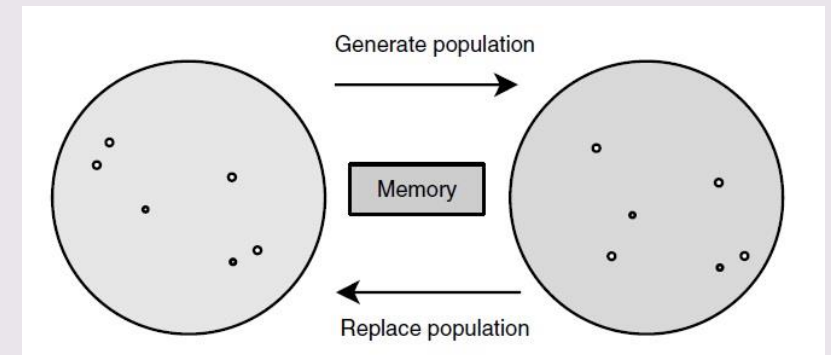


# P-HEURISTIKE

Heuristike bazirane na populaciji

# Zajednički koncepti heuristika baziranih na populaciji

- Započinju inicijalnom populacijom
  - *Kod nekih heuristika rešenja mogu biti prazna ili parcijalna*
- Iterativno se ponavlja generisanje nove populacije i zamena trenutne populacije
  - *Neke heuristike nemaju memoriju, pa se ovo obavlja na osnovu trenutne populacije*
- Većina P-heuristika je inspirisana prirodom
- Evolutionary Algorithms, Ant Colony Optimization, Scatter Search, Particle Swarm Optimization, Bee Colony, Artificial Immune Systems...



**Algorithm 3.1** High-level template of P-metaheuristics.

```
 $P = P_0$ ; /* Generation of the initial population */  
 $t = 0$ ;  
Repeat  
  Generate( $P_t$ ); /* Generation a new population */  
   $P_{t+1} = \text{Select-Population}(P_t \cup P_t')$ ; /* Select new population */  
   $t = t + 1$ ;  
Until Stopping criteria satisfied  
Output: Best solution(s) found.
```

# Swarm Optimization

- Sistem inteligentnih agenata inspirisanim kolektivnim ponašanjem socijalnih insekata i drugih životinja koji kolektivno žive
  - *Mravi, pčele*
  - *Vukovi, svetleći crvi, veverice, neke vrste ptica,*
  - *Čestice, koralji,*
- Pojam uveden 1989 od strane G. Beni i J. Wang

# Optimizacija kolonijom pčela

## *Bee Colony Optimization*

- Razvijeni su različiti algoritmi zasnovani na ponašanju pčela u prirodi:
  - *Artificial Bee Colony, Bee Swarm Optimization, Bee Hive, Honey Bee Colony Algorithm, Marriage Bee Colony, ...*
- BCO uveli Lučić, Teodorović, 2001
  - *BCO – osnovna konstruktivna metoda*
  - *BCOi (BCO improvement) – popravljanje postojećih rešenja*
- Pokazala se vrlo pouzdanom pri rešavanju kombinatornih problema, kontinualnih problema, problema koji sadrže neprecizne podatke, višekriterijumske optimizacije...

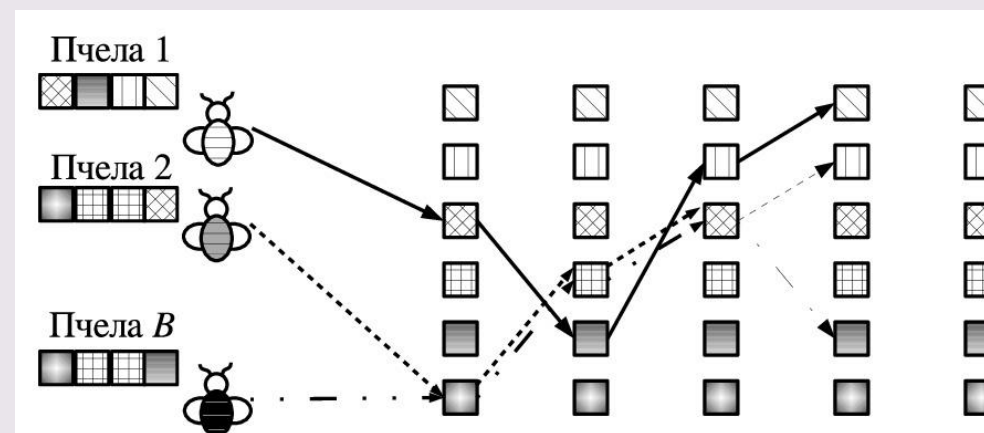
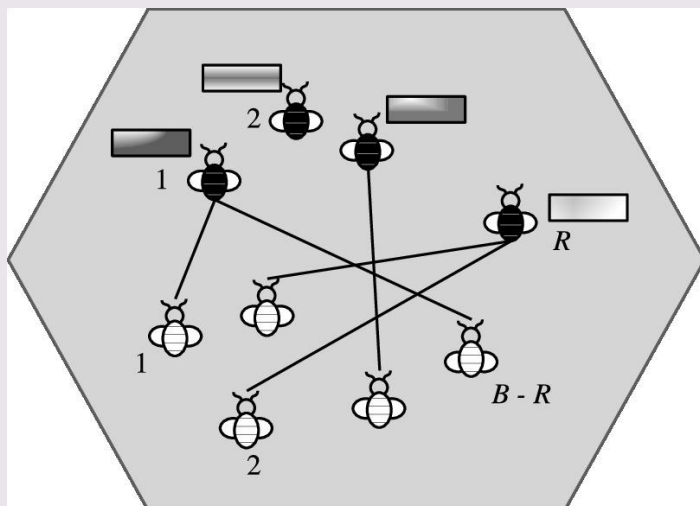
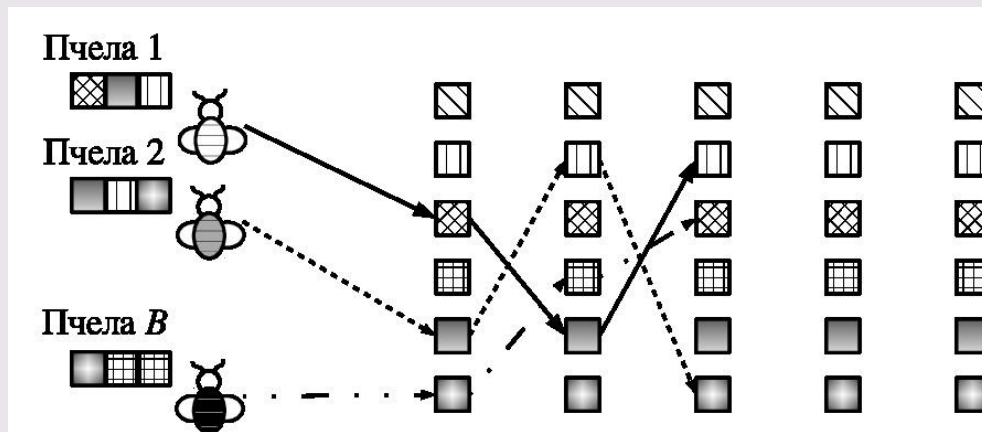
# Pčele u prirodi

- Pčele traže hranu pretražujući polja u okolini košnice
- Sakupljaju i skladište hranu
- U početnom koraku sakupljanja hrane pčele-istraživači pretražuju okolinu košnice, vraćaju se u košnicu i „plešući“ informišu druge pčele o lokaciji, količini i kvalitetu raspoložive hrane
- Ukoliko pčele odluče da napuste košnicu, one slede „uputstva“ pčela istraživača, a nakon povratka moguće je da pčela
  - *Napusti lokaciju i postaje neopredeljeni sledbenik*
  - *Nastavi sa sakupljanjem hrane, pri čemu ne regrutuje ostatak košnice*
  - *„Plesom“ regrutuje druge pčele pre nego što nastavi sa sakupljanjem hrane*
- Ukoliko više pčela istovremeno „plešući“ regrutuje nove pčele pitanje je kako pčele odlučuju koga će pratiti – verovatno odluka zavisi od kvaliteta hrane

# BCO

- Ideja je napraviti sistem multi-agenata (veštačkih pčela) koje će tražiti dobro rešenje
- Kolonija veštačkih pčela je često dosta mala – 5 do 15 pčela
- Tokom pretrage pčele sarađuju i razmenjuju informacije
- Svaka pčela je odgovorna za jedno rešenje
- Tokom algoritma smenjuju se dve faze:
  - *Let unapred (forward pass) – pretražuje se prostor rešenja i primenjuju koraci za izgradnju/popravljjanje rešenja*
  - *Let unazad (backward pass) – razmena informacija*
- Algoritam se odvija po iteracijama
- Tokom jedne iteracije let unapred i let unazad se smene NC puta

# BCO



# VCO

---

## Алгоритам 4.4: VCO алгоритам

---

**Do**

Иницијализација (празног) решења за сваку пчелу;

**For** ( $s = 0; s < NC; s ++$ )

//Лет унапред

**For** ( $b = 0; b < B; b ++$ )

Израчунавање могућих корака;

Случајан избор и примена једног корака;

//Лет уназад

**For** ( $b = 0; b < B; b ++$ )

Оцена (делимичног/комплетног) решења за пчелу  $b$ ;

**For** ( $b = 0; b < B; b ++$ )

Одлука о лојалности пчеле  $b$  тренутном решењу;

**For** ( $b = 0; b < B; b ++$ )

**If** ( $b$  је неодређена) случајан избор регрутера;

Оцена свих решења и проналажење најбољег;

**While** (критеријум за заустављање није испуњен);

---



# BCO – Ocena rešenja

- Ako je  $C_b$  vrednost funkcije cilja rešenja pčele  $b$ , normalizovana vrednost je

- Za problem minimizacije  $O_b = \frac{C_{max} - C_b}{C_{max} - C_{min}}, b = 1, 2, \dots, B$

- Za problem maksimizacije  $O_b = \frac{C_b - C_{min}}{C_{max} - C_{min}}, b = 1, 2, \dots, B$

# BCO - Odluka lojalnosti

- Nakon konstrukcije/popravljanja rešenja svaka pčela donosi odluku o lojalnosti
- Verovatnoća da pčela  $b$  ostaje lojalna svom rešenju

$$p_b^{u+1} = e^{-\frac{O_{max}-O_b}{u}}, b = 1, 2, \dots, B$$

- $O_b$  - normalizovana vrednost funkcije cilja za rešenje pčele  $b$
  - $O_{max}$  - maksimum svih normalizovanih vrednosti funkcije cilja
  - $u$  - brojač izvršenih letova unapred čije su vrednosti  $1, 2, \dots, NC$
- Na osnovu date formule i generatora slučajnih brojeva donosi se odluka za svaku pčelu posebno
    - Ako je generisani broj manji od dobijene verovatnoće pčela ostaje lojalna svom rešenju
    - U suprotnom pčela postaje neopredeljeni sledbenik

# BCO – Proces regrutacije

- Ako je pčela neopredeljeni sledbenik donosi odluku o tome kog regrutera će pratiti

$$p_b = \frac{O_b}{\sum_{k=1}^R O_k}, b = 1, 2, \dots, R$$

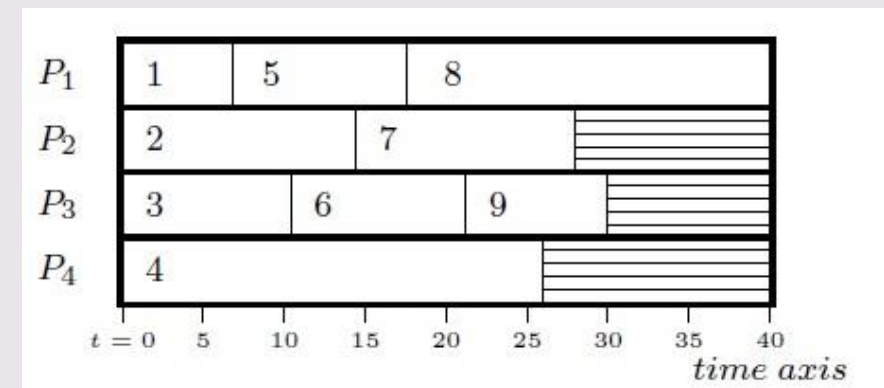
- $O_k$  - normalizovana vrednost  $k$ -tog reklamiranog rešenja
  - $R$  - broj regrutera
- Odluka se donosi korišćenjem rulet točka

# BCO – Globalno (kolektivno) znanje

- Među iteracijama ne postoji ramena informacija
  - *Omogućava diverzifikaciju prostora pretrage*
  - *Obećavajući delovi prostora pretrage mogu biti nedovoljno istraženi*
- Omogućiti razmenu iteracija među iteracijama – količina razmenjenih informacija se određuje eksperimentalno
- Propagirati najbolja rešenja za narednu iteraciju
- Na početku iteracije, za odabir inicijalne populacije, osim novih generisanih rešenja uzeti u obzir i rešenja iz prethodne populacije, kao i do tada najbolje generisano rešenje

# BCO za raspoređivanje procesa

- Neka je
  - $T = \{1, 2, \dots, n\}$  skup nezavisnih procesa
  - $P = \{1, 2, \dots, m\}$  skup identičnih procesora
  - $l_i$  vreme izvršavanja procesa  $i$  ( $i = 1, 2, \dots, n$ )
- Svaki proces treba da bude dodeljen tačno jednom procesoru i svaki proces se izvršava tačno jedanput
- Pronaći raspored procesa po procesorima tako da je ukupno vreme izvršavanja minimalno



# BCO za raspoređivanje procesa

## 1. Reprezentacija rešenja

- Matrica  $S_{m \times n}$  gde element  $s_{ji}$  predstavlja indeks  $i$ -tog procesa na  $j$ -om procesoru – iako redosled procesa nije bitan omogućava direktno enkodiranje rešenja
- Vektor  $A = \{a_1, \dots, a_m\}$  - broj procesa na svakom procesoru
- Vektor  $Y = \{y_1, \dots, y_m\}$  - vreme izvršavanja na svakom procesoru,  $y_j = \sum_{i=1}^{a_j} l_{s_{j0}}$
- Svakoj pčeli je na početku dodeljena po jedna struktura  $(S, A, Y)$

## 2. Funkcija cilja – maksimalni element vektora $Y$

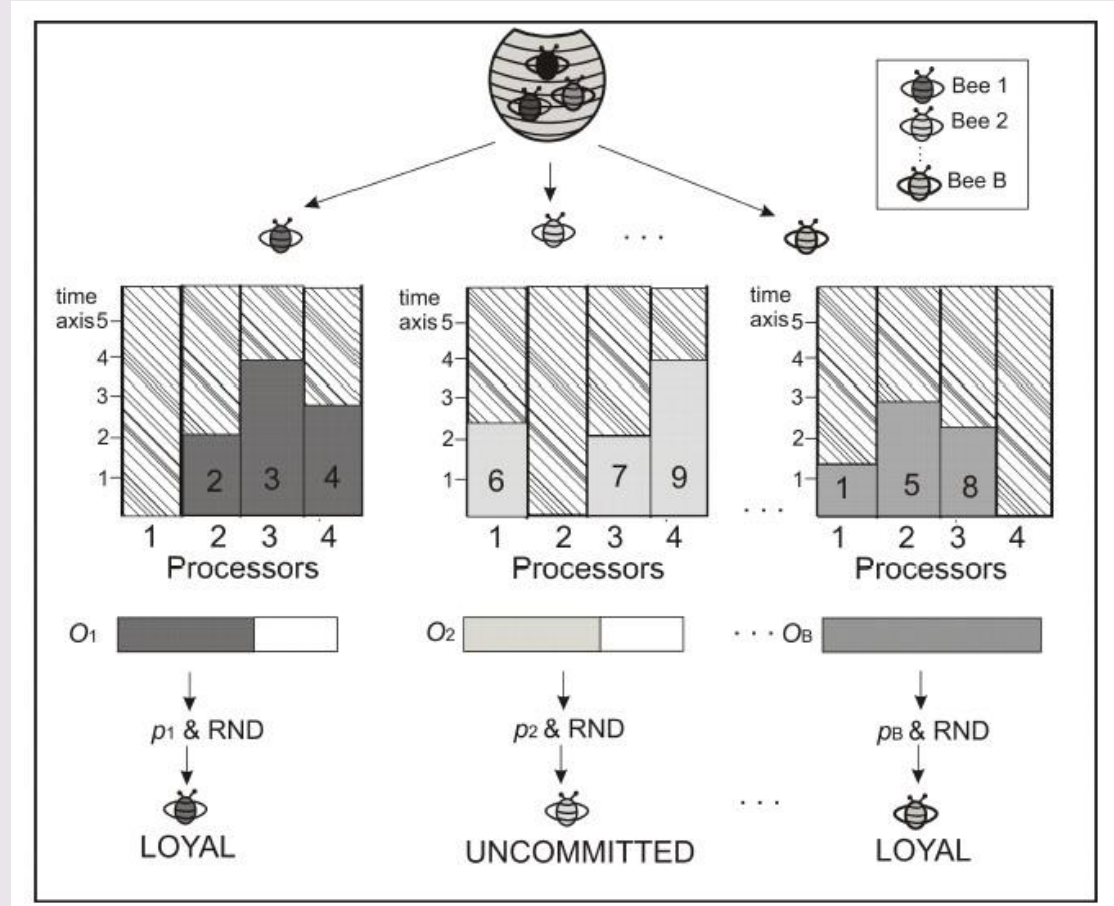
## 3. Inicijalna populacija – prazne strukture

# BCO za raspoređivanje procesa

4. Let unapred – u svakom letu unapred svaka svaka pčela bira  $NC$  parova proces-procesor i dodaje svom parcijalnom rešenju
  - Verovatnoća da pčela izabere proces  $i$  -  $p_i = \frac{l_i}{\sum_{k=1}^K l_k}$ ,  $i = 1, 2, \dots, n$ ,  $K$  – broj preostalih procesa – pčela bira proces na osnovu generatora slučajnih brojeva
  - Verovatnoća da pčela izabere procesor  $j$  -  $p_j = \frac{V_j}{\sum_{k=1}^m V_k}$ ,  $j = 1, 2, \dots, m$ ,  $V_j = \frac{\max y - y_j}{\max y - \min z}$
5. Let unazad – nije problemski zavistan, pa se obavlja na uobičajeni način

# BCO za raspoređivanje procesa

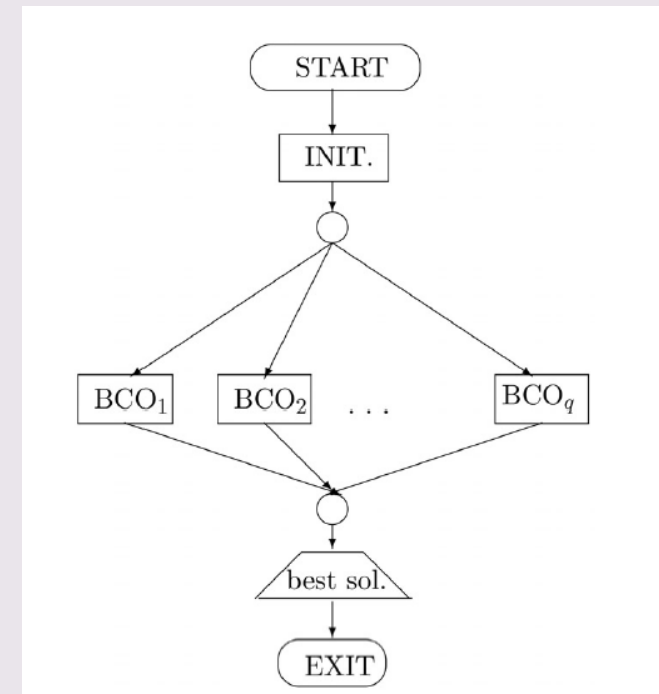
- Nakon 3 leta unapred za  $NC = 1$





# BCO – Distribuirana izvršavanja

- Nezavisna paralelna pokretanja BCO algoritma – Distributed BCO
  - *Kriterijum za zaustavljanje  $runtime/q$*
  - *Paralelizacija iteracija*
- Podela populacije na više manjih populacija  $B/q$  – BBCO
  - *Kriterijum za zaustavljanje nepromenjen*
- Kombinacija DBCO i BBCO – Multiple BCO



# BCO – Sinhrona paralelizacija

- Razmena informacija među manjim kolonijama koje se paralelno izvršavaju
  - *Fiksiran broj puta*
  - $n_{it}/(10 \cdot q)$
- Ukoliko pčela razvija rešenje koje ne može da „pobedi“ trenutno najbolje ona odustaje od svog rešenja i kreće iz početka

# BCO – Asinhrona paralelizacija

- Centralizovana razmena informacija – *central blackboard*
  - Ukoliko se najbolje rešenje u koloniji ne popravi u fiksiranom broj iteracija preuzima se rešenje sa centralne table
  - Ukoliko kolonija dođe do boljeg rešenja od onog na centralnoj tabli o tome obaveštava ostale kolonije, tj. upisuje na tablu rešenje
- Necentralizovana razmena informacija
  - Postoji više „tabli“
  - Ring arhitektura

