Setup Of Fuzzy Hybrid Particle Swarms

A heuristic approach

Nicolas Roy
University of Namur
Namur, Belgium
nicolas.roy@unamur.be

Charlotte Beauthier
Cenaero
Gosselies, Belgium
charlotte.beauthier@cenaero.be

Timotéo Carletti
University of Namur
Namur, Belgium
timoteo.carletti@unamur.be

Alexandre Mayer
University of Namur
Namur, Belgium
alexandre.mayer@unamur.be

ABSTRACT

This paper presents a framework for systematically investigating and designing fuzzy rulesets for Adaptive Fuzzy Particle Swarm Optimization (AFPSO) algorithms. Training is achieved through Gaussian Process (GP) supported by Gradient Boosted Regression Trees (GBRT). Meta-objective was defined by ranks on various benchmark functions. Validation benchmarks were also performed on GECCO '20 bound-constrained optimization competition. The resulting variants, particularly those controlling hybridization with Quantum Particle Swarm Optimization (QPSO) surpassed classical Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Differential Evolution (DE) on the training functions. Some level of generalization was also observed on most of the validation set.

CCS CONCEPTS

• **Theory of computation** → **Continuous optimization**; *Algorithm design techniques*;

KEYWORDS

ACM proceedings, PSO, Fuzzy Control, Heuristic, GBRT

ACM Reference Format:
Nicolas Roy, Charlotte Beauthier, Timotéo Carletti, and Alexandre Mayer. 2021. Setup Of Fuzzy Hybrid Particle Swarms: A heuristic approach. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459418

1 MOTIVATION

Research shows that controlling the parameters during the optimization process can improve performance. For example, in PSO, AFPSO variants are proposed[5]. However, design of the controller remains manual. We propose a framework using meta heuristic optimization for the design of fuzzy rulesets. An automated setup allows effortless adaptation of an algorithm to a set of problems. We also pursue a reasonable level of generalization or robustness.

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459418

2 CONCEPTS AND TECHNIQUES

Our AFPSO algorithm includes a Fuzzy Inference Engine (FIE) probing three variables from the optimization process and turning them into various parameters. At a conceptual level, it integrates mostly like the objective function evaluation, as illustrated in Figure 1.

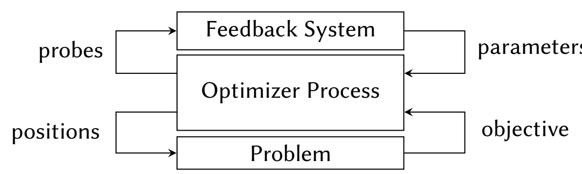

Figure 1: Fuzzy inference engine integration.

2.1 Particle Swarm Optimization

Our framework is a plug-in that could be used along any set of population heuristic with meta-parameters or existing AFPSO. We chose to work with the PSO paradigm given its popularity, robustness and simplicity. The original PSO was adapted from Craig Reynolds' boïds simulation by Kennedy and Russel Eberhart in 1995[1].

The swarm in PSO for bound-constrained, single objective minimization is a collection of S agents possessing a position in the search space, relationships with other agents and a cost. Particles also store in memory their best known position and its associated cost.

The update equations behind PSO are pretty straightforward: particles' positions are simply incremented by their speed at each iteration t.

The search mechanism lies in the definition of the speed in which the particle is attracted by two weighted *attractors* or known points: the agent's own best memory, with factor r_1c1 and the agent's best friend's best memory, with factor r_2c2. r_1 and r_2 are random variables sampled from a uniform distribution in $[0, 1]$, giving

$$v_i \mid_{t+1} = r_1 c_1 \left(m_i \mid_t - x_i \mid_t\right) + r_2 c_2 \left(m_{b_i} \mid_t - x_i \mid_t\right) + \omega v_i \mid_t \ .$$

While used separately, the two attractors allow for **exploitation**, locally or globally. Their interaction employs the information shared among the swarm to better **explore** the search space. The last term implements inertia from [2] by basing the current speed on a fraction ω of its previous value.

In the following, we refer to hybridization, which was achieved using QPSO[3] rather than PSO with probability h.

Table 1: Definition of noteworthy fuzzy variables

Name	Symbol	Low	Med.	High
Evals. Budget	τ	0.2	0.4	0.6
Proximity to b.f.	δ	0.3	0.5	0.8
Improvement	ϕ	-1.0	0.0	1.0
Inertia	ω	0.3	0.7	0.9
Hybridization	h	0.0	0.5	1.0
Lowest speed thr.	l	0.0	0.005	0.01

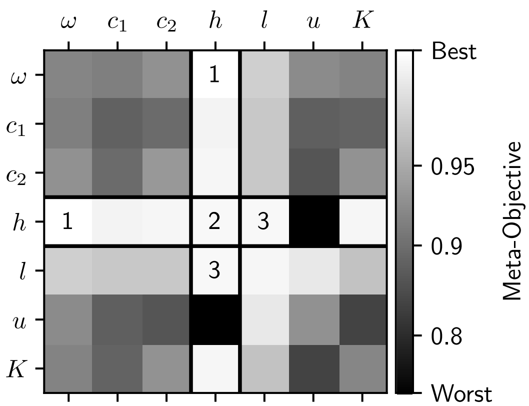

Figure 2: Efficiency of the different prototypes (pairs of controlled parameters). The meta-objective is represented using power-law normalization (γ = 8)

2.2 Fuzzy inference engine

A feedback engine design uses the variables described in Table 1, binding probes τ, δ, ϕ to parameters ω, h and l of PSO. Parameter l controls the minimal speed threshold. To preserve the simplicity and speed of PSO, we designed simple probes that are evaluated quickly. Initial definition was inspired by [5]. Probe τ measures the consumption of function evaluations and probe δ the distance from the agent to its best friend. Finally, probe ϕ measures the rank improvement of the agent among the swarm.

A GBRT assisted GP evolves the best design using 100 meta-objective evaluations for a given pair of parameters, the prototype and a benchmark. The meta-objective represents performance of an optimizer on 14 functions in 50 dimensions with 40e3 evaluations of the function shared by 40 agents. Functions are presented multiple times to the optimizer with transformations during training. Figure 2 indicates the three best combinations by their rank: ω&h(1), h(2) and h&l(3). Performance on those training functions is shown in Table 2 for the best prototypes of Figure 2 with an increased population size S. These rank show some robustness in the slightly different(S = 80) benchmark in Table 2.

3 RESULTS & CONCLUSIONS

We investigated the simultaneous control of two or more parameters. The performance for pairs of parameters are shown in Figure 2. Controlling the hybridization has a great and positive impact on performance.

We also assessed the generalization capability of the method on the GECCO'20 benchmark[4]. Results are briefly summarized in Table 3.

Table 2: Comparison between the mean results for our benchmark functions in 50 dimensions (S=80).

Func.	PSO	ω&h(1)	h(2)	h&l(3)
Sphere	5.28e−2	**0.00**	**0.00**	**0.00**
Ackley	6.48	**0.00**	**0.00**	2.28e−3
Rastr.	1.45e2	**0.00**	2.00	7.32
Rosen.	9.24e1	**4.59e1**	4.70e1	4.70e1
Stibl.	2.68e2	1.64e2	**1.02e2**	3.71e2
Schwef.	1.02e4	**8.93e3**	1.29e4	9.71e3
Chung	1.07	**0.00**	**0.00**	8.24e−7
Griew.	7.24e8	**0.00**	**0.00**	1.59e−5
Qing	4.98e7	3.68e3	3.59e3	**6.18e2**
Salom.	5.90e−1	**3.54e−1**	5.36e−1	5.91e−1
Hap. Cat	1.44	**9.99e−2**	1.20e−1	1.80e−1
Xin-Sh1	1.43e10	**7.94e9**	9.87e9	1.15e10
Xin-Sh2	**0.00**	**0.00**	**0.00**	**0.00**
Bnt.Cig.	9.86e9	**0.00**	**0.00**	1.38e3

Table 3: Comparison between the mean results on the GECCO'20 benchmark functions in 20 dimensions (S=80).

Func.	PSO	ω&h(1)	h(2)	h&l(3)
0	4.67e6	**0.00**	**0.00**	2.59e2
1	1.64e3	**3.35e2**	9.06e2	2.22e3
2	4.20e1	3.68e1	**2.63e1**	3.75e1
3	2.69	1.59	**1.33**	1.76
4	3.75e2	3.77e2	**5.45**	5.75e1
5	2.77e1	6.85	**3.69**	2.64e1
6	9.93	2.27e2	**2.85**	3.73e1
7	1.14e2	1.03e2	1.02e2	**1.00e2**
8	4.07e2	4.14e2	**4.06e2**	4.09e2
9	4.50e2	4.26e2	4.14e2	**4.14e2**

In this setup, the algorithm was far from its training conditions in terms of number of variables, agents and function evaluations but still managed to provide overall improvement to PSO, particularly the prototype based on the single control parameter h.

REFERENCES

[1] James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4. IEEE, 1942–1948.
[2] Yuhui Shi and Russell Eberhart. 1998. A modified particle swarm optimizer. In *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*. IEEE, 69–73.
[3] Jun Sun, Bin Feng, and Wenbo Xu. 2004. Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*, Vol. 1. IEEE, 325–331.
[4] CT. Yue. 2020. Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. (2020).
[5] Yuhui Shi and R. C. Eberhart. 2001. Fuzzy adaptive particle swarm optimization. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Vol. 1. 101–106 vol. 1.