Space-based Initialization Strategy for Particle Swarm Optimization

Liang Yin, Xiao-Min Hu (Corresponding Author) and Jun Zhang
Department of Computer Science, Sun Yat-sen University
Key Laboratory of Machine Intelligence and Sensor Networks, Ministry of Education
Key Laboratory of Software Technology, Education Department of Guangdong Province, P.R. China
xmhu@ieee.org

ABSTRACT
Particle Swarm Optimization (PSO) is a population-based stochastic optimization algorithm that has been applied to various scientific and engineering problems. Despite its fast convergence speed, the original PSO is easy to fall into local optima when solving multi-modal functions. To address this problem, we present a novel initialization strategy, namely Space-based Initialization Strategy (SIS), to help PSO avoid local optima. We embed SIS into the standard PSO and form a novel PSO variant named SIS-PSO. The performance of SIS-PSO is validated by 13 benchmark functions and the experimental results demonstrate that the SIS enables PSO to achieve faster convergence speed and higher solution accuracy especially in multi-modal problems.

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

General Terms
Algorithms, Performance, Experimentation.

Keywords
Multi-modal function, History information, Sub-region, SIS-PSO, Space-based initialization strategy.

1. INTRODUCTION
Particle swarm optimization (PSO) has been successfully applied to various scientific and engineering problems [1-4] for it performs well in solving unimodal problems. However, it is easy to fall into local optima due to the lack of strategies to jump out of local optima. There exist approaches for improving PSO such as via adopting time-varying parameters controlling strategies [5], combining PSO with other algorithms [6], or change the operations of PSO [7].

This paper studies the initialization strategy for PSO and proposes a novel initialization strategy, i.e., Space-based Initialization Strategy (SIS). SIS contains two primary operations: sampling operation and dividing operation, respectively. In each generation of the SIS, the sampling operation collects a fixed number of samples and the dividing operation finds out the best sample for further space division. By doing this, the range of large regions will diminish exponentially. As a result, the solution accuracy is improved. We embed SIS into the standard PSO and form a novel PSO variant named SIS-PSO.

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

2. MAIN PROCEDURE OF SPACE-BASED INITIALIZATION STRATEGY
Step 1) *Initialization*: Divide each dimension of the search space, S^i, into two equal sub-regions s_1^i and s_2^i. The boundary of s_1^i and s_2^i is $[l^i,(l^i+u^i)/2]$ and $[(l^i+u^i)/2, u^i]$. Each sub-region is associated with a probability *pr*, which is initialized as 0.5. The initial population *X* is empty.

Step 2) *Sampling*: Use the roulette wheel strategy to sample *p* solutions from the search space. In detail, each dimension ep^i of a sampled solution $EP = (ep^1,ep^2,\ldots,ep^n)$ is sampled by two methods, the possibility of each method is *Pg* and (1 - *Pg*). Detailed information is showed as equation (1):

$$ep^i = \begin{cases} random(l^i,u^i); & if\ \ \alpha < Pg; \\ random(l_k^i,u_k^i); & in\ other\ cases; \end{cases} \tag{1}$$

where α is a random number in (0,1) and $[l_k^i,u_k^i]$ is the boundary of sub-region s_k^i selected by the roulette wheel strategy.

Step 3) *Further division*: Suppose the best solution among the sample is denoted as *BP* $(bp^1,bp^2,\ldots,bp^n)$. For each sub-region s_j^i that bp^i belongs to, further split the sub-region into two and associate each new sub-region with a probability *pn*. The probabilities of the other existing sub-regions are all equal to $(1-2\times pn)/(c-1)$, where *c* represents each dimension of the search space S^i that has been divided into *c* sub-regions as predefined in this paper.

Step 4) *Population update*: If the number of solutions in the initial population is smaller than the predefined population size *PS*, insert the best solution *BP* into the initial population *X*. Otherwise, replace the worst solution in the initial population with *BP* in the case that *BP* is better than the worst solution in the initial population *X*.

Step 5) *Termination check*: The termination criterion is defined as the maximum number of fitness evaluations that can be used by SIS. If the termination criterion is satisfied, procedure of SIS is completed and *X* is returned to PSO as its initial population. Otherwise, return to step 2 for another iteration of SIS.

TABLE II: The Result of GPSO, CLPSO and SIS-PSO, in SIS-PSO, the SIS Run 60000 FEs and PSO Run 140000 FEs

Func	*FEs*	*GPSO*		*CLPSO*		*SIS-PSO*	
		Mean	*Stddev*	*Mean*	*Stddev*	*Mean*	*Stddev*
1	200000	1.55e-026	1.41e-051	2.62e-009	1.12e-009	**6.28e-042**	**8.93e-082**
2	200000	2.46e-018	1.47e-035	1.96e-006	4.88e-007	**1.35e-027**	**4.38e-054**
3	200000	3.37e+002	1.55e+006	1.04e+003	2.27e+002	**9.42e-002**	**4.40e-003**
4	200000	6.56e-001	1.09e-001	1.13e+001	1.26	**5.06e-002**	**6.49e-004**
5	200000	3.26e+003	2.60e+008	**1.81e+001**	**1.15e+001**	4.08e+001	1.63e+003
6	200000	0	0	0	0	**0**	**0**
7	200000	8.95e-003	1.38e-005	6.40e-003	1.62e-003	**2.05e-003**	**5.04e-007**
8	200000	-9818.	2.61e+005	-12569.5	3.83e-005	**-12569.5**	**1.09e-023**
9	200000	2.75e+001	1.02e+002	5.27e-003	3.74e-003	**0**	**0**
10	200000	3.01e-014	9.14e-028	1.67e-005	3.19e-006	**1.11e-014**	**3.77e-030**
11	200000	1.98e-002	8.80e-004	**7.04e-007**	**5.02e-007**	5.57e-002	1.80e-003
12	200000	6.91e-003	6.68e-004	1.10e-010	5.87e-011	**1.68e-032**	**4.63e-066**
13	200000	1.09e-003	1.08e-005	2.20e-009	9.48e-010	**7.75e-032**	**1.21e-062**

3. EXPERIMENT AND ANALYSIS

In the experiment, 13 benchmark functions are used for validating the performance of SIS-PSO. Among the functions, functions 1-7 are unimodal functions and functions 8-13 are multi-modal functions [8].

This paper also compares SIS-PSO with GSO and CLPSO. The parameters are set as TABLE I:

TABLE I: Parameter of GPSO, CLPSO and SIS-PSO:

Algorithm	*pg*	*p*	*pn*	*PS*	*w*	*c1*	*c2*	*v*
GPSO	—	—	—	50	0.9-0.4	2	2	0.2
CLPSO	—	—	—	50	0.9-0.4	2	2	0.2
SIS-PSO	0.975	100	0.49	50	0.9-0.3	2	2	0.05
SIS-PSO	0.975	100	0.49	50	0.9-0.3	2	2	0.05

where *pg* is the mechanism selection possibility of SIS, *p* is the sampling size of the SIS, *pn* is the possibilities of most recently obtained sub-regions, *PS* is the population size of PSO, *w* is the inertia weight of PSO, *v* is the maximum velocity of particles in PSO. Because the initial particles of SIS-PSO are likely surrounding the global optimum, the inertia weight and the maximum velocity in SIS-PSO are thus smaller than those in CLPSO and GPSO for achieving faster convergence rate and getting better solutions.

The result accuracy of SIS-PSO, GPSO and CLPSO is showed in TABLE II. In all functions, the convergence speed of SIS-PSO is much faster than CLPSO and GPSO. There are two reasons attributing to this phenomenon: one reason is that the SIS binary divided sub-regions contain great solutions, and focus on searching these sub-regions to exponentially increase the convergence rate; the other reason is that the inertia weight w and the maximum value of velocity v are decreased. In other words, SIS-PSO can increase the convergence rate by adjusting the parameters of PSO with regardless of the influence of local optima. Though in $f5$ and $f11$ the other algorithms perform better than SIS-PSO, they are worse than SIS-PSO in other 11 functions.

4. CONCLUSION

This paper presents a Space-based Initialization Strategy (SIS), which can gather the solutions around the global optimum with relatively low computational cost. This strategy is applied when initializing the PSO for assisting PSO to avoid local optima. Thirteen functions have been utilized to validate the performance of SIS-PSO and the paper makes a comparison with the GPSO and CLPSO. The results show that the SIS is efficient, especially for solving multi-modal functions.

5. ACKNOWLEDGEMENT

This work was supported in part by the National High-Technology Research and Development Program ("863" Program) of China under Grand No. 2013AA01A212, by the National Science Fund for Distinguished Young Scholars under Grant 61125205, by the National Natural Science Foundation of China under Grant 61070004, 61202130, by the NSFC Joint Fund with Guangdong under Key Project U1201258 and U1135005, Guangdong Natural Science Foundation S2012040007948, Fundamental Research Funds for the Central Universities No. 12lgpy47, Specialized Research Fund for the Doctoral Program of Higher Education 20120171120027.

REFERENCES

[1] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Netw.*, Perth, Australia, 1995, vol. 4, pp. 1942–1948.

[2] Wei-Neng Chen, Jun Zhang, H.S.H. Chung, Wen-Liang Zhong, Wei-gang Wu, Yu-Hui Shi, A novel set-based particle swarm optimization method for discrete optimization problems, *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 2, pp. 278-300, 2010.

[3] Yue-Jiao Gong, Jun Zhang, H.S.H. Chung, Wei-Neng Chen, Zhi-Hui Zhan, Yun Li, Yu-Hui Shi. An efficient resource allocation scheme using particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp.801-816, 2012.

[4] Yue-Jiao Gong, Jun Zhang, Ou Liu, Rui-Zhang Huang, H.S.H. Chung, Yu-Hui Shi. Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach. *IEEE Transactions on Systems, Man, and Cybernetics: Part C*, vol. 42, no. 2, pp. 254-267, 2012.

[5] Zhi-Hui Zhan, Jun Zhang, Yun Li, H.S.H. Chung. Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 39, no. 6, pp.1362-1381, 2009.

[6] J.J. Liang and P.N. Suganthan. Dynamic multi-swarm particle swarm optimizer with local search. In *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 522–528.

[7] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, vol.10, no.3, pp. 281- 295, 2006.

[8] Xin Yao, Yong Liu, Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, 1999.