PSO with Randomized Low-Discrepancy Sequences

Nguyen Xuan Hoai
Natural Computation Group
Military Technical Academy,
Hanoi, Vietnam
hoainx@lqdtu.edu.vn

Nguyen Quang Uy
Natural Computation Group
Military Technical Academy,
Hanoi, Vietnam
quanguyhn@yahoo.com

R.I. McKay
Structural Complexity Laboratory,
Seoul National University,
South Korea
rim@cse.snu.ac.kr

ABSTRACT
We initialize a global-best particle swarm with a Halton sequence, comparing it with uniform initialization on a range of benchmark function optimization problems. We see substantial improvements in performance, particularly with high complexity problems /small populations. Halton initialization yields equivalent performance to uniform initialization with substantially smaller populations.

Categories and Subject Descriptors
I.2.8 [**Computing Methodologies**]: Problem Solving, Control Methods and Search – *heuristic methods.*

General Terms: Experimentation

Keywords
Randomized Halton Sequence, Particle Swarm Optimization.

1. INTRODUCTION
Several studies in evolutionary computation (EC) have shown that different pseudo-random generators can have significant effects on performance [1-3]. However they still assume that uniform random initialization is appropriate. In Monte Carlo simulations, randomized low discrepancy sequences [4,5] have been found more effective. We trialed a randomized Halton sequence [6] for particle swarm initialization (PSO, [7]).

2. EXPERIMENTS AND RESULTS
We implemented the glbest algorithm [7], with the initial swarm values generated in two ways: uniform pseudo-random (U-PSO) with the random number generator of Press et al [8], and a randomized Halton (SH-PSO) sequence. Otherwise, the two algorithms are identical, both using Press' generator for later stages. The benchmark functions used were Spherical (**f_1**), Quadratic (**f_2**), Ackley (**f_3**), Griewank (**f_4**), Rastrigin (**f_5**), and Rosenbrock (**f_6**) [2].

We used a fixed budget, 20000 function evaluations, and swarm sizes 50, 100, and 200. We varied problem dimensionality over 10, 15, 20, 25, 30, 35, and 40. For each combination, we ran 100 runs and recorded mean best performance.. Due to space limits we only show the results (Table 1) with swarm size 50, and dimensionalities 10, 20, 30, and 40.

3. CONCLUSIONS AND FUTURE WORK
Halton sampling gives better results than uniform, particularly on problems of high dimensionality, with limited swarm size. Halton gives better exploration, equivalent to an increase in population. We plan to investigate this further, through diversity and local fitness landscape studies, in subsequent work.

Acknowledgements
This work was partly funded through a national research grant for fundamental sciences, grant number: 203106.

GECCO'07, July 7–11, 2007, London, England, United Kingdom
ACM 978-1-59593-697-4/07/0007.

REFERENCES
[1] Daida, J, Ampy, D, Ratanasavetavadhana, M, Li, H, & Chaudhri, O Challenges with Verification, Repeatability, and Meaningful Comparison in Genetic Programming: Gibson's Magic. In *Proceedings of GECCO 1999*, 1851-1858, 1999

[2] Kimura, S & Matsumura, K Genetic Algorithms using Low-Discrepancy Sequences In *Proc GECCO 2005,* 1341-1346, 2005.

[3] Meysenburg, M.M. & Foster, J.A. Random Generator Quality and GP Performance. In *Proceedings GECCO 1999*, 1121-1126, 1999.

[4] Gentle, E.J. *Random Number Generation and Monte Carlo Methods*, Springer-Verlag, 1998.

[5] Morokoff,W.J. & Caflisch,R.E. . Quasi-random sequences and their discrepancies. *SIAM J Sci Comp*, 15(6): 1251-1279, 1994.

[6] Wang, X. & Hickernell, F.J. Randomized Halton Sequences, *Mathematical and Computer Modelling*, 32: 887-899, 2000.

[7] Engelbrecht, A.P. Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, 2005.

[8] Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, chapter 7, 2002.

Table 1. Average Best Solution Values
Number of particles = 50 (400 generations)

	Dim.	SH-PSO	U-PSO
f_1	10	0	0
	20	0.21±0.31	0.24±0.36
	30	250±214	316±224
	40	$(5.0\pm2.2)*10^3$	$(7.6\pm3.6)*10^3$
f_2	10	0.003±0.007	0.003±0.005
	20	$(7.2\pm3.9)*10^3$	$(8.0\pm4.8)*10^3$
	30	$(1.4\pm0.4)*10^5$	$(1.7\pm0.6)*10^5$
	40	$(6.8\pm2.2)*10^5$	$(7.0\pm2.2)*10^5$
f_3	10	$(1.1\pm1.8)*10^{-6}$	$(1.0\pm1.3)*10^{-6}$
	20	0.33±0.39	0.29±0.33
	30	5.5±1.3	5.8±1.3
	40	12.4±1.6	13.7±2.0
f_4	10	0.079±0.037	0.080±0.043
	20	0.23±0.15	0.23±0.19
	30	1.56±0.48	1.71±0.50
	40	12.2±4.9\	18.0±8.1
f_5	10	3.3±2.1	3.5±2.4
	20	33.2±9.5	34.5±10.9
	30	131±34	137±37
	40	336±78	397±73
f_6	10	8.2±7.9	10.23±9.8
	20	32.4±18.0	38.4±22.3
	30	194±159	333±401
	40	$(8.3\pm1.6)*10^3$	$(10.8\pm9.5)*10^3$