Solving Engineering Design Problems by Social Cognitive Optimization

Xiao-Feng Xie and Wen-Jun Zhang

Institute of Microelectronics, Tsinghua University, 100084 Beijing, China
xiexf@ieee.org, zwj@tsinghua.edu.cn

Swarm systems are products of natural evolution. The complex collective behavior can emerge from a society of N autonomous cognitive entities [2], called as agents [5]. Each agent acquires knowledge in socially biased individual learning [4]. For human, the extrasomatic arbitrary symbols that manipulated by language allows for cognition on a grand scale [3], since agent can acquire social information that is no longer limited to direct observation to other agents. The individual learning then only plays secondary role due to the ubiquity and efficiency of social learning [1].

Social cognitive optimization (SCO) is based on human social cognition, which operates on a *symbolic space* (S) that encodes the problem. Each $\vec{x} \in S$ is a *knowledge point* that evaluated by the *goodness function* $F(\vec{x})$. The foundational entity of SCO is social cognitive (SC) agent, which includes a memory (M_D) and a set of action rules (R_A). Specially, the agent acquires *social sharing information* only from an external medium called *library* ($\underline{L}$), which stores N_L points. The cognition is realized by interplaying between learning and memory, which the M_D and the $\underline{L}$ are employed for storing knowledge for guiding future actions; and R_A are executed for acquiring memory.

Each SC agent is worked in iterated learning cycles. Suppose T is the number of maximum learning cycles. At the tth ($1 \le t \le T, t \in \mathbb{Z}$) learning cycle, its M_D stores the most recently point $\vec{x}^{(t)}$. The action rules (R_A) include: a) Selects a better point $\vec{\kappa}_B$ from $\underline{L}$ by a tournament size τ_B; b) Determines the model point X_M and the refer point X_R: if $F(\vec{\kappa}_B) \le F(\vec{x}^{(t)})$, then $X_M = \vec{\kappa}_B$, $X_R = \vec{x}^{(t)}$, else $X_R = \vec{\kappa}_B$, $X_M = \vec{x}^{(t)}$; c) Infers a new point $\vec{x}^{(t+1)}$: For the dth dimension: $x_d^{(t+1)} = U_{\mathbb{R}}(X_{R,d}, X_{B,d})$ is a random value between $X_{R,d}$ and $X_{B,d}$. Normally, $X_{B,d} = 2 \cdot X_{M,d} - X_{R,d}$ for the dth dimension. Besides, if it is required that $x_d^{(t+1)} \in [l_d, u_d]$, then the $X_{B,d}$ is replaced by both $\max(X_{B,d}, l_d)$ and $\min(X_{B,d}, u_d)$; d) Stores the $\vec{x}^{(t+1)}$ into M_D, and replaces a worse point $\vec{\kappa}_W$, which is selected from $\underline{L}$ by a tournament size τ_W, by its old $\vec{x}^{(t)}$. The default values of τ_B and τ_W are 2 and 4, respectively.

Fig 1 shows two SC models: a) Full sharing model (FSM): all agents shares with a common library ($\underline{L}$), which the evaluation times are $T_E = N_L + (1+T) \cdot N$; b) Partial sharing model (PSM): the *blackboard* ($\underline{B}$) serves as a central data repository, which the size is N_B. Each agent has own library ($\underline{L}$) and allows a specified *thinking time* (T_T). When $t = n_t \cdot T_T + 1 \quad (0 \le n_t \le T / T_T, n_t \in \mathbb{Z})$, each agent updates its $\underline{L}$ by selecting N_L points from $\underline{B}$ at random. Then each agent works with its M_D and $\underline{L}$ as $n_t \cdot T_T + 1 \le t \le (n_t + 1) \cdot T_T$. After the $t = (n_t + 1) \cdot T_T$ cycle is finished, each agent only updates the point with best goodness in its $\underline{L}$ into $\underline{B}$. For PSM, $T_E = N_B + T \cdot N$.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 261–262, 2004.

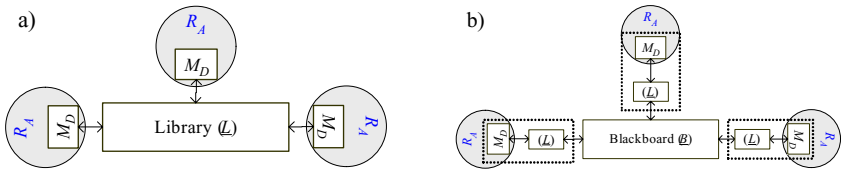

Fig. 1. SC models: a) full sharing model (FSM); b) partial sharing model (PSM)

Four engineering design problems [5] have been tested. They are speed reducer (*SR*), three-bar truss (*TB*), welded beam (*WB*), and tension spring (*TS*) problems. Specially, the *TS* is mixed-integer-continuous (MIC) problem. Table 1 summaries the global optimum F^*, the recently published results and the corresponding T_E by Ray et al. [5], and the results by SC models as T_E=20200, include FSM#1 (N=1, T=2E4, N_L=199), FSM#2 (N=40, T=500, N_L=160) and PSM (N=40, T=500, N_B=200, N_L=39, T_T=15) , which 500 runs were performed for each problem.

Table 1. Comparison of results between existing results [5] and SC models

F.	F^*	Ray et al. (T_E) [5]	FSM#1	FSM#2	PSM
SR	2994.471	2998.027 (110235)	2994.471	2994.471	2995.407
TB	263.8958	263.8989 (36113)	263.8972	263.8970	263.8963
WB	2.38113	2.96070 (64862)	2.47514	2.42723	2.39042
TS	0.012666	0.012923 (25167)	0.01365	0.01344	0.01283

It shows that even only one agent can get high quality solutions on three problems. Moreover, the models with N>1 are performed better than N=1 for most testing cases. For the MIC problem *TS*, the PSM shows better performance than FSM, which reduces the probability of the premature convergence by preventing some intermediate knowledge points from guiding the actions. At last, the PSM performs better than the published algorithm [5] for all problems in fewer evaluation times.

The SCO have few parameters that can be readily adjusting. The N can be set flexibly, even down to 1. Moreover, the trade-off between exploitation and exploration can be achieved by adjusting the size of library (N_L) while no significant impact on the T_E, which is mainly determined by the N and T.

References

1. Bandura, A.: Social Learning Theory. Prentice Hall, NJ (1977)
2. Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. Proc. Natl. Acad. Sci. USA, 99 (2002) 7280–7287
3. Flinn, M.V.: Culture and the evolution of social learning. Evolution and Human Behavior, 18 (1997) 23-67
4. Galef, B.G.: Why behaviour patterns that animals learn socially are locally adaptive. Animal Behaviour, 49 (1995) 1325-1334
5. Newell, A.: The knowledge level. Artificial Intelligence, 18 (1982) 87-127
6. Ray, T., Liew, K.M.: Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans. Evolutionary Computation, 7 (2003) 386-396