcal{F}} Z(X) \quad \text{lub} \quad Z_{OPT}(n) = \min_{X \in \mathcal{F}} Z(X) \quad (1)$$

gdzie  $Z(X)$  - funkcja celu,  $\mathcal{F}$  - zbiór rozwiązań dopuszczalnych.

Powszechnie przyjmuje się, że rozwiązanie dopuszczalne  $X$  jest wektorem liczb rzeczywistych,  $X = [x_1, x_2, \dots, x_n]$ ,  $X \in R^n$ . Zbiór rozwiązań dopuszczalnych  $\mathcal{F}$  jest na ogół określany poprzez pewien układ warunków, na przykład nierówności lub równań. Zbiór  $\mathcal{F}$  można przykładowo zapisać w postaci:

$$\begin{aligned} F_1(X) &\leq 0 \\ F_2(X) &\leq 0 \\ &\vdots \\ F_m(X) &\leq 0. \end{aligned}$$

Postać funkcji  $Z(X)$ ,  $F_1(X), \dots, F_m(X)$ , rozstrzyga z jakim zadaniem optymalizacyjnym mamy do czynienia. Często przyjmuje się, że

$$Z(X), F_1(X), \dots, F_m(X) : R^n \rightarrow R.$$

Rozwiązanie dopuszczalne  $X^*$ ,  $X^* \in \mathcal{F}$ , dla którego

$$Z(X^*) = \max_{X \in \mathcal{F}} Z(X) \quad \text{lub} \quad Z(X^*) = \min_{X \in \mathcal{F}} Z(X),$$

nazywamy rozwiązaniem optymalnym, a  $Z(X^*)$ ,  $Z_{OPT}(n) = Z(X^*)$ , jego wartością.

Przyjmując założenie, że funkcja celu i lewe strony ograniczeń są funkcjami rzeczywistymi zmiennych rzeczywistych możemy sformułować zadania:

- Programowania (mieszanego) całkowitoliczbowego:

$$z_{OPT}(n) = \max \left\{ \sum_{i=1}^n c_i \cdot x_i + \sum_{l=1}^p h_l \cdot y_l \right\} \quad (\text{funkcja celu})$$

przy ograniczeniach  $\sum_{i=1}^n a_{ji} \cdot x_i + \sum_{l=1}^p g_{jl} \cdot y_l \leq b_j; \quad j = 1, \dots, m,$  (2)

gdzie  $x_i, y_l \geq 0$ ,  $x_i$  - całkowite,  $i = 1, \dots, n$ ,  $l = 1, \dots, p$ ,

- Programowania binarnego

$$z_{OPT}(n) = \max \sum_{i=1}^n c_i \cdot x_i$$

przy ograniczeniach  $\sum_{i=1}^n a_{ji} \cdot x_i \leq b_j(n)$  (3)

gdzie  $j = 1, \dots, m$ ,  $x_i = 0$  lub  $1$ ,  $i = 1, \dots, n$ .

- Wielowymiarowe zadanie załadunku, jeżeli w (3) dodatkowo

$$c_i, a_{ji}, b_j \geq 0.$$

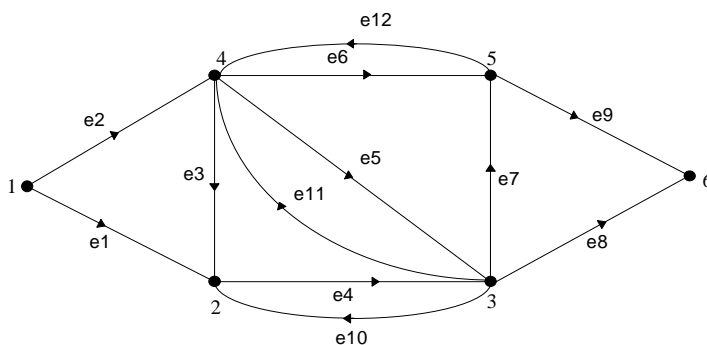
- Zadanie szeregowania prac z terminami zakończenia. Dla każdej pracy  $i$  są określone: *czas wykonania*  $t_i$ , *termin zakończenia*  $d_i(n)$  oraz *zysk*  $p_i$  z wykonania pracy w terminie. Zadanie optymalizacyjne polega na maksymalizacji ogólnego zysku z wykonania prac w terminie. Jeżeli  $d_1(n) \leq d_2(n) \leq \dots \leq d_n(n)$  to zadanie możemy zapisać w postaci wielowymiarowego zadania załadunku (3):

$$z_{OPT}(n) = \max \sum_{j=1}^n p_j x_j$$

przy ograniczeniach  $\sum_{j=1}^i t_j x_j \leq d_i(n), \quad i = 1, \dots, n$

gdzie  $x_j = 0$  lub  $1, \quad j = 1, \dots, n.$

- *Zadanie komiwojżera* polega na wyznaczeniu takiej drogi rozpoczynającej i kończącej się w zadanym wierzchołku grafu ważonego  $G$  i zawierającej każdy wierzchołek grafu dokładnie raz, że łączna jej długość jest minimalna.



Rysunek 1: Graf skierowany

## Algorytmy optymalizacyjne

Powszechnie przyjmuje się, że nadrzędnym celem każdego algorytmu rozwiązującego zadanie optymalizacyjne jest odnalezienie jego rozwiązania optymalnego  $X^*$  oraz wyznaczenie jego wartości  $Z_{OPT}(n)$ .

Algorytmem optymalizacyjnym, oznaczonym jako  $A$ , będziemy nazywać procedurę obliczeniową, która wyznacza pewne rozwiązanie  $X'$  zadania (1) z wartością rozwiązania  $Z_A(n) = Z(X')$ .

- Algorytm  $A$ , dla którego zawsze  $X' = X^*$  oraz  $Z_A(n) = Z_{OPT}(n)$  będziemy nazywali *algorytmem dokładnym*.
- Algorytm  $A$ , który wyznacza rozwiązanie zadania:  $X'$ , gdzie zawsze

$$X' \in \mathcal{F} \quad \text{oraz} \quad Z_A(n) \leq Z_{OPT}(n) \quad \text{lub} \quad Z_A(n) \geq Z_{OPT}(n),$$

nazywamy *algorytmem przybliżonym*.

Jeżeli dla pewnej stałej  $\varepsilon \geq 0$  zachodzi

$$|Z_A(n) - Z_{OPT}(n)| \leq \varepsilon,$$

albo, gdzie dodatkowo:  $Z_A(n), Z_{OPT}(n) > 0, 0 \leq \varepsilon \leq 1$ ,

$$1 - \varepsilon \leq \frac{Z_A(n)}{Z_{OPT}(n)} \leq 1 \quad \text{lub} \quad 1 - \varepsilon \leq \frac{Z_{OPT}(n)}{Z_A(n)} \leq 1,$$

to będziemy mówić, że algorytm przybliżony ma *gwarancję dokładności działania*. Inaczej jest on nazywany *algorytmem heurystycznym*. Mogą również występować algorytmy, które nie gwarantują uzyskania dla wszystkich realizacji danych zadania rozwiązań dopuszczalnych, to znaczy, że mogą istnieć  $X' \notin \mathcal{F}$ .

## Metody rozwiązywania

- Metoda podziału i oszacowań, której podstawowym celem jest zawężenie pełnego przeglądu zbioru rozwiązań dopuszczalnych przy wykorzystaniu oszacowań wartości rozwiązania optymalnego.
- Programowanie dynamiczne, gdzie rozwiązanie optymalne zadań o większym rozmiarze jest uzyskiwane poprzez rozwiązania optymalne mniejszych zadań pomocniczych.
- Metody zachłanne, które mają za zadanie dokonywanie najlepszych lokalnie wyborów. Metody tego typu są często stosowane w praktyce. W niektórych przypadkach można udowodnić ich optymalność, w innych mogą one mieć gwarantowaną dokładność obliczeń. W przypadku dużej części zadań optymalizacji dyskretnej metody zachłanne są metodami heurystycznymi.
- Grupa metod lokalnej poprawy, gdzie zasadnicza idea sprowadza się do uzyskiwania w kolejnych iteracjach, w otoczeniu aktualnie rozpatrywanego rozwiązania przybliżonego zadania, innego rozwiązania przybliżonego, o lepszej wartości funkcji celu. Uzyskane rozwiązanie staje się nowym aktualnym rozwiązaniem. Spośród metod lokalnej poprawy w literaturze większą uwagę poświęcono przeszukiwaniu losowemu, symulowanemu wychładzaniu, przeszukiwaniu z zakazami oraz algorytmom genetycznym.
- Metody programowania liniowego, które są bardzo użyteczne jako narzędzie pomocnicze w rozwiązywaniu wielu zadań optymalizacji dyskretnej. Na szczególną uwagę zasługują tutaj różne modyfikacje metody sympleks, algorytm elipsoidalny oraz projekcyjny.

## Nakład obliczeń

Przyjmując, że współczynniki zadań są zawsze wymiernymi liczbami rzeczywistymi, praktycznie każde zadanie optymalizacyjne może być reprezentowane jako nieskończony zbiór różnych realizacji jego danych

$$\{p_1, p_2, \dots, p_i, \dots\}, p_i = [\beta_1, \beta_2, \dots, \beta_v], \text{ gdzie } \beta_i \in R, i = 1, \dots, v.$$

Wymierna liczba rzeczywista  $c$  może być zapisana jako dwie liczby całkowite  $a$  i  $b$ , np.  $c = a/b$  (lub  $c = a.b$ ). Każda liczba całkowita  $a$  może zostać w jednoznaczny sposób zapisana w poniższej postaci

$$a = \sum_{i=0}^k \alpha_i \cdot 2^i \text{ gdzie } \alpha_i = 0 \text{ lub } 1, i = 1, \dots, k, k \leq \log_2 a < k + 1.$$

Liczba niewymierna może być zapisana z zadaną dokładnością jako liczba wymierna. Oznacza to, że każdą liczbę rzeczywistą (a więc również wektor liczb rzeczywistych) można przedstawić jako wektor o składowych binarnych. Zatem każda realizacja danych zadania  $p_i$  może być zapisana w jednoznaczny sposób jako wektor o postaci:

$$p_i = [\delta_1, \delta_2, \dots, \delta_{l_i}], \text{ gdzie } \delta_j = 0 \text{ lub } 1, j = 1, \dots, l_i.$$

Rozmiar realizacji danych zdania  $p_i$  wynosi  $l_i$ , co oznaczamy  $l(p_i) = l_i$ .

Jeżeli współczynniki  $\beta_1, \beta_2, \dots, \beta_v$ , mają ograniczone wartości, np. istnieją stałe  $\gamma \geq 0, \theta > 0, \gamma \leq |\beta_i| \leq \theta$ , to wtedy pojęcie rozmiaru danych zadania można uprościć, wystarczy operować zagregowanymi wielkościami takimi jak np.  $n$  (liczba zmiennych) i  $m$  (liczba ograniczeń).

Niech  $A$  będzie algorytmem wyznaczającym w skończonej liczbie iteracji rozwiązanie dla każdej z realizacji danych zadania  $p_i$ . *Nakład obliczeń* wykonywanych przez algorytm  $A$  może być wyrażony jako liczba elementarnych działań (takich jak dodawanie, odejmowanie, porównywanie, mnożenie lub dzielenie dwóch liczb) wykonywanych przez algorytm  $A$  w celu uzyskania rozwiązania  $p_i$ .

Przyjmijmy, że nakład obliczeń wykonywanych przez algorytm  $A$  jest zadany w postaci poniższej funkcji:

$$g_A(p_i) : \{p_1, p_2, \dots, p_i, \dots\} \rightarrow R_+.$$

Naszym celem jest uzyskanie oszacowania nakładu obliczeń wymaganego przez algorytm jako funkcji rozmiaru realizacji danych zadania  $l_i = l(p_i)$ , w taki sposób, aby ocena ta nie zależała od konkretnej postaci  $p_i$ . Warunek ten spełnia poniższa funkcja :

$$f_A(n) = \sup\{g_A(p_i) : l(p_i) = n\}.$$

Funkcja  $f_A(n)$  jest oszacowaniem od góry wymaganego przez algorytm  $A$  nakładu obliczeń dla każdej realizacji danych zadania  $p_i$ , o rozmiarze  $n$ ,  $i \in N$ . Ocena ta jest ważna dla wszystkich wartości  $n$ ,  $n \in N$ . Funkcja  $f_A(n)$  wyznacza *złożoność obliczeniową algorytmu  $A$* .

Dla dowolnych funkcji  $f(n), g(n) \geq 0, n \in N$ , będziemy oznaczać

$$f(n) = O(g(n)) \quad \text{jeżeli} \quad f(n) \leq c' \cdot g(n), \quad n \geq n';$$

$$f(n) = \Omega(g(n)) \quad \text{jeżeli} \quad c'' \cdot g(n) \leq f(n), \quad n \geq n'';$$

gdzie  $c', n' > 0$  oraz  $c'', n'' > 0$  stałe. Ponadto

$$f(n) = \Theta(g(n)) \quad \text{jeżeli} \quad f(n) = O(g(n)) \text{ oraz } f(n) = \Omega(g(n)).$$

Zgodnie z powyższą notacją dla  $f(n) = \sum_{i=0}^k a_i \cdot n^i$ ,  $a_k > 0$ , (funkcji wielomianowej) mamy

$$f(n) = O(n^k), f(n) = \Omega(n^k) \text{ oraz } f(n) = \Theta(n^k).$$

- Algorytm  $A$  dla zadania optymalizacyjnego będziemy nazywać *algorytmem o wielomianowej złożoności obliczeniowej*, jeżeli

$$f_A(n) = O(n^k), \text{ gdzie } k \geq 1 \text{ jest stałą.}$$

- Algorytm  $A$  dla zadania optymalizacyjnego będziemy nazywać *algorytmem o wykładniczej złożoności obliczeniowej*, jeżeli

$$f_A(n) = \Omega(\alpha^n) \text{ oraz } f_A(n) = O(\beta^n), \text{ gdzie } \alpha, \beta > 1 \text{ są stałymi.}$$

W przypadku algorytmów przybliżonych z gwarantowaną dokładnością  $\varepsilon$  w ocenie ich złożoności obliczeniowej często oprócz rozmiaru danych zadania, oznaczonego jako  $n$ , uwzględniana jest wartość  $1/\varepsilon$ .

- Algorytm przybliżony  $A$  uważany jest za *efektywny obliczeniowo* dla zadanej wartości dokładności  $\varepsilon$ , jeżeli  $f_A(n) = O(n^k \cdot \ln(\delta/\varepsilon))$ , gdzie  $k \geq 1, \delta > 0$  są stałymi.
- Algorytm przybliżony  $A$  nazywamy *wielomianowym schematem aproksymującym*, jeżeli dla dowolnego  $0 \leq \varepsilon \leq 1$  uzyskuje on rozwiązanie przybliżone zadania o gwarantowanej dokładności  $\varepsilon$  oraz  $f_A(n) = O(n^k)$ , gdzie  $k \geq 1$  jest stałą.

- Algorytm przybliżony  $A$  nazywamy *w pełni wielomianowym schematem aproksymującym*, jeżeli dla dowolnej wartości  $0 \leq \varepsilon \leq 1$  uzyskuje on rozwiązanie przybliżone zadania o gwarantowanej dokładności  $\varepsilon$  oraz  $f_A(n)$  jest ograniczona przez funkcję wielomianową od  $n$  oraz  $1/\varepsilon$ .

W przypadku algorytmów złożonych, składających się w istocie z różnych procedur obliczeniowych, ocenie może podlegać złożoność obliczeniowa każdej ze składowych rozważanego algorytmu.

### Złożoność zadań optymalizacyjnych

Problem *rozstrzygalności* dla zadania  $X$  stanowi para  $(D, V)$  gdzie  $V \subseteq D$  oraz elementy zbioru  $D$  są wektorami binarnymi. Zbiór  $D$  będziemy nazywać *zbiorem realizacji rozwiązań* dla zadania  $X$ , natomiast  $V$  będziemy nazywać *zbiorem dopuszczalnych realizacji rozwiązań* dla zadania  $X$ . Dla danego  $d \in D$  chcemy znać odpowiedź na pytanie, czy  $d \in V$ . Dla każdego  $d \in D$  odpowiedź brzmi tak lub nie.

Pomiędzy zadaniami optymalizacyjnymi i problemami decyzyjnymi istnieje bezpośredni związek.

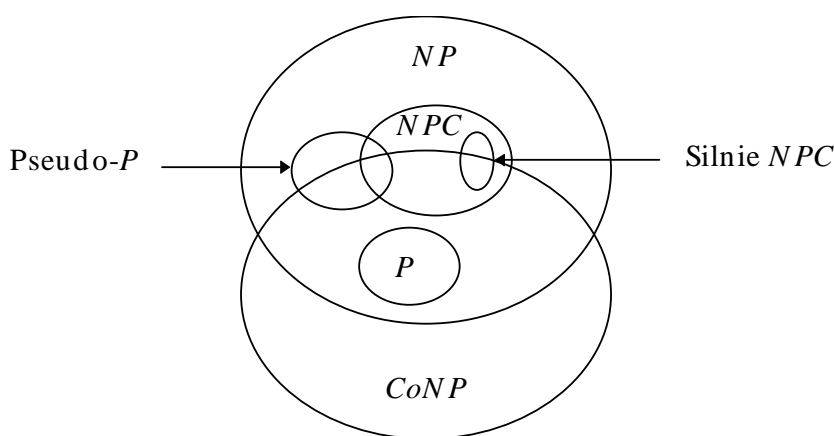
- Klasa problemów rozstrzygalności, gdzie dla każdego  $d \in V$  istnieje wielomianowy algorytm rozstrzygający fakt, że  $d \in V$ , nazywana jest klasą  $\mathcal{P}$ .
- Klasa problemów rozstrzygalności, gdzie dla każdego  $d \in V$  istnieje wielomianowy, niedeterministyczny algorytm rozstrzygający fakt, że  $d \in V$ , nazywana jest klasą  $\mathcal{NP}$ .
- Klasa  $\mathcal{CoNP}$  jest to klasa problemów rozstrzygalności  $X$  takich, że ich dopełnienia należą do klasy  $\mathcal{NP}$ , co można zapisać w poniższy sposób:

$$\mathcal{CoNP} = \{X : \bar{X} \in \mathcal{NP}\}, \text{ gdzie } \bar{X} = (\bar{V}, D).$$

- Problem rozstrzygalności  $X \in \mathcal{NP}$  będziemy nazywać  *$\mathcal{NP}$ -zupełnym*, jeżeli wszystkie problemy z klasy  $\mathcal{NP}$  są wielomianowo sprowadzalne do problemu  $X$ . Klasę problemów  $\mathcal{NP}$ -zupełnych będziemy oznaczać  $\mathcal{NPC}$ .
- Jeżeli problem rozstrzygalności odpowiadający zadaniu optymalizacyjnemu jest  $\mathcal{NP}$ -zupełny, to zadanie optymalizacyjne będziemy nazywać  *$\mathcal{NP}$ -trudnym*.



- Algorytm wymagający wielomianowego nakładu obliczeniowego przy zastosowaniu unarnego kodowania danych zadania nazywamy *algorytmem pseudowielomianowym*. Klasę takich zadań będziemy nazywać Pseudo- $\mathcal{P}$ .
- Problemy  $\mathcal{NP}$ -zupełne, dla których istnienie dokładnych algorytmów pseudowielomianowych oznacza  $\mathcal{NP} = \mathcal{P}$ , będziemy nazywali *silnie  $\mathcal{NP}$ -zupełnymi* i oznaczać Silnie  $\mathcal{NPC}$ . Odpowiadające im zadania optymalizacyjne będziemy nazywać silnie  $\mathcal{NP}$ -trudnymi.



Relacje klas  $\mathcal{NP}$ ,  $\mathcal{NPC}$ ,  $Co\mathcal{NP}$ ,  $\mathcal{P}$ , Silnie  $\mathcal{NPC}$  i Pseudo- $\mathcal{P}$

Powszechnie uważa się, że w klasie  $\mathcal{NP}$  istnieją zadania nie należące do klasy  $\mathcal{P}$  a więc, że

$$\mathcal{P} \neq \mathcal{NP}.$$

Natomiast, jeżeli zachodzi relacja  $\mathcal{NP} = \mathcal{P}$ , to wtedy

$$\mathcal{NP} = \mathcal{NPC} = Co\mathcal{NP} = \text{Silnie } \mathcal{NPC} = \mathcal{P} = \text{Pseudo-}\mathcal{P}.$$

## Analiza przypadku średniego

Analiza przypadku średniego jest alternatywnym sposobem oceny zadań i algorytmów optymalizacyjnych. Jej przeprowadzenie wymaga przyjęcia założenia, że niektóre dane zadania są realizacjami pewnych zmiennych losowych.

W procesie praktycznego rozwiązywania realizacji danych zadania optymalizacyjnego o rozmiarach  $n$  przy wykorzystaniu algorytmu  $A$  operujemy, między innymi, wielkościami takimi jak:

$$z_{OPT}(n), \quad z_A(n) \quad \text{oraz} \quad f_A(n).$$

Charakterystyki te stają się realizacjami pewnych zmiennych losowych, których zachowanie może być badane i poddane analizie.

Analiza przypadku średniego nosi na ogół charakter asymptotyczny. Zakładamy, że rozmiar zadania, dla ustalenia uwagi oznaczany jako  $n$ , wzrasta w sposób nieograniczony,  $n \rightarrow \infty$ . Analizowane jest graniczne zachowanie się ciągów zmiennych losowych opisujących takie wielkości jak np.  $z_{OPT}(n)$ ,  $z_A(n)$  oraz  $f_A(n)$  przy  $n \rightarrow \infty$ .

## Zadanie programowania liniowego

Do rozwiązywania zadań programowania liniowego powszechnie stosowany jest algorytm sympleks. W sensie analizy przypadku najgorszego ma on wykładniczą złożoność obliczeniową.

Jednym z bardzo istotnych wyników analizy przypadku średniego było wykazanie, że istnieją realizacje algorytmu sympleks mające w średnim przypadku wielomianową złożoność obliczeniową. Wynik ten potwierdził wysoką przydatność praktyczną algorytmu sympleks.

Algorytm elipsoidalny zaproponowany przez Chacziana ma wielomianową złożoność obliczeniową. Ten wynik pozwolił zakwalifikować zadanie programowania liniowego do klasy  $\mathcal{P}$ . Dużą przydatność praktyczną ma algorytm Karmakara i metody pokrewne (metody punktu wewnętrznego).

W ogólnym przypadku zadań optymalizacji ciągłej, szczególnie wieloekstremalnych, nie istnieją efektywne obliczeniowo metody dla rozwiązywania wszystkich zadań. Natomiast dla licznych zadań **programowania wypukłego** istnieją metody efektywne obliczeniowo.

## Zadanie komiwojażera

Zadanie komiwojażera jest  $\mathcal{NP}$ -trudnym zadaniem optymalizacji dyskretnej.

W klasycznej pracy Beardwood, Halton i Hammersley, rozważono nieskierowany graf ważony  $G$ , dla którego  $V = \{1, \dots, n\}$  - zbiór wierzchołków jest zadany

jako zbiór  $n$  punktów w kwadracie o bokach równych 1. Każdy z wierzchołków grafu  $i \in V$  jest zdefiniowany poprzez jego współrzędne  $(x_i, y_i)$ , gdzie  $0 \leq x_i, y_i \leq 1$ . Dla każdej krawędzi lub łuku  $e = (i, j)$  grafu  $G$  waga  $c_{ij}$  jest zdefiniowana jako najkrótsza odległość pomiędzy wierzchołkami:

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$$

gdzie  $x_1, x_2, \dots, x_n$  oraz  $y_1, y_2, \dots, y_n$  są realizacjami  $X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n$  - wzajemnie niezależnych zmiennych losowych o rozkładzie równomiernym w przedziale  $(0, 1]$ . Wtedy:

$$\lim_{n \rightarrow \infty} \frac{z_{OPT}(n)}{\sqrt{n}} = \beta \quad (\text{p.n.}), \quad \text{gdzie } \beta \text{ stała, } \beta \sim 0,765.$$