AIAA JOURNAL
Vol. 49, No. 7, July 2011

Swarm Heuristic for Identifying Preferred Solutions in Surrogate-Based Multi-Objective Engineering Design

Robert Carrese*
Royal Melbourne Institute of Technology, Bundoora, Victoria 3083, Australia
András Sóbester†
University of Southampton, Southampton, England SO17 1BJ, United Kingdom
and
Hadi Winarto‡ and Xiaodong Li§
Royal Melbourne Institute of Technology, Bundoora, Victoria 3083, Australia

DOI: 10.2514/1.J050819

Exploring the entire Pareto frontier of high-fidelity multidisciplinary problems can be prohibitive due to the excessive number of expensive evaluations required. The use of surrogate models offers promise toward managing such problems, which are restricted by a computational budget. In this paper, the kriging-assisted user-preference multi-objective particle swarm heuristic is presented, in which less accurate but inexpensive surrogate models are used cooperatively with the precise but expensive objective functions to alleviate the computational burden. A user-preference module is integrated into the optimization framework, which guides the swarm toward preferred regions of the Pareto frontier, thereby focusing all computing effort on identifying only solutions of interest to the designer. While providing a logical criterion to prescreen candidates for precise evaluation, the additional guidance provided by user-preferences guarantees an accelerated convergence rate. To depict the proficiency of the proposed framework, a suite of test problems, including the multidisciplinary cross-sectional design of a semimonocoque fuselage enclosing a pressurized cabin and payload bay, is presented. A parametric model is described that is capable of generating a broad range of double-lobe fuselage designs. The superiority of the kriging-assisted user-preference multi-objective particle swarm optimization algorithm over more traditional search methods to efficiently manage high-fidelity discontinuous design problems is highlighted.

Nomenclature

C	=	scaling coefficient
CAR	=	cargo lobe
c	=	constraint array
d_z	=	reference point distance
$\hat{d}_z$	=	kriging predicted reference point distance
f	=	fitness array
$\hat{f}$	=	kriging fitness
m	=	number of objectives
N	=	training dataset sample size
$N_{1,2}$	=	class function exponents
N_s	=	swarm population
n	=	number of decision variables
PAX	=	passenger lobe
$\mathbf{p}_g$	=	population global best position vector
$\mathbf{p}_i$	=	particle personal best position vector
Q	=	elitist archive
R	=	lobe radius
S	=	feasible design space
$\hat{s}$	=	kriging prediction error
t	=	time-step
$\mathbf{v}$	=	particle velocity vector
$\mathbf{X}$	=	training dataset sample position
$\mathbf{x}$	=	particle position vector
$\mathbf{Y}$	=	training dataset sample fitness
$\hat{y}$	=	kriging prediction
y, z	=	fuselage cross-sectional coordinates
$\bar{z}$	=	reference point
δ	=	preferred region solution spread
$\hat{\theta}$	=	correlation parameter
σ_v	=	von Mises stress

Received 2 August 2010; revision received 31 January 2011; accepted for publication 7 February 2011.

*Ph.D. Candidate, School of Aerospace Mechanical and Manufacturing; robert.carrese@rmit.edu.au. Student Member AIAA.

†Lecturer, Computational Engineering and Design; a.sobester@soton.ac.uk. Senior Member AIAA.

‡Associate Professor, School of Aerospace Mechanical and Manufacturing; hadi.winarto@rmit.edu.au. Senior Member AIAA.

§Senior Lecturer, School of Computer Science and Information Technology; xiaodong.li@rmit.edu.au.

I. Introduction

IN RECENT times, the usage of computational models in engineering design has greatly increased [1,2]. Engineers employ high-fidelity numerical modeling to simulate, within reasonable accuracy, how a complex system will behave. Furthermore, the ability to reflect changes in the behavior of a system by modifying certain input parameters has prompted the use of optimization techniques. Innovative design methods have since been developed, e.g., by drawing on the Darwinian model of the survival of the fittest, the ability of a flock of birds to move in unison to avoid a predator, or the ability of a neural network to detect patterns in complex input-output data relationships, etc. These *learning* techniques are formulated to find the most optimal system configuration to suit the preferences of the designer, in the shortest time possible. However, a significant challenge to the application of evolutionary heuristics in engineering design is the excessive number of expensive function evaluations required for convergence [1]. Even with the cooperative use of a surrogate model, the computational expense may still intensify when considering multiple competing objectives, where a host of solutions are possible. Therefore, control measures are proposed to ensure efficient frameworks that focus on identifying preferred solutions only.

A multi-objective optimization problem (MOP) follows the generic form:

$$\min_{\mathbf{b}\in S} f(\mathbf{x}) = \{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\} f_i\colon S \to \mathbb{R} \tag{1}$$

where $\mathbf{x}$ represents a decision vector of n inputs $\mathbf{x} = \{x_1, \ldots, x_n\}$ subject to the design space $S \in \mathbb{R}^n$ bounded by $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$, and any additional constraints. The vector f represents m conflicting objectives. Solutions to MOP are globally nondominated or Pareto-optimal, each offering a specific level of compromise between the objectives. The simplest approach to identify one possible solution on the Pareto front is to adopt the weighted sum approach, where all objectives are aggregated into a single scalar through weights [3,4]. Such methodologies allow for the use of proven optimization tools, such as the deterministic gradient-based methods. However, the weight terms are generally not known in advance, and it provides no flexibility to the designer to screen through alternative solutions. Furthermore, complete equivalence to an actual Pareto-optimal solution is not necessarily guaranteed [5]. Alternatively, we propose the use of a multi-objective particle swarm algorithm [5,6], a recent addition to the list of evolutionary multi-objective techniques [7]. This population-based heuristic is capable of identifying a host of nondominated solutions, providing the designer greater flexibility in selecting the most appropriate solution. The swarm heuristic is praised for its proficiency in managing discontinuous and multimodal problems, as well as its quick and simple implementation [5,8].

Despite these advantages, evolutionary algorithms may only deliver marginal performance for computationally challenging problems since they are not guided by any gradient information [1,9]. We argue that in most engineering applications, to explore the entire Pareto front is often unnecessary and the computational burden can be alleviated by incorporating the preferences of the designer. Recently, there has been increased interest in coupling classical interactive methods to population-based heuristics as an intuitive way of specifying user-preferences [10–12]. We propose the integration of one such interactive algorithm as a guidance mechanism for the swarm. A reference point, which is specified as an array of m aspiration values, is projected onto the Pareto landscape by the designer to guide the swarm toward solutions of interest. Unlike goal attainment methods, which make explicit reference to a target design [13], the reference point is a means of expressing the designer's preferred level of compromise, which can ideally be based on an existing or target design [14–16]. The swarm is guided by this information to confine its search to the preferred region of the Pareto front in the vicinity of the reference point. By establishing a preferred region (or solution spread), the designer is still provided with the flexibility to explore other interesting alternatives, which perhaps slightly deviate from the specific level of compromise as dictated by the reference point.

The integration of the reference point method has proven to overcome many limitations that plague conventional evolutionary algorithms, including many-objective problems [16,17]. Of particular significance is the ability to converge over large multimodal design spaces (which are typical of engineering design problems) and precision in the exploitation of individual solutions [14,15]. In this paper, we aim to progress further with the concept of user-preference optimization, by exploring the concept of surrogate-assisted optimization (SAO). Surrogate or metamodels are used to predict the response of a computationally intensive function at an unobserved location, based on observations at nearby locations. The aim of SAO is to use the precise objective functions and the inexpensive surrogate models cooperatively, in an effort to reduce the number of precise evaluations required for convergence. An active area of research is the development of prescreening criteria which determine, with sufficient confidence, which candidate designs are feasible for precise evaluation [2,18,19]. Such prescreening strategies are to be implemented with caution, so that a fair balance between searching less explored regions of the design space and exploitation of promising areas can be established. We demonstrate the relative ease in prescreening candidate designs for precise evaluation, by using the information provided by the reference point. This simple, yet logical, criterion is proficient in identifying solutions which are expected to provide improvement within the preferred region.

The paper is structured as follows. Section II provides a description of the user-preference multi-objective particle swarm algorithm, and the rationale for selecting leaders. Section III introduces surrogate modeling and describes a novel user-preference prescreening criterion. Section IV highlights the computational efficiency of the proposed framework for a high-fidelity design problem, compared with more traditional methods. Finally in Sec. V, conclusions are presented and avenues for future research are explored.

II. User-Preference Particle Swarm Algorithm

Particle swarm optimization (PSO) is perhaps the most widely researched swarm paradigm and was first introduced by Kennedy and Eberhart [6]. The PSO architecture is derived from the social-psychological tendency of individuals to learn from previous experience and emulate the success of others. Particles are represented by n-dimensional vectors $\mathbf{x}_i$ and $\mathbf{v}_i$, which are the particle position and velocity, respectively. From the performance rating provided by the objective solver, particles identify and exploit promising areas of S via coordinated movement. Although PSO was initially developed for single-objective optimization, it has since gained rapid popularity in multi-objective optimization [8].

There have been numerous modifications to the canonical particle swarm algorithm, which affect certain search characteristics [5]. Of significant importance is the swarm topology, which controls the level of communication between swarm particles. Recent research suggests that a local topology is more likely to overcome premature convergence so that particles do not commit prematurely to an inferior optimum [20]. Individual particles have limited global communication and the concept of a single global leader is replaced by a subset of one to many neighborhood leaders, to encourage diversity. We present a variant of the multi-objective particle swarm (MOPSO) heuristic, which incorporates user-defined preferences to direct all computing effort on preferred regions of the Pareto landscape. Implementing user-preferences allows us to adopt a more diverse search tactic, by ranking solutions according to their resemblance to the ideal compromise.

A. Particle Flight

The ith particle of the swarm is accelerated toward its personal best position $\mathbf{p}_i$ and the global (or neighborhood) best position $\mathbf{p}_g$. The particle velocity magnitude is initialized randomly in the interval $[\mathbf{x}_{\min} - \mathbf{x}_{\max}, \mathbf{x}_{\max} - \mathbf{x}_{\min}]$. The updated position and velocity vectors at time $t + 1$ are given by the following two equations [21]:

$$\mathbf{v}_{i,(t+1)} = \chi[\mathbf{v}_{i,(t)} + \mathbf{R}_1[0, \varphi_1] \otimes (\mathbf{p}_{i,(t)} - \mathbf{x}_{i,(t)}) + \mathbf{R}_2[0, \varphi_2] \otimes (\mathbf{p}_{g,(t)} - \mathbf{x}_{i,(t)})] \tag{2}$$

$$\mathbf{x}_{i,(t+1)} = \mathbf{x}_{i,(t)} + \mathbf{v}_{i,(t+1)} \tag{3}$$

where $\mathbf{R}_1[0, \varphi_1]$ and $\mathbf{R}_2[0, \varphi_2]$ are two functions returning a vector of uniform random numbers in the range $[0, \varphi_1]$ and $[0, \varphi_2]$, respectively. The constants φ_1 and φ_2 are set to $\varphi/2$ where $\varphi = 4.1$. The constriction factor χ applies a dampening effect as to how far the particle explores within the search space, given as $\chi = 2/|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|$. To avoid saturation of the search space boundary, in the event a particle leaves the search space, it is reflected in the opposite direction by the magnitude of confinement violation.

B. Topology

In a multi-objective environment, there are generally a host of nondominated solutions that are all considered equally optimal. At each time-step, the best representative front found by the particles is stored within an elitist archive. As the search progresses, nondominated solutions in the archive are updated and/or removed via the dominance criteria. Since there are now several identifiable *leaders* for each particle, additional guidance in ranking the archive solutions is necessary. Following the work of Wickramasinghe and

Li [10], a user-preference module is integrated to focus on preferred regions of the Pareto landscape. This module provides an intuitive criterion for selecting candidates for leadership, and assists the swarm to identify only solutions of interest to the designer. The guidance mechanism takes the form of a reference point $\bar{z}$, which is used to construct a distance metric to be minimized for $\mathbf{x} \in S$

$$\text{minimize } \; d_z = \max_{i=1:m}\{(f_i(\mathbf{x}) - \bar{z}_i)\} \tag{4}$$

where $\bar{z}_i$ is the ith component of the reference point or the aspiration value to the ith objective. The designer generally has no prior knowledge of the Pareto front, therefore reference points may be ideally placed in any feasible or infeasible region, as shown in Fig. 1. The reference point draws on the designer's experience to express a feasible compromise, rather than specific target values or goals. Similarly, the reference point distance metric ranks or assesses a particle's success using one single scalar, instead of an array of objective values. We apply the dominance criteria concurrently with Eq. (4) to find a feasible set of nondominated solutions with the most resemblance to $\bar{z}_i$, in terms of compromise.

At each population update, the archive members are first sorted based on the metric d_z, of which the highest ranking solutions are selected as candidates for leadership. Each individual particle is then assigned randomly to an archive member as the allocated leader [see Eq. (2)]. Although this procedure does not follow the canonical local topology [20], it promotes search diversity and provides the necessary selection pressure for particles to converge toward the preferred *region* around the reference point, rather than a single point. To maintain a high selection pressure on the archive members, a limited number of solutions are permitted for entry. If the archive limit is breached, lowest ranked solutions are removed. The solution spread along the Pareto front is controlled by δ, as shown in Fig. 1. This parameter is the maximum variance of the solutions' distance metric $\sigma(d_z)$. The extent of the solution spread is directly proportional to δ and evidently, as the value of δ increases, the influence of the location of $\bar{z}$ diminishes. Once the condition $\sigma(d_z) \leq \delta$ is satisfied, the algorithm uses a crowding distance operator to maintain a uniform spread [7,22].

C. Handling Constraints

When comparing particles for admission in the archive, a constraint-dominance procedure is applied following the work of Deb [7]. For each particle, $c_i = (c_1, \ldots, c_p)$ where p is the number of constraints and $c_i \geq 0$ is the violation of the constraint. A solution $\mathbf{x}$ constraint-dominates $\mathbf{y}$ if any of the following criteria are met:

1) Solution $\mathbf{x}$ is feasible and solution $\mathbf{y}$ is not.
2) Both solutions $\mathbf{x}$ and $\mathbf{y}$ are infeasible but $c(\mathbf{x}) < c(\mathbf{y})$.
3) Both solutions $\mathbf{x}$ and $\mathbf{y}$ are feasible but $d_z(\mathbf{x}) < d_z(\mathbf{y})$.

Therefore, if both solutions are deemed to satisfy the constraint values, the more *preferred* particle is admitted for entry. However, if both particles are infeasible, the particle with the overall least number/value of constraint violations is considered the better solution.

D. Mutation Operator

Despite the additional guidance provided by user-preferences, and the diversity inherent within the proposed topology, the search proficiency of the swarm may deteriorate when confronted with a highly multimodal problem [8,23]. A Gaussian mutation operator is applied to archive solutions if consistent improvement is not recorded.¶ The mutation operator is both effective at initially bypassing poorly performing areas of the design space, and generating new leaders as the search stagnates, or $\mathbf{v} \to 0$. The percentage of archive members selected for mutation steadily reduces as the archive reaches maximum capacity.

¶Improvement is measured by monitoring the mean value of d_z. If successive mean values are equal, mutation is triggered.

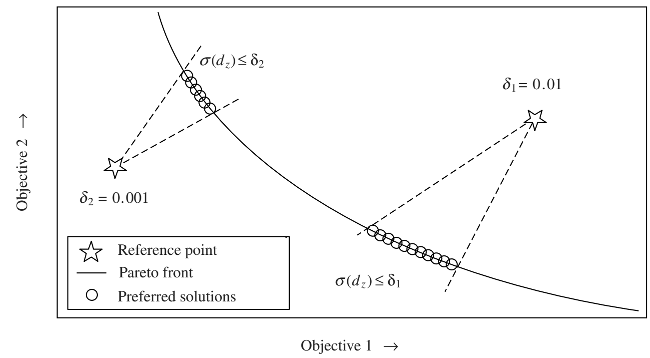


Fig. 1 User-preference particle swarm algorithm on a convex Pareto landscape.

E. UP-MOPSO Algorithm

The user-preference multi-objective particle swarm optimization (UP-MOPSO) algorithm is summarized in Fig. 2. The stopping criterion for the algorithm is based on the maximum number of function evaluations as specified by the user.

UP-MOPSO is the developed optimizer for our surrogate-assisted framework. We next describe how surrogates are implemented in this algorithm to manage problems constrained by a computational budget.

III. Surrogate-Assisted Optimization

Evolutionary optimization techniques are very reliable in obtaining a representation of the Pareto front for MOP. However, in modern engineering design, the objective array can be computationally demanding and thus the success of the optimization process is also dependent on the computational resources available. The cooperative use of a surrogate with the precise objective function offers promise toward managing design problems which are restricted by a computational budget.

A. Kriging Prediction

In most engineering problems, to construct a globally accurate surrogate of the original objective landscape is improbable due to the weakly correlated design space. It is more common to locally update the prediction accuracy of the surrogate as the search progresses toward promising areas of the design space [2]. For this purpose, the kriging method has received much interest, because it inherently considers confidence intervals of the predicted outputs. For a complete derivation of the kriging method, the readers are encouraged to follow the work of Jones [24] and Forrester et al. [2]. We provide a very brief introduction to the ordinary kriging method, which expresses the unknown function $y(\mathbf{x})$ as

$$y(\mathbf{x}) = \beta + z(\mathbf{x}) \tag{5}$$

where $\mathbf{x} = [x_1, \cdots, x_n]$ is the data location, β is a constant global mean value, and $z(\mathbf{x})$ represents a local deviation at the data location $\mathbf{x}$ based on a stochastic process with zero-mean and variance σ^2 following the Gaussian distribution. The approximation $\hat{y}(\mathbf{x})$ is obtained from

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{1}\hat{\beta}) \tag{6}$$

where $\hat{\beta}$ is the approximation of β, $\mathbf{R}$ is the correlation matrix, $\mathbf{r}$ is the correlation vector, $\mathbf{Y}$ is the training dataset of N observed samples at location $\mathbf{X}$, and $\mathbf{1}$ is a column vector of N elements of 1. The correlation matrix is a modification of the Gaussian basis function

$$R(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\sum_{k=1}^{n} \theta_k |x_k^i - x_k^j|^2\right) \tag{7}$$

where $\theta_k > 0$ is the kth element of the correlation parameter $\boldsymbol{\theta}$. Following the work of Jones [24], the correlation parameter $\boldsymbol{\theta}$ (and

1. Specify reference point $\bar{z}$ and solution spread δ
2. Initialize the swarm population:
   (a) The position $\vec{x}_i$, velocity $\vec{v}_i$ and personal best position $\vec{p}_i = \vec{x}_i$
   (b) Evaluate each particle in the population; time-step $t := 0$.
3. Time-step $t := t + 1$.
4. Apply domination criteria to update archive, Q:
   (a) Calculate the distance metric d_z (via Eq. (4)) of each swarm particle.
   (b) Identify non-dominated solutions and include in Q; **if** $\sigma(d_z) \le \delta$ **then** rank Q by min d_z, **else** rank Q by crowding distance.
   (c) **if** Q limit is breached, **if** $\sigma(d_z) \le \delta$ **then** rank Q by min d_z, **else** rank Q by crowding distance; lowest ranked members are removed.
5. Update swarm population:
   (a) Randomly assign highest ranked members of Q to particles.
   (b) Update particle position $\vec{x}_i$ and velocity $\vec{v}_i$ as per Eqs. (2) and (3); evaluate each particle in the population.
6. Apply mutation operator if consistent improvement in Q is not recorded.
7. Update $\vec{p}_i$ if dominates existing $\vec{p}_i$.
8. if **not** maximum number of evaluations reached **then goto** 3.

Fig. 2 UP-MOPSO pseudocode.

hence the approximations $\hat{\beta}$ and $\hat{\sigma}^2$) are estimated by maximizing the concentrated ln-likelihood of the dataset $\mathbf{Y}$, which is an n-variable single-objective optimization problem, solved using a quasi-Newton method [25]. The accuracy of the prediction $\hat{y}$ at the unobserved location $\mathbf{x}$ depends on the correlation distance with sample points $\mathbf{X}$. The closer the location of $\mathbf{x}$ to the sample points, the more confidence in the prediction $\hat{y}(\mathbf{x})$. The measure of uncertainty in the prediction is estimated as **

$$\hat{s}^2(\mathbf{x}) = \hat{\sigma}^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right] \tag{8}$$

B. Reference Point Prescreening Criterion

Training a kriging model from a training dataset is time-consuming and is $\mathcal{O}(N^3)$. Stratified sampling using a maximin Latin hypercube (LHS) methodology [2,26] is used to construct a global kriging approximation $[\mathbf{X}, \mathbf{Y}]$. The nondominated subset of $\mathbf{Y}$ is then stored within the elitist archive (see Sec. II.B). This ensures that candidates for global leadership have been precisely evaluated (or with negligible prediction error) and, therefore, offer no false guidance to other particles. Adopting the concept of individual-based control [27], kriging predictions are then used to prescreen each candidate particle after the population update (or after mutation) and subsequently flag them for precise evaluation or rejection. The kriging model estimates a lower-confidence bound (LCB) for the objective array as

$$\{\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_m(\mathbf{x})\}_{\mathrm{lb}} = [\{\hat{y}_1(\mathbf{x}) - \omega\hat{s}_1(\mathbf{x})\}, \ldots, \{\hat{y}_m(\mathbf{x}) - \omega\hat{s}_m(\mathbf{x})\}] \tag{9}$$

where $\omega = 2$ provides a 97% probability for $\hat{f}_{\mathrm{lb}}(\mathbf{x})$ to be the lower bound value of $\hat{f}(\mathbf{x})$. An approximation to the reference point distance, $\hat{d}_z(\mathbf{x})$, can thus be obtained using Eq. (4). This value, while providing a means of ranking each solution as a single scalar, also gives an estimate to the improvement that is expected from the solution. At time t, the archive member with the highest ranking according to Eq. (4) is recorded as $d_{\min}$. The candidate $\mathbf{x}$ may then be accepted for precise evaluation, and subsequent admission into the archive if $\hat{d}_z(\mathbf{x}) < d_{\min}$. Particles will thus be attracted toward the areas of the design space which provide the greatest resemblance to $\bar{z}$ and the direction of the search will remain consistent.

As the search begins in the explorative phase and the prediction accuracy of the surrogate model(s) is low, depending on the deceptivity of the objective landscape(s) there will initially be a large percentage of the swarm that is flagged for precise evaluation. Subsequently, as the particles begin to identify the preferred region and the prediction accuracy of the surrogate model(s) gradually increases, the prescreening criterion becomes increasingly difficult to satisfy, thereby reducing the number of flagged particles at each time-step. To restrict saturation of the dataset used to train the kriging models, a limit is imposed of $N = 200$ sample points where lowest ranked solutions according to Eq. (4) are removed.

C. Kriging UP-MOPSO Algorithm

To allow for the introduction of kriging and the corresponding reference point criterion, several modifications must be made to the original pseudocode, as shown in Fig. 2. The kriging UP-MOPSO, or K-UP-MOPSO, algorithm is summarized in Fig. 3.

D. Schaffer Test Function

Presented here are the results for the inexpensive convex Schaffer problem [19]. This mathematical test function is a multidimensional two-objective minimization problem of the form:

$$f_1(\mathbf{x}) = \frac{2}{\gamma}\cdot\left(\sum_{i=1}^{d}\mathbf{x}_i^2\right)^{\gamma/2} \tag{10}$$

$$f_2(\mathbf{x}) = \frac{2}{\gamma}\cdot\left[\sum_{i=1}^{d}(1 - \mathbf{x}_i)^2\right]^{\gamma/2} \tag{11}$$

**If $\mathbf{x} \subset \mathbf{X}$, it is observed from Eq. (8) that $\hat{s}(\mathbf{x})$ reduces to zero.

Table 1 Optimization results for the convex Schaffer test function

Algorithm	Function evaluations	$\min(d_z)$	$\mathrm{var}(d_z)$
NSGA-II	1000	7.6201 ± 1.8600	–
NSGA-II	5000	0.5371 ± 0.2823	–
UP-MOPSO	1000	0.1742 ± 0.0274	–
UP-MOPSO	5000	0.0553 ± 0.0009	0.0159 ± 0.0013
K-UP-MOPSO	150	0.0521 ± 0.0003	0.0152 ± 0.0002

where the design space range is $\mathbf{x} \in [0, 10]^d$, the number of dimensions $d = 10$ and $\gamma = 2$. The curvature of the Pareto front is scalable by the parameter γ. The Pareto front follows the equation

$$y_2 = (1 - y_1^{1/\gamma})^\gamma \qquad y_1 \in [0, 1] \tag{12}$$

The Schaffer test function is not characterized as deceptive; though a random initial population tends to be reasonably far from the global Pareto front, which generally prolongs the explorative phase of the search. By focusing exclusively on the preferred region and constructing a kriging model for each objective, the time spent in exploration and hence the number of function evaluations can be significantly reduced. Simulations are performed with a population of 100 individuals using the UP-MOPSO algorithm, the benchmark nondominated sorting genetic algorithm (NSGA-II) by Deb et al. [28] and the K-UP-MOPSO algorithm. For the latter, a kriging model is constructed for each objective based on a LHS sample of $N = 50$ design points. The reference point is selected as $\bar{z} = [0.2, 0.2]$ with a solution spread of $\delta = 0.015$. With no broadly applicable stopping criterion for multi-objective optimization, all algorithms stopped after a specified number of function evaluations. Results for the NSGA-II and the UP-MOPSO algorithms are recorded after 1000

1. Specify reference point $\bar{z}$ and solution spread δ
2. Construct global Kriging approximations
   (a) Construct global Kriging models [**X**, **Y**] for each objective/constraint where required using LHS of N samples.
   (b) Calculate d_z of each sample point [**X**, **Y**].
   (c) The non-dominated sample points of **Y** are stored within the archive, Q; rank Q by min d_z
3. Initialize the swarm population:
   (a) Initialize position $\vec{x}_i$, velocity $\vec{v}_i$ and personal best position $\vec{p}_i = \vec{x}_i$
   (b) Evaluate each particle in the population using Kriging; time-step $t := 0$.
4. Time-step $t := t + 1$.
5. Update swarm population:
   (a) Randomly assign highest ranked members of Q to particles.
   (b) Update $\vec{x}_i$ and $\vec{v}_i$; evaluate each particle in the population using Eq. (9).
   (c) Calculate the estimated lower bound reference point distance $\hat{d}_{z,\,i}$
6. Update Kriging models via pre-screening strategy
   (a) For the i-th particle: **if** $\hat{d}_{z,\,i} <$ highest ranked member of Q **then** precisely evaluate particle and update Kriging datasets ($N := N + 1$), **else** reject particle.
   (b) Update Kriging dataset [**X**,**Y**] and re-calculate d_z of each sample point.
   (c) If the sample size $N > 200$ design points, lowest ranked solutions according to d_z are removed.
7. Apply domination criteria to update archive, Q:
   (a) Identify non-dominated swarm particles with negligible error ($\hat{s}_i \to 0$) and include in Q.
   (b) Identify non-dominated solutions from training datasets and include in Q; **if** $\sigma(d_z) \le \delta$ **then** rank Q by min d_z, **else** rank Q by crowding distance.
   (c) **if** Q limit is breached, **if** $\sigma(d_z) \le \delta$ **then** rank Q by min d_z, **else** rank Q by crowding distance; lowest ranked members are removed.
8. Apply mutation operator if consistent improvement in Q is not recorded (mutated particles are precisely evaluated as per step 6).
9. Update personal best position if dominates existing personal best. $\vec{p}_i$ is recorded as the position which provided the minimal prediction error $\hat{s}_i$.
10. **if not** maximum number of evaluations reached **then goto** 4.

Fig. 3 K-UP-MOPSO pseudocode.

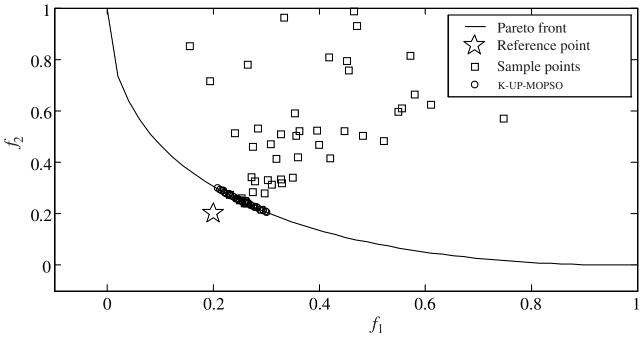


Fig. 4 K-UP-MOPSO on Schaffer function showing sample points in the immediate vicinity.

and 5000 function evaluations. For the K-UP-MOPSO algorithm, results are recorded after 150 function evaluations.

Statistical results (mean $\pm$ standard deviation) of 50 independent simulations are shown in Table 1. The value of the solution with minimum reference distance (i.e., $\min(d_z)$) is indicative of the closeness of the nondominated solution set to the global Pareto front. The K-UP-MOPSO consistently provides a more accurate solution set over the standard UP-MOPSO algorithm and NSGA-II algorithm at approximately 2% of the computational cost (the number of function evaluations refers to the sample points plus the additional update points). Results for the variance (i.e., $\mathrm{var}(d_z)$) of the nondominated solution set are also recorded; this variance is a measure of the solution spread. Since the NSGA-II algorithm explores the full extent of the Pareto front, the variance is not recorded. Larger values of the variance suggest a nonuniform scattering of solutions, while values that approach δ indicate the desired solution spread is attained.

It is clearly observed from Fig. 4 that the K-UP-MOPSO algorithm identifies a set of Pareto-optimal solutions in the immediate vicinity of the reference point. Also shown are selected sample points used to update the kriging models. The proficiency of the reference point prescreening criterion is evident from the sample point attraction toward the preferred region. This establishes a distinct search direction which is beneficial to prescreen prospective swarm particles to determine whether they are feasible for precise evaluation. It is also observed that only few precisely evaluated solutions reside on the Pareto front. This information is sufficient to predict a uniform spread of solutions on the Pareto front with negligible error.

E. Design of a Helical Compression Spring

For engineering design problems, the selection and use of a reference point to identify preferred solutions is more intuitive since it can reflect the designer's own preferred compromise. Presented in this section is a multi-objective constrained problem which was first proposed by Tudose and Jucan [29]. They solved it using a multi-objective genetic algorithm; a subsequent solution, using surrogate models, is described by Forrester et al. [2]. A helical compression spring is to be designed to work over a stroke of $h = 50$ mm with a corresponding load variation between $F_{\min} = 40$ N and $F_{\max} = 500$ N. ASTM A229/SAE J315 oil tempered wire is used with modulus of elasticity $E = 2.06 \times 10^5$ MPa, density $\rho = 7.87 \times 10^{-6}$ kg/mm^3, and rigidity modulus $G = 0.78 \times 10^5$ MPa. There are two conflicting objectives: 1) to minimize the mass, and 2) to maximize the fatigue life of the spring. Two constraints ensure that the spring does not fail in shear and in buckling, respectively. There are three design variables, whose ranges are shown in Table 2. The

Table 2 Ranges of variables for the design of a helical compression spring

Designation	Variable	Lower bound	Upper bound
1	d, mm	0.5	7
2	i	4	16
3	k_Δ	0.1	1.1

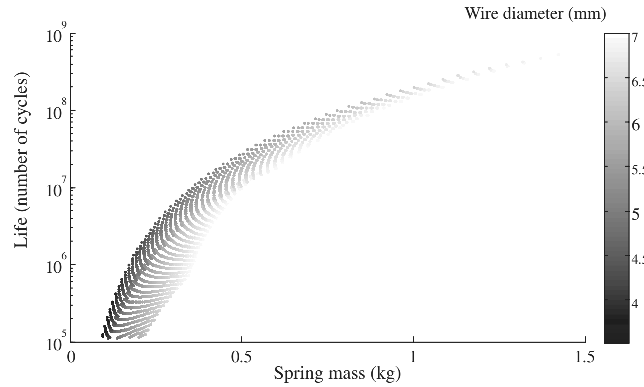


Fig. 5 Full factorial plan and the corresponding wire diameters of the feasible designs.

first is the wire diameter d. The second is the index i, defined as the ratio of the mean helix diameter (measured from the center of the wire) and the diameter d. The final variable is the maximum load intercoil distance coefficient k_Δ, which is the ratio of the distance between adjacent coils of the fully loaded spring and the diameter d.

Shown in Fig. 5 are the results of the evaluation of a 250,000 point full-factorial sampling plan. Only a marginal percentage (i.e., 66,139) of the designs have proven to be feasible, where they do not violate the two resistance constraints and they make geometrical sense (i.e., their variable triplets generate sensible springs). For a conventional (exhaustive) search, this would suggest a high number of unnecessary (or at least avoidable) evaluations are performed.

As the following results will indicate, the K-UP-MOPSO algorithm performs efficiently for this problem, in spite of the very large infeasible subdomain of the design space. Simulations are performed with a population of 100 individuals using the NSGA-II algorithm, the UP-MOPSO algorithm and the K-UP-MOPSO algorithm. For the latter, a kriging model is constructed for the objective and constraint functions based on a LHS sample of 20 design points, of which only 7 designs satisfy the constraints (note: designs that yield geometrically nonsensible springs are omitted). We specify the reference point as the ideal target design of 0.1 kg mass, and a life of 10^{10} cycles. The solution spread is specified as $\delta = 0.015$. For the NSGA-II and UP-MOPSO algorithms, results are recorded after 2000 function evaluations. For the K-UP-MOPSO algorithm, results are recorded after 50 function evaluations.

Statistical results (mean $\pm$ standard deviation) of 50 independent simulations are shown in Table 3. A representative simulation is shown in Fig. 6 compared with the Pareto front trend estimated from the full-factorial search of Fig. 5. The K-UP-MOPSO algorithm is observed to be far more proficient in comparison to the other algorithms, because it consistently obtains more accurate (and uniform) results at a fraction of the computational cost. Furthermore, the percentage of feasible solutions (i.e., sensible designs with no record of constraint violation) is sufficiently greater. Although the objectives and constraints are inexpensive to compute for this specific problem, if each experiment were a computer simulation or destructive test, the potential time and cost savings would be significant.

IV. Fuselage Cross-Sectional Design

As the main illustration of the K-UP-MOPSO algorithm, we shall consider the preliminary design of a semimonocoque fuselage enclosing a pressurized cabin and payload bay, using high-fidelity structural analysis. A fundamental design rule here is that any cross-sectional shape other than a circle is a stress compromise [30]. Any deviation from the circular shape forces the frames to carry a bending load (otherwise they are limited to maintaining the shape of the fuselage and breaking up the lengths of the longerons). Yet, competing drivers (primarily, the minimization of pressure drag by a reduction of cross-sectional area and the maximization of passenger comfort by increasing certain cabin dimensions) routinely demand other shapes.

Table 3 Optimization results for the helical spring design problem

Algorithm	Function evaluations	% Feasiblity	$\min(d_z)$	$\mathrm{var}(d_z)$
NSGA-II	2000	40.2 ± 0.5	0.2925 ± 0.0014	–
UP-MOPSO	2000	41.7 ± 0.7	0.2902 ± 0.0027	0.0152 ± 0.0023
K-UP-MOPSO	50	68.7 ± 0.2	0.2877 ± 0.0005	0.0151 ± 0.0002

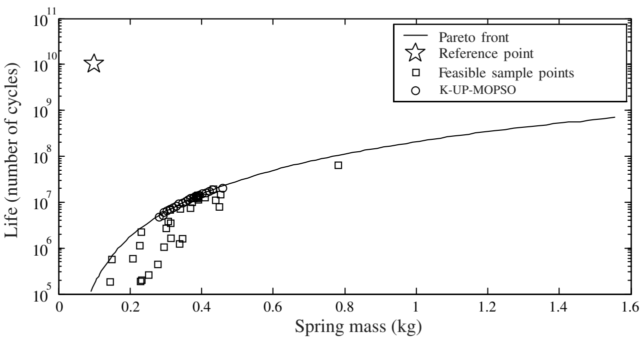


Fig. 6 K-UP-MOPSO on helical spring design problem showing feasible sample points.

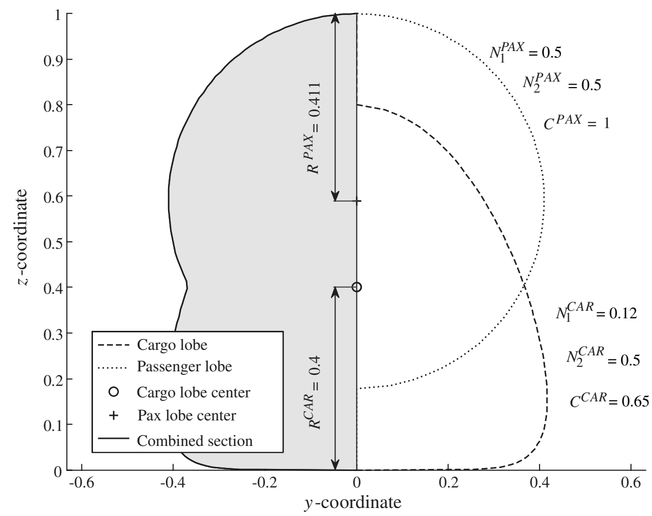


Fig. 7 Approximation to the wing-to-body fairing area of the Embraer E145.

The classic alternative is the two-lobe cross section. We refer to the upper lobe as the "passenger lobe" and the lower lobe as the "cargo lobe." The passenger cabin and the cargo bay are separated by the cabin floor, which carries tensile loads resulting from the pressurization, as well as the bending loads caused by the weight of the seats, passengers, etc. From a design optimization perspective we therefore need a parametric description capable of covering a broad range of two-lobe designs.

A. Features of a "Good" Parametric Geometry

Different practitioners may hold subtly different mental wish-lists under this heading, but these differences are usually in the ranking, rather than in the entries themselves. Here then, are the top three items on *a* list, colored, no doubt, by our own biases.

1) *Conciseness*: The cost of a conventional, black-box optimization process increases exponentially with the number of design variables. To tackle this "curse of dimensionality," one must limit the number of design variables; see Sóbester [31] for a more detailed treatment of this issue as well as a generic approach for building very low dimensionality geometries.

2) *Robustness*: The parametric geometry's ability, in terms of design space proportion, to yield physically and geometrically sensible shapes. A low degree of robustness will waste physics-based analyses on nonsensible designs during optimization. Robustness may be improved by restricting design variable domains, but regions of infeasibility are seldom rectangular, so this process may sacrifice flexibility.

3) *Flexibility*: This is the breadth of the range of shapes the parametric geometry is capable of generating. It is very hard to measure, and it is generally impossible to tell when a model has reached "sufficient" flexibility. This is because the necessary flexibility is determined by the vague and difficult to define concept of how "unusual" would a shape have to be for it to still be worth investigating.

For fuselage design, beyond the need to reproduce standard sections, generating *families* of objects with smoothly controllable deviations from circularity and multiple overlapping instances thereof in a concisely parameterized manner is advantageous. Such a parametric geometry would enable the Pareto analysis of the tradeoffs involved in deviations from circularity. As hinted earlier, these are usually driven by the competing goals of structural weight, drag minimization and comfort.[††]

[††]The study of the systematic description of such pseudocircular fuselage cross-section shapes goes back to the earliest days of computer aided design analysis [32].

B. Concise Cross-Section Model

Let us define the right-hand half of the generic airliner fuselage cross section in the y-z plane[‡‡] as the explicit function

$$y(z) = C_S \cdot \max[Y^{\mathrm{CAR}}(z), Y^{\mathrm{PAX}}(z)] + \Delta z, \qquad z \in [0, 1] \tag{13}$$

where the coordinates $y(z)$ are a composite depiction of the individual shape description[§§] of the cargo lobe (Y^{CAR}) and the passenger lobe (Y^{PAX})

$$Y^{\mathrm{CAR}}(z) = \begin{cases} C^{\mathrm{CAR}} z^{N_1^{\mathrm{CAR}}} (2R^{\mathrm{CAR}} - z)^{N_2^{\mathrm{CAR}}}, & z \in [0, 2R^{\mathrm{CAR}}] \\ 0 & \text{elsewhere} \end{cases} \tag{14}$$

$$Y^{\mathrm{PAX}}(z) = \begin{cases} C^{\mathrm{PAX}} [z - (1 - 2R^{\mathrm{PAX}})]^{N_1^{\mathrm{PAX}}} (1 - z)^{N_2^{\mathrm{PAX}}}, & z \in [1 - 2R^{\mathrm{PAX}}, 1] \\ 0 & \text{elsewhere} \end{cases} \tag{15}$$

In its most flexible form, the cross-sectional shape is described by ten variables, each with a clear and intuitive meaning. The upper (passenger) lobe and the lower (cargo) lobe have variable radii: R^{PAX} and R^{CAR}, respectively. The deviations from circularity are controlled by two exponents on each lobe: N_1^{PAX}, N_2^{PAX}, N_1^{CAR}, and N_2^{CAR}. Additional flexibility is enabled by a scaling coefficient on each lobe (C^{PAX} and C^{CAR}) (see Fig. 7 for an example). The section is normalized to a height of one; the coefficient C_S and the offset Δz help define the full size, correctly positioned section.

C. Cabin Design Constraints and Objectives

We seek to design a single-aisle fuselage cross section capable of accommodating 95 percentile U.S. male passengers seated six abreast, respecting the industry standard requirements in terms of aisle headroom, headroom under the overhead bins, overhead bin

[‡‡]We are viewing the section from the nose of the aircraft, with the vertical z-axis pointing upwards and the horizontal y-axis pointing to the port side.

[§§]The shape descriptions use class functions from Kulfan [33,34].

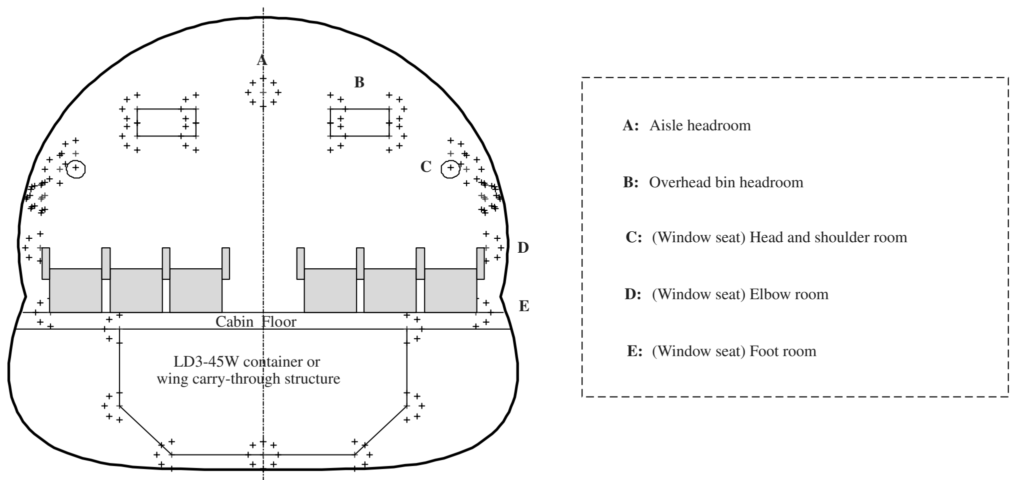


Fig. 8 Cabin space constraint points (denoted by + symbols) related to passenger space requirements based on the size of a 95 percentile U.S. male, as well as on the space needed for a fuselage frame of constant depth. The cargo lobe must accommodate the standard single-aisle container and/or the wing carry-through structure. A feasible cross-section shape is one that envelops all + symbols.

space, and window seat foot, shoulder, and headroom. We also seek to design the cargo bay to accommodate the standard LD3-45W container (see Fig. 8). We incorporate these conditions into a set of constraint points (marked by + symbols on Fig. 8), which must fall inside the section to ensure the satisfaction of all of these requirements. Formally, we define the design variable vector

$$\mathbf{x} = [R^{\mathrm{CAR}}, N_1^{\mathrm{CAR}}, N_2^{\mathrm{CAR}}, C^{\mathrm{CAR}}, R^{\mathrm{PAX}}, N_1^{\mathrm{PAX}}, N_2^{\mathrm{PAX}}, C^{\mathrm{PAX}}, W_S, W_A, W_{\mathrm{AR}}, H_B] \tag{16}$$

(see Table 4 for the definition and ranges of these variables) and, denoting the number of constraint points that fall *outside* the cross section with N_{out}, we set the constraint $N_{\mathrm{out}}(\mathbf{X}, C_S, \Delta Z) = 0$. The constraint is satisfied implicitly through the parametric model and is thus not handled directly by the optimizer, yielding a first-order discontinuity. As K-UP-MOPSO is inherently a population-based gradient-free approach, this approach is permissible.

For a given $\mathbf{x}$, we seek to wrap the constraint points as tightly as possible, so we obtain f_1, the cross-sectional (half) area as:

$$f_1(\mathbf{x}) = \min_{C_S, \Delta z} \int_0^1 y(z)\,\mathrm{d}z \quad \text{subject to } N_{\mathrm{out}} = 0 \tag{17}$$

where each evaluation of $f_1(\mathbf{x})$ requires a Nelder and Mead pattern search. The second objective f_2 is a measure of passenger comfort and we calculate it as:

$$f_2(\mathbf{x}) = 0.6W_{\mathrm{S}} + 0.1W_A + 0.1W_{\mathrm{AR}} + 0.2H_B \tag{18}$$

The third objective relates to the stress in the cabin structure. There are a variety of loads (e.g., bending loads, aerodynamic loads, etc.) that are dynamically applied to an aircraft fuselage in flight. Furthermore, the stress is nonuniformly distributed due to deviations in frame depth and circumferential size along various stations. We propose a more simplified stress analysis that provides a fairly sensible measure of the harshness of the stress-state of a candidate design. A two-dimensional finite element analysis is still required, which is a physics-based, high-fidelity analysis. Distributed loads are applied to the frame circumference and the floor to signify pressurization loads and weight loads, respectively.

A two-dimensional (linear static) finite element model is constructed using Nastran [35]. Numerical loading values are obtained from Niu [30] as 8.25 psi and 0.35 psi for the frame and floor, respectively. The I-beam and C-section, representing the floor and frame, respectively, are constructed with a series of beam elements of depth 0.125 m. The material is aluminum Al2024-T3 with modulus of elasticity $E = 7.5 \times 10^{10}$ Pa and Poisson ratio of $\nu = 0.33$. The model is analyzed to determine the von Mises stress σ_v

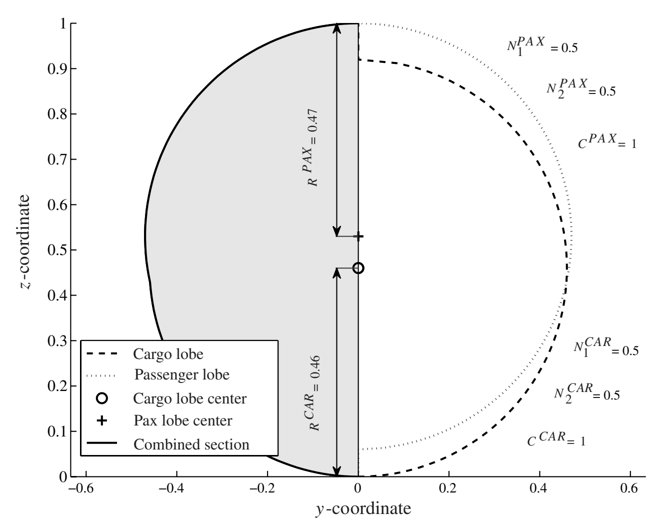


Fig. 9 Reference point 737-style fuselage cross section (normalized coordinates).

Table 4 The 12 cross-section definition design variables

Variable	Definition	Lower bound	Upper bound
R^{CAR}	See Eq. (14)	0.3	0.5
N_1^{CAR}	See Eq. (14)	0.1	0.5
N_2^{CAR}	See Eq. (14)	0.1	0.5
C^{CAR}	See Eq. (14)	0.5	1.2
R^{PAX}	See Eq. (15)	0.3	0.5
N_1^{PAX}	See Eq. (15)	0.1	0.5
N_2^{PAX}	See Eq. (15)	0.1	0.5
C^{PAX}	See Eq. (15)	0.5	1.2
W_{S}	Seat width	0	1
W_{A}	Aisle width	0	1
W_{AR}	Armrest width	0	1
H_{B}	Bin headroom	0	1

Table 5 Aspiration values for the reference point representative fuselage

Fuselage	Cross-sectional area, m^2	Comfort	Peak stress ($\times 10^9$ Pa)
Reference	6.9928	0.5	1.4908

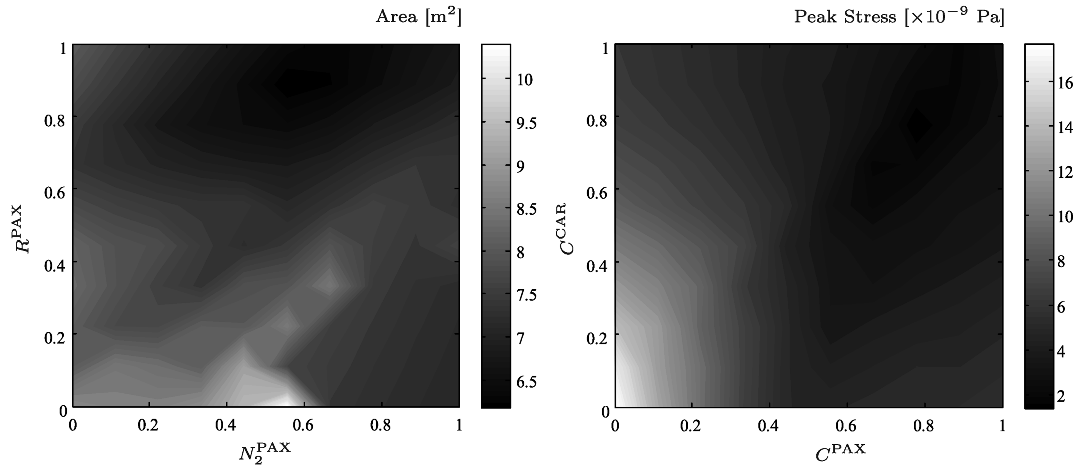
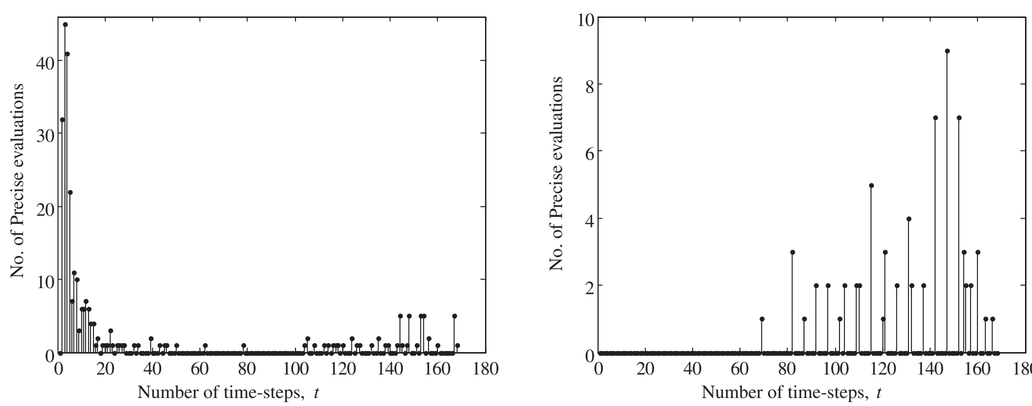


Fig. 10 Selected screening results for various parameters (normalized variable ranges).



a) Evaluations due to updates **b) Evaluations due to mutations**

Fig. 11 History of precise evaluations for the K-UP-MOPSO algorithm on the fuselage problem.

distribution around the frame. The simulation is completed by identifying the peak stress from the resultant stress distribution as the objective function, resulting in another discontinuity in the first-order. By minimizing the identified peak stress (where the location varies dependent on the cross-sectional shape), emphasis is indirectly placed on obtaining designs that exhibit a uniform stress distribution (i.e., circular cross sections)

$$f_3(\mathbf{x}) = \min\{\max[\sigma_v(y(z))]\} \tag{19}$$

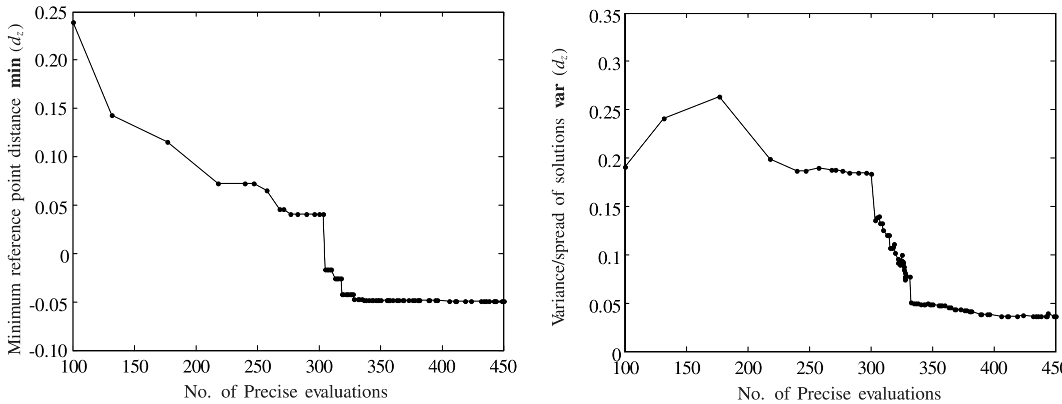
D. Numerical Example

To reiterate on the concept of using the K-UP-MOPSO algorithm to focus on and exploit our preferred compromise, we (as the designers) have selected our best approximation[¶¶] of the Boeing 737 fuselage cross section (see Fig. 9) as the reference point. This cross section has a slight hint of a double-lobe design where the cusp point coincides with the floor. The quasi-circular cross section implies a fairly uniform circumferential stress distribution while maintaining a relatively low cross-sectional area for reduced pressure drag. The aspiration values obtained from the representative cross section are given in Table 5. The K-UP-MOPSO algorithm will use these aspiration values to improve on the performance characteristics, while still maintaining a similar level of compromise. The solution spread is restricted to $\delta = 0.05$.

The allowable computational budget for this problem is directly related to the multimodality of the objective landscapes. The first-order discontinuity of the objectives places further limitations on the use of kriging models, since a larger number of sample points are required to accurately depict nonsmooth regions of the design space. Screening studies are a useful and simple technique to gain an understanding of the optimization landscape. Screening studies are performed by maintaining the baseline fuselage of the reference geometry, and then systematically varying two independent parameters to generate contour plots. Selected screening plots are shown in Fig. 10. The deceptive landscapes originate from the element of uncertainty in the location of the peak stress for each individual cross-sectional design, as well as the requirement of the cross section to enclose all constraint points. Adopting user-preferences, the kriging models can be entirely localized within the preferred region of the design space, thereby alleviating the irregularity of the landscape.

A swarm population of $N_s = 100$ particles was initialized; a common value used in previous optimization studies using MOPSO variants [22,23]. Since the majority of the swarm (at every population update) is calculated using the constructed kriging models, the population size is not an important consideration for the K-UP-MOPSO algorithm. The objective space was normalized for the calculation of the reference point distance by $(f_{\max} - f_{\min})$. A kriging model based on an LHS of $N = 100$ points was constructed for each objective. The accuracy of the global kriging models could be fairly low here as a result of possible discontinuities. However,

[¶¶]This case study is merely a demonstration of the algorithm on a realistic engineering design problem. We have only taken into account some of the key objectives and constraints here, so the actual results may be different from those obtained if the same exercise was conducted in an industrial setting.



a) History of most preferred solution **b) History of spread of solutions**

Fig. 12 Representation of the performance of K-UP-MOPSO for the fuselage problem.

since a user-preference module is adopted, the kriging models will focus entirely within the preferred region (as dictated by the prescreening criterion), providing a more accurate depiction of the localized area. We have thus imposed a computational budget of 450 precise evaluations, due to the first-order discontinuity.

A further 350 precise evaluations of the objective array were performed over $t \approx 170$ time-steps before the evaluation limit was breached. The additional 350 evaluations are composed of 280 particles and 70 mutations that satisfied the prescreening criterion, as shown in Fig. 11. It is observed from the stem-plot in Fig. 11a that the largest number of update points occur in the earlier stages of the search. This is due to the relatively poor prediction accuracy of the global kriging models. As the training datasets of the kriging models become localized within the identified preferred region, a reduced percentage of particles satisfy the prescreening criteria. This is opposite to the mutation stem-plot shown in Fig. 11b. As expected, during the explorative phase of the search, the set of nondominated solutions within the archive is continuously updating. As the search stagnates and there is less consistent improvement in the archive, particles are scheduled for mutation in an attempt to encourage further improvement.

Figure 12a features the progress of the solution with minimum reference point distance $d_{\min}$ as the number of precise evaluations escalates. For this simulation, 100 precise evaluations are required to reach within 65% of the most preferred solution*** and an additional 20 evaluations to reach within 15%. After approximately 235 evaluations, the search has reached within 3% and appears to have converged from this point only for slight decreases in $d_{\min}$ as the swarm exploits the preferred region. Figure 12b features the history of the solution spread. Although the algorithm does not obtain the desired spread of $\delta = 0.05$, a fairly consistent spread of $\approx$0.04 is observed after 230 evaluations. Table 6 demonstrates the proficiency of the K-UP-MOPSO algorithm against the UP-MOPSO and NSGA-II algorithms, with a similar search effort. The results clearly highlight the superiority of the K-UP-MOPSO algorithm.

The adept searching technique of K-UP-MOPSO is further demonstrated in Fig. 13a which features the 200 most recent solutions scheduled for precise evaluation. Attraction toward the preferred region dictated by the reference point is observed, which progressively becomes more focused and localized. Furthermore, few solutions appear to disturb the search direction of the algorithm, i.e., the trajectory of the search remains consistent. Hence, the reference point prescreening criterion proves to be very capable at filtering out solutions that do not reside within the preferred region. This argument also applies to particles that are scheduled for mutation, which is evidently nondestructive. Featured in Fig. 13b is the final set of 23 nondominated solutions. The optimization framework was successful in obtaining solutions that exhibit improvement over all three objectives compared with the reference point (i.e., the reference point is dominated). The solutions are clearly more inclined toward f_3, suggesting that managing the circumferential stress distribution is the most active objective.

In this study, our most preferred fuselage geometry is the solution that provides the most resemblance to the reference point in terms of compromise between the objectives. Table 7 features the percentage improvement obtained over each objective with respect to the reference geometry. The geometry of the preferred solution is featured in Fig. 14. It is observed that the cusp point is no longer as pronounced as the reference design. The preferred geometry follows a more circular form, which corresponds to the 33% recorded improvement in the stress. An evident reduction in area is visible from the passenger lobe as well as a more comfortable seating arrangement for passengers due to the elongation in the width. The circumferential stress distribution around the reference and preferred geometries is featured in Fig. 15. The von Mises stress is recorded at each circumferential station which is normalized with respect to the z-axis. In addition to a reduced overall operating stress, the stress distribution of the preferred geometry is very uniform compared with the reference design. The peak stress for both geometries occurs at the intersection of the floor.

Perhaps the greatest advantage of the K-UP-MOPSO algorithm is its ability to provide a spread of solutions to the designer. This concept offers flexibility to the designer to select a design that deviates slightly from the preferred compromise. As an example, we have analyzed the solution which provides additional improvements of 0.25 and 0.2%, respectively, in area and comfort, at the expense of a 9.5% increase in the peak stress. This is the solution with the best recorded area, as dictated by the extent of the solution spread.

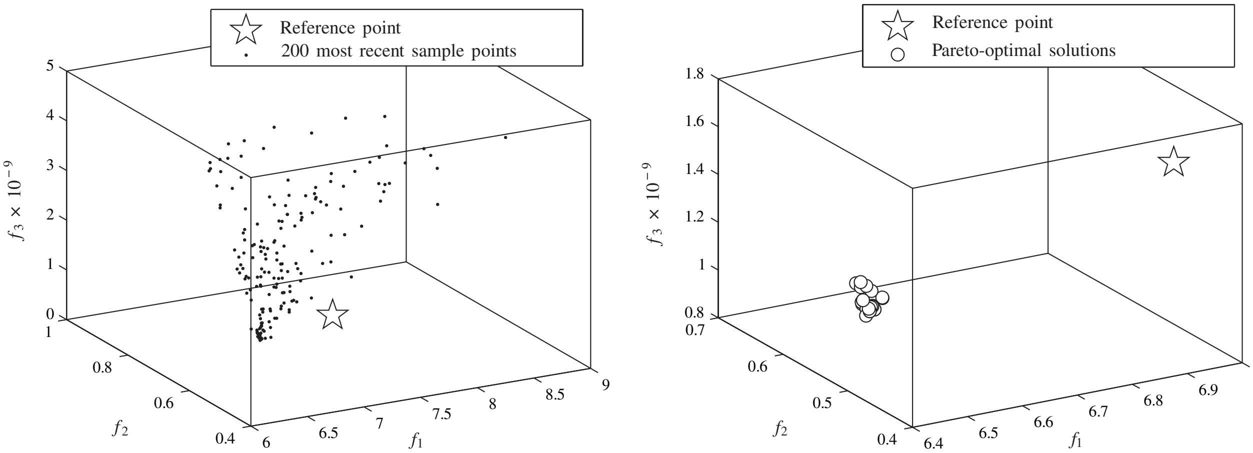
Table 6 Optimization results for the fuselage cross-sectional design

Algorithm	Function evaluations	$\min(d_z)$	$\mathrm{var}(d_z)$
NSGA-II	500	0.0649	–
UP-MOPSO	500	0.0440	0.1422
K-UP-MOPSO	450	−0.0497	0.0410

Table 7 Objective values for the preferred solution and the respective improvement over the reference design

Fuselage	Cross-sectional area, m^2	Comfort	Peak stress ($\times 10^{-9}$ Pa)
Reference	6.9928	0.5	1.4908
Preferred	6.4952	0.552	0.9915
% Improvement	7.1	10.4	33.5

***We refer to the *most preferred* solution as that with the minimum value of $d_{\min}$



a) Scatter plot of 200 most recent precise evaluations **b) Preferred region of the Pareto front**

Fig. 13 Final set of nondominated solutions and most recent sample points.

Figure 16 features the stress contour of this design, compared with the preferred solution and the reference design. For the design with minimal area, an evident increase in the peak stress, which is located at the floor intersection, is observed over the preferred design. Despite the nonuniformity in the circumferential stress distribution of the minimal area design, it clearly does not diverge any further than the reference design.

V. Conclusions

Integrating surrogate models into virtual engineering is receiving much interest, because it allows efficient management of computationally expensive design problems. In this paper a user-preference module is integrated into a kriging-assisted particle swarm algorithm to focus all computational effort on identifying only solutions of interest to the designer. The swarm is attracted to preferred regions of the Pareto front by specifying a reference in the objective space. A reference point distance metric functions as both a guidance mechanism for selecting leaders for the swarm, as well as a novel selection criterion for locally updating the kriging datasets. It is believed that this is the first attempt at integrating a user-interactive module on multiple levels to an evolutionary surrogate-assisted multi-objective optimization framework.

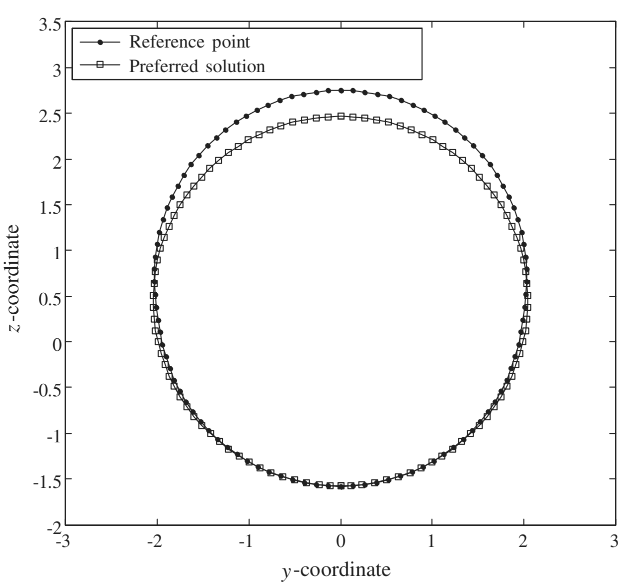


Fig. 14 Geometrical shape of the preferred design compared with the 737-style reference design.

The overarching goal of the algorithm development effort reported on here is to reduce the often prohibitive computational cost of multi-objective design search to the level of practical affordability in aerospace engineering problems. It has been shown through a mathematical test function and engineering design problems that driving a surrogate-assisted particle swarm toward a sector of special interest on the Pareto front can be an effective and efficient mechanism. As the main illustration of the algorithm, the preliminary cross-sectional design of a semimonocoque fuselage, characterized by a double-lobe parametric model, was considered. Competing drivers such as the uniformity in the circumferential stress distribution, the minimization of cross-sectional area, and the maximization of passenger comfort interacted to form a highly conflicting and deceptive optimization problem. Compared with more conventional methods, the advantages of the algorithm are conclusive, even when exposed to the apparent discontinuities in the objective landscapes. Final solutions obtained provide significant improvement over the reference geometry, and are clearly reflective of preferred interest.

The reference point criterion is capable of balancing exploration of the search space, versus maintaining a high selection pressure on solutions that exploit the preferred region. It is observed that there is a distinct attraction toward the preferred region dictated by the reference point, which implies that the reference point criterion is adept at filtering out solutions that will disrupt or deviate from the optimal search path. While the heuristic presented here proved effective over the chosen battery of tests, no mathematical proofs of convergence are available at this time, nor is the impact of the mutation operator fully understood on the convergence rate of the algorithm, potentially warranting more research in this direction. Further studies could also evaluate alternative stopping criteria, and the performance of the algorithm on larger-scale optimization problems.

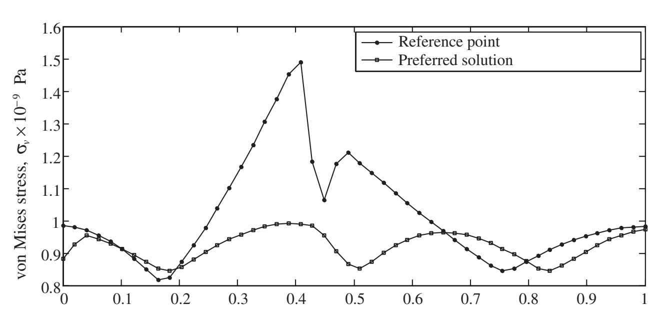
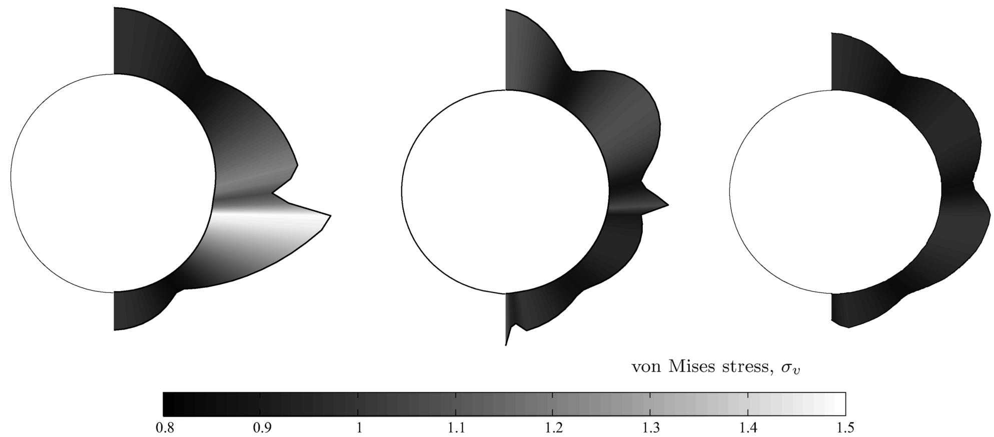


Fig. 15 The circumferential stress distribution over the preferred and reference designs.



a) Reference point **b) Best area design** **c) Preferred design**

Fig. 16 Circumferential stress contour around selected designs.

Acknowledgments

The assistance of ChunWang and Adrian Orifici of the School of Aerospace Mechanical and Manufacturing, Royal Melbourne Institute of Technology, in the Nastran simulation is gratefully acknowledged. Special thanks to Upali Wickramasinghe of the School of Computer Science and Information Technology, and Manas Khurana of the Sir Lawrence Wackett Centre for Aerospace Design Technology, both at Royal Melbourne Institute of Technology. The work of András Sóbester has been supported by the Royal Academy of Engineering and the Engineering and Physical Sciences Research Council through their Research Fellowship scheme. The contributions and helpful suggestions by the anonymous reviewers are also acknowledged.

References

[1] Keane, A. J., and Nair, P. B., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, Wiley, New York, 2005.

[2] Forrester, A., Sóbester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, New York, 2008.

[3] Epstein, B., and Peigin, S., "Efficient Approach for Multipoint Aerodynamic Wing Design of Business Jet Aircraft," *AIAA Journal*, Vol. 45, No. 11, 2007, pp. 2612–2621.
doi:10.2514/1.29307

[4] Nemec, M., Zingg, D. W., and Pulliman, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.
doi:10.2514/1.10415

[5] Engelbrecht, A. P., *Fundamentals of Computational Swarm Intelligence*, Wiley, New York, 2005.

[6] Kennedy, J., and Eberhart, R. C., *Swarm Intelligence*, Morgan Kaufmann, San Mateo, CA, 2001.

[7] Deb. K., *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, New York, 2001.

[8] Sierra, M. R., and Coello Coello, C. A., "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, Vol. 2, No. 3, 2006, pp. 287–308.
doi:10.1.1.138.1829

[9] C. Grosan, A. Abraham, and H. Ishibuchi, (eds.), *Hybrid Evolutionary Algorithms*, Springer–Verlag, Berlin, 2007.

[10] Wickramasinghe, U. K., and Li, X., "Integrating User Preferences with Particle Swarms for Multi-Objective Optimization," *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2008, pp. 745–752.

[11] Deb, K., and Sundar, J., "Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms," *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2006, pp. 635–642.

[12] Thiele, L., Miettinen, P., Korhonen, P. J., and Molina, J., "A Preference-Based Interactive Evolutionary Algorithm for Multiobjective Optimization," Helsinki School of Economics W-412, 2007.

[13] Basu, M., "A Simulated Annealing-Based Goal-Attainment Method for Economic Emission Load Dispatch of Fixed Head Hydrothermal Power Systems," *International Journal of Electrical Power and Energy Systems*, Vol. 27, No. 2, 2005, pp. 147–153.
doi:10.1016/j.ijepes.2004.09.004

[14] Carrese, R., Winarto, H., and Watmuff, J., "User-Preference Particle Swarm Algorithm for Airfoil Design Architecture," *27th International Congress of the Aeronautical Sciences*, ICAS 2010-2.6.4, 2010.

[15] Carrese, R., Winarto, H., and Li, X., "Integrating User-Preference PSO and Kriging Metamodeling for Airfoil Design," *49th AIAA Aerospace Sciences Meeting, Orlando*, AIAA 2011-1246, 2011.

[16] Wickramasinghe, U. K., Carrese, R., and Li, X., "Designing Airfoils Using a Reference Point Based Evolutionary Many-Objective Particle Swarm Optimization Algorithm," *IEEE World Congress on Computational Engineering*, IEEE Publications, Piscataway, NJ, 2010, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5586221 [accessed April 2011].

[17] Wickramasinghe, U. K., and Li, X., "Using a Distance Metric To Guide PSO Algorithms for Many-Objective Optimization," *Proceedings of the 11th Annual conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2009, pp. 667–674.

[18] Sóbester, A., Leary, S. J., and Keane, A. J., "On the Design of Optimization Strategies Based on Global Response Surface Approximation Models," *Journal of Global Optimization*, Vol. 33, No. 1, Sept. 2005, pp. 31–59.
doi:10.1007/s10898-004-6733-1

[19] Emmerich, M., and Naujoks, B., "Single- and Multi-Objective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 4, 2006, pp. 421–438.
doi:10.1109/TEVC.2005.859463

[20] Bratton, D., and Kennedy, J., "Defining a standard for particle swarm optimization," *Proceedings of the IEEE Swarm Intelligence Symposium*, IEEE Publications, Piscataway, NJ, 2007, pp. 120–127.

[21] Clerc, M., and Kennedy, J., "The Particle Swarm: Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 58–73.
doi:10.1109/4235.985692

[22] Raquel, C. R., and Naval, P. C., "An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization," *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2005, pp. 257–264.

[23] Sierra, M. R., and Coello Coello, C. A., "Improving PSO-Based Multi-Objective Optimization Using Crowding, Mutation and ϵ-Dominance," *Lecture Notes in Computer Science*, Springer, New York, 2005, pp. 505–519.

[24] Jones, D. R., "A Taxomony of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 345–383.
doi:10.1023/A:1012771025575

[25] Gano, S. E., Renaud, J. E., Martin, J. D., and Simpson, T. W., "Update Strategies for Kriging Models Used in Variable Fidelity Optimization," *Structural and Multidisciplinary Optimization*, Vol. 32, No. 4, 2006, pp. 287–298.
doi:10.1007/s00158-006-0025-y

[26] Mackay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis

of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.
doi:10.2307/1268522

[27] Jin, Y., "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Computing*, Vol. 9, No. 1, 2003, pp. 3–12.
doi:10.1007/s00500-003-0328-5

[28] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182–197.
doi:10.1109/4235.996017

[29] Tudose, L., and Jucan, D., "Pareto Approach in Multi-Objective Optimal Design of Helical Compression Springs," *Annals of the Oradea University*, Fascicle of Management and Technological Engineering, Vol. 6, No. 16, 2007.

[30] Niu, M. C. Y., *Airframe Structural Design*, Hong Kong Conmilit Press Ltd., Hong Kong, 2002.

[31] Sóbester, A., "Concise Airfoil Representation via Case-Based Knowledge Capture," *AIAA Journal*, Vol. 47, No. 5, 2009, pp. 1209–1218.
doi:10.2514/1.40119

[32] Polhamus, E. C., Geller, E. W., and Grunwald, K. J., "Pressure and Force Characteristics of Noncircular Cylinders as Affected by Reynolds Number with a Method Included for Determining the Potential Flow About Arbitrary Shapes," NASA Technical Rept. R-46, 1959.

[33] Kulfan, B. M., "Universal Parametric Geometry Representation Method," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 142–158.
doi:10.2514/1.29958

[34] Kulfan, B. M., "Fundamental Parametric Geometry Representations for Aircraft Component Shapes," AIAA Paper 2006-6948, 2006, pp. 1–45.

[35] MSC.Software Corporation, MD Nastran (NASA Structural Analysis) 2008.

T. Zang
Associate Editor