



OPEN

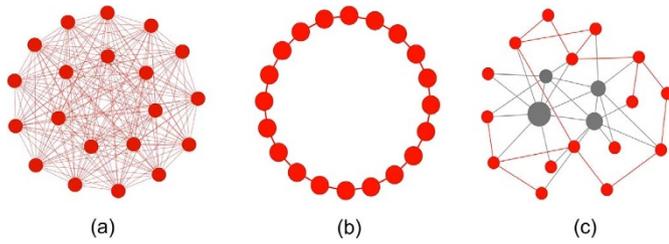
# Selectively-informed particle swarm optimization

SUBJECT AREAS:  
COMPLEX NETWORKS  
COMPUTER SCIENCEYang Gao<sup>1</sup>, Wenbo Du<sup>1</sup> & Gang Yan<sup>2</sup><sup>1</sup>School of Electronic and Information Engineering, Beihang University, Beijing 100191, People's Republic of China, <sup>2</sup>Center for Complex Network Research and Department of Physics, Northeastern University, Boston, MA 02115 USA.Received  
19 November 2014Accepted  
20 February 2015Published  
19 March 2015Correspondence and  
requests for materials  
should be addressed to  
W.D. (wenbodu@  
buaa.edu.cn)

Particle swarm optimization (PSO) is a nature-inspired algorithm that has shown outstanding performance in solving many realistic problems. In the original PSO and most of its variants all particles are treated equally, overlooking the impact of structural heterogeneity on individual behavior. Here we employ complex networks to represent the population structure of swarms and propose a selectively-informed PSO (SIPSO), in which the particles choose different learning strategies based on their connections: a densely-connected hub particle gets full information from all of its neighbors while a non-hub particle with few connections can only follow a single yet best-performed neighbor. Extensive numerical experiments on widely-used benchmark functions show that our SIPSO algorithm remarkably outperforms the PSO and its existing variants in success rate, solution quality, and convergence speed. We also explore the evolution process from a microscopic point of view, leading to the discovery of different roles that the particles play in optimization. The hub particles guide the optimization process towards correct directions while the non-hub particles maintain the necessary population diversity, resulting in the optimum overall performance of SIPSO. These findings deepen our understanding of swarm intelligence and may shed light on the underlying mechanism of information exchange in natural swarm and flocking behaviors.

Optimization<sup>1–3</sup> aims to seek the minimal or maximal point in the constrained parameter space of a system, which is highly challenging due to the increasing complexity of real problems we face in modern society. To solve real-world optimization problems researchers learned from the collective behaviors of social animals, yielding several intelligent algorithms<sup>4–6</sup>. Among those, particle swarm optimization (PSO), proposed by Kennedy and Eberhart<sup>5</sup>, is a typical swarm-intelligence algorithm that derives the inspiration from the self-organization and adaptation in flocking phenomena<sup>7–11</sup>.

In PSO, a flock of particles move in a constrained parameter space, interact with each other, and update their velocities and positions according to their own and their neighbors' experiences, searching for the global optimum. Owing to its simplicity, effectiveness and low computational cost, PSO has gained significant popularity and improvements. Most studies on improving the PSO fall into three categories. (1) Modifying the model coefficients. Shi and Eberhart introduced an inertia weight to reduce the restriction on velocity and better control the scope of search<sup>12</sup>. Later on, they employed fuzzy system and stochastic mechanism to better adapt the inertia weight<sup>13</sup>. Clerc and Kennedy introduced a constriction coefficient to ensure the convergence of the particles<sup>14</sup>. Trelea used dynamical system theory to analyze the PSO algorithm and derived the guidelines for choosing appropriate parameters<sup>15</sup>. Zhan *et al.* proposed an adaptive PSO in which model coefficients can vary according to evolutionary states<sup>16</sup>. (2) Considering the population structure. Kennedy showed that the sociometric structure and small-world manipulation interacted with function can produce a significant effect on performance<sup>17</sup>. Kennedy and Mendes examined the impact of topological structure more detailedly, leading to the identification of superior population configurations<sup>18</sup>. Liu *et al.* proposed the scale-free PSO (SFPSO) which employs degree-heterogeneous (scale-free) topologies and is able to significantly improve the optimization performance<sup>19</sup>. (3) Altering the interaction modes. Mendes *et al.* revised the way how each particle is influenced by its neighbors, resulting in the fully-informed PSO (FIPSO)<sup>20,21</sup> in which each particle learns from every individual in its neighborhood rather than the single best one. The performance of FIPSO is closely related to the population structure<sup>22</sup>. Liang *et al.* proposed the comprehensive learning PSO that allows each dimension of a particle to learn from different neighbors<sup>23</sup>. Li *et al.* proposed the adaptive learning PSO in which each particle can adaptively guide its behavior of exploration and exploitation<sup>24</sup>. They further proposed the self-learning PSO (SLPSO) that allows each particle to adaptively choose one of four learning strategies in different situations with respect to convergence, exploitation, exploration, and jumping out of the basins of attraction of local optima<sup>25</sup>.



**Figure 1 | Network structure representing the interactions between particles.** (a) A complete network with 20 nodes (particles). Each node connects to all others. (b) A ring network with 20 nodes. Each node links to its nearest two neighbors. (c) A scale-free network with 20 nodes, in which the node size represents the node degree, i.e. the number of edges associated with the node. It shows that most nodes have low degrees, yet there exist a few high-degree nodes (hubs).

However, most of the existing PSO algorithms treat all particles equally, prompting us to explore the impact of heterogeneous sight ranges: the hub particles (leaders) have a broad sight of the population; each non-hub particle (follower) has only a single source of information. The former would make the optimization process well guided by the leaders while the latter allows the followers to move without unnecessary interference. We found that our algorithm, selectively-informed PSO (SIPSO), taking into account the individuals' heterogeneity, can balance the exploration and the exploitation in the optimization process thus it achieves better performance.

In the following we will briefly introduce the PSO and its typical variants and then describe our SIPSO algorithm in detail.

**GPSO & LPSO.** For a minimum optimization problem with  $D$  independent variables and an objective function  $f(x)$ , the PSO algorithm represents the potential solutions with a flock of particles. Each particle  $i$  has a position  $x_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$  and a velocity  $v_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$  in the  $D$ -dimensional space. The goal is to find an optimal position  $x_i$  of any particle  $i$  that makes the objective function  $f(x)$  minimum. Initially the particles' positions and velocities

are generated randomly. Then, at each time step (iteration), each particle updates its position and velocity according to the following equations<sup>5</sup>:

$$v_i = \eta \cdot (v_i + U(0, c_1) \cdot (p_i - x_i) + U(0, c_2) \cdot (p_{n,i} - x_i)) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

where  $\eta = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}$ ,  $\varphi = c_1 + c_2 > 4$ . Here  $p_i$  is the best historical position found by particle  $i$ ,  $p_{n,i}$  is the best historical position found by  $i$ 's neighbors,  $c_1$  and  $c_2$  are the acceleration coefficients.  $U(a, b)$  is a random number drawn at each iteration from the uniform distribution  $[a, b]$ . Therefore,  $c_1$  and  $c_2$  balance the impacts of each particle's own and its neighbors' experiences, and  $\eta$  indicates the learning rate. Based on previous extensive analysis<sup>14</sup> we choose the appropriate settings as  $c_1 = c_2 = 2.05$  and  $\eta = 0.7298$ . Previous studies<sup>17,18,20-22</sup> have found that the interaction topology of particles has a great influence on final optimization results. Two versions of canonical PSO algorithm with different topologies are most commonly used: the GPSO with a fully connected network (Fig. 1(a)) and the LPSO with a ring (Fig. 1(b)). GPSO converges more rapidly than LPSO, yet, is more susceptible to be trapped at local optima<sup>17</sup>.

**FIPSO.** In the canonical PSO each particle is influenced by itself and the best-performed particle in its neighborhood. This "single-informed" strategy may ignore some important information from the remaining neighbors. Mendes *et. al.* hence proposed a "fully-informed" version of PSO (FIPSO)<sup>20,21</sup>, in which each particle adjusts its velocity according to the experiences of its all neighbors:

$$v_i = \eta \cdot \left( v_i + \frac{1}{k_i} \sum_{j \in \mathcal{N}(i)} U(0, \varphi) \cdot (p_j - x_i) \right) \quad (3)$$

where  $\mathcal{N}(i)$  is the node set of  $i$ 's neighbors,  $k_i$  is the number of  $i$ 's neighbors (i.e.,  $k_i$  is  $i$ 's degree and  $k_i = |\mathcal{N}(i)|$ ),  $p_j$  is the best historic

**Table 1 | Success rate**

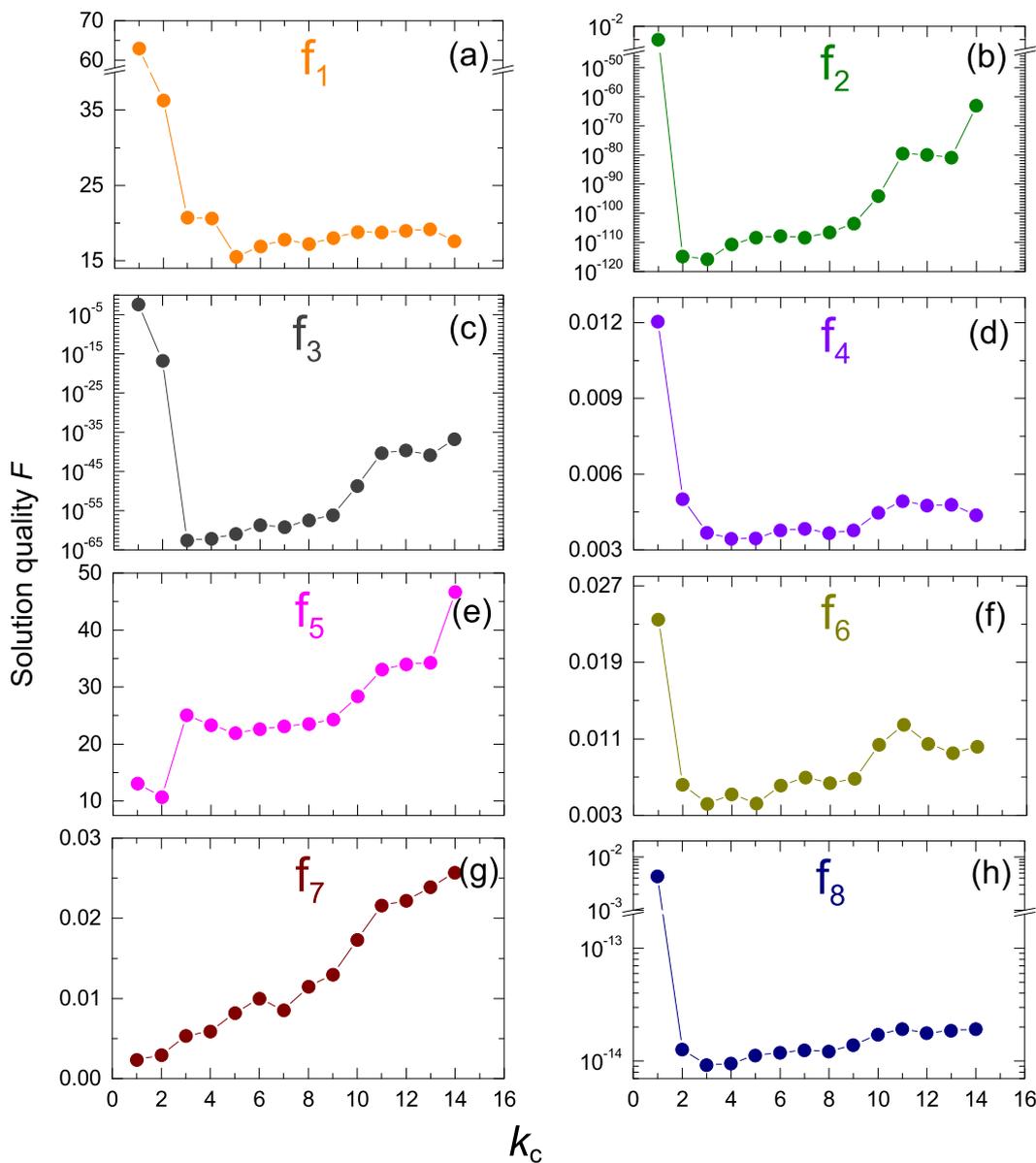
	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
GPSO	0.99	1.00	1.00	1.00	1.00	0.87	0.34	0.25
LPSO	0.92	1.00	1.00	1.00	1.00	1.00	0.71	1.00
SFPSO	0.96	1.00	1.00	1.00	1.00	0.99	0.66	0.98
GFIPSO	0	0	0	0.74	0.52	0	0.98	0
LFIPSO	0.99	1.00	1.00	1.00	0.88	1.00	1.00	1.00
SFIPSO	0.53	0.45	0.16	1.00	1.00	0.47	1.00	0.18
SLPSO	0.99	1.00	0.16	0.10	1.00	0.96	0.66	1.00
SIPSO	1.00 ( $k_c = 5$ )	1.00 ( $k_c = 3$ )	1.00 ( $k_c = 3$ )	1.00 ( $k_c = 4$ )	1.00 ( $k_c = 2$ )	1.00 ( $k_c = 3$ )	1.00 ( $k_c = 2$ )	0.99 ( $k_c = 3$ )
SIPSO ( $k_c^{fix} = 5$ )	1.00	1.00	1.00	1.00	1.00	0.99	0.96	0.99

**Table 2 | Solution quality**

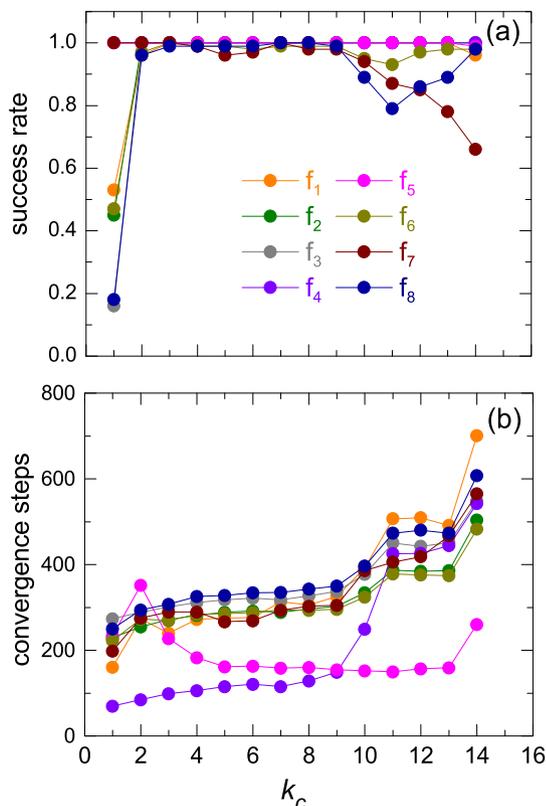
	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
GPSO	1.96e + 1	3.00e - 100	6.78e - 23	4.12e - 3	6.01e + 1	9.96e - 3	3.55e - 2	1.24e - 14
LPSO	2.31e + 1	8.60e - 47	7.90e - 27	9.46e - 3	5.65e + 1	5.56e - 3	2.80e - 2	<b>7.69e - 15</b>
SFPSO	1.76e + 1	8.61e - 64	1.99e - 37	4.37e - 3	4.67e + 1	1.02e - 2	2.57e - 2	1.93e - 14
GFIPSO	-	-	-	2.94e - 2	5.71e + 1	-	<b>1.41e - 3</b>	-
LFIPSO	2.59e + 1	3.7e - 11	7.03e - 7	5.60e - 3	8.32e + 1	<b>1.96e - 5</b>	8.87e - 3	2.96e - 6
SFIPSO	6.30e + 1	2.54e - 3	4.03e - 3	1.21e - 2	1.31e + 1	2.35e - 2	2.34e - 3	4.35e - 3
SLPSO	<b>2.28e - 2</b>	1.14e - 58	1.20e - 33	4.39e - 2	<b>0</b>	1.66e - 2	3.97e - 2	2.34e - 14
SIPSO	1.55e + 1	<b>1.64e - 116</b>	<b>2.86e - 63</b>	<b>3.45e - 3</b>	1.07e + 1	4.23e - 3	2.94e - 3	9.24e - 15
	( $k_c = 5$ )	( $k_c = 3$ )	( $k_c = 3$ )	( $k_c = 4$ )	( $k_c = 2$ )	( $k_c = 3$ )	( $k_c = 2$ )	( $k_c = 3$ )
SIPSO ( $k_c^{fix} = 5$ )	1.55e + 1	4.67 - 109	1.20e - 61	3.45e - 3	2.19e + 1	4.25e - 3	8.21e - 3	1.12e - 14



Table 3   Convergence speed		$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
GPSO		524.5	314.5	485.2	358.0	162.7	300.0	484.5	400.2
LPSO		856.4	660.0	764.8	1006.1	388.0	646.2	648.9	887.3
SFPSO		700.7	503.4	548.5	541.9	260.6	482.5	564.8	607.2
GFIPSO		-	-	-	<b>23.4</b>	<b>31.7</b>	-	<b>47.2</b>	-
LFIPSO		1985.1	2209.1	2594.9	1209.6	3367.2	2238.5	1773.0	2512.1
SFIPSO		<b>160.6</b>	<b>229.4</b>	<b>274</b>	70.0	234.7	<b>222.9</b>	198.1	<b>250.2</b>
SLPSO		854.2	1798.3	1851.7	4836.0	1762.1	1730.5	1146.1	1916.4
SIPSO		275.1 ( $k_c = 5$ )	271.2 ( $k_c = 3$ )	300.9 ( $k_c = 3$ )	105.6 ( $k_c = 4$ )	351.7 ( $k_c = 2$ )	269.2 ( $k_c = 3$ )	275.7 ( $k_c = 2$ )	307.8 ( $k_c = 3$ )
SIPSO ( $k_c^{fix} = 5$ )		275.1	288.5	317.3	115.0	161.9	285.7	266.6	327.6



**Figure 2 | The impact of  $k_c$  on the solution quality of the algorithm for eight benchmark functions.** In all sub-figures the vertical axes represent the solution quality for each function  $f$  by the average fitness of different runs  $F = \frac{1}{M} \sum_{j=1}^M [f(\mathbf{x}_k^j) - f(\mathbf{x}_{opt})]$ , where  $M$  is the number of independent runs,  $\mathbf{x}_k^j$  is the best solution found in  $j$ -th run, and  $\mathbf{x}_{opt}$  is the (known) optimum solution for the given function  $f$ . Therefore, the smaller the value of  $F$ , the better the performance of the algorithm.

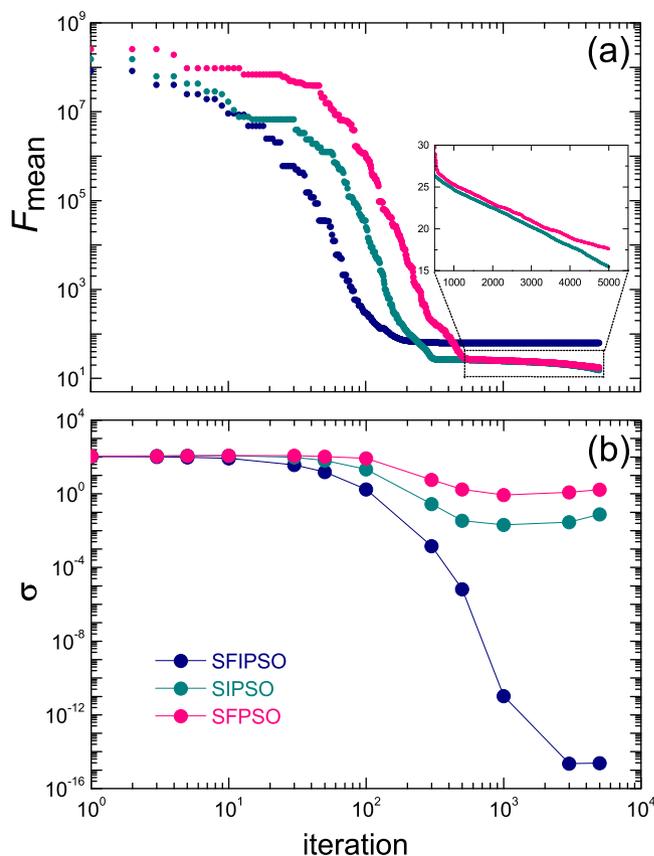


**Figure 3** | The impact of  $k_c$  on the success rate and convergence speed. (a) The success rate represented the proportion of successful runs. (b) The steps required to reach the goal value. The smaller the number of required steps, the higher the convergence speed.

position found by  $j$ . Studies<sup>21,22</sup> have revealed that, with appropriate parameter settings, the FIPSO can outperform the traditional PSO, but it is susceptible to the topology alteration. In some topologies the FIPSO may perform even worse than the canonical PSO.

**SFPSO & SFIPSO.** Recently, many natural and man-made networks have been found to exhibit scale-free property, *i.e.* the degree distribution is power-law<sup>26,27</sup>. Examples include neural networks<sup>28</sup>, citation networks<sup>29</sup>, World Wide Web<sup>30</sup>, Internet<sup>31</sup>, software engineering<sup>32</sup>, and on-line social networks<sup>33</sup>. In scale-free networks, only a few nodes are densely connected hubs and most nodes are low degree non-hub nodes, resulting in high heterogeneity of node's degrees (Fig. 1c). This discovery has triggered the interest of studying the impacts of underlying network structures on dynamical processes<sup>34–40</sup> and also of introducing scale-free topologies into evolutionary optimization algorithms<sup>19,41–43</sup>. In particular, Liu *et al.* investigated the influence of scale-free population structure on the performance of PSO<sup>19</sup>. Their results indicated that the scale-free PSO (SFPSO) outperforms the traditional GPSO and LPSO. In the following we also compare our algorithm to the fully-informed versions of SFPSO and GPSO (called SFIPSO and GFIPSO hereafter, respectively).

**SLPSO.** In most traditional PSO algorithms, a single learning mode is used for all particles, which may restrict the intelligence for a particular particle to deal with different situations. Li *et al.* proposed the self-learning PSO (SLPSO) that enables the particles to switch between four modes: exploitation, exploration, jumping out, and convergence<sup>25</sup>. Each mode has a set of operations to update the particles' velocity and position. A common strategy was introduced to allow each particle to adaptively choose the most suitable mode which depends on evolutionary stages and local



**Figure 4** | The mean fitness and the diversity of the swarm population during the optimization process. (a) The evolution of the mean fitness on the function  $f_1$ , *i.e.*,  $F_{mean} = \frac{1}{N} \sum_{j=1}^N [f_1(\mathbf{x}_i) - f_1(\mathbf{x}_{opt})]$  where  $N$  is the total number of particles,  $\mathbf{x}_i$  is the position of particle  $i$ , and  $\mathbf{x}_{opt} = \mathbf{1}$  is the optimum solution of  $f_1$ . The inset shows the last steps for SIPSO and SFPSO. (b) The evolution of the population diversity  $\sigma$  (see main text) during the optimization processes.

fitness landscape. Experimental comparisons showed that SLPSO outperforms several peer algorithms in terms of mean value, success rate and overall ranking, especially for some complex high-dimensional functions. Yet, three key parameters of SLPSO need to be chosen very carefully through a parameter tuning approach, as these parameters significantly affect the algorithm's performance. Note that in SLPSO, although each particle is able to switch between different modes, the learning strategy of choosing suitable modes is identical for all particles.

**Selectively-informed PSO.** The algorithms described above assumed that all particles are single-informed or fully informed, or adopt the same strategy for switching between different modes, overlooking the heterogeneity of individuals. Here we propose the selectively-informed PSO (SIPSO) algorithm that takes into consideration the heterogeneity of individuals' learning strategies. The population structure of our SIPSO is represented by a scale-free network (see **Methods**). And the learning strategy of each particle depends on its degree:

$$\mathbf{v}_i = \begin{cases} \eta \cdot \left( \mathbf{v}_i + \frac{1}{k_i} \sum_{j \in \mathcal{N}(i)} \mathbf{U}(0, \varphi) \cdot (\mathbf{p}_j - \mathbf{x}_i) \right), & \text{if } k_i > k_c \\ \eta \cdot \left( \mathbf{v}_i + \mathbf{U}(0, c_1) \cdot (\mathbf{p}_i - \mathbf{x}_i) + \mathbf{U}(0, c_2) \cdot (\mathbf{p}_{n,i} - \mathbf{x}_i) \right), & \text{if } k_i \leq k_c \end{cases} \quad (4)$$

where  $k_i$  is the degree of particle  $i$ ,  $k_c$  is the threshold to determine a particle fully- or single-informed. The densely-connected hubs



( $k > k_c$ ) are provided with more information to better lead the optimization process. The non-hub particles ( $k \leq k_c$ ) are less affected so that they can move in the search space with more freedom, maintaining the diversity of the population. Note that, when  $k_c = k_{min} - 1$ , all the particles are fully-informed thus the algorithm is degenerated to SFIPSO; when  $k_c = k_{max}$  all the particles take the canonical learning strategy, turning the algorithm to SFPSO. Here we are interested in the information selectivity, i.e.  $k_{min} - 1 < k_c < k_{max}$ . For example, in Fig. 1c, when  $k_c = 5$  the grey nodes (particles) with degree higher than 5 are fully-informed and the rest red nodes are single-informed.

## Results

**Overall performance.** We test the performance of our algorithm on eight widely-used benchmark functions  $f_{1-8}$  (see **Methods**) and compare it to other seven algorithms for three criteria: *success rate*, *solution quality*, and *convergence speed* (see **Methods**). Note that in SIPSO the optimal value of the degree threshold  $k_c$  varies for different test functions. We also show the results for a fixed threshold ( $k_c^{fix} = 5$ ) over all the functions.

Table 1 lists the comparison of success rate. Our algorithm SIPSO shows significant advantages, i.e., 99% on  $f_8$  and 100% on all the other functions. Even with a fixed threshold  $k_c^{fix} = 5$  the SIPSO also gets very satisfactory success rates.

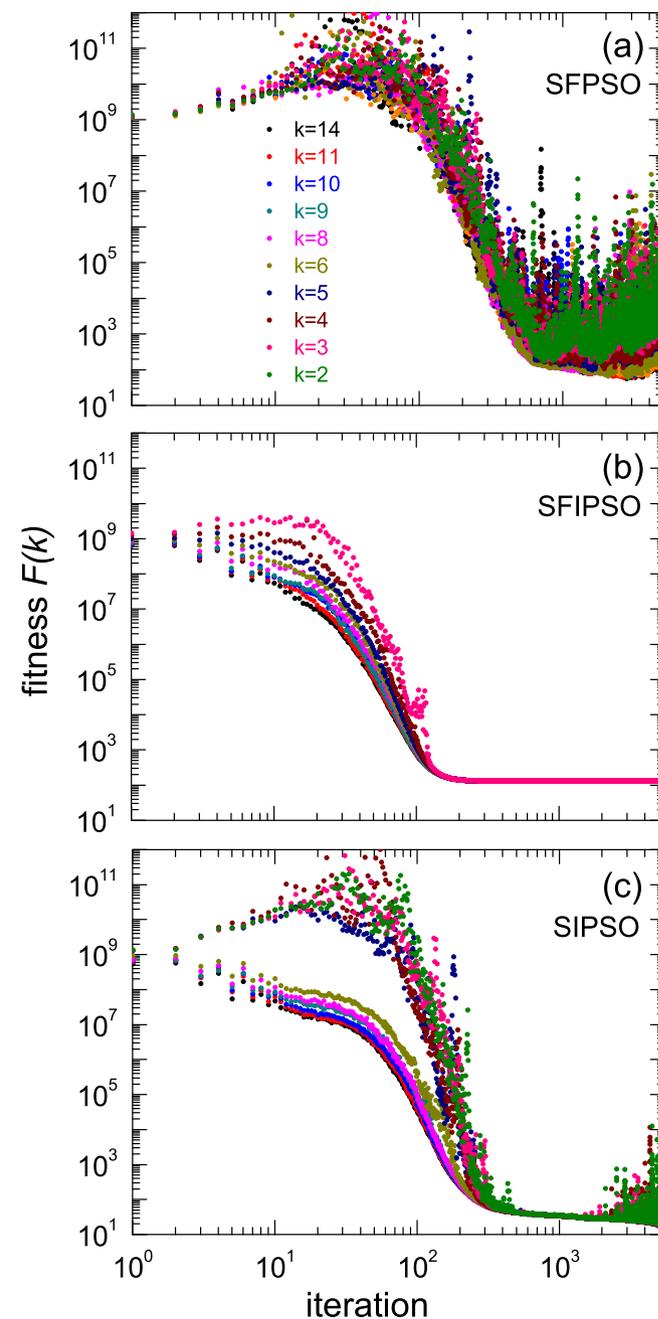
Table 2 lists the results in terms of solution quality. For each function, the best solutions are highlighted in bold and “–” means that the corresponding algorithm fails to reach the acceptable solution even once. For functions  $f_{2-4}$  our SIPSO remarkably outperforms the other algorithms, for  $f_1, f_5, f_6$  and  $f_8$  the SIPSO ranks 2<sup>nd</sup> of all the algorithms, while for  $f_7$  it ranks 3<sup>rd</sup>. When the degree threshold is fixed as  $k_c^{fix} = 5$ , the solution quality still ranks top 3 of all the algorithms over eight test functions.

Table 3 shows the convergence speed of each algorithm, represented by the steps required to reach the goal value. Thus the smaller the number of required steps, the higher the convergence speed. The best cases are marked in bold. Our SIPSO has a relatively fast convergence speed on all the functions, ranking 2<sup>nd</sup> on  $f_1, f_2, f_3, f_6$  and  $f_8$ , 3<sup>rd</sup> on  $f_4$  and  $f_7$ , 4<sup>th</sup> on  $f_5$ . SFIPSO has the fastest convergence speed on  $f_1, f_2, f_3, f_6$ , and  $f_8$ , and the GFIPSO converges fastest on  $f_4, f_5$  and  $f_7$ . It is worth noting that, faster convergence does not necessarily mean a better optimization trial. Actually, too fast convergence may lead to the problem of prematureness, i.e., being trapped at local optima. For example, as shown in Table 2 the solution qualities of SFIPSO and GFIPSO are really bad for most benchmark functions, although their convergence are very fast. In the fully-informed algorithms, each particle's information can be quickly transferred to all other individuals in the swarm thus the algorithms converge rapidly, resulting in prematureness. In contrast, in our SIPSO, only the hub particles are fully-informed and there are many non-hub particles taking the single-informed learning strategy to maintain the population diversity. Consequently, our SIPSO can achieve better performance with a satisfactory convergence speed.

**The impact of  $k_c$ .** As described above we find that for each function there is an optimal value of the threshold  $k_c$  with which our algorithm SIPSO performs best. Hence we investigate the impact of  $k_c$  on the performance for all eight benchmark functions. The results of solution quality, success rate and convergence speed are shown in Figs. 2 and 3. One can see that, for the solution quality on all functions except  $f_5$  and  $f_7$  SFPSO (the rightmost data point) outperforms SFIPSO (the leftmost data point), while for  $f_5$  and  $f_7$  it reverses. However, on all the functions except for  $f_7$ , neither SFIPSO nor SFPSO is able to obtain the best result. With  $k_c$  between  $k_{min}$  and  $k_{max}$  our algorithm SIPSO achieves the best performance (Fig. 2). Similar results for success rate are shown in Fig. 3(a). Our SIPSO has high success results on all functions with an appropriate  $k_c$ . As shown in

Fig. 3(b), increasing the number of fully-informed particles can significantly improve the convergence speed and our SIPSO has moderate speed of convergence.

**The microscopic point of view.** To uncover the underlying mechanism of our algorithm, we explore the optimization process from a microscopic point of view. We compare our SIPSO ( $k_{min} - 1 < k_c < k_{max}$ ) to SFIPSO ( $k_c = k_{min} - 1$ ) and SFPSO ( $k_c = k_{max}$ ) that are all on scale-free networks, excluding the influence of other factors. For the sake of simplicity, in the following we will present the results for the function  $f_1$ . The results for other functions are alike and not shown here.



**Figure 5 | The fitness of particles with different degrees during the optimization process.** Here  $F(k) = \frac{1}{N_k} \sum_{i=1}^N [f_1(\mathbf{x}_i) - f_1(\mathbf{x}_{opt})] \delta(k_i, k)$ , where  $N_k$  is the number of particles with degree  $k$ . And  $\delta(k_i, k) = 1$  if  $k_i = k$ , and 0 otherwise. (a) SFPSO, i.e.  $k_c = k_{max}$ ; (b) SFIPSO, i.e.  $k_c = k_{min} - 1$ . (c) SIPSO, i.e.  $k_{min} - 1 < k_c < k_{max}$ , here we show the evolution of  $F(k)$  for  $k_c = 5$ .



Table 4 | Benchmark functions

Name	Formula	D	Range	Optimum	Goal
Rosenbrock	$f_1(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	30	$[-30, 30]^p$	0	100
Sphere	$f_2(x) = \sum_{i=1}^D x_i^2$	30	$[-100, 100]^p$	0	0.01
Schwefel's P2.22	$f_3(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	30	$[-10, 10]^p$	0	0.01
QuadricNoise	$f_4(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0,1)$	30	$[-1.28, 1.28]^p$	0	0.05
Rastrigin	$f_5(x) = \sum_{i=1}^D x_i^2 - 10 \cos 2\pi x_i + 10$	30	$[-5.12, 5.12]^p$	0	100
Griewank	$f_6(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \frac{x_i}{\sqrt{i}} + 1$	30	$[-600, 600]^p$	0	0.05
Griewank	$f_7(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \frac{x_i}{\sqrt{i}} + 1$	10	$[-600, 600]^p$	0	0.05
Ackley	$f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^p$	0	0.01

First, we examine the mean fitness ( $F_{mean}$ ) of the swarm population during an optimization process, with the definition  $F_{mean} = \frac{1}{N} \sum_{i=1}^N [f_1(\mathbf{x}_i) - f_1(\mathbf{x}_{opt})]$  where  $N$  is the total number of particles,  $\mathbf{x}_i$  is the position of particle  $i$ , and  $\mathbf{x}_{opt} = \mathbf{1}$  is the optimum solution of  $f_1$ . As shown in Fig. 4(a) the SFIPSO has the fastest convergence as each particle uses full information from all of its neighbors, but it is trapped at some local optima in the early stage ( $\sim 150$  iterations). Despite their relatively low convergence SIPSO and SFPSO are able to achieve higher qualities of final solutions, and SIPSO is the best for the mean fitness.

Second, we compare the population diversity of SFPSO, SFIPSO and SIPSO, which indicates the extent of exploration during the searching process of the swarm. The population diversity is defined as<sup>45</sup>  $\sigma = \sqrt{\sum_{i=1}^N \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 / (N-1)}$  where  $N$  is the total number of particles, and  $\bar{\mathbf{x}} = \sum_{i=1}^N \mathbf{x}_i$  is the mean position (center) of the swarm. Thus, the larger the  $\sigma$ , more diverse is the swarm. And a very small  $\sigma$  means that all particles are aggregated together, diminishing the capability of exploration. As shown in Fig. 4(b), the diversity of SFIPSO decreases quickly to a very small value due to the information redundancy of the fully-informed learning. Consequently, SFIPSO is not able to escape once gets stuck at a local optimum. Both SFPSO and SIPSO have a high level of diversity during the optimization, which ensure the thorough search in the parameter space thus improve the probability of finding the global optimum.

Furthermore, we investigate the fitness of particles with different degrees, i.e.,  $F(k) = \frac{1}{N_k} \sum_{i=1}^N [f_1(\mathbf{x}_i) - f_1(\mathbf{x}_{opt})] \delta(k_i, k)$ , where  $N_k$  is the number of particles with degree  $k$ .  $\delta(k_i, k) = 1$  if  $k_i = k$ , and 0 otherwise. The particles in SFPSO have only one information source, which is very unstable during the optimization process. So the fluctuation of the particles' fitness in SFPSO are violent (Fig. 5(a)). In SFIPSO, all particles are fully-informed, making the algorithm converge fast but prematurely (Fig. 5(b)). Our SIPSO combines the advantages of the two algorithms. The fitness of hub particles monotonously decreases, indicating that the hubs play the role of guiding the swarm. On the contrary, the non-hub particles have oscillating fitness, maintaining the necessary diversity of the swarm (Fig. 5(c)). The two different roles of the particles in SIPSO result in the appropriate trade-off between the convergence speed and the population diversity.

## Discussion

Taking into account the heterogeneity of individuals behaviors in flocking we propose the Selectively-Informed Particle Swarm Optimization (SIPSO) algorithm. In SIPSO, the particles interact

with their neighbors and change the searching direction and speed by learning from the experiences of themselves and their neighbors. Each particle's learning strategy depends on its degree: the hubs are able to learn from all of their neighbors (fully-informed) while each non-hub particle learns from a single yet best-performed neighbor. Consequently, the hubs have bird's eye views of the swarm and can better lead the population; the non-hub particles are less influenced thus can search in the space with high freedom, maintaining the diversity of the population.

We test the performance of our SIPSO on eight benchmark functions. The results show that SIPSO has high success rate, high solution quality, and acceptable convergence speed. We examine the optimization process from a microscopic point of view and reveal that, indeed, there are two different roles that the particles play in the SIPSO. Moreover, our algorithm is able to balance the population diversity and the convergence speed during optimization processes, improving the overall performance in comparison with other seven algorithms.

It is worth noting that we do not introduce adaptation into our SIPSO algorithm, i.e., all parameters including  $k_c$  are set initially and do not change during the optimization process, but instead we discriminate the nodes with different degrees, in contrast to SLPSO which adopts adaptive strategies in search of the optimum. Despite the lack of adaptation, our SIPSO works very well in the benchmark test functions. This finding uncovers the importance of considering the individuals' heterogeneity in particle swarm optimization. Nevertheless, as shown in previous works (e.g., refs. 24, 25), adaptation can improve PSO's performance. It is fairly expected that adaptively tuning the value of  $k_c$  during the searching process could improve our SIPSO's performance, which deserves future pursuits.

## Methods

**Benchmark functions.** To make a comprehensive comparison to test the effectiveness of our algorithm we designed extensive experiments. We choose eight benchmark functions (Table 4) that have been widely used<sup>17,18,20,21,44</sup>. Functions  $f_1 - f_4$  are unimodal, which are relatively easy to solve. Functions  $f_5 - f_8$  are multi-modal with a large number of local optima so that the algorithm really suffers from being premature. Functions  $f_6$  and  $f_7$  are the same Griewank function with different dimensions. In fact,  $f_7$  is considered more difficult<sup>18</sup>. Column 2 shows the formula of the fitness function. Column 3 shows the dimension of the problem  $D$ . Column 4 gives the range that variables can take. In column 5 the optimum values of the problems are presented. Column 6 defines the goal value to judge whether a run (trial) is successful or not.

**Parameter settings.** The parameters of experiments are set as follows. The population size is 50. For each algorithm and each benchmark function, the experiment consists of 100 independent runs. The maximal iteration is 5000. For SFPSO, SFIPSO and SIPSO, the scale-free network has maximal degree 14 and minimal degree 2. We generate the scale-free networks by Barabási-Albert model<sup>46</sup>, which has two main mechanisms: growth and preferential attachment. Starting with  $m_0$  fully-connected nodes, at each time step we add a new node to the network and connect it to  $m$  existing



nodes ( $m < m_0$ ). The probability  $P_i$  that the new node is connected to an existing node  $i$  depends on  $i$ 's degree:  $P_i = \frac{k_i}{\sum_j k_j}$ , where  $j$  runs over all the existing nodes. Here we set the parameters  $m_0 = 4$  and  $m = 2$ .

**Criteria.** To compare the performance of different algorithms we use three criteria: solution quality, convergence speed, and success rate. The solution quality is the final fitness value at the end of 5000 iterations. The convergence speed is represented by the number of iterations required to reach the goal. Obviously, the larger the number of required iterations, the lower the convergence speed. The success rate is the fraction of successful runs. Both the solution quality and the convergence speed are average values over the successful runs.

- Holland, J. H. *Adaptation in natural and artificial system: An introductory analysis with applications to biology, control, and artificial intelligence*. (University of Michigan Press, Ann Arbor, 1975).
- Glover, F. & Laguna, M. *Tabu search* (Springer, USA, 1999).
- Van Laarhoven, P. J. & Aarts, E. H. *Simulated annealing*. (Springer, Netherlands, 1987).
- Dorigo, M., Maniezzo, V. & Colnari, A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **26**, 29–41 (1996).
- Kennedy, J. & Eberhart, R. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Netw.* **4**, 1942–1948 Perth, WA. (DOI:10.1109/ICNN.1995.488968) (1995).
- Karaboga, D. An idea based on honey bee swarm for numerical optimization. *Technical report, Erciyes University, Computer Engineering Department*. (2005) Available at: [http://mf.erciyes.edu.tr/abc/pub/tr06\\_2005.pdf](http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf). (Accessed: 19 December 2014).
- Heppner, F. & Grenander, U. A stochastic non-linear model for coordinated bird flocks. (ed. Krasner, S.) (AAAS Publications, 1990).
- Couzin, I. D., Krause, J., Franks, N. R. & Levin, S. A. Effective leadership and decision making in animal groups on the move. *Nature* **433**, 513–516 (2005).
- Conradt, L., Krause, J., Couzin, I. D. & Roper, T. J. “Leading according to need” in self-organizing groups. *Am. Nat.* **173**, 304–312 (2009).
- Nagy, M., Ákos, Z., Biro, D. & Vicsek, T. Hierarchical group dynamics in pigeon flocks. *Nature* **464**, 890–893 (2010).
- Vicsek, T. & Zafeiris, A. Collective motion. *Phys. Rep.* **517**, 71–140 (2012).
- Shi, Y. & Eberhart, R. A modified particle swarm optimizer. *IEEE World Congr. Comput. Intell., Anchorage, AK*. (DOI:10.1109/ICEC.1998.699146) (1998).
- Shi, Y. & Eberhart, R. Fuzzy adaptive particle swarm optimization. *CEC '01, Seoul, South Korea. IEEE Proc. Congr. Evol. Comput.* **1**, 101–106. (DOI:10.1109/CEC.2001.934377) (2001).
- Clerc, M. & Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **6**, 58–73 (2002).
- Trelea, I. C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inform Process Lett.* **85**, 317–325 (2003).
- Zhan, Z. H., Zhang, J., Li, Y. & Chung, H. H. Adaptive particle swarm optimization. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **39**, 1362–1381 (2009).
- Kennedy, J. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. *CEC '99, Washington, DC. IEEE Proc. Congr. Evol. Comput.* **3**, 1931–1938 (DOI:10.1109/CEC.1999.785509) (1999).
- Kennedy, J. & Mendes, R. Population structure and particle swarm performance. *CEC '02, Honolulu, Hawaii. IEEE Proc. Congr. Evol. Comput.* **2**, 1671–1676. (DOI:10.1109/CEC.2002.1004493) (2002).
- Liu, C., Du, W. B. & Wang, W. X. Particle swarm optimization with scale-free interactions. *PLoS ONE* **9**, e97822 (2014).
- Mendes, R., Kennedy, J. & Neves, J. Watch thy neighbor or how the swarm can learn from its environment. *IEEE Proc. Swarm Intell. Symp.* 88–94 (DOI:10.1109/SIS.2003.1202252) (2003).
- Mendes, R., Kennedy, J. & Neves, J. The fully informed particle swarm: simpler, maybe better. *IEEE Trans. Evol. Comput.* **8**, 204–210 (2004).
- Kennedy, J. & Mendes, R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.* **36**, 515–519 (2006).
- Liang, J. J., Qin, A. K., Suganthan, P. N. & Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **10**, 281–295 (2006).
- Li, C. H., & Yang, S. X. An Adaptive Learning Particle Swarm Optimizer for Function optimization. *CEC '09, Trondheim, Norway. IEEE Proc. Congr. Evol. Comput.* 381–388. (DOI:10.1109/CEC.2009.4982972) (2009).
- Li, C. H., Yang, S. X. & Trung, T. N. A Self-Learning Particle Swarm Optimizer for Global Optimization Problems. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **42**, 627–646 (2012).
- Albert, R. & Barabási, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47–97 (2002).
- Newman, M. E. J. The Structure and Function of Complex Networks. *SIAM Rev.* **45**, 167–256 (2003).
- Eguiluz, V. M., Chialvo, D. R., Cecchi, G. A., Baliki, M. & Apkarian, A. V. Scale-free brain functional networks. *Phys. Rev. Lett.* **94**, 018102 (2005).
- Redner, S. How popular is your paper? An empirical study of the citation distribution. *Eur. Phys. J. B* **4**, 131–134 (1998).
- Barabási, A.-L., Albert, R. & Jeong, H. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A* **281**, 69–77 (2000).
- Vázquez, A., Pastor-Satorras, R. & Vespignani, A. Large-scale topological and dynamical properties of the Internet. *Phys. Rev. E* **65**, 066130 (2002).
- Wen, L., Dromey, R. G. & Kirk, D. Software engineering and scale-free networks. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **39**, 845–854 (2009).
- Leskovec, J. & Horvitz, E. Planetary-Scale Views on an Instant-Messaging Network. *WWW '08, Beijing, China. ACM Proc. 17th Int. Conf. World Wide Web* 915–924. (DOI:10.1145/1367497.1367620) (2008).
- Barrat, A., Barthelemy, M. & Vespignani, A. *Dynamical processes on complex networks*. (Cambridge University Press, 2008).
- Song, C., Havlin, S. & Makse, H. A. Self-similarity of complex networks. *Nature* **433**, 392–395 (2005).
- Zhou, S. & Mondragón, R. J. The rich-club phenomenon in the Internet topology. *IEEE Commun. Lett.* **8**, 180–182 (2004).
- Boccaletti, S. *et al.* The structure and dynamics of multilayer networks. *Phys. Rep.* **544**, 1–122 (2014).
- Perc, M. & Szolnoki, A. Coevolutionary games - A mini review. *BioSystems* **99**, 109–125 (2010).
- Shen, H.-W., Cheng, X.-Q. & Fang, B.-X. Covariance, correlation matrix, and the multiscale community structure of networks. *Phys. Rev. E* **82**, 016114 (2010).
- Wu, Z.-X., Rong, Z. & Holme, P. Diversity of reproduction time scale promotes cooperation in spatial prisoner's dilemma games. *Phys. Rev. E* **80**, 036106 (2009).
- Gasparri, A., Panzari, S., Pascucci, F. & Ulivi, G. A spatially structured genetic algorithm over complex networks for mobile robot localisation. *Intelligent Service Robotics* **2**, 31–40 (2009).
- Giacobini, M., Preuss, M. & Tomassini, M. Effects of scale-free and small-world topologies on binary coded self-adaptive CEA. *EvoCOP '06, Budapest, Hungary. Lect. Notes Comput. SC.* **3906**, 86–98. (DOI:10.1007/11730095\_8) (2006).
- Kirley, M. & Stewart, R. An analysis of the effects of population structure on scalable multiobjective optimization problems. *GECCO '07, London, UK. ACM Proc. 9th Annu. Conf. of Genetic and Evol. Comput.* 845–852. (DOI:10.1145/1276958.1277124) (2007).
- Tang, K. *et al.* *Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Technical report, USTC, China.* (2007) Available at: <http://sci2s.ugr.es/programacion/workshop/Tech.Report.CEC2008.LSGO.pdf>. (Accessed: 19th November 2014).
- Deb, K. & Beyer, H. G. Self-adaptive genetic algorithms with simulated binary crossover. *Evol. Comput.* **9**, 197–221 (2001).
- Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).

## Acknowledgments

Y.G. and W.B.D. acknowledge the financial support from the National Natural Science Foundation of China (Grant Nos. 61201314 and 61221061). G. Y. acknowledges the financial support from NS-CTA sponsored by US Army Research Laboratory under Agreement No. W911NF-09-2-0053.

## Author contributions

Y.G., W.B.D. and G.Y. designed and performed the research, analyzed the results, and wrote the paper.

## Additional information

**Competing financial interests:** The authors declare no competing financial interests.

**How to cite this article:** Gao, Y., Du, W. & Yan, G. Selectively-informed particle swarm optimization. *Sci. Rep.* **5**, 9295; DOI:10.1038/srep09295 (2015).



This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder in order to reproduce the material. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>