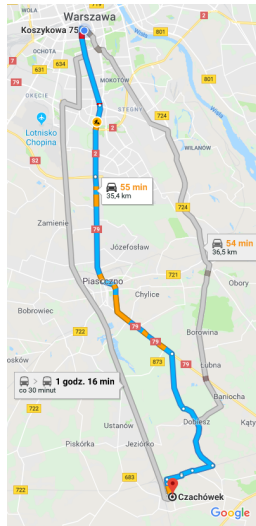
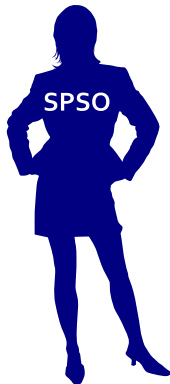


Jak wrócić dziś do domu?



Jak bym to rozwiązał?



vecteezy.com

Jak byśmy to rozwiązali?



vecteezy.com

Optymalizacja rojem cząstek: hybrydyzacja i samoadaptacja

Mateusz Uliński, Adam Żychowski, Jakub Fijałkowski,
Marcin Chudy, Piotr Jastrzębski, Marcin Bogucki, Artur
Niziołek, Jędrzej Krauze, Mateusz Zaborski,
Hubert Kordulewski,
Michał Okulewicz

Wydział Matematyki i Nauk Informatycznych
Politechnika Warszawska

Plan prezentacji

① Definicja problemu

② PSO

Hybrydyzacja

Adaptacja

③ Rozwój

Definition (Problem optymalizacyjny)

Problem optymalizacyjny $\mathcal{P} = (\Omega, f, \succ)$ jest zdefiniowany przez przestrzeń rozwiązań Ω , funkcję jakości^a $f : \Omega \rightarrow \mathbb{R}$ oraz relację $\succ \in \{<, >\}$. Rozwiązaniem problemu jest znalezienie zbioru elementów (elementu) $\mathcal{X} \subseteq \Omega$ spełniającego następujące warunki:

$$\mathcal{X} = \{x \in \Omega : \forall_{x' \in \Omega} f(x) \succeq f(x')\}$$

^aW zależności od dziedziny funkcja jakości może być też określana jako funkcja celu lub funkcja przystosowania.

Przestrzeń rozwiązań i typ problemu

Podczas tej prezentacji uznajemy, że:

$$\Omega := \mathbb{R}^n \tag{1}$$

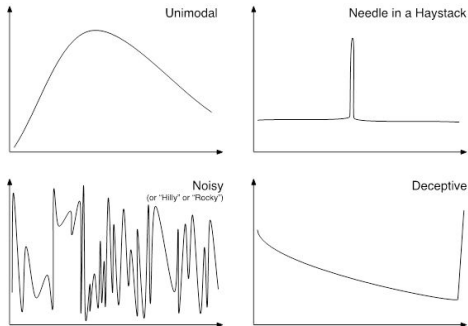
$$\succ := < \tag{2}$$

Jaki jest najlepszy algorytm optymalizacji globalnej?

- Przeszukiwanie losowe?
- *Hill climbing*?
- Pełzający simplex (Metoda Nelder-Mead)?
- Metody quasi-Newtonowskie (np. L-BFGS)?
- Algorytmy genetyczne / ewolucyjne?
- Inne metody populacyjne?
- Solvery programów liniowych (CPLEX itp.)?
- CMA-ES?

Jak rozwiązać wszystkie problemy jednym algorytmem?

<http://slideplayer.com/slide/4579308/>



- No free lunch in optimization (Wolpert and Macready, 1997)
- No free lunch in machine learning (Wolpert, 1996)

A zatem...?



A zatem...?

Wyberzmy taki, który daje
przyzwoite wyniki i umiemy go
parametryzować

Optymalizacja rojem cząstek

- Inspirowana: Reynolds (1987)
- Wprowadzona: Kennedy and Eberhart (1995)
- Usprawniana: Shi and Eberhart (1998),
- Porządkowana: Bratton and Kennedy (2007)
- Analizowana: Van Den Bergh and Engelbrecht (2010)

Operatory przeszukiwania

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i + \mathbf{v}_{t+1}^i. \quad (3)$$

$$\mathbf{v}_{t+1}^i = \omega \cdot \mathbf{v}_t^i + c_1 \cdot (\mathbf{p}_{best}^i - \mathbf{x}_t^i) + c_2 \cdot (\mathbf{neighbors}_{best}^i - \mathbf{x}_t^i) \quad (4)$$

Jak dobierać jej parametry i sąsiedztwo?

- Prace: Cleghorn, Clerc, Engelbreht, Harrison, Poli (Harrison et al., 2017)

Równanie Poli'ego na zbieżność PSO

$$c_1 + c_2 < \frac{24(1 - \omega)^2}{7 - 5\omega} \quad (5)$$

Rejon zbliżeności Poli'ego

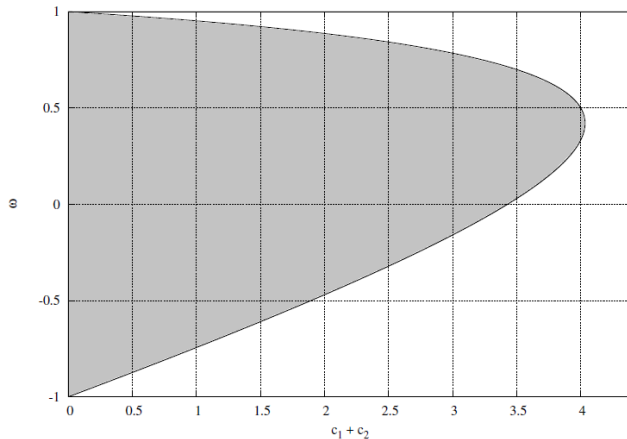


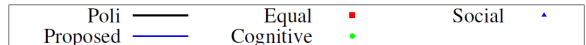
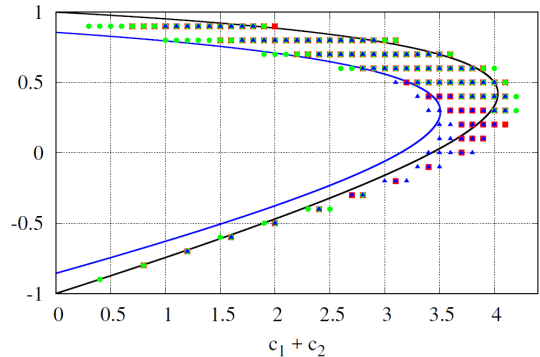
Fig. 1. Visualization of Poli's theoretically defined region for convergent behaviour of PSO parameters.

Od przybytku głowa nie boli

CONTROL PARAMETER VALUES

ω	c_1	c_2
0.7298	1.49618	1.49618
0.729	2.0412	0.9477
0.6	1.7	1.7
0.721	1.193	1.193
0.715	1.7	1.7
0.724	1.468	1.468
0.785	1.331	1.331
0.837	1.255	1.255
0.42	1.55	1.55
0.711897	1.711897	1.711897
0.5	1.90	1.90
0.6	1.80	1.80
0.1	0.950	2.850
-0.1	0.875	2.625

3



Od przybytku głowa nie boli c.d.

- Particle Swarm Optimization
 - Charged PSO
 - Quantum PSO
 - Fully-Informed Swarm
 - Local PSO
 - Unified PSO
 - Orthogonal Learning PSO
- Artificial Bee Colony
- Bat Algorithm
- Cuckoo search (via Lévy flights)
- Dog Group Wild Chase and Hunt Drive Algorithm
- Eagle strategy (using Lévy walk and firefly algorithms)
- Firefly algorithm
- Glowworm swarm based optimization

Sörensen na ratunek!

Metaheuristics – the metaphor exposed Sörensen (2015)

Z polityki publikacyjnej *Journal of Heuristics*

Proposing new paradigms is only acceptable if they contain innovative basic ideas, such as those that are embedded in classical frameworks like genetic algorithms, tabu search, and simulated annealing. The Journal of Heuristics avoids the publication of articles that repackage and embed old ideas in methods that are claimed to be based on metaphors of natural or manmade systems and processes. These so-called “novel” methods employ analogies that range from intelligent water drops, musicians playing jazz, imperialist societies, leapfrogs, kangaroos, all types of swarms and insects and even mine blast processes (Sörensen, 2013).

Ale co z tego wynika...?

Spróbujmy poszukać
alternatywnego sposobu
poprawy jakości działania
algorytmów przez ich
hybrydyzację

Spojrzenie uogólnione

- Cząsteczki nie muszą być identyczne
- Każda cząsteczka ma aktualną pozycję x , prędkość v oraz najlepszą pozycję p_{best} (a potencjalnie pełną historię)
- Wszystkie cząsteczki mogą mieć dostęp do x i p_{best} innych
- Okulewicz (2016); Uliński et al. (2018):

$$\begin{aligned} \mathbf{v}_{t+1}^i = & \omega \cdot \mathbf{v}_t^i + c_1 \cdot (\mathbf{p}_{best}^i - \mathbf{x}_t^i) \\ & + \sum_{k=1}^{|\mathcal{N}|} \sum_{j=1, j \neq i}^{|\text{particles}|} I(jth \in N_k(ith)) c'_{j,k} \cdot (\mathbf{p}_{best}^j - \mathbf{x}_t^i) \\ & + \sum_{k=1}^{|\mathcal{N}|} \sum_{j=1, j \neq i}^{|\text{particles}|} I(jth \in N_k(ith)) c''_{j,k} \cdot (\mathbf{x}_t^j - \mathbf{x}_t^i) \end{aligned} \quad (6)$$

Inne algorytmy: Ewolucja różnicowa

- Wprowadzona: Storn and Price (1997)
- Od razu z założeniem wysokopoziomowej parametryzacji (DE/rand/1/bin, DE/best/1/bin)

Operatory przeszukiwania

$$\mathbf{y}_t^{(i)} = \mathbf{x}_t^{(i)} + F \times (\mathbf{x}_t^{(i_2)} - \mathbf{x}_t^{(i_1)}) \quad (7)$$

$$\mathbf{u}_t^i = \text{Bin}_p(\mathbf{x}_t^{\text{best}}, \mathbf{y}_t^{(i)}) \quad (8)$$

$$\mathbf{u}_t^i = \begin{cases} \mathbf{u}_t^i & \text{if } f(\mathbf{u}_t^i) < f(\mathbf{x}_t^i) \\ \mathbf{x}_t^i & \text{otherwise} \end{cases} \quad (9)$$

DE jako cząstka w uogólnionym PSO

$$\mathbf{x}_{test}^{(i,t+1)} = \text{Bin}(\omega \mathbf{v} + (\mathbf{p}_{best} - \mathbf{x}_{test}^{(i,t)}), \mathbf{g}_{best}) \quad (10)$$

Mój przepis na hybrydę PSO + DE

- Losowe aplikowanie wzorów PSO (4) lub DE (10) niezależnie od wcześniejszej historii danego punktu
- Utrzymywanie cech cząsteczki: prędkości, położenia aktualnego i najlepszego znanego

Hybrydy PSO-DE



- Otwartym pytaniem pozostaje sposób wyboru procentowego udziału poszczególnych rodzajów cząsteczek / zachowań
- Przegląd metod adaptacji przydatnych w takim wyborze: Sharma et al. (2018)
- W GAPSO użyto zliczania wartości sukcesów poszczególnych rodzajów:

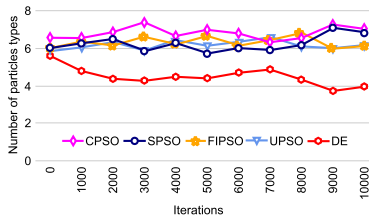
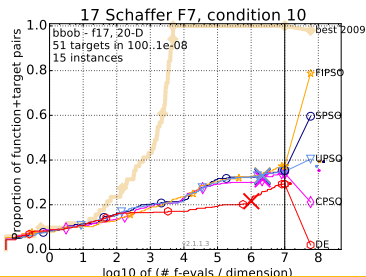
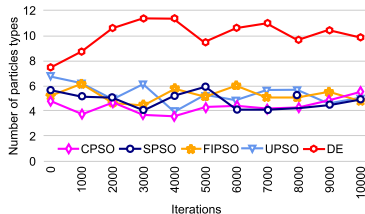
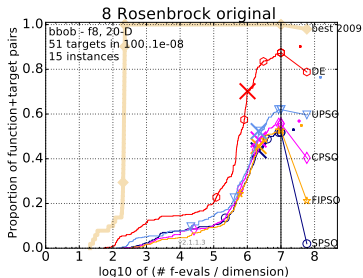
$$z_t^s = \max(0, \frac{f(\mathbf{p}_{best}^s) - f(\mathbf{x}_t^s)}{f(\mathbf{p}_{best}^s)}) \quad (11)$$

$$\hat{z}_t^p = \frac{1}{S_p * N_a} * \sum_{t=T}^{T-N_a} \sum_{s \in \text{swarm}_p} z_t^s \quad (12)$$

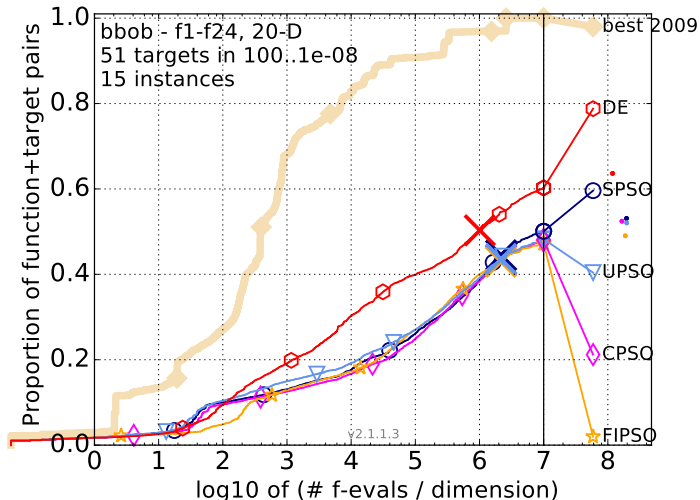
$$Z = \sum_p^P z_p \quad (13)$$

$$[\frac{z_1}{Z}, \frac{z_2}{Z}, \dots, \frac{z_P}{Z}] \quad (14)$$

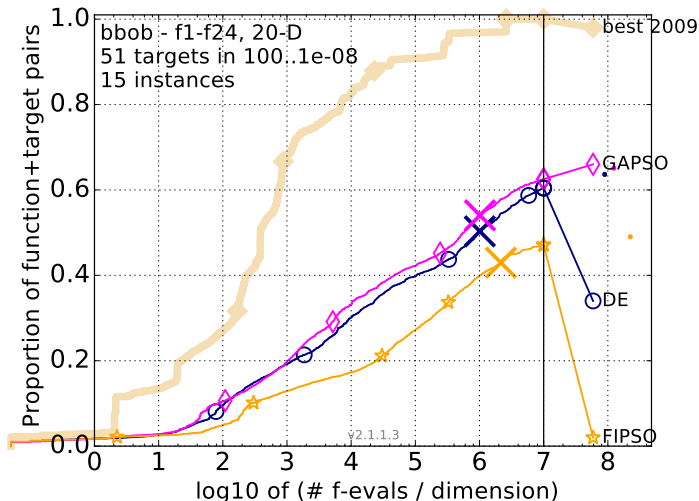
Garść wyników: BBOB



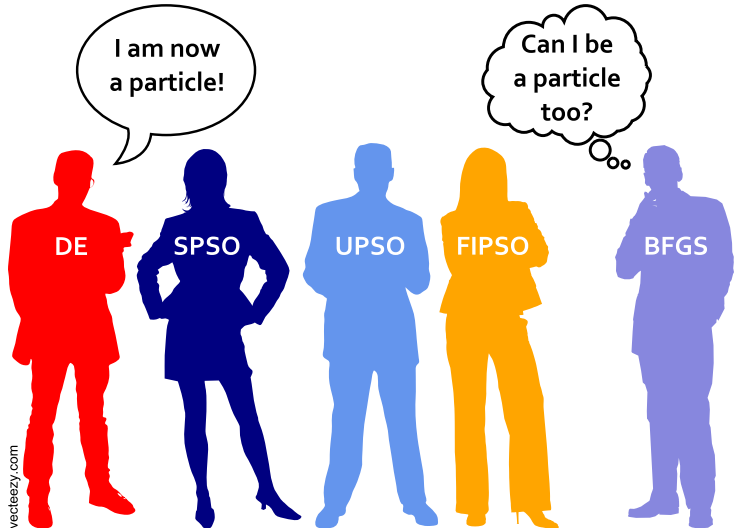
Garść wyników: BBOB



Garść wyników: BBOB



Co dalej...?



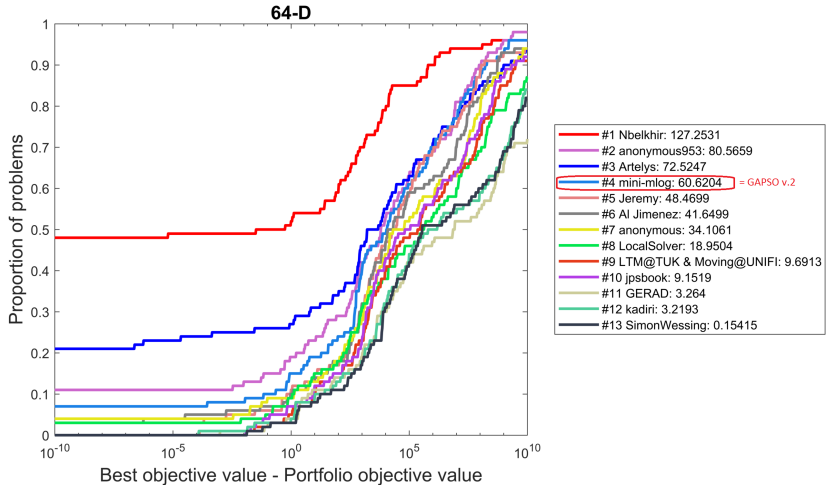
Garść wyników: BBComp 10BJ

rank	participant	method name	method description	software	paper	score ↑	sum of ranks ↓
1	Nbelkhir	Feature Based Algorithm Selection	Feature Based Algorithm Selection + algorithm scheduling: a set of 20 optimizers including CMA-ES, BFGS, ...			1077.06	3898
2	Danil Shkarupin					878.68	4407
4	avaneev	biteopt2017			link	744.583	4730
3	radka					817.503	4747
5	jpsbook					730.328	5093
6	Al Jimenez					578.137	5951
7	Poly Montreal	MADS + VNS + NM				509.345	6421
9	GERAD	PSD-MADS	Serial version of PSD-MADS, based on MADS and using the NOMAD software.	link	link	244.434	7665
10	mini-mlog	GAPSO	A swarm approach utilizing CPSO, SPSO, DE, Simplex and quasi-Newton methods	link	link	140.54	8320
11	Jeremy	Research algorithm				105.573	8591
8	Artelys					275.369	9915
12	LocalSolver					38.6977	10664
13	kadiri					10.7097	12314
14	djagodzi					6.53201	13656
15	...					3.63873	14834

Garść wyników: BBComp Expensive 10BJ

rank	participant	method name	method description	software	paper	score ↑	sum of ranks ↓
2	avaneev	biteopt2017			link	793.263	4144
1	Nbelkhir	gaussian processes + Local search	Robust optimization for finding initial search point + multiple local search			815.239	4966
3	Artelys					685.836	5631
4	LTM@TUK & Moving@UNIFI					509.01	5745
6	Jeremy	Research algorithm				413.758	5827
8	mini-mlog	GAPSO		link	link	383.728	6052
7	Al Jimenez					405.932	6171
5	LocalSolver	LocalSolver 8.0	LocalSolverBlackbox 8.0 with default parameters			426.768	6873
9	anonymous					240.166	6972
10	GERAD	PSD-MADS	Serial version of PSD-MADS, based on MADS and using the NOMAD software.	link	link	218.277	8007
11	jpsbook					134.86	8161
13	Simon Wessing	ALGSS	Approximately latinized generalized stratified sampling	link	link	28.588	11117
12	kadiri					41.4288	11330

Garść wyników: BBComp Expensive 10BJ 64D



Literatura I

Daniel Bratton and James Kennedy. Defining a Standard for Particle Swarm Optimization. (Sis):120–127, 2007.

Kyle Robert Harrison, Andries P. Engelbrecht, and Beatrice M. Ombuki-Berman. An adaptive particle swarm optimization algorithm based on optimal parameter regions. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, nov 2017. ISBN 978-1-5386-2726-6. doi: 10.1109/SSCI.2017.8285342. URL <http://ieeexplore.ieee.org/document/8285342/>.

James Kennedy and Russell C. Eberhart. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks. IV*, pages 1942–1948, 1995.

Literatura II

- Michał Okulewicz. Finding an Optimal Team. In *Position Papers of the 2016 Federated Conference on Computer Science and Information Systems*, pages 205–210, oct 2016. doi: 10.15439/2016F465. URL <https://fedcsis.org/proceedings/2016/pliks/465.pdf><https://fedcsis.org/proceedings/2016/drp/465.html>.
- Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, 1987. ISSN 0097-8930. doi: 10.1145/37402.37406. URL <http://doi.acm.org/10.1145/37402.37406>.
- Mudita Sharma, Manuel López-Ibáñez, and Dimitar Kazakov. Performance Assessment of Recursive Probability Matching for Adaptive Operator Selection in Differential Evolution. pages 321–333. Springer, Cham, sep 2018. doi: 10.1007/978-3-319-99259-4_26. URL http://link.springer.com/10.1007/978-3-319-99259-4_{_}26.

Literatura III

- Yuhui Shi and Russell C. Eberhart. Parameter selection in particle swarm optimization. *Proceedings of Evolutionary Programming VII (EP98)*, pages 591–600, 1998. doi: 10.1007/BFb0040810. URL <http://link.springer.com/10.1007/BFb0040810>.
- Kenneth Sörensen. Metaheuristics-the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015. ISSN 14753995. doi: 10.1111/itor.12001.
- Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. ISSN 0925–5001. doi: 10.1023/A:1008202821328. URL <http://dx.doi.org/10.1023/A{%}%3A1008202821328>.

Literatura IV

- Mateusz Uliński, Adam Żychowski, Michał Okulewicz, Mateusz Zaborski, and Hubert Kordulewski. Generalized Self-adapting Particle Swarm Optimization Algorithm. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3242, pages 29–40, 2018. doi: 10.1007/978-3-319-99253-2_3. URL http://link.springer.com/10.1007/978-3-319-99253-2_{_}3.
- Frans Van Den Bergh and Andries Petrus Engelbrecht. A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae*, 105(4):341–374, 2010. ISSN 01692968. doi: 10.3233/FI-2010-370.
- David H. Wolpert. The Lack of a Priori Distinctions between Learning Algorithms. *Neural Computation*, 8(7):1341–1390, 1996. ISSN 08997667. doi: 10.1162/neco.1996.8.7.1341.

Literatura V

David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): 67–82, 1997. ISSN 1089778X. doi: 10.1109/4235.585893.