

# A comparison of multi-objective optimisation metaheuristics on the 2D airfoil design problem

S. Mirjalili<sup>1</sup>

T. Rawlins<sup>2</sup>

J. Hettenhausen<sup>3</sup>

A. Lewis<sup>4</sup>

(Received 4 October 2012; revised 14 June 2013)

## Abstract

Variants of the multi-objective particle swarm optimisation (MOPSO) algorithm are investigated, mainly focusing on swarm topology, to optimise the well-known 2D airfoil design problem. The topologies used are global best, local best, wheel, and von Neumann. The results are compared to the non-dominated sorting genetic algorithm (NSGA-II) and multi-objective tabu search (MOTS) algorithm, and it is found that the attainment surfaces achieved by some of the MOPSO variants completely dominate those of NSGA-II. In general, the MOPSO algorithms also significantly improve diversity of solutions compared to MOTS. The

---

<http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6154> gives this article, © Austral. Mathematical Soc. 2013. Published June 30, 2013, as part of the Proceedings of the 16th Biennial Computational Techniques and Applications Conference. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

MOPSO algorithm proves its ability to exploit promising solutions in the presence of a large number of infeasible solutions, making it well suited to problems of this nature.

*Subject class:* 68

*Keywords:* Multi-objective particle swarm optimization; MOPSO; 2D airfoil design

## Contents

<b>1 Introduction</b>	<b>C346</b>
<b>2 Multi-objective particle swarm optimisation</b>	<b>C347</b>
<b>3 2D airfoil design problem</b>	<b>C349</b>
<b>4 Results and discussion</b>	<b>C351</b>
<b>5 Conclusion</b>	<b>C356</b>
<b>References</b>	<b>C358</b>

## 1 Introduction

There is increasing use of heuristic optimisation algorithms for real world problems, such as the 2D airfoil design. In the past, solutions of this problem were undertaken by a priori aggregation of objectives, such as lift and drag coefficients, into a single objective, the lift/drag ratio [1]. Maintaining the multi-objective formulation of the problem allows the exploration of the behaviour of the airfoil across a range of design parameters and operating conditions, but requires the use of more complex metaheuristics and a need to address conflicting objectives.

Particle swarm optimisation (PSO) [2] is readily applicable to many problems due to its simplicity and effectiveness. Inspired by the social behaviour of various species, it uses a number of particles (candidate solutions) moving around the search space to find the best solution. Particles are guided by their own best found solutions, as well as the best solution the swarm has obtained so far.

Different variants of the PSO algorithm focusing on the swarm topology have been developed. Some topologies are global best [2, 3], local best [3], wheel [4], and von Neumann [4]. In this article the 2D airfoil design problem is used to investigate the capabilities of these different variants of the multi-objective PSO (MOPSO) algorithm. A comparative study with the non-dominated sorting genetic algorithm (NSGA-II) [5] and multi-objective tabu search (MOTS) [6] is also conducted in order to verify the results.

Section 2 presents a brief introduction to MOPSO. Section 3 discusses the basic principles of the 2D airfoil design problem. The experimental results are demonstrated in Section 4. Finally, Section 5 concludes the work and suggests some directions for future research.

## 2 Multi-objective particle swarm optimisation

The multi-objective version of PSO was first proposed by Coello [7]. Following the same concepts of PSO, MOPSO employs a number of particles which move around the search space to find the best solution. Meanwhile, they all trace the best location (best solution) in their paths. In other words, particles consider their own non-dominated best solutions (pbest) as well as one of the non-dominated solutions the swarm has obtained so far (gbest) for updating position. The mathematical model is [8]

$$\mathbf{v}_i(\mathbf{t} + 1) = w\mathbf{v}_i(\mathbf{t}) + c_1 \text{rand}(\text{pbest}_i - \mathbf{x}_i(\mathbf{t})) + c_2 \text{rand}(\text{gbest} - \mathbf{x}_i(\mathbf{t})), \quad (1)$$

$$\mathbf{x}_i(\mathbf{t} + 1) = \mathbf{x}_i(\mathbf{t}) + \mathbf{v}_i(\mathbf{t} + 1), \quad (2)$$

where  $v_i(\mathbf{t})$  is the velocity of particle  $i$  at iteration  $\mathbf{t}$ ,  $w$  is the inertial weight,  $c_1$  is the cognitive coefficient,  $c_2$  is the social coefficient,  $\text{rand}$  is a random number in  $[0, 1]$ ,  $x_i(\mathbf{t})$  is the position of particle  $i$  at iteration  $\mathbf{t}$ ,  $\text{pbest}_i$  is the best non-dominated solution that the  $i$ th particle has found so far, and  $\text{gbest}$  indicates one of the non-dominated solutions the swarm has obtained so far. The first term of (1) provides stability for PSO. The second and third terms contribute to the algorithm's exploitation of known good solutions. An external archive is generally used for storing and retrieving the obtained Pareto optimal solutions.

This article explores the suitability of various algorithms when applied to a problem with competing objectives (design of an airfoil to maximise lift and minimise drag). Furthermore, the problem chosen proves to be particularly challenging due to the sparsity of feasible solutions. To address this issue a number of features were incorporated in the algorithm.

A mutation operation called turbulence is embedded in MOPSO to increase randomness and promote diversity of trial solutions. The proposed external archive was designed to save the non-dominated solutions obtained so far. It has two main components: an archive controller and a grid. The archive controller is responsible for deciding if a solution should be added to the archive or not. The grid is responsible for keeping the archive solutions as diverse as possible.

The MOPSO algorithm handles constraints whenever two solutions are being compared. In comparing two feasible solutions the non-dominance comparison is applied directly. In comparing feasible and infeasible solutions the feasible solution is selected. Among two infeasible solutions the solution with less constraint violation is chosen.

The MOPSO algorithm starts by randomly placing the particles in a problem space. Over the course of iterations, the velocities of particles are calculated using (1). After defining the velocities, the position of particles is calculated by (2). All non-dominated solutions are added to the archive. Finally, the search process is terminated by satisfying a stopping criterion.

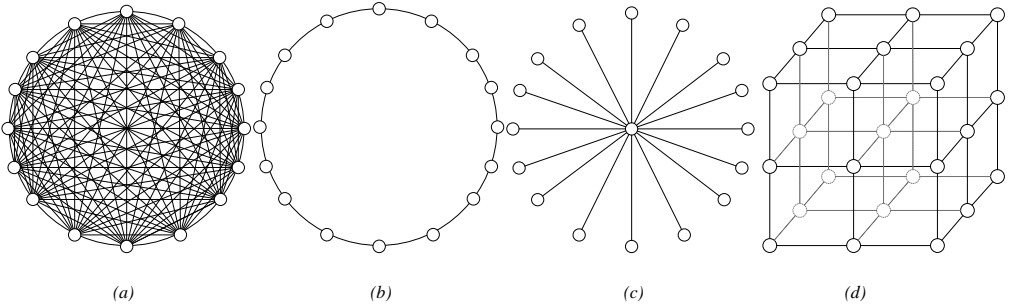


Figure 1: Different topologies for MOPSO.

The original version of MOPSO used the global (MOPSOg) topology illustrated in Figure 1(a). A topology dictates how particles share information. Some other topologies are: local (MOPSOl), Figure 1(b); wheel (MOPSOw), Figure 1(c); and von Neumann (MOPSOv), Figure 1(d). In the following sections the MOPSO algorithm is equipped with each of these topologies in turn and employed for optimising the shape of a 2D airfoil.

### 3 2D airfoil design problem

The 2D airfoil design problem is essentially one of optimising the shape of the airfoil to maximise lift and minimise drag. As visualised in Figure 2, the engines provide thrust for moving the aircraft forward, and thrust is converted to lift by the wings. However, air flow around the wing brings drag against the thrust, the magnitude being influenced by the shape of the airfoil. The objective functions (lift and drag coefficients) are  $F(C_L) = -C_L/C_{L,\text{datum}}$  and  $F(C_D) = C_D/C_{D,\text{datum}}$  where ‘datum’ is used for a NACA0012 airfoil section [9]. The NACA0012 airfoil is symmetrical. The 00 indicates that it has no camber and the 12 indicates that the airfoil has a thickness to chord length ratio of 12 (it is 12% as thick as it is long).

The freeware XFoil [10] was used for calculating the objective functions

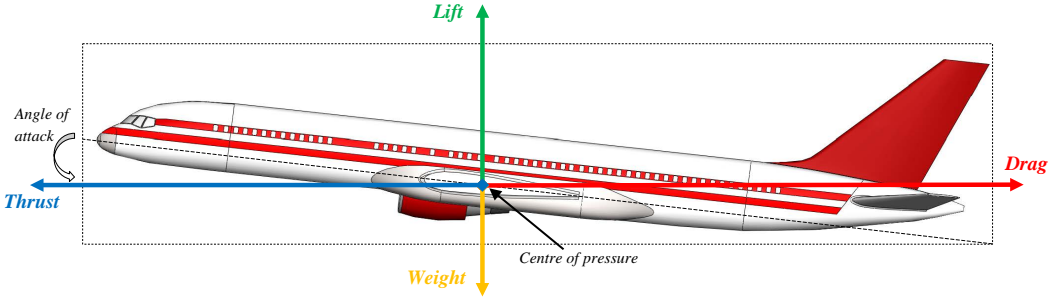


Figure 2: Lift and drag.

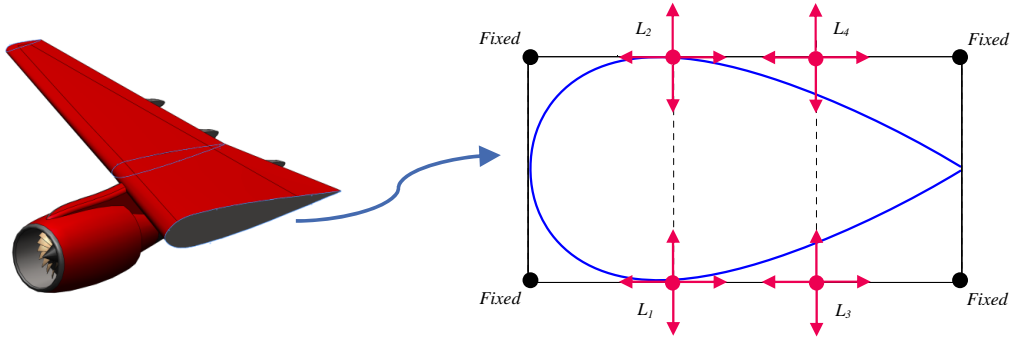


Figure 3: Schematic of 2D airfoil and FFD structure.

(lift and drag). As illustrated in Figure 3, the free form deformation (FFD) parametrisation method [11] is utilised for defining the shape of the airfoil using four control points. In this method the control points define the final shape of an airfoil curvature. In our case, the control points are allowed to move in the X and Y dimensions. Leading and trailing edges of the airfoil are fixed.

This problem is formulated as:

$$\begin{aligned}
 &\text{suppose} && \vec{x} = (L_1^x, L_1^y, L_2^x, L_2^y, L_3^x, L_3^y, L_4^x, L_4^y) \\
 & && = (P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8) ; \\
 &\text{maximise} && F(C_L) = -\frac{C_L}{C_{L,\text{datum}}} ; \\
 &\text{minimise} && F(C_D) = \frac{C_D}{C_{D,\text{datum}}} ; \\
 &\text{subject to} && \text{thickness, leading curvature;} \\
 &\text{parameter range} && -1 \leq P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8 \leq 1 .
 \end{aligned} \tag{3}$$

This problem has eight variables, each within the range  $[-1, 1]$ . There are two main constraints related to the thickness and leading curvature. The first constraint is to keep the required space for internal wing components. The second constraint is to minimise problems with flow separation and increase stability of performance at operating extremes. The operating conditions simulated test performance of the airfoil when close to stalling. Finally, the objectives are to maximise life and minimise drag.

According to our observations and the nature of aerodynamic problems, the constraints and method of shape manipulation lead to many infeasible solutions, which make the problem highly sensitive to variations in the shape parameters.

## 4 Results and discussion

In this section the MOPSO algorithm is equipped with the different topologies and compared with NSGA-II and MOTS. As is the convention, we set parameters  $c_1, c_2 = 2$  in (1) as per the advice of Shi and Eberhart [8]. The inertial weight  $w$  was decreased linearly from 0.9 to 0.4 for this algorithm as per Eberhart and Shi [12]. For NSGA-II, the crossover and mutation were chosen as blend and Gaussian [5]. List size, neighbourhood size, and movement method were set to 15, 20 and swap, respectively, as recommended by Jaeggi et al. [6].

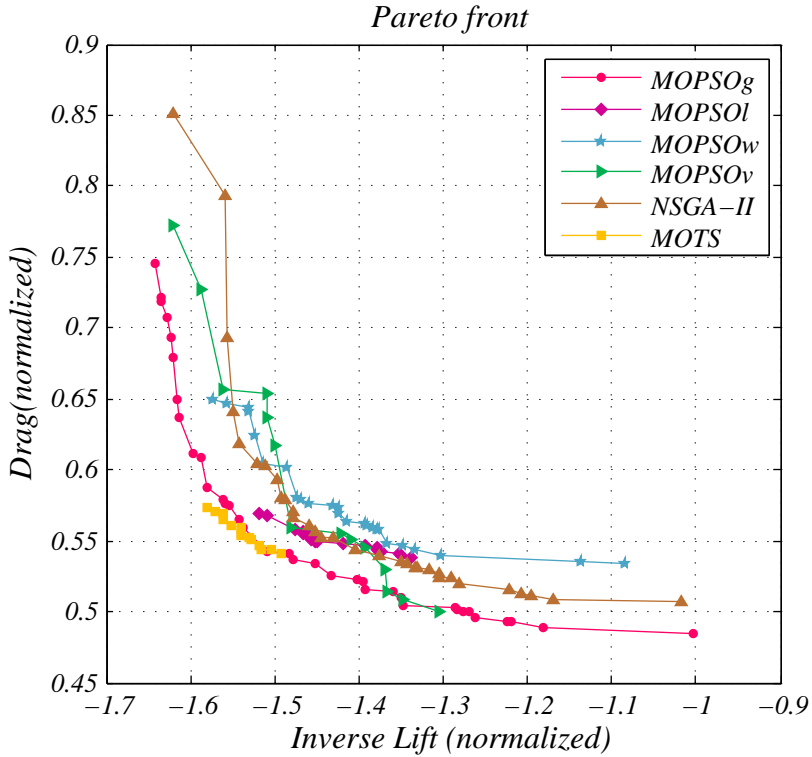


Figure 4: Attainment surfaces (the true Pareto front is unknown).

All random values were chosen from a uniform random distribution in the interval  $[0, 1]$ . The stopping criterion is maximum number of iterations (200) and all algorithms were implemented in Python. Finally, each algorithm was run on the problem four times and the best attainment surface selected.

As seen in Figure 4, the attainment surfaces achieved by most of the MOPSO variants completely dominated those of NSGA-II. The global topology yielded the best results, followed by local and von Neumann topologies. In other words, MOPSOg dominates some topologies, whereas MOPSOl dominates others.



The range of Pareto solutions obtained are shown in Figure 4. For example, MOPSOg provides lift from about  $-1.65$  to  $-1$  (expressed as a negative value to allow expression of the problem as one of minimisation) and drag from  $0.47$  to  $0.75$ , whereas MOPSOl provides lift from approximately  $-1.55$  to  $-1.29$  and drag from  $0.53$  to  $0.57$ . In general, all MOPSO algorithms have significantly greater range of solutions compared to MOTS. The reason for this might be due to the characteristics of the 2D airfoil problem search space.

The histories of particles within the search space, in terms of feasible and infeasible regions, are illustrated in Figure 5. The white squares indicate the infeasible solutions, whereas the black ones show the feasible solutions. This figure shows that in some cases more than half of the trial solutions tested were infeasible, suggesting that the search space of the 2D airfoil design problem has only sparse feasible regions. Considering this characteristic, a possible reason for the superior results of MOPSO algorithms is that they use established feasible solutions as guides for particles. Even if a particle strays into infeasible areas of the search space, it has a reference point in the feasible area to guide its search. However, MOTS avoids infeasible regions by temporarily adding them to a blacklist which makes it more difficult for it to cross infeasible areas to find new feasible solutions to exploit. Note that the blacklist in MOTS was established to store unpromising solutions with the purpose of avoiding possible infeasible regions around them. NSGA-II also does not provide good exploitation because it ignores solutions in the previous generations which might have helpful information about the feasible areas of the search space.

According to these results, MOPSO has an ability to exploit promising solutions (primarily the non-dominated solutions) in the presence of a large number of infeasible solutions, making it well suited to problems of this nature. Handling feasibility is a critical capability in optimisation, especially in aerodynamic problems.

The behaviour of particles in MOPSO variants with different topologies is also interesting to consider. As seen in Figure 5, at first MOPSOl finds just



Figure 5: Feasible and infeasible particles in the search space for, from left to right, topologies MOPSOg, MOPSOl, MOPSOw and MOPSOv. White squares indicate infeasible solutions and black squares feasible solutions.

a few feasible solutions in the initial population. Information about the feasible regions is communicated to nearest neighbours, slowly spreading out through the population, around the ring shown in Figure 1(b). A similar pattern is also observed for MOPSOv, but the spread of the information about feasible solutions is more rapid due to the greater degree interconnectivity in the network, as seen in Figure 1(d). In contrast, all particles in MOPSOg immediately have global knowledge of feasible particles since they are globally interconnected, as shown in Figure 1(a), and rapidly move into feasible regions. They do not appear to ‘cluster’ around just a few feasible regions like MOPSOl and, to a lesser extent, MOPSOv. Possibly ‘churning’ or rapid changing of guide particles (the particles  $p_{best}$  and  $g_{best}$  in (1)) leads to a greater degree of exploration and less focus on exploitation. Further investigation of particle dynamics is required to confirm this hypothesis. MOPSOw seems to show behaviour midway between the other MOPSO approaches. Initially few feasible solutions are found, but when an attractive feasible solution  $is$  found knowledge of it is immediately spread to all particles via the ‘hub’ of the wheel. It is not clear from the particle histories shown in Figure 5 alone whether this leads to intensive exploitation of just a few feasible solutions.

Since the investigated problem has two objectives the behaviour of the algo-

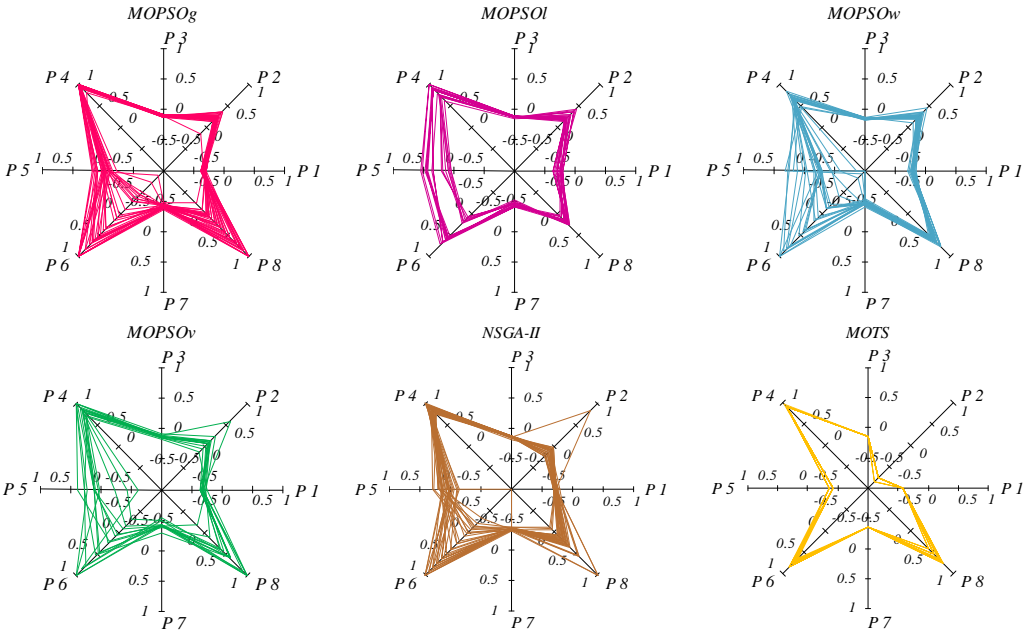


Figure 6: Parallel coordinates.

gorithms in finding the final optimum designs is not easy to extract. Therefore, we need some tools for extracting information from the results. One of the methods for post-analysis of the results is parallel coordinates figures [13]. Parallel coordinates were first proposed by Inselberg in 1985 [14] for visualising information. This method maps an  $n$  dimensional space to indexed subsets of a two dimensional space ( $\mathbb{R}^n \mapsto \mathbb{R}^2$ ). According to Kipouros et al. [13], the complexity of analysis is reduced by parallel coordinates. Moreover, this method of visualisation assists with easily extracting patterns and useful information from the results because it is more readily understandable for users. In addition, there is potential to reveal information about the nature of the problems. A parallel coordinates representation of the solutions found by various algorithms is presented in Figure 6.

Figure 6 shows that MOPSO algorithms provide much more diverse values for the parameters compared to MOTs. Among the MOPSO variants, MOPSOL shows discontinuous patterns for two of the parameters ( $P_5$  and  $P_6$ ) confirming it exploited just a few feasible regions, as the patterns in Figure 5 suggested. MOPSOw again shows results between those of MOPSOg and MOPSOL. However, it missed the diversity of values for  $P_8$  that were found by MOPSOg, so is possibly exploiting fewer feasible regions. Comparing the plots,  $P_6$  has a wide range of values compared to other parameters. This suggests that  $L_3^y$  and, to a lesser extent,  $P_8 = L_4^y$  adopt a wide range of values across solutions in the Pareto optimal set, from solutions with high lift to those with low drag. In other words, movements of the trailing control points in the Y direction play key roles in defining the final values of lift and drag. This concept is evident in the design of aircraft wings where the flaps on the trailing edge of a wing are adjustable, allowing aircraft to develop high lift during take-off and landing, but minimise drag when cruising.

The results using NSGA-II are similar to those of MOPSO, with similar ranges of parameters. This corresponds to the broad extent of the attainment surface for NSGA-II in Figure 4. However, as seen in Figure 4, MOPSOg and MOPSOL have attainment surfaces that are further converged than NSGA-II, dominating its results across the majority of the front, as was observed earlier.

The optimised shapes of airfoils are shown in Figure 7. The effect of the diverse solutions on the airfoil shapes are apparent. The MOPSO algorithm provides decision makers with a wide range of design options.

## 5 Conclusion

The effects of different topologies on MOPSO algorithms were investigated. The 2D airfoil design was selected to examine the algorithms and proved to be a challenging problem, due to the sparsity of feasible solutions. Also, a comparative study was undertaken with NSGA-II and MOTs.

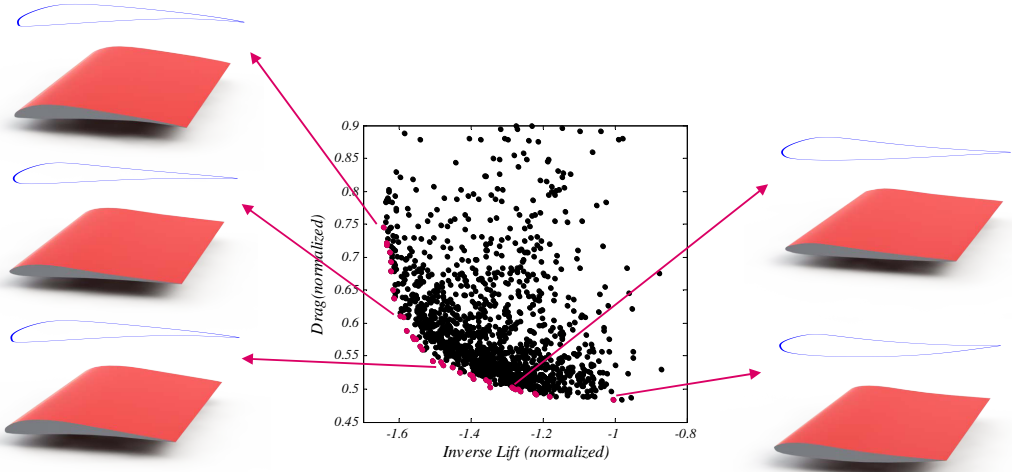


Figure 7: Search history of MOPSO front and the optimised airfoil shapes.

The results showed that MOPSO algorithms dominated the results of NSGA-II and provided much more diverse results compared to MOTS. The superior results of MOPSO appeared to be due to its superior ability to deal with the high number of infeasible solutions over the course of iterations. MOPSO appeared capable of crossing infeasible regions, whereas MOTS ignored search agents once they became infeasible and NSGA-II lost information about feasible solutions between generations. Finally, the results from different communication topologies in MOPSO revealed that topologies with lower degrees of interconnectivity tended to over-exploit the rare feasible solutions found at the expense of exploration, delivering poorer approximations to the Pareto front.

For future studies it would be interesting to further explore the reasons underlying the superior performance of MOPSO using a global communication topology. Further analysis of search agents may also find other reasons for the different performance of the algorithms.

## References

- [1] D. Abramson, A. Lewis, T. Peachey, and C. Fletcher. An automatic design optimization tool and its application to computational fluid dynamics. In *Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)*, pages 25–25. ACM, 2001. doi:[10.1145/582034.582059](https://doi.org/10.1145/582034.582059). C346
- [2] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995. doi:[10.1109/MHS.1995.494215](https://doi.org/10.1109/MHS.1995.494215). C347
- [3] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1671–1676. IEEE, 2002. doi:[10.1109/CEC.2002.1004493](https://doi.org/10.1109/CEC.2002.1004493). C347
- [4] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Evolutionary Computation, 1999. CEC'99. Proceedings of the 1999 Congress on*, volume 3. IEEE, 1999. doi:[10.1109/CEC.1999.785509](https://doi.org/10.1109/CEC.1999.785509). C347
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002. doi:[10.1109/4235.996017](https://doi.org/10.1109/4235.996017). C347, C351
- [6] D. M. Jaeggi, G. T. Parks, T. Kipouros, and P. J. Clarkson. The development of a multi-objective tabu search algorithm for continuous optimisation problems. *European Journal of Operational Research*, 185(3):1192–1212, 2008. doi:[10.1016/j.ejor.2006.06.048](https://doi.org/10.1016/j.ejor.2006.06.048). C347, C351
- [7] C. A. Coello Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Evolutionary*

- Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1051–1056. IEEE, 2002. doi:[10.1109/CEC.2002.1004388](https://doi.org/10.1109/CEC.2002.1004388).  
C347
- [8] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995. doi:[10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).  
C347, C351
- [9] E. N. Jacobs, K. E. Ward, and R. M. Pinkerton. NACA report no. 460: The characteristics of 78 related airfoil sections from test in the variable-density wind tunnel, 1933. [http://www.esdu.com/cgi-bin/ps.pl?sess=unlicensed\\_1130622070421svf&t=doc&p=naca\\_tr460](http://www.esdu.com/cgi-bin/ps.pl?sess=unlicensed_1130622070421svf&t=doc&p=naca_tr460).  
C349
- [10] M. Drela. XFOIL: An analysis and design system for low Reynolds number airfoils. In T. J. Mueller, editor, *Low Reynolds Number Aerodynamics*, volume 54 of *Lecture Notes in Engineering*, pages 1–12. Springer Berlin Heidelberg, 1989. doi:[10.1007/978-3-642-84010-4\\_1](https://doi.org/10.1007/978-3-642-84010-4_1).  
C349
- [11] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *ACM Siggraph Computer Graphics*, 20(4):151–160, 1986. doi:[10.1145/15886.15903](https://doi.org/10.1145/15886.15903). C350
- [12] R. C. Eberhart and Y. Shi. Particle swarm optimization: developments, applications and resources. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 81–86. IEEE, 2001. doi:[10.1109/CEC.2001.934374](https://doi.org/10.1109/CEC.2001.934374). C351
- [13] T. Kipouros, M. Mleczko, and A. M Savill. Use of parallel coordinates for post-analyses of multi-objective aerodynamic design optimisation in turbomachinery. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 2138. AIAA, 2008. doi:[10.2514/6.2008-2138](https://doi.org/10.2514/6.2008-2138). C355

- [14] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985. doi:[10.1007/BF01898350](https://doi.org/10.1007/BF01898350). C355

## Author addresses

1. **S. Mirjalili**, School of Information and Communication Technology, Griffith University, Australia.  
<mailto:seyedali.mirjalili@griffithuni.edu.au>
2. **T. Rawlins**, School of Information and Communication Technology, Griffith University, Australia.  
<mailto:tim.rawlins@griffithuni.edu.au>
3. **J. Hettenhausen**, School of Information and Communication Technology, Griffith University, Australia.  
<mailto:jan.hettenhausen@griffithuni.edu.au>
4. **A. Lewis**, School of Information and Communication Technology, Griffith University, Australia.  
<mailto:a.lewis@griffith.edu.au>