Optimisation of flight dynamic control based on many-objectives meta-heuristic: a comparative study

Sujin Bureerat, Nantiwat Pholdee* and Thana Radpukdee

*E-mail: nantiwat@kku.ac.th

Abstract. Development of many objective meta-heuristics (MnMHs) is a currently interesting topic as they are suitable to real applications of optimisation problems which usually require many objectives. However, most of MnMHs have been mostly developed and tested based on standard testing functions while the use of MnMHs to real applications is rare. Therefore, in this work, MnMHs are applied for optimisation design of flight dynamic control. The design problem is posed to find control gains for minimising; the control effort, the spiral root, the damping in roll root, sideslip angle deviation, and maximising; the damping ratio of the dutch roll complex pair, the dutch roll frequency, bank angle at pre-specified times 1 seconds and 2.8 second subjected to several constraints based on Military Specifications (1969) requirement. Several established many-objective meta-heuristics (MnMHs) are used to solve the problem while their performances are compared. With this research work, performance of several MnMHs for flight control is investigated. The results obtained will be the baseline for future development of flight dynamic and control.

1. Introduction

Recently, meta-heuristics (MHs) also known as evolutionary algorithms (EAs) are very popular in the applications to engineering design optimisation. The methods mostly developed according to inspiration of physical law or natural phenomena such as genetic evolutionary, food finding of animal or insect, etc [1-7]. The advantage of MHs is that they are global search optimisation. With the balance of exploitation and exploration, they can avoid local trapping. In addition, due to no requirements of function derivative of MHs, they can deal with almost any kind of objective function and design variables. Also, they can explore a Pareto front in a single optimisation run for the multiobjective (usually for 2-3 objectives) or many objective problem (more than 3 objectives). In this regard, MHs are becoming more popular than the classical gradient-based methods nowadays, particularly for multiobjective and many-objective optimisation problems. However, MHs usually have low search converge-ence and consistency due to some random strategies in its search procedure. The development of a new MH to eliminate this drawback is therefore challenging.

Development of a meta-heuristic for many-objective optimisation problems is a currently interesting topic as most of real engineering design problems usually need to optimise several
objective functions at the same time. Several ideas to develop MHs for many-objective problems (many objective meta-heuristics, MnMHs) have been presented, for example, modifying the selection and/or updating mechanisms [8, 9], reducing the spaces of objective functions [10-12], applying a two-archive technique [13, 14], using a point-preference approach [15-17]. Although several MnMHs have been proposed successfully, researchers have usually focused on implementing them for test functions problem without considering constraints [12]. Therefore, investigation of MnMHs performance for real engineering applications is still wide open.

This paper presents a comparative study of several MnMHs for many objective optimisation of flight dynamic control. The design problem is posed to find control gains as design variables. The objective functions are minimising the control effort, the spiral root, the damping in roll root, sideslip angle deviation, and maximising the damping ratio of the Dutch-roll complex pair, the Dutch-roll frequency, bank angle at pre-specified times 1 seconds and 2.8 second subjected to several constrained based on Military Specifications (1969) requirement. A number of many-objective meta-heuristics (MnMHs) including multiobjective evolutionary algorithm based on decomposition (MOEA/D) [18], improved two-archive algorithm (Two_Arch2) [13], hybridisation of real-code population-based incremental learning and differential evolution (RPBILDE) [19] preference-inspired co-evolutionary algorithm using goal vectors (PICEA-g) [17], and knee point driven evolutionary algorithm for many-objective optimization (KnEA) [16] are used to tackle the problem while C-indicator as detailed in [20] are used for performance assessment.

2. Flight dynamic control optimization problem
In this study, the optimisation problem of a flight control in lateral and directional modes is considered. The flight dynamic model used is the same as those presented in [21-23]. The motion of an aircraft can be described in three main axes; roll axis (longitudinal), Pitch axis (lateral) and Yaw as shown in Figure 1. The state equation represents the lateral/directional motion of an aircraft can be expressed as [21-23]:

\[ \dot{x} = Ax + Bu \]  

where \( x = [\beta, r, p, \phi]^T \)
- \( \beta \) = Sideslip
- \( r \) = Yaw rate,
- \( p \) = Roll rate,
- \( \phi \) = Bank angle,
- \( A \) = Kinetic energy matrix,
- \( B \) = Coriolis matrix,
- \( u \) = Control vector

The control vector, \( u \), can be expressed as;

\[ u = Cu_p + Kx \]

where \( u_p \) is a pilot’s control input vector while \( C \) and \( K \) are the gain matrices expressed as

\[
C = \begin{bmatrix} 1 & 0 \\ k_5 & 1 \end{bmatrix},
\]

\[
K = \begin{bmatrix} k_6 & k_1 & k_2 & 0 \\ k_7 & k_3 & k_4 & 0 \end{bmatrix}
\]

The parameters \( k_1-\cdots-k_7 \) are control gain coefficients which need to be determined.

Substitution eq. (2) into the eq. (1), the state equation for the lateral/directional motion of the aircraft is expressed as;

\[ \dot{x} = (A + BK)x + BCu_p \]
many-objective problem of the flight lateral/directional control is set to minimise; spiral root \( \lambda_s \), roll damping \( \lambda_r \), the control effort \( \sum_{i=1}^{7} k_i \), sideslip angle deviation \( \Delta \beta \), and maximise; damping ratio of Dutch-roll complex pair \( \xi_D \), Dutch-roll frequency \( \Omega_D \), bank angle at 1 and 2.8 seconds \( \phi(1) \) and \( \phi(2.8) \), subjected to several constraints for a fighter aircraft based on the Military Specifications (1969) requirement. The optimisation problem can be expressed as:

\[
\text{Min: } f(x) = \left\{ \lambda_s, \lambda_r, \sum_{i=1}^{7} k_i, \Delta \beta, \frac{1}{\xi_D}, \frac{1}{\omega_D}, \frac{1}{\phi(1)}, \frac{1}{\phi(2.8)} \right\}
\]

Subjected to:
\[
\begin{align*}
\lambda_s & \leq -0.01 \\
\lambda_r & \leq -3.75 \\
\xi_D & \geq 0.5 \\
\omega_D & \geq 1 \\
\phi(1) & \geq 90^\circ \\
\phi(2.8) & \geq 360^\circ \\
\Delta \beta & \leq 0.75^\circ
\end{align*}
\]

The parameter \( \lambda_s, \lambda_r, \xi_D \) and \( \Omega_D \) can be calculated based on the eigenvalues associated with the term of \( (A+BK) \) while \( \beta \) and \( \phi \) can be found from dynamic response. The design variables are control gains coefficient in the matrix \( K \) (\( x = \{ k1, k2, k3, k4, k5, k6, k7 \}^T \)). The kinetic energy matrix \( A \) and the Coriolis matrix \( B \) are defined as:

\[
A = \begin{bmatrix}
-0.2842 & -0.9879 & 0.1547 & 0.0204 \\
10.8574 & -0.5504 & -0.2896 & 0 \\
-199.8942 & -0.4840 & -1.6025 & 0 \\
0 & 0.1566 & 1 & 0
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0 & 0.0524 \\
0.4198 & -12.7393 \\
50.5756 & 21.6753 \\
0 & 0
\end{bmatrix}
\]

**Figure 1.** Three main axis of an aircraft.
3. Numerical Experiment

Five established meta-heuristics including, MOEA/D, Two-Arch2, RPBILDE, PICEA-g, and KnEA, are used to solve many-objective problems of the flight dynamic control problem described in Section 2. The MnMHs parameter settings used in this study (details of notations can be found in the corresponding references of each method) are given as [24]:

- MOEA/D: The algorithm is coded by authors of this paper based on the reference [18]. The number of neighbouring weight vectors, crossover and mutation probabilities are set as 6, 1.0, and 0.1 respectively.
- Two-Arch2: Use the code from H.Wang et al. [13]. The crossover probability and mutation probability are set to be 1 and 0.1, respectively.
- RPBILDE: The code is provided by the authors of [19]. Scaling factor, crossover probability, and probability of choosing element from offspring in crossover are set to be 0.8, 0.7 and 0.5, respectively.
- PICEA-g: Use the code from R.Wang et al. [17]. All optimization parameters such as SBX parameter, type of crossover, probability of crossover between a pair, probability of internal crossover, etc. are set as default values from [17].
- KnEA: used the default values and the parameter settings from X. Zhang et al. [16].

Each optimiser is used to find a Pareto optimal front of the problems as detailed in Section 2 for 10 optimisation runs. The population size is set to be 50 and the number of generations is set to be 200. For an MnMH using different population size, the search process will be terminated with the total number of function evaluations (FEs) equal to 50×200 FEs. In addition, as most of MnMHs used in this study is box-constrained many-objective optimiser, to deal with constraints, a penalty function is used. The function is expressed as follow;

\[ f_p(x) = (1 + \varepsilon_1 \cdot v)^{\varepsilon_2} \]

where \( f_p \) is a penalised function value, \( \varepsilon_1 \) and \( \varepsilon_2 \) are selected considering the exploration and exploitation rates of the search space.

\[ v = \sum_{i=1}^{n_c} v_i \]

where \( n_c \) and \( v_i \) are the number of constraints and constraints violation, respectively. The \( v_i \) value can be calculated as:

\[ v_i = \begin{cases} \max(\lambda_s + 0.01, 0) \\ \max(\lambda_R + 3.75, 0) \\ \max(0.5 - \xi_D, 0) \\ \max(1 - \omega_d, 0) \\ \max(90 - \phi(1), 0) \\ \max(360 - \phi(2.8), 0) \\ \max(\Delta \beta - 0.75, 0) \end{cases} \]

4. Results and discussions

After performing 10 optimisation runs of the five MnMHs on solving the flight dynamic control optimisation problem. The search performances are investigated based on the C-indicator. The C-indicator will compare a pair of optimisers which results in totally 10 pairs.

Figure 2 shows the box-plots of the C-values comparing of all optimisers. In the box-plots, the upper and lower lines show the maximum and minimum C-values while the middle line shows median of the C-values from 10 optimisation runs. The box-plot at row i and column j give the results of C (optimiser_i, optimiser_j) which is required to compare with C (optimiser_j, optimiser_i) whereas the
higher median of C the better. The quantitative assessment is given in Tables 1 where each value on row i and column j is the mean value of C (optimiser_i, optimiser_j). The comparison is evaluated in the same way as the box-plot comparison. For example, average C (MOEA/D, Two_arch2) = 0 while average C (Two_arch2, MOEA/D) = 0.041256. This means Two_arch2 is better than MOEA/D. From the table, it is found that Two_arch2 is the best optimiser while the others are ranked in the following sequences: (2) KnEA, (3) RPBILDE, (4) PICEA-g and (5) MOEA/D.

| MOEA/D       | 0  | 0.002816 | 0.0022 | 0.0018 |
|--------------|----|----------|--------|--------|
| 0.041256     | Two_Arch2 | 0.027603 | 0.004  | 0.0094 |
| 0.015269     | 0.0068 | RPBILDE  | 0.0028 | 0.007  |
| 0.0025       | 0    | 0.001021 | PICEA-g | 0.0006 |
| 0.010611     | 0.0014 | 0.008299 | 0.0022 | KnEA   |

Figure 2. Box plot of C-values

Table 1. Mean C-values of all MnMHs

5. Conclusion
Many-objective constrained optimisation of flight dynamic control has been proposed and five many-objective meta-heuristics are implemented to solve the problem. The comparative results shown that Two_Arch2 is the best performer based on C-indicator values. Future work will be directed to the development of a new powerful meta-heuristic for solving real application of flight dynamic and control.

Acknowledgement
Technical The authors are grateful for the support from the Thailand Research Fund (TRF).

References
[1] Socha K and Dorigo, M 2008 European Journal of Operational Research 185 1155
[2] Venter G.and Sobiesczanski-Sobieski J 2003 AIAA Journal 41 1583
[3] Storn R.and Price K 1997 Journal of Global Optimization 11 341
[4] Back T 1996 Evolutionary Algorithms in Theory and Practice (Oxford University Press: Oxford)
[5] Lindfield G and Penny J 1995 Numerical Methods Using MATLAB Ellis Horwood
[6] Baluja S 1994 Population-Based Incremental Learning: A Method for Integrating GeneticSearch Based Function Optimization and Competitive Learning In CMU-CS-94-163 t. r. Ed. (Carnagie Mellon University: Pittsburgh).
[7] Hwang C R 1988 Acta Applicandae Mathematicae 12 108.
[8] Deb K and Jain H 2012 In Evolutionary Computation (CEC), 2012 IEEE Congress on pp 1.
[9] Bader J and Zitzler E 2011 Evol. Comput 19 45.
[10] Bandyopadhyay S and Mukherjee A 2014 Evolutionary Computation, IEEE Transactions on 1
[11] Murata T and Taki A 2010 In 10th International Conference on Hybrid Intelligent Systems (HIS) 2010 pp 175
[12] Pholdee N Bureerat S and Yildiz A R 2017 International Journal of Vehicle Design 73 20
[13] Wang H, Jiao L and Yao X 2015 IEEE transactions on evolutionary computation 19 524
[14] Nuaekaew K, Artrit P, Pholdee N and Bureerat S 2017 Expert Systems with Applications 87 79
[15] Cheng R, Jin Y, Olhofer M, and Sendhoff B 2016
[16] Zhang X, Tian Y and Jin Y 2015 IEEE transactions on evolutionary computation 19 761
[17] Wang R, Purshouse R C and Fleming P J 2013 IEEE transactions on evolutionary computation 17 474
[18] Qingfu Z and Hui L 2007 IEEE Transactions on evolutionary computation 11 712
[19] Pholdee N. and Bureerat S 2013 Information Sciences 223 136
[20] Zitzler E, Deb K and Thiele L 2000 Evolutionary Computation 8 173.
[21] Tabak D, Schy A, Giesy D and Johnson K 1979 Automatica 15 595
[22] Adra S F, Hamody A I, Griffin I and Fleming P J A 2005 In IEEE Congress on Evolutionary Computation, 2005. vol 1 pp 1
[23] Rostami S and Neri, F 2016 Integrated Computer-Aided Engineering 23 313
[24] Pholdee N, Bureerat S, Jaroenapibal P and Radpukdee T 2017, In International Symposium on Neural Networks Springer pp 143.