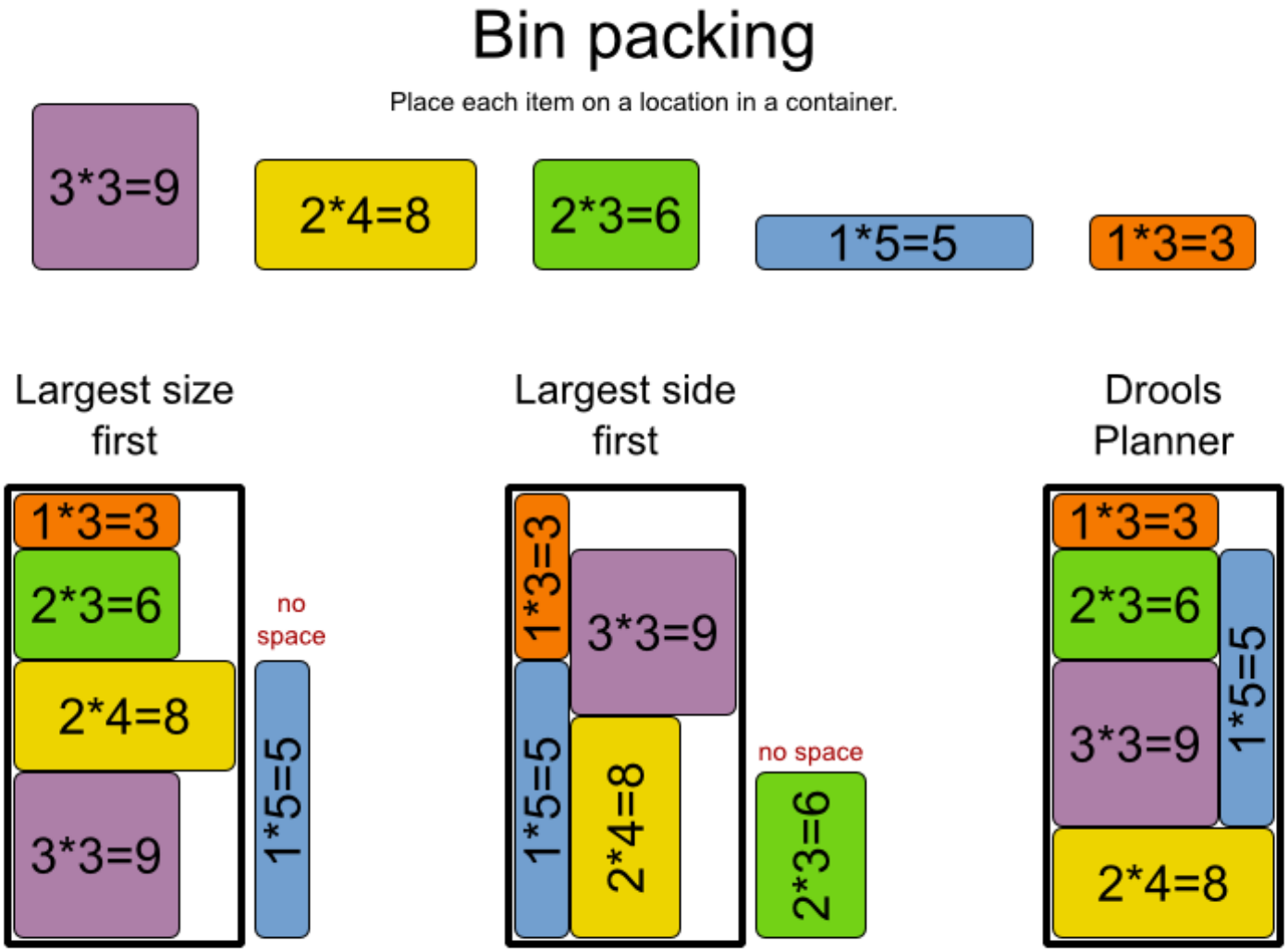
Optymalizacja

Marcin Jukiewicz Andrzej Gajda

Sztuczna Inteligencja

27.02.2019

Optymalizacja


Bin packing
Place each item on a location in a container.
3*3=9
2*4=8
2*3=6
1*5=5
1*3=3
Largest size first
1*3=3
2*3=6
2*4=8
3*3=9
no space
1*5=5
Largest side first
1*3=3
3*3=9
1*5=5
2*4=8
no space
2*3=6
Drools Planner
1*3=3
2*3=6
3*3=9
1*5=5
2*4=8

Problem optymalizacyjny

Jest to problem obliczeniowy, polegający na znalezieniu wartości maksymalnej, bądź minimalnej określonego parametru zadanego problemu.

Problem optymalizacyjny

Jest to problem obliczeniowy, polegający na znalezieniu wartości maksymalnej, bądź minimalnej określonego parametru zadanego problemu.

Instancja problemu

Jest to wystąpienie danego problemu optymalizacyjnego. Konkretny przypadek problemu optymalizacyjnego.

Problem optymalizacyjny

Jest to problem obliczeniowy, polegający na znalezieniu wartości maksymalnej, bądź minimalnej określonego parametru zadanego problemu.

Instancja problemu

Jest to wystąpienie danego problemu optymalizacyjnego. Konkretny przypadek problemu optymalizacyjnego.

Przestrzeń rozwiązań

Jest to zbiór zawierający wszystkie elementy, które mogą być rozwiązaniem problemu.

Problem optymalizacyjny

Jest to problem obliczeniowy, polegający na znalezieniu wartości maksymalnej, bądź minimalnej określonego parametru zadanego problemu.

Instancja problemu

Jest to wystąpienie danego problemu optymalizacyjnego. Konkretny przypadek problemu optymalizacyjnego.

Przestrzeń rozwiązań

Jest to zbiór zawierający wszystkie elementy, które mogą być rozwiązaniem problemu.

Funkcja celu

Jest to zadanie (niekoniecznie funkcja), dla którego poszukiwane jest rozwiązanie optymalne.

Mamy do dyspozycji siatkę metalową, z której możemy zbudować ogrodzenie. Długość tej siatki to 48 m.

Jaki **największy prostokątny** obszar możemy ogrodzić taką siatką?

Mamy do dyspozycji siatkę metalową, z której możemy zbudować ogrodzenie. Długość tej siatki to 48 m.

Jaki **największy prostokątny** obszar możemy ogrodzić taką siatką?

x, y — boki prostokąta
$2x + 2y = 48$ — obwód prostokąta
xy — pole prostokąta

współrzędne wierzchołka paraboli: p, q, gdzie:

$$p = \frac{-b}{2a}$$

$$q = \frac{-\Delta}{4a}$$

Startujesz z własnym biznesem gastronomicznym. Na początek ograniczysz się do dwóch pozycji menu: pizza i zapiekanki. Twoja specjalności to ser i pieczarki. Do każdej zapiekanki potrzebujesz 150 g sera i 100 g pieczarek. Do każdej pizzy potrzebujesz 230 g sera i 80 g pieczarek. Środki pozwalają zakupić Ci 4 kg sera i 2 kg pieczarek. Pizza ma kosztować 13 zł, a zapiekana 8 zł. Jak zmaksymalizować zysk?

Pewien sklep zoologiczny zajmuje się hodowlą i sprzedażą chomików oraz świnek morskich, sprzedając je po 8 i 25 zł za sztukę. Aby utrzymać przez tydzień jednego chomika, sklep musi wykorzystać 150 g trocin, 90 g karmy, a także zapewnić mu miejsce w klatce o powierzchni 200 cm^2 . Natomiast w przypadku świnki morskiej to zużycie wynosi odpowiednio 100 g trocin, 300 g karmy i przynajmniej 1000 cm^2 powierzchni życiowej. Wiemy także, że sklep może sobie pozwolić, aby zużyć w tygodniu maksymalnie 2 kg trocin i 3 kg karmy oraz posiada klatki i akwaria o łącznej powierzchni 15000 cm^2 . Zakładając, że wszystkie zwierzęta znajdują od razu nowych właścicieli, należy znaleźć liczbę chomików i świnek morskich, która powinna być hodowana tygodniowo przez sklep zoologiczny aby maksymalizował swój przychód.

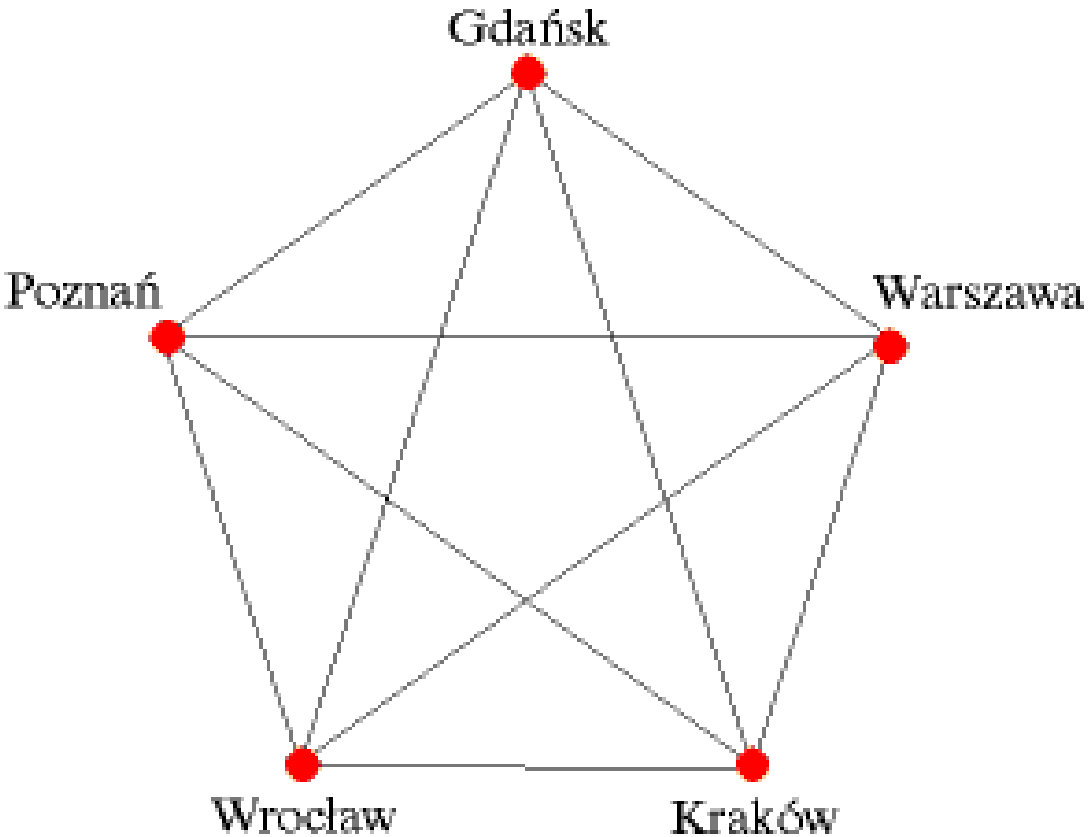
x_1 — liczba chomików
x_2 — liczba świnek morskich

Funkcja celu:

$$f(x_1, x_2) = 8x_1 + 25x_2$$

Ograniczenia:
$150x_1 + 100x_2 \leqslant 2000$
$90x_1 + 300x_2 \leqslant 3000$
$200x_1 + 1000x_2 \leqslant 15000$
$x_1 \geqslant 0$
$x_2 \geqslant 0$

max 15 kg		
radio	4 zł	12 kg
komputer	1 zł	2 kg
TV	3 zł	7 kg
telefon	2 zł	5 kg
kamień	1 zł	9 kg


Gdańsk
Poznań
Warszawa
Wrocław
Kraków

	Poznań	Warszawa	Gdańsk	Kraków	Wrocław	Toruń	Zielona G.
Poznań	1	310	333	444	168	161	139
Warszawa	310	1	358	294	363	212	450
Gdańsk	333	358	1	600	463	170	476
Kraków	444	294	600	1	270	441	429
Wrocław	168	363	463	270	1	298	156
Toruń	161	212	170	441	298	1	305
Zielona Góra	139	450	476	429	156	305	1

Obliczanie złożoności

Typy złożoności:

- $\mathcal{O}(1)$ stała
- $\mathcal{O}(n)$ liniowa
- $\mathcal{O}(log(n))$ logarytmiczna
- $\mathcal{O}(nlog(n))$ liniowo-logarytmiczna
- $\mathcal{O}(n^2)$ kwadratowa
- $\mathcal{O}(n^x)$ wielomianowa
- $\mathcal{O}(x^n)$ wykładnicza
- $\mathcal{O}(n!)$ typu silnia

Problem P

Problem decyzyjny, dla którego rozwiązanie można znaleźć w czasie wielomianowym.

Problem P

Problem decyzyjny, dla którego rozwiązanie można znaleźć w czasie wielomianowym.

Problem NP

Problem decyzyjny, dla którego rozwiązanie można zweryfikować w czasie wielomianowym.

Problem P

Problem decyzyjny, dla którego rozwiązanie można znaleźć w czasie wielomianowym.

Problem NP

Problem decyzyjny, dla którego rozwiązanie można zweryfikować w czasie wielomianowym.

Dlatego każdy problem P jest NP, przy czym nie wiadomo jednak czy istnieje problem NP niebędący P.

Dlaczego problemy mogą być trudne do rozwiązania?

- Problem jest bardzo złożony użycie modeli uproszczonych i rezultaty są bezużyteczne.
- Duża liczba możliwych rozwiązań przeszukanie całej przestrzeni rozwiązań dopuszczalnych w celu znalezienia najlepszego jest nierealne.
- Funkcja oceny jest obarczona niepewnością.
- Potencjalne rozwiązania mocno ograniczone już znalezienie jednego dopuszczalnego jest problemem.

Optimum lokalne

Optimum w określonym obszarze/zakresie/fragmencie problemu.

Optimum lokalne

Optimum w określonym obszarze/zakresie/fragmencie problemu.

Optimum globalne

Optimum dla całości analizowanego obszaru/zakresu/problemu.

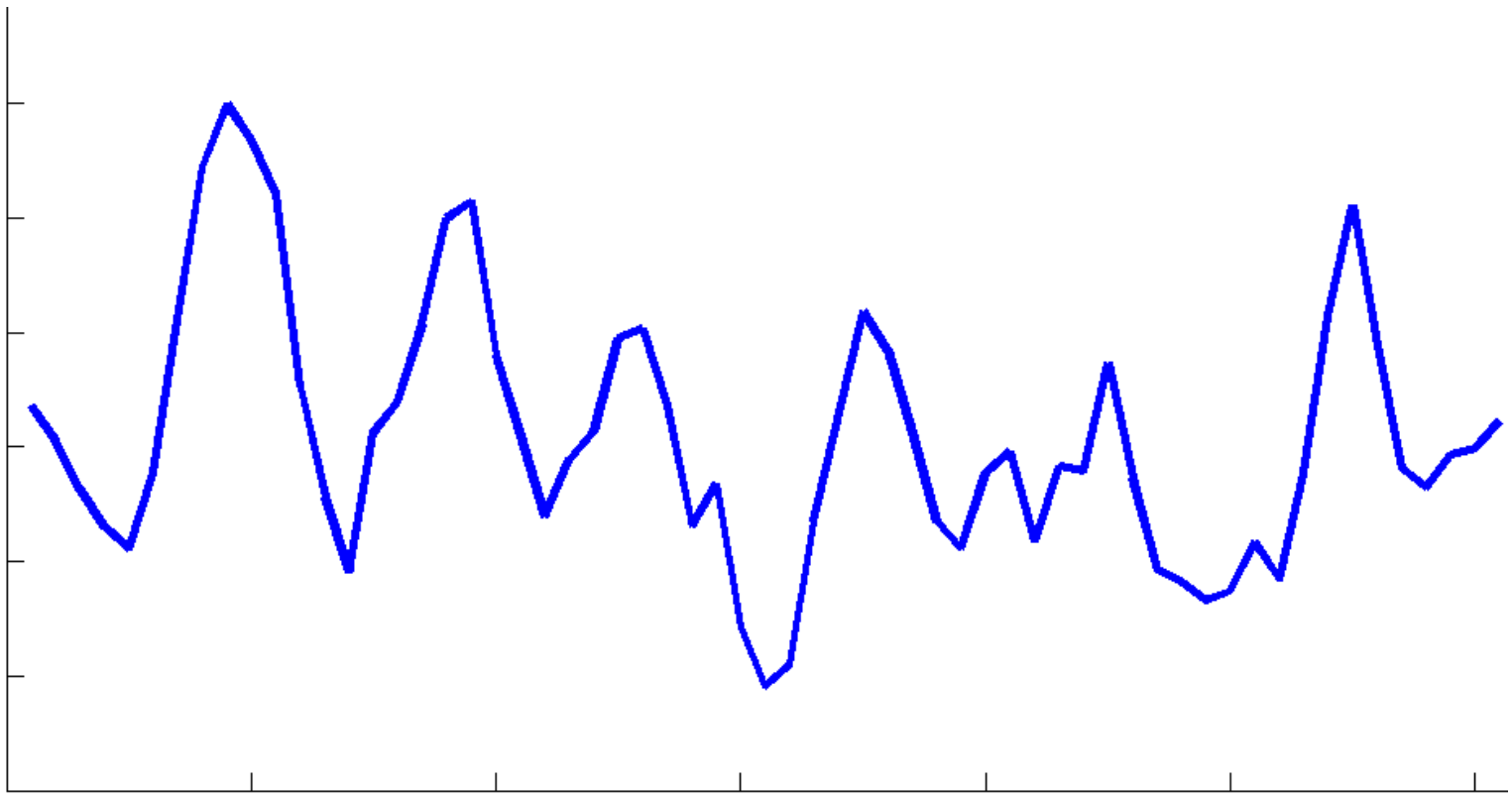
Optimum lokalne

Optimum w określonym obszarze/zakresie/fragmencie problemu.

Optimum globalne

Optimum dla całości analizowanego obszaru/zakresu/problemu.

Jaka jest relacja pomiędzy optimum lokalnym a globalnym?



Przykładem algorytmu dokładnego jest rozwiązanie kombinatoryczne problemu komiwojażera. W takim przypadku, rozwiązanie polega na wypisaniu wszystkich możliwych tras. Ile ich jest?

$$\frac{n!}{2n} = \frac{(n-1)!}{2}$$

Dla małej liczby punktów, wypisanie wszystkich możliwych rozwiązań nie wydaje się trudne, jednak problem się szybko komplikuje.

liczba punktów	liczba możliwych połączeń
4	3
5	12
6	60
7	360
8	2520
9	20160
10	181440
11	1814400
12	19958400

Heurystyki konstruujące (nie mylić z metaheurystykami) to typ algorytmów optymalizacji, które wykorzystują informacje zebrane dotychczas przez algorytm lub dane dostarczone z góry, aby wspomóc podjęcie decyzji, który potencjalne rozwiązanie powinno być testowane oraz jak stworzyć kolejne potencjalne rozwiązanie. Heurystyka jest zależna od rozwiązywanego problemu.

Artysta	Wynagrodzenie [zł]	Ocena krytyków
Maryla Rodowicz	25000	⋆⋆
Krzysztof Krawczyk	60000	⋆⋆⋆⋆⋆
Zenek Martyniuk	40000	⋆⋆⋆⋆
Edyta Górniak	40000	⋆⋆
Jan Kowalski	100	⋆

Metoda podziału i ograniczeń

Głównym założeniem metody podziału i ograniczeń (*branch-and-bound*) jest uporządkowane przeszukiwanie drzewa reprezentującego przestrzeń rozwiązań problemu oraz możliwość odcięcia gałęzi, które nie zawierają „dobrych" rozwiązań. Dzięki temu nie przeglądamy całego drzewa i możemy dzięki temu dzielić je i przeglądać tylko obiecujące obszary.

Ta metoda składa się z dwóch podstawowych procedur:

- rozgałęzianie (*branching*) — dzielenie zbioru rozwiązań reprezentowanego przez węzeł na rozłączne podzbiory, reprezentowane przez następników tego węzła
- ograniczanie (bounding) — pomijanie w przeszukiwaniu tych gałęzi drzewa, o których wiadomo, że nie zawierają optymalnego rozwiązania w swoich liściach

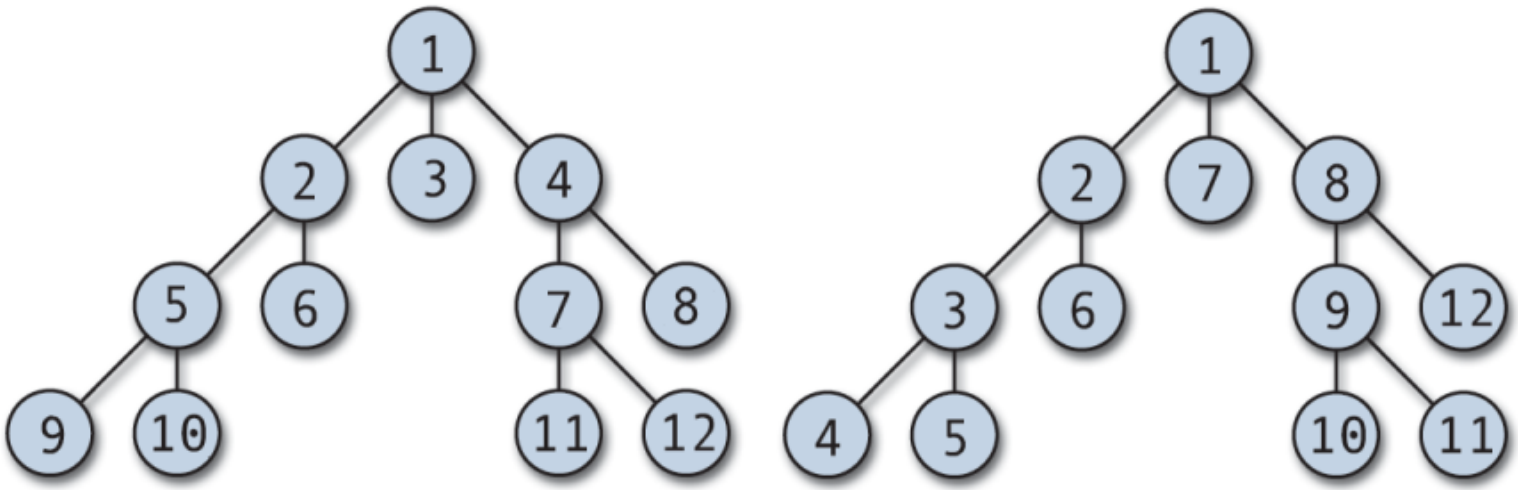
Ogólny schemat postępowania wygląda następująco:

1. zbiór wszystkich dopuszczalnych rozwiań jest dzielony na mniejsze podzbiory
2. dla każdego otrzymanego podzbioru obliczamy ograniczenie wartości funkcji celu
3. z dalszych podziałów eliminujemy podzbiory o ograniczeniu przekraczającym wartość funkcji celu

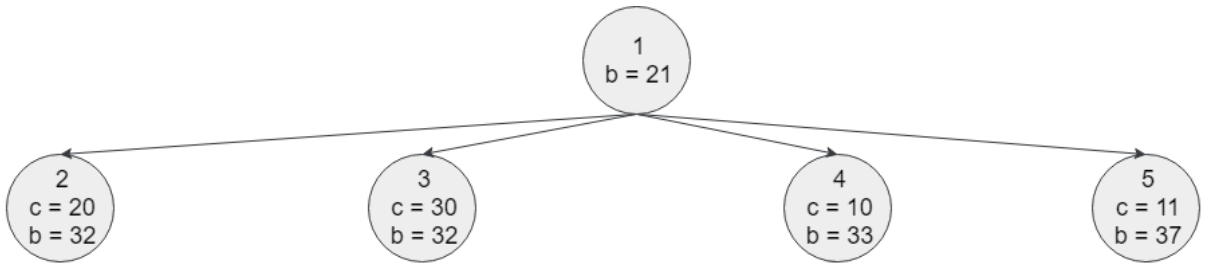
Metody przeszukiwania:

1. przeszukiwanie wszerz (*breadth-first search*) — polega na odwiedzeniu wszystkich osiągalnych wierzchołków z wierzchołka początkowego
2. przeszukiwanie w głąb (*depth-first search*) — polega na badaniu wszystkich krawędzi wychodzących z podanego wierzchołka
3. najpierw najlepszy (*best-first search*) — wybierane są te drogi, które wydają się prowadzić do najlepszego rozwiązania

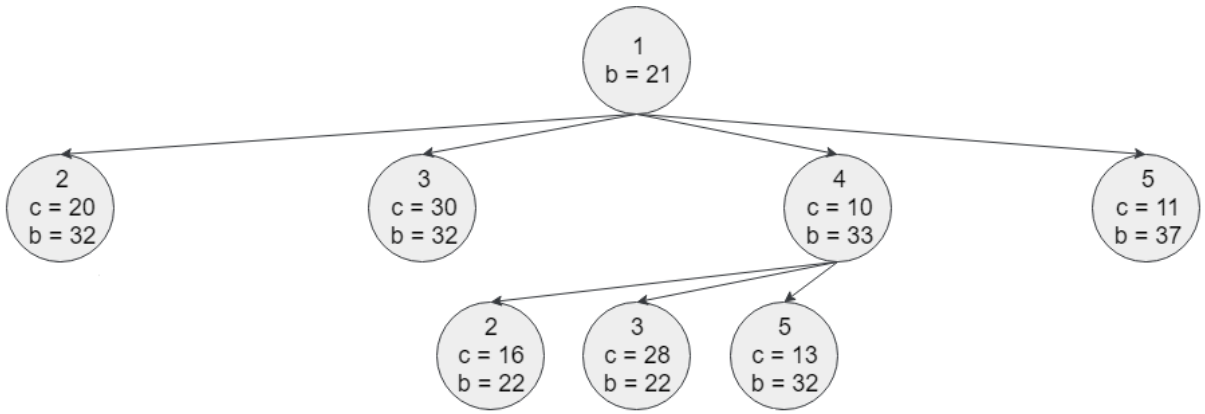
Przeszukiwanie wszerz i przeszukiwanie w głąb



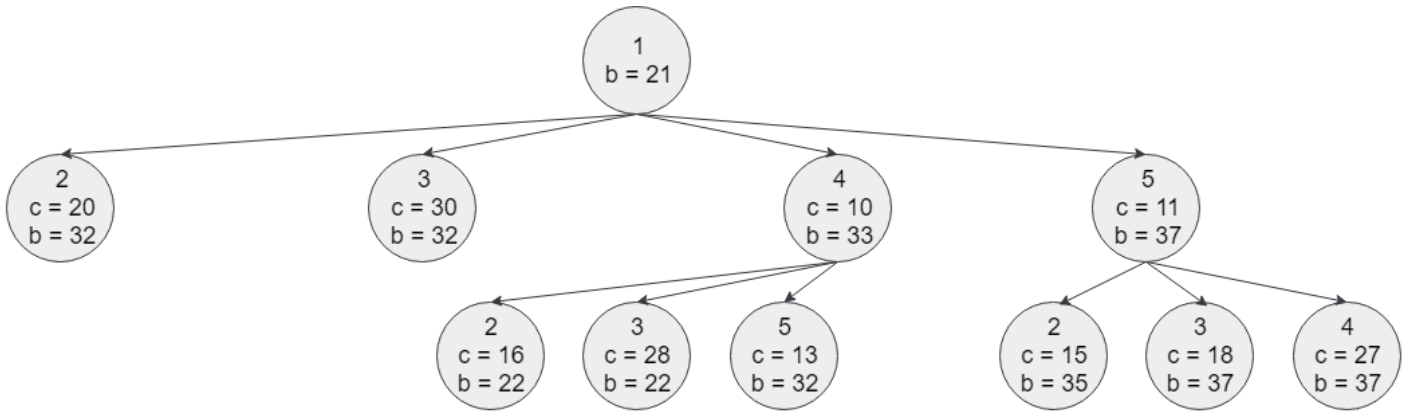
Metoda podziału i ograniczeń



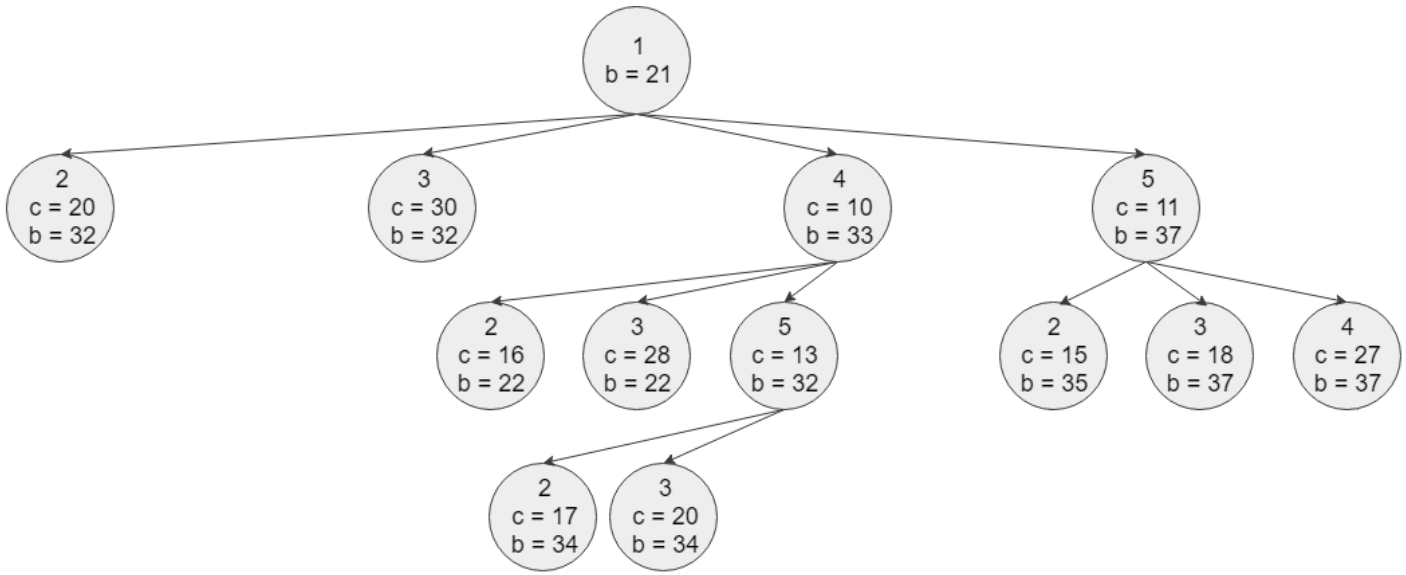
Metoda podziału i ograniczeń



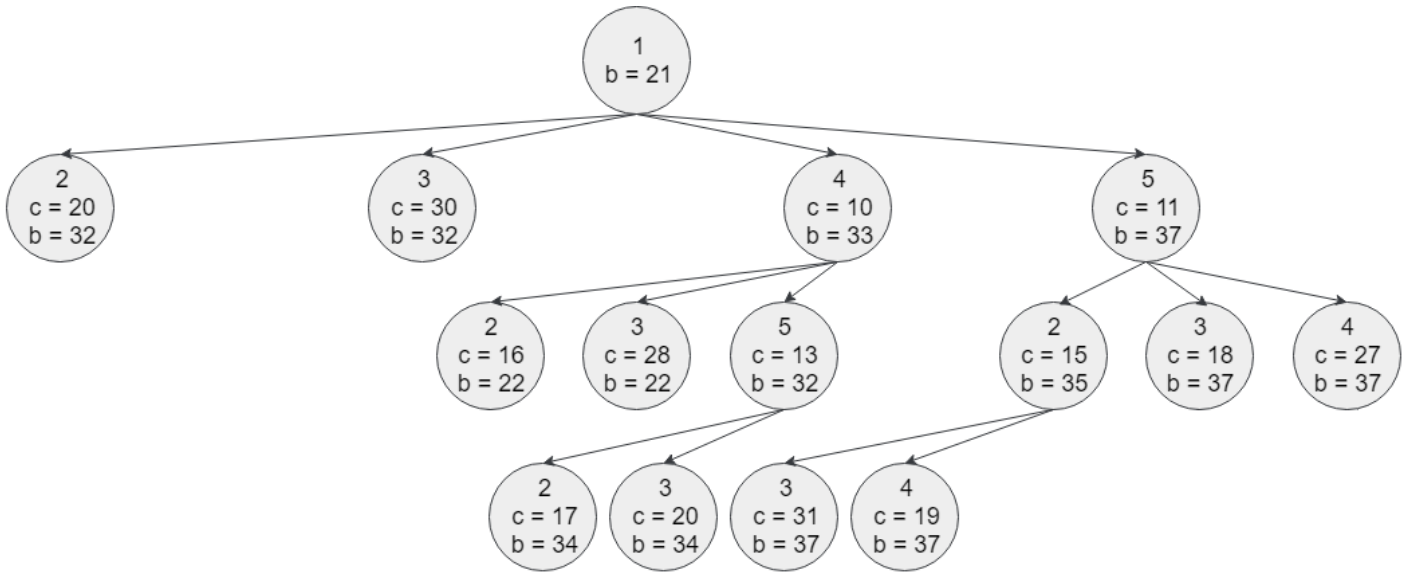
Metoda podziału i ograniczeń



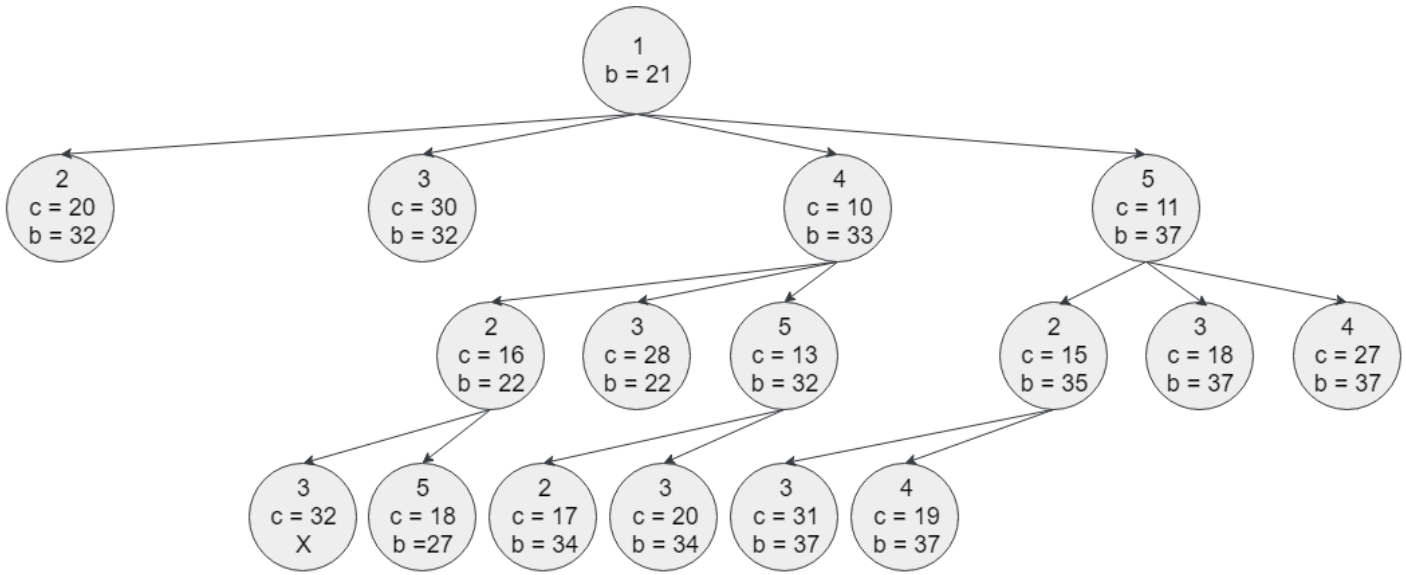
Metoda podziału i ograniczeń



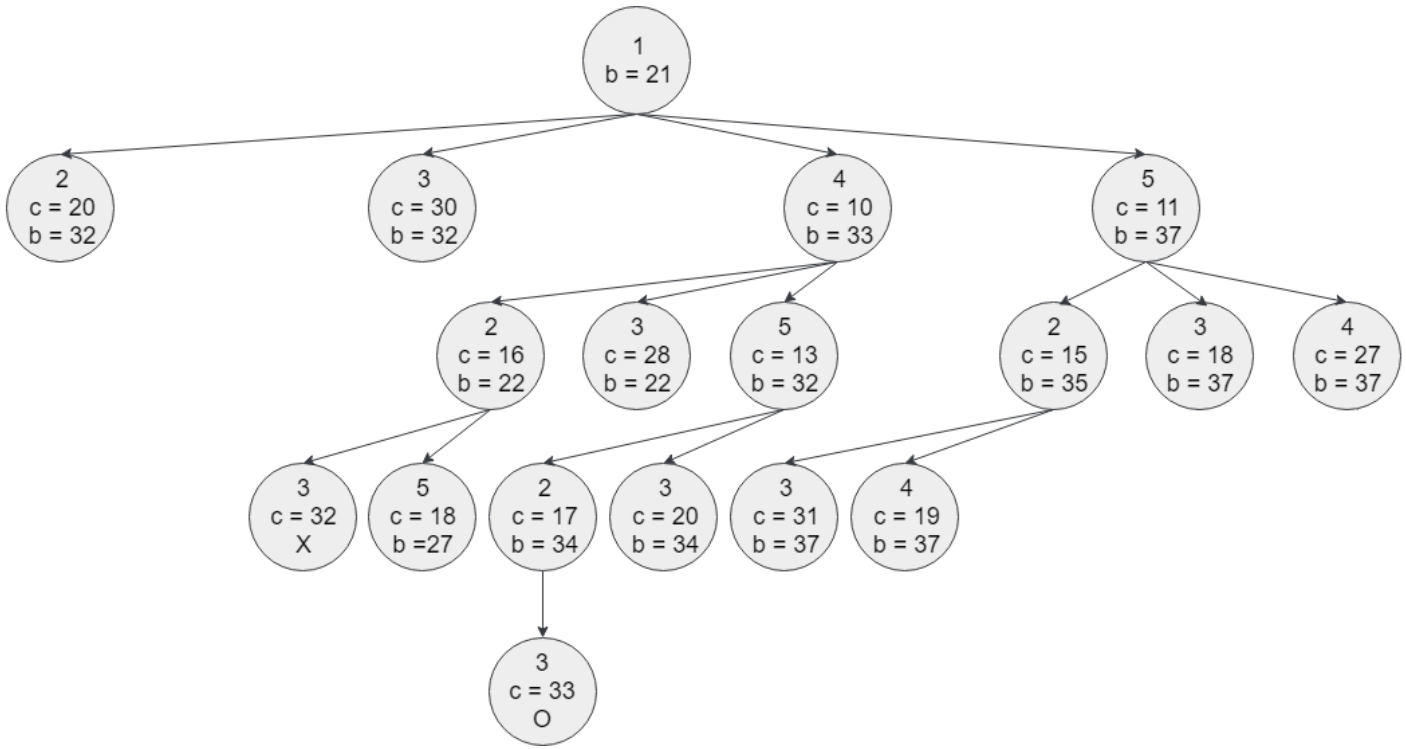
Metoda podziału i ograniczeń



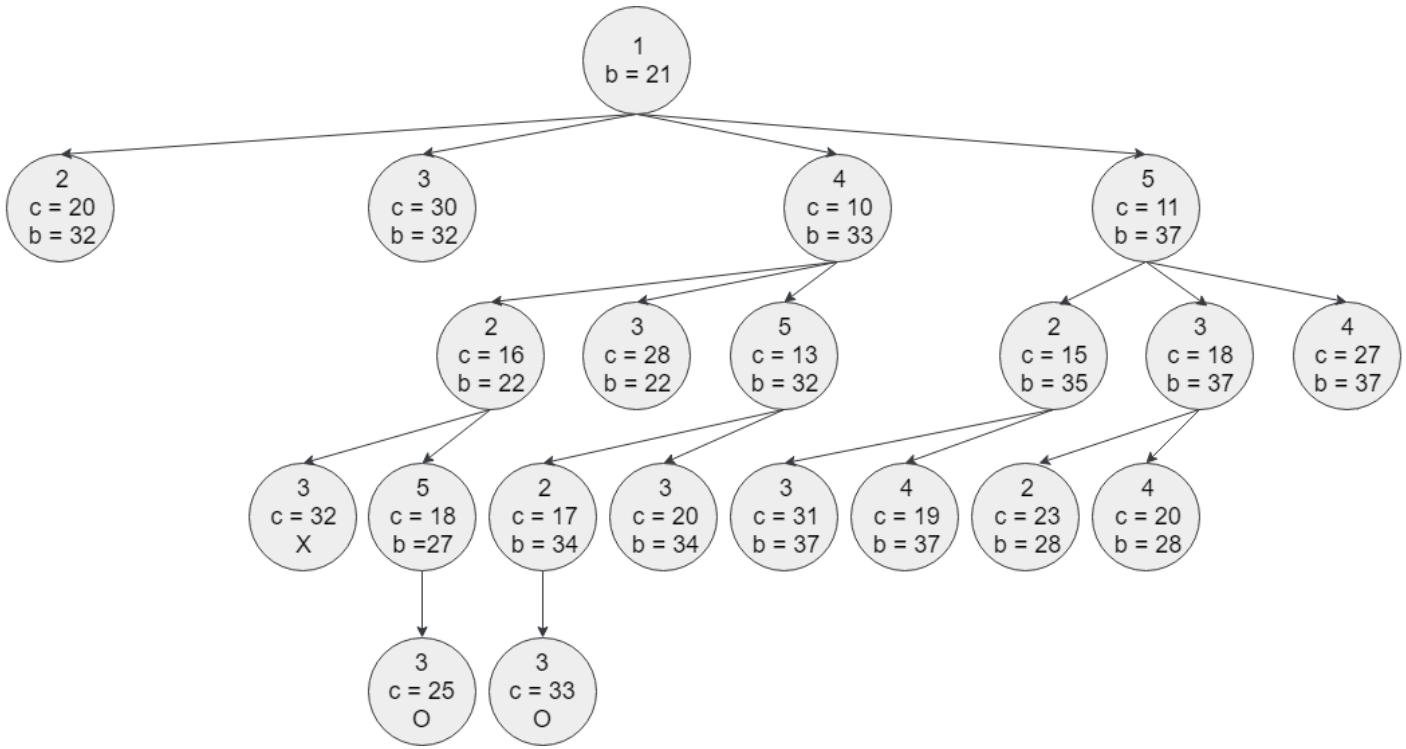
Metoda podziału i ograniczeń



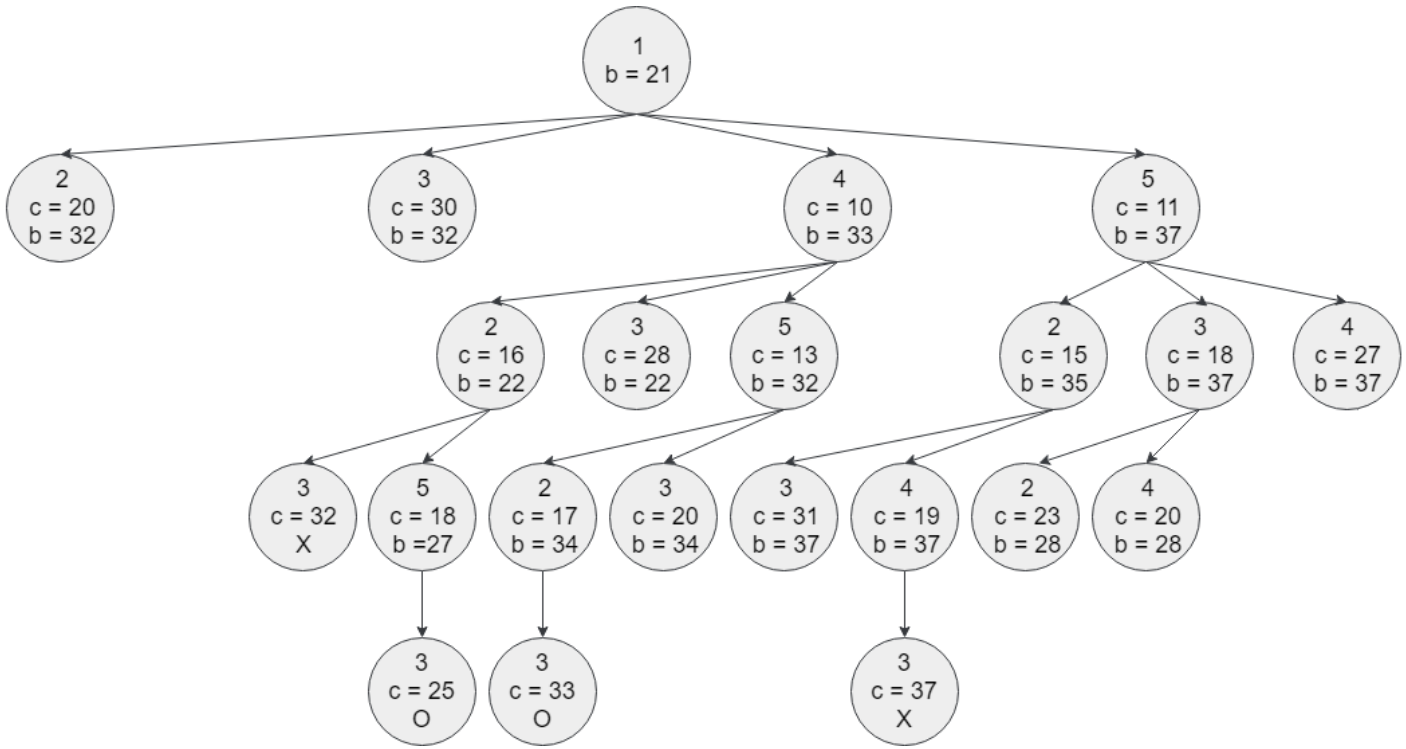
Metoda podziału i ograniczeń



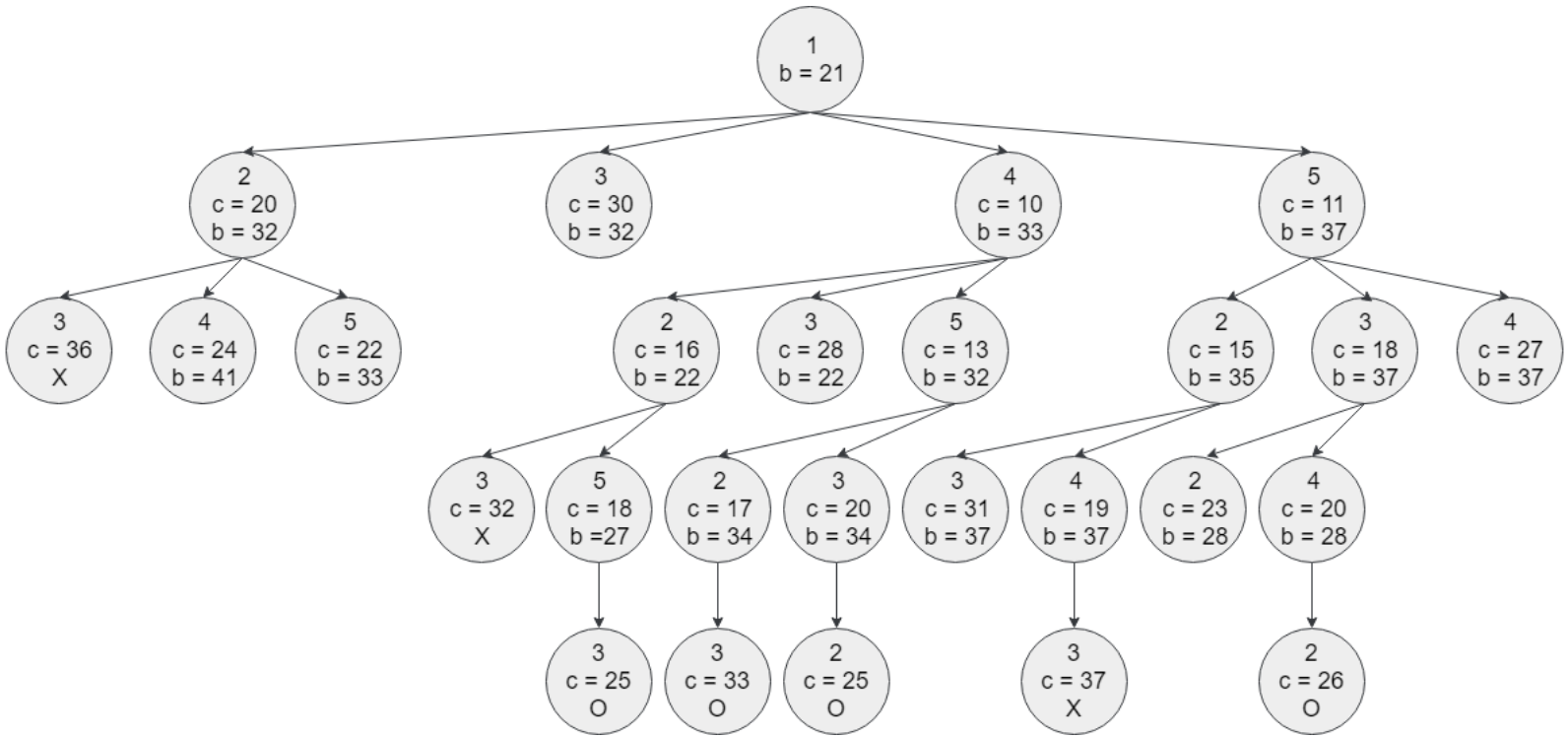
Metoda podziału i ograniczeń



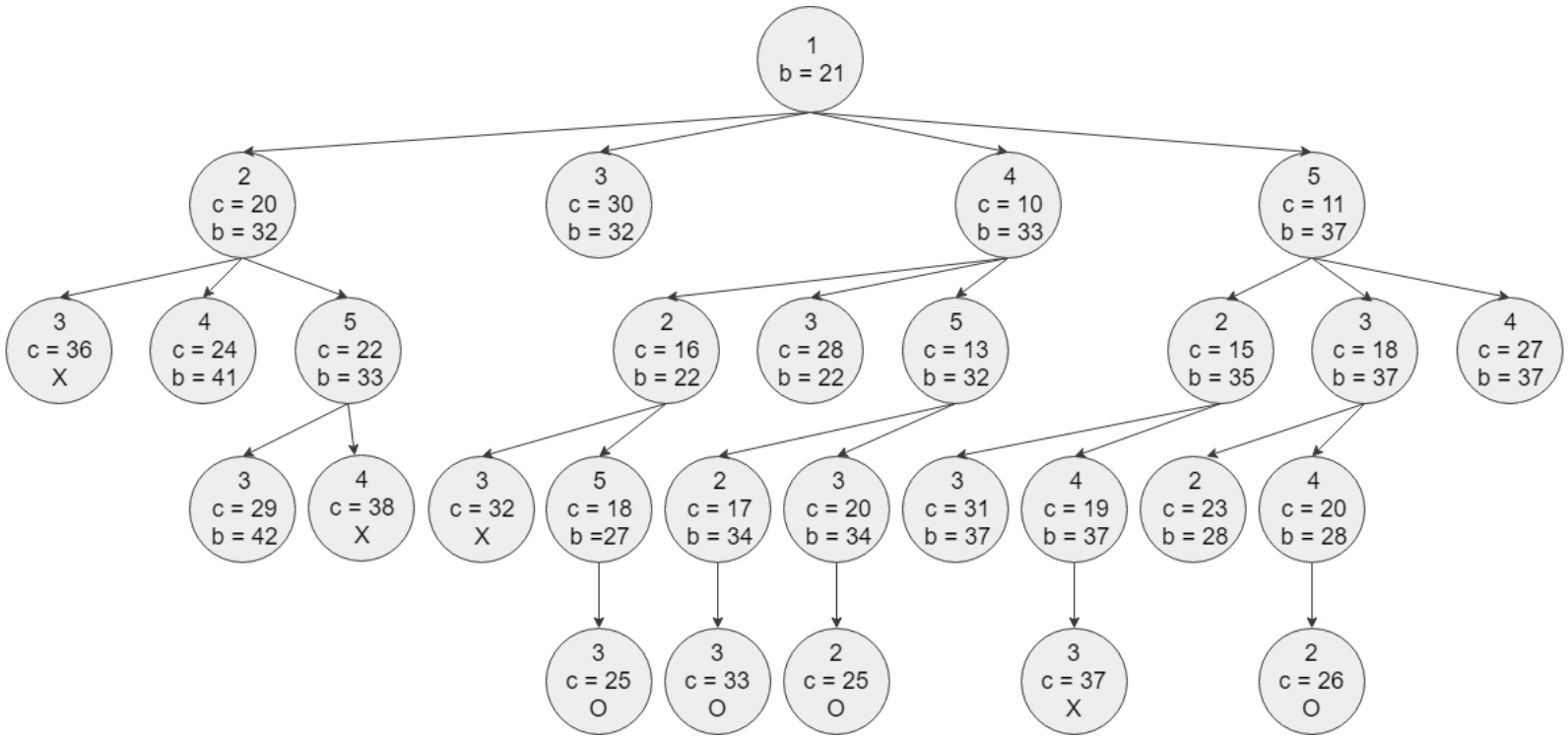
Metoda podziału i ograniczeń



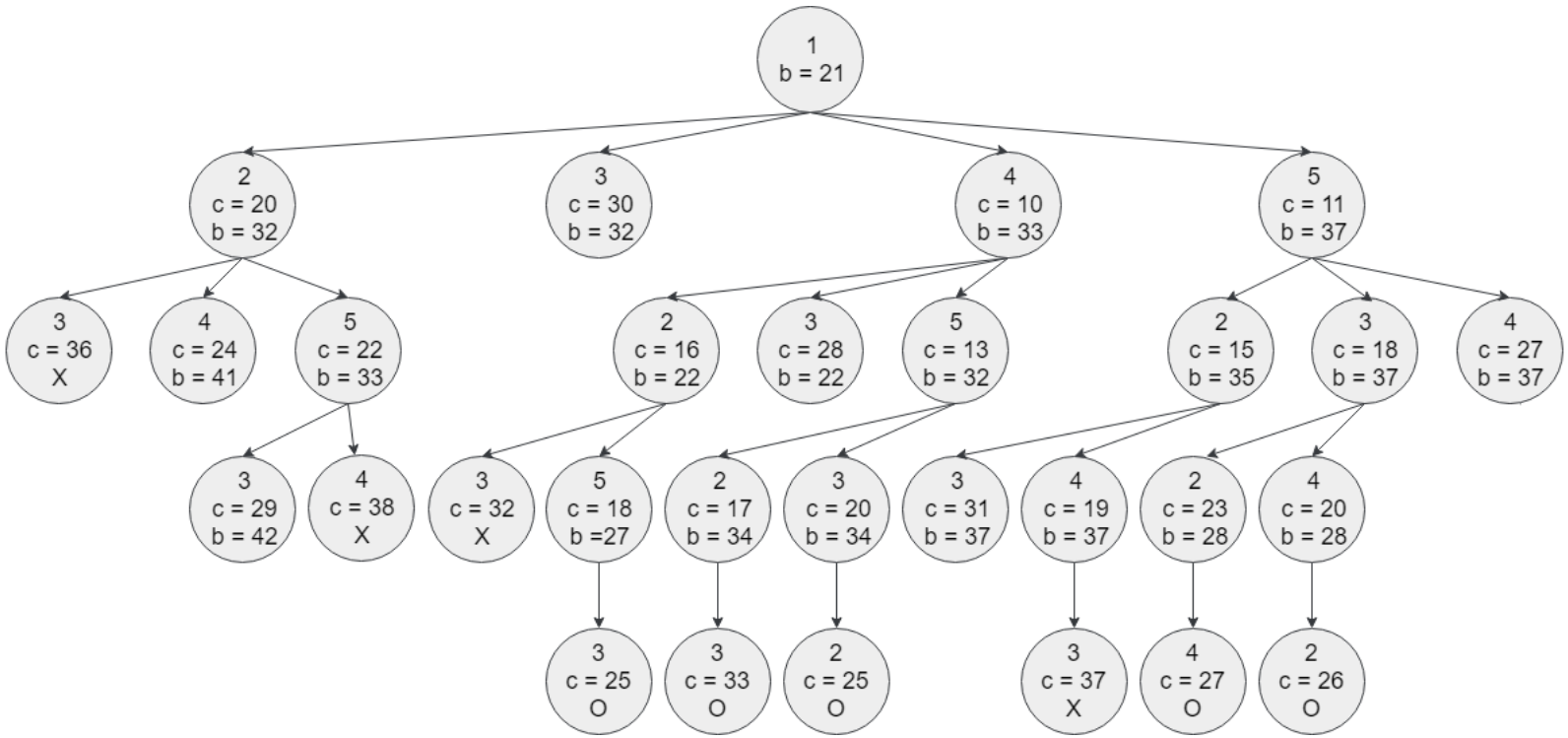
Metoda podziału i ograniczeń



Metoda podziału i ograniczeń



Metoda podziału i ograniczeń



Algorytm przeszukiwania lokalnego porusza się od rozwiązania do rozwiązania sąsiedniego w przestrzeni rozwiązań. Koniec przeszukiwania ustalany jest przez kryterium końca, np. znalezienie optimum, czas itp.

Niech X będzie zbiorem wszystkich rozwiązań. Sąsiedztwo rozwiązania x, $N(x) \subseteq X$, to zbiór rozwiązań leżących „blisko" x, czyli sąsiedztwo rozwiązania x to skończony zbiór rozwiązań podobnych do rozwiązania x.

Cechy sąsiedztwa:

- ograniczenie na rozmiar: dla każdego x, $N(x)$ zawiera co najmniej jedno rozwiązanie różnie od x; zbiór sąsiednich rozwiązań nie może być równy liczebnością przestrzeni wszystkich rozwiązań;
- podobieństwo sąsiadów: kolejne rozwiązanie nie powinno różnić się zbyt wiele tak, aby nie wymagało konstruowania zupełnie nowego rozwiązania;
- równouprawnienie: niezależnie od wyboru rozwiązania początkowego, powinno być możliwe osiągnięcie każdego rozwiązania należącego do pełnej przestrzeni rozwiązań.

Przykład sąsiedztwa dla problemu komiwojażera

k-zamiana (*k-swap*, *k-opt*): $N(x)$ jest zbiorem rozwiązań powstałych przez zamianę k wybranych miast miejscami.

2-zmiana miast 3 i 5:

Gdańsk-Warszawa-Kraków-Wrocław-Poznań-Gdańsk

Gdańsk-Warszawa-Poznań-Wrocław-Kraków-Gdańsk

Przykład sąsiedztwa dla problemu komiwojażera

Wymiana łuków: $N(x)$ jest zbiorem rozwiązań powstałych przez wstawienie k kolejnych miast w odwrotnej kolejności.

Wymiana łuków 2 i 5:

Gdańsk-Warszawa-Kraków-Wrocław-Poznań-Gdańsk

Gdańsk-Poznań-Wrocław-Kraków-Warszawa-Gdańsk

Algorytm zachłanny (*greedy*)

Algorytm wybiera pierwsze przejrzane rozwiązanie sąsiednie, które jest lepsze niż aktualne.

Algorytm zachłanny (*greedy*)

Algorytm wybiera pierwsze przejrzane rozwiązanie sąsiednie, które jest lepsze niż aktualne.

Algorytm stromy (*steepest*)

Algorytm wybiera najlepsze rozwiązanie spośród swoich sąsiadów, które jest lepsze niż aktualne.

Algorytm zachłanny (*greedy*)

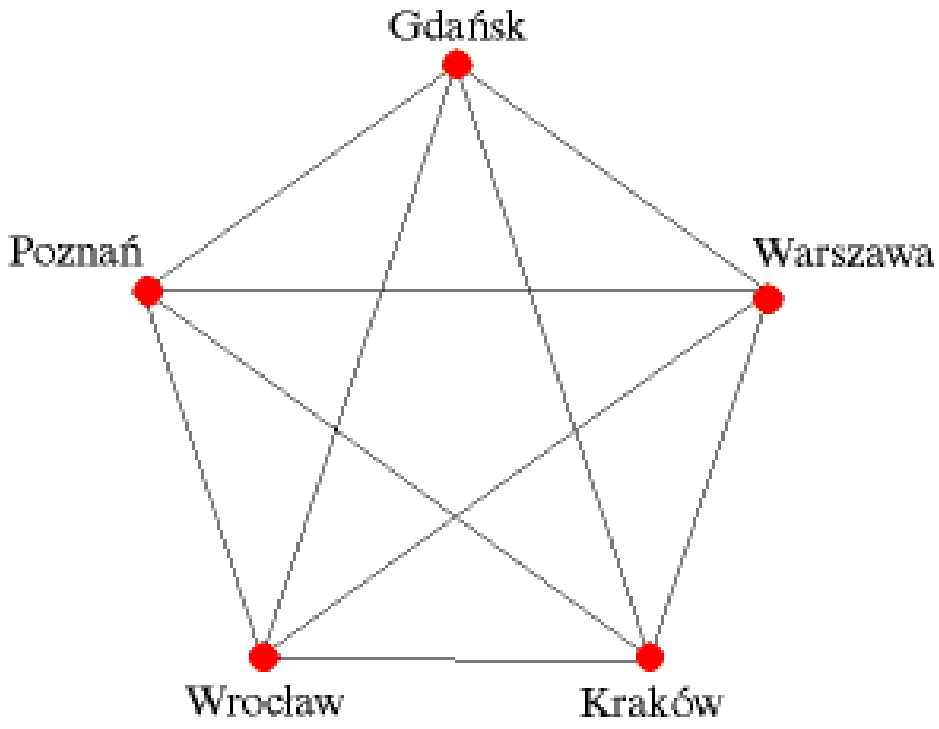
Algorytm wybiera pierwsze przejrzane rozwiązanie sąsiednie, które jest lepsze niż aktualne.

Algorytm stromy (*steepest*)

Algorytm wybiera najlepsze rozwiązanie spośród swoich sąsiadów, które jest lepsze niż aktualne.

Który jest „lepszy”? Co to znaczy „lepszy”? Czy one konstruują rozwiązania?

Przeszukiwanie lokalne: *greedy* i *steepest*
Przykład



	Poznań	Warszawa	Gdańsk	Kraków	Wrocław
Poznań	1	310	333	444	168
Warszawa	310	1	358	294	363
Gdańsk	333	358	1	600	463
Kraków	444	294	600	1	270
Wrocław	168	363	463	270	1

Greedy
Przykład

2490 Poznań, Kraków, Gdańsk, Wrocław, Zielona, Toruń, Warszawa, Poznań

wymiana	
Gdańsk-Kraków	-304

2186 Poznań, Gdańsk, Kraków, Wrocław, Zielona, Toruń, Warszawa, Poznań

wymiana	zysk
Kraków-Gdańsk	304
Wrocław-Gdańsk	155
Wrocław-Zielona	-193

1993 Poznań, Gdańsk, Kraków, Zielona, Wrocław, Toruń, Warszawa, Poznań

wymiana	zysk
Kraków-Zielona	191
Wrocław-Zielona	42
Gdańsk-Zielona	193
Toruń-Zielona	578
Warszawa-Zielona	-42

1951 Poznań, Gdańsk, Kraków, Warszawa, Wrocław, Toruń, Zielona, Poznań

wymiana	zysk
Kraków-Warszawa	227
Wrocław-Warszawa	-247

1704 Poznań, Gdańsk, Kraków, Wrocław, Warszawa, Toruń, Zielona, Poznań

wymiana	zysk
Kraków-Wrocław	345
Warszawa-Wrocław	247
Gdańsk-Wrocław	628
Toruń-Wrocław	308
Zielona-Wrocław	152
Kraków-Warszawa	335
Kraków-Gdańsk	464
Kraków-Toruń	500
Kraków-Zielona	483
Gdańsk-Warszawa	348
Toruń-Warszawa	292
Zielona-Warszawa	331
Toruń-Gdańsk	25
Zielona-Gdańsk	286
Zielna-Toruń	328

2490 Poznań, Kraków, Gdańsk, Wrocław, Zielona, Toruń, Warszawa, Poznań

wymiana	zysk
Kraków-Toruń	-507
Gdańsk-Warszawa	-425
Gdańsk-Zielona	-306
Kraków-Gdańsk	-304
Kraków-Zielona	-179

1983 Poznań, Toruń, Gdańsk, Wrocław, Zielona, Kraków, Warszawa, Poznań

wymiana	zysk
Zielona-Wrocław	-146
Kraków-Gdańsk	7
Warszawa-Wrocław	23
Warszawa-Zielona	36
Warszawa-Toruń	155

1837 Poznań, Toruń, Gdańsk, Zielona, Wrocław, Kraków, Warszawa, Poznań

wymiana	zysk
Kraków-Gdańsk	1
Warszawa-Wrocław	53
Kraków-Wrocław	120
Zielona-Wrocław	146
Warszawa-Zielona	152

Przykład z wykorzystaniem siatki dwuwymiarowej

	A	B	C	D	E	F
1	0	1	8	2	4	1
2	3	3	3	5	1	0
3	6	0	9	8	0	6
4	5	7	8	2	1	2
5	6	6	7	8	0	4
6	2	0	8	6	2	1

Załóżmy jakiś punkt startowy, jak będą wyglądać rozwiązania w zależności od:

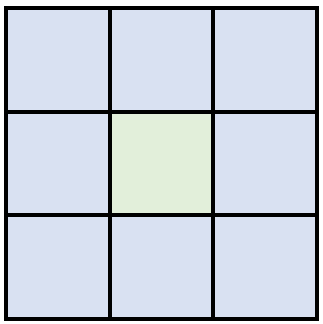
- algorytmu (*stepest*, *greedy*)
- sąsiedztwa (Moore'a, von Neumanna)
- zastosowania $>$ lub $\geqslant$

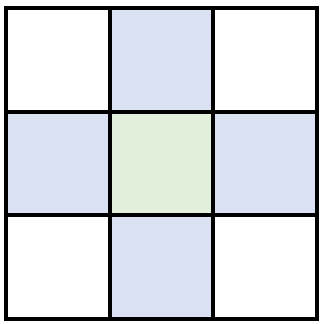
Sąsiedztwo Moore'a

Sąsiadem jest każda komórka, która z centralną graniczy krawędzią lub wierzchołkiem.

Sąsiedztwo von Neumanna

Sąsiadem jest każda komórka, która graniczy z komórką centralną krawędzią.





Wady i zalety

Zalety:

- proste w implementacji
- stosunkowo szybkie w działaniu
- stosowalne dla wielu problemów optymalizacji

Wady:

- wynik ich działania jest zależny od rozwiązania startowego
- bardzo duże ryzyko zatrzymania w optimum lokalnym
- znalezione rozwiązanie jest zwykle rozwiązaniem przybliżonym danego problemu
- ustalone sąsiedztwo

W jaki sposób poradzić sobie z wadami?

Symulowane wyżarzanie

Algorytm ten nazwę zawdzięcza analogii do zjawisk fizycznych.
W procesach stygnięcia metali zaobserwowano, że podczas stopniowego, powolnego ochładzania, cząsteczki ciała oddając energię rozkładają się (umiejscawiają) w sposób bardziej systematyczny i tworząc w ten sposób bardziej równomierne struktury. W przypadku szybkiego spadku temperatury, ich położenie jest bardziej chaotyczne.

Podstawowym wzorem wykorzystywanym w tym algorytmie jest:

$$P(\Delta E) = e^{\frac{-\Delta E}{T}}$$

gdzie:

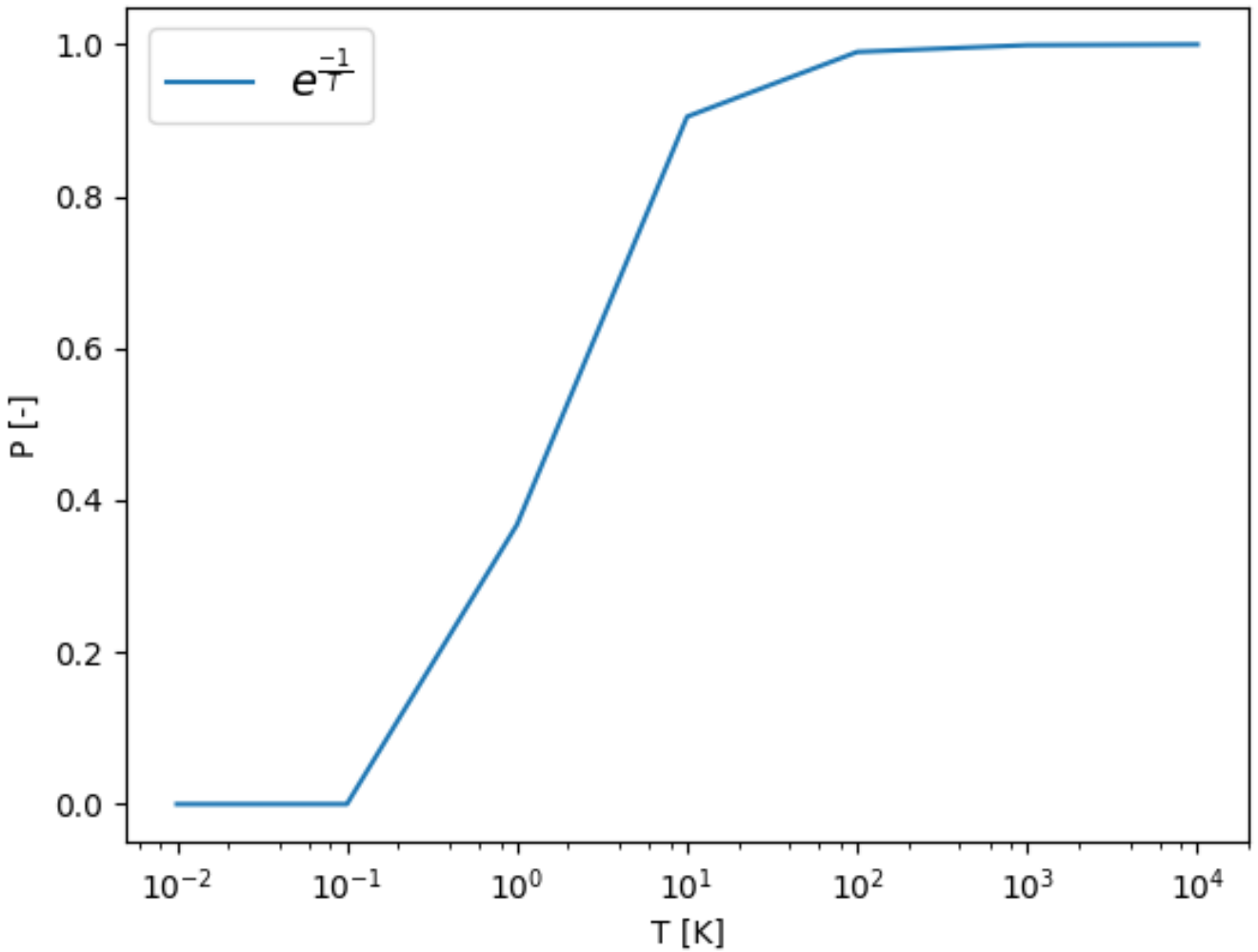
- ΔE — różnica energii $(E_i - E)$
- T — temperatura

Zasada dobierania nowych rozwiązań

s_i, s — aktualne rozwiązanie, najlepsze rozwiązanie

if $E_i > E$ **then**
    $s = s_i$
else
    if $e^{\frac{-(E_i - E)}{T}} > rand(0, 1)$ **then**
        $s = s_i$
    end if
end if

Symulowane wyżarzanie

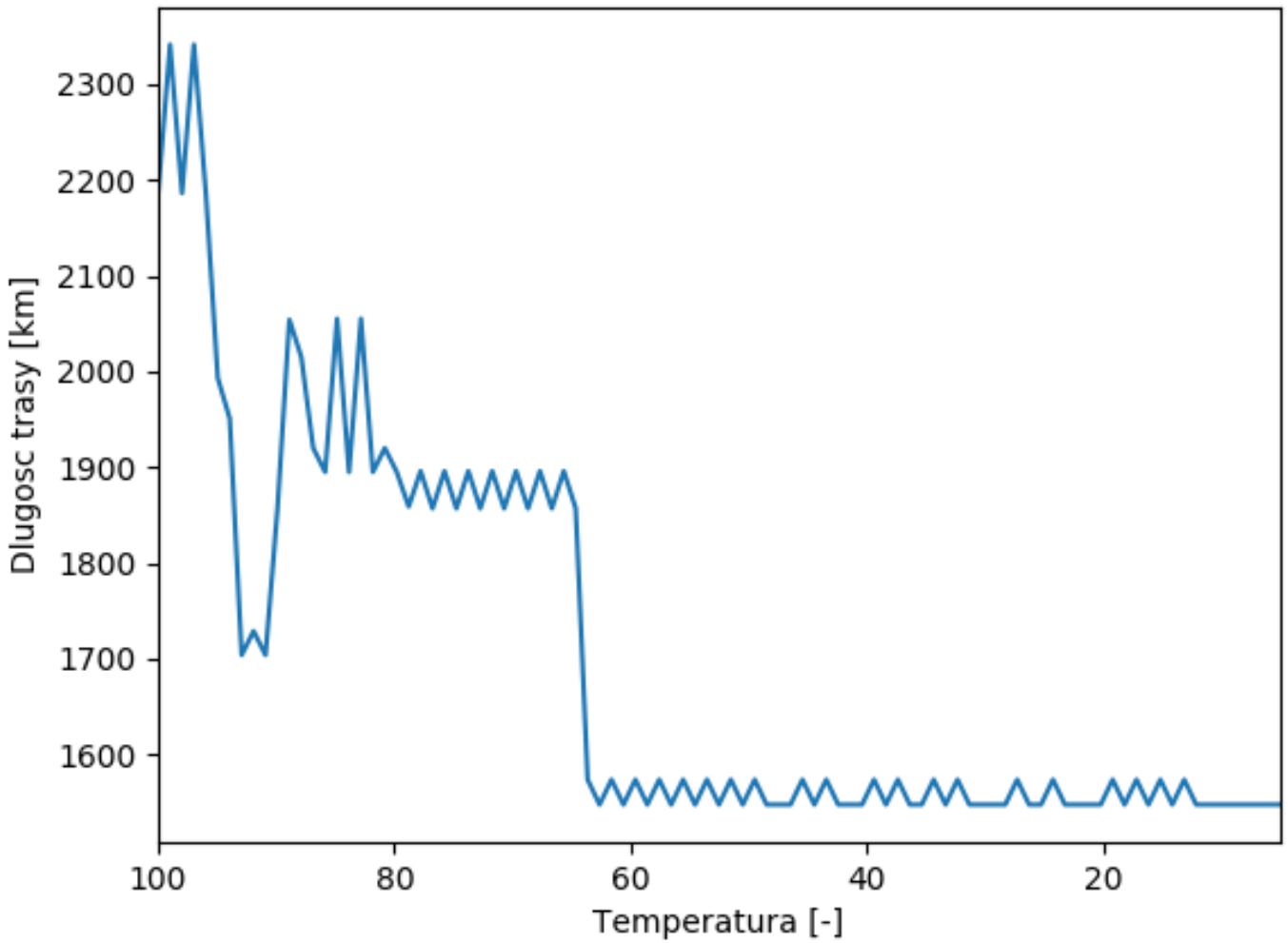


Współczynnik „temperatura” reguluje proces wyboru kolejnych rozwiązań, gdy kandydat na nowe rozwiązani jest gorszy, w sposób następujący:

- Jeżeli wartość współczynnika jest duża, to prawdopodobieństwo wyboru gorszego rozwiązania jest duże.
- Jeżeli wartość współczynnika jest mała, to prawdopodobieństwo wyboru nowego rozwiązania, gorszego od poprzednika, jest bardzo małe.
- Wartość współczynnika „temperatura” w kolejnych iteracjach algorytmu jest zmniejszana, więc w kolejnych krokach szansa przejścia do nowego, gorszego rozwiązania maleje. W końcowych iteracjach rozwiązanie stabilizuje się.

Przykład dla problemu komiwojażera:

- Wybierz początkowe rozwiązanie.
- Wybierz sąsiadujące rozwiązanie:
  - Jeżeli nowe rozwiązanie zmniejsza dystans do przebycia, zaakceptuj je jako nowe rozwiązanie.
  - Jeżeli nowe rozwiązanie zwiększa dystans do przebycia, zaakceptuj je jako nowe rozwiązanie zgodnie z określonym wcześniej prawdopodobieństwem
- Zmniejsz temperaturę.


Dlugosc trasy [km]
2300
2200
2100
2000
1900
1800
1700
1600
100
80
60
40
20
Temperatura [-]

Przeszukiwanie tabu

W tej metodzie przeglądamy przestrzeń rozwiązań, przechodząc zawsze do tego sąsiada, dla którego wartość funkcji oceny jest największa , o ile nie znajduje się on na liście punktów zabronionych (tabu). To różni ją od algorytmu *steepest*. Lista tabu zapamiętuje ruchy oraz częstość ich występowania, w celu unikania minimów lokalnych. Na takiej liście przechowujemy x ostatnio odwiedzonych punktów.

Kryterium aspiracji

Zastosowanie tabu list w pewnych sytuacjach może prowadzić do pomijania niektórych bardzo dobrych (ze względu na wartość funkcji celu) ruchów.

Kryterium aspiracji to pewien warunek, który mówi że jeśli w otoczeniu rozwiązania zostało znalezione bardzo dobre rozwiązanie, ale będące na liście tabu, to właśnie to rozwiązanie jest brane jako kolejne.

2490 Poznań, Kraków, Gdańsk, Wrocław, Zielona, Toruń, Warszawa, Poznań

	Gdańsk	Wrocław	Zielona	Toruń	Warszawa
Kraków				5	
Gdańsk					
Wrocław					
Zielona					
Toruń					

wymiana	zysk
Kraków-Toruń	-507
Gdańsk-Warszawa	-425
Gdańsk-Zielona	-306
Kraków-Gdańsk	-304
Kraków-Zielona	-179

1983 Poznań, Toruń, Gdańsk, Wrocław Zielona, Kraków, Warszawa, Poznań

	Gdańsk	Wrocław	Zielona	Toruń	Warszawa
Kraków				4	
Gdańsk					
Wrocław			5		
Zielona					
Toruń					

wymiana	zysk
Zielona-Wrocław	-146
Kraków-Gdańsk	7
Warszawa-Wrocław	23
Warszawa-Zielona	36
Warszawa-Toruń	155

1837 Poznań, Toruń, Gdańsk, Zielona, Wrocław Kraków, Warszawa, Poznań

	Gdańsk	Wrocław	Zielona	Toruń	Warszawa
Kraków	5			3	
Gdańsk					
Wrocław			4		
Zielona					
Toruń					

wymiana	zysk
Kraków-Gdańsk	1
Warszawa-Wrocław	53
Kraków-Wrocław	120
Zielona-Wrocław	146
Warszawa-Zielona	152

1838 Poznań, Gdańsk, Toruń, Zielona, Wrocław Kraków, Warszawa, Poznań

	Gdańsk	Wrocław	Zielona	Toruń	Warszawa
Kraków	4			2	
Gdańsk					
Wrocław			3		5
Zielona					
Toruń					

wymiana	zysk
Kraków-Gdańsk	-1
Warszawa-Wrocław	78
Kraków-Zielona	113
Zielona-Wrocław	152
Warszawa-Zielona	152

1990 Poznań, Gdańsk, Toruń, Zielona, Warszawa, Kraków, Wrocław Poznań

	Gdańsk	Wrocław	Zielona	Toruń	Warszawa
Kraków	3	5		1	
Gdańsk					
Wrocław			2		4
Zielona					
Toruń					

wymiana	zysk
Kraków-Wrocław	-286
Warszawa-Zielona	-152
Kraków-Gdańsk	-1
Toruń-Wrocław	22
Warszawa-Wrocław	36

Zalety:

- Pozwala na opuszczenie lokalnego minimum.
- Wykorzystuje informacje o poprzednich ruchach.

Wady:

- Jak ocenić ruch?
- Jak długo powinna trwać kadencja ruchu na liście?
- Jak dostosować inne parametry dla symulowanego wyżarzania?

Klasyfikacja algorytmów optymalizacji

- Dokładne
  - przeszukiwanie pełne (wyczerpujące)
  - programowanie matematyczne
  - BB
- Heurystyczne
  - metaheurystyki
    - losowe
    - przeszukiwania lokalnego
    - ewolucyjne, mrówkowe itp.
  - wyspecjalizowane dla danego problemu
- Aproksymacyjne

Dziękujemy za uwagę