

24.05.2010.

RP. 3. ciklus

$$t_c N^2 Z \sim E(t_c N^2 Z + t_s 2P + t_w 4NZP)$$

$$N = \delta(P) = ? \quad \text{npr. } N = P$$

$$t_c P^2 Z \sim E(t_c P^2 Z + t_s 2P + t_w 4P^2 Z) / : P^2$$

$$t_c Z \sim E(t_c Z + t_s \frac{2}{P} + t_w 4Z) \\ \underbrace{\sim}_{\approx \phi}$$

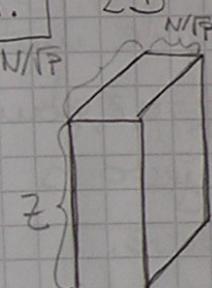
IZUČINKOVITOST?

$$\text{KOLIČINA POSLA: } E_c \cdot (N^2 \cdot Z)$$

$$\Rightarrow O(\phi^2)$$

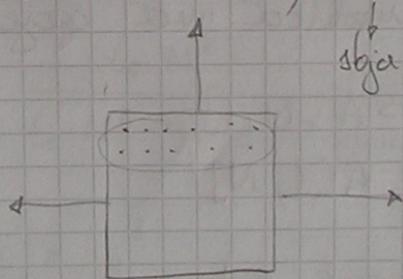
Pr. 1.

2D ATMOSFERSKI MODEL



$$\text{- RAČUNANJE: } T_p^i = L_p = t_c \frac{N^2}{P} \cdot Z$$

$$\text{- KOMUNIKACIJA: 4 PORUKE; } 2 \cdot Z \cdot \frac{N}{P}$$



$$T_p = t_c \frac{N^2}{P} \cdot Z + 4(t_s + t_w \cdot 2 \cdot Z \cdot \frac{N}{P})$$

$$t_c N^2 Z \sim E(t_c N^2 Z + 4t_s P + 8t_w N \cdot Z \cdot \sqrt{P})$$

$$\text{npr. } N = \sqrt{P}$$

$$t_c P Z \sim E(t_c P Z + 4t_s P + 8t_w P Z) / : P$$

$$t_c Z \sim E(t_c Z + 4t_s + 8t_w Z)$$

POSTOJANJE N^3

\Rightarrow IZOUČ. $O(9)$

[Pr. 2.] (KVAADRATNIJE MATRICE)



$$t_c N^3 \sim E(t_c N^3 + 2t_s P^{3/2} + 2t_w N^2 P^{1/2})$$

$$\text{Npr. } N = \sqrt{P}$$

$$t_c P^{3/2} \sim E(t_c P^{3/2} + 2t_s P^{3/2} + 2t_w P^{3/2}) : / P^{3/2}$$

$2(\sqrt{P}-1)(..)$

ZANEFERIMO

$$\text{POSAO} \sim N^3$$

$$\Rightarrow \text{IZOUČIN KOVITOST} = O(P^{3/2}) //$$

[Pr. 3.] Parallelni algoritam iterativno računa elemente matrice.

Izbači vrijednost svakog elementa računa se uz pomoć vrijednosti neposrednih susjeda gore i lijevo s tim da matrica ima stepene ove bridae.

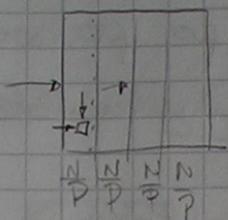
$$A[1,1] \rightarrow A[N,1]$$

$$A[1,N]$$

Trošak računanje jednog elementa kost će biti, krozite trajanje izvođenja na P procesora, te učinkovitost, izračunavljivost ako je matrica na procesore podijeljena po:

- a) stupnju,
- b) paralelizirajuće jedinice.

a)



$$t_R = t_c \cdot \frac{N^2}{P}$$

$$t_K = (t_s + t_w \cdot N) \cdot 1$$

$$E = \frac{t_c N^2}{t_c N^2 + t_s P + t_w NP}$$

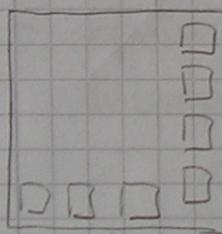
$$t_c N^2 \in E \left(t_c N^2 + t_s P + t_w NP \right) / \cdot P^2$$

$$N = P$$

$$t_c \sim E(t_c + \frac{t_s}{P} + t_w)$$

$$\Rightarrow \text{IZOUČENKOVÝ TOST} = O(P^2)$$

5) $t_R = \left(\frac{N}{P}\right)^2 \cdot t_c = \frac{N^2}{P} \cdot t_c$



$$t_K = 2(t_s + t_w \cdot \frac{N}{P})$$

$$E = \frac{t_c N^2}{t_c N^2 + 2t_s P + 2t_w NP}$$

$$N = \sqrt{P}$$

$$t_c P \in (t_c P + 2t_s P + 2t_w P) / \cdot P$$

$$t_c \sim E(t_c + 2t_s + 2t_w)$$

$$\text{IZOUČ} = O(P)$$

Pr. 4. Parallelno računalo plaća se 1€ po h po posao
Na raspodjeljiv je parallelni program sa trajanjem izvršenja

$T_p = 50 + \frac{150}{P}$ [h]. Dobit od rezultata ovisi o
brojučku dobivanja (rezultata i dobivena je razazom)

$$D = \max(0, 18(T_1 - T_p))$$
 [E]

Koje trajanje izvršenja nam donosi najveću moguću
zadužbu (dobit - troškovi) i na koliko procesora?

(Zadužba) $Z = D - C$ (dobit - troškovi)

$$Z = 18T_1 - 18T_p - P \cdot T_p$$

$$Z = 18T_1 - 18 \cdot 50 - \frac{18 \cdot 150}{P} - 50P - 150$$

$$Z = 18T_1 - 750 - \frac{2700}{P} - 50P$$

$$\frac{dZ}{dP} = \frac{2700}{P^2} - 50 = 0$$

$$P^2 = \frac{2700}{50} = 54$$

$$P = 7.348 \Rightarrow P = 7$$

↳ zbog zaduživanja

$$T_p = 71.43$$

$$C \approx 500$$

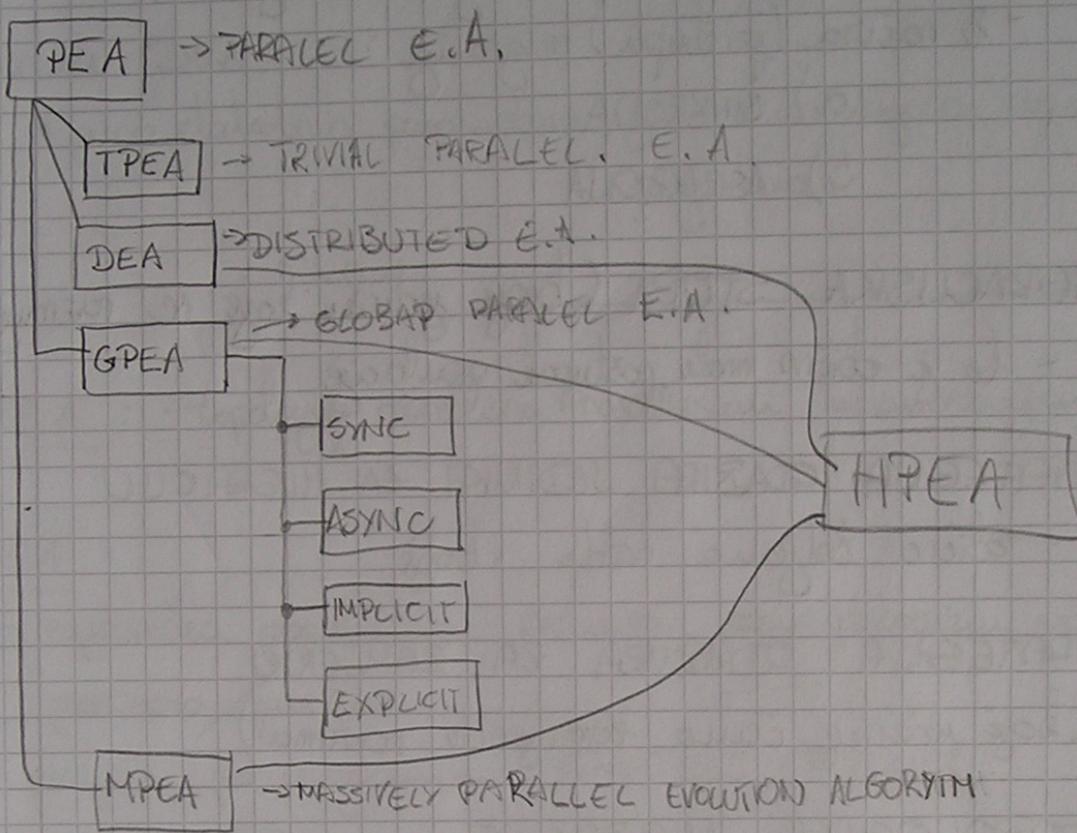
$$D = 2314$$

$$Z = 1814$$

31.05.2010.

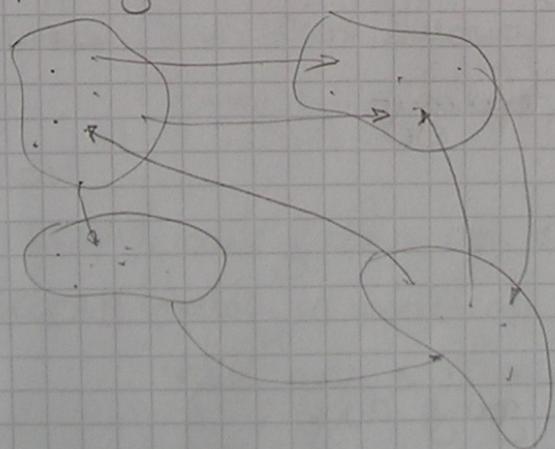
P.P.

- PARALELNI EVOLUCIJSKI ALGORITMI



TPEA - više neovisnih E.A. na razlicitim procesorima

DEA - populacija rješenja jedini dijeli se na subpopulacije - otoke (island model, structured E.A.)



- svaka subpopulacija izvodi slijedni E.A. (može biti i različiti)

- nekoliko dvačinih parametara:

1° MIGRACIJSKI INTERVAL (period razmjene jedinki među stanicama)

- može biti konstantan, slučajan ili ujetavan

- 2 načina (pristupa) migracije:

a) SINKRONA

b) ASINKRONA

2° MIGRACIJSKA STOPA (broje jedinki koje se razmjenjuju)

- to je obično mali postupak populacije

3° STRATEGIJA ODABIRA JEDINKI ZA MIGRACIJU

- obično odjedno nasta "bolje"

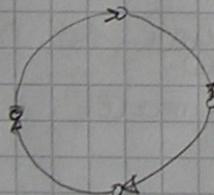
4° STRATEGIJA ODABIRA ZA ZAMJENU

(koji jedinci će učestvovati zamjeni)

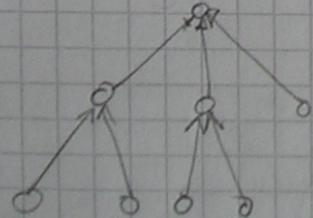
5° TOPOLOGIJA MIGRACIJE

(što će sase dati, što će od loga primati)

- povezana topologija



- stablosta topologija



- kombinacija granje dvije

- prednosti DEA :

- jednostavnost implementacije
- velika mogućnost superlinearnog ubrzanja.

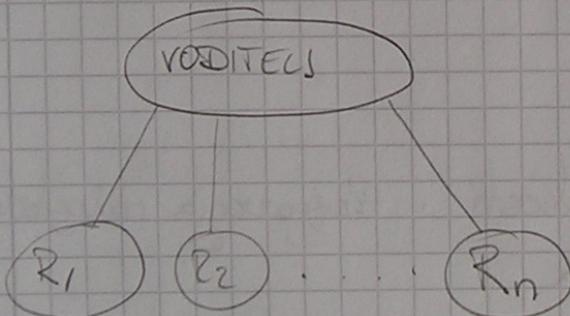
- nedostaci DEA :

- puno dodatnih parametara; jako veliki utjecaj na učinkovitost algoritma.

MPEA : - topologija određena hardverskom implementacijom procesora

GPEA :

- zajednička populacija evoluira pod utjecajem više procesora / procesa.
- podjela po ulogama : VODITELJ / RADNIK



- posao radnika : - u većini slučaja evaluacija
 - može biti i križanje, mutacija, ...

- koherencija :

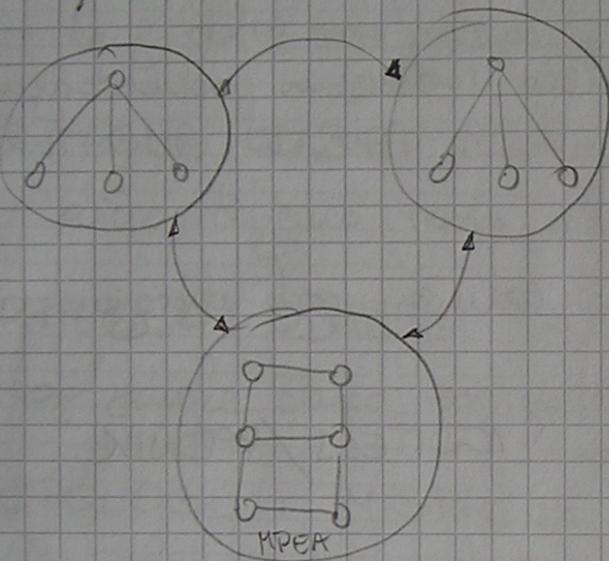
- SINKRONI (- voditelj čeka da svi radnici završe posao)
- ASINKRONI (- voditelj ne mora čekati na sve radnike,
 - voditelj radi s relativnoj jedinstvenosti)

- vrsta:

- EKSPlicitni (definicije su uloga i raspodjela posla)
- IMPLICitni (posao rachila se definira recimo o algoritmu)

HPEA : (Higrofizički Parallelni Evolucijski Algoritam)

- kombinacija nekoliko petrovnih modela
- DEA / GPEA



- prednosti:

- kombinacija prenosti i izbjegavanja nadegubaka određenih modela

ECF:

GP, ZEMRIS, FER, HR / ECF

DEA \rightarrow subpopulacija = MPI proces (1 : 1)

GPEA \rightarrow

- eksplicitna varijanta: 1 varijant / N rachila
- implicitni algoritam: 1 sljedeci algoritam / N rachila
 - raspodjela posla je odvojice celog algoritma.

- HPEA - svaka subpopulacija ima vlastiti HPI Communicator
- migracija: boristi globalni Communicator

07.06.2010.

P.P.

UBRZANJE PARALELNIH STOHALISTIČKIH ALGORITAMA

Primjer: E.A.

-VRSTE UBRZANJA:

- ABSOLUTNO (STRONG) → NEPRIMJENIVO!
- RELATIVNO (WEAK): → u odnosu na isti algoritam
↳ JEDINO MOGUĆE!

"ISTI ALGORITAM": slijedici E.A. Šeći simulirao rach
paralelnog E.A.

-TRAJANJE ALGORITMA: učetak zastavljanja:

- broj generacija / iteracija
- kvaliteta rješenja → NEPRAKTIČNO →
Nije zajamčena konvergencija do
NEKOGI RJEŠENJA!

-POSTUPAK MJESENJA:

1. Odrediti približnu kvalitetu rješenja (najčešće suboptimalna)
2. Prenuti algoritam u više instanci (~30)
3. Zastaviti sadrže medijan nekoliko rješenja svih
instanci dosegne zadani kvalitetu.

14.06.2010.

P_o P_o

1. U MPI programu svih N procesa treba odrediti.
K. O. Redoslijed ulaska u L.O. je određen
dinamički, svaki proces zna svoj redni broj, ali
ne i ostalih procesa. Određivanje sljeda procesa
ulaska rješili logoritamsku strošnu po
ritaju broja SEND operacija.

- U - br. process
 - ID - ID process
 - RBR - rechni broj process
 $(0, \dots, N-1)$

- $$\text{SCIJED} [] \quad \text{npr. RBR=2, ID=5}$$

- DSUED []

$$SLUED[i] = -1, \forall i$$

$$S \sqcup E [RBR] = ID$$

$\text{ZA } (i=0 \text{ DO } \log N)$

$$PAR = ID \text{ XOR } 2^{16} i$$

SEND (SLUED[], PAR);

RECV (PSL1LED[], PAR)); {

2A $\{j = \emptyset \text{ do } N-1\}$

AKO(P SUCED[j]! = -1)

$\text{SLUED}[j] = \text{PSLUED}[j];$

AKO (RBR == \emptyset)

| K.O. ; SEND (OK, SLUED[1]);

INAKE AKO (RBR == N - 1)

| RECV (OK, SLUED[N - 2]); KO;

INAKE

RECV (OK, SLUED[RBR - 1]); KO; SEND (OK, SLUED[RBR + 1]);

2.