

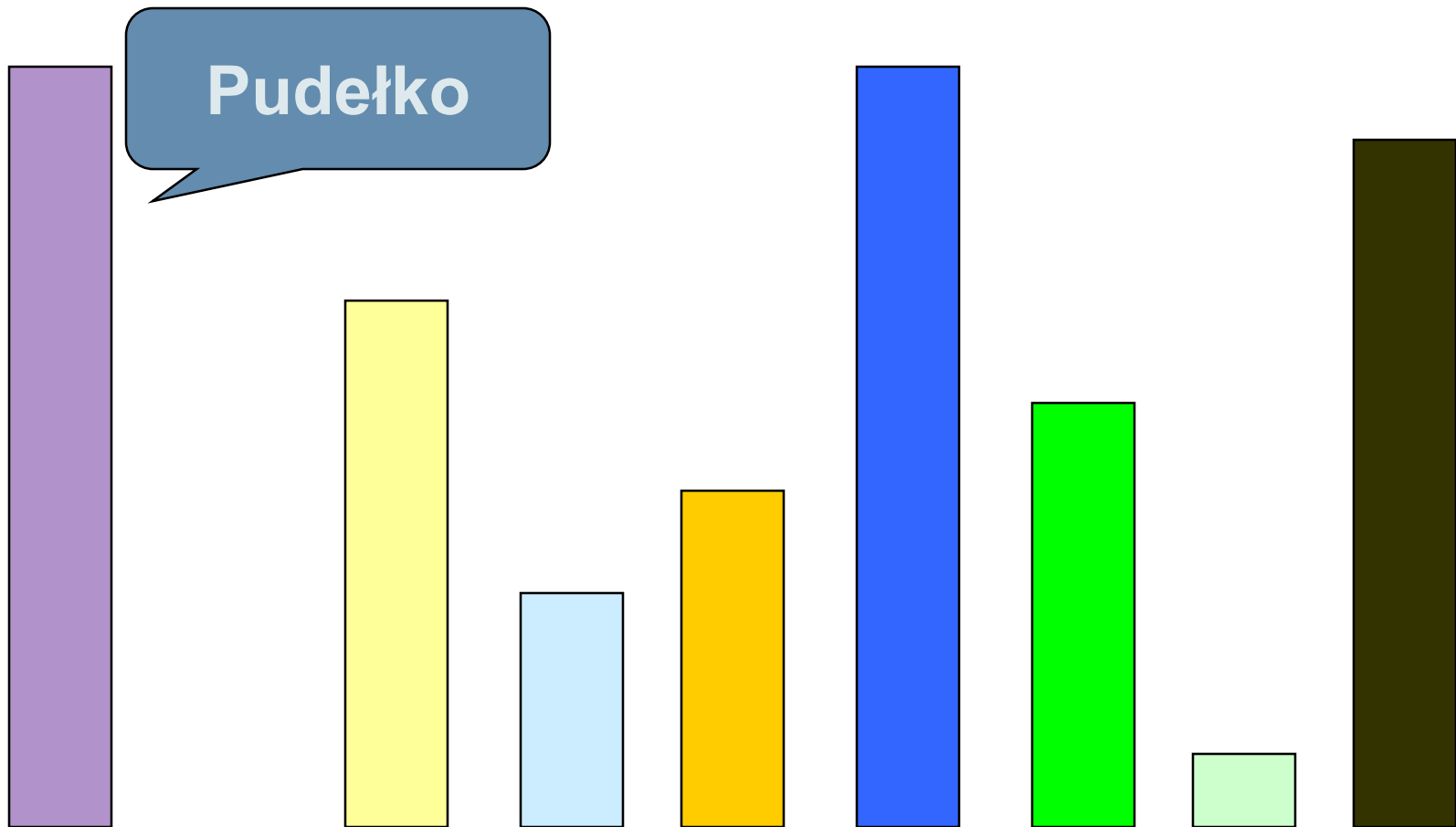
Algorytmy ewolucyjne w grupowaniu

Strategia ewolucyjna

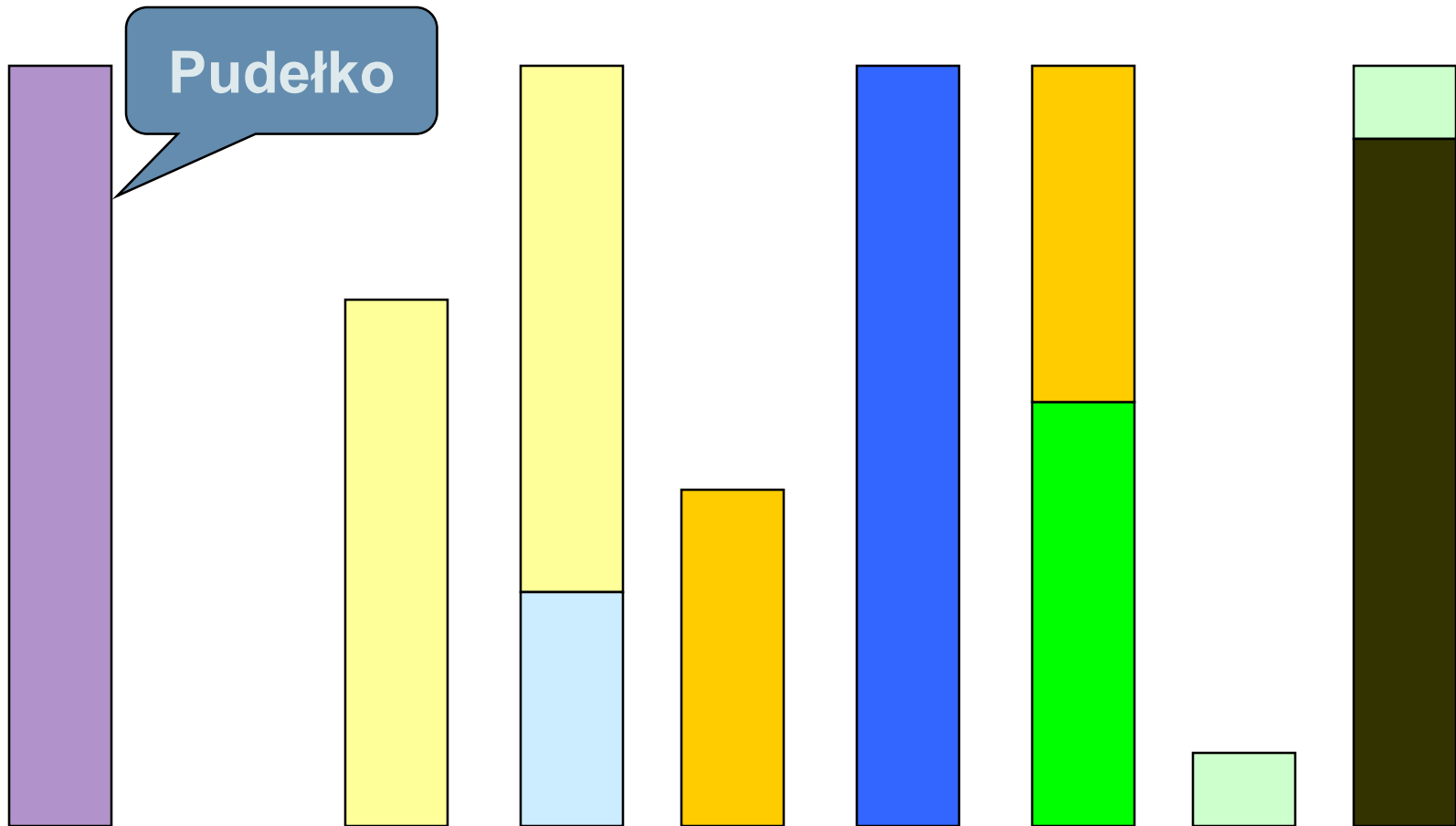
Problem pakowania pudełek (BPP)

Mamy n niepodzielnych obiektów, każdy o wadze (wartości) w_i ($1 \leq i \leq n$) oraz n pudełek (ang. bins) każde o pojemności C ($w_i < C$); należy tak rozmieścić obiekty w pudełkach, by liczba użytych pudełek była minimalna, przy założeniu, że pojemność pudełek nie będzie przekroczona.

Problem pakowania pudełek (BPP)



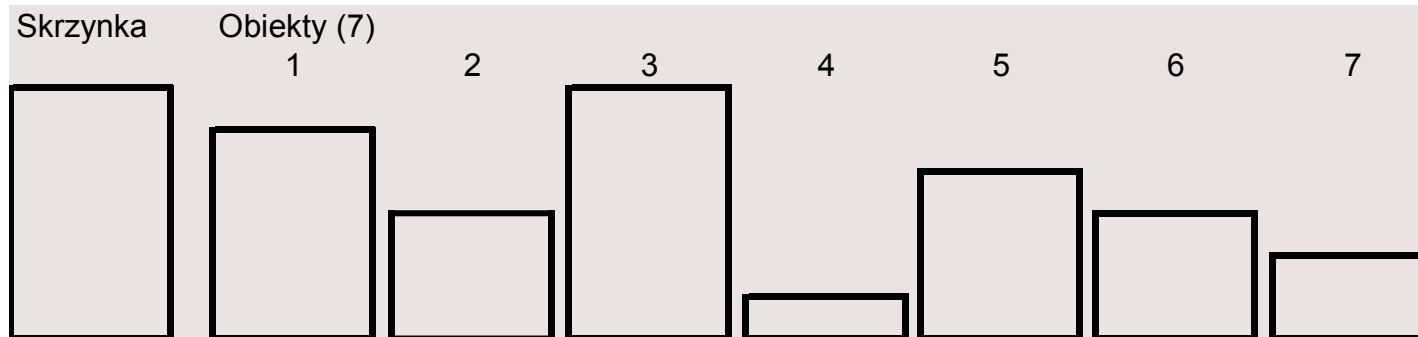
Problem pakowania pudełek (BPP)



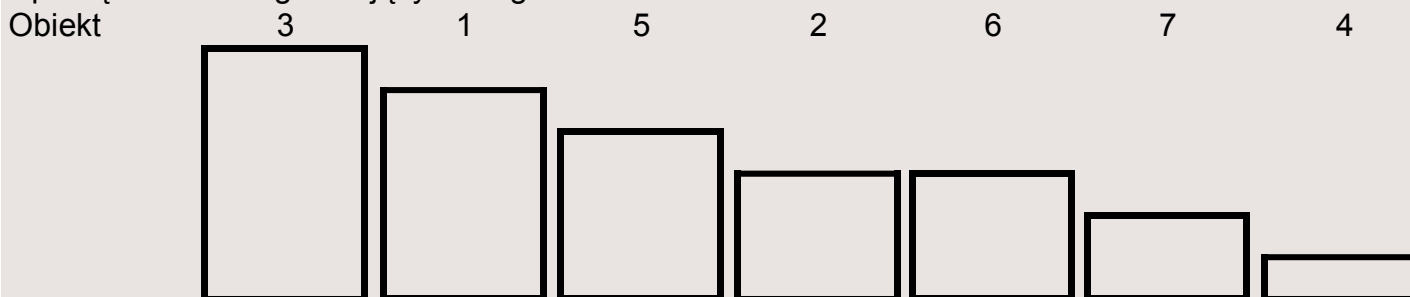
Znane algorytmy

- Proste algorytmy rozwiązujące w sposób przybliżony problem pakowania pudełek podali Garey i Johnson; spośród nich algorytm First Fit Decreasing (FFD), daje w wyniku rozwiązanie nie gorsze niż 22% od rozwiązania optymalnego.
 1. uporządkuj obiekty wg malejącej wartości wag w_i
 2. dla kolejnych skrzynek:
 - włoż do skrzynki pierwszy wolny obiekt,
 - dopóki nie przekroczona pojemność skrzynki dodaj do skrzynki następny wolny obiekt.

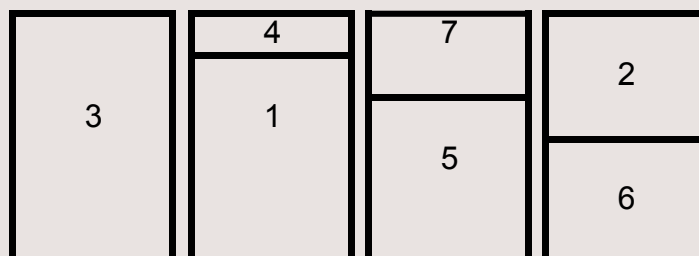
Znane algorytmy - FFD



Uporządkowanie wg malejących wag obiektów



Rozwiązanie problemu BBP algorytmem FFD (potrzebne 4 skrzynki)



Znane algorytmy

- W ostatnim okresie trwają intensywne prace nad zastosowaniem algorytmów ewolucyjnych do różnych problemów optymalizacji kombinatorycznej. W przypadku BPP opracowano specjalizowany algorytm genetyczny Falkenauera GGA bazujący na specjalizowanej reprezentacji problemu oraz specjalnie skonstruowanych operatorach pseudogenetycznych. Algorytm ten uważany jest za najlepszy algorytm przybliżony dla problemu pakowania pudełek.

Strategia ewolucyjna

- ▶ Wersja strategii ewolucyjnej jest wynikiem prac nad algorytmami ewolucyjnymi dla problemu szeregowania zadań; nie próbowano optymalizować żadnych parametrów algorytmu. Jest to $(1, \lambda)$ -ES, w której λ potomków jest generowanych z jednego rodzica za pomocą prostych mutacji. Krzyżowanie nie jest stosowane. Najlepszy z potomków zastępuje rodzica w nowej populacji. Ten sposób selekcji (rodzic nie konkuruje z potomkami) powoduje często pogorszenie rozwiązania wejściowego, ale poprawia efektywność algorytmu.

Strategia ewolucyjna

BEGIN

generate and evaluate starting solution;

best solution:= start solution; *counter*:=0;

REPEAT

FOR *i*:= 1 to *I* DO

BEGIN

copy parent to create child number *i*;

mutate child *i*;

evaluate child *i* according to fitness function *FF*;

END;

choose the best child based on fitness function value to be a new parent;

IF $FF(\text{new parent}) \leq FF(\text{best solution})$ THEN

counter:= *counter*+1

ELSE

counter:=0;

best solution:= new parent

END;

IF $\text{counter} \geq \text{Max}_C$ THEN

BEGIN

counter:=0;

choose the worst child based on objective function value to be a new parent

END;

UNTIL (*max. iterations*) or ($FF(\text{best solution}) = \text{optimum}$);

output the best solution;

END.

ES dla BPP

- **Reprezentacja rozwiązania problemu**

Lista elementów i separatorów, np. dla 7 elementów i 3 separatorów:

$$R_1 = (1, 3, 9, 8, 5, 2, 7, 10, 6, 4)$$

$$(1, 3); (5, 2, 7); (6, 4),$$

rozwiązanie

$$R_2 = (1, 10, 3, 8, 5, 2, 9, 7, 6, 4)$$

$$(1); (3); (5, 2); (7, 6, 4)$$

a rozwiązanie

$$R_3 = (1, 10, 3, 5, 9, 2, 7, 8, 6, 4)$$

$$(1); (3, 5); (2, 7); (6, 4)$$

ES dla BPP

- **Stosowane mutacje**

W algorytmie stosowano znane z literatury mutacje tj. wymiany, wstawiania i inwersji. Najistotniejszą cechą algorytmu, różniącą go od prób innych autorów, jest to, że **mutacjom podlegają zarówno elementy jak i separatory**.

- **Inne parametry**

Rozwiązanie początkowe – losowane.

Populacja – 30 rozwiązań, liczba iteracji – 15 000.

ES dla BPP

Funkcja dopasowania

Funkcja kary:
1 jeśli $F_i \leq C$
-1000 jeśli $F_i > C$

$$FD = \frac{\sum_{i=1}^N s \cdot \left(\frac{F_i}{C} \right)^2}{N}$$

Suma wag
elementów w
pudełku i
 $i=1..N$

Pojemność
pudełka

Liczba grup w
rozwiązaniu

Wyniki eksperymentów

- Do badań wzięto 8 zestawów po 20 przykładowych problemów zawartych w bibliotece Beasley'a. Problemy te podzielone są na dwie grupy:
 - ⇒ elementy o wagach wylosowanych z rozkładu równomiernego z przedziału $[20..100]$ mają być umieszczone w pudełkach o rozmiarach 150; liczba elementów wynosi 120, 250, 500 i 1000,
 - ⇒ elementy o wartościach z przedziału $[0,25..0,50]$ mają być umieszczone w pudełkach o rozmiarach 1; liczba elementów wynosi 60, 120, 249 i 501, z czego $1/3$ ma duże wagi, a $2/3$ - małe (poniżej $1/3$ rozmiaru pudełka).

Wyniki eksperymentów

- Dla każdego zestawu danych wykonywano trzy przebiegi, z których wybierano najlepsze rozwiązanie. W jednym przebiegu algorytm badał 450 000 rozwiązań.
- Ogólnie rzecz biorąc problemy oznaczone *binpack1-binpack3* oraz *binpack5-binpack7* okazały się łatwe dla naszego algorytmu: w każdym przypadku osiągnano rozwiązania optymalne

Wyniki eksperymentów

Zestaw *Binpack4*

No.	OPT	FFD	GGA	MTP	ES-S
1	399	403	399	403	403
2	406	411	406	410	409
3	411	416	411	416	415
4	411	416	411	416	417
5	397	402	397	401	402
6	399	404	399	402	403
7	395	399	395	398	399
8	404	408	404	406	407
9	399	404	399	402	404
10	397	404	397	402	402

Wyniki eksperymentów

Zestaw *Binpack8*

Lp.	OPT	FFD	GGA	MTP	ES-S
1	167	190	167	184	169
2	167	191	167	181	168
3	167	190	167	177	168
4	167	190	167	180	168
5	167	191	167	181	168
6	167	190	167	183	168
7	167	190	167	183	169
8	167	189	167	183	169
9	167	191	167	177	168
10	167	190	167	185	168

Wyniki eksperymentów

- ▶ Zaproponowany algorytm daje tylko nieznacznie gorsze rozwiązania niż - o wiele bardziej skomplikowany - GGA, natomiast dużo lepsze niż inne algorytmy przybliżone. W tej sytuacji trudno się zgodzić ze spostrzeżeniami Falkenauera oraz Khuriego i zespołu, że standardowa reprezentacja porządkowa oraz operatory genetyczne są nieodpowiednie dla problemów grupowania. Okazuje się, iż algorytmy ewolucyjne są na tyle elastyczną techniką, że wystarczy nieznaczna modyfikacja standardowego schematu (tu: separatory grup podlegają mutacji na równi z elementami), by osiągnąć zadowalające rezultaty dla różnorodnych problemów optymalizacji kombinatorycznej. Wyniki eksperymentów jeszcze raz potwierdziły siłę algorytmów ewolucyjnych, która polega na zdolności do generowania bardzo dobrych rozwiązań bez wnikania w strukturę problemu.