

Metaheuristic Algorithms: Optimal Balance of Intensification and Diversification

Xin-She Yang¹, Suash Deb² and Simon Fong^{3,*}

¹ School of Science and Technology, Middlesex University, London NW4 4BT, UK

² Department of Computer Science and Engineering, Cambridge Institute of Technology, Ranchi, India

³ Department of Computer and Information Science, University of Macau, Macau SAR

Received: 8 May. 2013, Revised: 12 Sep. 2013, Accepted: 13 Sep. 2013

Published online: 1 May. 2014

Abstract: In nature-inspired metaheuristic algorithms, two key components are local intensification and global diversification, and their interaction can significantly affect the efficiency of a metaheuristic algorithm. However, there is no rule for how to balance these important components. In this paper, we provide a first attempt to give some theoretical basis for the optimal balance of exploitation and exploration for 2D multimodal objective functions. Then, we use it for choosing algorithm-dependent parameters. Finally, we use the recently developed eagle strategy and cuckoo search to solve two benchmarks so as to confirm if the optimal balance can be achieved in higher dimensions. For multimodal problems, computational effort should focus on the global explorative search, rather than intensive local search. We also briefly discuss the implications for further research.

Keywords: Metaheuristic; Eagle strategy; Cuckoo search; Performance evaluation

1 Introduction

Many contemporary search algorithms for design optimization have been based on Turing's heuristic ideas. In fact, in addition to genetic algorithms and neural networks, there is a class of metaheuristic algorithms which are inspired by some successful characteristics of biological systems in nature [9,10]. Nature-inspired metaheuristic algorithms, especially those based on swarm intelligence, form an important part of contemporary global optimization algorithms [3,6,9,19]. Good examples are particle swarm optimization [6,20,24] and cuckoo search [12,22,23]. They work remarkably efficiently and have many advantages over traditional, deterministic methods and algorithms, and thus they have been applied in almost all areas of science, engineering and industry [4,10,21]. There are more than a dozen of swarm-based algorithms using the so-called swarm intelligence. For a detailed introduction, please refer to [7,10].

On the other hand, many design optimization problems in engineering and industry are highly nonlinear under stringent constraints, and thus often NP-hard. Therefore, to find optimal solutions to such optimization problems it is usually very challenging if not impossible.

In the last two decades, a dozen of new algorithms such as particle swarm optimization, differential evolution, ant and bee algorithms, harmony search, firefly algorithm and cuckoo search have appeared and they have shown great potential in solving tough engineering optimization problems [4,9], e-business service [25] and data mining problems [26].

In this paper, we intend to analyze the key components in metaheuristic algorithms so as to find the possibly optimal balance between these key components. It will further be confirmed by numerical experiments. To our knowledge, this is the first time in the literature to try to address the optimal balance of exploration and exploitation for metaheuristic algorithms. The paper is organized as follows: First, we briefly highlight the key components of exploration and exploitation in meta-heuristics, followed by the analysis of possible balance based intermittent search theory. Then, we introduce briefly the eagle strategy and cuckoo search. Finally, we use these algorithms to solve two test benchmarks so as to confirm the optimal choice of parameters.

* Corresponding author e-mail: ccfong@umac.mo

2 Intensification and Diversification in Metaheuristics

Meta-heuristics can be considered as an efficient way to produce acceptable solutions by trial and error to a complex problem in a reasonably practical time. The complexity of the problem of interest makes it impossible to search every possible solution or combination, the aim is to find a good feasible solution in an acceptable time scale. There is no guarantee that the best solutions can be found, and we even may not know whether an algorithm will work and why if it does work, though we may know the basic components that can help to work. The idea is to have an efficient but practical algorithm that will work most the time and is able to produce good quality solutions. Among the found quality solutions, it is expected that some of them are nearly optimal, though there is often no guarantee for such optimality.

However, many metaheuristics can typically have global convergence properties, and thus they usually can find the global optimality in practice within a relatively limited number of function iterations or evaluations. This makes it particular suitable for metaheuristic algorithms to solve global optimization. For example, cuckoo search uses not only a search technique with good convergence but also a randomization technique with more efficient Levy flights [12].

In principle, for a metaheuristic algorithm to be efficient, it has to have some special capabilities. One of such is to be able to generate new solutions that can usually be more likely to improve the previous/existing solutions and also be able to cover most important search areas where the global optimum may lie. Another capability is that an algorithm should be able to escape any local optimum so that it cannot be stuck in a local mode. A good combination may lead to good efficiency under appropriate conditions, and this often requires balancing two important components of any metaheuristics: exploration and exploitation. However, this itself is an unresolved optimization task.

The main components of any metaheuristic algorithms are: intensification and diversification, or exploitation and exploration [3,9,10]. Diversification means to generate diverse solutions so as to explore the search space on the global scale, while intensification means to focus on the search in a local region by exploiting the information that a current good solution is found in this region. This is in combination with the selection of the best solutions

2.1 Balance is the Key

An important component in modern metaheuristics is exploration, often by use of randomization [3,9], which enables an algorithm to have the ability to jump out of any local optimum so as to explore the search globally.

Randomization can also be used for local search around the current best if steps are limited to a local region. When the steps are large, randomization can explore the search space on a global scale. Fine-tuning the right amount of randomness and balancing local search and global search are crucially important in controlling the performance of any metaheuristic algorithm.

Randomization techniques can be a very simple method using uniform distributions and/or Gaussian distributions, or more complex methods as those used in Monte Carlo simulations. They can also be more elaborate, from Brownian random walks to Levy flights [12]. Some theoretical implications suggest that Levy flights can be the optimal search strategy for revising targets, while intermittent search strategy could be optimal for non-revisiting targets [1,2].

Exploitation is the use of local knowledge of the search and solutions found so far so that new search moves can concentrate on the local regions or neighborhood where the optimality may be close; however, this local optimum may not be the global optimality. Exploitation tends to use strong local information such as gradients, the shape of the mode such as convexity, and the history of the search process. A classic technique is the so-called hill-climbing which uses the local gradients or derivatives intensively.

Empirical knowledge from observations and simulations of the convergence behaviour of common optimization algorithms suggests that exploitation tends to increase the speed of convergence, while exploration tends to decrease the convergence rate of the algorithm. On the other hand, too much exploration increases the probability of finding the global optimality but with a reduced efficiency, while strong exploitation tends to make the algorithm being trapped in a local optimum. Therefore, there is a fine balance between the right amount of exploration and the right degree of exploitation. Despite its importance, there is no practical guideline for this balance. So essentially each algorithm uses different degrees of exploitation and exploration, often far from optimal [3]. Some algorithms may have intrinsically better balance among these two important components than other algorithms, that is one of the reasons why they may perform better [9,10].

2.2 Intermittent Search Theory

Even there is no guideline in practice, some preliminary work on the very limited cases exists in the literature and may provide some insight into the possible choice of parameters so as to balance these components [1,2,8]. Intermittent search strategies concern an iterative strategy consisting of a slow phase and a fast phase [1,2]. Here the slow phase is the detection phase by slowing down and intensive, static local search techniques, while the fast phase is the search without detection and can be considered as an exploration technique. For example, the static target detection with a small region of radius a in a

much larger spherical domain of radius R where $a = R$ can be modelled as a slow diffusive process in terms of random walks with a diffusion coefficient D .

It is worth pointing out that the following results assuming the targets are local optima or modes, and the objective functions are multimodal. For unimodal functions, there is no need to balance exploration and exploitation because exploitation should be used mainly in the search process. For convex unimodal, any local optimality found is also the global optimal solution, therefore, intensive local search exploiting local information and update is preferred. Intermittent switch between exploration stage and exploitation stage can be optimal for multimodal functions where the areas/volumes of the local modes are small, compared with the area/volume of the search domain. Thus, we are dealing with target modes with sparsity.

Let τ_a and τ_R be the mean times spent in intensive detection stage and the time spent in the exploration stage, respectively, in the 2D case [2]. The diffusive search process is governed by the mean first-passage time satisfying the following equations

$$D\nabla_r^2 t_1 + \frac{1}{2\pi\tau_a} \int_0^{2\pi} [t_2(r) - t_1(r)] d\theta + 1 = 0,$$

$$u \cdot \nabla_r t_2(r) - \frac{1}{\tau_R} [t_2(r) - t_1(r)] + 1 = 0,$$

where t_2 and t_1 are times spent during the search process at slow and fast stages, respectively, and u is the search speed [2].

After some lengthy mathematical analysis [1,2], the optimal balance of these two stages can be estimated as $r_{\text{optimal}} = \frac{\tau_a}{\tau_R} \approx \frac{D}{a^2} \frac{1}{[2 - \frac{1}{\ln(R/a)}]^2}$. Assuming that the search steps have a uniform velocity u at each step on average, the minimum times required for each phase can be estimated as $\tau_a^{\text{min}} \approx \frac{D}{2u^2} \frac{\ln^2(R/a)}{[2\ln(R/a)-1]}$, and $\tau_R^{\text{min}} \approx \frac{a}{u} \sqrt{\ln(R/a) - \frac{1}{2}}$.

When $u \rightarrow \infty$, these relationships lead to the above optimal ratio of two stages. An interesting observation is that the above results depend weakly on the domain size R , and thus balancing these two key components can lead to very efficient performance of the algorithm used [2]. In addition, increase the global exploration velocity u can also reduce the overall search time, and thus implicitly enhance the search efficiency of the algorithm.

It should be emphasized that the above result is only valid for 2D cases, and there is no general results for higher dimensions, except in some special 3D cases [1]. Now let us use this limited result to help choose the possible values of algorithm-dependent parameters in eagle strategy [11], as an example.

3 Eagle Strategy

Eagle strategy is a recent metaheuristic strategy for optimization, developed in 2010 by Xin-She Yang and Suash Deb [11]. More extensive studies have followed [14,5]. It uses a combination of crude global search and intensive local search employing different algorithms to suit different purposes. In essence, the strategy first explores the search space globally using a Levy flight random walk, if it finds a promising solution, then an intensive local search is employed using a more efficient local optimizer such as hill-climbing, differential evolution and/or cuckoo search. Then, the two-stage process starts again with new global exploration followed by a local search in a new region.

The advantage of such a combination is to use a balanced trade-off between global search which is often slow and a fast local search. Some trade-off and balance are important. Another advantage of this method is that we can use any algorithms we like at different stages of the search or even at different stages of iterations. This makes it easy to combine the advantages of various algorithms so as to produce better results. The main steps are outlined in 3.

```

Objective functions  $f_1(x), \dots, f_N(x)$ 
Initialization and random initial guess  $x^{t=0}$ 
while (stop criterion)
  Global exploration by randomization
  Evaluate the objectives and find a promising solution
  If  $p_e < \mathbf{rand}$ , switch to a local search
  Intensive local search around a promising solution via an
  efficient optimizer
    if (a better solution is found), Update the current
    best; end
  end
  Update  $t = t + 1$ 
end
Post-process the results and visualization. Pseudo code of
the eagle strategy.
    
```

Here the only parameter is p_e which controls the switch between local and global search. That is, it controls when to do exploitation and when to do exploration. It is worth pointing that this is a methodology or strategy, not an algorithm. In fact, we can use different algorithms at different stages and at different time of the iterations. The algorithm used for the global exploration should have enough randomness so as to explore the search space diversely and effectively. This process is typically slow initially, and should speed up as the system converges (or no better solutions can be found after a certain number of iterations). On the other hand, the algorithm used for the intensive local exploitation should be an efficient local optimizer. The idea is to reach the local optimality as quickly as possible, with the minimal number of function evaluations. This stage should be fast and efficient.

3.1 Cuckoo Search

Cuckoo search (CS) is one of the latest nature-inspired metaheuristic algorithms, developed in 2009 by Xin-She Yang and Suash Deb [12, 13]. CS is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lévy flights, rather than by simple isotropic random walks. This algorithm was inspired by the aggressive reproduction strategy of some cuckoo species such as the *ani* and *Guira* cuckoos, based on the following three idealized rules:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
- The best nests with high-quality eggs will be carried over to the next generations;
- The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

As a further approximation, this last assumption can be approximated by a fraction p_a of the n host nests are replaced by new nests (with new random solutions). Recent studies suggest that cuckoo search can outperform particle swarm optimization and other algorithms [13].

This algorithm uses a balanced combination of a local random walk and the global explorative random walk, controlled by a switching parameter p_a . The local random walk can be written as $x_i^{t+1} = x_i^t + s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t)$, where x_j^t and x_k^t are two different solutions selected randomly by random permutation, $H(u)$ is a Heaviside function, ε is a random number drawn from a uniform distribution, and s is the step size. On the other hand, the global random walk is carried out by using Lévy flights $x_i^{t+1} = x_i^t + \alpha L(s, \lambda)$, $L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}$, ($s \gg s_0 > 0$).

3.2 ES with CS

As ES is a two-stage strategy, we can use different algorithms at different stage. The large-scale coarse search stage can use randomization via Lévy flights. In the context of metaheuristics, the so-called Lévy distribution is a distribution of the sum of N identically and independently distribution random variables.

For the second stage, we can use differential evolution as the intensive local search. We know CS is a global search algorithm, it can easily be tuned to do efficient local search by limiting new solutions locally around the most promising region. Such a combination may produce better results than those by using pure CS only, as we will demonstrate this later. Obviously, the balance of local search (intensification) and global search (diversification) is very important, and so is the balance of the first stage and second stage in the ES.

3.3 Choices of Parameters

There are a few algorithm-dependent parameters. For cuckoo search, extensive parametric studies suggest that we use $n = 15$, $p_c = 0.5$ and $\lambda = 1.5$. In terms of the balance of exploration and exploitation, the most important parameter is p_e in the eagle strategy.

If we use the simple, isotropic random walks for local exploration, then we have $D \approx \frac{s^2}{2}$, where s is the step length with a jump during a unit time interval or each iteration step. From equation (2.2), the optimal ratio of exploitation and exploration in a special case of $R \approx 10a$ becomes $\frac{\tau_a}{\tau_R} \approx 0.2$. That is, we can use $p_e \approx 0.2$ in the eagle strategy.

In case of $R/a \rightarrow \infty$, we have $\tau_a/\tau_R \approx 1/8$, which implies that more times should spend on the exploration stage. It is worth pointing out that the naive guess of 50-50 probability in each stage is not the best choice. More efforts should focus on the exploration so that the best solutions found by the algorithm can be globally optimal with possibly the least computing effort.

In the case studies to be described below, we have used the eagle strategy with combination with cuckoo search to find the optimal solutions to two design benchmarks, and we found that the optimal ratio is between 0.15 to 0.25, which are roughly close to the above theoretical result. This may imply that ES with CS has an intrinsic ability of balancing exploration and exploitation close to true optimal.

4 Numerical Experiments and Design Optimization

There are a wide range of design benchmarks, and it is not possible to include even a good fraction of these benchmarks in a short paper. Therefore, we will use two well-selected case studies to demonstrate how metaheuristic algorithms perform for nonlinear global optimization problems.

4.1 Standing-Wave Function

Let us first use a multimodal test function to see how to find the fine balance between exploration and exploitation in an algorithm for a given task. A standing-wave test function can be a good example [10] $f(x) = 1 + \left\{ \exp[-\sum_{i=1}^d (\frac{x_i}{\beta})^{10}] - 2 \exp[-\sum_{i=1}^d x_i^2] \right\} \cdot \prod_{i=1}^d \cos^2 x_i$, which is multimodal with many local peaks and valleys. It has a unique global minimum at $f_{\min} = 0$ at $(0, 0, \dots, 0)$ in the domain $-20 \leq x_i \leq 20$ where $i = 1, 2, \dots, d$ and $\beta = 15$. In this case, we can estimate that $R = 20$ and $a \approx \pi/2$, this means that $R/a \approx 12.7$, and we have in the case of $d = 2$ $p_e \approx \tau_{\text{optimal}} \approx \frac{1}{2[2-1/\ln(R/a)]^2} \approx 0.19$. This indicate that the algorithm should spend 80% of its

Table 1: Variations of p_e and its effect on the solution quality.

p_e	0.5	0.4	0.3	0.2	0.1	0.05	0.025
f_{min}	8.1E-9	9.4E-11	1.2E-12	2.7E-14	7.9E-12	8.4E-11	4.9E-9

computational effort on global explorative search, and 20% of its effort on local intensive search.

For the eagle strategy with cuckoo search (ES with CS), we have used $n = 15$ and 1000 iterations. We have varied the switching probability p_e which essentially controls the exploration and exploitation in the strategy, and p_e can thus affect the solution quality. A set of 25 numerical experiments have been carried out for each value of p_e and the results are summarized in Table 1.

This table clearly shows that $p_e \approx 0.2$ provides the optimal balance of local exploitation and global exploration, which is consistent with the theoretical estimation.

Though there is no direct analytical results for higher dimensions, we can expect that more emphasis on global exploration is also true for higher dimensional optimization problems. Let us look a benchmark with seven design variables.

4.2 Cobb-Douglas Production Optimization

For a production of a series of products and the labour costs, the utility function can be written as $q = \prod_{j=1}^n u_j^{\alpha_j} = u_1^{\alpha_1} u_2^{\alpha_2} \dots u_n^{\alpha_n}$, where all the exponents α_j are non-negative, satisfying $\sum_{j=1}^n \alpha_j = 1$. This is the minimization of the utility minimize q subject to $\sum_{j=1}^n w_j u_j = K$, where $w_j (j = 1, 2, \dots, n)$ are known weights.

Using the Lagrange multiplier method to convert it to an unconstrained problem, we can obtain the solution $u_1 = \frac{K}{w_1 [1 + \frac{1}{\alpha_1} \sum_{j=2}^n \alpha_j]}$, $u_j = \frac{w_1 \alpha_j}{w_j \alpha_1} u_1$, where $(j = 2, 3, \dots, n)$. For example, in a special case of $n = 2$, $\alpha_1 = 2/3$, $\alpha_2 = 1/3$, $w_1 = 5$, $w_2 = 2$ and $K = 300$, we have $u_1 = 40$ and $u_2 = 50$.

As most real-world problems have certain uncertainty, we can now add some noise to the above problem. For simplicity, we just modify the constraint as $\sum_{j=1}^n w_j u_j = K(1 + \gamma \epsilon)$, where ϵ is a random number drawn from a Gaussian distribution with a zero mean and a unity variance, and $0 \leq \gamma \ll 1$ is a small positive number.

We now solve this optimization problem by the Eagle Strategy, and reformulating the equality by using their expectation [11]. For $\gamma = 0.01$, the results obtained by CS can be summarized in Table 2 where the values are provided with different problem size n and different numbers of iterations. We can see that the results converge at the optimal solution very quickly.

Table 2: Mean deviations from the optimal solutions.

size n	Iterations	deviations
2	1000	0.023
10	5000	0.040
50	5000	0.037
50	15000	0.019

5 Conclusions

Nature-inspired metaheuristic algorithms have now gained increasing popularity, which is partly due to their ability of dealing with nonlinear global optimization problems. In this paper, we have first highlighted the importance of key components of exploration and exploitation in metaheuristics, and then provided a simple, yet still practical estimate for the ratio of search times or efforts of exploitation and exploration stages. This estimate was based on the intermittent search theory in the 2D case. However, this simple estimation does provide some insight into the possibility that more efforts should be placed on the global exploration for multimodal problems. We then used two case studies to show that numerical experiments are consistent with the theoretical analysis.

Further research can focus on the extension of the 2D results to higher dimensions. It may also be very fruitful to investigate various type of optimization problems and see how the distribution of modes and targets can be associated with the performance of an algorithm, which may help to identify the best algorithms for each type of problem.

References

- [1] O. Bnichou, C. Loverdo, M. Moreau, and R. Voituriez, Two-dimensional intermittent search processes: An alternative to Levy flight strategies, *Phys. Rev., E*, **74**, 020102, (2006).
- [2] O. Bnichou, C. Loverdo, M. Moreau, and R. Voituriez, Intermittent search strategies, *Review of Modern Physics*, **83**, 81-129, (2011).
- [3] C. Blum and A. Roli, *Metaheuristics in combinatorial optimisation: Overview and conceptual comparison*, *ACM Comput. Surv.*, **35**, 268-308 (2003).
- [4] C. A. Floudas, and P. M. Pardalos, *Encyclopedia of Optimization*, 2nd Edition, Springer, (2009).
- [5] A. H. Gandomi, X. S. Yang, S. Talatahari, and S. Deb, Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization, *Computers & Mathematics with Applications*, **63**, 191-200 (2012).
- [6] J. Kennedy, and R. Eberhart, Particle swarm optimisation, in: *Proc. of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, 1942-1948 (1995).
- [7] R. S. Parpinelli, and H. S. Lopes, New inspirations in swarm intelligence: a survey, *Int. J. Bio-Inspired Computation*, **3**, 1-16 (2011).
- [8] M. F. Shlesinger, Random searching, *J. Phys. A.: Math. Theor.*, **42**, 434001 (2009).

- [9] X. S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, UK, (2008).
- [10] X. S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley and Sons, USA, (2010).
- [11] X. S. Yang, and S. Deb, Eagle strategy using Lvy walk and firefly algorithm for stochastic optimization, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) (Eds. J. R. Gonzalez et al.), SCI, **284**, 101-111 (2010).
- [12] X. S. Yang, and S. Deb, Cuckoo search via Lvy flights, Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009, India), IEEE Publications, USA, 210-214 (2009).
- [13] X. S. Yang, and S. Deb, Engineering optimisation by cuckoo search, Int. J. Mathematical Modelling and Numerical Optimisation, **1**, 330-343 (2010).
- [14] X. S. Yang, and S. Deb, Two-stage eagle strategy with differential evolution, Int. J. Bio-Inspired Computation, **4**, 1-5 (2012).
- [15] H. Parvin, H. Alizadeh, S. Parvin. A Scalable Classifier Ensemble Suitable for Multiclass Classification by Use of Pairwise Classifier Ensembles, Journal of Intelligent Computing, **3**, 1-13 (2012).
- [16] X. S. Yang, S. Deb, S. Fong, Accelerated Particle Swarm Optimization and Support Vector Machine for Income Prediction and Project Scheduling, Journal of Information Technology Review, **2**, 114-124 (2011).
- [17] H. Parvin, Z. Rezaei, S. Parvin. Creation of an Ensemble: Diversity Production Based Approach, Journal of E-Technology, **3**, 8-16 (2012).
- [18] T. Mekhaznia. Natural Heuristics for Cryptanalysis, Journal of Information Security Research, **3**, 81-88 (2012).
- [19] Y. Zhou, G. Zhou, J. Zhang, A Hybrid Glowworm Swarm Optimization Algorithm for Constrained Engineering Design Problems, Appl. Math. Inf. Sci., **7**, 379-388 (2013).
- [20] M. R. Girgis, T. M. Mahmoud, H. F. Abd El-Hameed and Z. M. El-Saghier. Routing and Capacity Assignment Problem in Computer Networks Using Genetic Algorithm. Information Science Letters, **2**, 13-25 (2013).
- [21] A. Chand. A heuristic approach to constraint optimization in timetabling. The South Pacific Journal of Natural Science, **20**, 64-67 (2002).
- [22] S. M. Abbas, Ilyas Saleem, Bilal Ahmed, Hunaina Khurshid. UWB Antenna with Parasitic Patch and Asymmetric Feed. Information Science Letters, **2**, 27-33 (2013).
- [23] uehui Lu, Xingri Jin, Seongjae Lee, Joo Yull Rhee, Won Ho Jang and Young Pak Lee, Passive and active control of a plasmonic mimic of electromagnetically induced transparency in stereometamaterials and planar metamaterials, Advances in Natural Sciences: Nanoscience and Nanotechnology, **1**, (2011).
- [24] Josh Barnes & Piet Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. Nature, **324**, 446-449 (1986).
- [25] M. E. J. Newman. Fast algorithm for detecting community structure in networks. Phys. Rev. E, **69**, (2004).
- [26] S. Fong, Opportunities and Challenges of Integrating Bio-inspired Optimization and Data Mining Algorithms, Swarm Intelligence and Bioinspired Computation, book chapter, Elsevier, (2013).



Xin-She Yang is Reader at Middlesex University (UK) and Adjunct Professor at Reykjavik University (Iceland). After obtained his DPhil in Applied Mathematics from Oxford University, he has worked at Cambridge University, National Physical Laboratory as a Senior Research Scientist. He is the Editor-in-Chief of International Journal of Mathematical Modelling and Numerical Optimisation (IJMMN O). He has more than 200 publications.



Suash Deb specializes in Soft Computing, Nanocomputing, Artificial Intelligence, Bioinformatics & the related fields & has published extensively in these areas. He received his BE in Mechanical Engineering from Jadavpur University, Kolkata, followed by M.Tech. in Computer Science from University of Calcutta and UNDP fellowship in Computer Science from Stanford University, USA. A Senior Member of IEEE & belonging to CIT, Ranchi, Prof. Deb has been the recipient of Bharat Excellence Award, Albert Einstein International Award for Scientific Excellence and also Rajiv Gandhi Education Excellence Award. He has also served as the Asian Expert of Advanced Research Project Agency (ARPA), Dept. of Defense, Federal Govt. of USA. He is currently on the Editorial Board of Numerous Intl. journals. He is the Editor-in-Chief of International Journal of Soft Computing & Bioinformatics, Regional Editor of Neural Computer & Applications & Advisory Editor of a no. of other journals. He was elected the President of the International Neural Network Society (INNS)- India Regional Chapter as well as the Secretary of the IEEE Computational Intelligence Society, Calcutta Chapter. He has also served as the Chair of the Task Force of Business Intelligence & Knowledge Management, IEEE Computational Intelligence Society. He has been the General Chair/ Technical Chair of numerous international events & travelled widely across the globe to deliver keynote address/plenary talk/tutorial talk etc. He is listed in a no. of WhosWho.



Simon Fong graduated from La Trobe University, Australia, with a 1st Class Honours BEng. Computer Systems degree and a PhD. Computer Science degree in 1993 and 1998 respectively. Simon is now working as an Associate Professor at the Computer and Information

Science Department of the University of Macau. He is also one of the founding members of the Data Analytics and Collaborative Computing Research Group in the Faculty of Science and Technology. Prior to joining the University of Macau, he worked as an Assistant Professor in the School of Computer Engineering, Nanyang Technological University, Singapore. Prior to his academic career, Simon took up various managerial and technical posts, such as systems engineer, IT consultant and e-commerce director in Melbourne, Hong Kong and Singapore. Some companies that he worked before include Hong Kong Telecom, Singapore Network Services, AES Pro-Data and United Oversea Bank, Singapore. Dr. Fong has published over 200 international conference and peer-reviewed journal papers, mostly in the area of intelligent applications and data-mining.