Comparative Performance and Scaling of the Pareto Improving Particle Swarm Optimization Algorithm

Stephyn G. W. Butcher
Johns Hopkins University
Baltimore, MD
steve.butcher@jhu.edu

John W. Sheppard
Montana State University
Bozeman, MT
john.sheppard@montana.edu

Brian K. Haberman
Johns Hopkins University - APL
Laurel, MD
bkhabs@cs.jhu.edu

ABSTRACT

The Pareto Improving Particle Swarm Optimization algorithm (PI-PSO) has been shown to perform better than Global Best PSO on a variety of benchmark problems. However, these experiments used benchmark problems with a single dimension, namely $32d$. Here we compare global best PSO and PI-PSO on benchmark problems of varying dimensions and with varying numbers of particles. The experiments show that PI-PSO generally achieves better performance than PSO as the number of dimensions increases. PI-PSO also outperforms PSO on problems with the same dimension but with the same *or fewer* particles.

KEYWORDS

Particle Swarm Optimization, Curse of Dimensionality, Hitchhiking, Pareto Improvement

ACM Reference Format:
Stephyn G. W. Butcher, John W. Sheppard, and Brian K. Haberman. 2018. Comparative Performance and Scaling of the Pareto Improving Particle Swarm Optimization Algorithm. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205721

1 INTRODUCTION

Potter and de Jong developed the Cooperative Coevolutionary Genetic Algorithm (CCGA) to combat hitchhiking in the GA by harnessing cooperative subspecies that solved different parts of the problem [3] . Van den Bergh and Engelbrecht adapted the CCGA to Particle Swarm Optimization (PSO) and devised the Cooperative PSO (CPSO) [6]. However, they noticed, as did Potter and de Jong, that although problem partitioning and cooperation attenuated hitchhiking, it introduced the possibility of *pseudo-optima* in problems with *epistasis* or variables with highly correlated values. Strasser *et al.* [5] re-introduced competition in the form of overlapping populations in FEA to solve both hitchhiking and combat pseudo-optima. As a family of algorithms, FEA has the added benefit of being able to use GA, PSO, and other optimizers.

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205721

Butcher *et al.* [1] re-evaluated this line of research and proposed that the main reason these "cooperative" algorithms succeeded was they implicitly harnessed a Pareto improving criterion for resolving conflicts in information exchange. Because of the way those algorithms reassembled partial solutions, they avoided hitchhiking in the full solution. By applying the same approach to the selection of the global best in PSO, Butcher *et al.* created a single population algorithm that accomplished the same result: Pareto Improving Particle Swarm Optimization (PI-PSO). The experiments comparing PI-PSO and PSO, however, were based on benchmark problems of a single dimensionality of $32d$ with a fixed number of candidates.

In contrast, in this paper, we illustrate the comparable performance and scaling of PI-PSO in two facets. First, by looking at problems with different dimensionalities instead of just $d = 32$ as in the original research, we investigate where PI-PSO falls relative to PSO in terms of the dimensionality's impact on search performance. Additionally, following [2], we used the same number of candidate solutions for both PSO and PI-PSO, 10 per dimension. We also evaluate the performance of PI-PSO and PSO across the various dimensions with different numbers of particles.

2 EXPERIMENTS

In Butcher *et al.* [1] we used 19 standard benchmark minimization problems chosen for their scalability. However, we only looked a problems with 32 dimensions and 10 particles per dimension (*ppd*). In those experiments, PI-PSO outperformed PSO on 16 of the 19 benchmark problems. However, in the face of the No Free Lunch Theorem [7] and the curse of dimensionality [4] the performance and scaling characteristics of PI-PSO relative to PSO are far from clear. In order to address this, we performed additional experiments the same 19 benchmark functions with 4, 8, 16, and 32 dimensions. We used 2, 4, 8, 16, 32, 64, 128, and 256 ppd for the two algorithms.

2.1 Results

We had three basic hypotheses for our experiments. The overall results are specified in bold.

(1) **Hypothesis 1** – Because of the way PI-PSO constructs the global best, PI-PSO will out perform PSO as the number of particles per dimension increase, given a dimensionality. **Held for 16 of 19 benchmarks**.
(2) **Hypothesis 2** – Because PI-PSO can extract more information from the population, it will be able to perform

Figure 1: Ackley-1 Benchmark Results

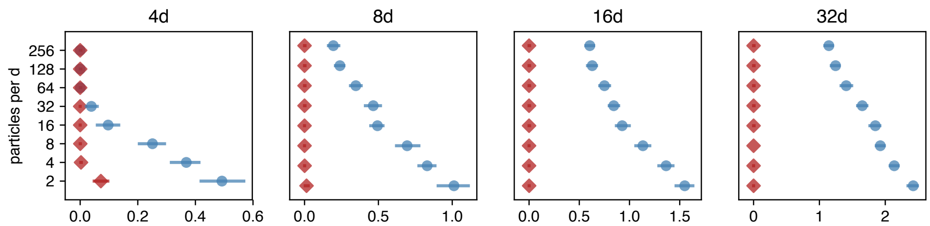
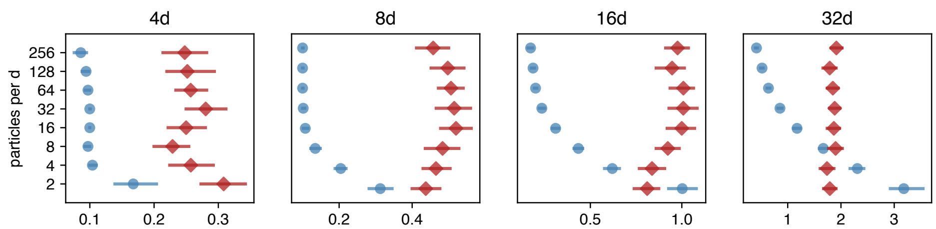


Figure 2: Salomon Benchmark Results



better than PSO on a problem of 32 dimensions with fewer particles. **Held for 15 of 19 benchmarks**.

(3) **Hypothesis 3** – Because we expect hitchhiking to be less of a problem on lower dimension problems, PI-PSO's advantage over PSO may be small on lower dimension problems and then increase as the dimensions increase. **Held for 12 of 19 benchmarks**.

Hypothesis 2 was important because the general approach is to keep candidate solutions the same for every algorithm [2]. However, we had concerns in our original work that PI-PSO had many more fitness evaluations than PSO. Our experiments are able to do both by showing the performance for each algorithm over a large range of particles per dimension for a given problem. In our experiments, 4 ppd for PI-PSO is approximately the same number of fitness evaluations as 128 ppd for PSO.

We show the specific findings for the Ackley-1 benchmark function in Figure 1, which were typical of the experiments supporting our hypotheses. The red/diamonds are PI-PSO; blue/dots, PSO. The lines are 95% confidence intervals. In contrast, the results for the Salomon benchmark function shown in Figure 2 are in part typical of those functions or parameterizations which did not support one or more hypotheses.

2.2 Discussion

Looking across all the problems and dimensions, there is no single parameterization that is always better than another. With 19 benchmarks and four possible dimensions, there are 76 sets of results. PI-PSO has the same results or better than PSO in 52 of them (68.4%) using 2 particles per dimension. However, this does not represent the best performance achievable by PI-PSO. In most cases, increasing particles per dimension, increases performance–for both PI-PSO and PSO. For PI-PSO, the problem lies in the cases where they do not.

3 CONCLUSION

We set out to explore the comparative performance and scalability of the Pareto Improving Particle Swarm Optimization (PI-PSO) algorithm against the standard *gbest* PSO. Although in most of the experiments, PI-PSO performed better than PSO there were some notable exceptions that suggest PI-PSO may be overly greedy in some cases. Future experiments will investigate the possibility of non-Pareto improving information exchanges.

REFERENCES

[1] Stephyn G. W. Butcher, John W. Sheppard, and Shane Strasser. 2018. Pareto Improving Selection of the Global Best in Particle Swarm Optimization. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. forthcoming.
[2] AP Engelbrecht. 2014. Fitness Function Evaluations: A Fair Stopping Condition?. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*. 1–8.
[3] Mitchell A Potter and Kenneth A De Jong. 1994. A cooperative coevolutionary approach to function optimization. In *Parallel Problem Solving from Nature—PPSN III*. Springer, 249–257.
[4] Francisco J Solis and Roger J-B Wets. 1981. Minimization by random search techniques. *Mathematics of operations research* 6, 1 (1981), 19–30.
[5] Shane Strasser, Nathan Fortier, John Sheppard, and Rollie Goodman. 2017. Factored Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* 21, 3 (2017), 281–293.
[6] Frans Van Den Bergh and Andries P Engelbrecht. 2001. Training product unit networks using cooperative particle swarm optimisers. In *Proceedings of International Joint Conference on Neural Networks*, Vol. 1. IEEE, 126–131.
[7] David H Wolpert and William G Macready. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 1 (1997), 67–82.