

# Heterogeni večagentni sistem za reševanje dinamičnih optimizacijskih problemov

Aleš Čep

Iztok Fister



# Pregled

- Opis problema
- DE, jDE in PSO
- Predlagani algoritem
- Poskusi in rezultati
- Zaključek



# Dinamični optimizacijski problemi

- Problemi, kjer se okolje med izvajanjem spreminja.
- Sprememba okolja se kaže v spremembi sistemskih nadzornih parametrov ali v spremembi ocenitvene funkcije.



# Diferencialna evolucija (*angl. Differential Evolution*)

- Algoritem uporabljamo za optimizacijo problemov z realno-kodirano predstavitvijo rešitev.
- Evolucijska zanka sestoji iz:
  - **mutacije** ('DE/rand/1/bin'),
  - **križanja** ter
  - **selekcije**.
- **jDE** (Brest in ostali, 2006):
  - nadzorna parametra  $F$  and  $CR$  sta dodana v predstavitev posameznika,
  - dodan je mehanizem za samoprilagajanje parametrov  $F$  and  $CR$ .



# Optimizacija z roji delcev (*angl.* Particle swarm optimization)

- Vzor je obnašanje jate ptičev, ki predstavljajo delce.
- Vsak delec v roju je definiran z lokacijo ter hitrostjo.
- Shranimo lokacijo najboljšo rešitev celotnega roja in najboljše lokacije posameznih delcev v roju.



# Predlagana metoda - splošno

- Heterogeni večagentni sistem
- Agent
- Velikost populacije
- Zaznavanje spremembe okolja
- Izmenjava informacij

# Agent

- Samostojna računska entiteta v večagentnem sistemu.
- Izbira optimizacijskega algoritma (*DE*, *jDE* ali *PSO*).
- V večagentnem sistemu agenti uporabljajo različne algoritme.
- Vsak ima svojo populacijo posameznikov.
- Velikost populacije enaka v vseh agentih (Čep in ostali, 2017).

Agent 1  
Algoritem  
Populacija

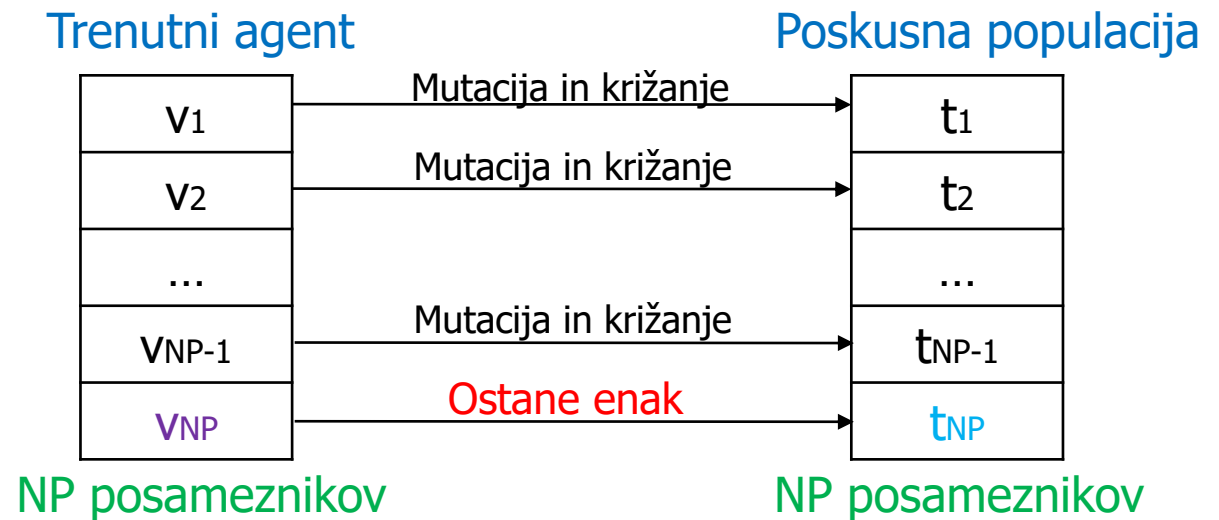
Agent 2  
Algoritem  
Populacija

Agent 3  
Algoritem  
Populacija

Agent *N*  
Algoritem  
Populacija

# Zaznavanje spremembe okolja

- Z mutacijo in križanjem ustvarimo NP-1 poskusnih posameznikov.
- Zadnji ostane enak, tj. nadzornik (angl. *sentinel*).
- Dinamično okolje simuliramo s spremembo ocenitvene funkcije ter parametrov okolja.



$f(\text{VNP}) \neq f(\text{tNP})$   
Zaznana sprememba okolja





# Raznolikost populacije v agentu

- Težava stagnacije populacije:

Majhna populacija hitro privede do majhne raznolikosti.

- **Raznolikost** izračunamo po postopku:

- določimo *povprečnega* posameznika,
- sledi seštevek Evklidskih razdalj med posamezniki v populaciji ( $x_i^{(G)}$ ) ter povprečnim posameznikom ( $a^{(G)}$ ):

$$a_j^{(G)} = \frac{\sum_{i=1}^{NP-1} x_{i,j}^{(G)}}{NP-1}$$

$$diversity = \sqrt{\sum_{j=1}^D \sum_{i=1}^{NP-1} (x_{i,j}^{(G)} - a_j^{(G)})^2}$$



# Izmenjava informacij

- Cilj je zmanjšati problem **stagnacije** populacije.
- Izmenjavo posameznikov izvedemo, ko raznolikost populacije pade pod določen **prag**.
- **Tip izmenjave** informacij je parameter sistema namenjen opredelitvi soseščin za izmenjavo teh. Definirali smo tri tipe soseščin:
  - *istovrstna* (*eq*),
  - *raznovrstna* (*neq*) in
  - *naključno-izbrana* (*rand*).

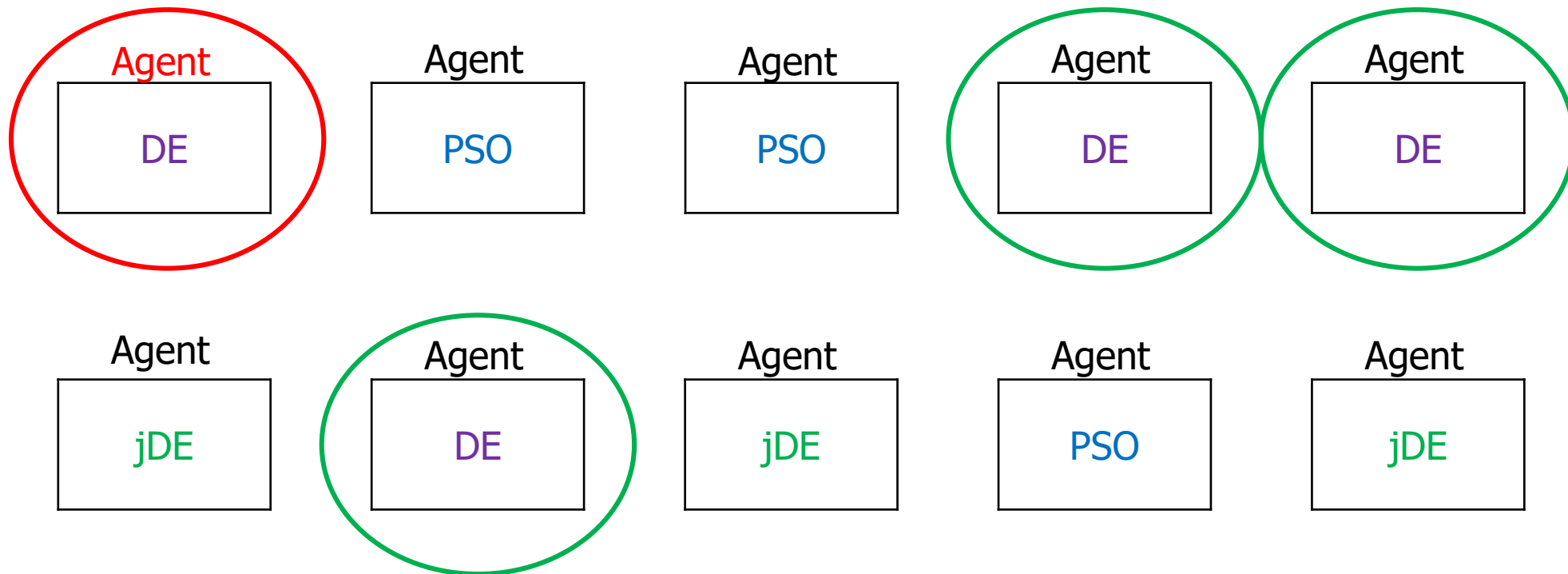
# Istovrstna soseščina

- Izmenjava lahko poteka samo med agenti, ki uporabljajo enak algoritem.



# Istovrstna soseščina

- Izmenjava lahko poteka samo med agenti, ki uporabljajo enak algoritem.



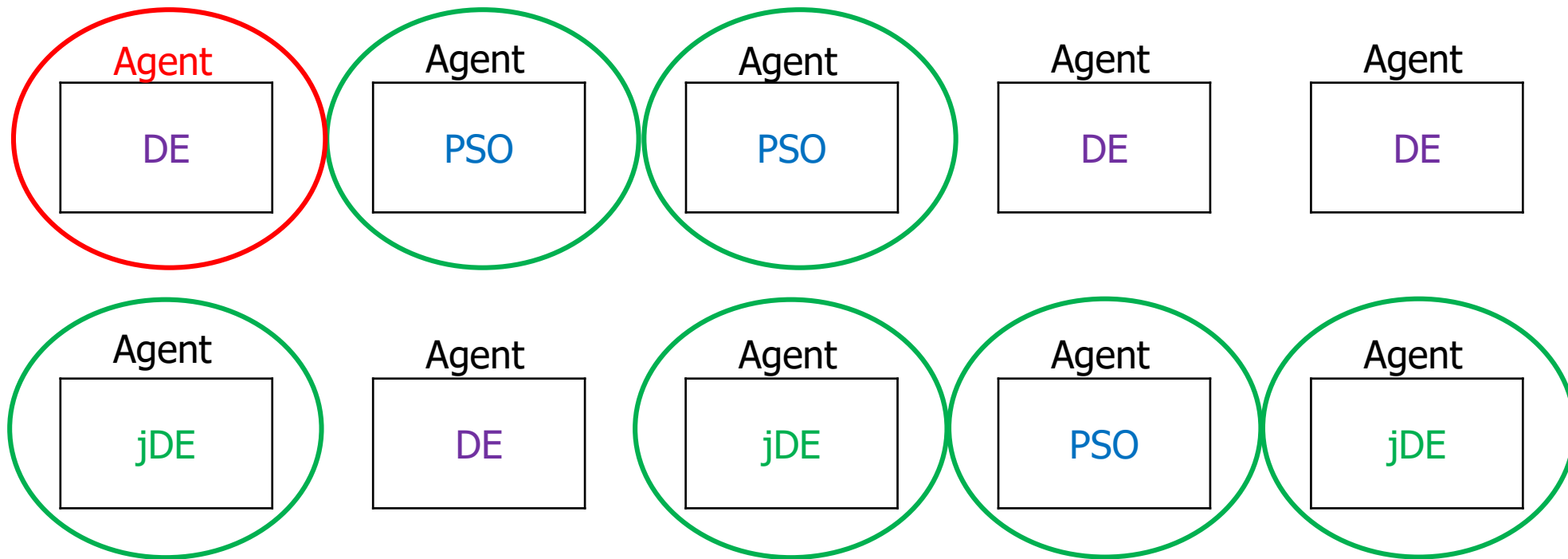
# Raznovrstna soseščina

- Izmenjava lahko poteka samo med agenti, ki uporabljajo različni algoritem.



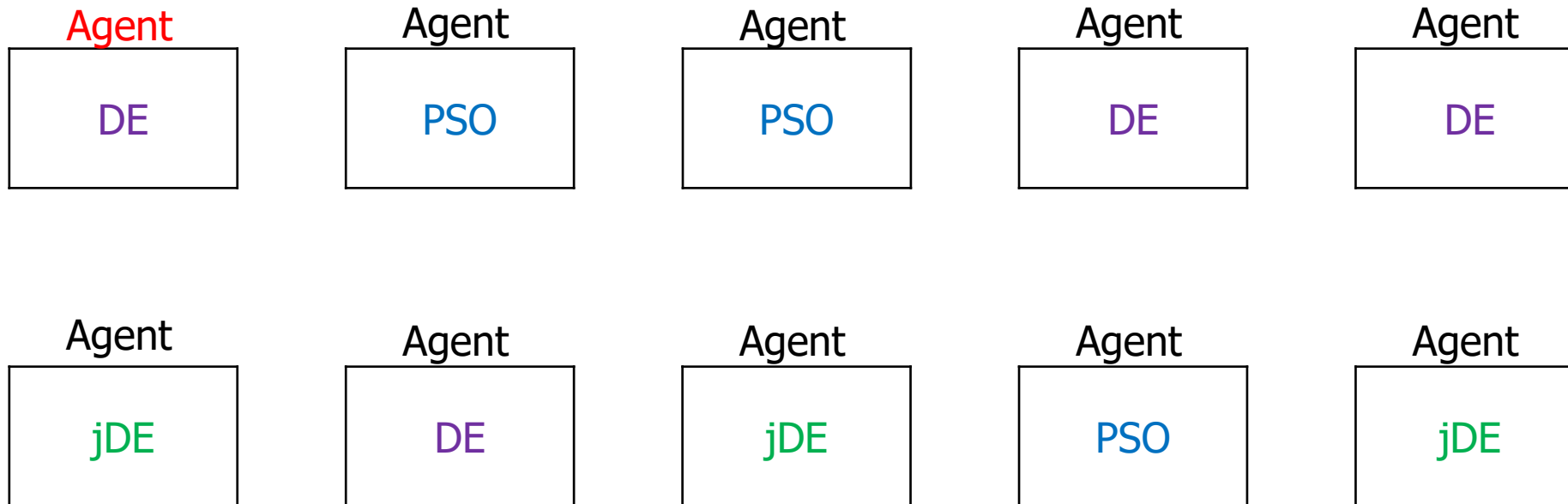
# Raznovrstna soseščina

- Izmenjava lahko poteka samo med agenti, ki uporabljajo različni algoritem.



# Naključno-izbrana soseščina

- Izmenjava lahko poteka med vsemi agenti, ne glede na izbran algoritem.



# Naključno-izbrana soseščina

- Izmenjava lahko poteka med vsemi agenti, ne glede na izbran algoritem.







# Algoritem za izmenjavo informacij

- Algoritem izmenjave za **agenta A** s prenizko raznolikostjo:
  1. Glede na **tip izmenjave** določimo soseščino agenta.
  2. Poiščemo najboljšega posameznika v vsakem agentu v soseščini.
  3. Izračunamo razdalje med najdenimi najboljšimi posamezniki ter najboljšim posameznikom v populaciji **agenta A**. Poiščemo najbolj oddaljenega posameznika (*Xdist*).
  4. V populaciji **agenta A** poiščemo *najbližja* posameznika. Slabšega izmed njiju zamenjamo z *Xdist*.

$$d(p, q) = \sqrt{\sum_{j=1}^D (p_j - q_j)^2},$$

# Predlagani večagentni sistem

- Inicializacija vseh agentov in njihov populacij.
- Ocenitev populacij v agentih.
- Za vsakega agenta v sistemu (dokler ni izpolnjen pogoj ustavljanja):
  - ustvarimo poskusno populacijo,
  - izvedemo ocenitev poskusne populacije in selekcijo,
  - je prišlo do spremembe okolja? => nova populacija,
  - posodobimo najboljšo rešitev v sistemu,
  - je raznolikost populacije prenizka? => migracija.

Inicializacija

Zanka



# Poskusi

- Izvedemo tri poskuse:
  - najti najboljšo konfiguracijo algoritmov in tipov izmenjave informacij,
  - poiskati optimalen tip izmenjave informacij,
  - predlagano metodo primerjati z obstoječimi rešitvami.
- CEC 2009 testno ogrodje za reševanje dinamičnih problemov (*Dynamic optimization benchmark CEC'09*).
- PC: Windows 10, Intel Core i7-4790, 16GB rama, C++.



# Ogrodje za izvajanje poskusov

- Tekmovanje CEC'2009 za reševanje dinamični problemov.
- 7 testnih funkcij.
- 7 tipov sprememb kontrolnih parametrov v sistemu.
- Skupaj je 49 testnih scenarijev.
- 20 neodvisnih ponovitev vsakega testa, da dobimo oceno (*mark*).
- Skupna ocena (*perf*):

$$perf = \sum_{i=1}^{49} mark_i \cdot 100$$



# Parametri

## ■ Večagentni sistem:

- Število agentov  $N$ : 10
- Velikost populacije  $NP$ : 20
- $Prag$ : 1

## ■ DE:

- $F$ : [0, 2]
- $CR$ : [0, 1]

## ■ PSO:

- $w$ : 0,729
- $c_1$ : 1,496
- $c_2$ : 1,496

## ■ jDE:

- $F_i$ : [0,1, 1]
- $CR_i$ : [0, 1]



# Poskus 1 – Iskanje najboljše konfiguracije

- **Cilj:** poiskati najboljšo konfiguracijo algoritmov in tipov izmenjave informacij.
- Iz nabora treh algoritmov smo dobili naslednje kombinacije:
  - DE in PSO, (5 : 5)
  - DE in jDE, (5 : 5)
  - PSO in jDE, (5 : 5)
  - DE, PSO in jDE. (3 : 3 : 4)
- Za vsako kombinacijo smo izvedli eksperiment s 3 tipi izmenjav, kar nam da skupaj 12 **variacij**.

# Rezultati poskusa 1

HeMAS	Exch. type	$F_1(10)$	$F_1(50)$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	Perf
DE_PSO	rand	0.08845	0.08617	0.07019	0.00898	0.07880	0.06527	0.05242	45.02752
	eq	0.08974	0.08809	0.06902	0.01064	0.06589	0.06054	0.06325	44.71810
	neq	0.09269	0.08720	0.06650	0.01287	0.06866	0.07235	0.05009	45.03783
DE_jDE	rand	0.09203	0.09285	0.09580	0.04501	0.09234	0.14219	0.09763	65.78644
	eq	0.09192	0.09242	0.09478	0.04536	0.09128	0.14044	0.09519	65.13922
	neq	0.09200	0.09283	0.09588	0.04500	0.09237	<b>0.14233</b>	0.09756	65.79691
PSO_jDE	rand	<b>0.09736</b>	<b>0.09515</b>	0.09459	0.04520	0.10090	0.13812	0.10078	67.20946
	eq	0.09414	0.09424	0.10434	0.04632	0.10322	0.13944	0.09702	67.87195
	neq	0.09664	0.09392	0.10284	<b>0.04804</b>	<b>0.11506</b>	0.13986	<b>0.10903</b>	<b>70.54091</b>
DE_jDE_PSO	rand	0.09384	0.09263	<b>0.10588</b>	0.03927	0.08725	0.13722	0.09179	64.78680
	eq	0.09410	0.09226	0.10206	0.03951	0.10731	0.13444	0.09606	66.57406
	neq	0.09402	0.09328	0.09937	0.04220	0.10009	0.13771	0.09775	66.44273
								Average:	61.24433

- Najboljše rezultate smo dobili s kombinacijo algoritmov **PSO** in **jDE** s tipom *različna soseščina*.

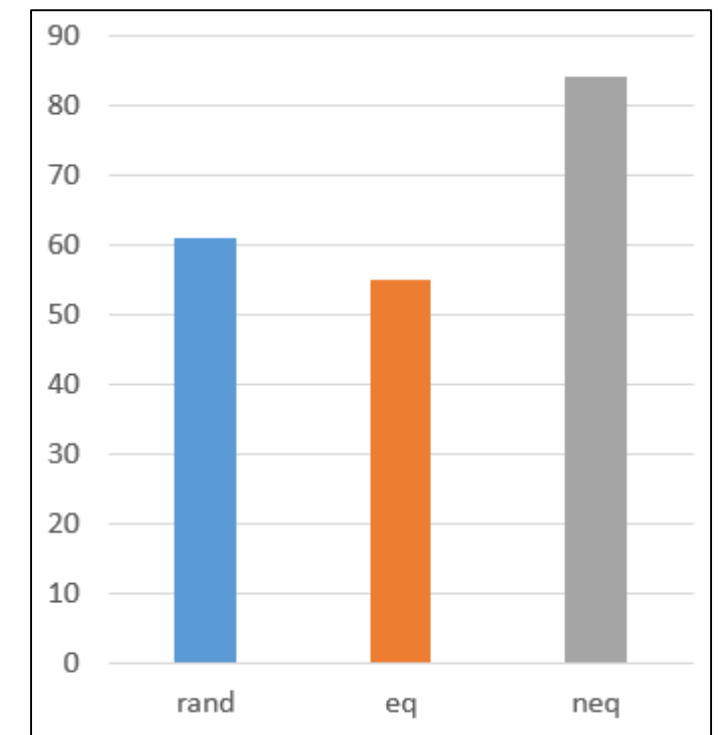
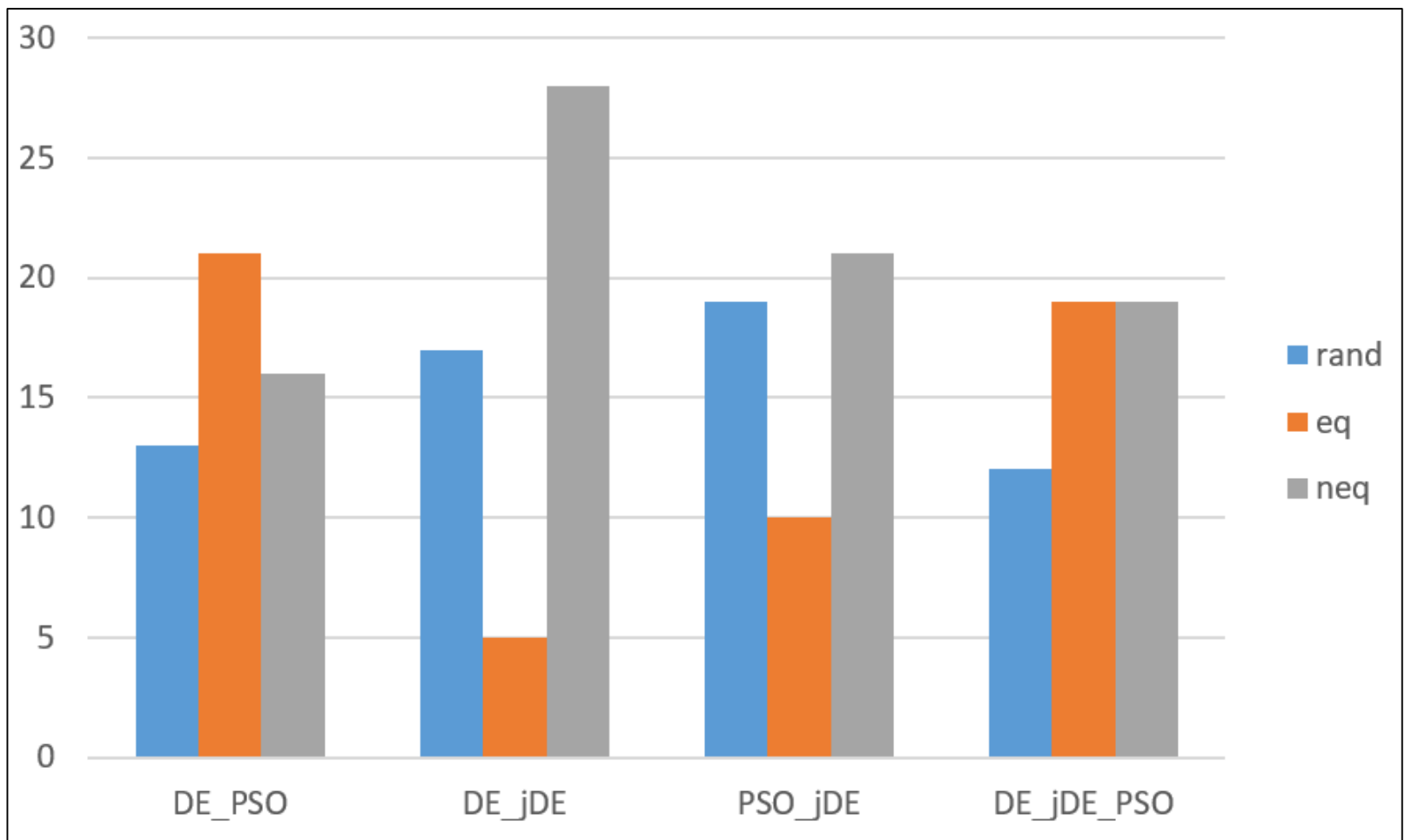
## Poskus 2 – Vpliv tipa izmenjave informacij

- **Cilj:** poiskati optimalen tip izmenjave informacij.
- Za vsako variacijo smo dobili 50 rezultatov (49 vmesnih rezultatov za vsako funkcijo in tip spremenbe + *perf*).
- Primerjali smo rezultate različnih tipov izmenjave informacij pri enaki kombinaciji algoritmov.





# Rezultati poskusa 2



# Poskus 3 – Primerjava z obstoječimi algoritmi

- **Cilj:** primerjava rezultatov večagentnih sistemov z rezultati obstoječih algoritmov za reševanje dinamičnih problemov.
- Primerjava je bila izvedena na istem testnem okolju.
- Primerjavo smo izvedli z naslednjimi algoritmi:
  - jDE za dinamične probleme [2] (jDE\*),
  - MLSDO [3],
  - mSQDEi [4],
  - PLBA [5],
  - DASA [6],
  - bCCEA [7] ter
  - Multi-DEPSO [8].

# Rezultati poskusa 3

Algorithm	$F_1(10)$	$F_1(50)$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	Perf
PSO_jDE- <i>neq</i>	0.09664	0.09392	0.10284	0.04804	0.11506	0.13986	0.10903	70.54091
jDE*	0.09418	0.09373	0.10908	0.05151	0.10779	0.14159	0.09939	69.72684
MLSDO	0.09783	0.09666	0.13930	0.06741	0.13539	0.15182	0.12435	<b>81.276</b>
mSQDE-i	<b>0.09940</b>	0.09790	<b>0.14520</b>	0.03353	<b>0.14195</b>	0.14006	0.11340	77.14583
PLBA	0.08157	0.07868	0.06817	0.02187	0.05903	0.08098	0.06693	45.72351
DASA	0.0916	0.09074	0.11245	0.03361	0.09902	0.13501	0.08959	65.202
bCCEA	0.06848	0.06551	0.04308	<b>0.11480</b>	0.03844	0.04656	0.03752	41.43782
Multi-DEPSO	0.09904	<b>0.09839</b>	0.14032	0.05119	0.13599	<b>0.15659</b>	<b>0.12748</b>	80.9

- Predlagani **PSO\_jDE-*neq*** je dosegel **boljše** rezultate kot jDE\*, PLBA, DASA in bCCEA.
- **Izziv** za prihodnost pa so algoritmi MLSDO, mSQDE-i in Multi-DEPSO.



# Povzetek

- Predstavili smo heterogeni večagentni sistem za reševanje dinamičnih optimizacijskih problemov.
- Agenti izvajajo različne optimizacijske algoritme (DE, PSO ali jDE), vsak s svojo populacijo.
- Agenti avtonomno zaznavajo spremembe v okolju.
- Težave s stagnacijo populacije smo omilil z merjenjem raznolikosti populacije ter migracijskim mehanizmom.
- Večagentni sistem je izboljšal rezultate (nekaterih) najsodobnejših algoritmov.



# Načrti za prihodnje

- Povečati nabor algoritmov.
- Izvesti analizo vpliva migracij posameznikov na raznolikost populacije.
- Obravnava rezultatov meritev raznolikosti.
- Dodati mehanizme za povečanje raznolikosti populacije tekom izvajanja (npr. *Novelty Search*).

# Hvala za vašo pozornost!



# Literatura

- [1] Cep A, Fister I (2017) Multi-agent system based on self-adaptive differential evolution for solving dynamic optimization problems. In: Proceedings of the 2017 4th Student Computer Science Research Conference - StuCoSReC, Koper: University of Primorska Press, str. 35-41, URL <http://www.hippocampus.si/ISBN/978-961-7023-41-1/mobile/index.html>.
- [2] Brest J, Zamuda A, Bošković B, Maučec MS, Žumer V (2009) Dynamic optimization using self-adaptive differential evolution. In: Evolutionary Computation, 2009. CEC'09. IEEE Congress on., IEEE, pp 415-422.
- [3] Lepagnot J, Nakib A, Oulhadj H, Siarry P (2013) A multiple local search algorithm for continuous dynamic optimization. Journal of heuristics 19(1):35-76.
- [4] Novoa-Hernandez P, Corona CC, Pelta DA (2013) Self-adaptive, multipopulation differential evolution in dynamic environments. Soft computing 17(10):1861-1881.
- [5] Hussein WA, Abdullah SNHS, Sahran S (2017) The patch-levy-based bees algorithm applied to dynamic optimization problems. Discrete dynamics in Nature and Society.
- [6] Korošec P, Šilc J (2009) The differential ant-stigmergy algorithm applied to dynamic optimization problems. In: Evolutionary Computation, 2009. CEC'09. IEEE Congress on., IEEE, pp 407-414.
- [7] Au CK, Leung HF (2014) Cooperative coevolutionary algorithms for dynamic optimization: an experimental study. Evolutionary Intelligence 7(4):201-218.
- [8] Zuo X, Xiao L (2014) A de and pso based hybrid algorithm for dynamic optimization problems. Soft Computing 18(7):1405-1424.