Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/geno20

Flower pollination algorithm: A novel approach for multiobjective optimization

Xin-She Yang[a], Mehmet Karamanoglu[a] & Xingshi He[b]

[a] School of Science and Technology, Middlesex University, The Burroughs, London NW4 4BT, UK

[b] School of Science, Xi'an Polytechnic University, Xi'an, PR China

Published online: 04 Oct 2013.

To cite this article: Xin-She Yang, Mehmet Karamanoglu & Xingshi He , Engineering Optimization (2013): Flower pollination algorithm: A novel approach for multiobjective optimization, Engineering Optimization, DOI: 10.1080/0305215X.2013.832237

To link to this article: http://dx.doi.org/10.1080/0305215X.2013.832237

Engineering Optimization, 2013
http://dx.doi.org/10.1080/0305215X.2013.832237

Flower pollination algorithm: A novel approach for multiobjective optimization

Xin-She Yang[a*], Mehmet Karamanoglu[a] and Xingshi He[b]

[a] *School of Science and Technology, Middlesex University, The Burroughs, London NW4 4BT, UK;*
[b] *School of Science, Xi'an Polytechnic University, Xi'an, PR China*

(*Received 18 April 2013; accepted 10 July 2013*)

Multiobjective design optimization problems require multiobjective optimization techniques to solve, and it is often very challenging to obtain high-quality Pareto fronts accurately. In this article, the recently developed flower pollination algorithm (FPA) is extended to solve multiobjective optimization problems. The proposed method is used to solve a set of multiobjective test functions and two bi-objective design benchmarks, and a comparison of the proposed algorithm with other algorithms has been made, which shows that the FPA is efficient with a good convergence rate. Finally, the importance for further parametric studies and theoretical analysis is highlighted and discussed.

Keywords: algorithm; benchmark; flower pollination algorithm; optimization; metaheuristics

1. Introduction

Real-world design problems in engineering and industry are usually multiobjective or multicriteria, and these multiple objectives often conflict with each other, which makes it impossible to use any single design option without compromise. Common approaches are to provide good approximations to the true Pareto fronts of the problem of interest so that decision makers can rank different options, depending on their preferences or their utilities (Abbass and Sarker 2002; Babu and Gujarathi 2007; Cagnina, Esquivel, and Coello Coello 2008; Deb 1999, 2001; Deb, Pratap, and Moitra 2000; Reyes-Sierra and Coello Coello 2006). Compared with single objective optimization, multiobjective optimization has its own additional challenging issues such as time complexity, inhomogeneity and dimensionality. It is usually more time consuming to obtain the true Pareto fronts because it is usually required to produce many points on the Pareto front for good approximations.

In addition, even if accurate solutions on a Pareto front can be obtained, there is still no guarantee that these solution points will be distributed uniformly on the front. In fact, it is often difficult to obtain the whole front without any part missing. For single objective optimization, the optimal solution can often be a single point in the solution space, while for bi-objective optimization, the Pareto front forms a curve, and for tri-objective cases, it becomes a surface. In fact, a higher-dimensional problem can have an extremely complex hypersurface as its Pareto front (Madavan

*Corresponding author. Email: x.yang@mdx.ac.uk

2002; Marler and Arora 2004; Yang 2010a; Yang and Gandomi 2012). Consequently, it is typically more challenging to solve such high-dimensional problems.

In the current literature of engineering optimization, a class of nature-inspired algorithms have shown their promising performance and have thus become popular and widely used, and these algorithms are mostly swarm intelligence based (Coello Coello 1999; Deb *et al.* 2002; Geem, Kim, and Loganathan 2001; Geem 2009; Ray and Liew 2002; Yang 2010a, 2010b, 2011a; Gandomi and Yang 2011; Gandomi *et al.* 2012). Metaheuristic algorithms such as particle swarm optimization, harmony search and cuckoo search are among the most popular (Geem 2009; Yang 2010). For example, harmony search, developed by Zong Woo Geem in 2001 (Geem, Kim, and Loganathan 2001; Geem 2006, 2009), has been applied in many areas such as highly challenging water distribution networks (Geem 2006) and discrete structural optimization (Lee *et al.* 2005). Other algorithms, such as the shuffled frog-leaping algorithm and particle swarm optimizers, have been applied to various optimization problems (Eusuff, Lansey, and Pasha 2006; He, Prempain, and Wu 2004; Huang 1996). There are many reasons for the popularity of metaheuristic algorithms, and the flexibility and simplicity of these algorithms certainly contribute to their success.

The main aim of this article is to extend the flower pollination algorithm (FPA), developed by Xin-She Yang in 2012 (Yang 2012), for single objective optimization to solve multiobjective optimization, and thus develop a multi-objective flower pollination algorithm (MOFPA). The rest of this article is organized as follows: Section 2 outlines the basic characteristics of flower pollination in nature and then introduces in detail the ideas of the flower pollination algorithm. Section 3 then presents the validation of the FPA by numerical experiments and a few selected multiobjective benchmarks. Then, in Section 4, two real-world design benchmarks are solved to design a welded beam and a disc brake, each with two objectives. Finally, some relevant issues are discussed and conclusions are drawn in Section 5.

2. Flower pollination algorithm

2.1. *Characteristics of flower pollination*

It is estimated that there are over a quarter of a million types of flowering plants in Nature and that about 80% of all plant species are flowering species. It still remains a mystery how flowering plants came to dominate the landscape from the Cretaceous period (Walker 2009). Flowering plants have been evolving for more than 125 million years and flowers have become so influential in evolution that it is unimaginable what the plant world would look like without flowers. The main purpose of a flower is ultimately reproduction via pollination. Flower pollination is typically associated with the transfer of pollen, and such transfer is often linked with pollinators such as insects, birds, bats and other animals. In fact, some flowers and insects have co-evolved into a very specialized flower-pollinator partnership. For example, some flowers can only attract and can only depend on a specific species of insects or birds for successful pollination.

Pollination can take two major forms: abiotic and biotic. About 90% of flowering plants depend on biotic pollination. That is, pollen is transferred by pollinators such as insects and animals. About 10% of pollination takes the abiotic form, which does not require any pollinators. Wind and diffusion help the pollination of such flowering plants, and grass is a good example of abiotic pollination (Glover 2007). Pollinators, sometimes called pollen vectors, can be very diverse. It is estimated there are at least 200,000 varieties of pollinators such as insects, bats and birds. Honeybees are a good example of pollinators, and they have also developed so-called flower constancy. That is, these pollinators tend to visit certain flower species exclusively while bypassing other flower species. Such flower constancy may have evolutionary advantages because this will maximize the transfer of flower pollen to the same or conspecific plants, thus maximizing the reproduction of the same flower species. Such flower constancy may be advantageous for pollinators as well, because they can be sure that a supply of nectar is available with their limited memory and minimum cost of learning, switching or exploring. Rather than focusing on some unpredictable but potentially more rewarding new flower species, flower constancy may require minimum investment cost and a more likely guaranteed intake of nectar (Waser 1986).

Pollination can be achieved by self-pollination or cross-pollination. Cross-pollination, or allogamy, means pollination can occur from the pollen of a flower of a different plant, while self-pollination is the fertilization of one flower, such as peach flowers, from pollen of the same flower or different flowers of the same plant, which often occurs when there is no reliable pollinator available. Biotic cross-pollination may occur over long distances, and pollinators such as bees, bats, birds and flies can fly long distances, thus this can be considered as global pollination. In addition, bees and birds may behave with Lévy flight behaviour, *i.e.* with jumping or flying distance steps obeying a Lévy distribution (Pavlyukevich 2007). Furthermore, flower constancy can be considered as an incremental stepping process using the similarity or difference of two flowers.

From the biological evolution point of view, the objective of flower pollination is the survival of the fittest and the optimal reproduction of plants in terms of numbers as well as the fittest. This can be considered as a plant species optimization process. All the above factors and processes of flower pollination interact so as to achieve the optimal reproduction of flowering plants. Therefore, this may motivate us to design new optimization algorithms.

2.2. *Flower pollination algorithm*

The flower pollination algorithm (FPA) — see Figure 1 — was developed by Xin-She Yang in 2012 (Yang 2012), inspired by the flow pollination process of flowering plants. The FPA has been extended to multi-objective optimization (Yang, Karamanoglu, and He 2013). For simplicity, the following four rules are used.

(1) Biotic cross-pollination can be considered as a process of global pollination, and pollen-carrying pollinators move in a way that obeys Lévy flights (Rule 1).
(2) For local pollination, abiotic pollination and self-pollination are used (Rule 2).
(3) Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved (Rule 3).
(4) The interaction or switching of local pollination and global pollination can be controlled by a switch probability $p \in [0, 1]$, slightly biased towards local pollination (Rule 4).

In order to formulate the updating formulas, the above rules have to be converted into proper updating equations. For example, in the global pollination step, flower pollen gametes are carried by pollinators such as insects, and pollen can travel over a long distance because insects can often fly and move over a much longer range. Therefore, Rule 1 and flower constancy (Rule 3) can be represented mathematically as

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma L(\lambda)(\boldsymbol{g}_* - \boldsymbol{x}_i^t), \tag{1}$$

where $\boldsymbol{x}_i^t$ is the pollen i or solution vector $\boldsymbol{x}_i$ at iteration t, and $\boldsymbol{g}_*$ is the current best solution found among all solutions at the current generation/iteration. Here γ is a scaling factor to control the step size.

Here $L(\lambda)$ is the parameter, more specifically the Lévy-flights-based step size, that corresponds to the strength of the pollination. Since insects may move over a long distance with various distance steps, a Lévy flight can be used to mimic this characteristic efficiently. That is, $L > 0$ is

---

Flower Pollination Algorithm (or simply Flower Algorithm)

Objective min *or* max $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$
Initialize a population of n *flowers/pollen gametes with random solutions*
Find the best solution $\boldsymbol{g}_*$ *in the initial population*
Define a switch probability $p \in [0, 1]$
while *(t <MaxGeneration)*
    for $i = 1 : n$ *(all* n *flowers in the population)*
      if *rand* $< p$,
        Draw a (d-dimensional) step vector L *which obeys a Lévy distribution*
        Global pollination via $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma L(\boldsymbol{g}_* - \boldsymbol{x}_i^t)$
      else
        Draw ϵ *from a uniform distribution in* $[0, 1]$
        Do local pollination via $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \epsilon(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t)$
      end if
      Evaluate new solutions
      If new solutions are better, update them in the population
    end for
      Find the current best solution $\boldsymbol{g}_*$
end while
Output the best solution found

Figure 1. Pseudo code of the proposed flower pollination algorithm (FPA).

drawn from a Lévy distribution

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0). \tag{2}$$

Here $\Gamma(\lambda)$ is the standard gamma function, and this distribution is valid for large steps $s > 0$. In theory, it is required that $|s_0| \gg 0$, but in practice s_0 can be as small as 0.1. However, it is not trivial to generate pseudo-random step sizes that correctly obey this Lévy distribution (2). There are a few methods for drawing such random numbers, and the most efficient one from our studies is the so-called Mantegna algorithm for drawing step size s by using two Gaussian distributions U and V by the following transformation (Mantegna 1994):

$$s = \frac{U}{|V|^{1/\lambda}}, \quad U \sim N(0, \sigma^2), \quad V \sim N(0, 1). \tag{3}$$

Here $U \sim (0, \sigma^2)$ means that the samples are drawn from a Gaussian normal distribution with a zero mean and a variance of σ^2. The variance can be calculated by

$$\sigma^2 = \left\{\frac{\Gamma(1+\lambda)}{\lambda\Gamma[(1+\lambda)/2]} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}}\right\}^{1/\lambda}. \tag{4}$$

This formula looks complicated, but it is just a constant for a given λ. For example, when $\lambda = 1$, the gamma functions become $\Gamma(1+\lambda) = 1$, $\Gamma[(1+\lambda)/2] = 1$ and

$$\sigma^2 = \left[\frac{1}{1 \times 1} \cdot \frac{\sin(\pi \times 1/2)}{2^0}\right]^{1/1} = 1. \tag{5}$$

It has been proved mathematically that the Mantegna algorithm can produce random samples that obey the required distribution (2) correctly (Mantegna 1994). By using this pseudo-random number algorithm, 50 step sizes have been drawn to form a consecutive 50 steps of Lévy flights as shown in Figure 2.

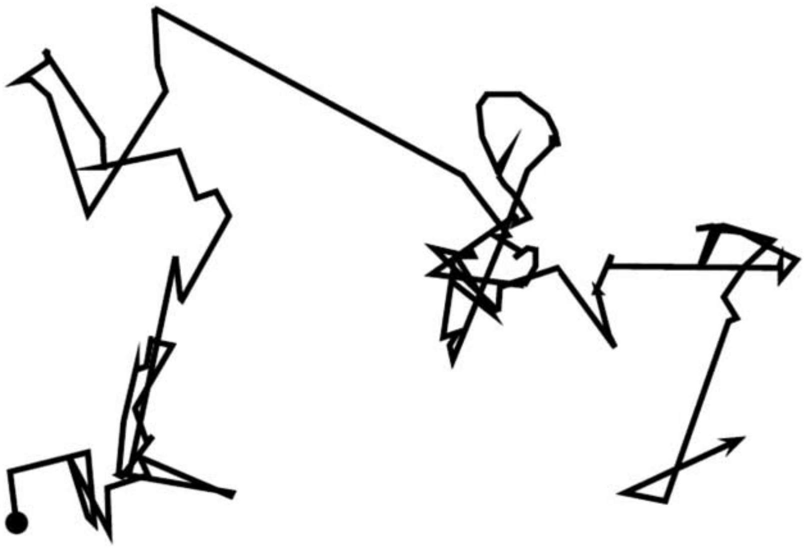

Figure 2. A series of 50 consecutive steps of Lévy flights.

For local pollination, both Rules 2 and 3 can be represented as

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \epsilon(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t), \tag{6}$$

where $\boldsymbol{x}_j^t$ and $\boldsymbol{x}_k^t$ are pollen from different flowers of the same plant species. This essentially mimics flower constancy in a limited neighbourhood. Mathematically, if $\boldsymbol{x}_j^t$ and $\boldsymbol{x}_k^t$ come from the same species or are selected from the same population; this equivalently becomes a local random walk if ϵ is drawn from a uniform distribution in [0, 1].

In principle, flower pollination activities can occur at all scales, both local and global. But in reality, adjacent flower patches or flowers in the not-so-far-away neighbourhood are more likely to be pollinated by local flower pollen than those far away. In order to mimic this feature, a switch probability (Rule 4) or proximity probability p can be effectively used to switch between common global pollination to intensive local pollination. To start with, a naive value of $p = 0.5$ may be used as an initially value. A preliminary parametric study showed that $p = 0.8$ may work better for most applications.

2.3. *Multiobjective flower pollination algorithm (MOFPA)*

A multiobjective optimization problem with m objectives can be written in general as

$$\text{Minimize} \quad f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}), \tag{7}$$

subject to the nonlinear equality and inequality constraints

$$h_j(\boldsymbol{x}) = 0, \quad (j = 1, 2, \ldots, J), \tag{8}$$

$$g_k(\boldsymbol{x}) \leq 0, \quad (k = 1, 2, \ldots, K). \tag{9}$$

In order to use the techniques for single objective optimization or extend the methods for solving multiobjective problems, there are different approaches to achieve this. One of the simplest ways is to use a weighted sum to combine multiple objectives into a composite single objective:

$$f = \sum_{i=1}^{m} w_i f_i, \tag{10}$$

with

$$\sum_{i=1}^{m} w_i = 1, \quad w_i > 0, \tag{11}$$

where m is the number of objectives and w_i $(i = 1, \ldots, m)$ are non-negative weights.

The fundamental idea of this weighted sum approach is that these weighting coefficients act as the preferences for these multiobjectives. For a given set $(w_1, w_2, \ldots, w_m)$, the optimization process will produce a single point of the Pareto front of the problem. For a different set of w_i, another point on the Pareto front can be generated. With a sufficiently large number of combinations of weights, a good approximation to the true Pareto front can be obtained. It is has been proved that the solutions to the problem with the combined objective (10) are Pareto optimal if the weights are positive for all the objectives, and these are also Pareto optimal to the original problem (7) (Miettinen 1999; Deb 2001). In practice, a set of random numbers u_i are first drawn for a uniform distribution $U(0, 1)$. Then, the weights w_i can be calculated by normalization. That is

$$w_i = \frac{u_i}{\sum_{i=1}^{m} u_i}, \tag{12}$$

so that $\sum_i w_i = 1$ can be satisfied. For example, for three objectives f_1, f_2 and f_3, three random numbers/weights can be drawn from a uniform distribution [0, 1], and they may be $u_1 = 0.2915$, $u_2 = 0.9147$ and $u_3 = 0.6821$ in one instance of sampling runs. Then, $\sum_i = 1.8883$, and $w_1 = 0.1544, w_2 = 0.4844, w_3 = 0.3612$. Indeed, $\sum_i w_i = 1.000$ is satisfied.

In order to obtain the Pareto front accurately with solutions relatively uniformly distributed on the front, random weights w_i should be used, which should be as different as possible (Miettinen 1999). From the benchmarks that have been tested, the weighted sum with random weights usually works well as can be seen below.

3. Validation and numerical experiments

There are many different test functions for multiobjective optimization (Zitzler and Thiele 1999; Zitzler, Deb, and Thiele 2000; Zhang *et al.* 2009), but a subset of some widely used functions provides a wide range of diverse properties in terms of the Pareto front and the Pareto optimal set. To validate the proposed MOFPA, a subset of these functions with convex, non-convex and discontinuous Pareto fronts has been selected, including seven single objective test functions and four multiobjective test functions, and two bi-objective design problems.

3.1. *Single objective test functions*

Before proceeding to solve multiobjective optimization problems, the algorithm should first be validated by solving some well-known single objective test functions. There are at least a hundred well-known test functions. However, there is no agreed set of test functions for validating new algorithms, though some reviews and literature do exist (Ackley 1987; Floudas *et al.* 1999; Hedar 2013; Yang 2010a). Here, a subset of seven test functions with diverse properties is used.

The Ackley function can be written as

$$f_1(\boldsymbol{x}) = -20\exp\left\{-\frac{1}{5}\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right\} - \exp\left\{\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right\} + 20 + e, \tag{13}$$

which has a global minimum $f_* = 0$ at $(0, 0, \ldots, 0)$.

The simplest of De Jong's functions is the so-called sphere function:

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2, \quad -5.12 \le x_i \le 5.12, \tag{14}$$

whose global minimum is obviously $f_* = 0$ at $(0, 0, \ldots, 0)$, and is unimodal and convex.

Easom's function:

$$f_3(\boldsymbol{x}) = (-1)^{d+1}\prod_{i=1}^{d}\cos(x_i)\exp\left\{-\sum_{i=1}^{d}(x_i - \pi)^2\right\}, \tag{15}$$

whose global minimum is $f_* = -1$ at $\boldsymbol{x}_* = (\pi, \ldots, \pi)$ within $-100 \le x_i \le 100$, has many local minima.

Griewank's function:

$$f_4(\boldsymbol{x}) = \frac{1}{4000}\sum_{i=1}^{d}x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -600 \le x_i \le 600, \tag{16}$$

whose global minimum is $f_* = 0$ at $\boldsymbol{x}_* = (0, 0, \ldots, 0)$, is highly multimodal.

Rastrigin's function:

$$f_5(\boldsymbol{x}) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)], \quad -5.12 \le x_i \le 5.12, \tag{17}$$

whose global minimum is $f_* = 0$ at $(0, 0, \ldots, 0)$, is highly multimodal.

Rosenbrock's function:

$$f_6(\boldsymbol{x}) = \sum_{i=1}^{d-1}[(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2], \tag{18}$$

has its global minimum $f_* = 0$ at $\boldsymbol{x}_* = (1, 1, \ldots, 1)$ in the domain $-5 \le x_i \le 5$, where $i = 1, 2, \ldots, d$.

Zakharov's function:

$$f_7(\boldsymbol{x}) = \sum_{i=1}^{d}x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{d}ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{d}ix_i\right)^4, \tag{19}$$

has its global minimum $f_* = 0$ at $(0, 0, \ldots, 0)$.

In order to compare the performance of the FPA with other existing algorithms, each algorithm is first tested using the most widely used implementation and parameter settings. For genetic algorithms (GAs), a crossover rate of $p_{\text{crossover}} = 0.95$ and a mutation rate of $p_{\text{mutation}} = 0.05$ are used (Holland 1975; Goldberg 1989; Yang 2010a). For particle swarm optimization (PSO), a version with an inertia weight $\theta = 0.7$ is used, and its two learning parameters $\beta_1 = \beta_2$ are

Table 1. Parameter values for each algorithm.

PSO	$n = 25$, $\theta = 0.7$, $\beta_1 = \beta_2 = 1.5$
GA	$n = 25$, $p_{\text{crossover}} = 0.95$, $p_{\text{mutation}} = 0.05$
FPA	$n = 25$, $\lambda = 1.5$, $\gamma = 0.1$, $p = 0.8$

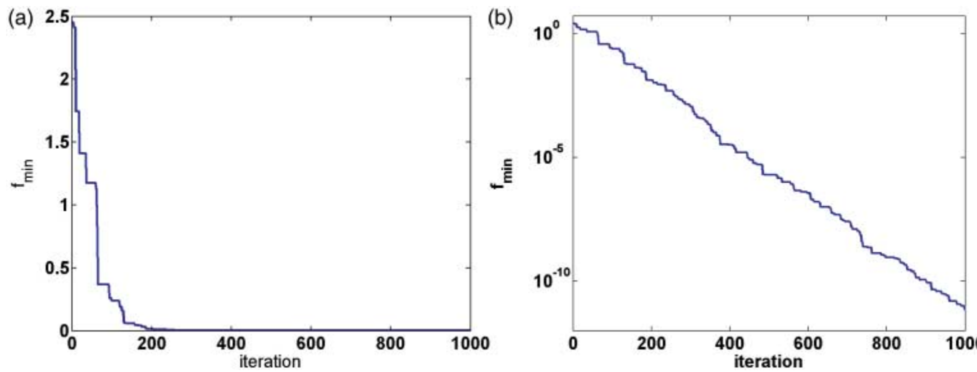

Figure 3. Convergence rate during iterations. The objective is plotted versus the iteration (left), and the same results are shown on a logarithmic scale (right).

Table 2. Comparison of algorithm performance: mean values.

Functions	GA	PSO	FPA
f_1	$8.29e^{-9}$	$7.12e^{-12}$	$5.09e^{-12}$
f_2	$6.61e^{-15}$	$1.18e^{-24}$	$2.47e^{-26}$
f_3	-0.9989	-0.9998	-1.0000
f_4	$5.72e^{-9}$	$4.69e^{-9}$	$1.37e^{-11}$
f_5	$2.93e^{-6}$	$3.44e^{-6}$	$4.52e^{-7}$
f_6	$8.97e^{-6}$	$8.21e^{-8}$	$6.19e^{-8}$
f_7	$8.77e^{-4}$	$1.58e^{-4}$	$9.53e^{-5}$

set as 1.5 (Kennedy and Eberhart 1995; Yang 2010a). Also, to ensure a fair comparison, the same population size should be used whenever possible. So $n = 25$ has been used for all three algorithms.

To get some insight into the parameter settings of the FPA, a detailed parametric study has been carried out by varying p from 0.05 to 0.95 with a step increase of 0.05, $\lambda = 1, 1.25, 1.5, 1.75, 1.9$ and $n = 5, 10, 15, \ldots, 50$. It has been found that $n = 25$, $p = 0.8$, $\gamma = 0.1$ and $\lambda = 1.5$ work for most cases. The parameter values used for all three algorithms are summarized in Table 1.

The convergence behaviour of genetic algorithms and PSO during iterations have been well studied in the literature. For the FPA, various statistical measures can be obtained from a set of runs. For example, for the Ackley function f_1, the best objective values obtained during each iteration can be plotted in a simple graph as shown in Figure 3 where a logarithmic plot shows that the convergence rate is almost exponential, which implies that the proposed algorithm is very efficient.

For a fixed population size $n = 25$, this is equal to the total number of function evaluations, which is 25,000. The best results obtained in terms of the means of the minimum values found are summarized in Table 2.

3.2. *Multiobjective test functions*

In the rest of the article, the parameters in the MOFPA are fixed, based on a preliminary parametric study, and $p = 0.8$, $\lambda = 1.5$, and a scaling factor $\gamma = 0.1$ are used. The population size $n = 50$ and the number of iterations is set to $t = 1000$. The following four functions will be tested.

- The ZDT1 function with a convex front (Zitzler and Thiele 1999; Zitzler, Deb, and Thiele 2000):

$$f_1(x) = x_1, \quad f_2(x) = g(1 - \sqrt{f_1/g}),$$
$$g = 1 + \frac{9\sum_{i=2}^{d} x_i}{d-1}, \quad x_1 \in [0,1],\ i = 2, \ldots, 30, \tag{20}$$

  where d is the number of dimensions. Pareto optimality is reached when $g = 1$.
- The ZDT2 function with a non-convex front:

$$f_1(x) = x_1, \quad f_2(x) = g\left(1 - \frac{f_1}{g}\right)^2,$$

  where g is the same as given in ZDT1.
- The ZDT3 function with a discontinuous front:

$$f_1(x) = x_1, \quad f_2(x) = g\left[1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1)\right],$$

  where g in functions ZDT2 and ZDT3 is the same as in function ZDT1. In the ZDT3 function, f_1 varies from 0 to 0.852 and f_2 from -0.773 to 1.
- The LZ function (Li and Zhang 2009; Zhang and Li 2007):

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j \in J_1}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{d}\right)\right]^2,$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j \in J_2}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{d}\right)\right]^2, \tag{21}$$

  where $J_1 = \{j | j \text{ is odd }\}$ and $J_2 = \{j | j \text{ is even }\}$ where $2 \le j \le d$. This function has a Pareto front $f_2 = 1 - \sqrt{f_1}$ with a Pareto set

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{d}\right), \quad j = 2, 3, \ldots, d, \quad x_1 \in [0,1]. \tag{22}$$

After generating 100 Pareto points by the MOFPA, the Pareto front generated by the MOFPA is compared with the true front $f_2 = 1 - \sqrt{f_1}$ of ZDT1 (see Figure 4).

Define the distance or error between the estimated Pareto front $\mathrm{PF^e}$ to its corresponding true front $\mathrm{PF^t}$ as

$$E_f = \|\mathrm{PF^e} - \mathrm{PF^t}\|^2 = \sum_{j=1}^{N}(\mathrm{PF}_j^{\mathrm{e}} - \mathrm{PF}_j^{\mathrm{t}})^2, \tag{23}$$

where N is the number of points.

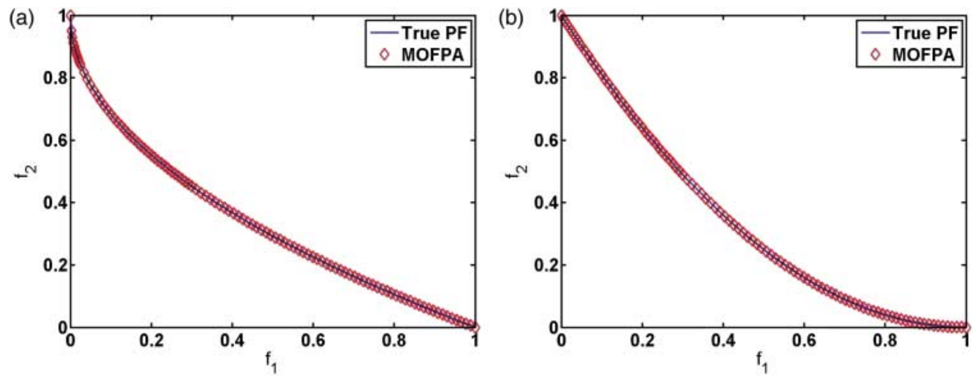

Figure 4. (a) Pareto front of test function ZDT1, and (b) Pareto front of test function ZDT2.

Table 3. Summary of results.

Functions	Errors (1000 iterations)	Errors (2500 iterations)
ZDT1	$1.1e^{-6}$	$3.1e^{-19}$
ZDT2	$2.7e^{-6}$	$4.4e^{-10}$
ZDT3	$1.4e^{-5}$	$7.2e^{-12}$
LZ	$1.2e^{-6}$	$2.9e^{-12}$

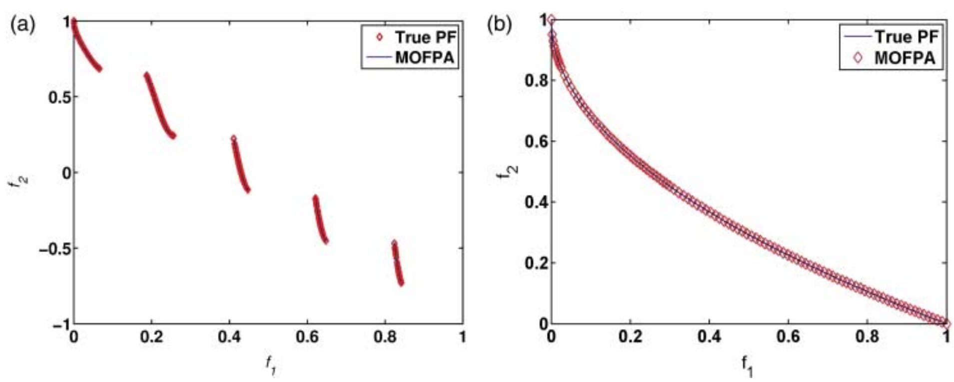

Figure 5. (a) Pareto front of test function ZDT3, and (b) Pareto front of test function LZ.

The variation of convergence rates or the convergence property can be viewed by plotting out the errors during iterations, as this measure is an absolute measure that depends on the number of points. Sometimes, it is easier to use a relative measure in terms of the generalized distance

$$D_g = \frac{1}{N}\sqrt{\sum_{j=1}^{N}(\mathrm{PF}_{ej} - \mathrm{PF}_j^{\mathrm{t}})^2}. \tag{24}$$

The results for all the functions are summarized in Table 3, and the estimated Pareto fronts and true fronts of other functions are shown in Figures 4 and 5. In all these figures, the vertical axis is f_2 and the horizontal axis is f_1.

Table 4. Comparison of D_g for $n = 50$ and $t = 500$ iterations.

Method	ZDT1	ZDT2	ZDT3	LZ
VEGA	$3.79e^{-02}$	$2.37e^{-03}$	$3.29e^{-01}$	$1.47e^{-03}$
NSGA-II	$3.33e^{-02}$	$7.24e^{-02}$	$1.14e^{-01}$	$2.77e^{-02}$
MODE	$5.80e^{-03}$	$5.50e^{-03}$	$2.15e^{-02}$	$3.19e^{-03}$
DEMO	$1.08e^{-03}$	$7.55e^{-04}$	$1.18e^{-03}$	$1.40e^{-03}$
Bees	$2.40e^{-02}$	$1.69e^{-02}$	$1.91e^{-01}$	$1.88e^{-02}$
SPEA	$1.78e^{-03}$	$1.34e^{-03}$	$4.75e^{-02}$	$1.92e^{-03}$
MOFPA	$7.11e^{-05}$	$1.24e^{-05}$	$5.49e^{-04}$	$7.92e^{-05}$

3.3. *Analysis of results and comparison*

In order to compare the performance of the proposed MOFPA with other established multiobjective algorithms, we have selected a few algorithms with available results from the literature. In cases where results are not available, the algorithms have been implemented using well-documented studies and then new results generated using these algorithms. In particular, other methods have also been used for comparison, including the vector evaluated genetic algorithm (VEGA) (Schaffer 1985), NSGA-II (Deb, Pratap, and Moitra 2000), multiobjective differential evolution (MODE) (Babu and Gujarathi 2007; Xue 2004), differential evolution for multiobjective optimization (DEMO) (Robič and Filipič 2005), multiobjective bees algorithms (Bees) (Pham and Ghanbarzadeh 2007), and the strength Pareto evolutionary algorithm (SPEA) (Deb *et al.* 2002; Madavan 2002). The performance measures in terms of generalized distance D_g are summarized in Table 4 for all the above major methods.

It is clearly seen from Table 4 that the proposed MOFPA obtained better results for almost all four cases.

4. Structural design examples

Design optimization, especially the design of structures, has many applications in engineering and industry. As a result, there are many different benchmarks with detailed studies in the literature (Kim, Oh, and Lee 1997; Pham and Ghanbarzadeh 2007; Ray and Liew 2002; Rangaiah 2008). In the rest of this article, the MOFPA will be used to solve two design case studies: design of a beam and a disc brake (Osyczka and Kundu 1995; Ray and Liew 2002; Gong, Cai, and Zhu 2009).

4.1. *Welded beam design*

The multiobjective design of a welded beam is a classical benchmark that has been solved by many researchers (Deb 1999; Ray and Liew 2002). The problem has four design variables: the width w and length L of the welded area, the depth d and thickness h of the main beam. The objective is to minimize both the overall fabrication cost and the end deflection δ.

The detailed formulation can be found in Deb (1999), Ray and Liew (2002) and Gong, Cai, and Zhu (2009). Here the main problem is rewritten as

$$\text{minimize } f_1(\boldsymbol{x}) = 1.104,71w^2L + 0.048,11dh(14.0 + L), \quad \text{minimize } f_2 = \delta, \tag{25}$$

subject to

$$g_1(\boldsymbol{x}) = w - h \le 0,$$
$$g_2(\boldsymbol{x}) = \delta(\boldsymbol{x}) - 0.25 \le 0,$$

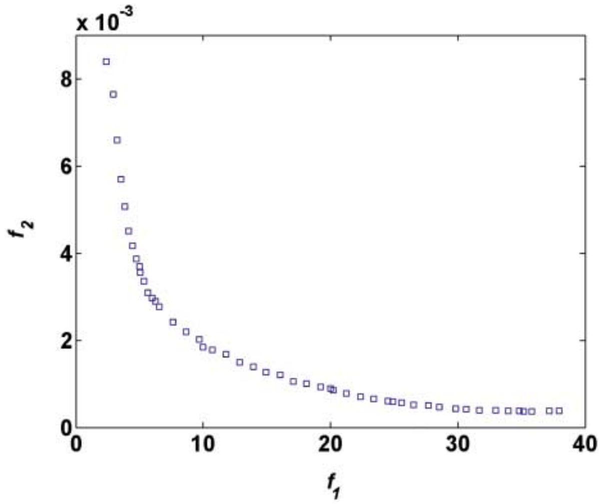

Figure 6. Pareto front for the bi-objective beam design where the horizontal axis corresponds to cost and the vertical axis corresponds to deflection.

$$\begin{aligned} g_3(\boldsymbol{x}) &= \tau(\boldsymbol{x}) - 13,600 \leq 0, \\ g_4(\boldsymbol{x}) &= \sigma(\boldsymbol{x}) - 30,000 \leq 0, \\ g_5(\boldsymbol{x}) &= 0.104,71w^2 + 0.048,11hd(14+L) - 5.0 \leq 0, \\ g_6(\boldsymbol{x}) &= 0.125 - w \leq 0, \\ g_7(\boldsymbol{x}) &= 6000 - P(\boldsymbol{x}) \leq 0, \end{aligned} \tag{26}$$

where

$$\begin{aligned} \sigma(\boldsymbol{x}) &= \frac{504,000}{hd^2}, \quad Q = 6000\left(14 + \frac{L}{2}\right), \\ D &= \tfrac{1}{2}\sqrt{L^2 + (w+d)^2}, \quad J = \sqrt{2}\, wL\left[\frac{L^2}{6} + \frac{(w+d)^2}{2}\right], \\ \delta &= \frac{65,856}{30,000hd^3}, \quad \beta = \frac{QD}{J}, \\ \alpha &= \frac{6000}{\sqrt{2}wL}, \quad \tau(\boldsymbol{x}) = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}, \\ P &= 0.614,23 \times 10^6\, \frac{dh^3}{6}\left(1 - \frac{d\sqrt{30/48}}{28}\right). \end{aligned} \tag{27}$$

The simple limits or bounds are $0.1 \leq L, d \leq 10$ and $0.125 \leq w, h \leq 2.0$. This design problem has been solved using the MOFPA. The approximate Pareto front generated by the 50 non-dominated solutions after 1000 iterations is shown in Figure 6. This is consistent with the results obtained by others (Ray and Liew 2002; Pham and Ghanbarzadeh 2007).

4.2. *Disc brake design*

The objectives are to minimize the overall mass and the braking time by choosing optimal design variables: the inner radius r, the outer radius R of the discs, the engaging force F and the number

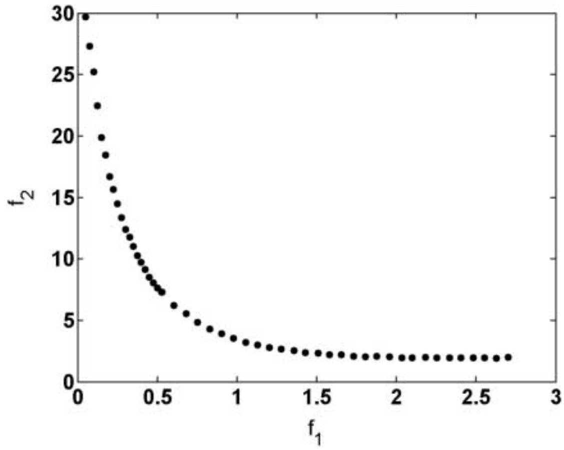

Figure 7. Pareto front of the disc brake design.

of the friction surface s. This is under design constraints such as the torque, pressure, temperature and length of the brake (Ray and Liew 2002; Pham and Ghanbarzadeh 2007).

This bi-objective design problem can be written as

$$\text{Minimize } f_1(\boldsymbol{x}) = 4.9 \times 10^{-5}(R^2 - r^2)(s-1), \quad f_2(\boldsymbol{x}) = \frac{9.82 \times 10^6 (R^2 - r^2)}{Fs(R^3 - r^3)}, \tag{28}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= 20 - (R - r) \leq 0, \\ g_2(\boldsymbol{x}) &= 2.5(s+1) - 30 \leq 0, \\ g_3(\boldsymbol{x}) &= \frac{F}{3.14(R^2 - r^2)} - 0.4 \leq 0, \\ g_4(\boldsymbol{x}) &= \frac{2.22 \times 10^{-3} F (R^3 - r^3)}{(R^2 - r^2)^2} - 1 \leq 0, \\ g_5(\boldsymbol{x}) &= 900 - \frac{0.0266 Fs(R^3 - r^3)}{(R^2 - r^2)} \leq 0. \end{aligned} \tag{29}$$

The simple limits are

$$55 \leq r \leq 80, \quad 75 \leq R \leq 110, \quad 1000 \leq F \leq 3000, \quad 2 \leq s \leq 20. \tag{30}$$

It is worth pointing out that s is discrete. In general, the MOFPA has to be extended in combination with constraint handling techniques so as to deal with mixed integer problems efficiently. However, since there is only one discrete variable, the simplest branch-and-bound method is used here.

In order to see how the proposed MOFPA performs for real-world design problems, the same problem has also been solved using other available multiobjective algorithms. Fifty solution points are generated using the MOFPA to form an approximatation to the true Pareto front after 1000 iterations, as shown in Figure 7.

A comparison of the convergence rates is plotted on the logarithmic scales in Figure 8. It can be seen clearly that the convergence rate of the MOFPA is the highest in an exponentially decreasing way. This suggests that the MOFPA provides better solutions in a more efficient way.

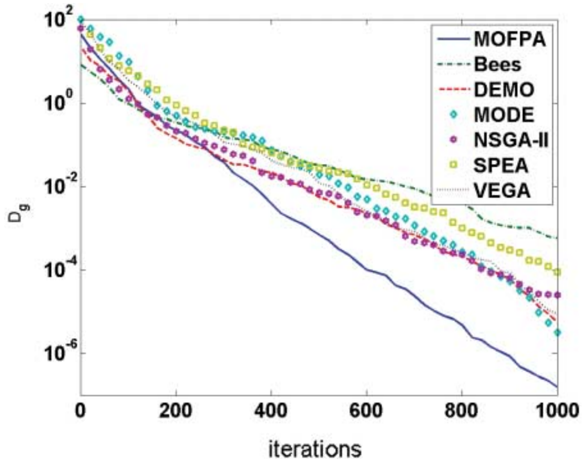

Figure 8. Convergence comparison for the disc brake design.

The above results for 11 test functions in total and two design examples suggest that the MOFPA is a very efficient algorithm for multiobjective optimization. The proposed algorithm can deal with highly nonlinear, multiobjective optimization problems with complex constraints and diverse Pareto optimal sets.

5. Discussions and conclusions

Multiobjective optimization in engineering and industry is often very challenging to solve, necessitating sophisticated techniques to tackle. Metaheuristic approaches have shown promise and popularity in recent years.

In the present work, a new algorithm, called the flower pollination algorithm, has been formulated for multiobjective optimization applications by mimicking the pollination process of flowering plants. Numerical experiments and design benchmarks have shown that the proposed algorithm is very efficient with an almost exponential convergence rate, based on comparison of the FPA with other algorithms for solving multiobjective optimization problems.

It is worth pointing out that mathematical analysis is highly necessary in future work so as to gain insight into the true working mechanisms of metaheuristic algorithms such as the MOFPA. The FPA has advantages such as simplicity and flexibility, and in many ways it has some similarity to cuckoo search and other algorithms with Lévy flights (Yang 2010a, 2011b); however, it is still not clear that why the FPA works well. In terms of number of parameters, the FPA has only one key parameter p together with a scaling factor γ, which makes the algorithm easier to implement. However, the nonlinearity in Lévy flights make it difficult to analyse mathematically. It can be expected that this nonlinearity in the algorithm formulations may be advantageous in enhancing the performance of an algorithm. More research may reveal the subtlety of this feature.

For multiobjective optimization, an important issue is how to ensure the solution points can distribute relatively uniformly on the Pareto front for test functions. However, for real-world design problems such as the design of a disc brake and a welded beam, the solutions are not quite uniform on the Pareto fronts, and there is still room for improvement. However, simply generating more solution points may not solve the Pareto uniformity problem easily. In fact, how to maintain a uniform spread on the Pareto front is still a challenging problem that requires more study. It may be useful as a further research topic to study other approaches for multiobjective optimization, such as the ϵ-constraint method, weighted metric methods, Benson's method, utility methods and evolutionary methods (Miettinen 1999; Coello Coello 1999; Deb 2001).

On the other hand, further studies could focus on more detailed parametric analysis and gain insight into how algorithm-dependent parameters can affect the performance of an algorithm. Furthermore, the linearity in the main updating formulas makes it possible to do some theoretical analysis in terms of dynamic systems or Markov chain theories, while the nonlinearity in terms of Lévy flights can make it difficult to analyse the FPA exactly. All these could form useful topics for further research.

References

Abbass, H. A., and R. Sarker. 2002. "The Pareto Differential Evolution Algorithm." *International Journal on Artificial Intelligence Tools* 11 (4): 531–552.

Ackley, D. H. 1987. *A Connectionist Machine for Genetic Hillclimbing*. Wisconsin, Madison: Kluwer Academic.

Babu, B. V., and A. M. Gujarathi. 2007. "Multi-Objective Differential Evolution (MODE) for Optimization of Supply Chain Planning and Management." In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, 25–28 September 2007, Singapore, 2732–2739. Singapore: IEEE Press. doi: 10.1109/CEC.2007.4424816.

Cagnina, L. C., S. C. Esquivel, and Carlos A. Coello Coello. 2008. "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer." *Informatica* 32 (2): 319–326.

Coello Coello, Carlos A. 1999. "An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends." In *Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC99)*, 6–9 July 1999, Washington, DC. Washington DC: IEEE Press. doi: 10.1109/CEC.1999.781901.

Deb, K. 1999. "Evolutionary Algorithms for Multi-Criterion Optimization in Engineering design." In *Evolutionary Aglorithms in Engineering and Computer Science*, 135–161. New York: Wiley.

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

Deb, K., A. Pratap, S. Agarwal, and T. Mayarivan. 2002. "A Fast and Elistist Multiobjective Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Deb, K., A. Pratap, and S. Moitra. 2000.. "Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA." In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 16–20 September 2000, Paris, 859–868. Berlin Heidelberg: Springer.

Eusuff, M., K. Lansey, and F. Pasha. 2006. "Shuffled Frog-Leaping Algorithm: A Memetic Metaheuristic for Discrete Optimization." *Engineering Optimization* 38 (2): 129–154.

Floudas, C. A., P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gumus, S. T. Harding, J. L. Klepeis, C. A. Meyer, and C. A. Scheiger. 1999. *Handbook of Test Problems in Local and Global Optimization*. Berlin Heidelberg: Springer.

Gandomi, A. H., and X.-S. Yang. 2011. "Benchmark Problems in Structural Optimization." In *Computational Optimization, Methods and Algorithms*, edited by Slawomir Koziel and Xin-She Yang, Vol. 356 of *Studies in Computational Intelligence*, 259–281. Berlin Heidelberg: Springer.

Gandomi, A. H., X.-S. Yang, S. Talatahari, and S. Deb. 2012. "Coupled Eagle Strategy and Differential Evolution for Unconstrained and Constrained Global Optimization." *Computers & Mathematics with Applications* 63 (1): 191–200.

Geem, Z. W. 2006. "Optimal Cost Design of Water Distribution Networks Using Harmony Search." *Engineering Optimization* 38 (3): 259–280.

Geem, Z. W. 2009. *Music-Inspired Harmony Search Algorithm: Theory and Applications*. Heidelberg: Springer.

Geem, Z. W., J. H. Kim, and G. V. Loganathan. 2001. "A New Heuristic Optimization: Harmony Search." *Simulation* 76 (2): 60–68.

Glover, B. J. 2007. *Understanding Flowers and Flowering: An Integrated Approach*. Oxford, UK: Oxford University Press.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Reading, MA: Addison-Wesley.

Gong, W. Y., Z. H. Cai, and L. Zhu. 2009. "An Effective Multiobjective Differential Evolution Algorithm for Engineering Design." *Structural and Multidisciplinary Optimization* 38 (2): 137–157.

He, S., E. Prempain, and Q. H. Wu. 2004. "An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems." *Engineering Optimization* 36 (5): 585–605.

Hedar, A. 2013. "*Test function web pages*." Accessed June 1, 2013. http://www-optima.amp.i.kyoto-u.ac.jp /member/student/hedar/Hedar_files/TestGO_files/Page364.htm.

Holland, J. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

Huang, G. H. 1996. "IPWM: An Interval Parameter Water Quality Management Model." *Engineering Optimization* 26 (1): 79–103.

Kennedy, J., and R. C. Eberhart. 1995. "Particle Swarm Optimization." In Vol. 4 of *Proceedings of the IEEE International Conference on Neural Networks*, 27 November–1 December 1995, Perth, Western Australia, 1942–1948. Piscataway, NJ: IEEE Press. doi: 10.1109/ICNN.1995.488968

Kim, J. T., J. W. Oh, and I. W. Lee. 1997. "Multiobjective Optimization of Steel Box Girder Bridge." In *Proceedings of the 7th KAIST-NTU-KU Trilateral Seminar/Workshop on Civil Engineering*, Kyoto, 1–3 December 1997. Piscataway, NJ: IEEE Press.

Lee, K. S., Z. W. Geem, S.-H. Lee, and K.-W. Bae. 2005. "The Harmony Search Heuristic Algorithm for Discrete Structural Optimization." *Engineering Optimization* 37 (7): 663–684.
Li, H., and Q. F. Zhang. 2009. "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II." *IEEE Transactions on Evolutionary Computation* 13: 284–302.
Madavan, N. K. 2002. "Multiobjective Optimization Using a Pareto Differential Evolution Approach." In Vol. 2 of *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, 1145–1150. Piscataway, NJ: IEEE Press.
Mantegna, R. N. 1994. "Fast, Accurate Algorithm for Numerical Simulation of Lévy Stable Stochastic Process." *Physical Review E* 49: 4677–4683.
Marler, R. T., and J. S. Arora. 2004. "Survey of Multi-Objective Optimization Methods for Engineering." *Structural and Multidisciplinary Optimization* 26: 369–395.
Miettinen, Kaisa M. 1999. *Nonlinear Multiobjective Optimization*. Vol. 12 of *International Series in Operations Research & Management Science*. Boston, MA: Kluwer Academic.
Osyczka, Andrzej, and Sourav Kundu. 1995. "A Genetic Algorithm-Based Multicriteria Optimization Method." In *Proceedings of the 1st World Congress on Structural and Multidisciplinary Optimization*, May 1995, Goslar, Germany, edited by Niels Olhoff and George I. N. Rozvany, 909–914. Oxford: Elsevier Science.
Pavlyukevich, I. 2007. "Lévy Flights, Non-Local Search and Simulated Annealing." *Journal of Computational Physics* 226: 1830–1844.
Pham, D. T., and A. Ghanbarzadeh. 2007. "Multi-Objective Optimisation Using the Bees Algorithm." In *3rd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007)*, 2–13 July 2007 Whittles, Dunbeath, Scotland.
Rangaiah, G. 2008. *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*. Singapore: World Scientific. http://dc247.4shared.com/doc/yhZtUlMZ/preview.html
Ray, L., and K. M. Liew. 2002. "A Swarm Metaphor for Multiobjective Design Optimization." *Engineering Optimization* 34 (2): 141–153.
Reyes-Sierra, M., and Carlos A. Coello Coello. 2006. "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art." *International Journal of Computational Intelligence Research* 2 (3): 287–308.
Robič, T., and B. Filipič. 2005. "DEMO: Differential Evolution for Multiobjective Optimization." In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, 9–11 March 2005, Guanajuato, Mexico, edited by Carlos A. Coello Coello, Arturo Hernández Aguirre and Eckart Zitzler. In Vol. 3410 of *Lecture Notes in Computer Science*, 520–533. Heidelberg, Germany: Springer.
Schaffer, J. David. 1985. "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms." In *Proceedings of the 1st International Conference on Genetic Aglorithms*, 24–26 July 1985, Carnegie-Mellon University, Pittsburgh, PA, 93–100. Hillsdale, NJ: L. Erlbaum Associates.
Walker, M. 2009. "How Flowers Conquered the World." *BBC Earth News*, 10 July 2009. http://news.bbc.co.uk/earth/hi/earth_news/newsid_8143000/8143095.stm.
Waser, N. M. 1986. "Flower Constancy: Definition, Cause and Measurement." *The American Naturalist* 127 (5): 596–603.
Xue, F., 2004. "Multi-Objective Differential Evolution: Theory and Applications." PhD diss., Rensselaer Polytechnic Institute, Troy, NY.
Yang, X.-S. 2010a. *Engineering Optimization: An Introduction with Metaheuristic Applications*. Hoboken, New Jersey: Wiley.
Yang, X.-S. 2010b. "A New Metaheuristic Bat-Inspired Algorithm." In *Proceedings of Nature-Inspired Cooperative Strategies for Optimization (NICSO 2010)*, edited by J. R. González et al. In Vol. 284 of *Studies in Computational Intelligence*, 65–74. Berlin: Springer.
Yang, X.-S. 2011a. "Bat Algorithm for Multi-Objective Optimisation." *International Journal of Bio-Inspired Computation* 3 (5): 267–274.
Yang, X.-S. 2011b. "Review of Meta-Heuristics and Generalised Evolutionary Walk Algorithm." *International Journal of Bio-Inspired Computation* 3 (2): 77–84.
Yang, X.-S. 2012. "Flower Pollination Algorithm for Global Optimization." In *Unconventional Computation and Natural Computation*, edited by J. Durand-Lose and N. Jonoska, Vol. 7445 of *Lecture Notes in Computer Science*, 240–249. Berlin: Springer.
Yang, X.-S., and A. H. Gandomi. 2012. "Bat Algorithm: A Novel Approach for Global Engineering Optimization." *Engineering Computations* 29 (5): 464–483.
Yang, X.-S., M. Karamanoglu, and X. S. He. 2013. "Multi-Objective Flower Algorithm for Optimization." *Procedia Computer Science* 18: 861–868.
Zhang, Q. F., and H. Li. 2007. "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition." *IEEE Transactions on Evolutionary Computation* 11 (6): 712–731.
Zhang, Q. F., A. M. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. 2009. *Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition*. Technical Report CES-487. Colchester, UK: University of Essex.
Zitzler, E., K. Deb, and L. Thiele. 2000. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *IEEE Transactions on Evolutionary Computation* 8 (2): 173–195.
Zitzler, E., and L. Thiele. 1999. "Multiobjective Evolutonary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *IEEE Transactions on Evolutionary Computation* 3 (4): 257–271.