Techniki optymalizacji

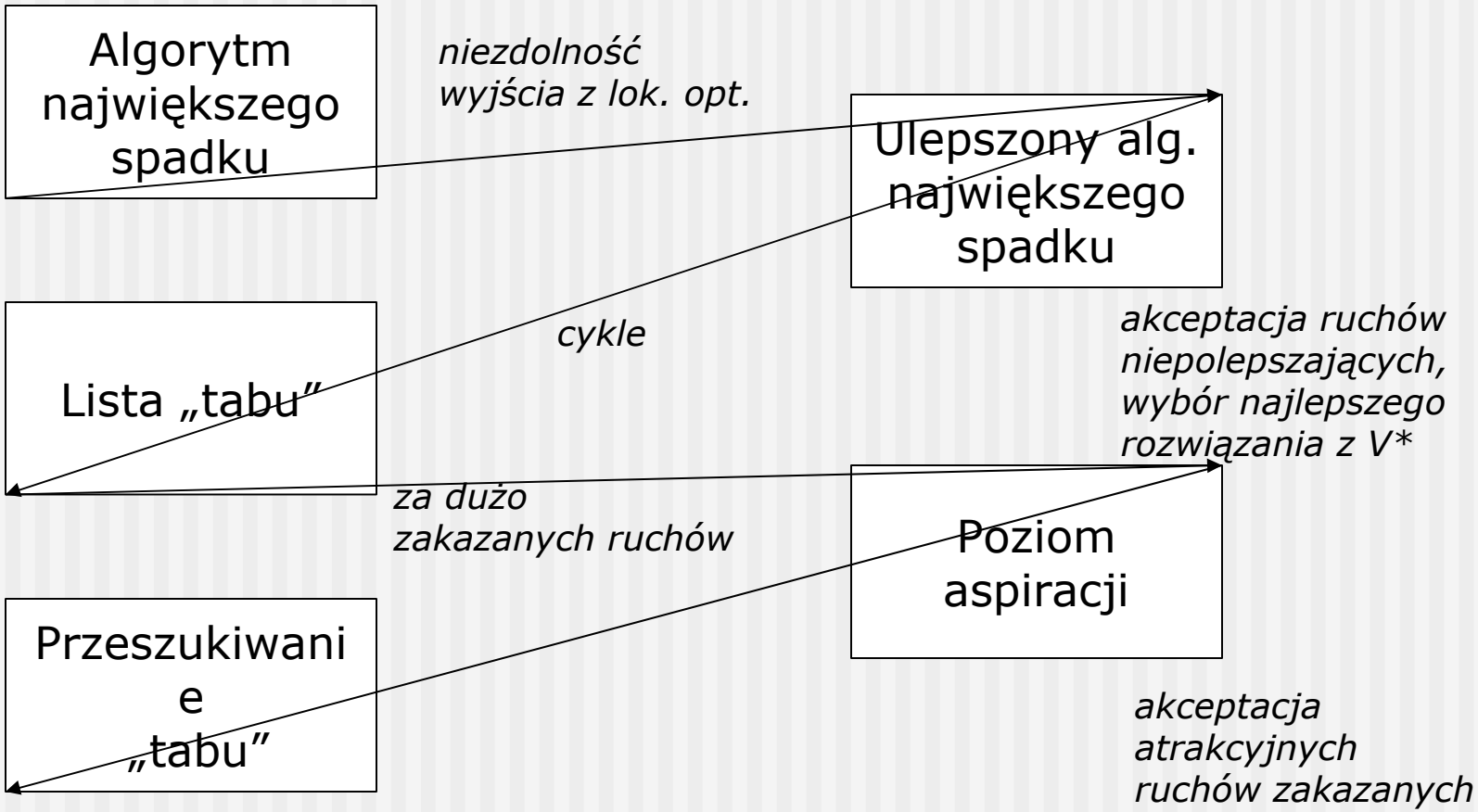
Przeszukiwanie „tabu”

Maciej Hapke
maciej.hapke at put.poznan.pl

Obserwacje

- Lokalne optima nie muszą być dobrymi rozwiązaniami
- Lokalne optima mogą się grupować, leżeć blisko siebie

Naturalny sposób powstania algorytmu

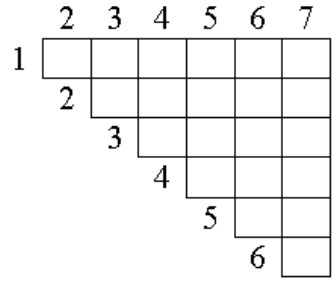


Przeszukiwanie „tabu”

- Główna idea – wykorzystanie pamięci
- Zapamiętywanie rozwiązań czy ruchów

Przykład zapamiętywania ruchów „tabu”

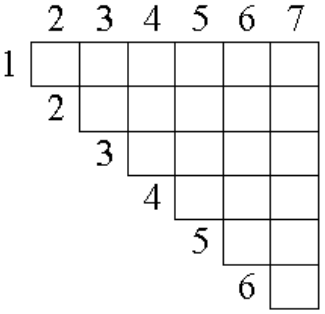
Struktura listy tabu



Wewnątrz numer kadencji (ilość iteracji tabu do końca).

Iteracja 0 (punkt startowy, kryterium maksymalizacji)
Rozwiązanie bieżące (wartość = 10)

2	5	7	3	4	6	1

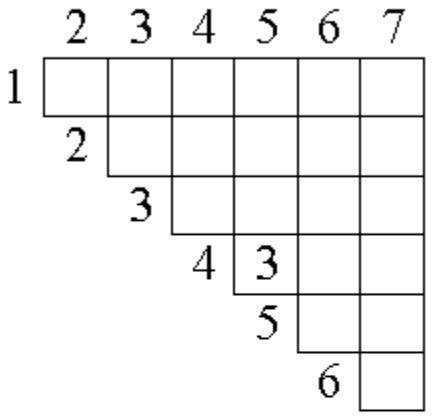


ruch		
5	**4**	**6**
7	4	4
3	6	2
2	3	0
4	1	-1

Iteracja 1

Rozwiązanie bieżące (wartość = 16)

2	4	7	3	5	6	1

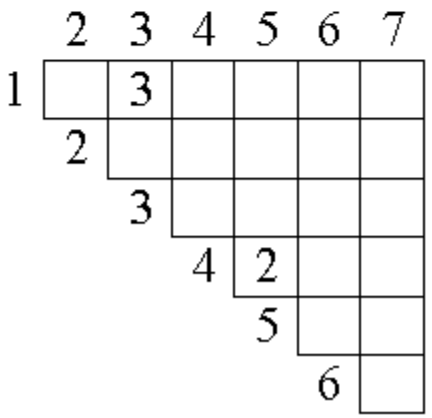


ruch		
3	**1**	**2**
2	3	1
3	6	-1
7	1	-2
6	1	-4

Iteracja 2

Rozwiązanie bieżące (wartość = 18)

2	4	7	1	5	6	3



ruch		
1	3	-2
2	**4**	**-4**
7	6	-6
4	5	-7
5	3	-9

Iteracja 3

Rozwiązanie bieżące (wartość = 14)

4	2	7	1	5	6	3

	2	3	4	5	6	7		ruch		
1		2					T	**4**	**5**	**6**
2		3						5	3	2
3								7	1	0
4			1					1	3	-3
5								2	6	-6
6										

Iteracja 4

Rozwiązanie bieżące

wartość = 20

5	2	7	1	4	6	3

	2	3	4	5	6	7	ruch		
1		1					7	1	0
2		2					4	3	-3
3							6	3	-5
4			3				5	4	-6
5							2	6	-8
6									

Recency-based memory vs. frequency-based memory

Częstość wystąpień poszczególnych ruchów wykorzystywana do rozrzucenia obliczeń po całej przestrzeni rozwiązań dopuszczalnych (dywersyfikacja). Ruchy często powtarzane uzyskują karę w przypadku gdy nie polepszają wartości rozwiązania. Zwykle częstość ta jest w jakiś sposób normalizowana (dzielenie przez wykonaną lub maksymalną liczbę iteracji).
Dywersyfikacja jest przydatna tylko w określonych warunkach (np. nie ma żadnych polepszeń).

Iteracja 26
Rozwiązanie bieżące
wartość = 20

5	2	7	1	4	6	3

	1	2	3	4	5	6	7	
1				3				T
2								
3	3							
4	1	5						
5		4		4				
6			1					
7	2			3				

ruch	v	p
1 4	3	3
2 4	-1	-6
3 7	**-3**	**-3**
1 6	-5	-5
6 5	-4	-6

Algorytm przeszukiwania „tabu”

```
procedure PRZESZUKIWANIE_TABU;
begin
  INICJALIZUJ(xstart,xbest,T);
  x:=xstart;
  repeat
    GENERUJ(V∈Sx);
    WYBIERZ(x');      {min f w V + asp}
    UAKTUALNIJ_LISTĘ_TABU(T);
    if f(x')≤f(xbest) then
      xbest:=x';
    x:=x';
    end;
  until WARUNEK_STOPU;
end;
```

Obserwacje

- Deterministyczny
  - Zły wybór strategiczny jest lepszy niż dobry wybór losowy (daje jakieś informacje, wiedza wzrasta)
- Zbiór V (lista kandydatów)