2011 International Symposium on Nonlinear Theory and its Applications
NOLTA2011, Kobe, Japan, September 4-7, 2011

Discrete Particle Swarm Optimizers for Numerical Analysis of Dynamical Systems

Toshimichi Saito† and Masafumi Kubota†

†EE Dept., Hosei University, Tokyo 184–8584, Japan

ABSTRACT

This paper studies application of discrete particle swarm optimizers (DPSO) to multi-solution problems (MSP) in discrete dynamical systems. The algorithm consists of two stages: the global search for constructing local sub-regions each of which includes one target solution and the local search that operates in parallel in the local subspaces to find all the target solutions. Performing basic numerical experiment, the algorithm efficiency is confirmed.

The particle swarm optimizer (PSO) is an population-based optimization algorithm inspired by flocking behavior of living beings. The particles represent potential solutions and moves to find the target solutions according to experience of its own and neighbors' history [1]. The PSO is simple in concept, is easy to implement and is applicable to various systems including artificial neural networks [2]-[5]. This paper studies a DPSO for application to MSP [6]-[8] in discrete dynamical systems [9]. The DSPO operates in a discrete-valued search space consisting of lattice points in order to optimize a discrete objective function. The DPSO has several advantages in reliable operation, reproducible results, robust hardware implementation and so on [1] [10]. The MSP is inevitable in practical/potential applications [6]-[8]. The target of our DPSO is a set of fixed points corresponding to the minima of the objective function. The DSPO does not try to find the exact optimum solutions but to find desired approximate solutions precisely and speedily. The algorithm consists of two stages. The first stage is global search in a discrete search space of rough lattice points. Applying the particles with ring topology, the DPSO tries to find the local subspaces (LSSs) each of which has one approximate solution that satisfies a criterion. The second stage is local search for finding the target solution in each LSS having finer lattice points than that in the global search. The algorithm operates in parallel in all the LSSs and tires to find all the approximate solutions that satisfies a criterion.

Let us consider the DPSO to find multiple fixed points in the following 2-dimensional map:

$$\begin{aligned} x_1(n+1) &= x_2(n) + a x_1(n) + b\tan^{-1} x_1(n) \equiv f(x_1, x_2) \\ x_2(n+1) &= -c x_1(n) \equiv g(x_1, x_2) \end{aligned} \tag{1}$$

where n is the discrete time. Such maps are important study objects and search of fixed points is basic problem in analysis of nonlinear dynamical systems. For simplicity, we fix the parameters $a = 0.55$, $b = 4$ and $c = 0.9$. In this case the map has three fixed points: $\boldsymbol{x}_s^1 = (-3.91284, 3.52156)$ (Sol1, stable); $\boldsymbol{x}_s^2 = -\boldsymbol{x}_s^1$ (Sol2, stable); $\boldsymbol{x}_s^3 = (0, 0)$ (Sol3, unstable). Adding magnitudes of f and g, we obtain the positive definite function F whose minimum value 0 gives the target fixed points as shown in Fig. 1 (a):

$$\begin{aligned} &F(x_1, x_2) = |f(x_1, x_2) - x_1| + |g(x_1, x_2) - x_2| \\ &S_0 = \{x|\ |x_1| \le 5,\ |x_2| \le 5\ \} \end{aligned} \tag{2}$$

where S_0 is the search space. It should be noted that our object is not finding the fixed points.

In the global search, the search space S_0 is discretized into $m_1 \times m_1$ lattice points. Let t_1 denote a search step. At $t_1 = 1$, we assign N_1 particles of ring topology on S_0 as shown in Fig. 1(b). Let $\boldsymbol{x}_i$ be the i-th particle positions and let $\boldsymbol{v}_i$ be the i-th particle velocity. They are updated:

$$\begin{aligned} \boldsymbol{v}_i(t_1+1) &= \omega \boldsymbol{v}_i(t_1) + r_1(\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i(t_1)) \\ &\qquad + r_2(\boldsymbol{x}_{lbest_i} - \boldsymbol{x}_i(t_1)) \\ \boldsymbol{x}_i(t_1+1) &= \boldsymbol{x}_i(t_1) + \boldsymbol{v}_i(t_1+1) \end{aligned} \tag{3}$$

where $i = 1 \sim N$. r_1 and r_2 are random values in $[0, \gamma_1]$ and $[0, \gamma_2]$, respectively. $\boldsymbol{x}_{pbest_i}$ is the i-th personal best such that $F(\boldsymbol{x}_{pbest_i})$ is the minimum for $\boldsymbol{x}_i$ in the past history. $\boldsymbol{x}_{lbest_i}$ is the i-th local best that is the best of the personal bests in the neighbor of the i-th particle. After this update, the position $\boldsymbol{x}$ is converted to the closest lattice points in S_0. The personal bast is updated:

$$\boldsymbol{x}_{pbesti} = \boldsymbol{x}_i(t_1) \text{ if } F(\boldsymbol{x}_i(t_1)) < F(\boldsymbol{x}_{pbest_i}) \tag{4}$$

Following the personal bests, the local bests are also updated. If $F(\boldsymbol{x}_i) < C_1$ for some i then $\boldsymbol{x}_i$ is declared as a first approximate solution (AS1, see Fig. 1 (c)) where C_1 is the first criterion. The position bmx_i is declared as a tabu lattice point and is prohibited to revisit. $\boldsymbol{x}_i$ is reset to a lattice point (Fig. 1(c)). $\boldsymbol{v}_i$, $pbest_i$ and $lbest_i$ are all reset. We repeat such procedure up to the the maximum time step t_{m1}.

In order to make the LSSs each of which includes the target solution, we select top K of the AS1s. We then construct K candidates of LSSs consisting of $m_l \times m_l$ lattice points in square shape cantered at each AS1. If two or more squares overlap then the subset centered at the smallest AS1 is survived and the other subsets are removed. If we obtain more

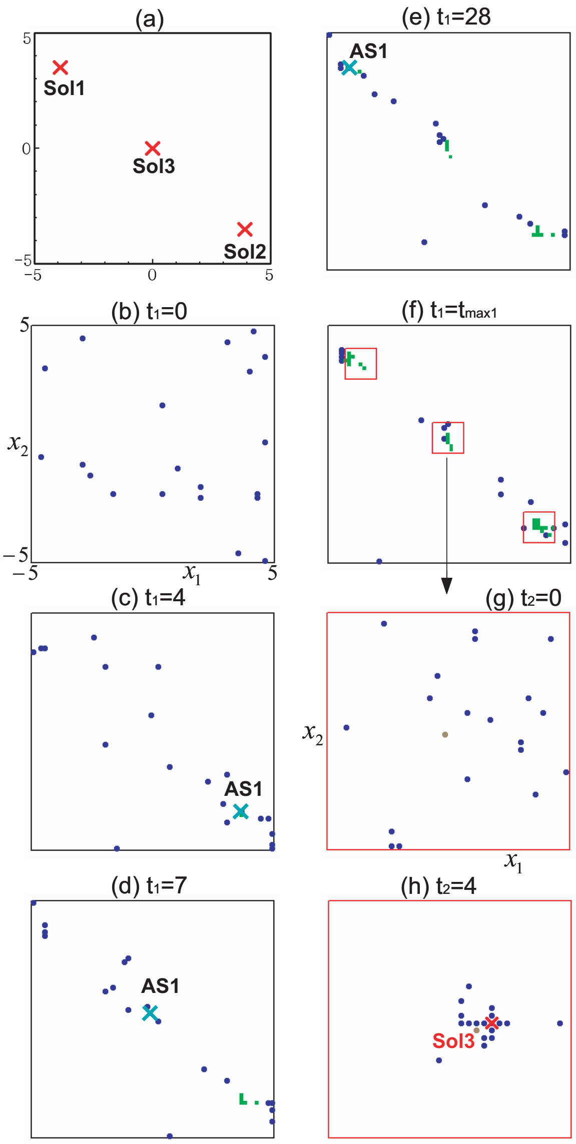

Figure 1: Particles movement of F for $N_1 = N_2 = 20$, $m_1 = 64$, $m_l = 8$, $m_2 = 32$, $\omega = 0.7$, $\gamma_1 = \gamma_2 = 1.4$, $C_1 = 0.5$, $K = 30$, $C_2 = 0.04$ and $t_{m1} = t_{m2} = 50$. (a) three fixed points (solutions, the red crosses) of the 2D map, (b) initialization for global search, (c) to (e) global search process, (f) three local subspaces (LSSs). (g) initialization for local search, (h) the approximate solution

than three survived subsets, we select subsets centered at top 3 of AS1s. The three subsets are the LSSs as shown in Fig. 1 (f). If the three subsets include the three solutions then the global search is said to be successful. In the local search, each LSS is re-discretized into $m_2 \times m_2$ lattice points The DSP operates on each LSS in parallel where N_2 particles of complete graph are assigned on the LSS. If some particle satisfies $F(\boldsymbol{x}) < C_2$ then algorithm is terminated successfully, where C_2 is the criterion of the approximate solution. Otherwise, we repeat such procedure up to the the maximum time step t_{m2}.

Performing basic numerical experiments as shown in Fig, 1, we have confirmed that the both global search and local search can realize over 90% success rate: if the parameters are selected suitably, the DPSO can operate almost successfully. Basically, there exists a trade-off between the success rate and computation cost and the DPSO can reduce the computation cost depending on the resolution. It should be noted that this is a first step to develop a novel DPSO for MSP. Future problems are many, including the following: role of key parameters, Analysis of search process, detailed performance evaluation, and application to bifurcation analysis.

References

[1] A. P. Engelbrecht, Fundamentals of computational swarm intelligence, Willey, 2005.

[2] M. P. Wachowiak, R. Smolikova, Y. uZheng, J. M. Zurada and A. S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3. pp. 289-301, 2004

[3] F. Teixeira and A. Romariz, Digital filter arbitrary magnitude and phase approximations-statistical analysis applied to a stochastic-based optimization approach, in Proc. IEEE Congress Evol. Comput., pp. 4089-4096, 2008.

[4] A. B. van Wyk and A. P. Engelbrecht, Overfitting by PSO trained feedforward neural networks, in Proc. IEEE Congress Evol. Comput., pp. 2672 - 2679, 2010.

[5] K. Kawamura and T. Saito, Design of switching circuits based on particle swarm optimizer and Hybrid Fitness Function, in Proc. 36th Annu. Conf. IEEE Ind. Electron. Soc., pp. 1099-1103, 2010.

[6] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3, pp. 211-224, 2004.

[7] D. Parrott and X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Trans. Evol. Comput., 10, 4. pp. 440-458, 2006

[8] S. Yang and C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, IEEE Trans. Evol. Comput., 14, 6. pp. 959-974, 2010

[9] E. Ott, Chaos in Dynamical Systems, Cambridge Univ. Press, 1993.

[10] Z. Sevkli and F. E. Sevilgen, Discrete Particle Swarm Optimization for the orienteering problem, in Proc. IEEE Congress Evol. Comput., pp. 1937-1944, 2010.