Investigating Particle Swarm Optimisation Topologies for Edge Detection in Noisy Images

Mahdi Setayesh[1], Mengjie Zhang[1], and Mark Johnston[2]

[1] School of Engineering and Computer Science
{mahdi.setayesh,mengjie.zhang}@ecs.vuw.ac.nz
[2] School of Mathematics, Statistics and Operations Research
Victoria University of Wellington,
PO Box 600, Wellington, New Zealand
mark.johnston@msor.vuw.ac.nz

Abstract. This paper investigates the effects of applying different well-known static and dynamic neighbourhood topologies on the efficiency and effectiveness of a particle swarm optimisation-based edge detection algorithm. Our experiments show that the use of different topologies in a PSO-based edge detection algorithm does not have any significant effect on the accuracy of the algorithm for noisy images in most cases. That is in contrast to many reported results in the literature which claim that the selection of the neighbourhood topology affects the robustness of the algorithm to premature convergence and its accuracy. However, the fully connected topology in which all particles are connected to each other and exchange information performs more efficiently than other topologies in the PSO-based based edge detector.

Keywords: particle swarm optimisation, edge detection, noisy images, neighbourhood topology.

1 Introduction

Edges as low level features in an image contain important information that are utilised in image analysis and computer vision systems. Many algorithms have been proposed to detect edges for different applications using various different paradigms such as curve fitting [4], optimization of a criterion [3], statistical testing [9] and soft computing [2] to detect edges. The selection of an edge detection algorithm for a particular application depends on its performance in variant environmental conditions (such as illumination and noise) and the requirements of the system of interest (such as real time ability, continuity of edges, thinness of edges and scale insensitivity).

PSO as a meta-heuristic method has been used to successfully solve global optimisation problems and was introduced by Kennedy and Eberhart in 1995 [7]. The main general advantages of PSO in comparison with other heuristic methods such as genetic algorithms, are ease of its implementation, fewer

D. Wang and M. Reynolds (Eds.): AI 2011, LNAI 7106, pp. 609–618, 2011.

operators, a limited memory for each particle and high speed of convergence [1]. As PSO has a high capability to optimise noisy functions [12], it has been successfully applied to many problems in noisy environments, such as image segmentation and vision tracking [20].

We previously applied two PSO-based algorithms with different encoding schemes and fitness functions to noisy binary images containing simple shapes, such as rectangles, squares, circles, crosses and triangles [16]. Their performances were acceptable in the binary images but they were inefficient and did not operate well on non-binary images. We revised our PSO-based algorithm through developing a new encoding scheme and fitness function and examined it on real images corrupted by two different types of noise (Gaussian and impulsive) [19]. The main idea in this algorithm was to maximise **interset** distance between the average pixel intensities of two regions separated by a continuous edge represented by a particle and minimise **intraset** distances within both regions. Our experiments showed that this version could outperform the Canny algorithm as a Gaussian filter-based algorithm especially in the images with high levels of noise. However, it produced jagged edges and its overall performance was worse than robust-rank order (RRO) algorithm as a statistical-based edge detection algorithm and was slower than the Canny and RRO algorithms. We changed the fitness function of our PSO-based algorithm through considering a larger area around each single pixel on a continuous edge than the previous version of our algorithm and introducing a curvature cost of a continuous edge to reduce the effect of producing jagged edges [19]. The experiments showed that the new revised version could detect edges more accurate, more continuous, smoother than the older version introduced in [17], the Canny and RRO algorithms. But, the new algorithm was still slower than Canny and RRO. We introduced a discrete constrained PSO-based algorithm with two constraints and used a penalising method to handle these constraints to detect edges [18]. Our experiments showed that the new algorithm is faster than the algorithm presented in [19] and there was no significant difference between the localisation accuracy of the algorithms. In all experiments, we have utilised the fully connected graph as a neighbourhood structure in our PSO-based edge detection algorithm and never investigated the influence of the chosen topology on the performance of the algorithm.

In many cases, researchers use the same social topologies (fully connected and ring graph) in the PSO algorithm to solve an optimisation problem, but there is a strong relationship between the selection of the social topology and the robustness of the algorithm to premature convergence [5]. Therefore it is needed to investigate which topology is more efficient and more accurate for the PSO-based edge detection algorithm and how we can improve accuracy or speed of the algorithm. In this paper, the influence of the chosen topology on the accuracy and speed of the PSO-based edge detection algorithm will be investigated.

2 Background

2.1 Particle Swarm Optimisation

PSO as a branch of swarm intelligence was inspired by the social behavior of animals and simulated a simplified social model such as flocking of birds and schooling of fish.

In PSO, there is a population of m particles. The position of i^{th} particle in an n-dimensional search space at time t is represented as the vector $\boldsymbol{X}_i(t) = (x_{i1}(t), x_{i2}(t), ..., x_{in}(t))$. The position of the particle is influenced by its own experience (particle and memory influence) and that of its neighbours (swarm influence). Each particle of the population has a velocity represented by $\boldsymbol{V}_i(t)$ that is used to update $\boldsymbol{X}_i(t)$ at each iteration of PSO as in equation (1).

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t) + \boldsymbol{V}_i(t+1). \tag{1}$$

The velocity is updated according to three components: current motion influence, particle memory influence, and swarm influence:

$$\boldsymbol{V}_i(t+1) = w\boldsymbol{V}_i(t) + C_1 r_1(\boldsymbol{X}_{pbest_i} - \boldsymbol{X}_i(t)) + C_2 r_2(\boldsymbol{X}_{leader} - \boldsymbol{X}_i(t)) \tag{2}$$

where r_1 and r_2 are uniform random variables between 0 and 1; w denotes the inertia weight that controls the impact of the previous velocity; C_1 and C_2 are the self and swarm confidence learning factors respectively; $\boldsymbol{X}_{pbest}$ represents the personal best position of each particle so far; and $\boldsymbol{X}_{leader}$ is the position of the leader which is the particle that is defined by a neighbourhood topology and guides other particles toward better regions of the search space.

2.2 Neighbourhood Topologies

An important feature of the PSO algorithm is the topology which defines how particles are connected to each other as an information sharing or exchanging mechanism [14]. A topology defines the social structure among a swarm's particles. The topology specifies the leader of each particle based on a typical neighbourhood graph. There are several typical neighbourhood topologies that have been proposed in the literature as follows:

- **Fully connected graph (FCG):** In this case, each particle is fully connected to the other particles (the opposite of the empty topology) [7]. In this topology, each particle is influenced by the best particle of the entire swarm (gbest), as well as its own past experience (pbest). In this case, the leader is global best particle ($leader = gbest$ in equation (2)). This topology is shown in Figure 1(a).
- **Local best graph (LBG):** There are k immediate neighbours for each particle in the graph [7]. It means each particle has a local best particle among k particles within its neighbourhood. In this topology, each particle is influenced by a leader in its local neighbourhood plus its own past experience (pbest). In this case, the leader is called the local best (lbest) particle. This topology is shown in Figure 1(b). This topology does not need to be symmetrical.

- **Ring topology (RT):** This topology is a special representation of the local best topology in which $k = 2$ [7]. It means, each particle has just two particles in its neighbourhood as shown in Figure 1(c).
- **Star graph (SG):** In this case, one particle is just connected to all other particles [14] as shown in Figure 1(d). It is called the focal particle. In this topology, particles are isolated from each other and they communicate through the focal particle. This topology is sometime called the wheel topology. In this topology, $leader = focal$ in equation (2).
- **Tree-based graph (TBG):** each particle corresponds to one node in a tree [14]. An example of this topology is shown in Figure 1(e). In this case, the leader of each particle is its parent in the tree. Whenever each child particle finds a solution better than the best particle found by its parent, the child and parent particle are exchanged. In this topology, $leader = pbest_{parent}$ in equation (2).
- **The von Neumann topology (VNT):** in this case, each particle has four neighbours within its neighbourhood and exchanges the information with them [14]. These particles are usually located in its four different directions. An example of this topology in 2-dimensional search space is shown in Figure 1(f).

In some papers (e.g., [10], [5]), it has been indicated that if the neighbourhood size of a particle increases, the performance of the swarm may deteriorate. On the other hand, if it decreases, the run time of the algorithm may be increased. In [11], it has been shown that there is a strong relation between the chosen topology for the PSO algorithm and its robustness to premature convergence to optimise some benchmark fitness functions. It has been pointed out that the main cause of premature convergence in the PSO algorithm is the kind of topology which is chosen for it [5]. In many applications of PSO algorithm, the fully connected or local best graph is mostly utilised.

2.3 PSO-Based Edge Detection Algorithm

The most important goals of the PSO-based edge detection algorithm is to detect continuous edges in noisy images. Therefore, to reduce broken edges, we proposed an encoding scheme for the particles where each particle represents the global structure of a continuous edge [19]. This edge partitions an area of an image into two regions, the light and dark regions as can be seen in Figure 2(b), such that it maximises **interset** distance between the average pixel intensities of two regions and minimises **intraset** distances within both regions.

A continuous edge is encoded into a particle as $\langle\langle o_1, o_2\rangle, \langle m_1, m_2, \ldots, m_{max/2}\rangle, \langle m_{max/2+1}, \ldots, m_{max}\rangle\rangle$, where $max + 1$ is the number of pixels on the edge. The encoding scheme has three parts: the offsets of the closest edge to each pixel of the image ($\langle o_1, o_2\rangle$) and two sets of movement direction sequences from the pixel ($\langle m_1, m_2, \ldots, m_{max/2}\rangle$ and $\langle m_{max/2+1}, m_{max/2+2}, \ldots, m_{max}\rangle$). The values of two offsets (o_1 and o_2) are integers ranging from 0 to $SqrSize - 1$ and m_i ranging from 0 to 7. Here, m_i shows the movement direction from a pixel to one of

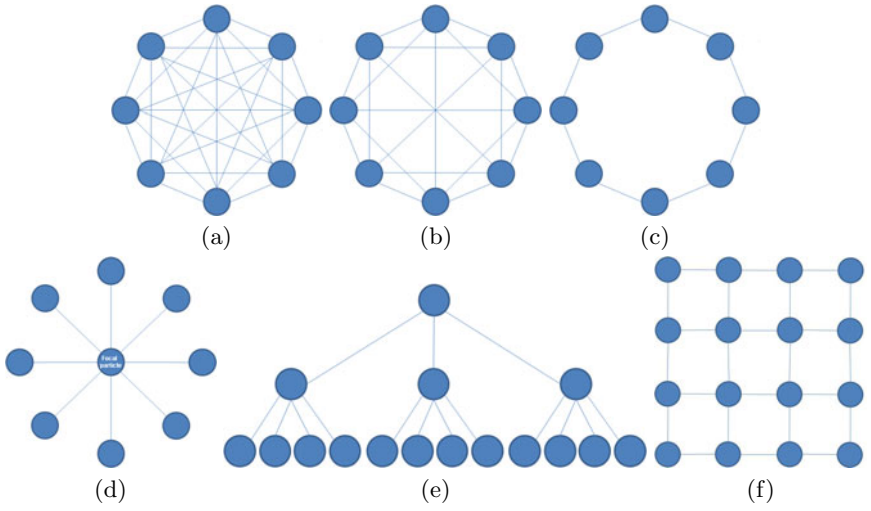

Fig. 1. Some well-known topologies used in PSO: (a) FCG (b) LBG, (c) RT, (d) ST, (e) TBG and (f) VNT

the eight possible adjacent pixels in its neighbourhood along the continuous edge as shown in Figure 2(a). For example, the particle encoding for the continuous edge in Figure 2(b) can be seen in Figure 2(c).

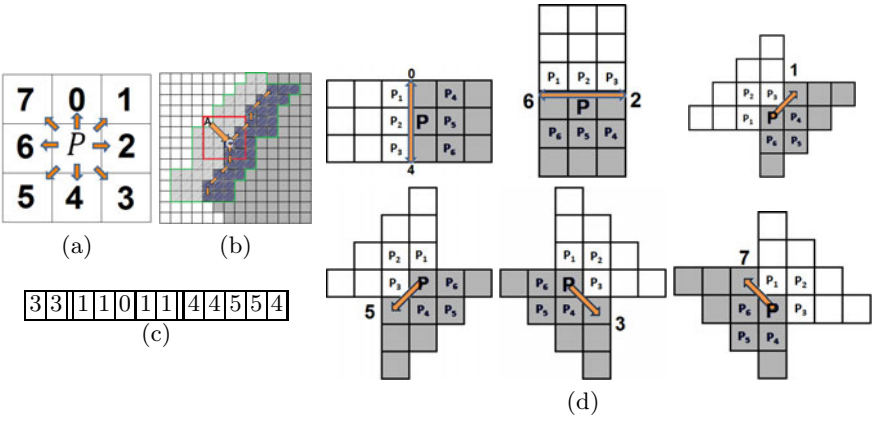

Fig. 2. The particle encoding scheme [18]: (a) eight movement directions from a pixel P; (b) an example of a curve with two regions; (c) the particle representing the curve with $max = 10$; (d) eight moving ways from pixel P to its neighbours

For evaluation of each particle at each generation of the PSO algorithm, at the first step, the intraset and interset distances of each single pixel on the continuous edge represented by the particle are calculated by the equations proposed in [18] according to eight moving ways from the pixel to its neighbours (see Figure 2(d)). Then the possibility score ($PScore$) and curvature cost factor ($CCost$) of the curve fitting on a continuous edge are computed as the proposed equations in [18]. The score in conjunction with the curvature cost factor as equation (3) is used for evaluation of each particle [18],

$$Fitness(C) = PScore(C) - CCost(C) \tag{3}$$

subject to two constraints:

$$Cross(C) = 0 \; and \; PScore(C) > HP$$

where C is the curve represented by a particle, $Cross(C)$ shows how many times the curve C crosses itself and HP is a threshold value which is defined by the user. The curves, represented by the particles, may sometimes intersect themselves, so we set a constraint $Cross(C) = 0$. On the other hand, $PScore(C)$ should be larger than HP to avoid false alarms. Therefore, $PScore(C) > HP$ as another constraint should be satisfied in the PSO algorithm. We proposed a non-stationary and multi-stage penalising method to handle these two constraints in [18]. In all experiments that we have arranged so far, the fully connected topology has been used in the PSO-based algorithm. Therefore, to evaluate the influence of using different topologies on the accuracy of the algorithm, we change the velocity equation in [18] as the equation (2) in order to specify the leader's position according to the chosen topology which defines the neighborhood structure of each particle in the PSO algorithm.

3 Experiment Design

We will compare the performance of PSO with six topologies for edge detection in noisy environment. We will describe the image set first and then the performance measure used in this paper.

To investigate the influence of chosen topology on the efficiency and effectiveness of the algorithms, we apply the algorithm on a set of benchmark images including four real images (Saturn, multi-cube, wall and road). The real images and their ground truth edge maps are available from [6]. The size of each image is 256×256 pixels and their resolution is 8 bits per pixel. These images are shown in Figure 3. All images are corrupted by two different types of noise. The noise probability for the impulsive noise ranges from 0.1 to 0.5 with a step size of 0.1. The peak signal-to-noise ratio (PSNR) value ranges from 0 to $22dB$ with a step size of $4dB$ for the Gaussian noise.

To compare the accuracy of the PSO-based edge detection algorithm with different described neighbourhood topologies, Pratt's Figure of Merit (PFOM)

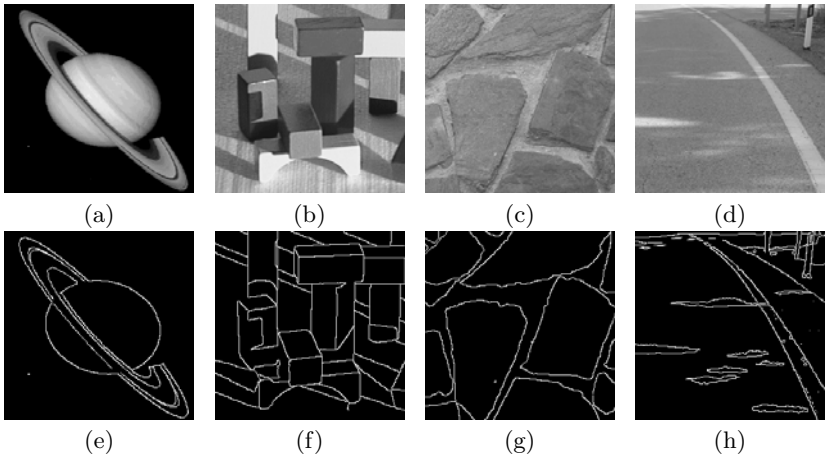

Fig. 3. (a)–(h) four real image from the UCO university and (e)–(h) their manual ground truth images [6]

is used as a quantitative measure. This measure is commonly utilised to compare the *localisation accuracy* of edge detection algorithms [13]. This measure is defined by equation (4),

$$R_{PFOM} = \frac{1}{max(I_I, I_A)} \sum_{i=1}^{I_A} \frac{1}{1 + \beta d(i)^2} \tag{4}$$

where I_I and I_A indicate the number of ideal and actual edge points in the ground truth and the generated edge map images, $d(i)$ is the distance between the pixel i in the generated edge map and the nearest ideal edge point in the ideal edge map, and β is a constant scale factor which is typically set to $\frac{1}{9}$. The ideal value of R_{PFOM} is 1.0 and the minimum could be very small. A larger value indicates stronger performance.

We use the values $w = 0.7298$, $C_1 = 1.4962$, $C_2 = 1.4962$ for PSO parameters in equation (2). The population size was set at 50 and the maximum number of iterations was 200. These values were chosen based on common settings [8]. In the PSO-based edge detection algorithm, the minimum length of a continuous edge, $max+1$ was set at 21, $SqrSize$ at 4, and HP at 0.5 [18]. For the tree-based topology, the branching factor was set at 3 [15].

4 Results

Table 1 shows PFOM estimated from the resulting images after applying the the PSO-based algorithm with different topologies. G6, G10, G14, G18 and G22 represent PSNR from 6dB to 22dB for Gaussian noise and N0.1, N0.2, N0.3, N0.4 and N0.5 represent noise probability from 0.1 to 0.5 for impulsive noise.

The columns FCG, LBG, RT, SG, TBG and VNT show the 95% confidence intervals for the localisation accuracy of the PSO-based algorithm with the fully connected, local best, ring, star, tree-based and von Neoman topologies after 30 runs for each image in each noise level. The Student two paired t-test was used to compare the pairwise accuracy means of the topologies. Alternative hypothesis was inequality of the means. The statistical analysis showed that the null hypothesis was accepted in most cases, i.e, there is often no significant difference between their means. This suggests that the topology does not have any influence on the accuracy of the algorithm.

Table 1. Comparison of accuracy of the PSO-based algorithm with different topologies

Image	Noise Level	95% Confidence Interval for Accuracy					
		FCG	LBG	RT	SG	TBG	VNT
Sat	G22	0.7728±0.0032	0.7688±0.0033	0.7684±0.0037	0.7649±0.0059	0.7568±0.0048	0.7533±0.0032
Sat	G18	0.8534±0.0026	0.8582±0.0022	0.8595±0.0018	0.8547±0.0022	0.8561±0.0040	0.8584±0.0021
Sat	G14	0.7846±0.0028	0.7867±0.0056	0.7924±0.0075	0.7953±0.0075	0.7918±0.0099	0.7976±0.0094
Sat	G10	0.8832±0.0032	0.8817±0.0023	0.8871±0.0073	0.8899±0.0050	0.8836±0.0071	0.8802±0.0068
Sat	G6	0.7674±0.0028	0.7668±0.0033	0.7630±0.0050	0.7660±0.0065	0.7707±0.0071	0.7670±0.0074
Cube	G22	0.6182±0.0032	0.6178±0.0020	0.6263±0.0008	0.6287±0.0013	0.6270±0.0011	0.6229±0.0011
Cube	G18	0.6466±0.0025	0.6399±0.0018	0.6359±0.0011	0.6384±0.0081	0.6414±0.0003	0.6440±0.0019
Cube	G14	0.5166±0.0030	0.5145±0.0018	0.5152±0.0024	0.5099±0.0086	0.5121±0.0042	0.5089±0.0062
Cube	G10	0.6333±0.0027	0.6316±0.0053	0.6346±0.0044	0.6344±0.0053	0.6368±0.0054	0.6301±0.0054
Cube	G6	0.5892±0.0027	0.5851±0.0036	0.5860±0.0015	0.5819±0.0040	0.5774±0.0016	0.5767±0.0014
Wall	G22	0.7466±0.0029	0.7585±0.0051	0.7602±0.0035	0.7453±0.0041	0.7659±0.0034	0.7649±0.0016
Wall	G18	0.7470±0.0030	0.7504±0.0033	0.7495±0.0008	0.7367±0.0008	0.7372±0.0094	0.7463±0.0034
Wall	G14	0.7913±0.0034	0.7933±0.0039	0.7977±0.0054	0.7922±0.0026	0.7939±0.0031	0.7939±0.0021
Wall	G10	0.8063±0.0030	0.8115±0.0000	0.8034±0.0003	0.7953±0.0002	0.7894±0.0006	0.7957±0.0004
Wall	G6	0.7805±0.0028	0.7820±0.0039	0.7756±0.0009	0.7796±0.0019	0.7772±0.0048	0.7789±0.0037
Street	G22	0.8091±0.0027	0.8086±0.0034	0.8046±0.0013	0.8101±0.0013	0.8208±0.0032	0.8158±0.0027
Street	G18	0.7440±0.0031	0.7510±0.0036	0.7515±0.0014	0.7442±0.0036	0.7607±0.0034	0.7514±0.0083
Street	G14	0.7468±0.0029	0.7480±0.0038	0.7562±0.0052	0.7524±0.0075	0.7532±0.0081	0.7537±0.0075
Street	G10	0.6412±0.0032	0.6388±0.0029	0.6318±0.0037	0.6337±0.0047	0.6352±0.0026	0.6347±0.0092
Street	G6	0.7502±0.0035	0.7539±0.0021	0.7504±0.0035	0.7448±0.0062	0.7353±0.0073	0.7314±0.0085
Sat	N0.1	0.4218±0.0027	0.4209±0.0063	0.4305±0.0045	0.4289±0.0073	0.4215±0.0065	0.4149±0.0080
Sat	N0.2	0.4701±0.0027	0.4693±0.0027	0.4677±0.0026	0.4656±0.0013	0.4712±0.0026	0.4746±0.0035
Sat	N0.3	0.4836±0.0029	0.4913±0.0028	0.4918±0.0030	0.4866±0.0038	0.4845±0.0021	0.4885±0.0055
Sat	N0.4	0.1912±0.0031	0.1886±0.0034	0.1873±0.0026	0.1934±0.0028	0.1932±0.0018	0.1933±0.0025
Sat	N0.5	0.1925±0.0027	0.1904±0.0013	0.1897±0.0005	0.1964±0.0026	0.1997±0.0012	0.2002±0.0018
Cube	N0.1	0.5698±0.0030	0.5707±0.0023	0.5706±0.0016	0.5699±0.0030	0.5700±0.0049	0.5685±0.0064
Cube	N0.2	0.5356±0.0029	0.5417±0.0022	0.5383±0.0023	0.5263±0.0094	0.5289±0.0002	0.5267±0.0011
Cube	N0.3	0.5344±0.0029	0.5463±0.0025	0.5515±0.0032	0.5457±0.0016	0.5369±0.0082	0.5427±0.0094
Cube	N0.4	0.4066±0.0025	0.4077±0.0010	0.4045±0.0022	0.4024±0.0045	0.3967±0.0024	0.3995±0.0027
Cube	N0.5	0.2914±0.0031	0.3035±0.0041	0.3028±0.0072	0.2909±0.0081	0.3061±0.0075	0.3012±0.0098
Wall	N0.1	0.4772±0.0025	0.4788±0.0040	0.4736±0.0075	0.4749±0.0046	0.4773±0.0009	0.4834±0.0014
Wall	N0.2	0.4887±0.0028	0.4922±0.0021	0.5005±0.0064	0.4881±0.0069	0.5002±0.0062	0.4964±0.0062
Wall	N0.3	0.5822±0.0030	0.5841±0.0027	0.5834±0.0037	0.5740±0.0040	0.5735±0.0095	0.5777±0.0065
Wall	N0.4	0.4400±0.0027	0.4389±0.0030	0.4322±0.0028	0.4374±0.0036	0.4431±0.0017	0.4433±0.0037
Wall	N0.5	0.2564±0.0030	0.2578±0.0007	0.2579±0.0033	0.2568±0.0022	0.2585±0.0006	0.2530±0.0017
Street	N0.1	0.5421±0.0030	0.5414±0.0019	0.5563±0.0020	0.5548±0.0003	0.5538±0.0005	0.5438±0.0010
Street	N0.2	0.3814±0.0030	0.3835±0.0020	0.3831±0.0026	0.3900±0.0028	0.3908±0.0085	0.3792±0.0039
Street	N0.3	0.4565±0.0028	0.4582±0.0039	0.4561±0.0023	0.4595±0.0010	0.4491±0.0001	0.4468±0.0046
Street	N0.4	0.4133±0.0022	0.4082±0.0017	0.4139±0.0027	0.4067±0.0035	0.3983±0.0002	0.3989±0.0001
Street	N0.5	0.2755±0.0033	0.2689±0.0046	0.2713±0.0046	0.2731±0.0053	0.2764±0.0037	0.2759±0.0035

Table 2 gives the 95% confidence intervals for the number of fitness function evaluations of the PSO-based algorithm with the described topologies after 30 runs for each image at each noise level. Statistic analysis showed that the number of fitness function evaluations of the algorithm with the fully connected graph is less than that of the PSO-based algorithm with the other topologies. This implies that the algorithm with the fully connected graph is faster than the algorithm with the other topologies. However, the accuracy of the algorithm with the fully connected graph topology does not have any significant difference with those of the algorithm with the other topologies. This suggests that the fully connected graph is the best topology for the PSO-based algorithm in terms of efficiency.

Table 2. Comparison of the number of the fitness function evaluations for PSO with different topologies

Image	Noise Level	Number of Fitness Function Evaluation					
		FCG	LBG	RT	SG	TBG	VNT
Sat	N0.1	354948±589	356011±571	358978±582	359835±585	358119±634	356935±654
Sat	N0.2	385045±1069	386010±1056	389105±1001	389909±1143	388017±969	387110±1055
Sat	G22	342871±414	343867±426	346829±476	347672±373	345831±412	344754±411
Sat	G14	367991±733	368995±746	372131±666	372950±651	371062±834	369955±736
Cube	N0.1	345031±361	346036±311	348974±323	350176±448	347902±393	347018±415
Cube	N0.2	374948±589	375972±582	379052±475	380050±701	377892±591	376899±639
Cube	G22	342654±297	343700±335	346492±353	347747±319	345544±245	344688±335
Cube	G14	362553±519	363527±497	366575±432	367584±522	365590±474	364609±508
Wall	N0.1	365010±1239	365996±1277	369095±1236	370103±1249	368229±1214	366920±1232
Wall	N0.2	394998±1472	395949±1451	399121±1473	400005±1493	397851±1515	396965±1476
Wall	G22	351359±918	352454±925	355419±817	356295±888	354417±994	353319±919
Wall	G14	378192±1319	379201±1338	382203±1383	383247±1327	381161±1372	380146±1317
Street	N0.1	294990±503	295954±477	299021±559	300025±541	297816±473	297039±500
Street	N0.2	324903±693	325845±680	328845±741	329780±801	327720±626	326989±753
Street	G22	271502±376	272557±382	275574±393	276416±359	274472±402	273500±375
Street	G14	310648±710	311589±662	314668±735	315693±728	313716±645	312486±709

5 Conclusions

For the PSO-based edge detection algorithm with two different constraints, it was demonstrated that the fully connected topology is the superior to the other described topologies in terms of efficiency. However the accuracy of the PSO-based edge detection algorithms was not influenced by the use of different topologies and there is no significant difference among their accuracies. These results are in contrast to the comments in the literature that the fully connected neighbourhood topology may converge to a local optima since all particles are connected together and they quickly communicate and share acquired information in the swarm. The results also showed that if the size of the particle neighbourhood is increased in the PSO-based edge detection algorithm, the algorithm speeds up meanwhile the accuracy of the algorithm is not significantly changed.

References

1. Rashidi, M.R.A., El-Hawary, M.E.: A survey of particle swarm optimization applications in electric power systems. Trans. Evol. Comp. 13(4), 913–918 (2009)

2. Baştürk, A., Günay, E.: Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. Expert Syst. Appl. 36(2), 2645–2650 (2009)
3. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
4. Chen, G., Hong Yang, Y.H.: Edge detection by regularized cubic b-spline fitting. IEEE Transactions on Systems, Man and Cybernetics 25(4), 636–643 (1995)
5. Czogalla, J., Fink, A.: Particle Swarm Topologies for Resource Constrained Project Scheduling. In: NICSO. SCI, vol. 236, pp. 61–73. Springer, Heidelberg (2009)
6. Fernández-García, N.L., Carmona-Poyato, A., Medina-Carnicer, R., Madrid-Cuevas, F.J.: Images from automatic generation of consensus ground truth for comparison of edge detection techniques, http://www.uco.es/~ma1fegan/investigacion/imagenes/ground-truth.html
7. Kennedy, F., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
8. Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization for integer programming. In: Proceedings of the 2002 Congress on Evolutionary Computation, pp. 1582–1587. IEEE Press (2002)
9. Lim, D.H.: Robust edge detection in noisy images. Comput. Stat. Data Anal. 50(3), 803–812 (2006)
10. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation 8, 204–210 (2004)
11. Montes de Oca, M.A., Stützle, T.: Convergence behavior of the fully informed particle swarm optimization algorithm. In: GECCO, pp. 71–78. ACM Press, New York (2008)
12. Pan, H., Wang, L., Liu, B.: Particle swarm optimization for function optimization in noisy environment. Applied Math. and Compu. 181(2), 908–919 (2006)
13. Pratt, W.: Digital Image Processing. Wiley Interscience (2007)
14. Reyes-Sierra, M., Coello Coello, C.A.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research 2(3), 287–308 (2006)
15. Schor, D., Kinsner, W., Anderson, J.: A study of optimal topologies in swarm intelligence. In: 23rd Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–8 (2010)
16. Setayesh, M., Johnston, M., Zhang, M.: Edge and Corner Extraction Using Particle Swarm Optimisation. In: Li, J. (ed.) AI 2010. LNCS, vol. 6464, pp. 323–333. Springer, Heidelberg (2010)
17. Setayesh, M., Zhang, M., Johnston, M.: Improving edge detection using particle swarm optimisation. In: Proceedings of the 25th International Conference on Image and Vision Computing. IEEE Press, New Zealand (2010)
18. Setayesh, M., Zhang, M., Johnston, M.: Detection of continuous, smooth and thin edges in noisy images using constrained particle swarm optimisation. In: GECCO, pp. 45–52 (2011)
19. Setayesh, M., Zhang, M., Johnston, M.: Edge detection using constrained discrete particle swarm optimisation in noisy images. In: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, pp. 246–253. IEEE Press (2011)
20. Zhao, J., Li, Z.: Particle filter based on particle swarm optimization resampling for vision tracking. Expert Systems with Applications 37(12), 8910–8914 (2010)