Parallel evolutionary algorithm for constrained single and multi objective optimisation–Part II: Multi/many-objective test cases and applications

Dorival M. Pedroso, *Member, IEEE*

***Abstract*—This paper is a follow-up of Part I that introduced an algorithm to solve nonlinear optimisation problems. The proposed code is named goga and employs a number of strategies to be efficient and accurate. Additionally, the repeatability of results is targeted. In this part, the performance in solving constrained multi objective optimisation problems is assessed. It is demonstrated that most tests are passed with excellent accuracy, efficiency and repeatability properties. Some applications are studied in this part as well; these include: topology optimisation of trusses and environmental economical dispatch in power generation. In the topology problem, a method based on the proposed out-of-range functions is described for handling failures of the finite element calculations.**

***Index Terms*—genetic algorithm, differential evolution, topology optimisation, environmental/economic dispatch, finite element method**

I. Introduction

Part I [1] has introduced the algorithm for solving nonlinear optimisation problems in detail. It was demonstrated that the resulting code was able to solve a number of single objective problems with many constraints and some two objective problems with no constraints, apart from the limits of the primary variables ($\underset{\sim}{x}$); i.e. *box constraints*. In particular, the ability to obtain the same result from different runs was demonstrated with 1000 samples and observing a small standard deviation.

In this Part II, two objective problems with constraints and three objective optimisation problems are studied. In addition, some tests with up to 20 objectives are investigated. At the end, two applications are presented. The first one is the shape and topology optimisation of truss structures with a mix of integers and real numbers that requires a robust strategy to deal with failures of the underlying finite element calculations. The second application is the interesting environmental economic dispatch problem in electricity generation.

As in Part I, the source code of all test cases and applications is available in https://github.com/cpmech/goga.

In all tests, the accuracy is assessed by comparing the numerical versus analytical results. The analytical optimal Pareto front is represented by $\psi(f_0, f_1, f_2, \ldots) = 0$. A root mean square error is then computed as follows

$$E = \sqrt{\frac{1}{n}\left[\sum_{i=0}^{n-1} \psi(f_0^i, f_1^i, f_2^i, \ldots)^2\right]} \tag{1}$$

where f_j^i is the numerical solution i for objective function j and n indicates the number of feasible solutions on the optimal front with $\phi_i = 0$. The spread is visually inspected in plots.

II. Test cases: Constrained two objective

A series of problems presented in [2] where a pair of generator functions is cleverly designed in order to produce challenging constrained problems is considered. These problems are symbolised by *CTPi* and include some with local minima and nonuniform distribution of points on the Pareto-optimal fronts. The interesting problem *TNK* from [3] is studied as well.

Compared to the solutions presented in Part I, the challenge to the evolutionary optimiser is hence increased. Nine problems are considered as described below. The number of solutions for these problems is chosen as $N_{sol} = 120$ in order to better draw the optimal front. The differential evolution coefficient C_{DE} is set equal to $C_{DE} = 0.1$ because this value seemed to give better results when solving the problems in this section. A statistical analysis is presented just after all problems are described.

Problem TNK [2] has two constraints and a discontinuous optimal front. $N_x = 2$ and the problem is defined by

$$\begin{aligned}
f_0 &= x_0 \\
f_1 &= x_1 \\
g_0 &= x_0^2 + x_1^2 - 1 - 0.1 \cos\left[16 \operatorname{atan}(x_0/x_1)\right] \\
g_1 &= 0.5 - (x_0 - 0.5)^2 - (x_1 - 0.5)^2
\end{aligned} \tag{2}$$

where the limits of variables are

$$0 \le x_i \le \pi \quad (i \in [0, 1]) \tag{3}$$

The results are plotted in Fig. 1 where it can be seen a good response: accurate and spread. In addition, all solutions satisfy the constraints.

The next *CTPi* problems are all defined with $N_x = 10$ and two groups; i.e. $N_{cpu} = 2$. In addition, the limits of variables for all of them are

$$0 \le x_i \le 1 \quad (i \in [0, N_x]) \tag{4}$$

School of Civil Engineering, The University of Queensland, St Lucia QLD 4072, Australia, e-mail:d.pedroso@uq.edu.au

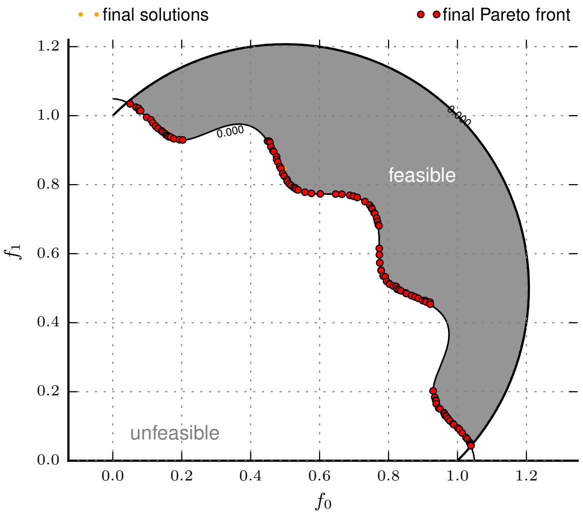


Fig. 1. TNK: Results.

TABLE I
COEFFICIENTS FOR CTP2-CTP7

P	θ	a	b	c	d	e
CTP2	$-0.2\,\pi$	0.2	10.0	1.0	6.0	1.0
CTP3	$-0.2\,\pi$	0.1	10.0	1.0	0.5	1.0
CTP4	$-0.2\,\pi$	0.75	10.0	1.0	0.5	1.0
CTP5	$-0.2\,\pi$	0.1	10.0	2.0	0.5	1.0
CTP6	$0.1\,\pi$	40.0	0.5	1.0	2.0	-2.0
CTP7	$-0.05\,\pi$	40.0	5.0	1.0	6.0	0.0

Problem CTP1 [2], as considered in this work, has two constraints and is defined by

$$\begin{aligned} c_0 &= 1 + \textstyle\sum_{i=1}^{N_x-1} x_i \\ f_0 &= x_0 \\ f_1 &= c_0 \, \exp\left(-x_0/c_0\right) \\ g_0 &= f_1 - a_0 \, \exp\left(-b_0 \, f_0\right) \\ g_1 &= f_1 - a_1 \, \exp\left(-b_1 \, f_0\right) \end{aligned} \tag{5}$$

where the following auxiliary coefficients are selected

$$a_0 = 0.858 \quad b_0 = 0.541 \quad a_1 = 0.728 \quad b_1 = 0.295 \tag{6}$$

Good results are obtained as illustrated in Fig. 2.

Problems CTP2-CTP7 [2] have one constraint and, in this work, are all defined by the following expressions

$$\begin{aligned} c_0 &= 1 + \textstyle\sum_{i=1}^{N_x-1} x_i \\ f_0 &= x_0 \\ f_1 &= c_0 \left(1 - f_0/c_0\right) \\ c_1 &= \cos\left(\theta\right) \left(f_1 - e\right) - \sin\left(\theta\right) f_0 \\ c_2 &= \sin\left(\theta\right) \left(f_1 - e\right) + \cos\left(\theta\right) f_0 \\ c_3 &= \sin\left(b \, \pi \left(c_2\right)^c\right) \\ g_0 &= c_1 - a \left(\left|c_3\right|\right)^d \end{aligned} \tag{7}$$

where a, b, c, d and θ are auxiliary coefficients that help to design many combinations. The selected values are listed in Table I.

Problem CTP2 has a number of narrow discontinuous Pareto-optimal fronts as illustrated in Fig. 3. The presented code can obtain satisfactory results every time it is run.

Problem CTP3 is more challenging because the optimal front reduces to a set of single points (Fig. 4).

Problem CTP4 (Fig. 5) is also more challenging because the 'path' available for points to move towards the best front is very narrow. The proposed code is able to obtain some reasonable results.

Problem CTP5 is similar to Problem CTP3 but has a nonuniform distribution of points along the optimal front. The results are plotted in Fig. 6 and the response is similar to CTP3; i.e. the points approach the optimal front with good spread.

Problem CTP6 has bands defining feasible and infeasible regions; hence additional problems would arise if the solver was not able to properly handle and 'jump over' constraints. Fig. 7 show the results where the optimal front lies along the constraint line.

Problem CTP7 (Fig. 8) has also multiple unfeasible regions. The Pareto-optimal front is discontinuous and follows a straight line defined by $f_1 = 1 - f_0$.

Problem CTP8 [2] is now introduced. This problem is similar to the previous CTP ones; however two constraints using the same generator are considered. For the sake of clarity, the equations are written below

$$\begin{aligned} c_1 &= \cos\left(\theta_1\right) \left(f_1 - e\right) - \sin\left(\theta_1\right) f_0 \\ c_2 &= \sin\left(\theta_1\right) \left(f_1 - e\right) + \cos\left(\theta_1\right) f_0 \\ c_3 &= \sin\left(b \, \pi \left(c_2\right)^c\right) \\ g_0 &= c_1 - a \left(\left|c_3\right|\right)^d \\ d_1 &= \cos\left(\theta_2\right) \left(f_1 - E\right) - \sin\left(\theta_2\right) f_0 \\ d_2 &= \sin\left(\theta_2\right) \left(f_1 - E\right) + \cos\left(\theta_2\right) f_0 \\ d_3 &= \sin\left(B \, \pi \left(d_2\right)^C\right) \\ g_1 &= d_1 - A \left(\left|d_3\right|\right)^D \end{aligned} \tag{8}$$

where c_0, f_0 and f_1 are the same as in (7). The auxiliary coefficients are

$$\begin{aligned} a = 40 \quad b &= 0.5 \quad c = 1 \quad d = 2 \quad e = -2 \\ A = 40 \quad B &= 2.0 \quad C = 1 \quad D = 6 \quad E = \;\; 0 \end{aligned} \tag{9}$$

and

$$\theta_1 = 0.1 \, \pi \qquad \theta_2 = -0.05 \, \pi \tag{10}$$

The results are presented in Fig. 9 where it can be observed that there are 'islands' of feasible and unfeasible regions and that the optimal front is discontinuous. The proposed code seems to also work well in this case.

Each problem is run 1000 times (samples) generating data for the statistical analysis. Once again, T_{sys} is the computer (system) time spent in running all 1000 samples of one particular problem; hence considering the 1000×500 loops.

The statistics with respect to error (see Part I [1]) are presented in Table II. It can be observed that the accuracy is reasonable and the repeatability characteristics are quite good for all tests. By visually inspecting the spread after a number of runs, it can be also concluded that the optimiser produces a good spread of points along the optimal Pareto front.

TABLE II
CONSTRAINED TWO OBJECTIVE PROBLEMS.

P	settings	settings/info	error	histogram ($N_{samples} = 1000$)
TNK	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 1m48.688s$	$E_{min} = 3.84264 \cdot 10^{-3}$ $E_{ave} = 6.18744 \cdot 10^{-3}$ $E_{max} = 3.28255 \cdot 10^{-2}$ $E_{dev} = \mathbf{1.6509 \cdot 10^{-3}}$	[-0.05,-0.03) | 0 [-0.03,-0.01) | 0 [-0.01,0.01) | 984 ###################### [0.01,0.03) | 15 # [0.03,0.05) | 1 # [0.05,0.06) | 0 [0.06,0.08) | 0 count = 1000
CTP1	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 2m57.587s$	$E_{min} = 6.26927 \cdot 10^{-4}$ $E_{ave} = 1.77085 \cdot 10^{-3}$ $E_{max} = 8.57021 \cdot 10^{-3}$ $E_{dev} = \mathbf{5.9251 \cdot 10^{-4}}$	[-0.05,-0.03) | 0 [-0.03,-0.02) | 0 [-0.02,-0.00) | 0 [-0.00,0.01) | 1000 ###################### [0.01,0.03) | 0 [0.03,0.04) | 0 [0.04,0.06) | 0 count = 1000
CTP2	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 3m3.308s$	$E_{min} = 9.09529 \cdot 10^{-4}$ $E_{ave} = 2.29671 \cdot 10^{-3}$ $E_{max} = 1.75664 \cdot 10^{-2}$ $E_{dev} = \mathbf{1.1425 \cdot 10^{-3}}$	[-0.05,-0.03) | 0 [-0.03,-0.02) | 0 [-0.02,0.00) | 20 # [0.00,0.02) | 980 ###################### [0.02,0.03) | 0 [0.03,0.05) | 0 [0.05,0.07) | 0 count = 1000
CTP3	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 2m57.239s$	$E_{min} = 1.60425 \cdot 10^{-3}$ $E_{ave} = 5.46932 \cdot 10^{-3}$ $E_{max} = 1.88064 \cdot 10^{-2}$ $E_{dev} = \mathbf{2.1154 \cdot 10^{-3}}$	[-0.05,-0.03) | 0 [-0.03,-0.01) | 0 [-0.01,0.00) | 5 # [0.00,0.02) | 995 ###################### [0.02,0.04) | 0 [0.04,0.05) | 0 [0.05,0.07) | 0 count = 1000
CTP4	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 3m1.658s$	$E_{min} = 1.01057 \cdot 10^{-2}$ $E_{ave} = 3.04897 \cdot 10^{-2}$ $E_{max} = 1.09641 \cdot 10^{-1}$ $E_{dev} = \mathbf{9.8181 \cdot 10^{-3}}$	[-0.04,-0.01) | 0 [-0.01,0.02) | 62 ## [0.02,0.05) | 880 ###################### [0.05,0.07) | 54 ## [0.07,0.10) | 3 # [0.10,0.13) | 1 # [0.13,0.16) | 0 count = 1000
CTP5	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 3m0.661s$	$E_{min} = 1.65470 \cdot 10^{-3}$ $E_{ave} = 5.53119 \cdot 10^{-3}$ $E_{max} = 1.35005 \cdot 10^{-2}$ $E_{dev} = \mathbf{1.6121 \cdot 10^{-3}}$	[-0.05,-0.03) | 0 [-0.03,-0.02) | 0 [-0.02,0.00) | 0 [0.00,0.02) | 1000 ###################### [0.02,0.03) | 0 [0.03,0.05) | 0 [0.05,0.06) | 0 count = 1000
CTP6	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 2m55.433s$	$E_{min} = 6.43497 \cdot 10^{-2}$ $E_{ave} = 1.19604 \cdot 10^{-1}$ $E_{max} = 3.52878 \cdot 10^{-1}$ $E_{dev} = \mathbf{2.8841 \cdot 10^{-2}}$	[0.01,0.07) | 4 # [0.07,0.13) | 656 ###################### [0.13,0.18) | 306 ########## [0.18,0.24) | 29 # [0.24,0.29) | 3 # [0.29,0.35) | 1 # [0.35,0.40) | 1 # count = 1000
CTP7	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 3m10.85s$	$E_{min} = 4.09044 \cdot 10^{-3}$ $E_{ave} = 6.29365 \cdot 10^{-3}$ $E_{max} = 5.41844 \cdot 10^{-2}$ $E_{dev} = \mathbf{4.2813 \cdot 10^{-3}}$	[-0.05,-0.02) | 0 [-0.02,-0.00) | 0 [-0.00,0.02) | 990 ###################### [0.02,0.04) | 3 # [0.04,0.06) | 7 # [0.06,0.08) | 0 [0.08,0.10) | 0 count = 1000
CTP8	$N_{sol} = 120$ $N_{cpu} = 4$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 2$ $C_{DE} = 0.1$ $N_{eval} = 60120$ $T_{sys} = 3m2.203s$	$E_{min} = 5.24050 \cdot 10^{-2}$ $E_{ave} = 9.14681 \cdot 10^{-2}$ $E_{max} = 6.46996 \cdot 10^{-1}$ $E_{dev} = \mathbf{3.4673 \cdot 10^{-2}}$	[0.00,0.10) | 785 ###################### [0.10,0.20) | 204 ###### [0.20,0.30) | 6 # [0.30,0.40) | 4 # [0.40,0.50) | 0 [0.50,0.60) | 0 [0.60,0.70) | 1 # count = 1000

III. TEST CASES: THREE OBJECTIVE

Nine three-objective optimisation problems are studied in this section. Eight tests are unconstrained (except for the limits on $\boldsymbol{x}$) and one has one constraint. As before, the number of iterations is $t_{max} = 500$. The number of trial solutions is set to $N_{sol} = 200$ for all tests in order to better depict the optimal Pareto front in the 3D space defined by (f_0, f_1, f_2).

The set of tests introduced in [4] are considered. These tests are known as *DTZLi*. In addition, the convex version of *DTZL2* presented in [5] (see also [6]) is solved. This version is named here *DTZL2x*. A constraint expression is added to *DTZL2* and the resulting test named *DTZL2c*. Finally, a modification to *DTLZi* is made by using the expressions of a superquadric thus defining the *SUQ1* and *SUQ2* tests. In all problems, the variables are limited according to

$$0 \leq x_i \leq 1 \quad (i \in [0, N_x]) \tag{11}$$

Problem DTZL1 has 7 unknowns and is defined by

$$\begin{aligned} c &= 100 \left\{ 5 + \sum_{i=2}^{6} \left[(x_i - 0.5)^2 - \cos\left(20\,\pi\,(x_i - 0.5)\right) \right] \right\} \\ f_0 &= x_0\, x_1\, (1 + c)/2 \\ f_1 &= x_0\, (1 - x_1)\, (1 + c)/2 \\ f_2 &= (1 - x_0)\, (1 + c)/2 \end{aligned} \tag{12}$$

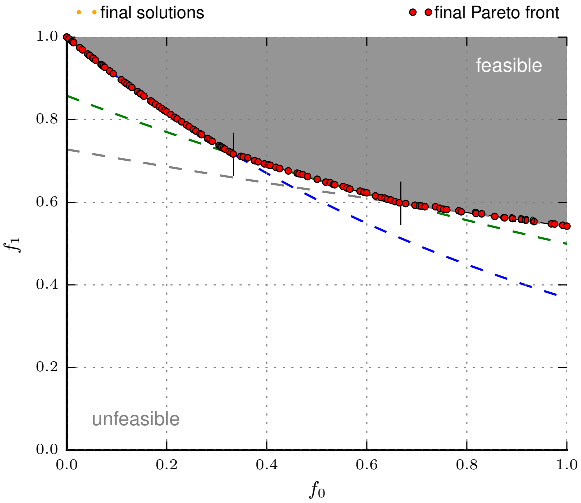


Fig. 2. CTP1: Results.

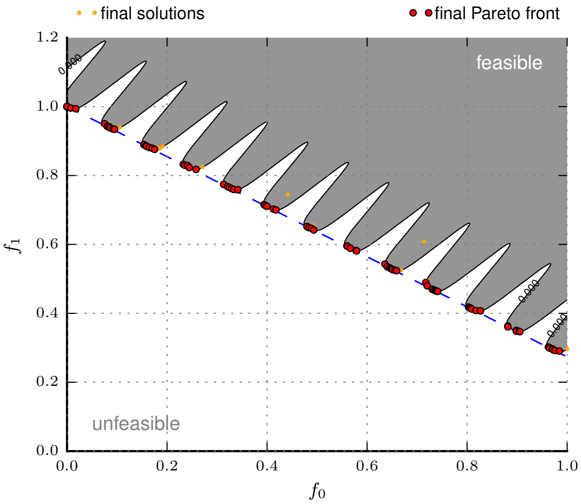


Fig. 3. CTP2: Results.

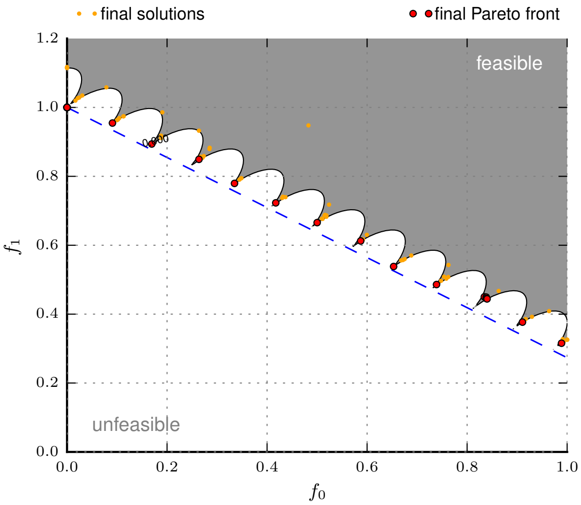


Fig. 4. CTP3: Results.

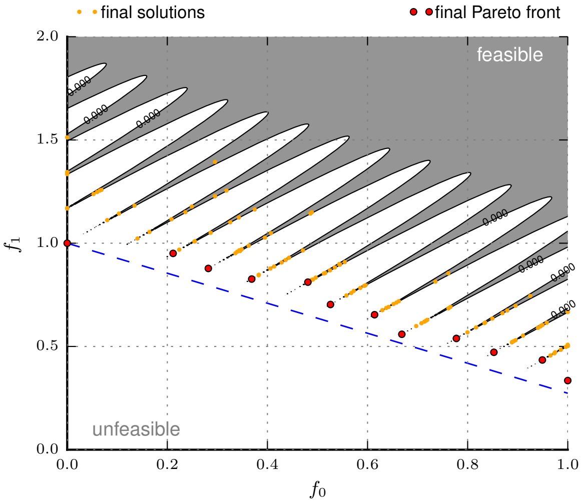


Fig. 5. CTP4: Results.

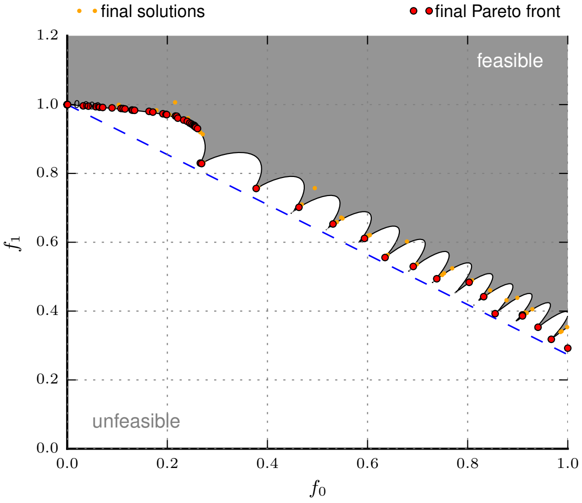


Fig. 6. CTP5: Results.

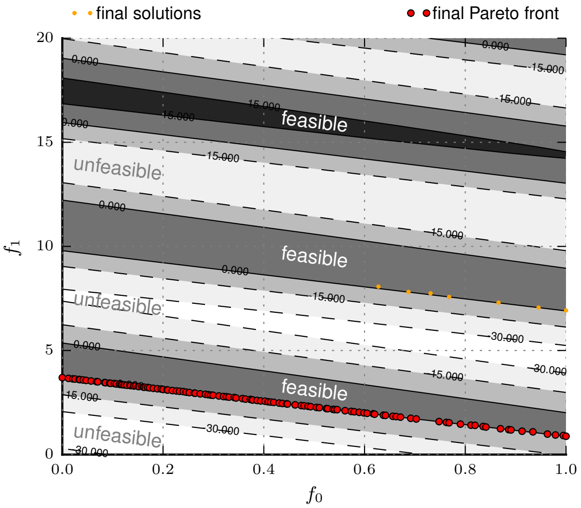


Fig. 7. CTP6: Results.

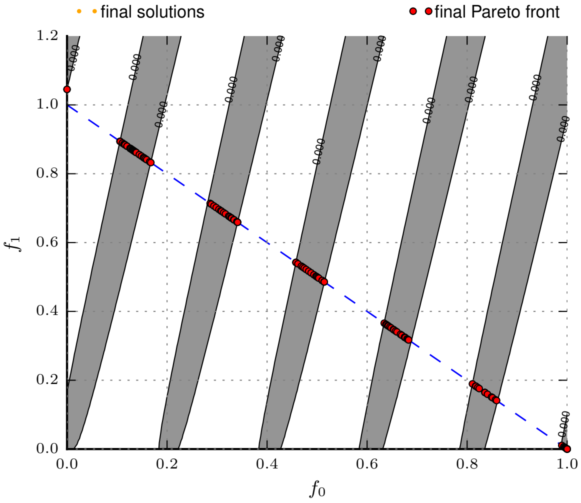


Fig. 8. CTP7: Results.

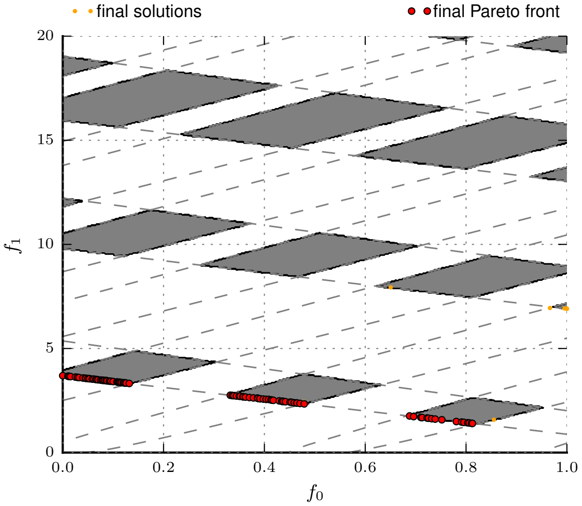


Fig. 9. CTP8: Results.

The optimal Pareto front is a flat surface described by

$$\psi(f_0, f_1, f_2) = f_0 + f_1 + f_2 - 0.5 \tag{13}$$

The results of one run are illustrated in Figs. 10 and 11 where it can be seen that the trial solutions (spheres) are close to the optimal front indicated by the plane.

Problem DTZL2 has 12 unknowns and is defined by

$$\begin{aligned} c &= \textstyle\sum_{i=2}^{11} (x_i - 0.5)^2 \\ f_0 &= (1+c)\,\cos(0.5 x_0 \pi)\,\cos(0.5 x_1 \pi) \\ f_1 &= (1+c)\,\cos(0.5 x_0 \pi)\,\sin(0.5 x_1 \pi) \\ f_2 &= (1+c)\,\sin(0.5 x_0 \pi) \end{aligned} \tag{14}$$

The optimal Pareto front is the surface of a sphere described by

$$\psi(f_0, f_1, f_2) = f_0^2 + f_1^2 + f_2^2 - 1 \tag{15}$$

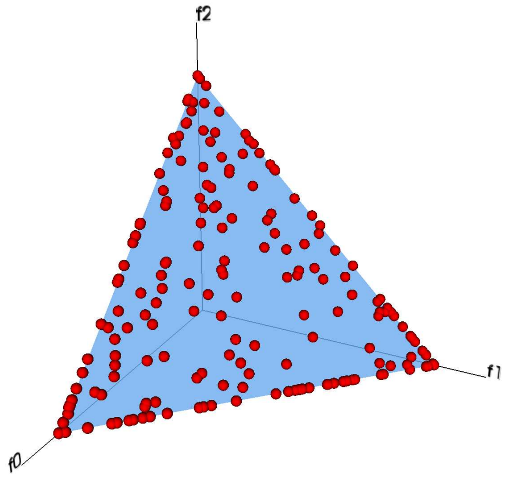


Fig. 10. DTLZ1: Pareto-optimal front (view 1).

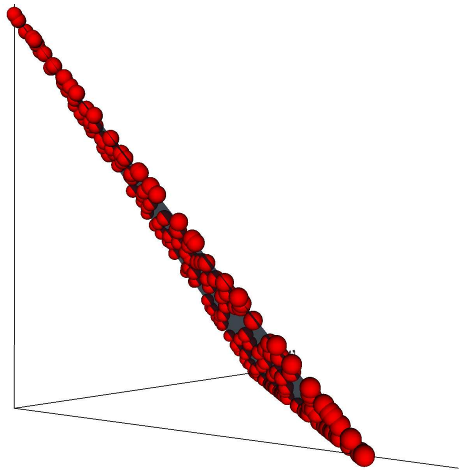


Fig. 11. DTLZ1: Pareto-optimal front (view 2).

The solutions are illustrated in Fig. 12 and 13 and also allow to conclude that the results are reasonable—a statistical analysis is presented later on.

Problem DTLZ3 has 12 unknowns, is similar to the previous problem; but is more difficult because it has a number of local fronts parallel to the global one. The problem is defined by

$$\begin{aligned} c &= 100\left\{10 + \sum_{i=2}^{11}\left[(x_i - 0.5))^2 - \cos\left(20\,\pi\,(x_i - 0.5)\right)\right]\right\} \\ f_0 &= (1+c)\,\cos(0.5 x_0 \pi)\,\cos(0.5 x_1 \pi) \\ f_1 &= (1+c)\,\cos(0.5 x_0 \pi)\,\sin(0.5 x_1 \pi) \\ f_2 &= (1+c)\,\sin(0.5 x_0 \pi) \end{aligned} \tag{16}$$

where the optimal front is also the surface of a sphere as in DTLZ2. The results are very similar to the ones from DTLZ2.

Problem DTLZ4 has 12 unknowns and is similar to DTLZ2; however with a bias in the search space that could cause solutions to be concentrated at planes normal to the coordinates

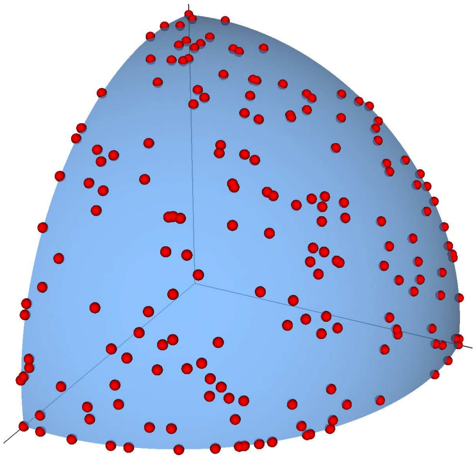


Fig. 12. DTLZ2: Pareto-optimal front (view 1).

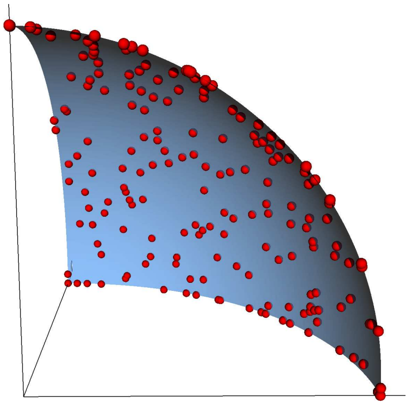


Fig. 13. DTLZ2: Pareto-optimal front (zoom).

axes. The problem is defined by

$$\begin{aligned} c &= \textstyle\sum_{i=2}^{11}(x_i-0.5)^2 \\ f_0 &= (1+c)\,\cos\left(0.5x_0^{\alpha}\pi\right)\,\cos\left(0.5x_1^{\alpha}\pi\right) \\ f_1 &= (1+c)\,\cos\left(0.5x_0^{\alpha}\pi\right)\,\sin\left(0.5x_1^{\alpha}\pi\right) \\ f_2 &= (1+c)\,\sin\left(0.5x_0^{\alpha}\pi\right) \end{aligned} \tag{17}$$

with $\alpha = 100$ and the optimal front is the surface of a sphere as in DTL2. The results are also similar to the ones from DTLZ2.

Problem DTLZ2x has 12 unknowns and is a convex version to DTLZ2—an exact convex counterpart is given in SUQ1 below. The problem is defined by

$$\begin{aligned} c &= \textstyle\sum_{i=2}^{11}(x_i-0.5)^2 \\ f_0 &= \left[(1+c)\,\cos\left(0.5x_0\pi\right)\,\cos\left(0.5x_1\pi\right)\right]^4 \\ f_1 &= \left[(1+c)\,\cos\left(0.5x_0\pi\right)\,\sin\left(0.5x_1\pi\right)\right]^4 \\ f_2 &= \left[(1+c)\,\sin\left(0.5x_0\pi\right)\right]^2 \end{aligned} \tag{18}$$

Note the exponents 4 and 2 that basically modify DTLZ2. The optimal Pareto front is described by

$$\psi(f_0, f_1, f_2) = \|f_0\|^{0.5} + \|f_1\|^{0.5} + f_2 - 1 \tag{19}$$

and the results are illustrated in Fig. 14 and 15 where a good convergence can be observed.

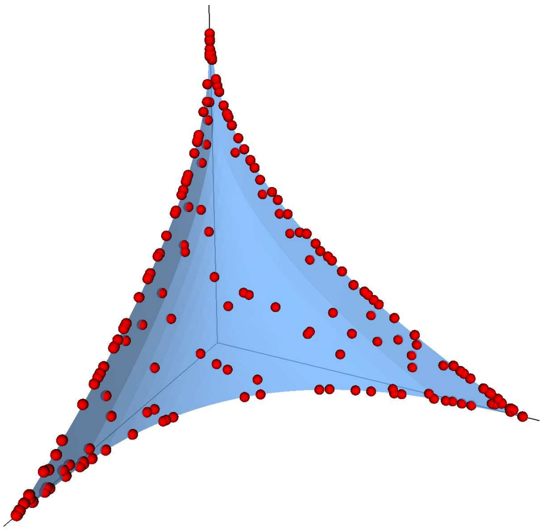


Fig. 14. DTLZ2x: Pareto-optimal front (view 1; f_2 pointing up).

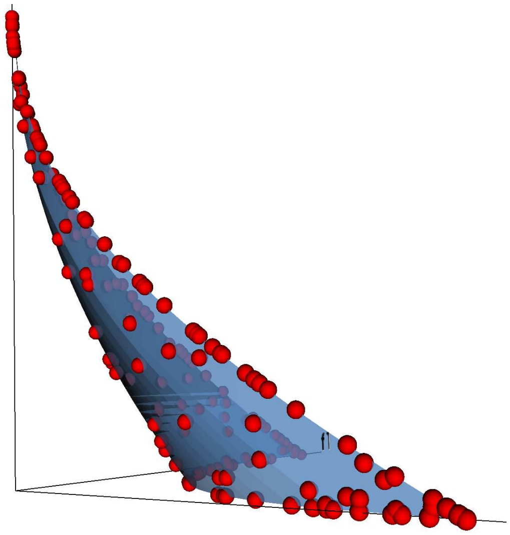


Fig. 15. DTLZ2x: Pareto-optimal front (view 2; f_2 pointing up).

Problem DTLZ2c is the same as DTLZ2 with the addition of one inequality constraint function. This function ($g_0(\boldsymbol{x})$) defines a cone in the f_i space that is aligned with its diagonal. The constraint is defined by

$$g_0 = \tan\alpha - \frac{\sqrt{(f_0-f_1)^2+(f_1-f_2)^2+(f_2-f_0)^2}}{f_0+f_1+f_2} \tag{20}$$

where α equals half the cone's opening angle. Here, $\alpha = 15$ has been selected. In this way, the feasible solutions must lie on a circular patch of the optimal front. The results are illustrated in Figs. 16 and 17.

Problems SUQi: A superquadric version of DTLZ2 is now introduced. The expressions in 14 are used, except that the sine and cosine functions are replaced by the following functions

$$\mathrm{sinX}\left(w;m\right) = \mathrm{sign}\left(\sin w\right)|\sin w|^{m} \tag{21}$$

and

$$\mathrm{cosX}\left(w;m\right) = \mathrm{sign}\left(\cos w\right)|\cos w|^{m} \tag{22}$$

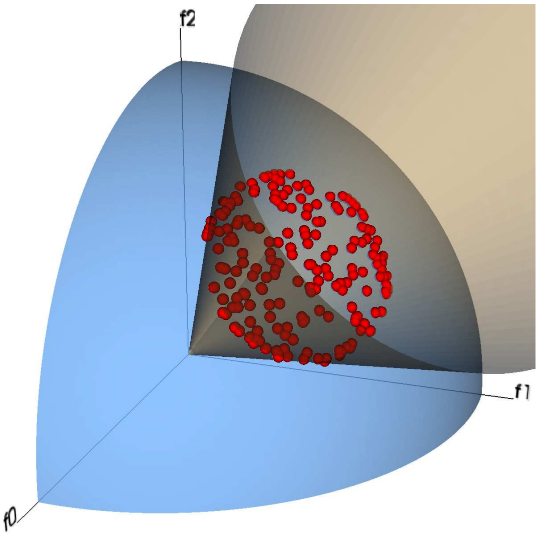


Fig. 16. DTLZ2c: Pareto-optimal front (view 1).

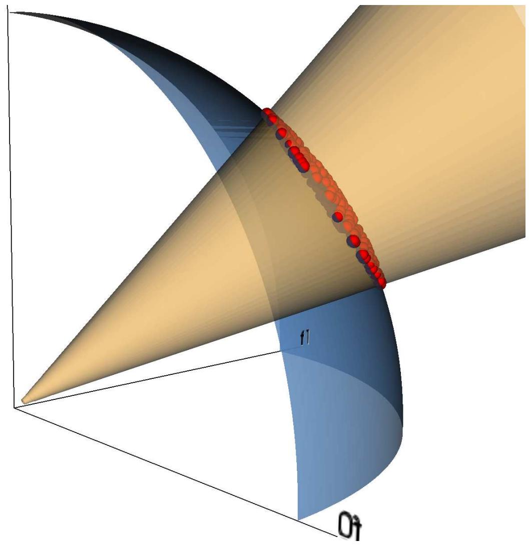


Fig. 17. DTLZ2c: Pareto-optimal front (view 2).

respectively. In this way, the *SUQi* problems are defined by

$$\begin{aligned} c &= \textstyle\sum_{i=2}^{11}(x_i-0.5)^2 \\ f_0 &= (1+c)\ \mathrm{cosX}\,(0.5x_0\pi;2/a)\ \mathrm{cosX}\,(0.5x_1\pi;2/a) \\ f_1 &= (1+c)\ \mathrm{cosX}\,(0.5x_0\pi;2/b)\ \mathrm{sinX}\,(0.5x_1\pi;2/b) \\ f_2 &= (1+c)\ \mathrm{sinX}\,(0.5x_0\pi;2/c) \end{aligned} \tag{23}$$

where a, b and c are the coefficients of the superquadric. The error related to the Pareto-optimal front is then

$$\psi(f_0,f_1,f_2)=|f_0|^a+|f_1|^b+|f_2|^c-1 \tag{24}$$

The above expressions are quite versatile because they can produce concave, convex and many kinds of twisted surfaces in the space of objective values. Views of the Pareto-optimal fronts and numerical results are shown in Figs. 18 and 19 for SUQ1 and in Figs. 20 and 21 for SUQ2 where it can be concluded that the results are satisfactory (error and spread).

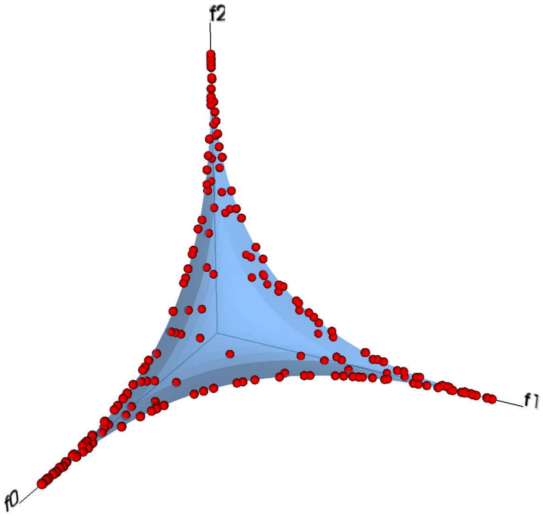


Fig. 18. SUQ1: Pareto-optimal front (view 1).

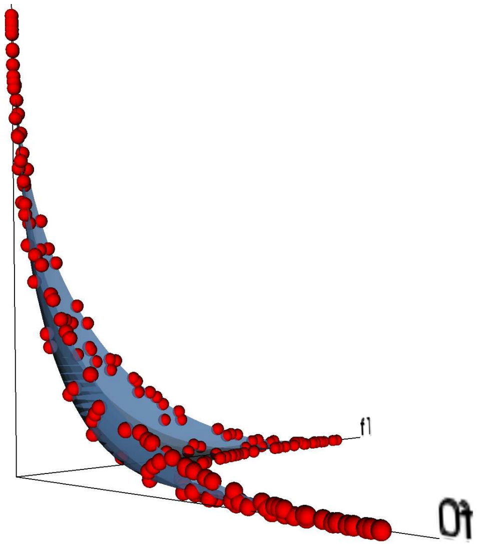


Fig. 19. SUQ1: Pareto-optimal front (view 2).

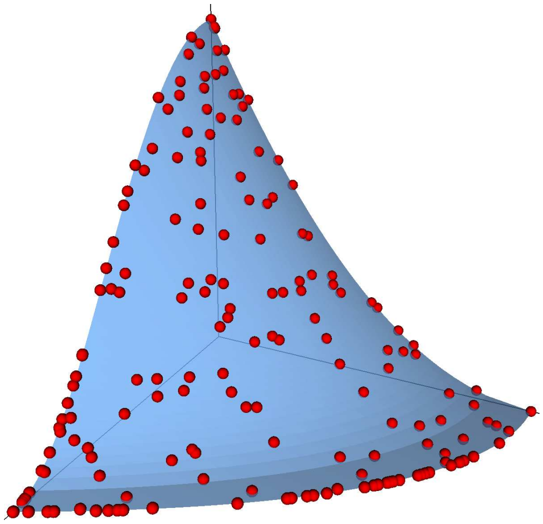


Fig. 20. SUQ2: Pareto-optimal front (view 1).

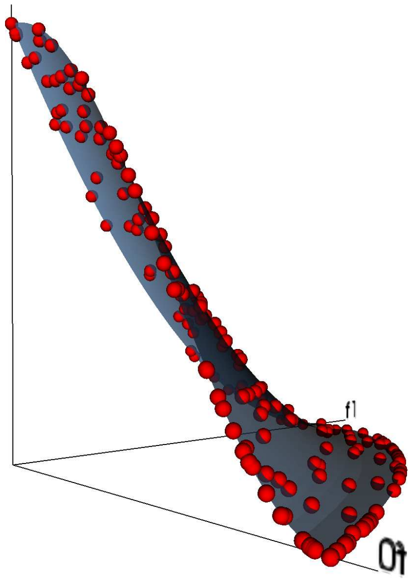
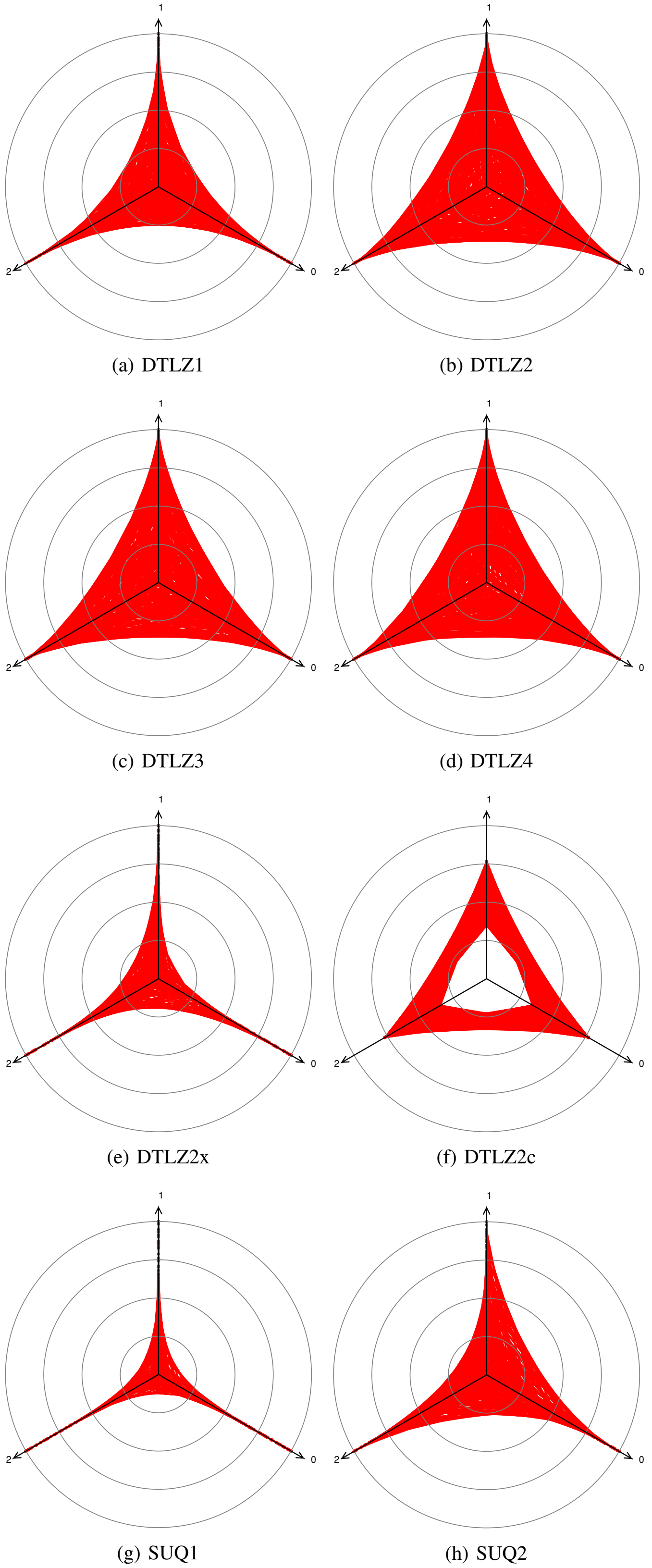


Fig. 21. SUQ2: Pareto-optimal front (view 2).

A visualisation tool is developed in order to assist on verifying the ability of the code to cover all objective functions. This tool, called here *starplot*, is quite simple and nothing more than a plot of the normalised objective values in a polar coordinates graph. For each feasible trial solution with $\phi = 0$ (best Pareto-optimal front), a line plot is drawn by connecting the f_i values. The axes for the f_i values are drawn as in a star with equal angles. The angle between axes is hence $\Delta\theta = 2\,\pi / N_f$. Because the f_i points are connected, an area is created in the diagram. For reference, four circles are drawn; each corresponding to a normalised objective value of 0.25, 0.5, 0.75 and 1.0. The normalisation factor is equal to 1.0 for all tests, except for DTLZ1 where it is equal to 0.5. A good coverage of objective functions is hence indicated by lines reaching the outer circles with higher frequency.

The starplots for all DTLZi and SUQi tests are shown in Fig. 22 where it can be observed that the three objective values are well represented in all tests. Symmetry is observed in DTLZ1-4, DTLZ2c and SUQ1. The superquadric SUQ2 does not have any symmetry in particular. It is interesting to observe the difference between DTLZ2x (convex) and SUQ1 (superquadric) with respect to symmetry. In DTLZ2c, the region of values is restricted because of the constraint. In this case, the combination of f_i values must be over 0.3 and smaller than a value close to 0.75.

To assess the accuracy and repeatability of results, 1000 samples are run for each problem and the statistics with respect to the root mean square error are computed. The results are collected in Table III. It can be observed that the errors and standard deviations reached machine precision with the exception of DTLZ3 which is a little harder due to the local optima. With more iterations, the error is reduced. The constraint in DTLZ2c does not cause problems to the solver; remember that the algorithm [1] prioritises satisfaction of constraints first hence trial solutions are 'pulled' into the conical space.



(a) DTLZ1

(b) DTLZ2

(c) DTLZ3

(d) DTLZ4

(e) DTLZ2x

(f) DTLZ2c

(g) SUQ1

(h) SUQ2

Fig. 22. DTLZi and SQUi: Results (starplots).

TABLE III
UNCONSTRAINED AND CONSTRAINED THREE OBJECTIVE PROBLEMS.

P	settings	settings/info	error	histogram ($N_{samples} = 1000$)
DTLZ1	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 4m52.233s$	$E_{min} = 0$ $E_{ave} = 1.6570 \cdot 10^{-15}$ $E_{max} = 1.2953 \cdot 10^{-14}$ $E_{dev} = \mathbf{1.6528 \cdot 10^{-15}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000
DTLZ2	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 6m0.016s$	$E_{min} = 3.6796 \cdot 10^{-16}$ $E_{ave} = 5.8872 \cdot 10^{-16}$ $E_{max} = 5.7672 \cdot 10^{-14}$ $E_{dev} = \mathbf{2.2489 \cdot 10^{-15}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000
DTLZ3	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 6m34.574s$	$E_{min} = 2.9022 \cdot 10^{-6}$ $E_{ave} = 1.3281 \cdot 10^{-3}$ $E_{max} = 2.7062 \cdot 10^{-1}$ $E_{dev} = \mathbf{1.1339 \cdot 10^{-2}}$	[-0.05,0.00) | 948 ##################### [0.00,0.06) | 50 ## [0.06,0.11) | 0 [0.11,0.16) | 0 [0.16,0.22) | 0 [0.22,0.27) | 1 # [0.27,0.32) | 1 # count = 1000
DTLZ4	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 6m14.143s$	$E_{min} = 3.7208 \cdot 10^{-16}$ $E_{ave} = 5.0316 \cdot 10^{-16}$ $E_{max} = 1.9920 \cdot 10^{-14}$ $E_{dev} = \mathbf{7.4111 \cdot 10^{-16}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000
DTLZ2x	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 5m56.295s$	$E_{min} = 3.7276 \cdot 10^{-16}$ $E_{ave} = 5.1045 \cdot 10^{-16}$ $E_{max} = 1.7755 \cdot 10^{-14}$ $E_{dev} = \mathbf{6.8437 \cdot 10^{-16}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000
DTLZ2c	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 5m57.551s$	$E_{min} = 4.2624 \cdot 10^{-16}$ $E_{ave} = 7.9684 \cdot 10^{-16}$ $E_{max} = 3.8322 \cdot 10^{-14}$ $E_{dev} = \mathbf{1.7491 \cdot 10^{-15}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000
SUQ1	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 6m6.578s$	$E_{min} = 1.2471 \cdot 10^{-16}$ $E_{ave} = 2.3090 \cdot 10^{-16}$ $E_{max} = 2.7153 \cdot 10^{-14}$ $E_{dev} = \mathbf{1.0141 \cdot 10^{-15}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000
SUQ2	$N_{sol} = 200$ $N_{cpu} = 5$ $t_{max} = 500$ $\Delta t_{exc} = 50$	$N_f = 3$ $C_{DE} = 0.01$ $N_{eval} = 100200$ $T_{sys} = 5m57.774s$	$E_{min} = 2.0478 \cdot 10^{-16}$ $E_{ave} = 2.8488 \cdot 10^{-16}$ $E_{max} = 7.3056 \cdot 10^{-15}$ $E_{dev} = \mathbf{2.5706 \cdot 10^{-16}}$	[-0.05,-0.04) | 0 [-0.04,-0.02) | 0 [-0.02,-0.01) | 0 [-0.01,0.01) | 1000 ##################### [0.01,0.02) | 0 [0.02,0.04) | 0 [0.04,0.05) | 0 count = 1000

IV. TEST CASES: MULTI OBJECTIVE

A final set of tests is carried out prior to presenting some applications. Now, problem DTLZ2 is re-analysed with more than 3 objectives. The following numbers of are selected: $N_f \in \{5, 7, 10, 15, 20\}$. The number of unknowns is also increased according to

$$N_x = N_f + 10 \tag{25}$$

Thus, $N_x \in \{15, 17, 20, 35\}$. The number of trial solutions is kept constant though: $N_{sol} = 200$ and 4 groups are employed. It is clear that the problem with more objectives and unknowns are more difficult to be solved. The problems are identified as *DTLZ2mI* where I means the number of objectives.

The starplots corresponding to the DTLZ2mI tests are illustrated in Fig. 23 where it can be observed that all objective values are well attained in most tests; i.e. a good coverage is observed. As an exception, tests with 13 or more objective values have less values reaching 1. Note that the number of trial solutions was kept constant and equal to 200. With a few more iterations (t_{max}) and number of unknowns, the response is improved. This has not been done here for the sake of showing how the difficulty increases with the number of objective functions.

With regards to the statistics of errors, Table IV presents the results for 1000 samples. As shown in this table, it is remarkable the ability of the proposed evolutionary algorithm to obtain very small errors every time it is run. It is possible to observe that the error increases in a nonlinear trend with the number of objective functions; this is plotted in Fig. 24. It is actually interesting to see that the rate decreases considering that the number of trial solutions and iterations was kept constant. This means that the performance is not bat at all.

TABLE IV
UNCONSTRAINED MANY OBJECTIVE PROBLEMS.

<table>
<tr><th>P</th><th>settings</th><th>settings/info</th><th>error</th><th>histogram ($N_{samples} = 1000$)</th></tr>
<tr><td>DTLZ2m5</td><td>$N_{sol} = 200$
$N_{cpu} = 4$
$t_{max} = 500$
$\Delta t_{exc} = 50$</td><td>$N_f = 5$
$C_{DE} = 0.01$
$N_{eval} = 100200$
$T_{sys} = 14m45.197s$</td><td>$E_{min} = 1.4183 \cdot 10^{-13}$
$E_{ave} = 2.3364 \cdot 10^{-12}$
$E_{max} = 4.2463 \cdot 10^{-10}$
$E_{dev} = \mathbf{1.5261 \cdot 10^{-11}}$</td><td>[-0.05,-0.04) | 0
[-0.04,-0.02) | 0
[-0.02,-0.01) | 0
[-0.01,0.01) | 1000 #####################
[0.01,0.02) | 0
[0.02,0.04) | 0
[0.04,0.05) | 0
count = 1000</td></tr>
<tr><td>DTLZ2m7</td><td>$N_{sol} = 200$
$N_{cpu} = 4$
$t_{max} = 500$
$\Delta t_{exc} = 50$</td><td>$N_f = 7$
$C_{DE} = 0.01$
$N_{eval} = 100200$
$T_{sys} = 16m51.432s$</td><td>$E_{min} = 1.4620 \cdot 10^{-11}$
$E_{ave} = 1.8510 \cdot 10^{-10}$
$E_{max} = 7.1792 \cdot 10^{-9}$
$E_{dev} = \mathbf{4.9185 \cdot 10^{-10}}$</td><td>[-0.05,-0.04) | 0
[-0.04,-0.02) | 0
[-0.02,-0.01) | 0
[-0.01,0.01) | 1000 #####################
[0.01,0.02) | 0
[0.02,0.04) | 0
[0.04,0.05) | 0
count = 1000</td></tr>
<tr><td>DTLZ2m10</td><td>$N_{sol} = 200$
$N_{cpu} = 4$
$t_{max} = 500$
$\Delta t_{exc} = 50$</td><td>$N_f = 10$
$C_{DE} = 0.01$
$N_{eval} = 100200$
$T_{sys} = 19m59.056s$</td><td>$E_{min} = 3.2089 \cdot 10^{-9}$
$E_{ave} = 4.5553 \cdot 10^{-8}$
$E_{max} = 4.6091 \cdot 10^{-6}$
$E_{dev} = \mathbf{1.8920 \cdot 10^{-7}}$</td><td>[-0.05,-0.04) | 0
[-0.04,-0.02) | 0
[-0.02,-0.01) | 0
[-0.01,0.01) | 1000 #####################
[0.01,0.02) | 0
[0.02,0.04) | 0
[0.04,0.05) | 0
count = 1000</td></tr>
<tr><td>DTLZ2m13</td><td>$N_{sol} = 200$
$N_{cpu} = 4$
$t_{max} = 500$
$\Delta t_{exc} = 50$</td><td>$N_f = 13$
$C_{DE} = 0.01$
$N_{eval} = 100200$
$T_{sys} = 24m0.127s$</td><td>$E_{min} = 1.9350 \cdot 10^{-7}$
$E_{ave} = 2.7087 \cdot 10^{-6}$
$E_{max} = 6.5875 \cdot 10^{-4}$
$E_{dev} = \mathbf{2.1491 \cdot 10^{-5}}$</td><td>[-0.05,-0.04) | 0
[-0.04,-0.02) | 0
[-0.02,-0.01) | 0
[-0.01,0.01) | 1000 #####################
[0.01,0.02) | 0
[0.02,0.04) | 0
[0.04,0.05) | 0
count = 1000</td></tr>
<tr><td>DTLZ2m15</td><td>$N_{sol} = 200$
$N_{cpu} = 4$
$t_{max} = 500$
$\Delta t_{exc} = 50$</td><td>$N_f = 15$
$C_{DE} = 0.01$
$N_{eval} = 100200$
$T_{sys} = 25m23.117s$</td><td>$E_{min} = 1.7667 \cdot 10^{-6}$
$E_{ave} = 3.2496 \cdot 10^{-5}$
$E_{max} = 1.2908 \cdot 10^{-2}$
$E_{dev} = \mathbf{4.1277 \cdot 10^{-4}}$</td><td>[-0.05,-0.03) | 0
[-0.03,-0.02) | 0
[-0.02,-0.00) | 0
[-0.00,0.01) | 1000 #####################
[0.01,0.03) | 0
[0.03,0.05) | 0
[0.05,0.06) | 0
count = 1000</td></tr>
<tr><td>DTLZ2m20</td><td>$N_{sol} = 200$
$N_{cpu} = 4$
$t_{max} = 500$
$\Delta t_{exc} = 50$</td><td>$N_f = 20$
$C_{DE} = 0.01$
$N_{eval} = 100200$
$T_{sys} = 30m25.228s$</td><td>$E_{min} = 1.2063 \cdot 10^{-4}$
$E_{ave} = 1.0556 \cdot 10^{-3}$
$E_{max} = 1.6046 \cdot 10^{-2}$
$E_{dev} = \mathbf{1.6438 \cdot 10^{-3}}$</td><td>[-0.05,-0.03) | 0
[-0.03,-0.02) | 0
[-0.02,-0.00) | 0
[-0.00,0.02) | 1000 #####################
[0.02,0.03) | 0
[0.03,0.05) | 0
[0.05,0.07) | 0
count = 1000</td></tr>
</table>

V. APPLICATION: TRUSS SHAPE AND TOPOLOGY OPTIMISATION

The proposed code seems to work well over a range of situations. To further investigate its performance and the combination of real and integer representations, the problem of optimising trusses in structural engineering is solved in this section. In particular, the efficiency of the parallel code is demonstrated.

The problem is known as shape and topology optimisation of trusses made up of slender rods (bars). The optimal cross sectional area (shape) of rods and the connections (topology) among them such that equilibrium is guaranteed are to be found. Two objectives that can be considered are: (1) minimum deflection when the structure is under loads; and (2) the overall minimum weight. Better satisfying one goal induces a failure in satisfying the other one. Hence a Pareto optimal front is to be determined. The structure has to be stable as well; therefore limits on maximum stresses must be followed.

The example studied in [2], [7]–[10] and illustrated in Fig. 25 is considered. The Young's modulus of all bars is $E = 10^4$ [units of pressure] and the density is $\rho = 0.1$ [units of density]. In this figure, the structure is known as *ground* structure because it serves as a basis for the generation of other combinations. It is important to note that a minimum number of joints (nodes) and bars (elements) must be present at all times in order to allow the structure to carry the prescribed load. In this particular problem, the fixed nodes and the nodes with applied loads must be present.

A simple expression is available to assess the degree of *mobility* in case a load-carrying structure cannot be established. This expression can be used to filter out probable unfeasible trial solutions and is given by

$$M = 2\,n - m - d \tag{26}$$

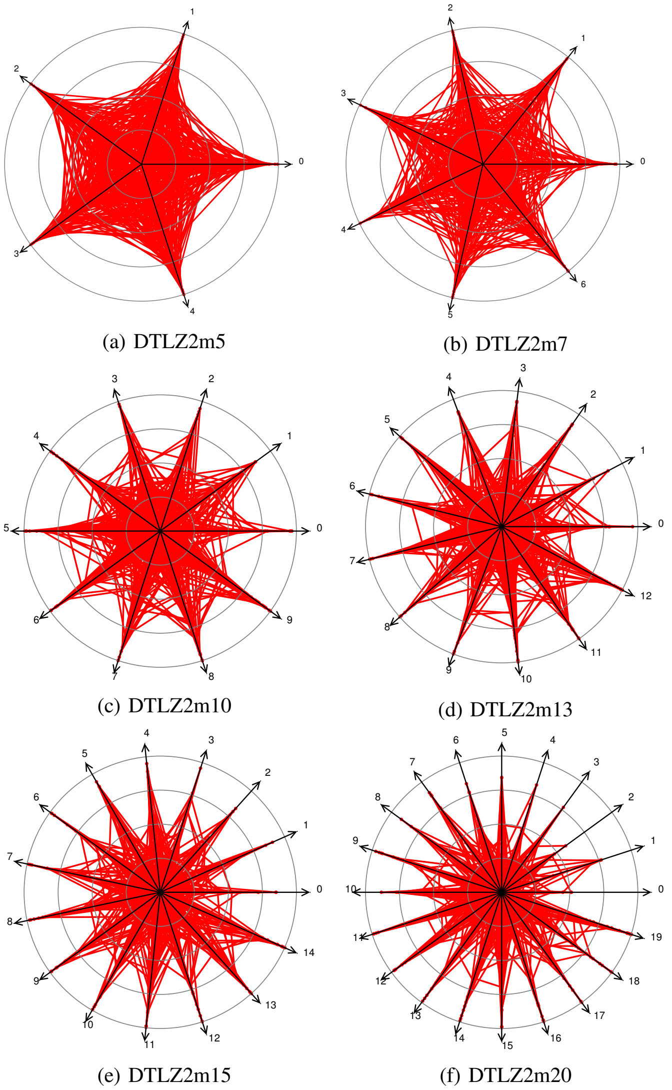
where M is the degree of mobility (positive values mean mobile), n is the number of nodes in the structure, m is the number of (active) rods, and d is the number of fixed degrees of freedom (e.g. 4 here: $2 \times \{u_x, u_y\}$).

Integers are used for the topology description (connection of bars) and real numbers for the cross sectional areas. In this case, integers simply serve as Boolean variables where 1 means a connection is active and 0 otherwise. There are 10 bars (10 areas); thus the number of real numbers $\underset{\cdot}{\boldsymbol{x}}$ is $N_x = 10$ and the number of integers $\underset{\cdot}{\boldsymbol{y}}$ is $N_y = 10$. The limits are

$$0.09 \le x_i \le 35 \quad i \in [0, 9] \tag{27}$$

i.e. the minimum and maximum areas are 0.09 and 35 [units of area], respectively.

The objective values (f_0: total weight; f_1: deflection) are computed after running a finite element (FEM) simulation to calculate the maximum deflection $\delta = |u_y^{max}|$ at $x = 720$ and $y = 0$. This value is also limited by a maximum allowance



(a) DTLZ2m5 (b) DTLZ2m7

(c) DTLZ2m10 (d) DTLZ2m13

(e) DTLZ2m15 (f) DTLZ2m20

Fig. 23. DTLZ2mI: Results (starplots).

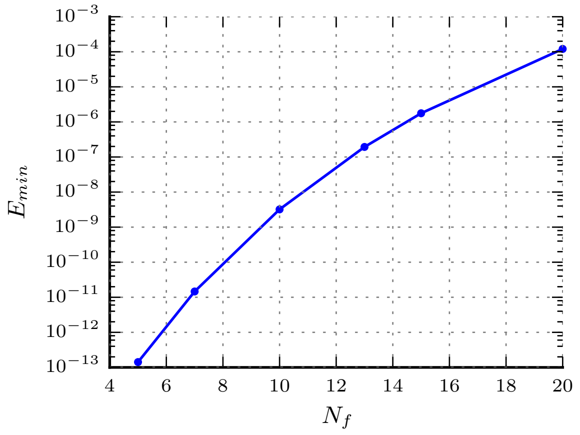


Fig. 24. DTLZ2mI: Increase of error with number of objectives.

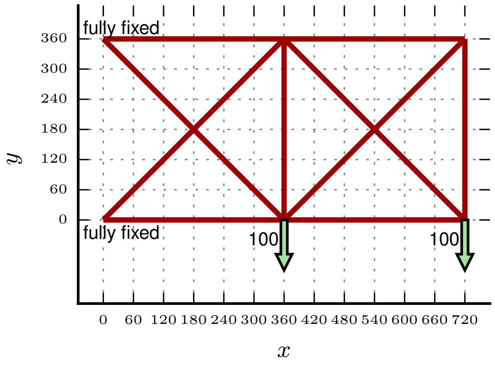


Fig. 25. "Ground" mesh with all 10 rods active.

of $\delta_{awd} = 5.6$ [units of displacement]. The maximum allowed tensile or compressive axial stress σ_{max} (from FEM) at any bar is $\sigma_{awd} = 35$ [units of pressure]. The bar lengths are L_i.

Considering the general optimisation problem equation in Part I, the required expressions are (with $f_i = f_i(\underset{\bullet}{\boldsymbol{x}}, \underset{\bullet}{\boldsymbol{y}})$ and $u_i = u_i(\underset{\bullet}{\boldsymbol{x}}, \underset{\bullet}{\boldsymbol{y}})$)

$$\begin{aligned}
f_0 &= \sum_{i=0}^{N_{active}} \rho\, x_i\, L_i \\
f_1 &= \delta \\
u_0 &= \langle M \rangle \\
u_1 &= 1 \text{ if FEM failed; 0 otherwise} \\
u_2 &= \langle \delta - \delta_{awd} \rangle \\
u_3 &= \langle |\sigma_{max}| - \sigma_{awd} \rangle
\end{aligned} \tag{28}$$

i.e. there are four constraints (out-of-range/OOR) functions u_i. Therein, $\langle x \rangle$ is the ramp function defined by

$$\langle x \rangle = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases} \tag{29}$$

The use of the OOR functions is quite convenient. Since the comparison between trial solutions first consider the number of constraint violations N_{viol}, the OOR functions can be set in a way such that the worse case gets higher N_{viol} by assigning greater than zero values to u_i (e.g. 1.0). Note also that the OORs are compared using the `ParetoComparison`. Therefore, this allows a strategy to rank trial solutions that cannot be even used in the FE simulation. The strategy is implemented as detailed in Algorithm 1.

Simple crossover and mutation rules for integers are employed [11] with probability $P_c = 0.5$ and $P_m = 0.01$, respectively. The number of cuts during crossover is 1 and the number of bit changes during mutation is 1. $N_{sol} = 200$, $t_{max} = 500$ and $\Delta t_{exc} = 50$ are selected. The default differential evolution coefficient value $C_{DE} = 0.8$ is employed. The number of groups N_{cpu} is varied from 1 to 16. Simulations are carried out in a 16-core Intel(R) Xeon(R) CPU E5-2687W @ 3.10GHz (Debian-Sid/GNU/Linux).

A typical Pareto-optimal front obtained with goga is shown in Fig. 26. A reference line from [2] is also shown in blue. Some examples of trusses are drawn near the weight-deflection

Algorithm 1. `RunFEA`($\underset{\cdot}{x}$, $\underset{\cdot}{y}$) runs finite element analysis and returns OVA and OOR values.

```
Input: trial solution: Areas=$\underset{\cdot}{x}$, EnabledConnections=$\underset{\cdot}{y}$
Output: $f_i$ and $u_i$
compute $n$ = number of nodes
set connectivity based on $\underset{\cdot}{y}$
initialise FEA stage
check for required vertices
if not all required vertices are present then
    $u_0,\ u_1,\ u_2,\ u_3 \leftarrow 1+2\,n,\ 1,\ 1,\ 1$
    return
compute mobility $M$
if $M > 0$ then
    $u_0,\ u_1,\ u_2,\ u_3 \leftarrow M,\ 1,\ 1,\ 1$
    return
set elements' cross-sectional areas
compute total weight $W$
run FE analysis (FEA)
if FEA failed then
    $u_0,\ u_1,\ u_2,\ u_3 \leftarrow 0,\ 1,\ 1,\ 1$
    return
extract maximum deflection from FEA
extract maximum magnitude of stress from FEA
$f_0,\ f_1 \leftarrow W,\ \delta$
$u_0,\ u_1,\ u_2,\ u_3 \leftarrow 0,\ 0,\ \langle\delta-\delta_{awd}\rangle,\ \langle|\sigma_{max}|-\sigma_{awd}\rangle$
```

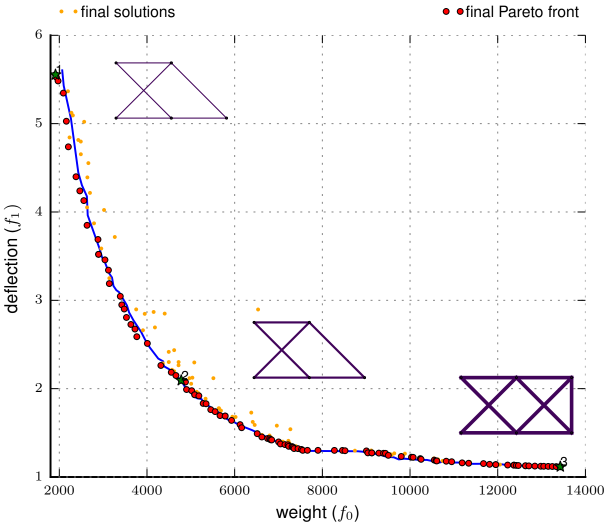


Fig. 26. Shape and topology optimisation. Results.

point (marked with a green star). The thickness of the line is proportional to the optimal cross sectional area. The results match the reference ones reasonably well and feature some spread.

The efficiency of the code can be assessed with the aid of Fig. 27 where the speedup (S_{up}) behaviour is illustrated. This quantity is computed by means of

$$S_{up}(N_{cpu}) = \frac{T^1_{sys}}{T^{N_{cpu}}_{sys}} \tag{30}$$

where T^1_{sys} is the computer time using one group (i.e. CPU) and $T^{N_{cpu}}_{sys}$ is the computer time using N_{cpu} groups. The real computer time is also shown in the figure by means of a gray line corresponding to the right-hand-side scale; the maximum time is around 40 seconds, for instance.

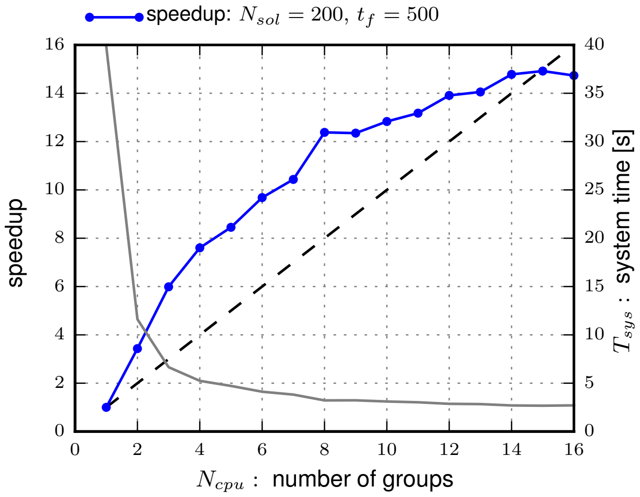


Fig. 27. Shape and topology optimisation. Efficiency.

It can be clearly observed that the speedup is better than ideal up to 14 CPUs. The reason for this excellent behaviour is in part due to a faster execution of the non-optimised implementation of the `Metrics` routine. This routine needs to perform several comparisons between trial solutions to find the closest neighbours in addition to determine the Pareto front indices. For example, by disabling the FEM computations, a slightly similar behaviour is observed where a very high speedup happens as well. Regardless, the parallel implementation is very helpful and the efficiency will only improve by using a better metrics routine (e.g. with a binary search or octree algorithms).

VI. Application: Economic Emission Load Dispatch

Economic emission load dispatch (or environmental/economic dispatch EED) is an interesting multi objective optimisation problem of great importance in power generation. The problem involves the design of optimal power flow observing the cost of operation and the emission of pollutants. These two objectives are contradictory and a number of constraints such as the maximum capacity of generators and the total power balance must be satisfied. The problem has been studied over several years; for example, some papers from 1987 to 2015 are: [12]–[28].

The fuel cost f_0 of a thermal unit is approximated by a quadratic function according to

$$f_0 = \sum_{i=0}^{N-1} \left(a_i + b_i\,P_i + c_i\,P_i^2\right) \tag{31}$$

where N is the number of generators and $x_i := P_i$ is the (unknown) active output power of generator i. Therein, a_i, b_i and c_i are fitting parameters.

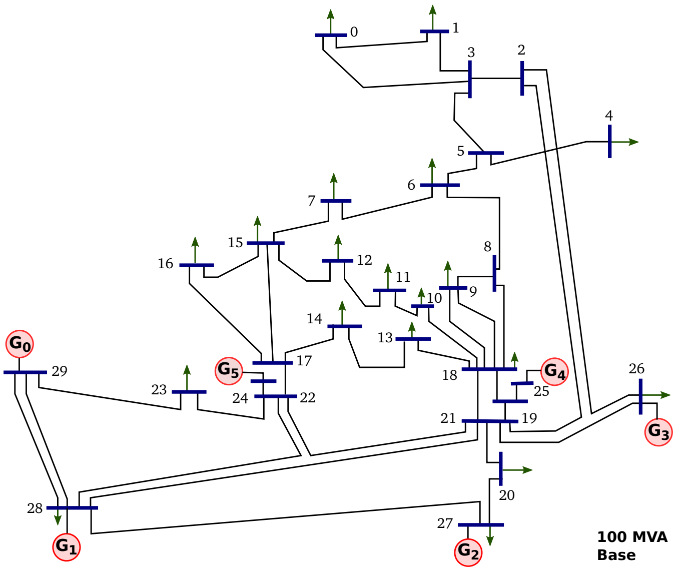


Fig. 28. IEEE 30 bus test system.

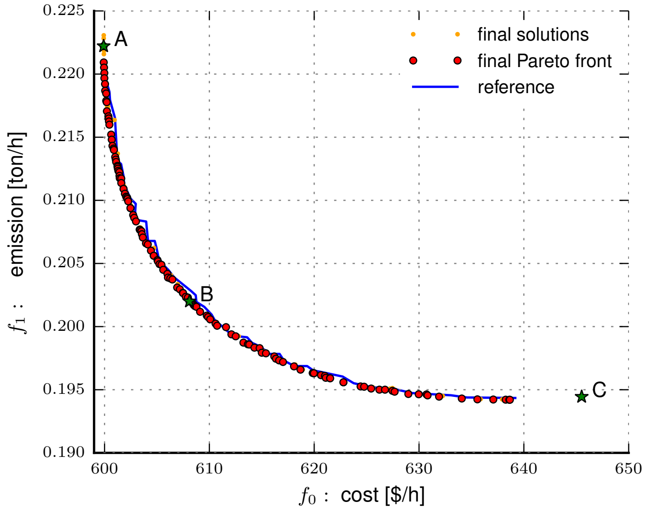


Fig. 29. Economic emission load dispatch. Lossless.

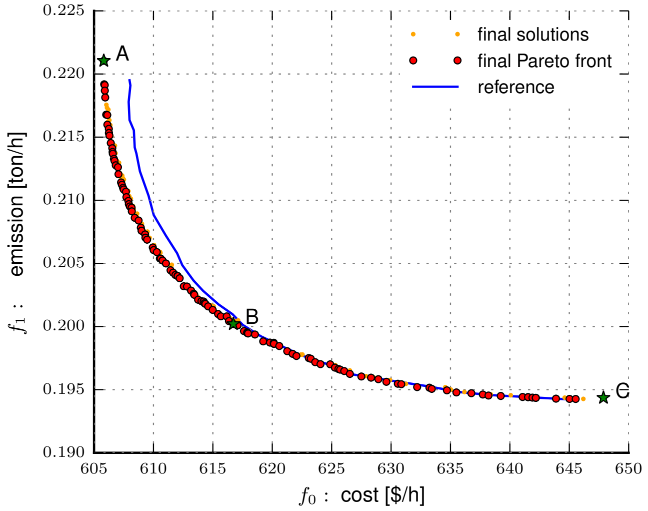


Fig. 30. Economic emission load dispatch. Lossy case.

The corresponding emission (e.g. of Nitrogen-Oxide NOx) f_1 is also modelled as a function of the power output by means of

$$f_1 = \sum_{i=0}^{N-1} \left[\alpha_i + \beta_i\, P_i + \gamma_i\, P_i^2 + \zeta_i\, \exp\left(\lambda_i\, P_i\right)\right] \tag{32}$$

where α_i, β_i, γ_i, ζ_i and λ_i are fitting parameters.

The power output P_i is limited by the generation capacity and an equality constraint h_0 must be taken into account. This constraint is given by the power balance involving the total power generation $\sum P_i$ of the system, the total load demand P_{demand}, and the total transmission loss P_{loss} as in

$$h_0 = \left(\sum_{i=0}^{N-1} P_i\right) - P_{demand} - P_{loss} \equiv 0 \tag{33}$$

The transmission loss P_{loss} is a nonlinear function of the power outputs P_i and requires the solution of the power flow equations. It depends on other variables such as the bus voltage magnitudes and angles. Nonetheless, a simple expression is available to approximate P_{loss} as follows

$$P_{loss} = B_{00} + \sum_{i=0}^{N-1} B_{0i}\, P_i + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_i\, B_{ij}\, P_j \tag{34}$$

The economic-emission dispatch of the IEEE 30 Bus Test Case is studied here. The sketch is illustrated in Fig. 28 and is based on [12]. The required fitting parameters are listed in Table V including the capacity limits. These values are based on data from [12] and [14] (note that there is a typo error in [14]: the α coefficient for his G_4 in Table 1).

The demand load is $P_{demand} = 2.834$ and the B coefficients related to IEEE 30 bus system are listed in Table VI; see e.g. [29]–[31]

Two cases are considered: a lossless and a lossy situation. In the first one, P_{loss} is set to zero making the problem a little easier. The solution is obtained with $N_{sol} = 200$, $t_{max} = 500$, $N_{cpu} = 4$ and $\Delta t_{exc} = 50$. In addition, the control for the equality constraint h_0 is $\epsilon_h = 10^{-3}$ and the differential evolution coefficient is $C_{DE} = 0.8$ (default value).

The Pareto-optimal front for the first case is illustrated in Fig. 29 where it can be observed that the results match well the reference solution from [15]. The front for the second case is illustrated in Fig. 30 where the reference solution from [15] is slightly less optimal.

Points corresponding to the minimum cost, a compromise between cost and emission, and the minimum emission are indicated in these figures by A, B and C, respectively. The point values are listed in Table VII for the lossless case and in Table VIII for the lossy case. It is interesting to see that the cost is higher in the lossy case which is reasonable. The balance errors are also listed in the tables under the h_0 column. The errors fall within the range specified by ϵ_h as required.

TABLE V
IEEE 30 BUS: GENERATORS PARAMETERS.

G	cost			emission					output limits	
	a	b	c	α	β	γ	ζ	λ	P_i^{min}	P_i^{max}
0	10	200	100	$4.091 \cdot 10^{-2}$	$-5.554 \cdot 10^{-2}$	$6.490 \cdot 10^{-2}$	$2 \cdot 10^{-4}$	2.857	0.05	0.5
1	10	150	120	$2.543 \cdot 10^{-2}$	$-6.047 \cdot 10^{-2}$	$5.638 \cdot 10^{-2}$	$5 \cdot 10^{-4}$	3.333	0.05	0.6
2	20	180	40	$4.258 \cdot 10^{-2}$	$-5.094 \cdot 10^{-2}$	$4.586 \cdot 10^{-2}$	$1 \cdot 10^{-6}$	8.000	0.05	1.0
3	10	100	60	$5.326 \cdot 10^{-2}$	$-3.550 \cdot 10^{-2}$	$3.380 \cdot 10^{-2}$	$2 \cdot 10^{-3}$	2.000	0.05	1.2
4	20	180	40	$4.258 \cdot 10^{-2}$	$-5.094 \cdot 10^{-2}$	$4.586 \cdot 10^{-2}$	$1 \cdot 10^{-6}$	8.000	0.05	1.0
5	10	150	100	$6.131 \cdot 10^{-2}$	$-5.555 \cdot 10^{-2}$	$5.151 \cdot 10^{-2}$	$1 \cdot 10^{-5}$	6.667	0.05	0.6

TABLE VI
IEEE 30 BUS: B COEFFICIENTS.

B_{00}					
0.00098573					
B_{0i}					
−0.0107	0.0060	−0.0017	0.0009	0.0002	0.0030
B_{ij}					
0.1382	−0.0299	0.0044	−0.0022	−0.0010	−0.0008
−0.0299	0.0487	−0.0025	0.0004	0.0016	0.0041
0.0044	−0.0025	0.0182	−0.0070	−0.0066	−0.0066
−0.0022	0.0004	−0.0070	0.0137	0.0050	0.0033
−0.0010	0.0016	−0.0066	0.0050	0.0109	0.0005
−0.0008	0.0041	−0.0066	0.0033	0.0005	0.0244

TABLE VII
ECONOMIC EMISSION DISPATCH: LOSSLESS CASE.

point	cost	emission	P_0	P_1	P_2	P_3	P_4	P_5	h_0
A	599.9131	0.222215	0.117399	0.303156	0.522061	1.020305	0.519831	0.350272	$9.7579 \cdot 10^{-4}$
B	608.0962	0.201999	0.236179	0.370779	0.530887	0.715596	0.550713	0.429157	$6.8968 \cdot 10^{-4}$
C	645.5148	0.194445	0.423533	0.478256	0.538390	0.325404	0.528290	0.539496	$6.3076 \cdot 10^{-4}$

TABLE VIII
ECONOMIC EMISSION DISPATCH: LOSSY CASE.

point	cost	emission	P_0	P_1	P_2	P_3	P_4	P_5	h_0
A	605.8174	0.221047	0.121975	0.285915	0.573947	0.999557	0.523837	0.353772	$8.3726 \cdot 10^{-4}$
B	616.7220	0.200202	0.246116	0.372354	0.554951	0.665870	0.568470	0.452289	$1.2542 \cdot 10^{-4}$
C	647.8771	0.194360	0.437091	0.468949	0.551263	0.381422	0.560829	0.471015	$8.6251 \cdot 10^{-5}$

VII. CONCLUSIONS

This paper confirms the ability of the proposed evolutionary algorithm (EA) to solve a wide range of multiple optimisation problems. Several test cases are studied in order to observe the performance. The tests include constrained two objective problems, three objective problems and cases with up to 20 objectives. The tests were already presented and studied in other research papers and form a useful basis for investigations. These include the CTPi and the DTLZi sets. A small addition to the set is presented by employing a superquadric in DTLZ2. This allows the generation of many convex, concave and twisted Pareto-optimal fronts.

The results exhibit good accuracy and excellent repeatability characteristics. For example, almost all tests (excluding CTP8 and DTLZ3) give nearly identical results after 1000 runs. Some tests even achieve machine precision. Starplots are drawn to illustrate how well all objective functions are covered. With these plots, it is shown that most tests have a good coverage of objectives although, as the number of objective functions increases, the coverage decreases as do accuracy. If the number of trial solutions and/or iterations is increased, this situation is improved.

The selection of the differential evolution coefficient C_{DE} is arbitrary. The default value of 0.8 is suggested although a reduction of this value should be tried with a couple of runs because it may improve the results as it did for CTPi and DTLZi tests.

Two applications are studied: topology optimisation of trusses and the economic emission dispatch. For the first, it is demonstrated that the use of the out-of-range functions help with the handling of not only constraints but also failures of parts of the code that are required to compute the objective values. In this case, this part is a call to an external library to perform a finite element analysis that may fail if the input data is not adequate. The second application involves an equality constraint (the power balance) that makes the problem a little more challenging. It is demonstrated that the proposed code works well with both applications.

By running the topology optimisation problem with different numbers of groups, it is observed an ideal speedup where the computer time is dramatically reduced; hence illustrating the great advantage of using parallel computing in EAs.

Acknowledgment

The support from the Australian Research Council under grant DE120100163 is gratefully acknowledged.

References

[1] D. M. Pedroso, "Parallel evolutionary algorithm for constrained single and multi objective optimisation - Part I: Algorithm, single and two-objective test cases," *IEEE Trans. Evol. Comput.*, vol. -, no. -, pp. –, 2016, submitted.

[2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[3] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, "Ga-based decision support system for multicriteria optimization," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 2, Oct 1995, pp. 1556–1561 vol.2.

[4] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multi-objective Optimization*, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer, 2005, pp. 105–145.

[5] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[6] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.

[7] W. S. Ruy, Y. S. Yang, G. H. Kim, and Y. S. Yeun, "Topology design of truss structures in a multicriteria environment," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 4, pp. 246–258, 2001.

[8] C.-Y. Wu and K.-Y. Tseng, "Truss structure optimization using adaptive multi-population differential evolution," *Structural and Multidisciplinary Optimization*, vol. 42, no. 4, pp. 575–590, 2010.

[9] N. Noilublao and S. Bureerat, "Simultaneous Topology, Shape, and Sizing Optimisation of Plane Trusses with Adaptive Ground Finite Elements Using MOEAs," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–9, 2013.

[10] R. Cazacu and L. Grama, "Steel Truss Optimization Using Genetic Algorithms and FEA," *Procedia Technology*, vol. 12, pp. 339–346, 2014.

[11] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley, 1989.

[12] R. Yokoyama, S. H. Bae, T. Morita, and H. Sasaki, "Multiobjective Optimal Generation Dispatch Based on Probability Security Criteria." *IEEE Trans. on Power Systems*, vol. 3, no. 1, pp. 317–324, 1987.

[13] A. Farag, S. Al-Baiyat, and T. C. Cheng, "Economic load dispatch multiobjective optimization procedures using linear programming techniques," *IEEE Trans. on Power Systems*, vol. 10, no. 2, pp. 731–738, 1995.

[14] M. a. Abido, "A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch," *Int. Journal of Electrical Power and Energy Systems*, vol. 25, no. 2, pp. 97–105, 2003.

[15] M. Abido, "Multiobjective evolutionary algorithms for electric power dispatch problem," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 315–329, 2006.

[16] M. AlRashidi and M. El-hawary, "Economic Dispatch with Environmental Considerations using Particle Swarm Optimization," *Large Engineering Systems Conf. on Power Engineering*, pp. 41–46, 2006.

[17] S. Kumar and R. Naresh, "Nonconvex economic load dispatch using an efficient real-coded genetic algorithm," *Applied Soft Computing*, vol. 9, no. 1, pp. 321–329, 2009.

[18] S. Dhanalakshmi, S. Kannan, K. Mahadevan, and S. Baskar, "Application of modified NSGA-II algorithm to Combined Economic and Emission Dispatch problem," *Int. Journal of Electrical Power and Energy Systems*, vol. 33, no. 4, pp. 992–1002, 2011.

[19] L. Bayón, J. M. Grau, M. M. Ruiz, and P. M. Suárez, "The Exact Solution of the Environmental / Economic Dispatch Problem," *IEEE Trans. on Power Systems*, vol. 27, no. 2, pp. 723–731, 2012.

[20] N. A. Rahmat, I. Musirin, and A. F. Abidin, "Differential Evolution Immunized Ant Colony Optimization (DEIANT) technique in solving economic emission dispatch," *Int. Conf. on Techn., Informatics, Management, Engrg., and Environment (TIME-E)*, pp. 198–202, 2013.

[21] Y.-F. Li, N. Pedroni, and E. Zio, "A Memetic Evolutionary Multi-Objective Optimization Method for Environmental Power Unit Commitment," *IEEE Trans. on Power Systems*, vol. 28, no. 3, pp. 2660–2669, 2013.

[22] A. Jubril, O. Komolafe, and K. Alawode, "Solving multi-objective economic dispatch problem via semidefinite programming," *IEEE Trans. on Power Systems*, vol. 28, no. 3, pp. 2056–2064, 2013.

[23] H. Bilil, G. Aniba, and M. Maaroufi, "Probabilistic economic/environmental power dispatch of power system integrating renewable energy sources," *Sustainable Energy Technologies and Assessments*, vol. 8, pp. 181–190, 2014.

[24] B. Jeddi and V. Vahidinasab, "A modified harmony search method for environmental/economic load dispatch of real-world power systems," *Energy Conversion and Management*, vol. 78, pp. 661–675, 2014.

[25] A. Jubril, O. Olaniyan, O. Komolafe, and P. Ogunbona, "Economic-emission dispatch problem: A semi-definite programming approach," *Applied Energy*, vol. 134, pp. 446–455, 2014.

[26] S. Sayah, A. Hamouda, and A. Bekrar, "Efficient hybrid optimization approach for emission constrained economic dispatch with nonsmooth cost curves," *Int. Journal of Electrical Power & Energy Systems*, vol. 56, no. 0, pp. 127–139, 2014.

[27] K. Bhattacharjee, A. Bhattacharya, and S. Halder, "Electrical Power and Energy Systems Backtracking search optimization based economic environmental power dispatch problems," *Int. Journal of Electrical Power and Energy Systems*, vol. 73, pp. 830–842, 2015.

[28] H. Zhang, D. Yue, X. Xie, S. Hu, and S. Weng, "Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch," *Applied Soft Computing*, vol. 34, pp. 312–323, 2015.

[29] S. Özyön, H. Temurta, B. Durmu, and G. Kuvat, "Charged system search algorithm for emission constrained economic power dispatch problem," *Energy*, vol. 46, no. 1, pp. 420–430, 2012.

[30] B. Pal, P. Biswas, and A. Mukhopadhyay, "Using genetic algorithm to goal programming model of solving economic-environmental electric power generation problem with interval-valued target goals," in *Mathematical Modelling and Scientific Computation*. Springer, 2012, vol. 283, pp. 156–169.

[31] L. Wang and C. Singh, "Balancing risk and cost in fuzzy economic dispatch including wind power penetration based on particle swarm optimization," *Electric Power Systems Research*, vol. 78, no. 8, pp. 1361–1368, 2008.

Dorival Pedroso See Part I for Bio.