Editorial

Xin-She Yang

Mathematics and Scientific Computing,
National Physical Laboratory,
Teddington, TW11 0LW, UK
E-mail: xin-she.yang@npl.co.uk

Biographical notes: Xin-She Yang received his DPhil in Applied Mathematics from Oxford University, and he has been the recipient of Garside Senior Scholar Award in Mathematics at Oxford. He worked at Cambridge University for five years and currently a Senior Research Scientist at National Physical Laboratory. He has written a dozen books and published more than 140 papers. He is the inventor of a few metaheuristic algorithms, including accelerated PSO, bat algorithm, eagle strategy, firefly algorithm, cuckoo search and virtual bee algorithm. He is also the Editor-in-Chief of *Int. J. Mathematical Modelling and Numerical Optimisation*.

1 Swarm intelligence and metaheuristics

Metaheuristic algorithms form an important part of contemporary global optimisation algorithms, computational intelligence and soft computing. These algorithms are usually nature-inspired with multiple interacting agents. A subset of metaheuristcs are often referred to as swarm intelligence (SI)-based algorithms, and these SI-based algorithms have been developed by mimicking the so-called SI characteristics of biological agents such as birds, fish, humans and others. For example, particle swarm optimisation was based on the swarming behaviour of birds and fish (Kennedy and Eberhart, 1995), while the firefly algorithm was based on the flashing pattern of tropical fireflies (Yang, 2008, 2009) and cuckoo search algorithm was inspired by the brood parasitism of some cuckoo species (Yang and Deb, 2010).

New algorithms inspired by natural phenomena, especially biological systems, appear almost every year (Kennedy and Eberhart, 1995; Yang, 2009, 2010a). Even more studies on the extension and improvements on existing algorithms by introducing new components and new applications (Cui and Zeng, 2005; Cui and Cai, 2009; Yang, 2010b). The literature on these topics is vast, and interested readers can refer to Yang (2010b) and the references listed in the papers in this special issue.

The responses to the call for papers of this special issue were overwhelming. As metaheuristics have permeated into many areas and have many diverse applications, it is impossible to cover even a good fraction of the topics and algorithms. Among many high-quality submissions, we can only select a few papers due to the limitation of the space in a single issue. The choice was on based the quality of the papers, the coverage of topics and the state of the art. They may provide a small snapshot of the latest developments, but may also provide hints and inspiration for further research. In addition, in the last section of this editorial, I list a few important open problems so that interested readers can start to ponder these profound questions.

2 Why metaheuristics?

New researchers often ask 'Why metaheuristics?'. Indeed, this is a fundamental question to ask in the first steps of solving a given problem. How do we choose the best algorithm and why?

We are often puzzled and often surprised by the excellent efficiency of contemporary nature-inspired algorithms. Seemingly simple algorithms can work 'magic', even for very tough global optimisation problems. Many elaborate and sophisticated conventional algorithms often do not work well, despite the fact that conventional algorithms have been well tested for many years. New SI-based metaheuristics often work much better in practice, even though we may not understand why these algorithms actually work. Empirical observations, vast literature and some preliminary convergence analysis all suggest that metaheuristics do work well. Loosely speaking, the success and popularity of metaheuristics can be attributed to the following three factors: algorithm simplicity, ease for implementation, and solution diversity.

Almost all metaheuristic algorithms look simple, and their fundamental characteristics are often derived, directly and indirectly, from nature. For example, cuckoo search was based on the simple fact of cuckoo brooding behaviour in combination with Lévy flights (Yang and Deb, 2010). This leads to a simple and yet powerful algorithm with a balance of two major components: diverse exploration and intensive exploitation. Diverse solutions are generated by random permutation and Lévy flights, which increases the diversity of the quality solutions. On the other hand, local intensive exploitation uses intensive local selective random walks in combination with the selection of the best solutions (elitism), which ensures the algorithm will converge to the global optimality.

Due to the simplicity nature of metaheuristics, they are relatively easy to implement in any programming language. In fact, most algorithms can be coded in fewer than a hundred lines in most programming languages. Such simple algorithms, once implemented properly, can subsequently deal with quite a wide range of optimisation problems without reprogramming.

A key factor may be the balance between solution diversity and solution speed. Ideally, we wish to find the global best solution with the minimum computing effort. For a simple problem, especially a unimodal convex problem, efficient algorithms do exist. For example, conventional algorithms such as hill-climbing or steepest descent methods can find the best solutions in an efficient way. However, real-world problems are not linear, and they certainly are not convex. The multimodality and complexity of the problem of interest may mean that we cannot find the global optimality with 100% certainty, unless in a very few limited classes of problems. To reduce the computing time, we often sacrifice the diversity of solutions, and consequently, we may be doing a local search, or the search process is trapped in a local optimum. In order to escape the local optima, we have to increase the diversity of new solutions so as to potentially reach the true global optimality. The diversity of the solutions in the search process can be achieved in many ways, though randomisation and stochastic intervention are often used in most metaheuristics. Now a natural question is how to ensure the proper degree of diversity in the solutions?

In fact, two major components in metaheustics are local exploitation and global exploration. Local exploitation uses local information obtained in local search, and tries to ensure the maximum convergence, while global exploration tends to explore different feasible regions in the whole search to ensure the global optimality can be achieved with the maximum likelihood. Obviously, these two components are conflicting, and we have to main a tradeoff or balance.

Now the question what is the best balance between these two components. This is an important question, but without satisfactory answer at the moment. More studies in this area are highly needed.

3 Can algorithms be intelligent?

The popularity of metaheuristics often prompts readers to ask 'Can algorithms be intelligent?' The short answer is 'possibly'.

Artificial intelligence has been an active research area for more than half a century, and new areas such as computational intelligence are going strong as ever. However, unless a Turing test can be really passed in the future, truly intelligent algorithms may be still a long way to go. Obviously, we can define the intelligence by different degrees of mimicking the human intelligence. In that sense, we have been trying to incorporate 'intelligence' in the smart algorithms of metaheuristics gradually and incrementally, with some promising results.

First, use of memory in the form of selection of the best solutions, elitism and Tabu search is a hint of some intelligence. After all, memory is an important part of human intelligence.

Second, connectionism, interactions and share information can also be considered as 'intelligence'. Many algorithms such as artificial neural networks use interactions and connectionism to link inputs to outputs in a complex, implicit manner. In many metaheuristics, multiple agents often can share the best solutions found so far so that new search and solutions are guided by such information.

Thirdly, many algorithms use the so-called SI by use certain rules derived from swarm behaviour. These rules essentially ensure the interactions between multiple agents are guided by local information such as the flashing light used in the firefly algorithm (Yang, 2008, 2009) or the individual best solution in history found by individual particles in the particle swarm optimisation (Kennedy and Eberhart, 1995). Mathematically speaking, these interacting agents form biased interacting Markov chains whose convergence rate can be influenced by the structure of the algorithms.

Finally, an algorithm can be called 'smart' if it somehow can automatically adjust its behaviour according to the landscape of the objective functions and the information obtained during the search process. If an algorithm with automatic parameter tuning can adjust its algorithm-dependent parameters automatically so as to increase the rate of convergence and reduce the computing cost (Yang, 2011), it may implicitly act as an 'intelligent' way.

Obviously, truly intelligent algorithms may only emerge in the far future, however, whatever the forms they may take, they will have a profound impact in almost every area of science, engineering and industrial applications.

4 Important open problems

Despite the increasing popularity of metaheuristics, many crucially important questions remain unanswered. These open questions span a diverse range of areas. Here I highlight a few but relevant open problems.

- *Framework:* Convergence analysis has been fruitful, however, it is still highly needed to develop a unified framework for algorithmic analysis and convergence.
- *Exploration and exploitation:* Two important components of metaheuristics are exploration and exploitation or diversification and intensification. What is the optimal balance between these two components?
- *Performance measure:* To compare two algorithms, we have to define a measure for gauging their performance. At present, there is no agreed performance measure, but what are the best performance measures?

- *Free lunches:* No-free-lunch theorems have not been proved for continuous domains for multiobjective optimisation. For single-objective optimisation, continuous free lunches are possible, is this true for multiobjective optimisation? In addition, no free lunch theorem has not been proved to be true for problems with NP-hard complexity. If free lunches exist, what are their implications in practice and how to find the best algorithm(s)?
- *Knowledge:* Problem-specific knowledge always helps to find an appropriate solution? How to quantify such knowledge?
- *Intelligent algorithms:* A major aim for algorithm development is to design better, intelligent algorithms for solving tough NP-hard optimisation problems. What do mean by 'intelligent'? What are the practical ways to design truly intelligent, self-evolving algorithms?

References

Cui, Z.H. and Cai, X.J. (2009) 'Integral particle swarm optimisation with dispersed accelerator information', *Fundam. Inform.*, Vol. 95, No. 4, pp.427–447.

Cui, Z.H. and Zeng, J.C. (2005) 'A modified particle swarm optimisation predicted by velocity', *GECCO 2005*, pp.277–278.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimisation', in *Proc. of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, pp.1942–1948.

Yang, X.S. (2008) *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK.

Yang, X.S. (2009) 'Firefly algorithms for multimodal optimisation', in O. Watanabe and T. Zeugmann (Eds.): *Lecture Notes in Computer Science*, Vol. 5792, pp.169–178, *Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications, SAGA 2009*.

Yang, X.S. (2010a) 'Firefly algorithm, stochastic test functions and design optimisation', *Int. J. Bio-Inspired Computation*, Vol. 2, No. 2, pp.78–84.

Yang, X.S. (2010b) *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley and Sons, USA.

Yang, X.S. (2011) 'Chaos-enhanced firfly algorithm with automatic parameter tuning', *Int. J. Swarm Intelligence Research*, Vol. 2, No. 4, pp.1–11.

Yang, X.S. and Deb, S. (2010) 'Engineering optimisation by cuckoo search', *Int. J. Math. Modelling & Num. Optimisation*, Vol. 1, No. 4, pp.330–343.