

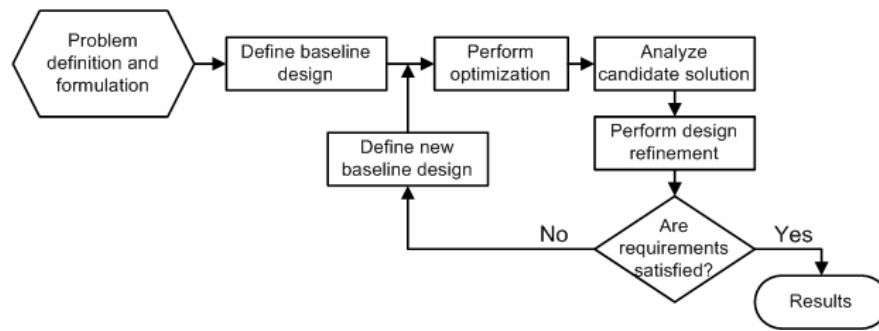


Figure 5.4. Optimization included in the design cycle.

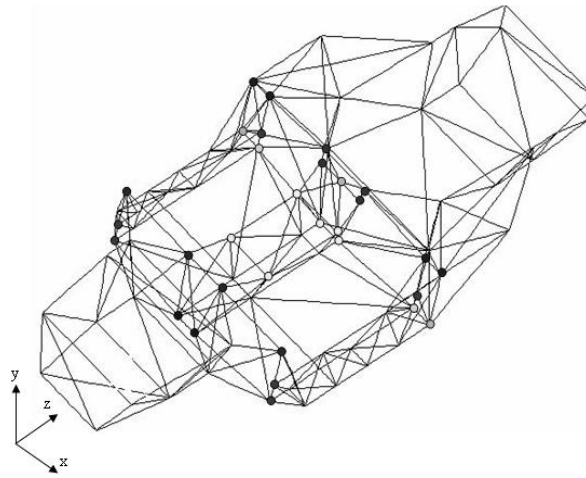


Figure 5.5. Nodal points for optimization.

Table 5.2. Possible values of the external diameter.

Design variable values	1	2	3	4	5
Corresponding diameter $\left[\times 10^{-3}m\right]$	38.10	41.27	44.45	47.60	50.80

Four different formulations were tested in the first “optimization” step of the design cycle. Table 5.3 gives the description of each of the four formulations. Static penalties and weighted sum methods were used to handle the constrained optimization, and the multi-objective problems, respectively.

Table 5.3. Formulation of the optimization problem.

Scenario	Description	Formulation
#1	Optimization problem is defined as the minimization of a dummy objective function, while both the mass and the stiffness are considered as design constraints.	Objective: $J(\mathbf{x}) = 0$ Mass constraint: $G_1(\mathbf{x}) = \frac{M}{M_{limit}} - 1.0 \leq 0$ Stiffness constraint: $G_2(\mathbf{x}) = 1.0 - \frac{K}{K_{limit}} \leq 0$
#2	Mass reduction is an objective function while stiffness increase is taken as a constraint.	Mass objective: $F(\mathbf{x}) = \frac{M}{M_{limit}}$ Stiffness constraint: $G(\mathbf{x}) = 1.0 - \frac{K}{K_{limit}} \leq 0$
#3	Stiffness increase is an objective while mass reduction is a constraint.	Stiffness objective: $F(\mathbf{x}) = -\frac{K}{K_{limit}}$ Mass constraint: $G(\mathbf{x}) = \frac{M}{M_{limit}} - 1.0 \leq 0$
#4	Multi-objective optimization: mass reduction and stiffness increase.	Mass objective: $F_1(\mathbf{x}) = \frac{M}{M_{limit}}$ Stiffness objective: $F_2(\mathbf{x}) = -\frac{K}{K_{limit}}$

3. Results and Discussion

Table 5.4 contains the optimal results obtained in the first “optimization” step, in which all heuristic methods were set up with a population size of 50 and number of iterations of 50, $M_{limit} = 100 [Kg]$ and $K_{limit} = 1900 [Kgf \cdot m/deg]$. According to Table 5.4, although all algorithms performed well, considering both mass and torsional stiffness, PSO using formulation #1 showed to be the best choice. This way, this is the only configuration for the “optimization” step in a second iteration of the design cycle. Table 5.5 summarizes all information about the only two cycles required to achieve an optimal design. In the second iteration, PSO was set up with a population size of 100 and number of iterations of 100, $M_{limit} = 80 [Kg]$ and $K_{limit} = 2835 [Kgf \cdot m/deg]$.

The framework illustrated in Figure 5.4 showed to be capable of improving the baseline design in every cycle of optimization.

Table 5.4. Optimal results for the first step.

Method	Formulation	Mass		Torsional stiffness	
		Value [Kg]	%of reduction	Value [$Kgf \cdot m/deg$]	% of increasing
ACO	#1	107.24	23.1	1897.7	9.8
	#2	115.93	16.9	1937.8	12.1
	#3	100	28.3	1765.08	2.1
	#4	105.31	24.5	1937.8	12.1
GA	#1	107.96	22.6	1774.54	2.7
	#2	126.34	9.45	1937.8	12.1
	#3	100.03	28.3	1809.56	4.7
	#4	100	28.3	1937.8	12.1
PSO	#1	106.08	24	2106.25	21.9
	#2	110.01	21.2	1937.8	12.1
	#3	108.55	22.2	1933.4	11.9
	#4	105.63	24.3	1868	8.1

Table 5.5. Summary of the optimization process.

Design cycle iteration	Designs	Mass [Kg]	Torsional stiffness [$Kgf \cdot m/deg$]	DOF of the model
#1	Baseline	139.5	1728.1	19167
	Candidate solution	106.1	2106.3	18609
#2	New baseline	105.7	2266	18675
	New candidate solution	109	2880	18333
Final design refinement	Optimal	104	2810	17643
% Improvement (reduction for mass and increasing for stiffness)		25.5	62.6	---

4. Summary and Conclusions

This application demonstrated the use of heuristic optimization algorithms directly coupled with finite element models to solve the design optimization of a three-dimensional vehicular structure. The investigation showed the possibility of using the present techniques in real world environment. A combination of continuous and discrete design variables was

successfully handled by the algorithms. It is also important to say that the algorithms were able to handle constraint functions and performed satisfactorily in the context of multi-criteria optimization design. The results are very encouraging in the sense that other system requirements, embracing for example the gravity center, dynamic and manufacture constraints, will be included in the optimization problem in further research work.

5.1.1. Optimization of Aircraft Structural Components by using Heuristic Algorithms and Multi-Fidelity Approximations

This application explores the use of the multi-fidelity framework coupled with heuristic optimization methods in order to reduce the computational cost in the on-line phase of the optimization. In this work, a suite of nature-inspired optimization methods, namely ACO, GA, PSO, and LC, is applied in the design of a flat pressure bulkhead reinforced by an array of beams. Variable fidelity is proposed as compromise solution between numerical performance and computational cost. This allows faster linear analyses during the optimization; whilst a reduced set of expensive non-linear analyses are run “off-line”, enhancing the linear results according to the physics of the structure. Preliminary results of this application was reported in Viana et al. (2007d), the full report is in Viana et al. (2008d).

1. Component Description and Formulation

From the structural viewpoint, the pressurized cabin of a modern aircraft is a system of sealed pressure vessels containing an atmosphere near to that of sea level and its functional requirements include (Bruhn, 1973): (i) transmission of internal and external flight loads; (ii) necessity for non-structural cutouts such as doors and windows; (iii) shape efficiency for both aerodynamics and space allocation; and (iv) minimum structural weight. More specifically, devices called pressure bulkheads close the extremities of pressurized cabins. According to Niu and Niu (1999), pressure bulkheads should have a dome-like structure in preference to a flat one. It is also known, however, that some requirements, mostly related to space availability, impose the adoption of the flat geometry. This case study is although geared towards the non-ideal condition, which is a flat bulkhead. For a more realistic analysis, it should be considered that when the carrying of transverse loads entails large (more than a few tenths of the plate thickness) deflections in the same direction (see Figure 5.1), the plates deform into curved surfaces, giving rise to mid-surface stresses. Still, the edge supports resist the in-plane movements of the plates, leading to further membrane (mid-surface) action.

From the analysis point of view, all previous discussion justifies the use of non-linear finite element modeling to deal with such complex calculations. On the other hand, within a design optimization framework, several analyses are required iteratively and, due to their usually high computational cost, a feasibility obstacle can arise. In this case study, variable fidelity is used to overcome this issue. Faster linear analyses are employed during the optimization; and a reduced set of expensive non-linear analyses are run “off-line.”

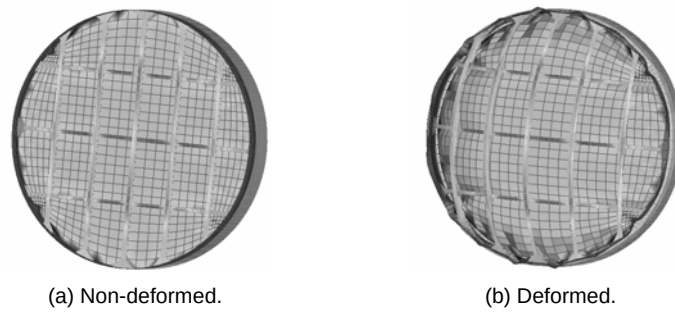


Figure 5.1. Displacement post-processing indicating large deflections at a pressurized flat bulkhead reinforced by beams.

2. Numerical Optimization of the Pressure Bulkhead

The optimization framework based on heuristic methods and on variable fidelity analyses, as described in Chapter IV, is used here. Table 5.1 to Table 5.3 contain useful design data, such as the general information about the pressure bulkhead; design variable descriptions and their respective boundary specifications; and finally, response descriptions and respective values for the baseline design and target design as well. All this information is important for the good understanding and definition of the optimization problem. It is worth mentioning that in Table 5.2, the normalization was performed to preserve interests in the technological content of this work.

The sets of four design variables and five responses that describe the pressure bulkhead modeling are listed in Table 5.2 and Table 5.3, respectively. The target values were chosen to be 80% of those at the baseline design (since all of them are quantities to be minimized). The bulkhead mass is set to be 2.5 times more important than the other responses (i.e. the weight coefficients are $w_i = 1$, for $i = 1$ to 4 and $w_5 = 2.5$). The baseline responses are chosen as the worst accepted values, thus forcing the optimizer to move away from them in a clear attempt to improve the design.

Table 5.1. Pressure bulkhead description.

Web material	Al 2024
Number of horizontal reinforcement beams	3
Number of vertical reinforcement beams	5
Reinforcement beams material	Al 7050
Pressure differential	Equivalent to 30,000 ft

Table 5.2. Design variables values normalized by the bulkhead diameter (normalization makes the design variables non-dimensional).

Design variable	Lower bound	Baseline	Upper bound
x_1 : width of reinforcement beams	0.01	0.02	0.03
x_2 : height of reinforcement beams	0.040	0.05	0.06
x_3 : thickness of reinforcement beams	0.0008	0.0012	0.0016
x_4 : web thickness	0.0008	0.0012	0.0016

Table 5.3. Baseline design and target design response values.

Objectives	Baseline value	Target value
$f_1(\mathbf{x})$: ratio of ultimate tension stress at the reinforcement beams with respect to allowable	1.01	0.8
$f_2(\mathbf{x})$: ratio of ultimate compression stress at the reinforcement beams with respect to allowable	0.85	0.68
$f_3(\mathbf{x})$: web maximum displacement, normalized with respect to the bulkhead diameter	0.116	0.928
$f_4(\mathbf{x})$: maximum limit tension stress with respect to yield limit	0.19	0.15
$f_5(\mathbf{x})$: pressure bulkhead mass	22.56	18.05

For the sake of simplicity, the variable fidelity is implemented with a second order PRS model for the difference between the high and low fidelity analyses of responses $f_1(\mathbf{x})$ to $f_4(\mathbf{x})$. Since $f_5(\mathbf{x})$ is the pressure bulkhead mass, it does not require any correction (its fidelity does not depend on whether the calculations are linear or non-linear; eventually, small differences may appear due to either rounding or truncation).

In order to keep the computational cost low (in terms of low-fidelity simulations), instead of a multi-objective approach for generating the Pareto front, the responses are combined into a functional whose minimization implies that all responses tend to a target value and depart from a non-desirable one. This simplifies the solution of the problem and can be considered acceptable given the general scope of this work. *Compromise Programming* (discussed in Chapter II) was used:

$$J(\mathbf{x}) = \sqrt{\sum_{i=1}^{n_{obj}} \left(w_i \frac{(f_i(\mathbf{x}) - f_i^*(\mathbf{x}))}{f_i^{worst}(\mathbf{x}) - f_i^*(\mathbf{x})} \right)^2} \quad (0.1)$$

where:

- $J(\mathbf{x})$ is a compromise objective function.
- $f_i(\mathbf{x})$ is the i -th response of interest, in a total of n_{obj} .
- $f_i^*(\mathbf{x})$ is the target value of the i -th response.
- $f_i^{worst}(\mathbf{x})$ is the worst (avoidable) value accepted for the i -th response.
- w_i is the weighting factor applied for the i -th response of interest.

The target and avoidable values are not necessarily design goals, but play an important role at the optimization problem by defining the tendency of the desired optima with respect to the baseline design. In the context of the present work, it is a useful approach since the heuristic techniques are defined for unconstrained and single-objective problems only. During the optimization task, each of the responses is calculated through a linear static finite element analysis (low-fidelity) and corrected by the PRS that correlate the result with the one obtained by means of non-linear finite element analysis (high-fidelity).

The samples for the regression are generated through a Central Composite Design CCD, as described in Box et al. (1978), is one of the most commonly used DOEs for quadratic polynomial response surfaces. CCD consists of three distinct sets of experimental runs: (1) a factorial (perhaps fractional) design in the factors studied, each having two levels; (2) the central point, whose values of each design variable are the medians of the design space; and (3) a set of axial points, experimental runs identical to the centre points except for one design variable, which will take on values both below and above the median of the design space, and typically both outside their range (all variables are varied in this way). In this application, CCD demanded 25 finite element simulations (both linear and non-linear). Following the idea presented in Chapter IV, most of the off-line computational effort is due to these runs, apart from isolated validations. According to this procedure, the obtained PRS are shown in the following equations:

$$\begin{aligned} \hat{\delta}_1(x) = & -3.07513 + 1.57573x_1 + 0.75861x_2 + 0.97382x_3 + 0.23060x_4 \\ & -0.41144x_1^2 - 0.11233x_2^2 - 0.23423x_3^2 + \\ & -0.18688x_1x_2 - 0.32223x_1x_3 - 0.23890x_1x_4 - 0.14728x_2x_3 \end{aligned} \quad (0.2)$$

$$\begin{aligned} \hat{\delta}_2(x) = & -5.09962 + 1.68555x_1 + 1.33653x_2 + 0.96027x_3 + \\ & -0.63840x_3^2 - 0.29912x_1x_2 \end{aligned} \quad (0.3)$$

$$\hat{\delta}_3(x) = -7.77607 + 0.41315x_2 + 5.39596x_4 - 2.36998x_4^2 \quad (0.4)$$

$$\hat{\delta}_4(\mathbf{x}) = -18.5759 - 2.6548x_1 - 1.2855x_2 - 2.4576x_3 + 6.3808x_4 + \\ -1.2053x_4^2 - 1.0776x_1x_3 \quad (0.5)$$

Table 5.4 gives the comparison of the adjusted correlation coefficients, R_a^2 , of the PRS models for the responses, $\hat{f}(\mathbf{x})$, and for the correction surrogates, $\hat{\delta}(\mathbf{x})$. R_a^2 primarily measures the statistical accuracy of the PRS surrogates (the closer R_a^2 is to 1, the better). Mathematically:

$$R_a^2 = 1 - \frac{\sum_{i=1}^p (y_i - \hat{y}_i)^2}{(n_{dv} - p - 1)} \frac{(n_{dv} - 1)}{\sum_{i=1}^p (y_i - \bar{y})^2}, \quad (0.6)$$

where p is the number of data points, y_i is the observed data, $\hat{y}_i$ is the prediction of the PRS model, and $\bar{y}$ is the mean of the observed data.

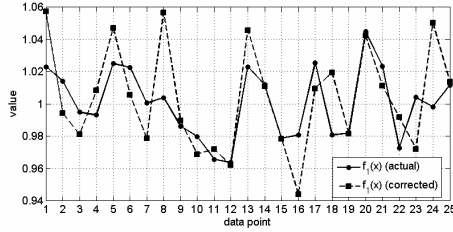
The success in using variable-fidelity over the direct meta-modeling of the responses can be observed in Table 5.4.

Table 5.4. Comparison of the adjusted correlation coefficients.

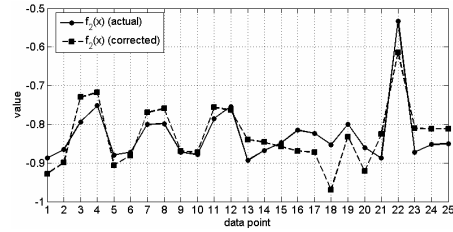
Surrogate of the response	$\hat{f}_1(\mathbf{x})$	$\hat{f}_2(\mathbf{x})$	$\hat{f}_3(\mathbf{x})$	$\hat{f}_4(\mathbf{x})$
	0.65	0.34	0.93	0.87
Correction surrogate	$\hat{\delta}_1(\mathbf{x})$	$\hat{\delta}_2(\mathbf{x})$	$\hat{\delta}_3(\mathbf{x})$	$\hat{\delta}_4(\mathbf{x})$
	0.99	0.96	0.982	0.952

As expected from the theoretical analysis, it can be observed that (i) due to the non-linearities, it pays to use the variable-fidelity framework over the PRS models of the responses; and (ii) the correction PRS models adjust very well to the data. As a consequence, the corrected response values (low-fidelity analyses and correction PRS surrogates) and the actual non-linear results (high-fidelity analyses) have very close variation in the case of all four responses of interest, as shown in Figure 5.2. As a final verification, the data corresponding to the 25 runs of the CCD was used to calculate two sets of correlation coefficients, as shown in Table 5.5. All sets of data move in the same direction (positive correlations) and, in general, the correlation is greater when the correction surrogates are used with the linear results. Given the improvement on the predicting capabilities of the low-

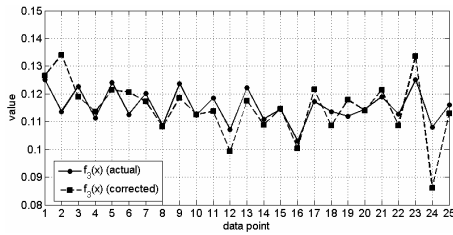
fidelity models due to the use of the correction surrogates adopted on the variable fidelity framework, it is now safe to perform the intended optimization procedures.



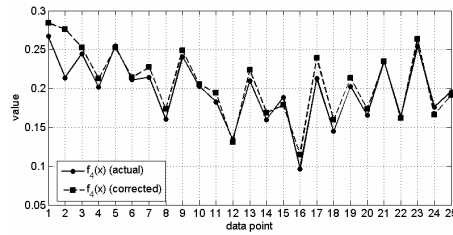
(a) Ratio of maximum tension stress at the reinforcement beams with respect to allowable one.



(b) Ratio of minimum compression stress at the reinforcement beams with respect to allowable one.



(c) Web maximum misplacement, normalized with respect to the bulkhead diameter.



(d) Maximum limit tension stress with respect to yield limit.

Figure 5.2. Variation of actual and corrected values of responses $f_1(\mathbf{x})$ to $f_4(\mathbf{x})$.

Table 5.5. Correlation coefficients of non-corrected and corrected linear results with respect to actual non-linear results.

Response	Uncorrected linear to non-linear correlation coefficients	Corrected linear to non-linear correlation coefficients
$f_1(\mathbf{x})$	51,32%	66,64%
$f_2(\mathbf{x})$	28,94%	79,28%
$f_3(\mathbf{x})$	70,09%	87,86%
$f_4(\mathbf{x})$	61,26%	94,62%

One of the aims of this work is to show the performance of the heuristic algorithms when combined with variable fidelity in a real world problem. Given the random nature of the algorithms used, results may vary from one run to another (or at least present slight differences). This way, in the first part of the study, each algorithm is tested 10 times (each time with different initial populations). The success of the algorithms in performing the initial global search is observed if the results of these repetitions are close to each other. Table 5.6 shows the setup used for each of the heuristic optimization algorithms. In all cases, each run of the algorithms will demand 1,500 low-fidelity function evaluations.

Secondly, a set of initial designs given by the heuristic methods fed the Nelder-Mead simplex direct search (NMSDS, detailed in Appendix D) in order to finish the local search.

NMSPDS was chosen because it is a zero-order method as well and thus it has no privilege of information when compared with the heuristic optimization algorithms. Finally, one of the final designs is used to validate the results of the optimization performed with variable fidelity analysis. This is done by checking the differences between the uncorrected and corrected linear (low-fidelity) simulations with respect to the non-linear one (high-fidelity).

Table 5.6. Setup for heuristic optimization methods.

General	Population size	50
	Iterations	30
	Stop criterion	Maximum. number of iterations
ACO	Dissolving rate	1.25
GA	Number of individuals for elitism	2
	Selection function	simpleGASelectionRoulette
	Crossover function	simpleGACrossoverHeuristic
	Crossover fraction	0.8
	Mutation function	simpleGAMutationUniform
	Migration direction	'forward'
	Migration interval	20
	Migration fraction	0.2
PSO	Inertia (w)	1.4
	Self trust (c_1)	1.5
	Swarm trust (c_2)	2.5
	DT	1
	Vmax	0.2
	Mass extinction factor	0.975
	Coefficient of Variation	1
	Swarm Subset	0.2

3. Results and Discussion

In the first part of the study, the performance and convergence of the heuristic algorithms is tested. Figure 5.3 and Figure 5.4 show an arbitrary run executed with each of them. They help as an initial comparison basis to analyze the behavior of the used techniques. Figure 5.3 shows how the best, the mean and the standard deviation of the objective function values for the population of ACO, GA and PSO evolve during the optimization procedure. Figure 5.4 illustrates the evolution of the LC along the iterations.

Figure 5.4-(a) shows which heuristic is conducting the optimization process at a given iteration. Figure 5.4-(b) shows the transitions due to its self-adaptation skills.

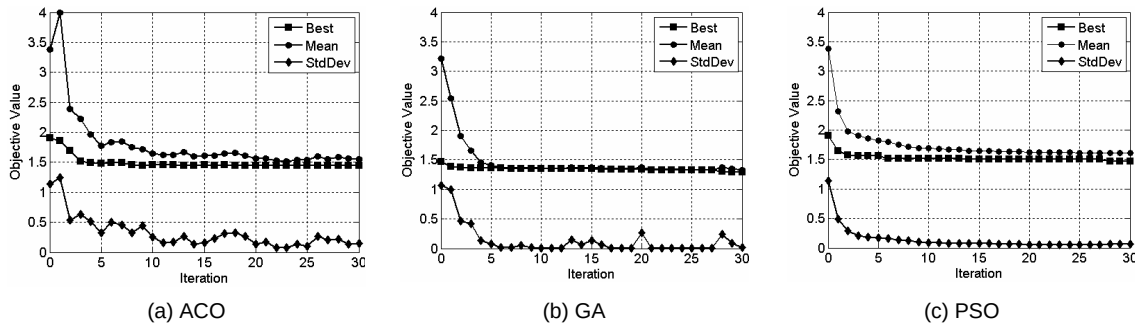


Figure 5.3. Sample of one run of the optimization for each basic algorithm.

In all cases shown above, good convergence both in terms of results (represented by the best value of the objective function) and in terms of the dispersion of the population (represented by the mean and standard deviation of the objective function) can be observed.

Figure 5.4-(a) illustrates how GA and PSO alternate in providing to LC the optimal solution. During the first 4 iterations, this is done by PSO, then the solution is few times updated by GA. Figure 5.4-(b) shows how this process is reflected in the split of the population between GA and PSO. Since most of the time GA performs better than PSO, it can be observed that the population trends to perform GA rather than PSO.

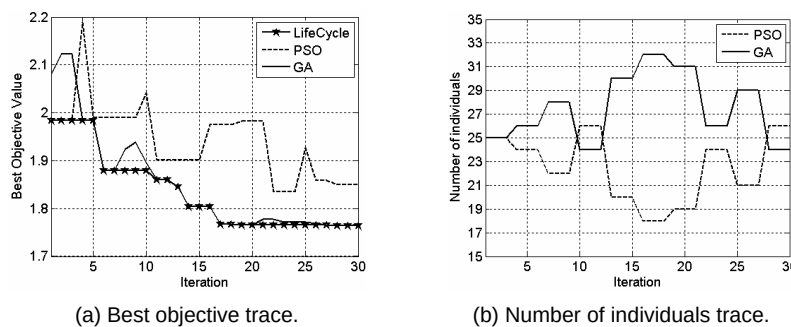


Figure 5.4. Sample of one run of the optimization for the LC algorithm.

Given the random nature of the algorithms used, the profiles of Figure 5.3 and Figure 5.4 may vary from one run to another (final results may also vary slightly). This way, for the sake of the statistical analysis, each run was repeated 10 times and the best, worst, average, standard deviation and coefficient of variation of the results from each of the 10 repetitions were recorded. The average indicates the central tendency which the heuristic results tend to. The standard deviation indicates the stability of the optimal solutions, measuring the degree of convergence. The average value for is the average of values of the functional over the 10 runs; it does not correspond to the value obtained when running the analysis with

the average design. Table 5.7 reports this information for each of the heuristic methods. The best and average results are shown in bold face. Table 5.7 shows that all algorithms converged fairly well. While GA achieved the best result, LC presents better dispersion (smaller differences among the best, average and worst designs). This is an evidence of the LC robustness, somehow expected, since LC is intended to take advantage of multiple heuristics simultaneously and to reduce the bias of a poorly performed algorithm. Altogether, Table 5.7 also indicates the success of all algorithms in pointing to a close region of the design space.

Table 5.7. Optimization results for all heuristic algorithms (performed alone).

Algorithm	Statistics	Design variable				Objective function $J(\mathbf{x})$
		$x_1 \times 10^{-3}$	$x_2 \times 10^{-3}$	$x_3 \times 10^{-3}$	$x_4 \times 10^{-3}$	
ACO	Best	25	41.94	1.5	1.08	1.44
	Average	24.96	40.19	1.5	1.03	1.73
	Worst	24.96	39.4	1.5	1.02	1.76
	StdDev	0.035	0.73	0.0001	0.02	0.10
GA	Best	29.08	42.87	1.65	0.98	1.3
	Average	26.70	38.37	1.63	1.00	1.67
	Worst	22.5	47.4	1.47	1.06	1.9
	StdDev	2.55	5.65	0.09	0.04	0.16
PSO	Best	24.89	41.35	1.49	1.07	1.47
	Average	24.61	40.72	1.49	1.04	1.76
	Worst	24.3	38.87	1.48	1.06	1.81
	StdDev	0.42	1.59	0.007	0.03	0.10
LC	Best	25.00	40.28	1.50	1.03	1.764
	Average	24.62	40.47	1.5	1.04	1.774
	Worst	24.38	41.05	1.5	1.05	1.785
	StdDev	0.4	0.64	0.002	0.01	0.009

Table 5.8 shows what happens when running the objective function with the average result found by each algorithm. The response values are all obtained by linear analysis (low fidelity) and further on properly corrected. Indeed, despite the apparent differences among the average values (which suggests small advantage for GA), the four averages are statistically equivalent at a 95% significance level, since all of them belong to the corresponding confidence interval that goes from 1.667 to 1.828.

Table 5.8. Comparison of the average design and the corresponding responses for each algorithm.

Design		Algorithm			
		ACO	GA	PSO	LC
Design variables	$x_1 \times 10^{-3}$	24.96	26.703	24.614	24.624
	$x_2 \times 10^{-3}$	40.191	38.367	40.717	40.473
	$x_3 \times 10^{-3}$	1.5	1.625	1.49	1.499
	$x_4 \times 10^{-3}$	1.03	1.003	1.038	1.036
Responses (linear corrected)	$f_1(\mathbf{x})$	0.96	0.92	0.96	0.96
	$f_2(\mathbf{x})$	-0.72	-0.66	-0.72	-0.72
	$f_3(\mathbf{x}) \times 10^{-3}$	109.42	109.44	109.23	109.28
	$f_4(\mathbf{x})$	0.18	0.18	0.18	0.18
	$f_5(\mathbf{x})$	21.56	21.82	21.60	21.6
Functional	$J(\mathbf{x})$	1.77	1.67	1.78	1.77

In the second part of the study, the efficiency of heuristics in providing an initial guess for a classical algorithm is tested. Table 5.9 shows the results when running another step of optimization using NMSDS with initial designs as given by Table 5.8. All final designs present even closer values for the design variables, individual responses and respective functional, $J(\mathbf{x})$. Indeed, the initial designs were in a close region of the design space.

Table 5.9. Results when using NMSDS with initial designs given by different heuristics.

Design		Algorithm			
		ACO + NMSDS	GA + NMSDS	PSO + NMSDS	LC + NMSDS
Design variables	$x_1 \times 10^{-3}$	29.93	29.96	29.93	29.94
	$x_2 \times 10^{-3}$	33.06	33.04	33.04	33.04
	$x_3 \times 10^{-3}$	1.73	1.73	1.73	1.73
	$x_4 \times 10^{-3}$	0.96	0.96	0.96	0.96
Responses (linear corrected)	$f_1(\mathbf{x})$	0.89	0.89	0.89	0.89
	$f_2(\mathbf{x})$	-0.67	-0.67	-0.67	-0.67
	$f_3(\mathbf{x}) \times 10^{-3}$	112.34	112.35	112.34	112.35
	$f_4(\mathbf{x})$	0.17	0.17	0.17	0.17
	$f_5(\mathbf{x})$	21.79	21.79	21.79	21.79
Functional	$J(\mathbf{x})$	1.64	1.64	1.64	1.64

As seen in Chapter IV, the final step of the variable-fidelity approach is the validation of the optimization outcomes. This is done by running the high-fidelity simulations using the optimal design. Table 5.10 shows the results of an arbitrary run of the LC algorithm as performed lonely (simulation #10) and the results of LC + NMSDS (as in Table 5.9). In general, the non-linear and linear corrected simulations present just small differences for responses $f_1(\mathbf{x})$ to $f_4(\mathbf{x})$ ($f_5(\mathbf{x})$ is the pressure bulkhead mass and it does not depend on the level of fidelity). The similar results (bold face in Table 5.10) show once again that the candidate solutions found by the heuristic methods lay in a neighborhood of the design space, where the optimal may be. From the design viewpoint, the changes introduced by the optimizers seem to be sound choices. Material has been removed from the web to the reinforcement beams, in a clear effort to minimize the mass, which was given highest priority by means of the 2.5 weighting factor in the *Compromise Programming*. Most important, the optimizers were capable of finding a design that overall improved the involved quantities, despite their inherent conflicting nature. The small increase in the displacement is secondary in view of the improvements observed in the other responses.

Table 5.10. Validation of the optimal designs given by LC and LC + NMSDS.

Design		Response				
		$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$ $\times 10^{-3}$	$f_4(\mathbf{x})$	$f_5(\mathbf{x})$
Baseline (Non-linear)		1.01	-0.85	116.0	0.2	22.56
LC (simulation #10, $J(\mathbf{x}) = 1.76$)	Linear uncorrected	3.42	-5.35	1.71	0.09	20.95
	Linear corrected	0.92	-0.66	109.44	0.17	20.95
	Non-linear	0.91	-0.65	117.5	0.17	20.95
	Optimization effect (%)	-9.90	-23.53	+1.29	-15.00	-7.14
LC + NMSDS ($J(\mathbf{x}) = 1.64$)	Linear uncorrected	0.06	-0.0709	1.85	0.0016	21.79
	Linear corrected	0.89	-0.67	112.35	0.17	21.79
	Non-linear	0.88	-0.62	118.20	0.18	21.79
	Optimization effect (%)	-12.87	-27.06	1.90	-10.00	-3.41

4. Summary and Conclusions

In this application, the use of intensive computing heuristic techniques for the optimal design of aircraft structural components was described. It was explored (i) coupling of non-linear high-fidelity and linear low-fidelity analyses through the variable fidelity approach; and (ii) the performance of different heuristic optimization methods.

The study allows the following conclusions:

- Variable fidelity approach enabled the use of intensive computing heuristic techniques and expensive non-linear analyses.
- The four applied nature-inspired methods converged in the sense of pointing out the same region of the design space as a candidate to contain the most suitable design. This means that these techniques can be used for initial exploration of the design space, with the purpose of identifying the improvement trends and ruling out the inadequate design choices.

5.1. Inverse Problems

5.1.1. Aircraft Longitudinal Stability and Control Derivatives Identification by using LifeCycle and Levenberg-Marquart Optimization Algorithms

This application explores the use of global (heuristic) and local (classical) optimization algorithms *in solo* performance and in a cascade-type approach to accomplish the parameter identification of the longitudinal motion of a real aircraft by using the Output Error Method. The first algorithm is the LC Model (composed by Genetic Algorithms and Particle Swarm Optimization). The second one is the gradient-based technique named Levenberg-Marquardt algorithm (LM, detailed in Appendix D), which is a variant of the Gauss-Newton method. Flight test data, performed with a military aircraft (Xavante AT-26 FAB 4509), were used to feed the Output Error Method. In this context, both optimization algorithms were tested, in solo performance and in a cascade-type approach. This application shortly reported Góes et al. (2007), and later detailed in Viana et al (2008c).

1. Experimental Data

The identification of longitudinal stability derivatives from flight test data were performed using a set of experimental flight data performed with a military aircraft, namely the Xavante (AT-26 FAB 4516), as illustrated in Figure 5.1. The AT-26 FAB 4509 flight test instrumentation system installed on the aircraft is composed by an inertial sensor, an anemometric boom, a GPS sensor, a surface position sensor; and a data acquisition and record unity. It provides the following measures: elevator deflection (input data in the identification procedure), angle-of-attack, pitch rate (both output data).



Figure 5.1. Xavante AT-26 FAB 4516.

The angle-of-attack (an airflow measurement) requires an air data probe installed in a region of the aircraft in which the minimum disturbance of the inflow air by the aircraft is encountered. The pitch rate (being an inertial quantity) requires an inertial measurement unit

generally installed near the center-of-gravity of the air vehicle in order to reduce the necessity of some corrections in acceleration measurements. The elevator deflection (a displacement) is generally measured by potentiometers directly installed in the aircraft control surfaces. Table 5.1 gives some information about the experimental work to evaluate some of the AT-26 FAB 4509 characteristics.

Table 5.1. AT-26 FAB 4509 data acquisition setup.

Measurement	Acquisition Rate	Acquisition Accuracy
Angle-of-attack	32 hz	0.11 %
Pitch Rate	32 hz	0.5 %
Elevator Deflection	16 hz	1.0 %

A single and manually applied longitudinal flight test maneuver was collected. It was specified in order to excite, mainly, the short period longitudinal dynamic model. Figure 5.2 shows the elevator deflection used as input in the identification procedure.

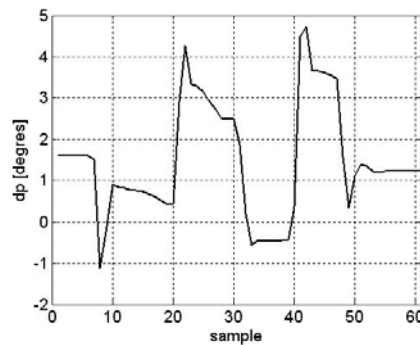


Figure 5.2. Elevator deflection.

2. Identification Problem Formulation

Figure 5.3 shows a scheme of the output error method, one of the most used estimation methods for aerodynamic parameter estimation from flight test data (Iliff, 1989) and the method used in this application. The output error method has several desirable statistical properties, including its applicability to nonlinear dynamical systems and the proper accounting of measurements noise (Iliff, 1989). In addition, it has no specific requirement about the optimization method. Even though most of works in the literature reports the use of classical methods, there are no limitations in using other approaches. In this scenario, three optimization strategies, described in Table 5.2, are explored to solve the identification problem.

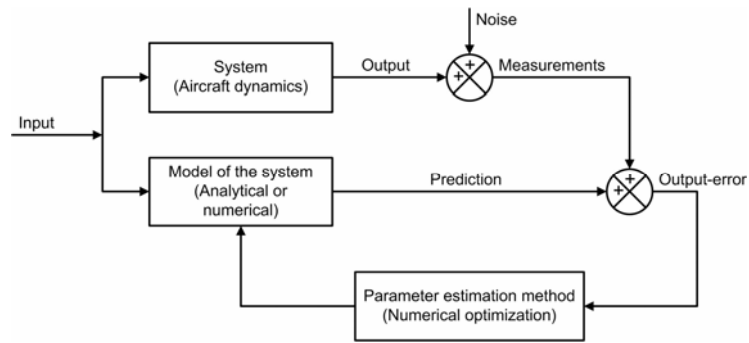


Figure 5.3. Block diagram of the estimation procedure.

Table 5.2 Optimization strategies used in the output error scheme.

Strategy	Optimization Method	Comments
#1	LC	The main explored characteristics are the global search capability and the low level of knowledge about the search space.
#2	LM	As a classical method, local search capabilities and small number of function evaluations are the main explored advantages.
#3	LC and LM	The result of a first global search performed by LC is used as an initial guess in the LM method. This cascade approach aims to make use of the main advantages of both methods.

As can be found in Góes et al. (2007) and Viana et al (2008c), the longitudinal equations of motion of the aircraft for the short period mode are described in terms of 10 variables, which constitute the search space defined in Table 5.3. Parameters Z_α , Z_q , M_α , M_q , Z_{δ_e} , and M_{δ_e} are the stability and control derivatives, which represent the aerodynamic model of the aircraft. Parameters b_{x_1} , b_{x_2} , b_{y_1} , and b_{y_2} comprise measurement biases in the output and input variables, and initial conditions on the state variables.

Table 5.3. 10-dimensional search space.

Parameter	Search space	Parameter	Search space
Z_α	$[-5 \ 0]$	M_{δ_e}	$[-10 \ 0]$
Z_q	$[0 \ 5]$	b_{x_1}	$[-0.1 \ 0.1]$
M_α	$[-5 \ 0]$	b_{x_2}	$[-0.1 \ 0.1]$
M_q	$[-5 \ 0]$	b_{y_1}	$[-0.1 \ 0.1]$
Z_{δ_e}	$[0 \ 5]$	b_{y_2}	$[-0.1 \ 0.1]$

The model is considered to be known, and the identification process consists in determining the parameter vector, $\mathbf{x}$, which gives the best prediction of the output signal $y(t)$, using optimization criteria. The attainment of an estimate through optimization of a functional based on the prediction error of the plant involves the minimization of a nonlinear function. Thus, LC, LM or a composition of both is used here to estimate the parameters of the model. Therefore, the functional involves the prediction error:

$$e(k) = \hat{y}(k) - y(k), \quad (0.1)$$

where $\hat{y}(k)$ is the output prediction based on the estimate $\hat{\mathbf{x}}$ of the parameter vector $\mathbf{x}$ at the k sample.

Consider an identifiable dynamic system represented by a model $M(\mathbf{x})$ and output y . Suppose that $p(y|\mathbf{x})$ is the conditional probability Gaussian distribution of the random variable y with dimension m , mean $f(\mathbf{x})$ and covariance R , with dimension $m \times m$. $p(y|\mathbf{x})$ is known as the likelihood functional. Goodwin and Payne (1977) attribute this name due to the fact that it is a measure of the probability of occurrence of the observation y for a given parameter $\mathbf{x}$. Thus, the likelihood functional is:

$$p(y|\mathbf{x}) = \frac{1}{(2\pi)^{m/2} |R^T|^{n/2}} \cdot \exp \left\{ -\frac{1}{2} \sum_{k=1}^n [e(k, \mathbf{x})]^T [R^T]^{-1} [e(k, \mathbf{x})] \right\}. \quad (0.2)$$

The Maximum Likelihood Estimate (MLE) is defined as the value of $\mathbf{x}$ which maximizes this functional, in such a way that the best estimate of $\mathbf{x}$ is:

$$\hat{\mathbf{x}} = \text{ArgMaxp}(y|\mathbf{x}) \quad (0.3)$$

The maximization of $p(y|\mathbf{x})$ is equivalent to the minimization of $J(\mathbf{x})$, which is given by:

$$J(\mathbf{x}) = \sum_{k=1}^n \frac{1}{2} \left\{ [e(k, \mathbf{x})]^T [R^T]^{-1} [e(k, \mathbf{x})] + \ln |R| \right\}. \quad (0.4)$$

3. Results and Discussion

Given the random nature of LC and the dependency of the initial guess of LM, the results may vary from one run to another. This way, for the sake of a reasonable statistical analysis, each one of the three strategies was repeated 20 times. For strategy #1, LC was set with different and random initial population of 50 individuals and maximum number of 200

iterations. For strategy #2, LM was set with different and initial guesses randomly picked within the design space. Finally, for strategy #3, the results obtained when running LC were used as initial guesses for LM. Besides the best and the mean results, a dispersion measure called coefficient of variation (COV) was used for comparing the performance of the different strategies. COV is computed by dividing the standard deviation by the mean (i.e., it intends to filter the order of magnitude).

Table 5.4 to Table 5.6 report a set of identification results obtained following the three different optimization strategies. It is worth mentioning that the mean value for J is the mean value of the functional over the 20 runs; it does not correspond to the value obtained when running the analysis with the mean values of the design variables.

Table 5.4 shows the best and the mean results as well as COV. Results are in a close region of the design space for strategy #1.

Table 5.4. Results for strategy #1 (LC, 50 individuals and 200 iterations).

	Best run	Mean	COV
Z_α	-1.94	-2.23	-0.40
Z_q	1.00	0.99	0.41
M_α	-2.85	-2.01	-0.65
M_q	-0.83	-1.42	-0.53
$Z_{\delta e}$	2.74	2.57	0.37
$M_{\delta e}$	-5.78	-7.20	-0.26
b_{x_1}	-0.06	-0.01	-6.07
b_{x_2}	0.1	0.1	0.08
b_{y_1}	0.1	0.09	0.13
b_{y_2}	0.04	0.06	0.31
J	7.5×10^{-9}	2.1×10^{-8}	0.41

For strategy #2, most of the runs did not find a feasible solution. Out of the 20 runs, 2 ended with an indetermination for the value of the functional J , 2 other runs ended with $J = +\infty$ and $J = -\infty$, and only 3 ended up with feasible solutions (revealing strong dependence on the initial guess when executing LM). Consequently, Table 5.5 was build computing only the remaining 16 runs. In addition to this, both mean and COV values reveal the dependency on the initial guess. Figure 5.4 helps the interpretation of these results.

Table 5.5. Results for strategy #2 (LM with initial guesses randomly picked within the design space).

	Best run	Mean	COV
Z_α	-0.70	-20.87*	-1.19
Z_q	0.23	6.45*	3.31
M_α	-2.37	-2.26	-11.73
M_q	-1.18	1.29*	4.13
$Z_{\delta e}$	0.38	2.89*	10.4
$M_{\delta e}$	-7.16	0.002*	1.8×10^4
b_{x_1}	-0.01	-1.23	-2.92
b_{x_2}	0.20	1.88*	4.94
b_{y_1}	0.08	2.75*	6.80
b_{y_2}	0.02	2.63*	2.73
J	1.2×10^{-11}	7.5×10^{149}	4

*: boundary violation.

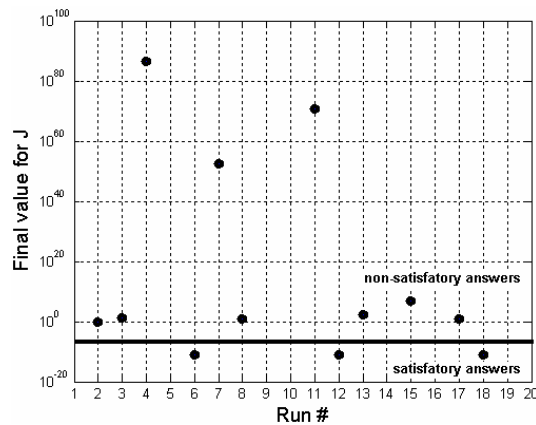


Figure 5.4. Final values of J for strategy #2 (LM).

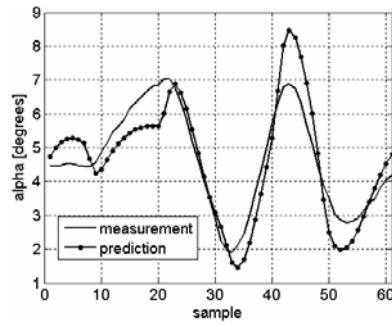
Table 5.6 shows performance when using strategy #3. It is easy to see the capability of the strategy in finding nearly same solutions. Differently from results shown in Table 5.4 and Table 5.5, here there is very small dispersion and all runs end up in feasible solutions.

Now that is clear that the winning approach is the strategy #3, the question that arises is how important the improvements driven by LM are. The answer is given when the predictions are compared. This way, Figure 5.5 and Figure 5.6 illustrate the prediction capabilities of strategy #1 (LC alone) and strategy #3 (LC+LM). Figure 5.5 helps to see

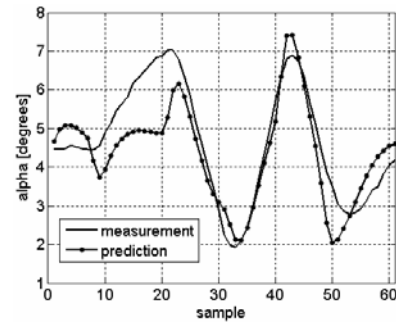
that even though relatively close, the results found by LC are not good enough as in Figure 5.6, which illustrates the success of the LC+LM strategy.

Table 5.6. Results for strategy #3 (initial guesses for LM provided by previous run of LC).

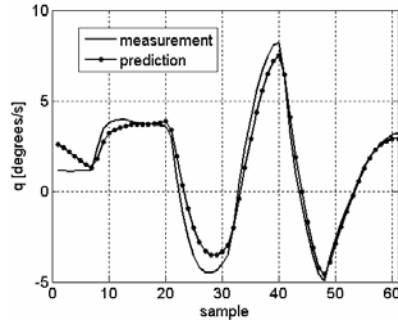
	Best run	Mean	COV
Z_α	-0.7012	-0.7012	-0.07×10^{-3}
Z_q	0.2308	0.2308	-0.10×10^{-3}
M_α	-2.3688	-2.3689	-0.2×10^{-3}
M_q	-1.1760	-1.1760	-0.2×10^{-3}
$Z_{\delta e}$	0.3841	0.3841	0.2×10^{-3}
$M_{\delta e}$	-7.1653	-7.1652	-0.1×10^{-3}
b_{x1}	-0.0093	-0.0093	-0.3×10^{-3}
b_{x2}	0.2037	0.2037	0.05×10^{-3}
b_{y1}	0.0774	0.0774	0.05×10^{-3}
b_{y2}	0.0182	0.0182	0.09×10^{-3}
J	1.2×10^{-11}	1.2×10^{-11}	2.2×10^{-5}



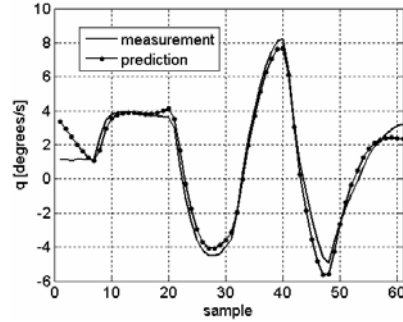
(a) Angle of attack (best case).



(b) Angle of attack (worst case).



(c) Pitch angular rate (best case).



(d) Pitch angular rate (worst case).

Figure 5.5. LC performance.

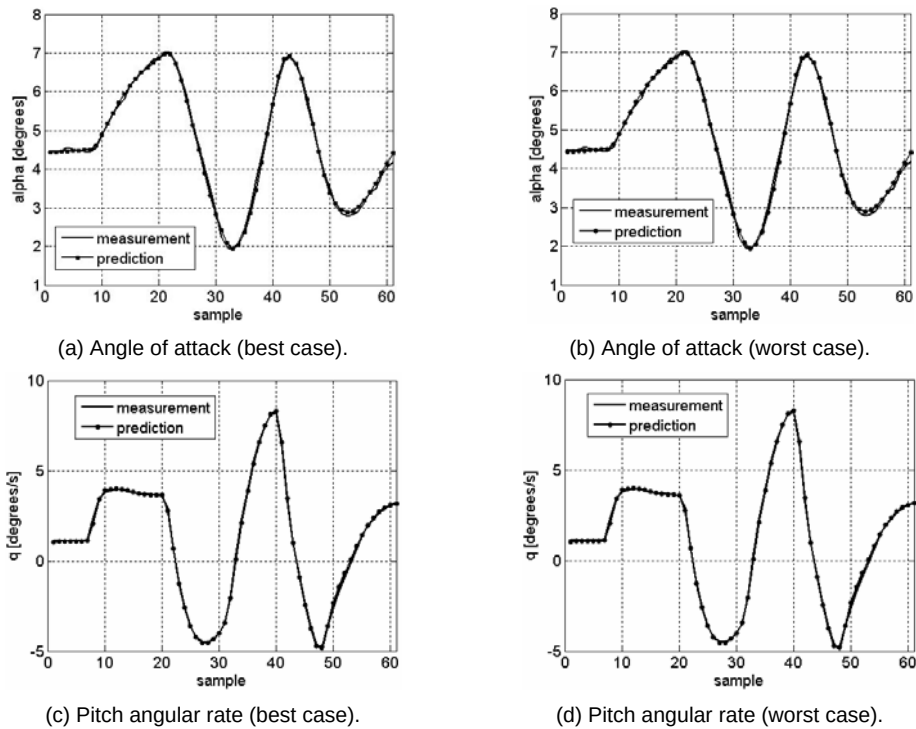


Figure 5.6. LC+LM performance.

4. Summary and Conclusions

This application presented an identification procedure to determine the longitudinal stability and control derivatives of a military aircraft. Within the framework of the Output Error method, the tested optimization algorithms are based on LC and LM methods. In the present application, GA and PSO represented the phases of the LC algorithm. The individual performances of both the algorithms were tested and then a cascade-type scheme was proposed aiming at taking advantage of the global search capabilities of LC and the local search capabilities of LM. Finally, the experimental investigation illustrated the possibility of using the proposed technique in real world environment. The results are very encouraging in the sense that more complex models, embracing non-linearities, will be analyzed in further research.

5.1.2. Identification of a Non-Linear Landing Gear Model using Heuristic Optimization

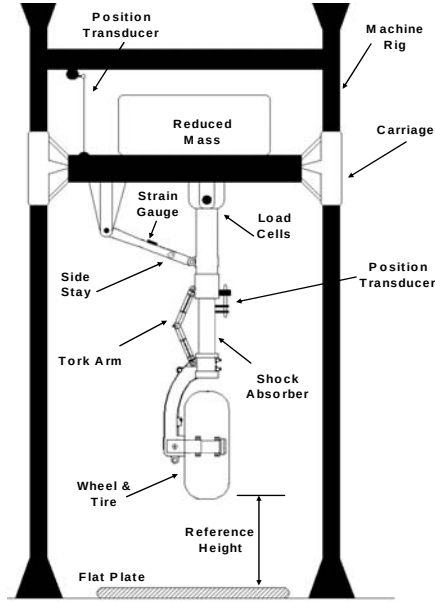
This application explores the parameter identification of a non-linear landing gear model proposed as an optimization problem solved by a heuristic optimization method, namely the LC algorithm (with GA and PSO representing the phases of the LC algorithm). The model is described in terms of the landing gear geometry, internal volumes and areas, shock absorber travel, tire type, and gas and oil characteristics of the shock absorber. The polytropic coefficient of the gas and the damping coefficient of the shock absorber are assumed as being unknown: they are considered as design variables. As an illustration, experimental drop test data, obtained under zero horizontal speed, were used in the non-linear landing gear model updating of a small aircraft. Preliminary results of this application was reported in Zanini et al. (2007), the full report is in Viana et al. (2008e).

5. Drop Test Arrangements and Non-Linear Landing Gear Model

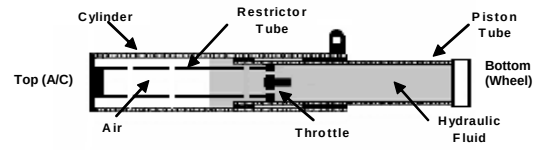
Figure 5.7-(a) illustrates a simplified drawing of the drop test installation and landing gear components. The landing gear is attached to a carriage that runs vertically between two rigs. This carriage is pre-loaded with the equivalent aircraft reduced mass. Position transducers are used to determine the vertical displacement of the carriage and deflection of the shock absorber. The difference between these parameters gives the tire deflection. Three-dimensional load cells together with strain gages are used to obtain the impact loads transferred to the aircraft. It is important to point out that, in this arrangement, no apparatus is used to simulate the aerodynamic lift.

The reduced mass used in the drop test represents the portion of the aircraft mass loaded on the landing gear during the landing impact. This mass can be calculated based on the aircraft static configuration (aircraft attitude on the ground) and certification requirements and it is further corrected to take into account the aircraft lift. A telescopic-type landing gear with an oil-pneumatic shock absorber was used. The main components of the oil-pneumatic shock absorber are shown in Figure 5.7-(b). The lower chamber contains hydraulic fluid that is forced to flow through small orifices (throttle) as a result of the piston displacement during landing. The impact energy is dissipated in this process and the air contained in the upper chamber compressed. After the maximum shock absorber deflection, part of the hydraulic fluid flows back into the lower chamber due to the expansion of the compressed air. In this extension stage, the numbers of orifices through which the fluid flows must be enough to guarantee that the tire stays in contact with the runway avoiding the rebound effect. The shock absorber considered in the present work uses eight orifices for compression and four orifices during extension. The “unsprung” mass corresponds to the sum of all movable parts

of the landing gear as tire, wheel, piston tube, fork, etc. Part of the hydraulic fluid mass is also computed in the “unsprung” mass.



(a) Drop test sketch.



(b) Oil-pneumatic shock absorber.

Figure 5.7. Drop test arrangement.

Further details about the non-linear modeling of the landing gear can be found in Viana et al. (2008e). In the scope of this text, it is enough to give general ideas involved in the formulation of the equations of movement. Figure 5.8 shows the two degrees of freedom landing gear model used in this study. In this model, M_1 and M_2 represent respectively the reduced and “unsprung” masses. K_1 and b correspond to the pneumatic spring and damping coefficients of the shock absorber. The tire is represented by a pneumatic spring K_2 . The vertical displacements of both reduced and “unsprung” masses are named as Z_1 and Z_2 , respectively. The shock absorber deflection Z is obtained by the difference between the displacements Z_1 and Z_2 .

Viana et al. (2008e) shows details on the modeling of quantities that are used in the formulation of the optimization problem, namely, the shock absorber spring force (F_{K_1}), the shock absorber damping force (F_b), tire spring force (F_{K_2}), the vertical displacement of the reduced mass (X_1), the tire deflection (X_3), the shock absorber deflection (Z), and the values of the maximum vertical forces (F_{V_1} and F_{V_2}). F_{V_1} is the vertical force that corresponds to the first half of the shock absorber deflection stage. F_{V_2} is the vertical force

that corresponds to the second half of the deflection stage. Figure 5.9 illustrates the behavior of both F_{V_1} and F_{V_2} .

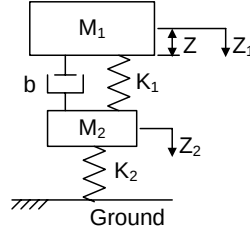


Figure 5.8. Landing gear model graphical representation.

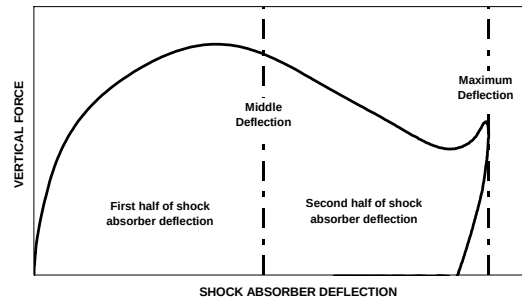


Figure 5.9. Vertical force vs. deflection curve.

The equations of motion that represent the landing gear dynamic behavior are obtained by combining the Lagrangian and Rayleigh's dissipation functions and replacing the terms that represent the shock absorber spring force (F_{K_1}), tire spring force (F_{K_2}) and damping force (F_b). Thus, after some algebraic manipulation:

$$\begin{cases} \ddot{Z}_1 = -\frac{1}{M_1} \cdot F_{K_1} - \frac{1}{M_1} \cdot F_b + g \quad , \\ \ddot{Z}_2 = \frac{1}{M_2} \cdot F_{K_1} - \frac{1}{M_2} \cdot F_{K_2} + \frac{1}{M_2} \cdot F_b + g \quad . \end{cases} \quad (0.5)$$

In order to obtain the numerical solution, it is assumed that $X_1 = Z_1$, $X_2 = \dot{Z}_1 = \dot{X}_1$, $X_3 = Z_2$, and $X_4 = \dot{Z}_2 = \dot{X}_3$.

It is also possible to write four first order differential equations, as presented below:

$$\begin{cases} \dot{X}_1 = X_2 \quad , \\ \dot{X}_2 = -\frac{1}{M_1} \cdot F_{K_1} - \frac{1}{M_1} \cdot F_b + g \quad , \\ \dot{X}_3 = X_4 \quad , \\ \dot{X}_4 = \frac{1}{M_2} \cdot F_{K_1} - \frac{1}{M_2} \cdot F_{K_2} + \frac{1}{M_2} \cdot F_b + g \quad . \end{cases} \quad (0.6)$$

Shock Absorber Spring Force (F_{K1})

The non-linear spring characteristic of the shock absorber is taken into account by considering the compression of the air and of the hydraulic fluid interconnected with a common force. Considering a fluid with uniform mass density and constant secant bulk modulus (hydraulic fluid stiffness), the force due to hydraulic fluid compression is given by:

$$F_{K1} = P_0 \cdot A \cdot \left(\frac{v_0}{v_0 - A \cdot s_1} \right)^n = \frac{\beta \cdot A^2 \cdot s_2}{V_0} \quad , \quad (0.7)$$

where:

- P_0 is the absolute air pressure for the fully extended shock absorber,
- A is the piston reference area,
- v_0 is the air volume for the fully extended shock absorber,
- s_1 is the shock absorber deflection due to the air compression,
- n is the polytrophic coefficient (constant value between 1.0 and 1.4 for adiabatic and isothermal compression, respectively),
- β is the secant bulk modulus,
- s_2 is the shock absorber deflection due to hydraulic fluid compression, and
- V_0 is the hydraulic fluid volume for the fully extended shock absorber.

The sum of the deflections obtained from air and fluid compression, (s_1) and (s_2) respectively, corresponds to the difference between the reduced and “unsprung” masses displacements. Figure 5.10-(b) illustrates the absorber force versus deflection.

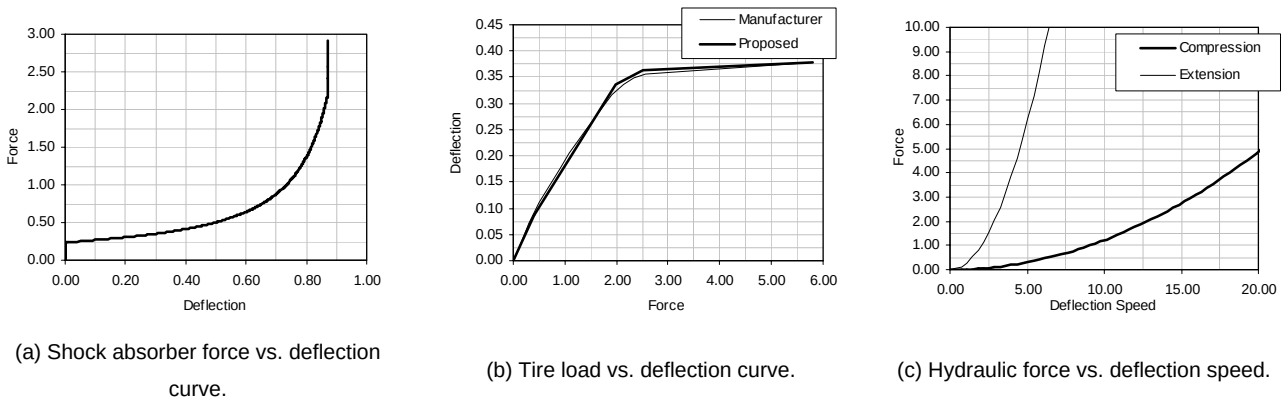


Figure 5.10. Details of the landing gear model.

Tire Spring Force (F_{K_2})

The tire was modeled as a pure spring element. The tire load versus deflection curve supplied by the manufacturer was divided into four linear regions. Therefore, the tire stiffness assumed four different values depending on the tire deflection at each simulation step. A comparison between the proposed force vs. deflection curve and that obtained from the tire manufacturer is presented in Figure 5.10-(b).

Hydraulic Resistance Force (F_b)

The hydraulic resistance in the shock absorber resulted from the difference of pressure associated with the fluid flow through the orifice. In the considered landing gear, the orifice area is enough in relation to the diameter of the shock absorber so that the jet velocities and Reynolds numbers are sufficiently large, exhibiting a fully turbulent flow. As a result, the damping force varies with the square of the telescoping velocity rather than linearly with the velocity. As a consequence, the hydraulic resistance (F_b) is given by:

$$F_b = \frac{\rho \cdot A_h^3 \cdot \dot{Z}^2}{2 \cdot (C_d \cdot A_n)^2} = \frac{\dot{Z}}{|\dot{Z}|} \cdot DS \cdot g \cdot \dot{Z}^2 \quad (0.8)$$

where ρ is the mass density of the hydraulic fluid, A_h is the hydraulic area, $\dot{Z}$ is the shock absorber deflection velocity, C_d is the coefficient of discharge, A_n is the net orifice area,

DS is a constant given by $DS = \frac{\rho \cdot A_h^3}{2 \cdot g \cdot (C_d \cdot A_n)^2}$, and g is the acceleration of gravity.

Figure 5.10-(c) is obtained by using the quadratic relation between the hydraulic force and the shock absorber deflection speed and by considering that the number of orifices reduces during shock absorber extension to avoid rebound effect.

Vertical Force Transferred to the Reduced Mass (F_v)

The vertical force transferred to the reduced mass was based on the D'Alembert's principle. Thus:

$$F_v = M_1 \cdot g - M_1 \cdot \ddot{Z}_1 \quad (0.9)$$

6. Identification Problem Formulation

The parameters that cannot be accurately determined by theoretical methods were selected for identification: the polytropic coefficient (n), and the damping parameter (DS). The corresponding identification problem consists in finding the set of design variables $\mathbf{x} = [n \ DS]^T$ that minimizes the objective function given by:

$$J(\mathbf{x}) = \frac{|X_{1trial}^{\max} - X_{1exp}^{\max}|}{X_{1exp}^{\max}} + \frac{|X_{3trial}^{\max} - X_{3exp}^{\max}|}{X_{3exp}^{\max}} + \frac{|Z_{trial}^{\max} - Z_{exp}^{\max}|}{Z_{exp}^{\max}} + \frac{|F_{V1trial}^{\max} - F_{V1exp}^{\max}|}{F_{V1exp}^{\max}} + \frac{|F_{V2trial}^{\max} - F_{V2exp}^{\max}|}{F_{V2exp}^{\max}} \quad (0.10)$$

The polytropic coefficient ranges between 1.0 and 1.4 for isothermal and adiabatic processes, respectively. The discharge coefficient was theoretically calculated by taking into account the diameters of the orifices of the shock absorber and the Reynolds' number for the upstream flow condition. The bounds of the search space for the damping parameter were arbitrarily defined as $\pm 15\%$ of the theoretical value DS_{TH} .

7. Results and Discussion

LC was used as the algorithm to solve the optimization problem. It was set up with 10 individuals, running along 50 iterations. The identified values are the following: polytropic coefficient $n = 1.05$, and damping parameter $DS = 622 \text{ kg.s}^2/\text{m}^2$. Currey (1984) recommended using a polytropic coefficient of 1.1 for shock absorbers without separator piston between gas and oil. Since this value is used only as reference and some variations can occur, for the studied shock absorber, which has oil in contact with the gas, the obtained value of 1.05 is considered as a good estimate. Similarly, the obtained damping parameter is approximately $+9\%$ above of the theoretical value, which represents also a good matching.

Figure 5.11 shows the graphics obtained for the simulations with the identified parameters and the drop test cases. Table 5.7 compares some experimental and simulated quantities. Again, the good performance and robustness of LC together with the proposed mathematical model were demonstrated. As can be seen, the agreement between experimental and simulated curves is satisfactory. The magnitude of the obtained loads variation is acceptable for a landing gear in the context of airplane design. Specifically for the tire deflection and vertical force, given by X_3 and F_v , respectively, small differences

between experimental and theoretical results are found. This is mainly due to the flexibility effect of the landing gear fork that was not considered in the proposed mathematical model.

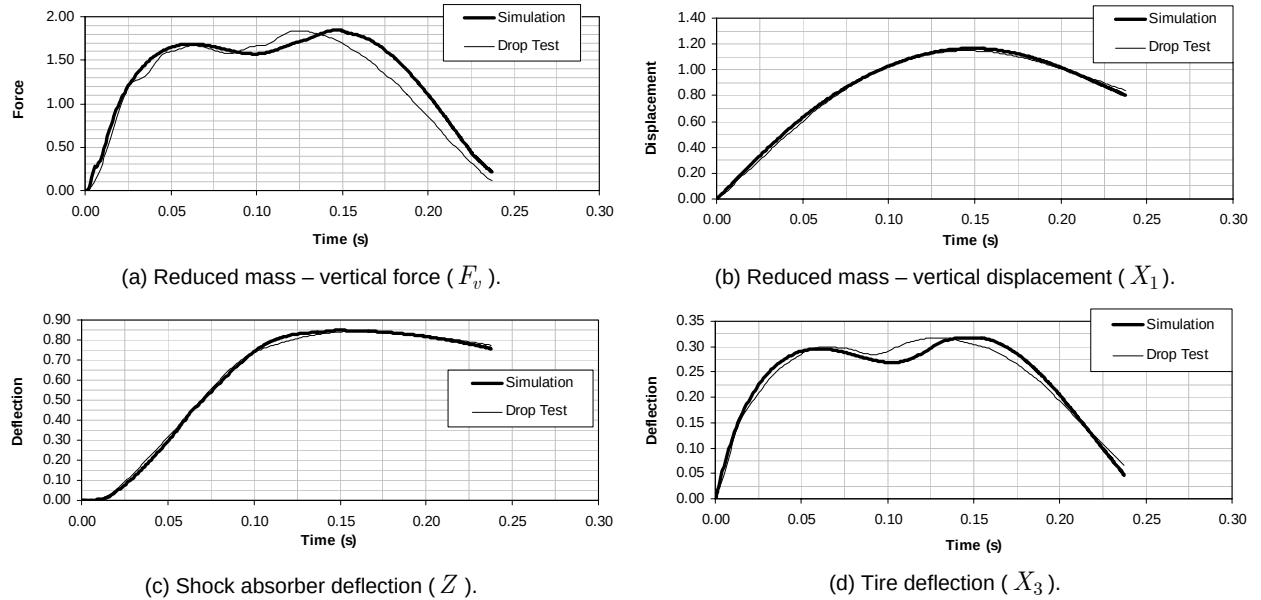


Figure 5.11. Comparison between simulation and experimental data.

Table 5.7. Experimental and simulated results for the landing gear.

Parameter	Drop Test	Simulation	Error (%)
$F_{V_1}^{\max}$ (N)	56,650.4	57,781.0	2.0
$F_{V_2}^{\max}$ (N)	65,212.3	63,278.5	−3.0
X_1 (m)	0.315	0.318	1.2
X_3 (m)	0.088	0.082	−6.1
Z (m)	0.229	0.231	1.1

8. Summary and Conclusions

This application was dedicated to the use of a heuristic optimization technique to solve the inverse problem represented by the identification of an aircraft landing gear model. For this aim, it was shown that the multi-objective function formulation makes use of both a set of measures that describe the behavior of the system and its numerical counterparts to build a representative functional, which once minimized gives a possible solution for the identification problem. The behavior of the landing gear was simulated as a non-linear model. A good match between the theoretical results and those obtained from the experimental drop-test was found. Therefore, it can be concluded that the proposed identification approach

was proved successful; encouraging future research involving more complex models, such as the ones in which wheel rotation for the representation of drag loads and landing gear flexibility are included (in addition to the two degrees of freedom considered in the present model).

5.2. Coupling Actual High-Fidelity and Surrogate Models

5.2.1. Coupling Heuristic Optimization and High-Fidelity Function Analysis to Improve Surrogate Models

This section shows how to couple heuristic optimization methods and high-fidelity analysis to improve surrogate models. This was discussed during the Section 4.4 of Chapter IV. The proposed approach is based on (i) the use of a large set of surrogates; (ii) the online update of the surrogates; and (iii) the use of *BestPRESS* for global search. In few words, the first variation updates the set of surrogates after refining any initial DOE with a pre-defined number of points. The second variation also refines any initial DOE with a pre-defined number of points; however, the update is performed every time a new point is added to the set. Here, DE was used to conduct the optimization and the performance of these approaches is studied through a set of analytical functions.

1. Numerical Experiments

A. Basic Surrogates

Table 5.8 shows the 24 different basic surrogates used during the investigation (see Chapter II for more details). The toolbox of Lophaven et al. (2002), the native neural networks Matlab toolbox (Mathworks Contributors, 2002), the code developed by Gunn (1998), and the SURROGATES toolbox of Viana and Goel (2008) were used to execute the KRG, RBNN, SVR, and PRS algorithms, respectively. SURROGATES ToolBox is also used for an easy manipulation of all these different codes. No attempt was made to improve the predictions of any surrogate by fine tuning its respective parameters. Six different Kriging surrogates are generated by varying the regression and correlation models, and 16 different SVR surrogates are created by varying the kernel function, the loss function (ε -insensitive or quadratic) and the SVR parameters (C and ε). The suggestions of Cherkassky and Ma (2004) are followed for the computation of C and ε . Additionally, for SVR, details about the kernels are given in Table 2.1. See Chapter II for more details about each surrogate technique.

B. Performance Measures

Since surrogates with different statistical assumptions are used, the root mean square error (*RMSE*, defined in Sub-section 2.2.2 of Chapter II) is chosen as measure of fit quality for the surrogates. From the set of basic surrogates, two derived surrogates are used to check the performance. They are *BestPRESS* and the most accurate surrogate of the initial

DOE, $BestRMSE$ (basis of comparison for all other surrogates). $BestRMSE$ would be selected if the performances of the surrogates were known *a priori*. In order to check the effectiveness of the strategies in adding points to the initial DOE, $BestRMSE$ is selected just once, i.e., the best surrogate of the ensemble as fitted with the initial set of data.

Table 5.8. Information about the set of 24 basic surrogates.

Surrogates		Modeling technique	Details
1	KRG-Poly0-Exp	Kriging model	Poly0, Poly1, and Poly2 indicate zero, first, and second order polynomial regression model, respectively. Exp and Gauss indicate general exponential and Gaussian correlation model, respectively. In all cases, $\theta_0 = 10 \times \mathbf{1}_{ndv \times 1}$, and $10^{-2} \leq \theta_i \leq 200$, $i = 1, 2, \dots, ndv$. 6 different Kriging surrogates are chosen by varying the regression and correlation models.
2	KRG-Poly0-Gauss		
3	KRG-Poly1-Exp		
4	KRG-Poly1-Gauss		
5	KRG-Poly2-Exp		
6	KRG-Poly2-Gauss		
7	PRS2	Polynomial response surface	Full model of degree 2.
8	RBNN	Radial basis neural network	$Goal = (0.05\bar{y})^2$ and $Spread = \frac{1}{3}$.
9	SVR-Anova-E-Full	Support vector regression	Anova, ERBF, GRBF and Spline indicate the kernel function (ERBF and GRBF kernel functions were set with $\sigma = 0.5$). E and Q indicate the loss function as ε -insensitive and quadratic, respectively. Full means that $C = \infty$ and $\varepsilon = 1 \times 10^{-4}$, while Short01 and Short02 mean that $\varepsilon = \frac{\sigma_y}{\sqrt{k}}$ and $\varepsilon = 3\sigma_y \sqrt{\frac{\ln(k)}{k}}$, in that order, while $C = 100 \max(\bar{y} + 3\sigma_y , \bar{y} - 3\sigma_y)$, where $\bar{y}$ and σ_y are the mean value and the standard deviation of the function values at the design data, respectively. 16 different SVR surrogates are chosen by varying the kernel function, the loss function (ε -insensitive or quadratic) and the SVR parameters (C and ε) define these surrogates.
10	SVR-Anova-E-Short01		
11	SVR-Anova-E-Short02		
12	SVR-Anova-Q		
13	SVR-ERBF-E-Full		
14	SVR-ERBF-E-Short01		
15	SVR-ERBF-E-Short02		
16	SVR-ERBF-Q		
17	SVR-GRBF-E-Full		
18	SVR-GRBF-E-Short01		
19	SVR-GRBF-E-Short02		
20	SVR-GRBF-Q		
21	SVR-Spline-E-Full		
22	SVR-Spline-E-Short01		
23	SVR-Spline-E-Short02		
24	SVR-Spline-Q		

Each basic surrogate and $BestPRESS$ are compared with the best surrogate of the initial DOE ($BestRMSE$). $\%difference$ is defined such as the percent difference by choosing a specific model over $BestRMSE$:

$$\%difference = 100 \frac{RMSE_{BestRMSE} - RMSE_{Surr}}{RMSE_{BestRMSE}}. \quad (0.11)$$

At the initial DOE, for *BestPRESS*, it is expected that $\%difference \leq 0$, which means that there may be losses and the best case scenario is when one of the basic surrogates (hopefully *BestPRESS*) coincides with *BestRMSE*. With the final set of points, $\%difference$ for *BestPRESS* may turn to positive, which express improvement over *BestRMSE*.

C. Analytical Examples

To test the effectiveness of the various approaches, a set of analytical functions widely used as benchmark problems in optimization is employed (Dixon and Szegö, 1978). These are:

1. Branin-Hoo function (2 variables):

$$y(\mathbf{x}) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10, \quad (0.12)$$

$$-5 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 15.$$

2. Camelback function (2 variables):

$$y(\mathbf{x}) = \left(\frac{x_1^4}{3} - 2.1x_1^2 + 4 \right) x_1^2 + x_1 x_2 + (4x_2^2 - 4) x_2^2, \quad (0.13)$$

$$-3 \leq x_1 \leq 3, \quad -2 \leq x_2 \leq 2.$$

3. Hartman functions (3 and 6 variables):

$$y(\mathbf{x}) = - \sum_{i=1}^q a_i \exp \left(- \sum_{j=1}^m b_{ij} (x_j - d_{ij})^2 \right), \quad (0.14)$$

$$0 \leq x_j \leq 1, \quad j = 1, 2, \dots, m.$$

Hartman3, with 3 variables and Hartman6 with 6 variables were used. For both, $q = 4$, $\mathbf{a} = [1.0 \quad 1.2 \quad 3.0 \quad 3.2]$, and other parameters are given in Table 5.9.

Table 5.10 shows details about the data set generated for each test function. Naturally, the number of points used to fit surrogates increase with dimensionality. Since quality of fit may vary with the initial DOE, results based on 100 different instances of latin hypercube designs (created with the MATLAB latin hypercube function `lhsdesign`, set with the "maxmin" option with 100 iterations) for all the problems are presented. For all examples, 5 different latin hypercube designs are used to test the surrogates (also created with

lhsdesign, but set with the “maxmin” option and only 10 iterations). The $RMSE$ is taken as the mean of the values for the 5 DOEs. The number of extra points presented in Table 5.10 refers to those points allocated by the heuristic optimization. The number of points for update of surrogates refers to when the approach suggested on Figure 4.4 (Chapter IV) performs the step illustrated on Figure 4.4-(b). k -folds refers to the value of k used for the k -fold strategy of computation of the $PRESS$ errors. This way, for the Hartman6 function, when the number of points is equal to 60, $k = 20$, when it is 63, $k = 21$, and so on.

Table 5.9. Parameters used in Hartman function.

Hartman3	$\mathbf{B} = \begin{bmatrix} 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \\ 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \end{bmatrix}$ $\mathbf{D} = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}$
Hartman6	$\mathbf{B} = \begin{bmatrix} 10.0 & 3.0 & 17.0 & 3.5 & 1.7 & 8.0 \\ 0.05 & 10.0 & 17.0 & 0.1 & 8.0 & 14.0 \\ 3.0 & 3.5 & 1.7 & 10.0 & 17.0 & 8.0 \\ 17.0 & 8.0 & 0.05 & 10.0 & 0.1 & 14.0 \end{bmatrix}$ $\mathbf{D} = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$

Table 5.10. Specifications for the Latin hypercube DOEs.

Test problem	# design variables	# points for fitting	# extra points	# points for update of surrogates	k -folds	# points for test (in each of the 5 DOEs)
Branin-Hoo	2	12	6	13, 14, ..., 18	13, 14, ..., 18	2000
Camelback	2	12	6	13, 14, ..., 18	13, 14, ..., 18	2000
Hartman3	3	20	10	21, 22, ..., 30	21, 22, ..., 30	2000
Hartman6	6	56	14	60, 63, 66, 69, 70	20, 21, 22, 23, 14	2000

In addition, the $RMSE$ on the neighborhood of the optimum of each function is also studied. The setup for this part is given in Table 5.11. The DOEs were generated with the MATLAB lhsnorm function (latin hypercube sample with a normal distribution) set with $\mu = \mathbf{x}^*$, and $\sigma = 0.001 \times \text{diag}(\mathbf{1}_{n_{dv} \times 1})$.

Table 5.11. Point of optimum and number of points used to estimate the $RMSE$ at the region close to the optimum.

Test problem	$\mathbf{x}^*$	# points for test
Branin-Hoo	$\begin{bmatrix} 9.4248 & 2.4750 \end{bmatrix}^T$	1000
	$\begin{bmatrix} 3.1416 & 2.2750 \end{bmatrix}^T$	1000
	$\begin{bmatrix} -3.1416 & 12.2750 \end{bmatrix}^T$	1000
Camelback	$\begin{bmatrix} 0.0898 & -0.7127 \end{bmatrix}^T$	1000
	$\begin{bmatrix} -0.0898 & 0.7127 \end{bmatrix}^T$	1000
Hartman3	$\begin{bmatrix} 0.1 & 0.5559 & 0.8522 \end{bmatrix}^T$	1000
Hartman6	$\begin{bmatrix} 0.2017 & 0.15 & 0.4769 & 0.2753 & 0.3117 & 0.6573 \end{bmatrix}^T$	1000

Figure 5.12 illustrates the complexity of the problems. For the two-dimensional cases, plots using the test points reveal the presence of high gradients. For all other cases, the test points were used to obtain boxplots of the functions, which show variation in the values of the functions by more than one order of magnitude. See Appendix C for details about boxplots.

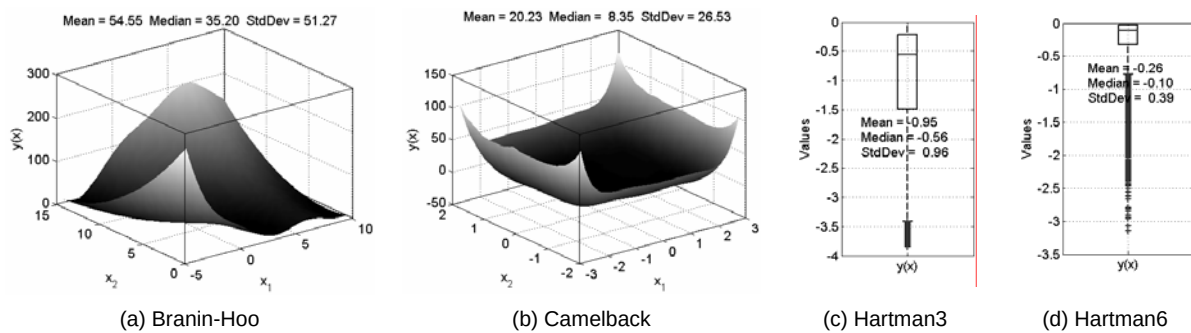


Figure 5.12. Plot of test functions.

Table 5.12 shows the setup used for DE, the method used inside the algorithm proposed in Chapter IV, Figure 4.3, for all test functions.

2. Results and Discussion

The numerical experiments are intended to:

- measure the improvement due to the addition of points in the data set in the (i) $RMSE$; (ii) region close to the optimum;

- check the effectiveness of the two different strategies proposed in Figure 4.3 in adding points to the initial data set; and
- explore how to take advantage of a large set of surrogates.

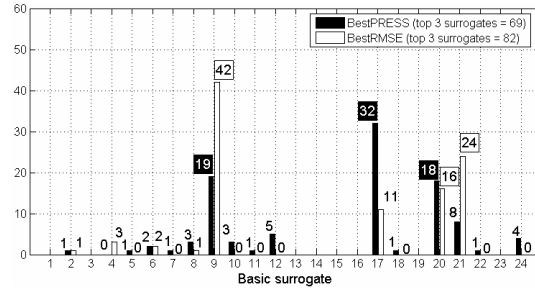
Table 5.12. Setup for DE algorithm.

DE scheme	Amplification Factor	Crossover Probability	Max. number of Iterations	Population size
'rand-to-best/1/exp'	0.8	0.5	20	Same number of points used to fit surrogates (see Table 5.10).

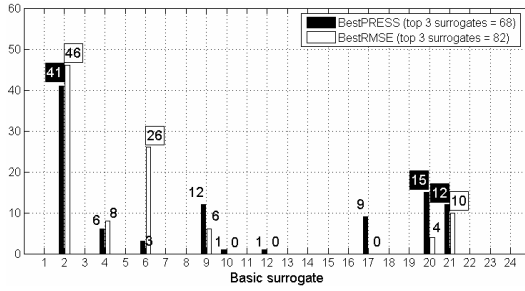
The first bullet is first quantified by comparing the correlation between $PRESS_{RMS}$ and $RMSE$ across the 24 surrogates (i.e., for a given DOE, the correlation is computed between the vectors of $PRESS_{RMS}$ and $RMSE$ values for the different surrogates) and the effect on the performance measures. The second bullet is quantified by the $\%difference$ for each of the 24 surrogates and $BestPRESS$ as compared with the best surrogate ($BestRMSE$) obtained with the initial set of data.

Figure 5.13 shows the frequency of best $PRESS_{RMS}$ and the best $RMSE$ for each of the surrogates for the Branin-Hoo function. It can be observed that for both $PRESS_{RMS}$ and $RMSE$, the best surrogate depends (i) on the problem, i.e. no single surrogate or even modeling technique is always the best; (ii) on the DOE, i.e. for the same problem, the surrogate that performs the best can vary from DOE to DOE; (iii) adding points to the initial DOE may change the identity of the best surrogates; and (iv) the best 3 surrogates according to both $PRESS_{RMS}$ and $RMSE$ tend to be the same.

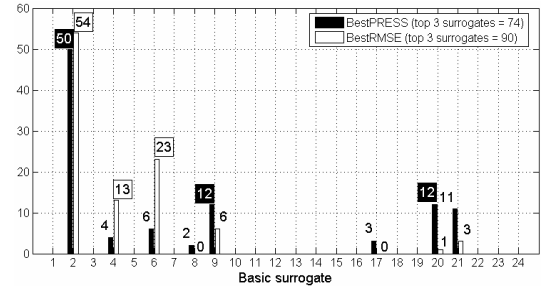
Figure 5.14 shows a histogram of the correlations between $PRESS_{RMS}$ and $RMSE$ for different surrogates, where the histograms represents the 100 DOEs. The Branin-Hoo and Hartman6 functions are chosen for illustrating what happens in low and high dimension, respectively. In a given DOE, the correlation is computed between the sets of $PRESS_{RMS}$ and $RMSE$ values. While none of the strategies to add points seems to be significantly better than the other, the correlation appears to improve with the number of points.



(a) Branin-Hoo, 12 points.

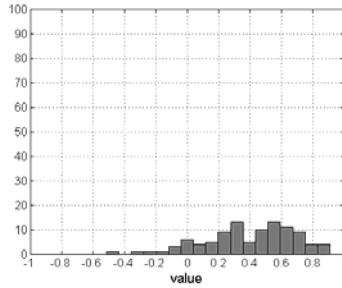


(b) Branin-Hoo, 18 points. Actual evaluation phase, without update of surrogates.

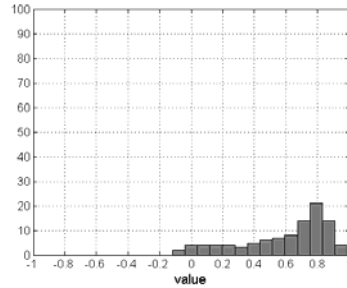


(c) Branin-Hoo, 18 points. Actual evaluation phase, with update of surrogates.

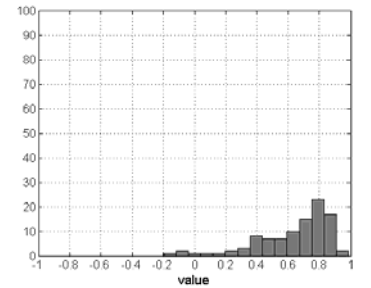
Figure 5.13. Frequency, in number of DOEs (out of 100), of best $PRESS_{RMS}$ and $RMSE$ for each basic surrogate for the Branin-Hoo function (numbers on the ordinate indicate the surrogate as in Table 5.8).



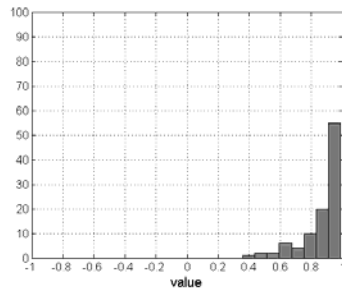
(a) Branin-Hoo, 12 points.



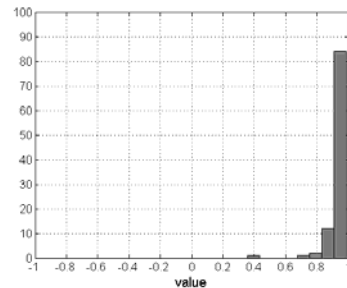
(b) Branin-Hoo, 18 points. Actual evaluation phase, without update of surrogates.



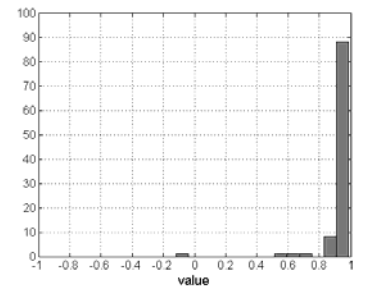
(c) Branin-Hoo, 18 points. Actual evaluation phase, with update of surrogates.



(d) Hartman6, 56 points.



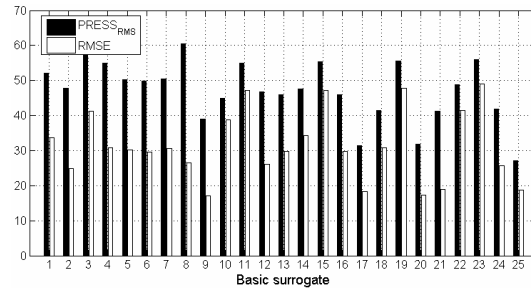
(e) Hartman6, 70 points. Actual evaluation phase, without update of surrogates.



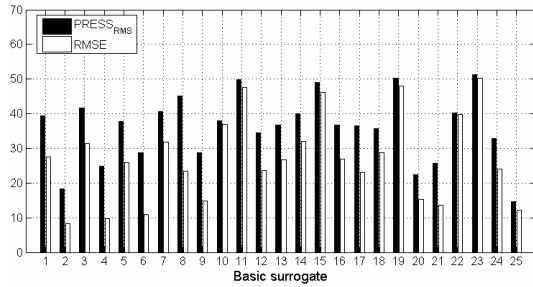
(f) Hartman6, 70 points. Actual evaluation phase, with update of surrogates.

Figure 5.14. Frequency of the correlation between $PRESS_{RMS}$ and $RMSE$ (out of 100 experiments) for the Branin-Hoo (low-dimension) and Hartman6 (high-dimension).

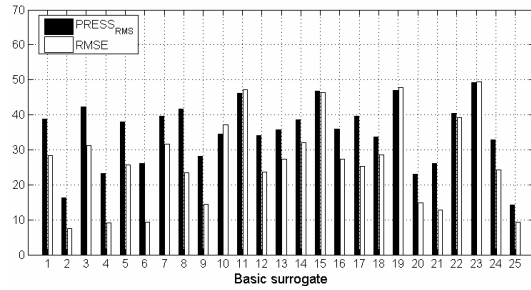
Figure 5.15 complements Figure 5.14, showing the median of $PRESS_{RMS}$ and $RMSE$ over the 100 DOEs.



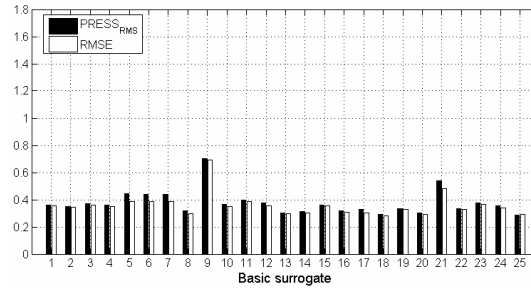
(a) Branin-Hoo, 12 points.



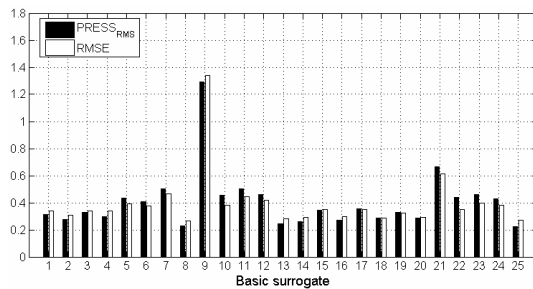
(b) Branin-Hoo, 20 points. Actual evaluation phase, without update of surrogates.



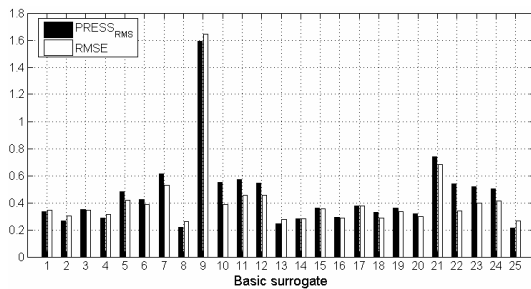
(c) Branin-Hoo, 20 points. Actual evaluation phase, with update of surrogates.



(d) Hartman6, 56 points.



(e) Hartman6, 70 points. Actual evaluation phase, without update of surrogates.



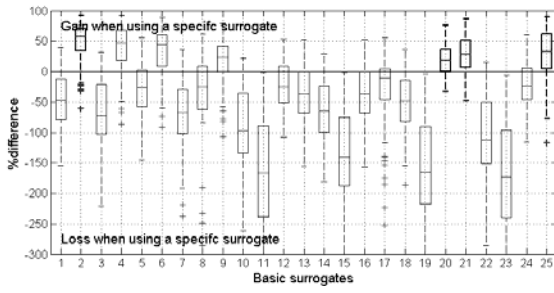
(f) Hartman6, 70 points. Actual evaluation phase, with update of surrogates.

Figure 5.15. Median over 100 DOEs of $PRESS_{RMS}$ and $RMSE$ (lower values indicate better fits) for basic surrogates for the analytical examples (numbers on the ordinate indicate the surrogate as in Table 5.8; the 25th surrogate is $BestPRESS$).

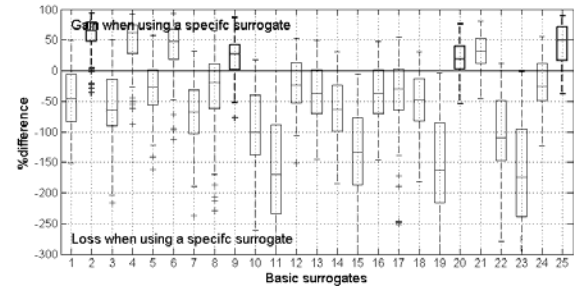
Once again, while there is no significant difference on the strategies, the agreement between both criteria improves with increasing number of points. Figure 5.15 and Figure 5.14 show that not only the correlation is good but also the actual values of $PRESS_{RMS}$ and $RMSE$ agreed well, especially for an increasing number of points. Altogether, $PRESS_{RMS}$ is good just for filtering out bad surrogates when there are few points (low-dimensional problems, i.e., 2 and 3 variables), and it can identify the sub-set of the best surrogates when there are more points available. Up to this point, while the benefits of the DOE refinement are already clear, there is no significant difference in performance when the two different strategies illustrated in Figure 4.4 are compared. In addition, since the identity of the best surrogate is DOE and problem dependent, it is also clear the advantage of using a large set of surrogates and then $BestPRESS$ as a way to prevent against bad fitted surrogates.

Next, the improvement over the $BestRMSE$ by adding points is studied. Figure 5.16 and Figure 5.17 illustrates the $\%difference$ for all basic surrogates and $BestPRESS$ for the examples of the Branin-Hoo and Hartman6 functions. For Figure 5.16, test data were generate as specified in Table 5.10. For Figure 5.17, test data were generate as specified in Table 5.11. Contrarily to what can be expected, adding points to the initial DOE does not always improve the performance of the basic surrogates. On the other hand, $BestPRESS$ shows to be not only more robust but also capable to gain with the extra simulations performed. It is also clear that, especially in high dimensions, the strategies for adding points benefit the region of the optima more intensively than the whole design space. Even though this is a trivial observation, it is worth to point out the potential of the $BestPRESS$ for optimization.

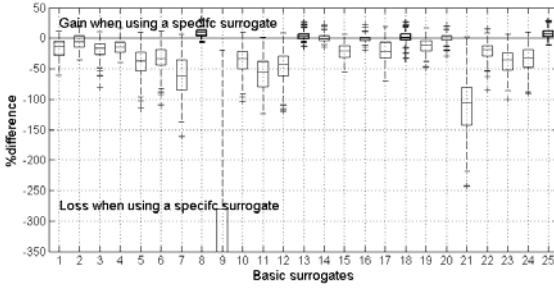
Table 5.13 provides the mean, median and standard deviation of the $\%difference$ in the $RMSE$ for the best 3 surrogates and $BestPRESS$. Test data were generate as specified in Table 5.10. The positive or negative signs of $\%difference$ indicate gain or loss in terms of $RMSE$ for the specific surrogate compared with the best surrogate of the original DOE. In general, $BestPRESS$ is the second best surrogate. It can be observed that (i) over the design space, the gains are limited by both the number of design variables (see the contrast between Branin-Hoo and Hartman6) and the complexity (see the contrast between Branin-Hoo and Camelback); and (ii) at the region of the optima, $BestPRESS$ always presented gains higher than 30%.



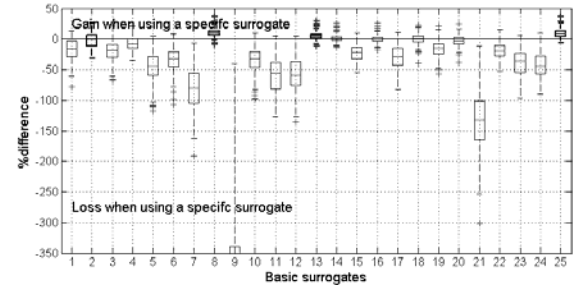
(a) Branin-Hoo, 20 points. Actual evaluation phase, without update of surrogates.



(b) Branin-Hoo, 20 points. Actual evaluation phase, with update of surrogates.

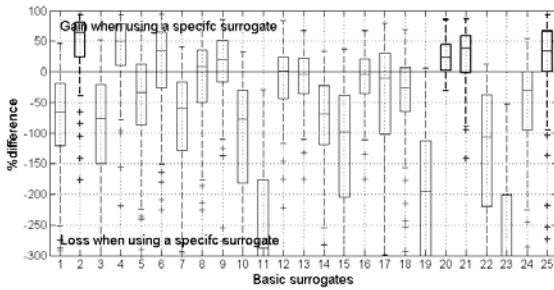


(c) Hartman6, 70 points. Actual evaluation phase, without update of surrogates.

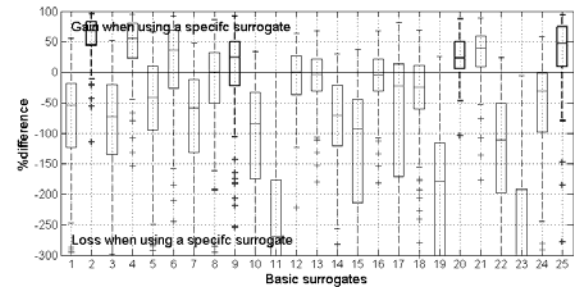


(d) Hartman6, 70 points. Actual evaluation phase, with update of surrogates.

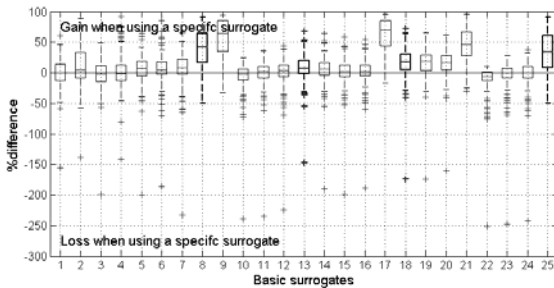
Figure 5.16. $\%difference$ in the $RMSE$, defined in Eq. (0.11), over the design space (numbers on the ordinate indicate the surrogate as in Table 5.8; the 25th surrogate is *BestPRESS*. See Appendix C for details about boxplots).



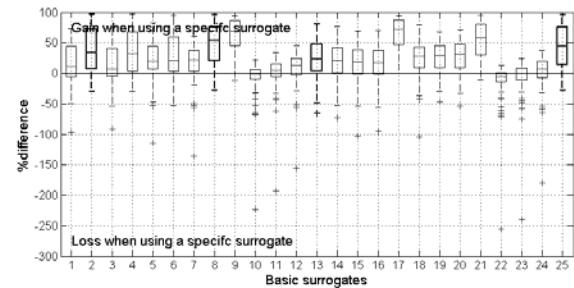
(a) Branin-Hoo, 20 points. Actual evaluation phase, without update of surrogates.



(b) Branin-Hoo, 20 points. Actual evaluation phase, with update of surrogates.



(c) Hartman6, 70 points. Actual evaluation phase, without update of surrogates.



(d) Hartman6, 70 points. Actual evaluation phase, with update of surrogates.

Figure 5.17. $\%difference$ in the $RMSE$, defined in Eq. (0.11), in the region of the optima (numbers on the ordinate indicate the surrogate as in Table 5.8; the 25th surrogate is *BestPRESS*. See Appendix C for details about boxplots).

Table 5.13. %difference in the $RMSE$, defined in Eq. (0.11), of the best 3 basic surrogates (according to how often they have the best $PRESS_{RMS}$, see Figure 5.13) and $BestPRESS$ (for the basic surrogates, the numbers indicate the identity as in Table 5.8)

Problem	Surrogate	Freq. of best $PRESS_{RMS}$	Over the design space			In the region of the optima		
			Median	Mean	StdDev	Median	Mean	StdDev
Branin-Hoo (without update of surrogates)	2	41	57	50	31	64	47	51
	20	15	19	20	25	23	25	26
	21	12	28	28	29	39	21	78
	BestPRESS	---	33	31	39	34	23	67
Branin-Hoo (with update of surrogates)	2	50	66	60	26	68	57	37
	9	12	27	21	32	24	-8	117
	20	12	19	21	26	23	26	31
	BestPRESS	---	49	43	33	47	35	57
Camelback (without update of surrogates)	1	11	13	11	22	14	-26	115
	5	19	31	29	19	40	24	62
	6	16	16	20	15	64	39	111
	BestPRESS	---	6	8	22	48	36	49
Camelback (with update of surrogates)	1	13	13	10	22	10	-35	123
	5	20	32	30	18	48	29	61
	17	15	-8	-18	41	73	62	32
	BestPRESS	---	6	9	21	46	31	68
Hartman3 (without update of surrogates)	2	49	29	27	24	40	31	56
	4	14	18	14	25	38	25	77
	13	14	13	14	16	-4	-30	87
	BestPRESS	---	21	21	25	33	25	67
Hartman3 (with update of surrogates)	2	46	32	26	29	63	48	46
	8	7	5	-14	88	31	21	63
	13	37	16	17	16	6	-8	79
	BestPRESS	---	24	24	19	34	31	55
Hartman6 (without update of surrogates)	8	67	10	9	7	42	40	30
	13	9	3	3	6	8	8	25
	18	18	2	1	9	18	16	28
	BestPRESS	---	8	7	7	34	35	31
Hartman6 (with update of surrogates)	2	6	-1	-3	13	34	37	35
	8	73	10	10	7	54	48	33
	13	17	4	5	6	23	26	29
	BestPRESS	---	8	9	7	45	44	34

3. Summary and Conclusions

In this application, the use of multiple surrogates, heuristic optimization methods, and extra high-fidelity simulations for minimum $RMSE$ in meta-modeling were explored. This was done by (i) the generation of a large set of surrogates and the use of $PRESS_{RMS}$ as a

criterion for surrogate selection; and (ii) the use of heuristic optimization for the generation of extra points for high-fidelity simulation. In addition, two different strategies for updating the surrogates were studied. The first one does not include the update of the surrogates after the inclusion of each point in the data set. Contrarily, the second one updates the set of surrogates every time a new point is included in the data set.

Based on a set of standard test functions, the study leads to the conclusion that the benefits of both strategies depend on dimensionality and number of points:

- *BestPRESS* is efficient avoiding poorly fitted surrogates and then indicating new points for extra simulation.
- While over the design space, the gains are limited by the number of design variables and the complexity of the function, both approaches presented above are robust in presenting gains in the region where the optima are located.

Additionally, there is no significant difference between the strategies used for adding points. This will be a subject for further research on this topic.

5.3. Combinatorial Optimization

5.3.1. On How to Implement an Affordable Optimal Latin Hypercube

It is worth mentioning that the reported methodology for obtaining Optimal Latin Hypercube was implemented during an exchange program at Vanderplaats Research and Development Inc., from May/2006 to July 2006. The resulting code is now part of the commercial package VisualDOC, one of the company's optimization software. This research was also properly reported in Viana et al. (2007a) and Viana et al. (2007c).

1. The Optimal Latin Hypercube

Chapter II has already discussed the importance of the Latin Hypercube design for computer generated experiments. Unfortunately, generating an Optimal Latin Hypercube design results in a difficult optimization problem that is traditionally solved by time consuming non-gradient based methods, for example a genetic algorithm. Solutions reported in the literature often exceed several hours for large number of points and large number of design variables (Morris and Mitchell, 1995; Ye et al., 2000; and Bates et al., 2004). Such a high computational cost limits the practical use of this important design of experiments.

The Optimal Latin Hypercube is obtained through the optimization problem of searching for a design $\mathbf{X}^*$ (a design matrix with n points and n_{dv} variables), which minimizes the objective function $f(\mathbf{X})$:

$$\mathbf{X}^* = \min f(\mathbf{X}) . \quad (0.15)$$

The objective function is formulated to achieve the required uniform space-filling property and, as a result, to avoid situations such as that illustrated in Figure 2.3-(b). In this application, the ϕ_p criterion (Morris and Mitchell, 1995; Jin et al., 2005) was used for the objective function; which leads to the maximization of the point-to-point distance in the design (Johnson et al., 1990). A design is called a ϕ_p -optimal design, if it minimizes:

$$\phi_p = \left[\sum_{i=1}^s \Theta_i d_i^{-p} \right]^{1/p} , \quad (0.16)$$

where:

- p is a positive integer, with a very large p , the ϕ_p -criterion is equivalent to the *maximin* distance criterion (Jin et al., 2005);
- d_i 's are distinct distance values with $d_1 < d_2 < \dots < d_s$;
- J_i is the number of pairs of points in the design separated by d_i ; and
- s is the number of distinct distance values.

By sorting all the point-to-point distance d_{ij} ($i \geq 1, j \leq n, i \neq j$), the distance list ($d_1, d_2, \dots, d_s$) and the index list ($\Theta_1, \Theta_2, \dots, \Theta_s$) can be obtained. To close the ϕ_p definition, the inter-sited distance can be expressed as follows:

$$d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{k=1}^{n_{dv}} |x_{ik} - x_{jk}|^2 \right]^{1/2}. \quad (0.17)$$

Up to this point, only the optimization problem was defined. However, in order to efficiently obtain an Optimal Latin Hypercube design, more than the problem definition is required. An appropriate optimization algorithm, with an affordable computational cost for evaluating the objective function, is also important. To overcome the high computational cost associated with the existing approaches the method used in this work is supported by (a) an adaptation and an enhancement of a global search algorithm, i.e., the Enhanced Stochastic Evolutionary Algorithm (i.e., ESEA, already presented in Chapter III), and (b) an efficient method for evaluating the objective function to reduce the computational cost.

2. Efficient Approach for Evaluating the Optimality Criterion

Since the objective function is evaluated whenever a new design of experiments is constructed, the efficiency of this evaluation becomes very important for creating the Optimal Latin Hypercube design within a reasonable time frame. Consider the evaluation of the ϕ_p based on Eq. (0.16). It can be seen that this process includes three parts, i.e.:

1. evaluation of all the point-to-point distances;
2. sorting of these distances to obtain a distance list and index list; and
3. evaluation of the ϕ_p value.

However, it can be observed that after an exchange ($x_{i_1k} \leftrightarrow x_{i_2k}$) only elements in rows i_1 and i_2 and columns i_1 and i_2 are changed in the distance matrix, $\mathbf{D}$. Thus, if Eq.

(0.16) could be written to take advantage of this fact, an efficient way for the calculation of ϕ_p is provided. It would avoid unnecessary calculations and the sorting required by Eq. (0.16). In this case, the new ϕ_p is computed by:

$$\phi'_p = \left[\phi_p^p + \sum_{1 \leq j \leq n, j \neq i_1, i_2} \left((d'_{i_1 j})^{-p} - (d_{i_1 j})^{-p} \right) + \sum_{1 \leq j \leq n, j \neq i_1, i_2} \left((d'_{i_2 j})^{-p} - (d_{i_2 j})^{-p} \right) \right]^{1/p}, \quad (0.18)$$

where:

- $d'_{i_1 j} = d'_{j i_1} = \left[(d_{i_1 j})^2 + s(i_1, i_2, k, j) \right]^{1/2}$;
- $d'_{i_2 j} = d'_{j i_2} = \left[(d_{i_2 j})^2 + s(i_1, i_2, k, j) \right]^{1/2}$; and
- $s(i_1, i_2, k, j) = |x_{i_2 k} - x_{j k}|^2 - |x_{i_1 k} - x_{j k}|^2$.

Table 5.14 provides an indication of possible savings in computational time when using Eq. (0.18) to evaluate the objective function. T_{full} is the objective function calculated through Eq. (0.16) and $T_{enhanced}$ is the enhanced approach, using Eq. (0.18). Table 5.14 shows the wall-clock time.

Table 5.14. Time comparison between two ways of calculating the objective function, adapted from Jin et al. (2005).

Latin Hypercube	12 x 4	25 x 4	50 x 5	100 x 10
$T_{enhanced}/T_{full}$	0.45	0.19	0.08	0.03

3. Structured Latin Hypercube Design

This section discusses the methodology proposed during this doctoral research for obtaining Optimal Latin Hypercube designs without using formal optimization. Using this methodology, the Optimal Latin Hypercube designs are obtained with minimal computational effort and in real time. The methodology exploits patterns of point locations for Optimal Latin Hypercube designs. Small building blocks with one or several points in each are used to recreate these patterns by taking into account the dimensionality of the problem and the required number of design points.

In its current form, the obtained Structured Latin Hypercube designs provided by this new methodology in two dimensional spaces cannot be improved using non-gradient optimization methods. In higher dimensional design spaces, the Structured Latin Hypercube designs generated using the proposed methodology are still good, but can be improved using non-gradient optimization methods. Even though the designs can be improved for higher dimensional problems, the proposed methodology still provides a powerful and practical tool for obtaining good Optimal Latin Hypercube designs in real time (at most, seconds).

The approach is quite simple and it is based on the hope that the simple n_{dv} -dimensional Latin Hypercube that can be constructed from a n_{dv} -dimensional seed design. Instead of a formal description of the approach, a practical example will be used to explain the methodology of creating a Structured Latin Hypercube design. Consider the case in which is desired a 16×2 Latin Hypercube, i.e., 16 points in 2 dimensions. First, a small Latin Hypercube design will be constructed to be used as a seed in the process. Figure 5.18 shows the some examples of 2-dimensional seed designs. Figure 5.18-(a) shows the seed used in this example. This seed can be as simple as just a $1 \times n_{dv}$ design (where n_{dv} is the number of dimensions of the problem, i.e. the number of design variables).

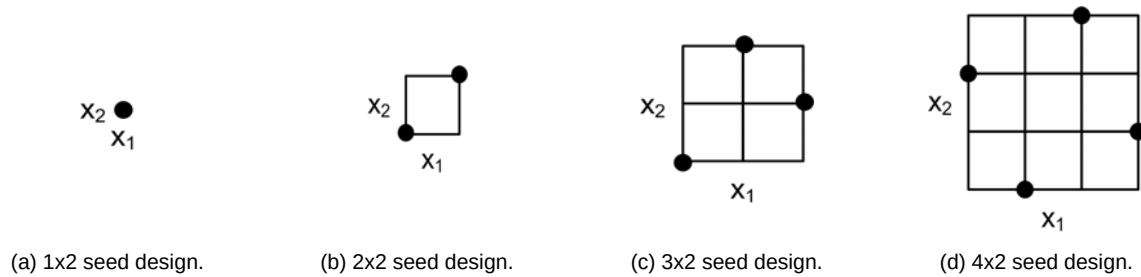


Figure 5.18. Examples of seed designs for 2 design variables.

Second, the design space is divided into blocks, in such a way that each dimension is divided in the same number of blocks. The result is that each block can be filled using the seed design (defined previously). It is clear that these processes are inter-dependent. The seed size, i.e., the number of points in the seed design, and the final design size will determine the number of blocks in each dimension. In general, the following relations must be observed:

$$p = n_{blocks} \times n_{seed} , \quad (0.19)$$

$$n_{blocks} = (n_{divisions})^{n_{dv}} , \quad (0.20)$$

$$n_{seed} = \frac{p}{n_{blocks}} . \quad (0.21)$$

For the seed design from Figure 5.18-(a), Figure 5.19 shows how the 16×2 Latin Hypercube mesh will be divided into blocks. It is important to point out that the fact that each block has four rows and four columns of the Latin Hypercube mesh does not mean that each block will have four points at the end of the process. Instead of that, this is a way to ensure the minimal distance between points in the final Latin Hypercube design.

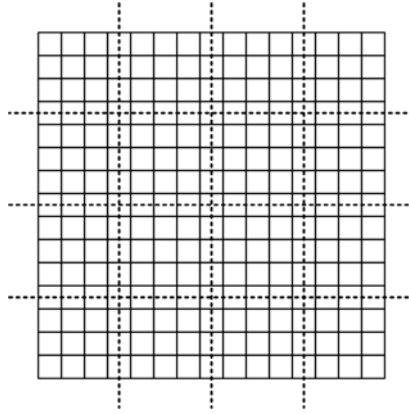


Figure 5.19. 16×2 Latin Hypercube mesh divided into blocks.

The seed design must be properly placed into each of the blocks. Figure 5.20 illustrates how this process works. The first step is to properly scale the “seed design” and then place it at the origin. Next, a set of shifts must be performed. The first one is to shift the seed to consecutive blocks following one of the dimensions. The second one is to shift the origin of the seed inside the mesh of the block. There is a coupling between these two processes. If the block shift is performed in the rows, the seed-origin shift must be performed in the columns, and vice-versa. This process is repeated until one of the dimensions is fulfilled. After that, the whole set of points placed in that dimension can be used to feedback the “shifting” process that continues fulfilling the next dimensions.

The biggest advantage of this approach is that there are no calculations to perform. All operations can be viewed as translations of an n_{seed} -points in the n_{dv} -dimensional hypercube. Although efficient for generating large designs, the previous algorithm fails to provide the flexibility for the number of points that the user may want. The approach described above is limited by the relationship:

$$p = (n_{divisions})^{n_{dv}} , \quad (0.22)$$

which follows directly from the Eqs. (0.20) and (0.21) for the smallest seed size of $n_{seed} = 1$. It means that the described algorithm is restricted to have the number of points to be equal to the integer power of the number of design variables.

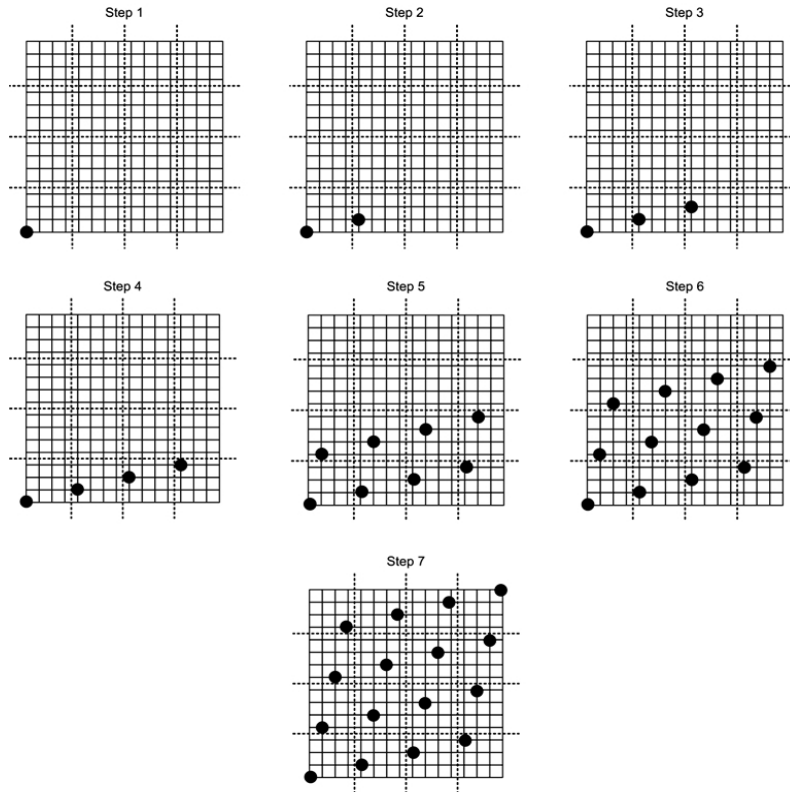


Figure 5.20. The process of creating the 16×2 Structured Latin Hypercube design.

The fact that the algorithm is capable of generating well distributed points that fill the design space well is used to overcome this deficiency. To generate the Structured Latin Hypercube design with any number of points, the first step is to generate the design that has at least the required number of points using the algorithm described above. If after generating the design we obtain the required number of points, the process is completed. If after applying the Structured Latin Hypercube algorithm the number of point is larger than required, a shrinking process is used to reduce the number of points to the desired one. The points are one-by-one removed from the initial Structured Latin Hypercube design discarding points that are the farthest from the center of the hypercube. The reason that such criterion works is that this criterion preserves the kernel of the initially generated design formed by well distributed points.

4. Results and Discussion

Figure 5.21 shows both the initial Latin Hypercube, i.e. before optimization, and the final Optimal Latin Hypercube, which is obtained by solving the optimization problem. The initial Latin Hypercube is a random design with good one-dimensional projective properties (in other words, there is only one point for each level), but with a poor space-filling property. This is typically for Latin Hypercube designs. In contrast, the Optimal Latin Hypercube design maintains the projective property while providing an excellent space filling property.

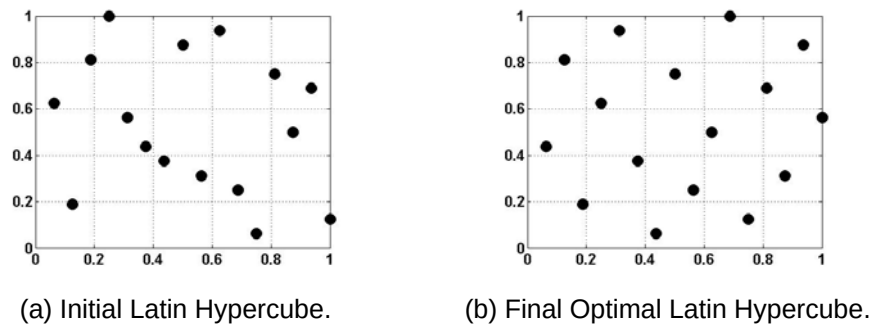


Figure 5.21. Latin Hypercube before and after optimization.

The appeal of the proposed methodology for generating Structured Latin Hypercube design is that virtually no computational time is required to create good Latin Hypercube designs. This empirical approach can be used either to quickly obtain a Latin Hypercube design with good space-filling properties or to generate a good starting point for a formal Optimal Latin Hypercube optimization procedure (like ESEA).

Table 5.15 shows the performance comparison of the ESEA procedure of generating Optimal Latin Hypercube designs starting from three different initial designs:

1. the worst design, where the points are located along the diagonal of the design space;
2. Structured Latin Hypercube design;
3. a random initial design.

For all cases, the stopping criterion used was a maximum number of 100 iterations.

Table 5.15 also allows to directly comparing the values of criterion for the Structured Latin Hypercube design with the Optimal Latin Hypercube design obtained with ESEA.

Note that for the case of two design variables (225×2 and 1024×2), the optimization procedure was not able to improve the ϕ_p -criterion of the Structured Latin Hypercube design. This indicates that for the 2D cases, the proposed empirical methodology of creating the Structured Latin Hypercube design produced the optimum results at no computation cost. For the higher dimensional cases, the formal optimization was able to make relatively small

improvements to the ϕ_p -criterion of the Structured Latin Hypercube design. This is an indication that the Structured Latin Hypercube design generated using the proposed methodology provides a reasonable approximation to the Optimal Hypercube designs in high dimensions.

Table 5.15. Optimal Latin Hypercube designs generated using ESEA from three different initial designs.

Design size	Performance	Worst case	Structured case	Random case
225×2	Time [s]	143	68	147
	Iterations	251	101	237
	ϕ_p initial	0.5571	0.0752	0.5111
	ϕ_p final	0.0756	0.0752	0.0752
1024×2	Time [s]	11155	3868	7838
	Iterations	284	101	202
	ϕ_p initial	0.5743	0.0353	0.507
	ϕ_p final	0.0421	0.0353	0.0426
256×4	Time [s]	313	372	469
	Iterations	283	336	424
	ϕ_p initial	0.2793	0.0166	0.0385
	ϕ_p final	0.0110	0.0109	0.0108
243×5	Time [s]	609	556	550
	Iterations	547	498	494
	ϕ_p initial	0.2232	0.0130	0.0267
	ϕ_p final	0.0069	0.0068	0.0068
1024×10	Time [s]	34283	27772	29421
	Iterations	601	489	518
	ϕ_p initial	0.2297	0.0044	0.0109
	ϕ_p final	0.0023	0.0023	0.0023

Finally, Figure 5.22 shows how the shrinking process works. In order to generate a 18×2 Latin Hypercube, a 27×2 Optimal Latin Hypercube initially feeds the shrinking algorithm. The space filling property is preserved on the final design.

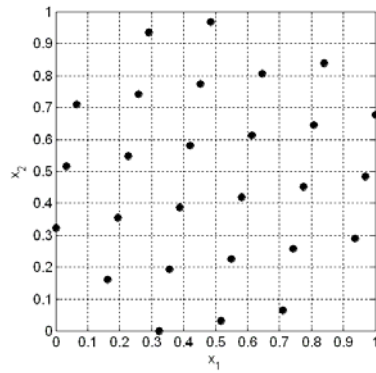
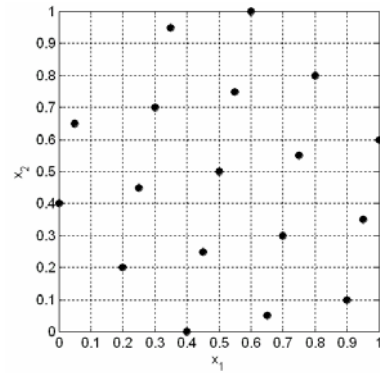
(a) Initial 27×2 Latin Hypercube.(b) Final 18×2 Latin Hypercube.

Figure 5.22. Shrinking process.

5. Summary and Conclusions

The application discussed two affordable algorithms for constructing the Optimal Latin Hypercube Design of Experiments. The first one uses the ESEA and an efficient method for evaluating the optimality criteria (ϕ_p). The second one uses an empirical approach to creating Structured Latin Hypercube designs that are good approximations of Optimal Latin Hypercube designs.

Test cases in low dimension show that the Structured Latin Hypercube approach produces designs that could not be improved using a formal optimization approach. In higher dimensions, the formal optimization approach could make small improvements to the designs obtained from the empirical approach. Future work will include both the discussion about the influence of the “seed” on the final design and enhancements on the algorithm in order to work better in higher dimensions.

CHAPTER VI

SUMMARY, CONCLUSIONS AND FUTURE WORK

This chapter summarizes this research work, presents its main conclusions and contributions and discusses the prospects for future work.

As stated in the beginning, the main goal of this contribution was to develop methodologies for applying heuristic optimization techniques to the solution of problems in engineering. Firstly, fundamental concepts in surrogate modeling and numerical optimization were revisited. The set of activities performed during the solution of an optimization problem as well as general concepts on the formulation of the optimization problem as treated by heuristic optimization algorithms were discussed. Then, the heuristic optimization methods used in this research were presented. Besides the basic version of these methods, some theoretical contributions were also introduced. At this point, challenges behind the use of heuristic methods in real world problems, such as the large number of function evaluations and the proper optimization problem formulation, were analyzed. Other than the direct use of actual models, three different approaches for effective management of models with different levels of fidelity were illustrated. They are the direct use of surrogate models, the use of the variable fidelity framework, and the coupling of both actual and surrogate models. It is clear that surrogate models appear as an alternative to alleviate high computational costs associated with heuristic optimization methods. Finally, a set of applications solved during this research showed the success of using heuristic optimization techniques. Important lessons from this work and the scope of future work are briefly presented as follows.

First, a summary of the theoretical contributions to the basic heuristic optimization algorithms is given:

- DE and GA implemented with multiple instances of operators: the developed computational code allows the use of multiple schemes for both DE and GA. For instance, DE can be simultaneously set up with a combination of DE/rand/1/bin and DE/best/1/exp; or GA can be set up with two different selection operators (e.g. roulette and tournament).
- Additional randomness for ACO and DE: the general framework, as presented by Figure 3.6, makes possible to include additional randomness in these two algorithms,

whose basic implementations have limited capabilities of avoiding premature convergence.

- Dispersion measure on the normalized design space: in opposition to schemes based on the function space, the proposed approach prevents difficulties in detecting aggregation of the population when the function space contains several minima. In addition, this is also useful in the context of inverse problems, where besides the objective function, the set of design variables define the solution.
- Implementation of ESEA in an environment of commercial software development to generate the optimal Latin hypercube.

About mixing surrogates, extra actual function evaluations, and heuristic optimization algorithms, during the present doctoral research, two variations of a scheme for updating the surrogates were proposed. The following aspects are explored: (i) large set of different surrogates is used during both off-line and on-line phases of the optimization; (ii) update of the surrogates during the on-line phase of the optimization through a pre-defined number of extra actual function evaluations (high-fidelity analysis); and (iii) *BestPRESS* (i.e. surrogate with the smallest *PRESS* value of the set) is used for global search. Remembering that *PRESS* is used as an estimator of the *RMSE*, the idea is to choose the surrogate with the best global approximation capabilities. The proposed variations differ in the update of the surrogates. While one updates the ensemble every time a new point is included in the data set, the other one performs the updating after the set of new points is generated.

Now, a summary and lessons from the applications:

1. Three-dimensional vehicular structure design optimization: this first application demonstrated the use of heuristic optimization algorithms directly coupled with finite element models to solve the design optimization of a three-dimensional vehicular structure. In the context of this doctoral work, this is an example of direct use of high-fidelity models. It is also important to say that the heuristic algorithms have successfully handled (i) a combination of continuous and discrete design variables, and (ii) constrained multi-criteria optimization. The results were encouraging regarding the possibility of adding system requirements, such as dynamic and manufacture constraints, during the optimization task.
2. Optimization of aircraft structural components by using heuristic algorithms and multi-fidelity approximations: in this application, the coupling of non-linear high-fidelity and linear low-fidelity analyses through the variable fidelity approach allowed the use of

intensive computing heuristic techniques for the optimal design of aircraft structural components. The study allows concluding that the use of variable fidelity together with heuristic optimization methods is a successful approach. The reasons are: (i) they do not require gradient information (which implies that the resources can be directly used for the search, and that there is no propagation of the errors due to the computation of the gradients based on corrected responses); and (ii) they have the trend to find the global or near global solution (which reduces the need of running multiple times the optimization problem).

3. Aircraft longitudinal stability and control derivatives identification by using LifeCycle and Levenberg-Marquardt optimization algorithms: this application presented an identification procedure to determine the longitudinal stability and control derivatives of a military aircraft by coupling together heuristic (global) and classical (local) optimization algorithms within the framework of the Output Error method. The individual performances of both the algorithms were tested and then a cascade-type scheme was proposed aiming at taking advantage of the global and local search capabilities of the individual algorithms. The results were very encouraging in the sense of adding complexity to the models, by embracing non-linearities for example, in further research.
4. Identification of a non-linear landing gear model using heuristic optimization: this application was dedicated to the identification of an aircraft landing gear model by using a heuristic optimization technique. A set of measures that describe the behavior of the system and its numerical counterparts are used together to build a representative functional, which once minimized gives a possible solution for the identification problem. A good match between the theoretical results and those obtained from the experimental drop-test was found. Encouraging future research involving more complex models, such as the ones in which the wheel rotation for the representation of drag loads and the flexibility of the landing gear are to be included (in addition to the two degrees of freedom considered in the present model).
5. Improving surrogate models by coupling heuristic optimization and extra high-fidelity function evaluations: in this application, the use of multiple surrogates, heuristic optimization methods, and extra high-fidelity simulations for minimization of the $RMSE$ in meta-modeling were explored. The key points of the methodology are (i) the generation of a large set of surrogates and the use of $PRESS_{RMS}$ as a criterion for

surrogate selection; and (ii) the use of heuristic optimization for the generation of extra points for high-fidelity simulation. Two different strategies for updating the surrogates were studied (with and without the update of the surrogates after the inclusion of each point in the data set). Based on a set of standard test functions, the study leads to the conclusion that (i) the benefits of both strategies depend on dimensionality and number of points; (ii) *BestPRESS* efficiently indicates new points for extra simulation; and (iii) while the gains over the design space are limited by the number of design variables and the complexity of the function, the gains in the region of the optima are less sensitive.

6. On how to implement an affordable Optimal Latin Hypercube: the application discussed how to obtain the Optimal Latin Hypercube Design of Experiments by using either the ESEA and an efficient method for evaluating the optimality criteria (ϕ_p), or an empirical approach (creating Structured Latin Hypercube designs, which are good approximations of Optimal Latin Hypercube designs). Test cases in low dimension show that the Structured Latin Hypercube approach produces designs that could not be improved using a formal optimization approach. In higher dimensions, the formal optimization approach could make small improvements to the designs obtained from the empirical approach.

More generally, during this doctoral research the following points were achieved:

- The use of non-conventional methods for system identification was explored. Heuristic algorithms presented a robust alternative to the solution of problems in which experimental data are corrupted or in which the models are not sufficiently known.
- The use of heuristic methods coupled with statistical tools for the solution of design problems was consolidated. This approach has been used (i) as a way to overcome the limitations of classic methods and (ii) as an alternative to decrease the computational costs that are commonly associated with heuristic optimization methods.
- The use of the developed techniques to solve various representative real world problems.
- The implementation of both a general-purpose optimization toolbox (SIMPLE Toolbox) and a surrogate modeling toolbox (SURROGATES Toolbox).

The main learning is that, as it also happens with classical optimization algorithms, the success in using heuristic methods is highly dependent on the formulation of the problem. This also includes the level of fidelity, the quality of information already known, and the

computational budget. In this sense, the choice of fidelity levels to be used and the proper handling strategy are both problem dependent. Additionally, since a large number of points of the design space is naturally visited when using heuristic algorithms, it is natural that the proper analysis of the whole set of points represents a complementary phase.

Aiming at amplifying the employment of heuristic optimization algorithms in real world problems, the following suggestions for future research work are given:

1. Heuristic optimization algorithms:

- Comparison of the performance of different DE schemes against simultaneous use of DE schemes. The same study for GA (regarding GA operators for that matter) should be implemented. At the end of this proposed study, the following question should be answered: what is the best strategy to be followed, a single scheme or a composition of different operators (and how many)?
- Development of a self-tuning approach for selecting different DE or GA schemes according to the performance in the early stages of the optimization. Independently of single or multiple operators, is it possible to choose the operators based on the first iterations of the optimization task?
- Study on the effectiveness of the randomness operator for ACO and DE. When use it and when not?
- LC outperforms the worst suited heuristic for a given problem. However, it also hurts the best one. This way, is it possible to speed up the selection of a heuristic method for a given problem?

2. Heuristic optimization in real world problems: consolidation of these algorithms through different test cases. Based on a larger set of applications, is it possible to build a general procedure that first evaluates the potential of using heuristic algorithms for a specific application and then gives general directions about their use?

3. Optimal Latin Hypercube: discussion about the influence of the “seed” on the final design and enhancements on the algorithm for the Structural Latin Hypercube, in order to work better in higher dimension problems.

4. Surrogate improvement through DOE refinement and heuristic optimization:

- Enhancements on the update algorithm, especially in the phase of surrogate updates.
- Application of the proposed approaches to real world problem.

APPENDIX A

SIMPLE OPTIMIZATION TOOLBOX

This is a short appendix about the SIMPLE Toolbox (SimpleToolBox), a set of functions for numerical optimization. The development is part of this PhD. For further reference, please check Viana and Steffen (2008).

A.1. What is the SIMPLE Optimization Toolbox?

SIMPLE Optimization ToolBox (SimpleToolBox) is a software that provides tools for creating, editing and solving general optimization problems. This way, you have access to a set of routines that can be easily used to deal with engineering design, inverse and multidisciplinary optimization problems. The current version includes:

- Multi-objective functions: weighted sum, compromise programming, and weighted min-max implementations.
- Constrained optimization functions: static penalties and dynamic penalties implementations.
- Solvers: Ant Colony Optimization, Differential Evolution, Genetic Algorithm, Particle Swarm Optimization, and LifeCycle Method.
- Report Generators: several plot functions, e.g. stopping criteria, objective statistics, and trace of the best score.

Examples:

- Mathematical functions: single objective, multi-objective, and constrained optimization.
- Design: PID controller, pressure vessel, and welded beam.
- Identification: fault diagnosis, frequency and time domain identification.

Figure A.1 illustrates the functionalities available in SimpleToolBox.

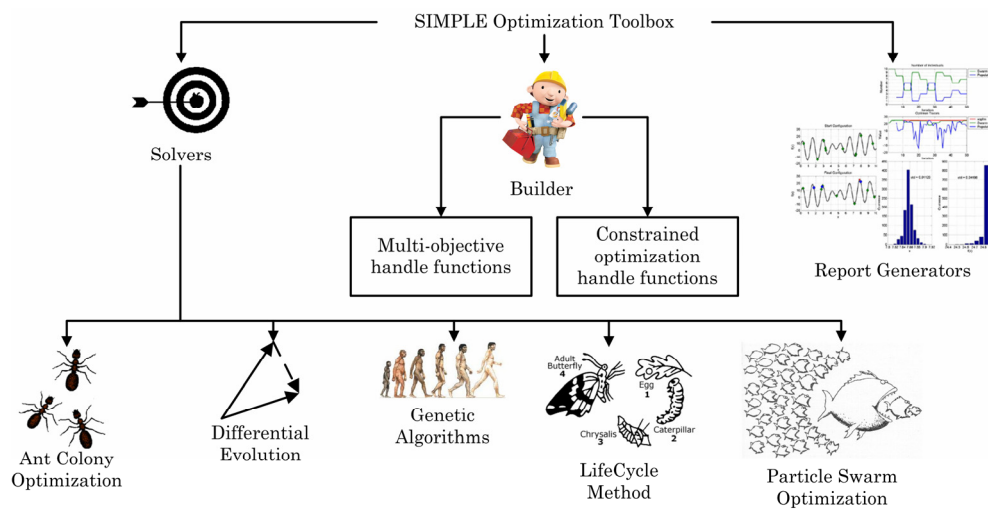


Figure A.1. SimpleToolBox facilities.

A.2. Installation and Uninstallation

Both the installation and the uninstallation processes are quite “simple”. To install SimpleToolBox, the first step is to download from <<http://fcchegury.110mb.com>>. Then unzip the SimpleToolBox, open a MATLAB terminal and go to the directory where it is the SimpleToolBox (for example, D:\projects\matlab\SimpleToolBox4p0, however, there is no preference for where the user should unzip it). Then type:

```
>> cd setup\
>> setup
```

Figure A.2 shows what you would see.

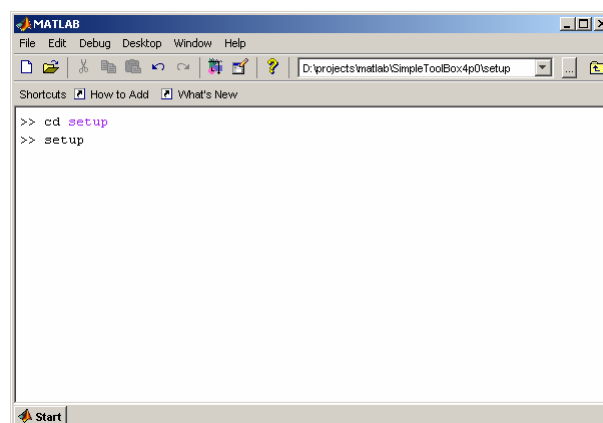


Figure A.2. First step of installation (or uninstallation).

At this moment, the setup routine will help the user to install (or uninstall) the current version of the SimpleToolBox (see Figure A.3).

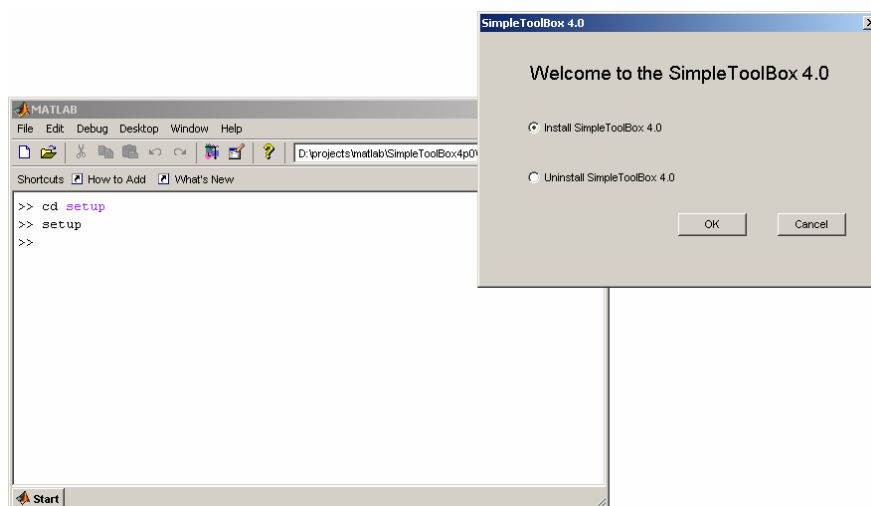


Figure A.3. Second step of installation (or uninstallation).

Observations:

1. The SimpleToolBox has been tested in both MATLAB® Version 7.0 Release 14 and MATLAB® Version 6.5 Release 13.
2. In practice, either the installation or uninstallation will change only the MATLAB search path. It means the NO files will be copied or moved.

To check the current version of the SimpleToolBox, open a MATLAB terminal and type:

```
>> srgtsVersion
```

Figure A.4 shows what you see for the current version.

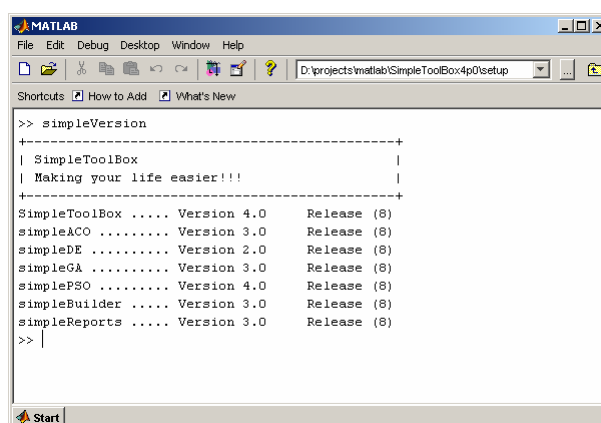


Figure A.4. SimpleToolBox version.

A.3. Help on Functions

All MATLAB functions in toolbox have prefix “simple” (except setup.m). To get help on a specific function just type:

```
>> help function_name
```

See Figure A.5 for an example.

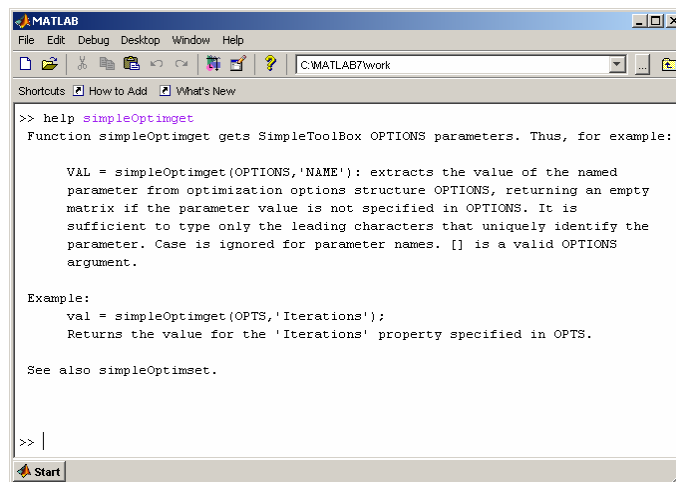


Figure A.5. Help on simpleOptimget.

A.4. Bug Reports

To check the history of bug reports, go to BugBuster at <<http://fcchegury.110mb.com>>.

A.5. Simpletoolbox/Surrogatestoolbox User's List

If the user wants to receive periodical information about both the SimpleToolBox and the SimpleToolBox, he/she can join to the SimpleToolBox/SurrogatesToolBox User's List. It is easy; just send an e-mail to Felipe (fcchegury@yahoo.com) with the following data: full name, e-mail address, and affiliation.

By doing that, the user will be informed about the state of development of those toolboxes, including updates on the code, change of standards, tendencies, etc. If the user no longer wants to receive e-mails from this list, he/she just have to send me an e-mail to Felipe (fcchegury@yahoo.com) asking to exclude his/her address from the list.

APPENDIX B

SURROGATES TOOLBOX

This is a short appendix about the SURROGATES Toolbox (SurrogatesToolBox), a set of functions for surrogate modeling. The first functions were developed during the PhD work of Dr. Tushar Goel (Goel, 2007). During a exchange program in 2007, the development was conducted together with the present research. For further reference, please check Viana and Goel (2008).

B.1. What is the SURROGATES Toolbox?

SURROGATES Toolbox (SurrogatesToolBox) provides tools for creating and editing different surrogate models (commonly used to replace expensive simulations of engineering problems). The current version includes:

Design of Experiments:

- Face centered cubic design (FCC).
- Combination of a face centered cubic design and a ordinary (but improved) latin hypercube design (FCCLH).
- Mixed-level full-factorial design (FFD).
- Latin hypercube sampling (LHS is especially suitable for very large designs, when the native MATLAB function run out of memory).
- Combination of latin hypercube and D-optimal design (LHDOptimal).
- Filling of user-defined design with a latin hypercube design (LHFFilling).
- Selection of points from a user-defined design according to a specific optimality criterion (OptimalSubDoE can use "MaxMin" or "DOptimal" criteria).

Surrogates: Kriging, polynomial response surface, radial basis neural network, support vector regression, and weighted average surrogates.

Performance Metrics:

- Based on test points: correlation coefficient, RMS error, maximum absolute error, and error matrix.
- Based on data points: PRESS and cross-validation errors.

Global Sensitivity Analysis using two different approaches: Monte Carlo and Gaussian Quadrature.

Other capabilities:

- Selection of cross-validation sets following the “k-fold” strategy.
- Generation of a text report file.

Examples:

- Design of experiment examples.
- Braninhoo function.
- Engineering problem 1: Diffuser.
- Engineering problem 2: Radial turbine.
- Data structure update (from previous version to the current version of the toolbox).

Figure B.1 illustrates the functionalities available in SurrogatesToolBox.

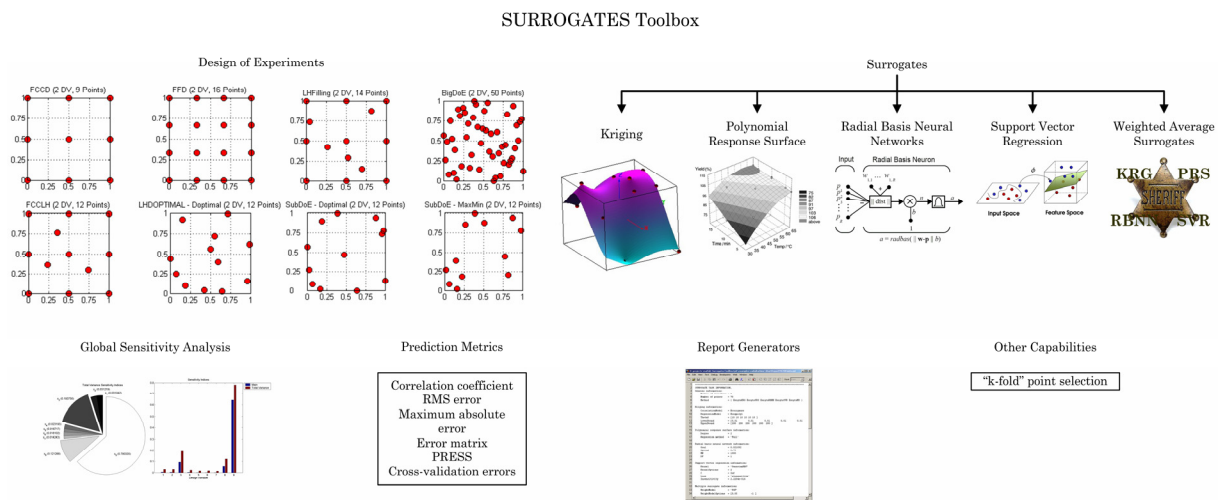


Figure B.1. SurrogatesToolBox facilities.

B.2. Installation and Uninstallation

Both the installation and the uninstallation processes are quite “simple”. To install SurrogatesToolBox, the first step is to download from <http://fchegury.110mb.com>. Then unzip the SurrogatesToolBox, open a MATLAB terminal and go to the directory where it is the SurrogatesToolBox (for example, D:\projects\matlab\SurrogatesToolBox1p1, however, there is no preference for where the user should unzip it). Then type:

```
>> cd setup\  
>> setup
```

Figure B.2 shows what you would see.

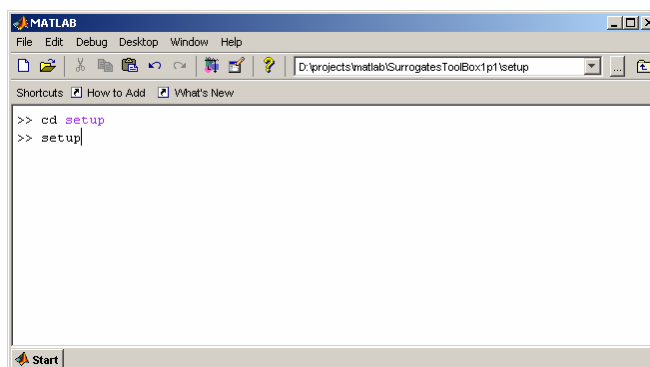


Figure B.2. First step of installation (or uninstallation).

At this moment, the setup routine will help the user to install (or uninstall) the current version of the SurrogatesToolBox (see Figure B.3).

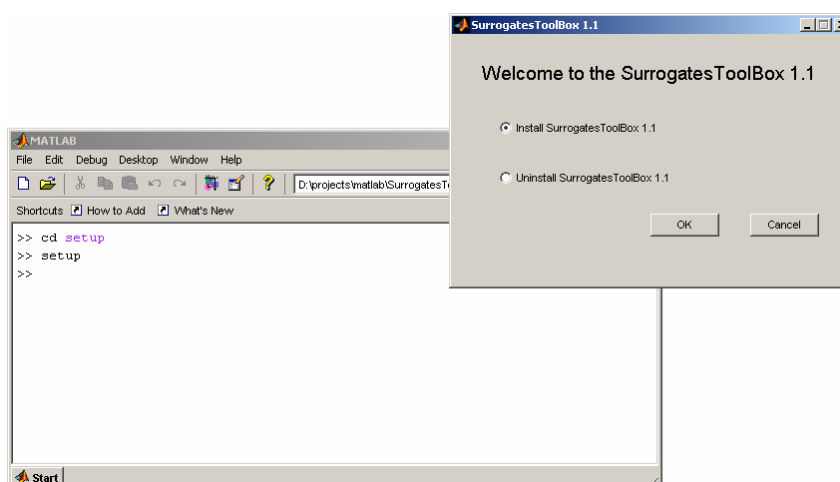


Figure B.3. Second step of installation (or uninstallation).

Observations:

1. The SurrogatesToolBox has been tested in both MATLAB® Version 7.0 Release 14 and MATLAB® Version 6.5 Release 13.
2. In practice, either the installation or uninstallation will change only the MATLAB search path. It means the NO files will be copied or moved.

To check the current version of the SurrogatesToolBox, open a MATLAB terminal and type:

```
>> srgtsVersion
```

Figure B.4 shows what you see for the current version.

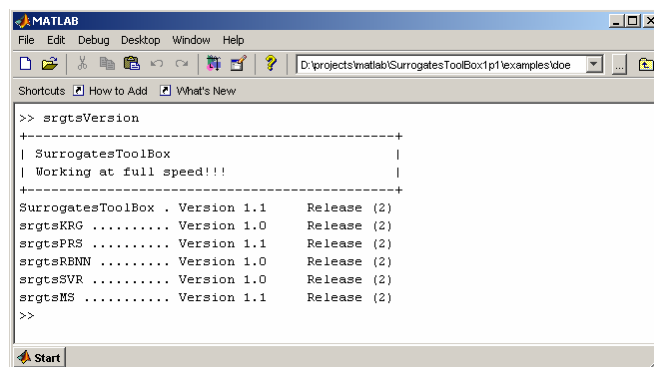


Figure B.4. SurrogatesToolBox version.

B.3. Help on Functions

All MATLAB functions in toolbox have prefix “srgts” (except `setup.m`). To get help on a specific function just type:

```
>> help function_name
```

See Figure B.5 for an example.

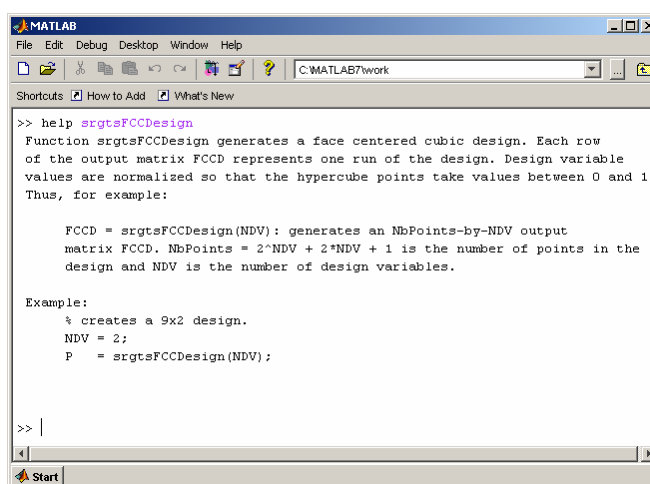
B.4. Bug Reports

To check the history of bug reports, go to BugBuster at <http://fchegury.110mb.com>.

B.5. Simpletoolbox/Surrogatestoolbox User's List

If the user wants to receive periodical information about both the SimpleToolBox and the SurrogatesToolBox, he/she can join to the SimpleToolBox/SurrogatesToolBox User's List. It is easy; just send an e-mail to Felipe (fchegury@yahoo.com) with the following data: full name, e-mail address, and affiliation.

By doing that, the user will be informed about the state of development of those toolboxes, including updates on the code, change of standards, tendencies, etc. If the user no longer wants to receive e-mails from this list, he/she just have to send me an e-mail to Felipe (fchegury@yahoo.com) asking to exclude his/her address from the list.



```

MATLAB
File Edit Debug Desktop Window Help
C:\MATLAB7\work

Shortcuts How to Add What's New

>> help srgtsFCCDesign
Function srgtsFCCDesign generates a face centered cubic design. Each row
of the output matrix FCCD represents one run of the design. Design variable
values are normalized so that the hypercube points take values between 0 and 1.
Thus, for example:

    FCCD = srgtsFCCDesign(NDV): generates an NbPoints-by-NDV output
matrix FCCD. NbPoints = 2*NDV + 2*NDV + 1 is the number of points in the
design and NDV is the number of design variables.

Example:
% creates a 9x2 design.
NDV = 2;
P = srgtsFCCDesign(NDV);

>>

```

Figure B.5. Help on srgtsFCCDesign.

APPENDIX C

BOXPLOTS

In a boxplot, the box is defined by lines at the lower quartile (25%), median (50%), and upper quartile (75%) values. Lines extend from each end of the box to show the coverage of the rest of the data (i.e., they are plotted at a distance of 1.5 times the inter-quartile range in each direction or the limit of the data, if the limit of the data falls within 1.5 times the inter-quartile range). Outliers are data with values beyond the ends of the lines by placing a “+” sign for each point.

See an example given in the MATLAB tutorial (MathWorks Contributors, 2002). The following command lines create a box plot of car mileage grouped by countries.

```
>> load carsmall  
>> boxplot(MPG, Origin)
```

Figure C.1 shows the results of the boxplot command.

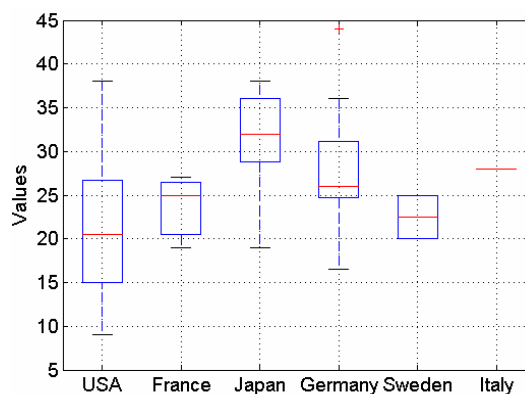


Figure C.1. Example of boxplot figure.

APPENDIX D

NELDER-MEAD SIMPLEX DIRECT SEARCH AND LEVENBERG-MARQUADT OPTIMIZATION METHODS

D.1. Nelder-Mead Simplex Direct Search

As seen in Lagarias et al. (1998), the Nelder-Mead simplex direct search (NMSDS) does not use numerical or analytic gradients. This method uses the simplex concept, which in the n_{dv} -dimensional space is characterized by the $n_{dv} + 1$ distinct points in its vertices (e.g., if $n_{dv} = 2$, a simplex is a triangle). At each step of the search, a new point in the neighborhood of the current simplex is generated. One of the vertices is replaced by this new point if the new function value is smaller than the values at the vertices of the simplex (which generates a new simplex). This step is repeated until the diameter of the simplex is less than a specified tolerance. The MATLAB *fminsearch* function (Mathworks Contributors, 2002), set with the default options, is used as implementation of the NMSDS algorithm.

D.2. Levenberg-Marquadt Algorithm

The Levenberg-Marquardt method (LM) is a second order variant of the Gauss-Newton method (Gill et al., 1981). This method, although complex, is suitable for a quadratic cost function, and is expected to converge quickly. First, $f(\mathbf{x})$ is approximated by a parabolic function $f_L(\mathbf{x})$ under the condition $\mathbf{x}_L$ (retaining only the 3 first Taylor series terms):

$$f_L(\mathbf{x}) \cong f_L(\mathbf{x}_L) + (\mathbf{x} - \mathbf{x}_L)^T \nabla_{\mathbf{x}}^T f(\mathbf{x}_L) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_L)^T [\nabla_{\mathbf{x}}^2 f(\mathbf{x}_L)](\mathbf{x} - \mathbf{x}_L). \quad (\text{D.1})$$

Let the Jacobian of $f(\mathbf{x})$ be denoted by $\mathbf{J}$, then the LM method searches in the direction given by the solution $\mathbf{p}$ to the equations:

$$(\mathbf{J}_k^T \mathbf{J}_k + \lambda_k \mathbf{I}) \mathbf{p}_k = -\mathbf{J}_k^T f_k, \quad (\text{D.2})$$

where λ_k are nonnegative scalars and $\mathbf{I}$ is the identity matrix.

The LM algorithm can be interpreted in the following manner: for small values of λ_k it behaves as the Gauss-Newton algorithm, while for high values of λ_k it behaves as the steepest gradient algorithm.

In the scope of this work, either NMSDS or LM are employed with initial designs given by the candidate solutions of the heuristic optimizers. This cascade-type scheme using a heuristic algorithm for global search and a classical algorithm for local search is believed to reduce the changes of fail because of local minima.

REFERENCES

AIRD, F. **Race Car Chassis Design and Construction**. Osceola, USA: Motorbooks International, 1997.

ALEXANDROV, N. M.; LEWIS, R. M.; GUMBERT, C. R.; GREEN, L. L.; and NEWMAN, P. A. **Optimization with Variable-fidelity Models applied to Wing Design**, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, USA, 1999, ICASE Technical Report N. 99-49.

ANSYS CONTRIBUTORS, ANSYS ICEM CFD Documentation Manuals, ANSYS Inc, 2006.

BATES, S.; SIENZ, J.; and TOROPOV, V. Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Apr., 2004, Palm Springs, USA. **Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference**.

BISHOP, C. M. **Neural Networks for Pattern Recognition**. Oxford University Press, 1995. pp. 364-371.

BOX, G. E. P.; HUNTER, W. G.; and HUNTER, J. S. **Statistics for Experimenters**. 1st edition. New York, USA: John Wiley & Sons, 1978.

BRUHN, E. F. **Analysis and Design of Flight Vehicle Structures**. Jacob Pub, 1973.

CANTU-PAZ, E. Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms. **Journal of Heuristics**. Vol. 7, N. 4, pp. 311-334, Jul. 2001.

CHENG, B.; and TITTERINGTON, D.M., 1994, Neural Networks: a Review from a Statistical Perspective. **Statistical Science**, Vol. 9, N. 1, pp. 2-30, Feb., 1994.

CHERKASSKY, V., and MA, Y. Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. **Neural Networks**. Vol. 17, N. 1, pp. 113-126, 2004.

CLARKE, S. M.; GRIEBSCH, J. H.; and SIMPSON, T. W. Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. **Journal of Mechanical Design**. Vol. 127, Issue 6, pp.1077-1087, Nov. 2005.

COELLO, C. A. C. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. **Knowledge and Information Systems**. Vol. 1, N. 3, pp. 129-156, 1999.

COELLO, C. A. C. Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. **Computer Methods in Applied Mechanics and Engineering**. Vol. 191, N. 11-12, pp.1245-1287, Jan. 2002.

COELLO, C. A. C. An Introduction to Evolutionary Algorithms and their Applications. **Lecture Notes in Computer Science**. Vol. 3563, pp.425-442, Aug. 2005.

CURREY, N. S. **Landing Gear Design Handbook**. Lockheed-Georgia Company, USA, 1984.

DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**. 1st edition. Chichester, UK: Wiley, Jun. 2001.

DIXON, L. C. W.; and SZEGÖ, G. P. **Towards global optimization 2**. North-Holland, Amsterdam, The Netherlands, 1978.

DORIGO, M. **Optimization, Learning and Natural Algorithms**. 1992. PhD thesis, Politecnico di Milano, Italy.

DORIGO, M. and GAMBARDELLA, L. M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. **IEEE Transactions on Evolutionary Computation**, Vol. 1, N. 1, pp. 53-66, Apr. 1997.

DORIGO, M., MANIEZZO, V., and COLORNI, A. The Ant System: Optimization by a Colony of Cooperating Agents. **IEEE Transactions on Systems, Man, and Cybernetics-Part B**. Vol. 26, N. 1, pp.1-13, Feb. 1996.

FLORES, J. E. R.; VIANA, F. A. C.; RADE, D. A.; and STEFFEN Jr, V. Identification of External Forces in Mechanical Systems by using Lifecycle Model and Stress-Stiffening Effect. **Mechanical Systems and Signal Processing**. Vol. 21, Issue 7, pp.2900-2917, Oct. 2007.

FOURIE, P. C. and GROENWOLD, A. A. The Particle Swarm Optimization Algorithm in Size and Shape Optimization. **Structural and Multidisciplinary Optimization**. Vol. 23, N. 4 pp. 259-267, May, 2002.

GANO, S. E.; RENAUD, J. E.; and SANDERS, B. Variable Fidelity Optimization using a Kriging Based Scaling Function. IN: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. Aug. 30 – Sep. 1, 2004, Albany, USA. **Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference**.

GEN, M. and CHENG, R. **Genetic Algorithms and Engineering Optimization**. New York, USA: Wiley-Interscience, Dec., 1999.

GERACE, S. A.; KASSAB, A. J.; and DIVO, E. An Automated Approach to Multiobjective Shape Optimization for Engineering Design Problems. IN: INVERSE PROBLEMS, DESIGN AND OPTIMIZATION SYMPOSIUM, Apr. 16-18, 2007, Miami, USA. **Proceedings of IPDO2007**.

GILL, P. R.; MURRAY, W.; and WRIGHT, M. H. The Levenberg-Marquardt Method. **Practical Optimization**. London: Academic Press, pp. 136-137, 1981.

GOEL, T. **Multiple Surrogates and Error Modeling in Optimization of Liquid Rocket Propulsion Components**. 2007. PhD Thesis, Department of Mechanical and Aerospace Engineering, University of Florida, USA.

GOEL, T.; HAFTKA, R. T.; SHYY, W.; and QUEIPO, N. V. Ensemble of Surrogates. **Structural and Multidisciplinary Optimization**. Vol. 33, N. 3, pp. 199-216, Mar. 2007.

GÓES, L. C. S.; MACIEL, B. C. O.; NETO, N. S. B.; VIANA, F. A. C.; and STEFFEN Jr, V. On the Use of a Global Search Method and a Gradient Based Method for the Identification of Aircraft Longitudinal Stability and Control Derivatives. IN: INVERSE PROBLEMS, DESIGN AND OPTIMIZATION SYMPOSIUM, Apr. 16-18, 2007, Miami, USA. **Proceedings of IPDO2007**.

GOODWIN, G. C. and PAYNE, R. L. **Dynamic System Identification - Experiment Design and Data Analysis**. Academic Press, New York, USA, 1977.

GUNN, S. R. **Support Vector Machines for Classification and Regression**. Image Speech and Intelligent Systems Research Group, University of Southampton, UK, Technical Report, 1998.

HAFTKA, R. T. and GÜRDAL, Z. **Elements of Structural Optimization**. 3rd edition. Dordrecht, Netherlands: Kluwer Academic Publishers, 1992.

HAUPT, R. L. and HAUPT, S. E. **Practical Genetic Algorithms**. 2nd edition. New York, USA: Wiley-Interscience Publication, May 2004.

ILIFF, K. W. Parameter Estimation for Flight Vehicles. **Journal of Guidance, Control and Dynamics**, Vol. 12, N. 5, pp. 609-622, 1989.

IMAN, R. L. and CONOVER, W. J. Small Sample Sensitivity Analysis Techniques for Computer Models, with an Application to Risk Assessment. **Communications in Statistics, Part A. Theory and Methods**. Vol. 17, pp.1749-1842, 1980.

JIN, R., CHENA, W. and SUDJANTO, A. An Efficient Algorithm for Constructing Optimal Design of Computer Experiments. **Journal of Statistical Planning and Inference**. Vol. 134, N. 1, pp. 268-287, Sep. 2005.

JIN, Y.; OLHOFFER, M.; and SENDHOFF, B. A Framework for Evolutionary Optimization with Approximate Fitness Functions. **IEEE Transactions on Evolutionary Computation**, Vol. 6, No. 5, pp. 481-494, 2002.

JOHNSON, M.; MOORE, L.; and YLVISAKER, D. Minimax and Maximin Distance Designs. **Journal of Statistics Planning and Inference**, Vol. 26, pp. 131-148, 1990.

KAELO, P. and ALI, M. M. A Numerical Study of Some Modified Differential Evolution Algorithms. **European Journal of Operational Research**. Vol. 169, Issue 3, pp.1176-1184, Mar. 2006.

KELLER, G. R. **Hydraulic System Analysis**. Published by the Editors of Hydraulics & Pneumatics Magazine. USA, 1974.

KENNEDY, J. and EBERHART, R. C. Particle Swarm Optimization. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS. Nov. 1995, Perth, Australia. **Proceedings of the 1995 IEEE International Conference on Neural Networks**. pp. 1942-1948.

KOHAVER, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. FOURTEENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE. Aug. 20-25, 1995, Montréal, Québec, Canada. **Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence**, pp. 1137-1143.

KOSSIAKOFF, A. and SWEET, W. N. **Systems Engineering Principles and Practice**. 1st edition. Hoboken, USA: Wiley-Interscience, Dec. 2002.

KRINK, T. and LØVBERG, M. The LifeCycle Model: Combining Particle Swarm Optimisation, Genetic Algorithms and Hill Climbers. **Lecture Notes in Computer Science**. Vol. 2439, pp.621-630. Feb. 2004.

LAGARIAS, J.C., REEDS, J. A., WRIGHT, M. H., and WRIGHT, P. E. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. **SIAM Journal of Optimization**, Vol. 9 N. 1, pp. 112-147, 1998.

LIANG, K.-H.; YAO, X.; and NEWTON, C. Evolutionary Search of Approximated N-dimensional Landscapes. **International Journal of Knowledge-Based Intelligent Engineering Systems**, Vol. 4, No. 3, pp. 172-183, 2000.

LOPHAVEN, S. N., NIELSEN, H. B., and SØNDERGAARD, J. **DACE – A MATLAB Kriging Toolbox**, Informatics and Mathematical Modelling, Technical University of Denmark, Technical Report IMM-TR-2002-12, 2002.

NEMHAUSER, G. L. and WOLSEY, L. A. **Integer and Combinatorial Optimization**. New York, NY, USA: Wiley-Interscience, 1988.

MARDUEL, X.; TRIBES, C.; and TRÉPANIER, J. Y. Variable-Fidelity Optimization: Efficiency and Robustness. **Optimization Engineering**. Vol. 7, pp.479–500, 2006.

MARLER, R. T. and ARORA, J. S. Survey of Multi-Objective Optimization Methods for Engineering. **Structural and Multidisciplinary Optimization**. Vol. 26, N. 6, pp.369–395, Apr. 2004.

MARTIN, J. D. and SIMPSON, T. W. Use of Kriging Models to Approximate Deterministic Computer Models. **AIAA Journal**. Vol. 43, N. 4, pp.853-863, 2005.

MATHWORKS CONTRIBUTORS, **MATLAB® The language of technical computing**. The MathWorks Inc., 2002.

MCKAY, M.D.; BECKMAN, R.J.; and CONOVER, W.J. A Comparison of Three Methods for Selecting Values of Input Variables from a Computer Code. **Technometrics**, Vol. 21, pp. 239-245, 1979.

MECKESHEIMER, M.; BOOKER, A. J.; BARTON, R. R.; and SIMPSON, T. W. Computationally Inexpensive Metamodel Assessment Strategies. **AIAA Journal**. Vol. 40, N. 10, pp.2053-2060, Oct. 2002.

MICHALEWICZ, Z. Genetic Algorithms, Numerical Optimization and Constraints. In: 6th INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, July 15-19 1995, Pittsburgh, USA. **Proceedings of the 6th International Conference on Genetic Algorithms**. pp. 151-158.

MICHALEWICZ, Z. and FOGEL, D. B. **How to Solve it: Modern Heuristics**. 1st edition. New York, USA: Springer-Verlag, Jan. 2000.

MONMARCHÉ, N.; RAMAT, E.; DROMEL, G.; SLIMANE, M.; and VENTURINI, G. **On the Similarities between AS, BSC and PBIL: toward the Birth of a New Meta-heuristic**, Ecole d'Ingénieurs en Informatique pour l'Industrie, Université de Tours, Technical Report E3i, 215, 1999.

MONTGOMERY, D.C. **Design and Analysis of Experiments**. 4th edition. New York, USA: John Wiley & Sons, 1997.

MORRIS, M. D. and MITCHELL, T. J. Exploratory Designs for Computational Experiments. **Journal of Statistics Planning and Interference**. Vol. 43, pp. 381-402, 1995.

MYERS, R.H. and MONTGOMERY, D.C. **Response Surface Methodology: Process and Product optimization using Designed Experiments**. 2nd edition. New York, USA: John Wiley & Sons, 1995.

NELLES, O. **Nonlinear System Identification: from Classical Approaches to Neural Networks and Fuzzy Models**. New York, USA: Springer-Verlag, Jan., 2001.

NIU, M. C and NIU, M. **Airframe Structural Design: Practical Design Information and Data on Aircraft Structures**. 2nd edition. Adaso Adastra Engineering Center, 1999, pp. 398.

OLIVEIRA, F. C. G. **CONTRIBUIÇÃO AO DESENVOLVIMENTO DE UMA ESTRUTURA VEICULAR TIPO SPACEFRAME USANDO MÉTODO DOS ELEMENTOS FINITOS E MÉTODOS HEURÍSTICOS DE OTIMIZAÇÃO NUMÉRICA**. 2007. MSc Dissertation, School of Mechanical Engineering, Federal University of Uberlândia, Brazil.

OLIVEIRA, F. C. G. ; VIANA, F. A. C.; BORGES, J. A. F.; and STEFFEN Jr, V. Three-dimensional Vehicular Structure Design combining Finite Element Modeling and Numerical

Optimization. IN: 19TH INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, Brasilia, Brazil, Nov. 5-9, 2007. **Proceeding of COBEM 2007**.

ONG, Y. S., NAIR, P. B., and KEANE, A. J. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. **AIAA Journal**. Vol. 41, No. 4, Apr. 2003.

OWEN, A.B. Orthogonal Arrays for Computer Experiments, integration and visualization. **Statistica Sinica**. Vol. 2, pp.439-452, 1992.

PAPADIMITRIOU, C. H. and STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. Englewood Cliffs, New Jersey, USA: Prentice-Hall, Inc, 1998.

PARSOPOULOS, K. E. and VRAHATIS, M. N. Recent approaches to global optimization problems through Particle Swarm Optimization. **Natural Computing**. Vol. 1, N. 2-3, pp. 235-306, Jun. 2002.

POURTAKDOUST, S. H. and NOBAHARI, H. An Extension of Ant Colony System to Continuous Optimization Problems. **Lecture Notes in Computer Science**. Vol. 3172/2004, pp. 294-301, Nov., 2004.

PRICE, K, STORN, R., and LAMPINEN, J. **Differential Evolution: A Practical Approach to Global Optimization**. 1st edition. New York, USA:Springer, Dec. 2005.

QUEIPO, N. V.; HAFTKA, R. T.; SHYY, W.; GOEL, T.; VAIDYANATHAN, R.; and TUCKER, P. K. Surrogate-Based Analysis and Optimization. **Progress in Aerospace Sciences**. Vol. 41, N. 1, pp.1-28, Jan. 2005

RASMUSSEN, T. K. **Improving Particle Swarm Optimization by Hybridization of Stochastic Search Heuristics and Self-Organized Criticality**. May 2002. 80 pages. PhD thesis, University of Aarhus, Department of Computer Science, Denmark.

RONKKONEN, J.; KUKKONEN, S.; and PRICE, K. Real-parameter optimization with differential evolution. IN: 2005 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, Sep. 2-5, 2005, Edinburgh, Scotland. **Proceedings of the 2005 IEEE Congress on Evolutionary Computation**, Vol. 1, pp. 506-513.

SAAB, Y. G. and RAO, Y. B. Combinatorial Optimization by Stochastic Evolution. **IEEE Transactions on Computer-Aided Design**, Vol. 10, pp. 525–535, 1991.

SACKS, J.; WELCH, W. J.; MITCHELL, T. J.; and WYNN, H. P. Design and Analysis of Computer Experiments. **Statistical Science**, Vol. 4, N. 4, pp.409-435, Nov. 1989.

SHI, Y. and EBERHART, R. Parameter Selection in Particle Swarm Optimization. **Lecture Notes in Computer Science**, Vol. 1447, pp. 591-600, 1998.

SIMPSON, T. W.; BOOKER, A. J.; GHOSH, D.; GIUNTA, A. A.; KOCH, P. N.; and YANG, R. J. Approximation Methods in Multidisciplinary Analysis and Optimization: a Panel Discussion, **Structural and Multidisciplinary Optimization**. Vol. 27, N. 5, pp 302-313, Jul. 2004.

SIMPSON, T. W.; POPLINSKI, J.; KOCH, P. N.; and ALLEN, J. K. On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments. IN: DESIGN THEORY AND METHODOLOGY. Sep. 14-17, 1997, Sacramento, USA. **ASME-DETC97/DTM-3881**.

SIMPSON, T. W.; POPLINSKI, J. D.; KOCH, P. N.; and ALLEN, J. K. Metamodels for Computer-based Engineering Design: Survey and recommendations. **Engineering with Computers** , Vol. 17, N. 2, pp 129-150, Jul. 2001.

SMITH, M. **Neural Networks for Statistical Modeling**. 1st edition. New York, USA: Von Nostrand Reinhold, 1993.

SMOLA, A. J. and SCHOLKOPF, B. A Tutorial on Support Vector Regression. **Statistics and Computing**. Vol. 14, N. 3, pp. 199-222, Aug. 2004.

SOCHA, K. ACO for Continuous and Mixed-Variable Optimization. **Lecture Notes in Computer Science**. Vol. 3172/2004, pp. 25-36, Nov., 2004.

STORN, R., On the Usage of Differential Evolution for Function Optimization. IN: 1996 BIENNIAL CONFERENCE OF THE NORTH AMERICAN FUZZY INFORMATION PROCESSING SOCIETY, Jun. 19-22, 1996, Berkeley, USA. **Proceedings of the 1996 Biennial Conference of the North American Fuzzy Information Processing Society**. pp. 519-523.

STORN, R. and PRICE, K. **Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces**. International Computer Science Institute, Berkley, Technical Report TR-95-012, Mar. 1995.

STORN, R. and PRICE, K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**, Vol. 11, N. 4, pp.341-359, Dec. 1997.

THOMPSON, L. L.; RAJU, S. and LAW, E. H. Design of a Winston Cup Chassis for Torsional Stiffness. IN: MOTORSPORTS ENGINEERING CONFERENCE AND EXPOSITION, Nov. 16-19, 1998, Dearborn, USA. **1998 Motorsports Engineering Conference Proceedings**.

VAN VELDHUIZEN, D. A. and LAMONT, G. B. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. **Evolutionary Computation**. Vol. 8, N.2, pp.125-147, 2000.

VANDERPLAATS, G. N. **Numerical Optimization Techniques for Engineering Design**. 4th edition. Colorado Springs, USA: Vanderplaats Research and Development, Inc., 2005.

VAPNIK, V. The Nature of Statistical Learning Theory. New York, USA: Springer, 1995.

VENKATARAMAN, S. and HAFTKA, R. T. Structural Optimization Complexity: What Has Moore's Law Done for Us? **Structural and Multidisciplinary Optimization**. Vol. 28, N. 6, pp.375-387, Dec. 2004.

VENTER, G. and SOBIESZCZANSKI-SOBIESKI, J. Particle Swarm Optimization. **AIAA Journal**. Vol. 41, N.8, pp.1583-1589, 2003.

VIANA, F. A. C.; BALABANOV, V.; VENTER, G.; GARCELON, J.; and STEFFEN Jr, V. Generating Optimal Latin Hypercube Designs In Real Time. IN: 7th WORLD CONGRESS ON STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION, May 21-25, 2007a, Seoul, Korea. **Proceedings of the 7th World Congress on Structural and Multidisciplinary Optimization**. pp.2310-2315.

VIANA, F. A. C. and GOEL, T. **SURROGATES Toolbox User's Guide**. 2008. Disponível em: <<http://fcheurgy.110mb.com>>. Acesso em: fevereiro de 2008.

VIANA, F. A. C.; HAFTKA, R. T., STEFFEN Jr, V.; BUTKEWITSCH, S.; and LEAL, M. F. Optimal use of Multiple Surrogate for Reduced RMS Error in Meta-model. IN: 2008a NSF ENGINEERING RESEARCH AND INNOVATION CONFERENCE, Jan. 7-10, 2008a, Knoxville, USA. **Proceedings of 2008 NSF Engineering Research and Innovation Conference**.

VIANA, F. A. C.; HAFTKA, R. T., STEFFEN Jr, V.; BUTKEWITSCH, S.; and LEAL, M. F. Ensemble of Surrogates: a Framework based on Minimization of the Mean Integrated Square Error. IN: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Apr., 2008b, Schaumburg, Illinois, USA. **Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference**.

VIANA, F. A. C., MACIEL, B. C. O., NETO, N. S. B., OLIVEIRA, M. F., STEFFEN Jr, V., and GÓES, L. C. S. Aircraft Longitudinal Stability and Control Derivatives Identification by Using Life Cycle and Levenberg-Marquardt Optimization Algorithms. **Inverse Problems in Science and Engineering**. Accepted for publication, 2008c.

VIANA, F. A. C., OLIVEIRA, F. C. G., BORGES, J. A. F., and STEFFEN Jr, V. Differential Evolution Applied to the Design of a Three-Dimensional Vehicular Structure. IN: IDETC/CIE 2007-ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Sep. 4-7, 2007b, Las Vegas, USA. **Proceedings of IDETC/CIE 2007 ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference**

VIANA, F. A. C. and STEFFEN Jr, V. Natural Methods Applied to Direct and Inverse Problems. IN: Workshop on Nonlinear Phenomena Modeling and their Applications, May 2-4, 2005, Rio Claro, Brazil. **Proceedings of the Workshop on Nonlinear Phenomena Modeling and Their Applications**.

VIANA, F. A. C. and STEFFEN Jr, V. **SIMPLE Optimization Toolbox User's Guide**. 2008. Disponível em: <<http://fchegury.110mb.com>>. Acesso em: fevereiro de 2008.

VIANA, F. A. C., STEFFEN Jr, V., VENTER, G., and BALABANOV, V. On How to Implement an Affordable Optimal Latin Hypercube. IN: 19th International Congress of Mechanical Engineering: COBEM 2007, Nov. 5-9, 2007c, Brasília, Brazil. **Proceedings of 19th International Congress of Mechanical Engineering**.

VIANA, F. A. C.; STEFFEN Jr, V.; BUTKEWITSCH, S.; and LEAL, M. F. About the Optimum Design of an Aircraft Pressure Bulkhead by using Multi-Fidelity and LifeCycle Algorithm . IN: INVERSE PROBLEMS, DESIGN AND OPTIMIZATION SYMPOSIUM, Apr. 16-18, 2007d, Miami, USA. **Proceedings of IPDO2007**.

VIANA, F. A. C.; STEFFEN Jr, V.; BUTKEWITSCH, S.; and LEAL, M. F. Optimization of Aircraft Structural Components by using Nature-Inspired Algorithms and Multi-Fidelity Approximations. **Journal of Global Optimization**. Accepted for publication, 2008d.

VIANA, F. A. C.; STEFFEN Jr, V.; ZANINI, M. A. X.; MAGALHÃES, S. A.; and GÓES, L. C. S. Identification of a Non-Linear Landing Gear Model using Nature-Inspired Optimization. **Shock and Vibration**. Accepted for publication, 2008e.

VITALI, R.; HAFTKA, R. T.; and SANKAR, B.V. Multi-fidelity Design of Stiffened Composite Panel with a Crack. **Structural and Multidisciplinary Optimization**, Vol. 23, N. 5, pp. 347-356, Jun. 2002.

VR&D CONTRIBUTORS. **VisualDOC Design Optimization Software, Version 6.0, Getting Started Manual**. Colorado Springs, USA: Vanderplaats Research and Development, Inc., 2006.

WANG G. G. and SHAN, S. Review of Metamodeling Techniques in Support of Design Optimization. **Journal of Mechanical Design**. Vol. 129, pp.370-380, 2007.

WASSERMAN, P. D. **Advanced Methods in Neural Computing**. 1st edition. New York, USA: Van Nostrand Reinhold, May 1993.

YE, K. Q., LI, W. and SUDJANTO, A. Algorithmic Construction of Optimal Symmetric Latin Hypercube Designs. **Journal of Statistics Planning and Interference**. Vol. 90, pp. 145-159, 2000.

ZANINI, M. A. X.; MAGALHÃES, S. A.; GÓES, L. C. S.; VIANA, F. A. C.; and STEFFEN Jr, V. On the use of Nature-Inspired Optimization Techniques applied to the Model Updating of a Landing Gear. In: ICED2007 - INTERNATIONAL CONFERENCE ON ENGINEERING DYNAMICS, April 16-18, 2007Algarve, Portugal. **Proceedings of ICED2007**.

ZERPA, L.; QUEIPO, N. V.; PINTOS, S.; SALAGER, J. An Optimization Methodology of Alkaline-surfactant-polymer Flooding Processes using Field Scale Numerical Simulation and Multiple Surrogates. **Journal of Petroleum Science & Engineering**. Vol. 47, Issues 3-4, pp.197-208, Jun. 2005.

ZHOU, Z.; ONG, Y. S.; NAIR, P. B.; KEANE, A. J.; and LUM, K. Y. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. **IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications And Reviews**. Vol. 37, No. 1, Jan. 2007.

ZITZLER, E. and THIELE, L. Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach. **IEEE Transactions on Evolutionary Computation**. Vol. 3, Issue 4, pp.257-271, Nov. 1999.

VIANA, F. A. C., 2005, **Técnicas de Meta-Modelagem e Métodos Heurísticos de Otimização Aplicados a Problemas de Projeto e Identificação**. 2008. 144 páginas. Tese de Doutorado. Universidade Federal de Uberlândia, Uberlândia, Brazil.

RESUMO EXTENDIDO EM PORTUGUÊS

Avanços na capacidade de processamento computacional popularizaram a otimização numérica como uma ferramenta de engenharia. Contudo, eles aparentemente também favoreceram o aumento na complexidade das simulações. Como resultado, o custo computacional de simulações complexas de alta fidelidade em engenharia dificultam o uso exclusivo de simulações em otimização. Esta pesquisa de doutorado representa um esforço em combinar técnicas de otimização global e meta-modelagem como uma forma de usar racionalmente os recursos computacionais e aumentar o nível de informação obtida durante a tarefa de otimização. Aspectos positivos dos métodos heurísticos de otimização, tais como o potencial para otimização global, facilidade para programação, e robustez foram explorados. Estes métodos também foram combinados com modelos de fidelidade variável, que reduz o esforço computacional intrínseco aos algoritmos heurísticos de otimização. As técnicas mencionadas acima foram usadas na solução dos problemas contínuo-discretos do projeto ótimo de uma estrutura veicular e componentes estruturais aeronáuticos; identificação de derivadas de controle e estabilidade longitudinal de aviões e modelo não linear de trem de pouso; e melhoramento de meta-modelos via adição de simulações; bem como na solução do problema combinatorial do hipercubo latino ótimo. Para a solução de problemas de otimização contínuo-discreta Otimização por Colônia de Formigas (OCF), Evolução Diferencial (ED), Algoritmos Genéticos (AG), Otimização por Enxame de Partículas, e Modelo do Ciclo de Vida foram usados. Para a solução do problema de otimização combinatorial o Algoritmo Estocástico Evolucionário Melhorado (AEEM) foi usado. Como resultado, algumas contribuições teóricas foram introduzidas à versão básica dos algoritmos heurísticos estudados, incluindo: (i) uso simultâneo de diferentes instâncias de operadores para o ED e AG; (ii) aleatoriedade adicional para OCF e ED; (iii) operadores de aleatoriedade disparados por uma medida de dispersão; e (iv) implementação do hipercubo latino ótimo em um ambiente de desenvolvimento de software comercial. Ao final da pesquisa, a principal lição é que, assim como também acontece com algoritmos clássicos de otimização, o sucesso no uso de métodos heurísticos é altamente dependente do problema, nível de fidelidade das simulações, nível das informações já conhecidas, e, obviamente, recursos computacionais. Desta forma, o uso de fidelidade variável juntamente

com métodos heurísticos de otimização é uma estratégia bem sucedida, uma vez que métodos heurísticos não requerem informação sobre o gradiente (isto é, os recursos são diretamente usados na busca e não há propagação dos erros devido ao custo dos gradientes); e eles têm a tendência em encontrar a solução global ou próximo dela. Em alguns casos, uma combinação em cascata de métodos de otimização heurísticos e clássicos pode compor uma estratégia viável para aproveitar as capacidades de busca global e local dos algoritmos individuais. Ao acoplar simulações de alta fidelidade e otimização heurística para o melhoramento de meta-modelos, as estratégias propostas neste trabalho de doutorado indicam eficientemente novos pontos para simulações; e enquanto os ganhos sobre o espaço de projeto são limitados pelo número de variáveis e pela complexidade da função, os ganhos na região dos ótimos são menos sensíveis. Finalmente, uma vez que um grande número de pontos no espaço de projeto é naturalmente visitado no uso de algoritmos heurísticos, é natural que uma análise apropriada do conjunto de pontos seja uma fase complementar ao final da otimização.

Palavras-Chave: Métodos heurísticos de otimização, meta-modelagem, otimização de projeto, problemas inversos, modelagem por fidelidade variável.