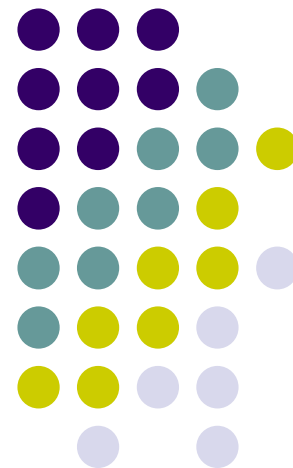


# Projektowanie systemów wbudowanych: kosynteza metodą programowania genetycznego oraz przydział nieprzewidzianych zadań

mgr inż. Adam Górski





# System wbudowany

## Struktura systemu wbudowanego

- Jednostki obliczeniowe (ang. Processing Elements):
  - uniwersalne (ang. Programmable Processors)
  - specjalizowane (ang. Hardware Cores)
- Kanały komunikacyjne

# Zastosowanie systemów wbudowanych



- Technologia kosmiczna
- Nowoczesne samochody
- Telefony komórkowe
- Sprzęt RTV/AGD
- Rakiety balistyczne

# Projektowanie systemów wbudowanych (wg. De Micheli, Gupta)



- Tworzenie modelu
- Implementacja
- Walidacja

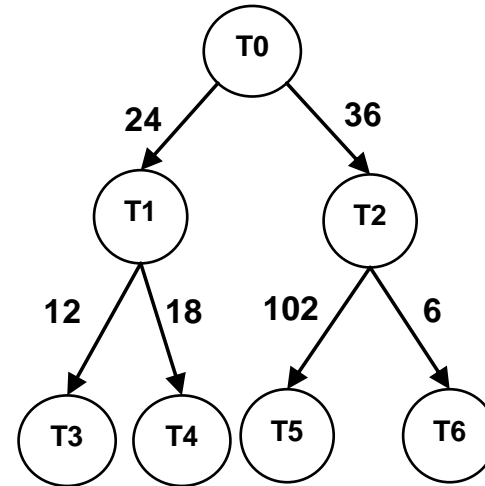
# Specyfikacja systemu wbudowanego w postaci grafu zadań



$v_i \in V$

$e_i \in E$

$$t_{ij} = \frac{e_{ij}}{b_{ij}}$$





## Przykładowa baza zasobów

Zadanie	PP1 C=200		PP2 C=250		HC1	
	t	c	t	c	t	c
T0	100	12	103	10	30	100
T1	87	23	60	34	5	93
T2	45	18	50	20	3	60
T3	90	9	86	8	10	95
T4	35	5	40	8	2	40
T5	22	1	30	3	2	55
T6	190	25	150	30	25	150
CL1, b=6	c=3		c=2		c=10	

# Kosynteza



Optymalizowane parametry:

- koszt (powierzchnia)
- czas
- pobór mocy

# Algorytmy kosyntezy systemów wbudowanych



- Konstrukcyjne
- Rafinacyjne



# Koszt systemu



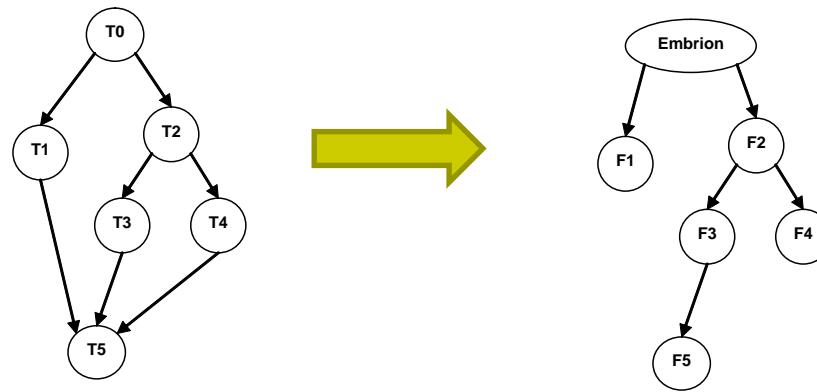
$$C_o = \sum_{i=1}^m C_{PE_i} + \sum_{j=1}^n C_j + \sum_{k=1}^p \sum_{l=1}^{P_k} C_{CL_k, PC_l}$$

# Programowanie genetyczne

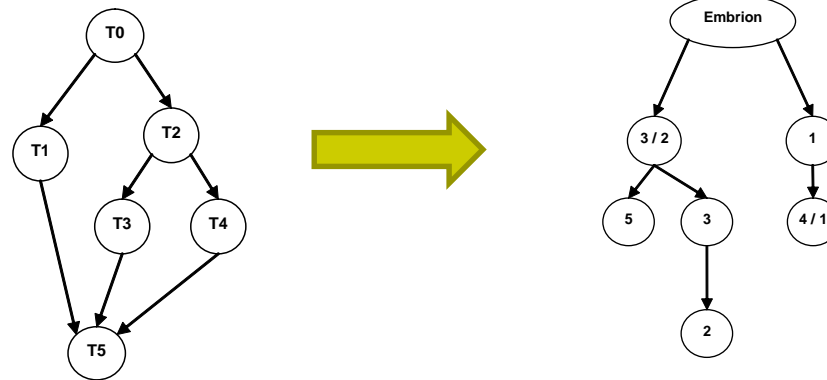


- Rozróżnienie między genotypem (drzewo) oraz fenotypem (gotowe rozwiązanie)
- Ewolucji podlega sposób otrzymania rozwiązania

# Sposób otrzymania genotypu (algorytm konstrukcyjny)



# Sposób otrzymania genotypu (algorytm rafinacyjny)



# Węzeł



- Alokacja zasobu (krok opcjonalny)
- Przyporządkowanie zasobu dla zadania (krok obowiązkowy)
- Alokacja kanału komunikacyjnego (krok opcjonalny)
- Przyporządkowanie kanału komunikacyjnego dla zasobów (krok obowiązkowy)
- Szeregowanie zadań (listowe) (krok obowiązkowy jeśli do jednego zasobu jest przyporządkowane więcej niż jedno zadanie)

# Funkcje konstruuujące system (zasoby obliczeniowe)



1. Używany zasób (0,6)
  - najtańszy (0,2)
  - najszybszy (0,2)
  - $\min(k \cdot t)$  (0,2)
  - najdłużej bezczynny (0,2)
  - taki sam jak dla poprzednika (0,2)
2. Najtańszy zasób (0,1)
3. Najszybszy zasób (0,1)
4. Najmniejszy iloczyn czasu i kosztu (0,1)
5. Najrzadziej używany (0,1)

# Funkcje konstruuujące system (kanały komunikacyjne)



1. Używany kanał komunikacyjny (0,5)
  - najtańszy (0,3)
  - najszybszy (0,3)
  - $\min(k \cdot t)$  (0,2)
  - najdłużej bezczynny (0,2)
2. Najtańszy kanał komunikacyjny (0,2)
3. Największa przepustowość (0,2)
4. Najrzadziej używany (0,1)

# Funkcje rafinujące system (zasoby)



1. Największy zysk kosztu (0,1)
2. Najmniejszy wzrost czasu obliczeń (0,1)
3. Zadanie z największym iloczynem czasu i kosztu przesunąć do zasobu, na którym ten iloczyn jest najmniejszy (0,4)
4. Pierwsze zadanie z listy najczęściej używanego PE na najrzadziej używany zasób (0,2)
5. Najdroższa implementacja zadania z najdroższej ścieżki na najtańszą implementację tego zadania (0,2)



# Funkcje rafinujące system (kanały komunikacyjne)



1. Największy zysk kosztu (0,4)
2. Największa przepustowość (0,3)
3. Najrzadziej używany (0,3)

# Wielkość pokolenia



$$\Pi = \alpha^* n^* e$$

$$P = \frac{\Pi - r}{\Pi}$$

# Ewolucja (algorytm konstrukcyjny)



- Klonowanie

$$\Phi = \beta * \Pi$$

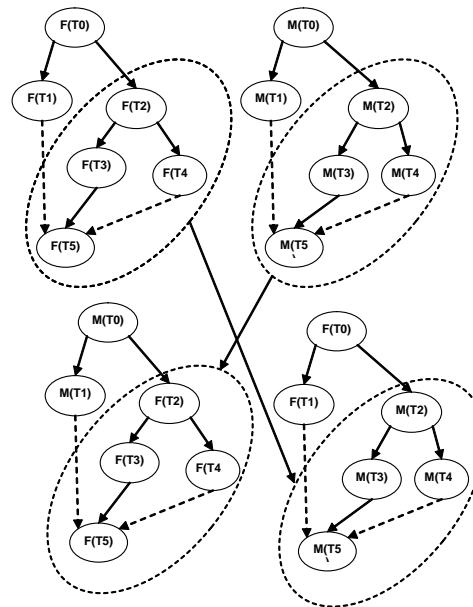
- Krzyżowanie

$$\Psi = \gamma * \Pi$$

- Mutacja

$$\Omega = \delta * \Pi$$

$$\beta + \gamma + \delta = 1$$



# Ewolucja (algorytm rafinacyjny)



- Klonowanie

$$\Phi = \beta * \Pi$$

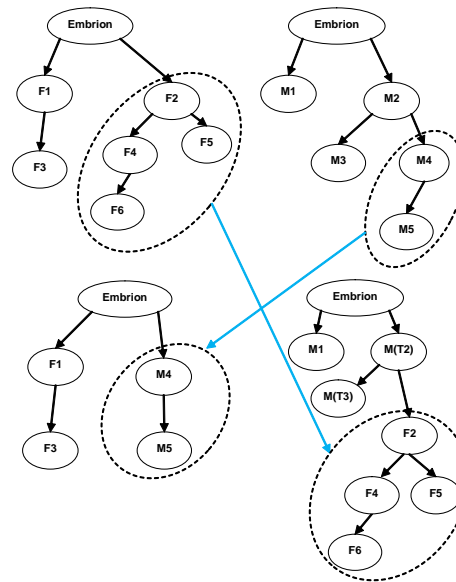
- Krzyżowanie

$$\Psi = \gamma * \Pi$$

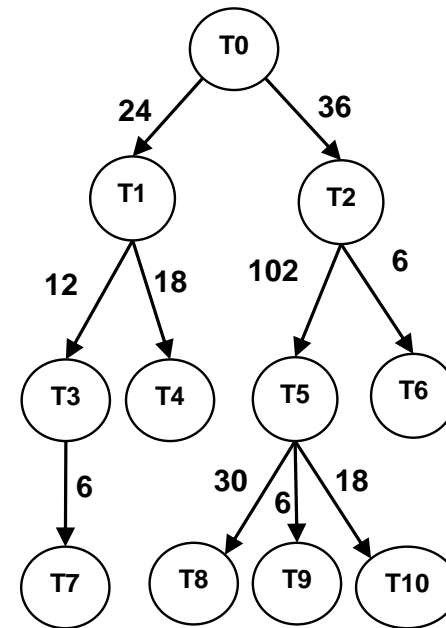
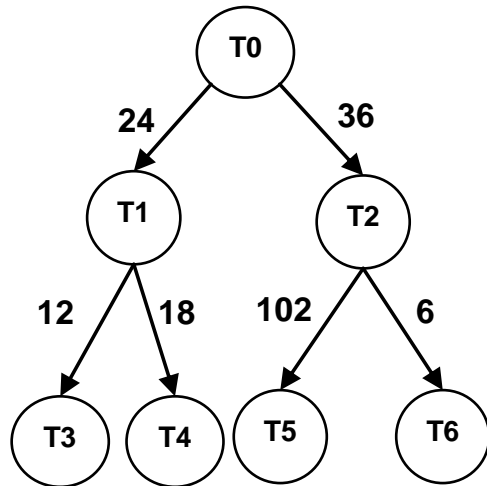
- Mutacja

$$\Omega = \delta * \Pi$$

$$\beta + \gamma + \delta = 1$$



# Zadania dodatkowe (przypadek szczególny)

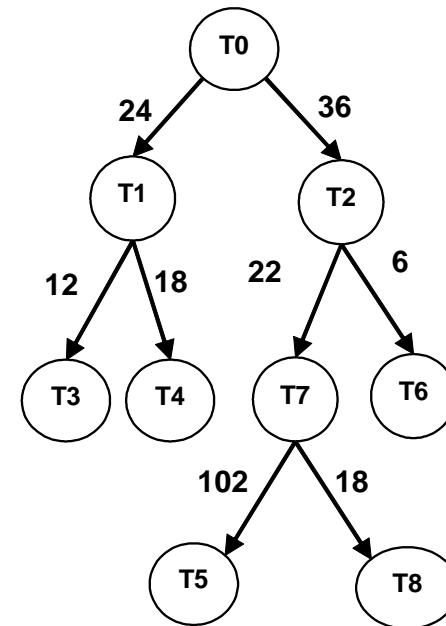
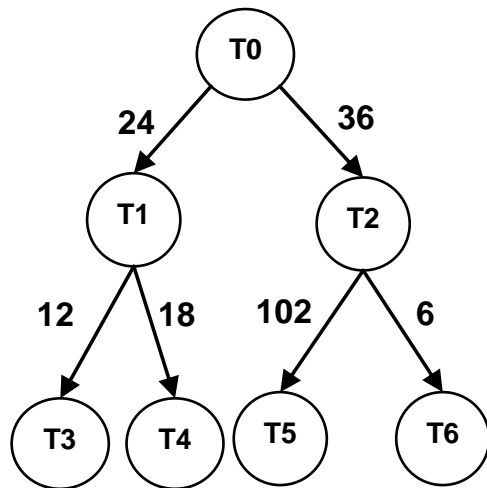




## Przykładowa baza danych (zadania dodatkowe)

Zadanie	PP1		PP2	
	t	c	t	c
T7	80	15	90	10
T8	40	8	55	4
T9	230	22	200	19
T10	130	25	150	18

# Zadania dodatkowe (przypadek ogólny)



# Algorytm

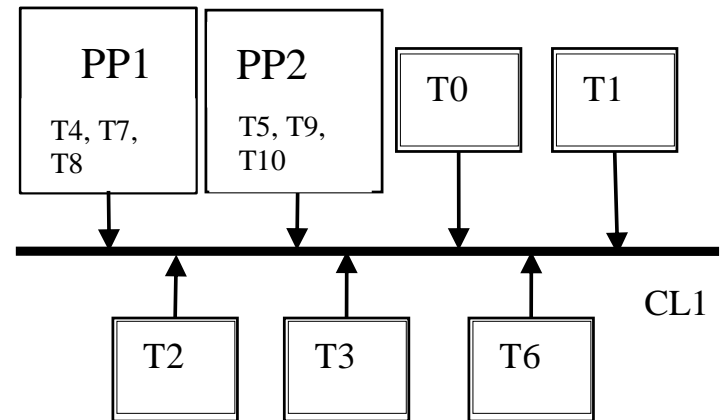
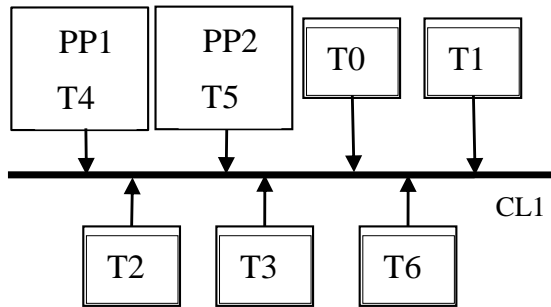


$$V_i = t * c$$

$$C_o = \sum_{i=1}^m C_{PE_i} + \sum_{j=1}^n C_j + \sum_{k=1}^p \sum_{l=1}^{P_k} C_{CL_k, PC_l} + \sum_{s=1}^z C_s$$



# Przykład





**Dziękuję za uwagę**