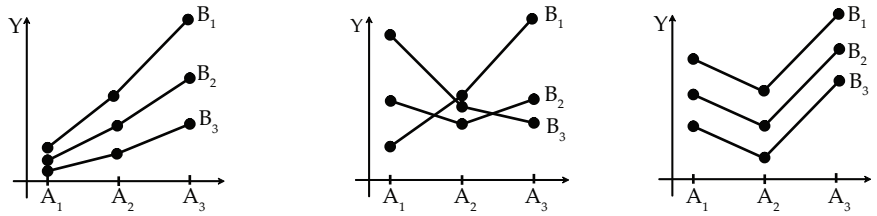


(a) Synergistic Interaction (b) Anti-synergistic Interaction (c) No interaction

Figure 3.28 Interaction graphs with 2 discrete variables at 3 levels. Adapted from (Phadke, 1989).

The interaction between two factors, *e.g.,* Factor A and Factor B, is computed using a matrix with dimensions equal to $a \times b$ which is filled with the following coefficients:

$$C_{A_i B_j} = \frac{Y_{ij\ldots}}{r} \tag{3.46}$$

In the simple example presented before, this matrix was represented by all 9 performances computed, because there were only 2 factors, but this is not valid in general of course. In this case $Y_{ij\ldots}$ indicates the sum of the $r = c \times \cdots \times m \times l$ responses with the Factor A at level i and Factor B at level j. For each level of A, for instance, b average performances can be plotted against the objectives values, providing the so-called interaction graphs, see Figure 3.28. When the lines of an interaction graph are not parallel it indicates the presence of synergistic (as in the previous example) or anti-synergistic effects, *i.e.,* interactions. A synergistic effect is present when the improvement of a performance given the variation of a factor is enhanced by the variation of another one. An anti-synergistic effect is the exact opposite (Phadke, 1989). In Figure 3.28, the higher-order behavior of the objective to the variation of the variable levels is indicated by the fact that the lines are not perfectly straight over the three levels of variable A, for instance.

The interactions between continuous and discrete variables, eventually detected by sensitivity analysis, can be graphically presented using a mix of contour plots, or single-variable trends, and linear graphs, as will be shown in the following subsection.

One last remark about the physical interpretation of linear graphs. There is a logical difference between the results obtained in case of *ordinal* and *categorical* discrete variables. In case of ordinal discrete variables, *e.g.,* the *number* of batteries in a satellite, the factor effect may indicate a certain increasing or decreasing trend of the performance given the variation of the factor. For instance, the mass of the power subsystem increases as the number of batteries increase. In case of categorical variables instead, *e.g.,* the *type* of batteries to be implemented, the effect identified with ANOVA and with the linear graphs may not be realistic anymore. The variation of the mass of the satellite, in this case, depends on the order in which the type of batteries are considered in the experimentation.

This aspect has an implication on the type of matrix design selected for sampling the subspace formed by the discrete variables only. In principle all the combinations of *categorical* design factors shall be experimented. Each one of these combinations represents a different system architecture that needs to be explicitly assessed. For the *ordinal* design factors instead, fractional-factorial designs may suffice to compute their effect on the output. However, this does not always have to be the case, thus accurate matrix-design selection has to be made by the engineering team depending on the type of problem at hand.

3.3.3 Test case: satellite system for Earth-observation, visualization of the design region

In Section 3.2.6 we analysed the model of the satellite system for Earth-observation using RBSA. The results demonstrated that there are few parameters, among these selected for the analysis, that influence the performance more than others. In this subsection we demonstrate that with no additional computational effort, we can elaborate the results in a graphical way to better support the engineering team in selecting a suitable baseline design. The purpose of this subsection is to use the results of RBSA to show the performance trends under the effect of the most influential factors.

In the interaction graph of Figure 3.29(a) the two discrete variables related to the orbit of the satellite are considered. For each level of A and B the average value of the equatorial coverage is plotted. The number of days for a repeating ground-track and the total number of orbits in that time period have a synergistic effect on the coverage. In particular, as expected with a higher orbit (*e.g.*, 13 orbits in 1 day and $H = 1258.6$ km) the average equatorial coverage is larger compared to a case with a lower orbit (*e.g.*, 29 orbits in 2 days and $H = 725.2$ km). The combinations of factors levels A1-B3 (*i.e.*, 15 orbits in 1 day), A2-B3 (*i.e.*, 30 orbits in 2 days), and A3-B3 (*i.e.*, 45 orbits in 3 days) lead to the same configuration since the altitude of the orbit is the same, $H = 567.5$ km.

In Figure 3.29(b) we present the *coverage* and the *resolution* performances as a function of the *minimum elevation angle* (factor D) and the *instrument aperture diameter* (factor C). The solid lines represent the mission configuration A3-B2, while the dashed lines represent the mission configuration A1-B1. The light-gray area represents the revisit time constraint for the A3-B2 configuration, set as 100% of equatorial coverage in 24 h. The dark-gray area represents the same constraint for the A1-B1 configuration. A higher orbit (dashed lines in Figure 3.29(b)) allows to meet the re-visit constraint with a larger minimum elevation angle thus also improving the resolution performance at the edge of the swath. For the A3-B2 configuration, with $\epsilon = 30^\circ$ and the instrument aperture diameter equal to 0.7 m the resolution at the edge of the swath is 12.7 m/pixel, and 1.26 m/pixel at subsatellite point. For the A1-B1 configuration, instead, the resolution at subsatellite point is slightly worse, *i.e.*, 2.2 m/pixel, but at the edge of the swath a resolution of 7 m/pixel can be obtained. Further, for an A1-B1 configuration, the fact that the minimum elevation angle can be up to 30° gives the satellite the possibility to actually observe over the entire geometrical swath width with the maximum possible slewing angle, *i.e.*, $(E) = 50^\circ$, and at a higher resolution than an A3-B2 configuration.

The aperture diameter of the instrument, paradoxically, plays a more relevant role in the determination of the data rate, thus on the down-link margin than on the actual resolution, as demonstrated by the sensitivity analysis. Indeed, in Figure 3.29(d) the down-link margin constraint is plotted as a function of the instrument aperture diameter and the minimum elevation angle, for the configuration A1-B1 and with $(H) = 30$ W and $(I) = 1$ m. An A3-B2 configuration would push the coverage constraint down, with the result of allowing less flexibility in selecting the instrument aperture diameter. The effect on the cost is plotted in Figure 3.29(c). The assumption is that a higher orbit would require less maneuvers for pointing the instrument of the satellite in one particular direction and the effect is in a reduced cost (difference between the solid and the dashed lines). The constraint on the launcher-mass availability is mainly driven by the instrument aperture diameter. Indeed the mass and power consumption of the payload is scaled with the diameter, and so does the mass of the satellite and its cost. The *Delta II* class of launchers allows for enough flexibility until the payload aperture diameter of about 0.9 m.

The triangles in Figure 3.29 represent a tentative selection of the baseline. In particular, an A1-B1 architecture has been selected, with $(C) = 0.7$ m, $(D) = 30^\circ$, $(E) = 50^\circ$, $(F) = 120$ s, $(G) = 10000$, $(H) = 30$ W, $(I) = 1$ m, $(J) = 2$, $(K) = 2$, $(L) = 1$. With these settings of

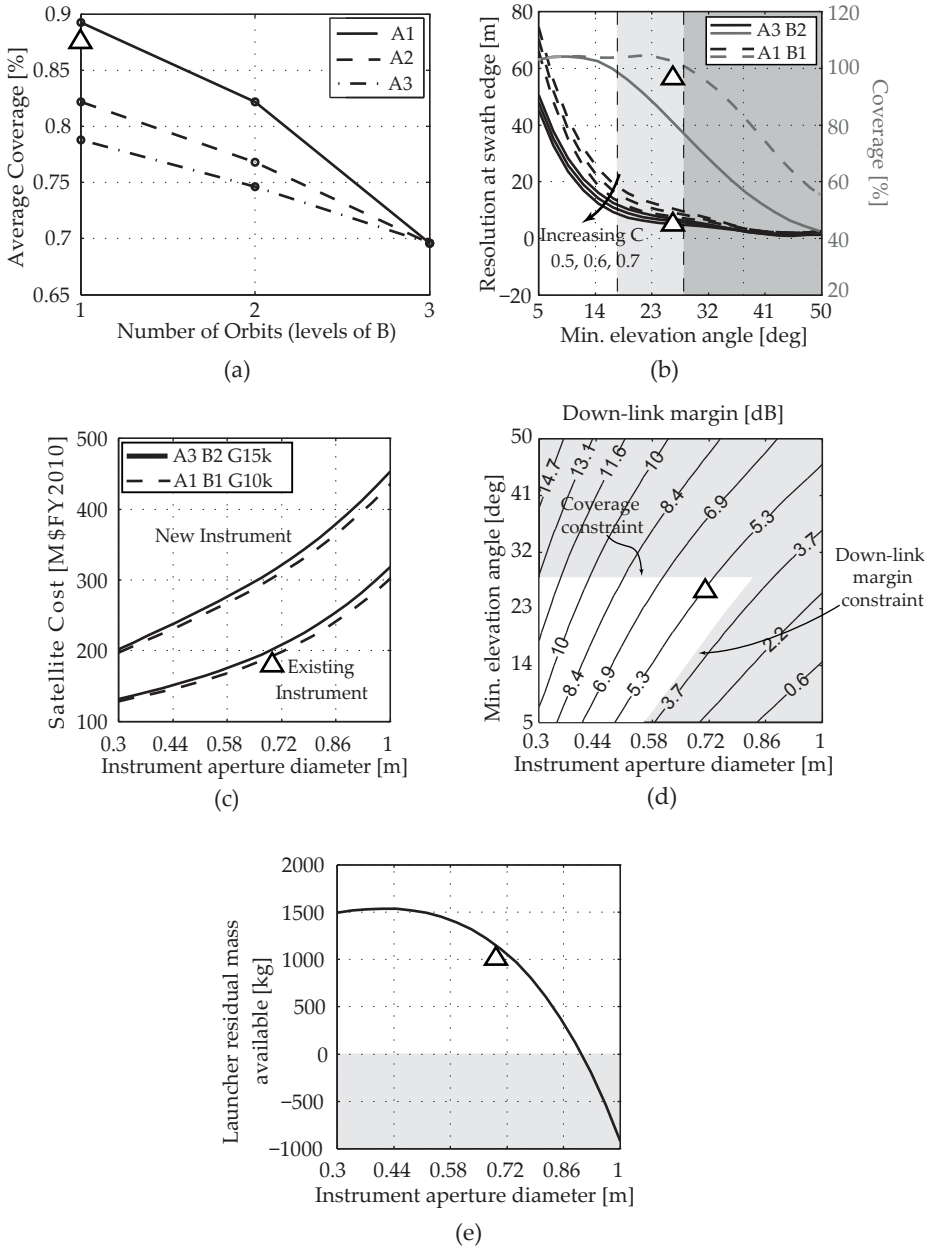


Figure 3.29 Analysis main results. Δ is a tentatively selected baseline. The light-gray area of (b) represents the revisit time constraint for the A3-B2 configuration, set as 100% of equatorial coverage in 24 h. The dark-gray area of (b) represents the same constraint for the A1-B1 configuration.

the design variables a confirmation experiment was performed on the model. The simulation yields to a cost of the satellite of 188 $M\$(FY2010)$, a mass of 1330 kg and an overall power consumption of 1 kW. The resolution at the edge of the swath is 7.3 m/pixel and 2.2 m/pixel at sub-satellite point. The equatorial coverage after 24 h is 100% and the down-link margin is 4.1 dB. In principle, selecting a design point from the graph obtained using regression analysis does not necessarily provide a precise result: a regression error may be present. For this reason, we advice to always use a confirmation experiment once a certain baseline is selected. In this case the results from the verification experiment are very close to the values that can be read from the graphs in Figure 3.29. This indicates that the sampling technique and the regression analysis provided reliable results. Sensitivity analysis and graphical support in the

form of contour plots, variable trends and interaction graphs enabled a thorough reasoning on the phenomena involved. This allows us to quickly select a system baseline that meets the constraints balancing the objectives under analysis.

With the information on the behavior of the system, made available thanks to the sampling of the design space, and given the availability of surrogate models obtained with regression analysis, it would be possible to run an optimization process to find the best possible baseline option. At this stage, and with local design settings, optimization would not be the most efficient way of selecting a baseline. Computational effort would be spent on a limited portion of the design space. This would be beneficial only if the designer would be confident that the best possible solution is within the selected design region. The current baseline solution is selected on the basis of the graphs that have been shown in this section.

3.4 Uncertainty analysis and robust design

The sampling techniques and analysis and visualization methods presented so far demonstrated to be helpful in determining the settings of the design variables that provide the performance of the system as required and that allow the system to meet the constraints. The variables considered in the previous sections are all deterministic in nature. This means that they are controllable factors whose level can be selected by the designer and can be obtained during the manufacturing and/or operations of the system. As mentioned in the introduction of this thesis, very often during conceptual design it can be the case that some factors are only known in a probabilistic sense: they are uncertain. The purpose of the design is to obtain combinations of controllable design-factor levels that provide *good* performance also in the presence of these uncertain factors. Uncertainty analysis and robust design are often considered complementary design activities implemented for determining the performance of the system under uncertain operating conditions. In particular, uncertainty analysis is the study of the uncertain distribution characteristics of the model output under the influence of the uncertainty distributions of the model inputs. With these settings, the purpose of uncertainty analysis is to *simply* propagate the uncertainty through the model. When the analysis presents both controllable and uncontrollable factors, the latter being intrinsically uncertain parameters (*e.g.,* operating environmental conditions), the purpose of the uncertainty analysis is to obtain settings of the controllable design variables that optimize the performances while at the same time minimize the impact of the uncertainties on the system. In this case we talk about robust design.

In general, uncertainty can be classified in two types: stochastic and epistemic. The stochastic, or aleatory, uncertainty describes the inherent variability associated with a certain phenomenon. It is usually modeled by stochastic processes when there is enough information to determine the probability distributions of the variables. For instance, the life-time of a certain component of the system is provided with a certain probability distribution (*i.e.,* failure rate) by the manufacturer. This failure rate is determined on a statistical basis by testing *many* components. In these cases one has detailed information about the uncertainty related to the life-time of that component. The epistemic uncertainty is characterized, instead, by the lack of knowledge about a specific characteristic of the system. In these cases usually uniform distribution is used to describe the uncertainty, but this procedure has been largely criticized. The main reason is that a phenomenon for which there is lack of knowledge cannot be represented by any specific probability distribution (Helton *et al.*, 2006).

For the design of a complex system, in case of both epistemic and stochastic uncertainty, probability theory alone is considered to be insufficient for a complete representation of the implications of the uncertainties on the performances. Therefore, in the following subsections we

introduce sampling methods and analysis techniques for propagating the uncertainty through the model, in the presence of both stochastic and epistemic uncertain factors.

3.4.1 The unified sampling method

In this subsection we introduce a modified implementation of the Sobol' sampling technique. A Sobol' sequence only allows to uniformly sample in the design space. Uniform distributions are the only necessary distributions to use in the presence of deterministic design variables, as discussed in the previous sections. The unified sampling technique, instead, allows to cope with any type of epistemic and stochastic distributions of the uncertain factors, typical when the focus of the analysis is that of propagating the uncertainty throughout the model.

The problem of determining the probability distribution of the output, given the probability distributions of the inputs of a model, is related to the computation of a multi-dimensional integral, similar in the form to the expression of Eq. (3.7). A direct numerical integration or the analytical solution of the integral can become practically infeasible with already few uncertain variables. Therefore, the direct Monte-Carlo simulation is amongst the most widely adopted methods for uncertainty analysis, since it does not require any type of *manipulation* of the model. When it comes to long-running (*i.e.,* computationally expensive) models, as is usually the case for complex space systems in a collaborative environment, the method of Monte Carlo, using random-sampling techniques, has the recognized disadvantage of being computationally expensive, since it generally requires a large number of simulations to compute the mean, the variance and a precise distribution of the response (Rubinstein, 1981). Helton and Davis (2003) compare Latin Hypercube Sampling with a random sampling technique for the propagation of uncertainty into mathematical models. Their analysis corroborates the original results obtained by McKay *et al.* (1979), and demonstrates that stratified sampling (*i.e.,* LHS) provides more stable Cumulative Distribution Functions (CDFs) of the output than random sampling, with the result that less samples are required for a given accuracy in the determination of the CDFs.

As discussed previously, also epistemic uncertainty must be considered for the design of a complex system. Thus, for the development of the unified sampling technique presented in this section we inherit some ideas and some nomenclature from the *evidence theory* derived from the initial work of Dempster (1967, 1968) and Shafer (1976). When lack of knowledge about a certain system behavior is present, and when the available historical and statistical sources are sparse, the engineering team is forced to evaluate and combine different data sources not perfectly tailored to the purpose at hand based on judgmental elements. Structured expert judgment is increasingly accepted as scientific input in quantitative models, and it is dealt with in a number of publications, see, for instance Cooke (1991) and O'Hagan and Oakley (2004). The result of the combination of expert judgments on the uncertainty of a specific phenomenon leads to the creation, for every single uncertain factor, of so-called *Basic Probability Assignments* (BPAs). The BPAs represent the level of confidence that the engineering team has in the fact that the value of the factor of interest lies in a certain interval of possible values. The uncertainty interval is divided into n subsets and for each of them a certain belief, or probability, that the actual value of the uncertain parameter will lie within that subset is assigned. The set of the n beliefs form the BPA for the factor under analysis. Consider for instance the epistemic uncertain factor A in Figure 3.30(a). The uncertainty interval of factor A (given on the x-axis) is equal to $[0, 1]$, divided into 3 subsets $[0, 0.2] \cup [0.2, 0.5] \cup [0.5, 1]$. Suppose that the judgment of the engineering team-members on the uncertainty structure of factor A leads to the conclusion that the actual value of A will lie in the subset $[0, 0.2]$ with a probability equal to 0.4, in the subset $[0.2, 0.5]$ with a probability equal to 0.3 and in the subset $[0.5, 1]$ with a probability of 0.3. Thus the BPA of factor A is equal to $[0.4, 0.3, 0.3]$ and its cumulative function is given on the y-axis of Figure 3.30(a).

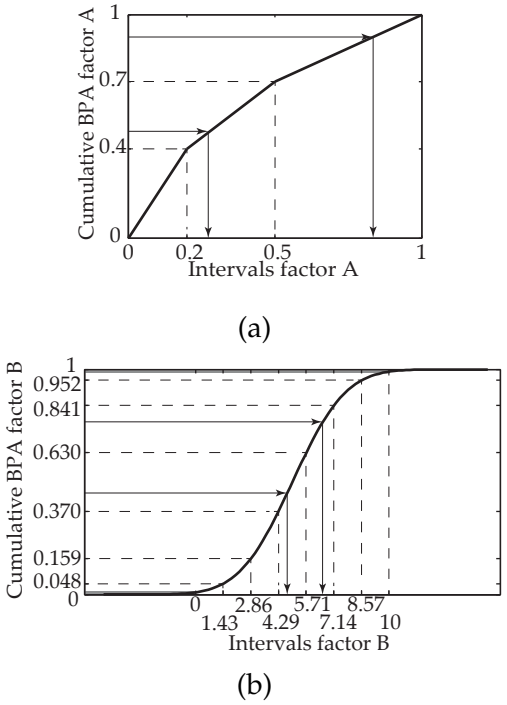


Figure 3.30 Representation of the cumulative distributions of (a) the epistemic uncertain variable, and (b) the stochastic (normal) uncertain variable. The dashed lines connect the BPAs to the relative uncertainty intervals. The arrows represent the projection of the sample points from the BPSs domain to the uncertainty-intervals domain.

To be able to do sampling in the presence of epistemic and stochastic variables at the same time, we shall unify the sampling procedure. The idea is to extend the concept of the BPA also to the stochastic variables in such a way to obtain a unique representation of the uncertainty structure of the inputs. For a stochastic variable the cumulative distribution function is continuous. The procedure we implemented foresees the discretization of the cumulative distribution of the stochastic factors. If the uncertainty interval of the stochastic factor is discretized into m subsets, then the discretized CDF can be expressed in the form of BPAs as in the case of the epistemic uncertain factors. Consider, for instance, the normally distributed uncertain factor B of Figure 3.30(b). Its uncertainty interval is equal to $[0, 10]$, divided into 7 subsets, for instance, as shows on the x-axis of Figure 3.30(b). The subsets are $[0, 1.43] \cup [1.43, 2.86] \cup [2.86, 4.29] \cup [4.29, 5.71] \cup [5.71, 7.14] \cup [7.14, 8.57] \cup [8.57, 10]$. Using the inverse CDF of the normal distribution we map these intervals to the associated BPAs: $[0.0480, 0.1110, 0.2106, 0.2608, 0.2106, 0.1110, 0.0480]$. The cumulative BPAs are given on the y-axis of Figure 3.30(b).

In the case of stochastic uncertainty, there is the possibility of having infinite tails of the distributions, as in the case of the normal one. To be able to do sampling between a minimum and a maximum value of the design factor, we shall truncate the tails of the distribution. Consider, for instance, a normal distribution. If the tails are truncated at 3σ, for instance, 0.27 $\%$ of the values expected for distribution are neglected. Therefore an error is introduced in the sampling procedure. However, if the minimum and the maximum values of the uncertainty intervals represent a high percentile, *e.g.*, 0.95 and 0.05, or 0.99 and 0.01 (as in the case of factor A), the error is acceptably small in most of the cases. In Figure 3.30(b) the gray areas represent the error that arises when considering a truncated normal distribution. The probabilities of the first and the last intervals are overestimated by a quantity equal to the smallest truncation percentile (0.01 in this case).

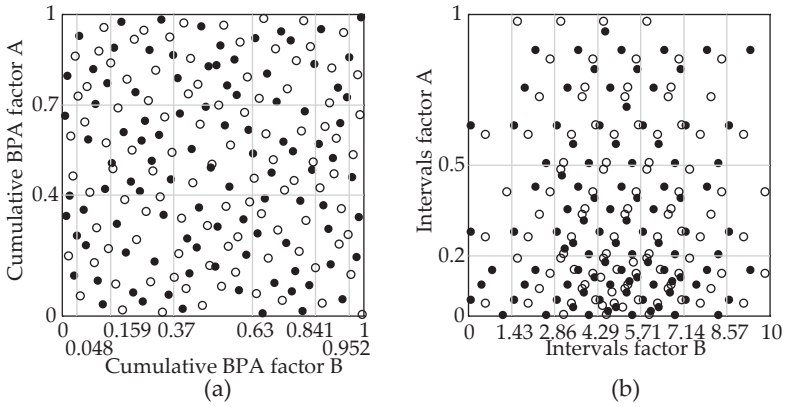


Figure 3.31 Unified sampling method. Representation of (a) the uniform sampling in the BPAs domain, and (b) the corresponding sample points in the uncertainty-intervals domain.

The unified sampling method, in the presence of both stochastic and epistemic factors, is executed in two steps. First, a uniform sampling on the space formed by the cumulative values of the BPAs is executed, Figure 3.31(a). In Figure 3.31(a) the x-axis and the y-axis represent the cumulative BPA structure of Factor A and B as given on the x-axis of Figure 3.30(b) and (a) respectively. Then, each sample point of Figure 3.31(a) is mapped to the corresponding point in the uncertainty-interval domain, Figure 3.31(b). This passage from the BPA domain to the uncertainty-intervals domain is also represented by the arrows in Figure 3.30. The CDF of each factor is used for this transformation. Adopting this 2-steps procedure, the final sample (*i.e.*, Figure 3.31(b)) is collected according to the mixed aleatory/epistemic probability distribution of the factors.

Experience and common sense tell that the more BPA intervals, the better the approximation of the output probability-distribution. However, in the case of epistemic-factor uncertainty the number of BPA intervals depends on the degree of knowledge of the engineering team on the behavior of the factors themselves. If the initial uniform sampling is performed according to a stratified technique (*e.g.*, LHS), the resulting response CDF will be more stable than what could be obtained by using a random technique, as demonstrated by Helton and Davis (2003) and McKay *et al.* (1979). Further, if a Sobol' sequence is implemented, all the advantages already discussed in the previous chapters would still hold. This is particularly true if seen from the perspective of computing the sensitivity analysis using the RBSA, which is directly applicable if the unified sampling method is used. The computation of sensitivity analysis under uncertainty settings allows to identify the contribution of the inputs to the uncertainty in the analysis output, so to drive the effort in better describing the uncertainty of only the most relevant factors.

Verification of the unified sampling method

The unified sampling method has been verified with the test functions provided by Helton and Davis (2003):

$$f_1(U,V) = U + V + UV + U^2 + V^2 + U \cdot \min\left(\exp\left(3V\right), 10\right) \\ U \in [1.0, 1.5];\ V \in [0,1]; \tag{3.47}$$

$$f_2(U,V) = U + V + UV + U^2 + V^2 + U \cdot g(V) \\ U \in [1.0, 1.5];\ V \in [0,1]; \tag{3.48}$$

with

$$h(V) = (V - 11/43)^{-1} + (V - 22/43)^{-1} + (V - 33/43)^{-1}$$
$$g(V) = h(V) \;\; if \;\; |h(V)| < 10$$
$$g(V) = 10 \;\; if \;\; |h(V)| \geq 10$$
$$g(V) = -10 \;\; if \;\; |h(V)| \leq -10$$

The expression in Eq. (3.47) is monotonic for positive values of U and V, while the expression in Eq. (3.48) is monotonic for positive values of U and non-monotonic for positive values of V. In the following figures, from Figure 3.32 to 3.35, we compare the performance obtained in estimating the output uncertainty of the models in Eqs. (3.47) and (3.48) when using the unified sampling method, random sampling, and Latin hypercube sampling. In each one of these figures we show six graphs. On the left-hand side of the figures the comparison between LHS and unified sampling is shown. On the right-hand side we present the results of the comparison between the performance obtained with random sampling and unified sampling. In each sub-figure we calculate the estimated output CDF of the two equations, using 30 replicated samples (*i.e.,* 30 replicated set of simulations) of size 25, 50, and 100, going from top to bottom.

In Figure 3.32 the arguments U and V are assumed uncorrelated and uniformly distributed over the intervals. Already with 25 sample points the estimated CDF computed using the results obtained from the unified sampling technique, the black lines, is very close to the one computed with 100 sample points. This is not the case for LHS and random sampling. The CDFs computed using the unified sampling method are always within and in the center of the estimates obtained with the 30 replicates of the other two sampling methods. In particular, one can observe that for an increasing sample size the CDF estimates obtained with LHS and Random Sampling tend toward the estimate obtained using the unified sampling method. This means that it is possible to estimate a precise output cumulative distribution function with much less sample points if the unified sampling technique is used, instead of LHS or random sampling.

The results in Figure 3.33 are also obtained assuming arguments U and V to be uncorrelated and uniformly distributed over the intervals. In Figures 3.34 and 3.35, instead, the arguments U and V are assumed normally distributed, with the intervals representing the 0.01 and 0.99 percentiles.

In general the same conclusions can be drawn also for the other figures of the verification presented in this section. The main purpose of uncertainty propagation is establishing the uncertain distribution of the output of a model, given the uncertainty distribution of its input factors. The unified sampling methods demonstrates that this can be done with a reduced computational effort is compared to commonly used LHS or random sampling techniques.

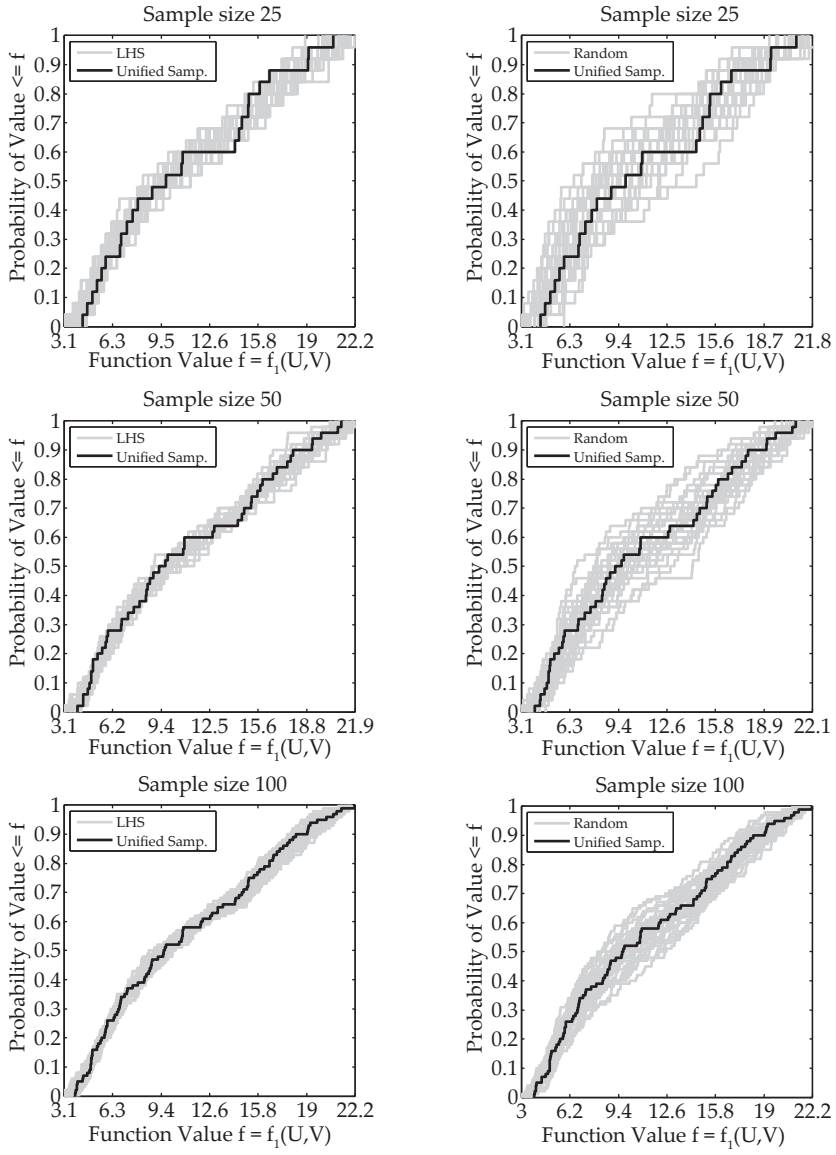


Figure 3.32 Comparison of estimated CDFs for Eq. 3.47 with uniform distribution of the parameters obtained with 30 replicated samples of size 25,50, and 100 using Latin Hypercube, Random, and unified sampling.

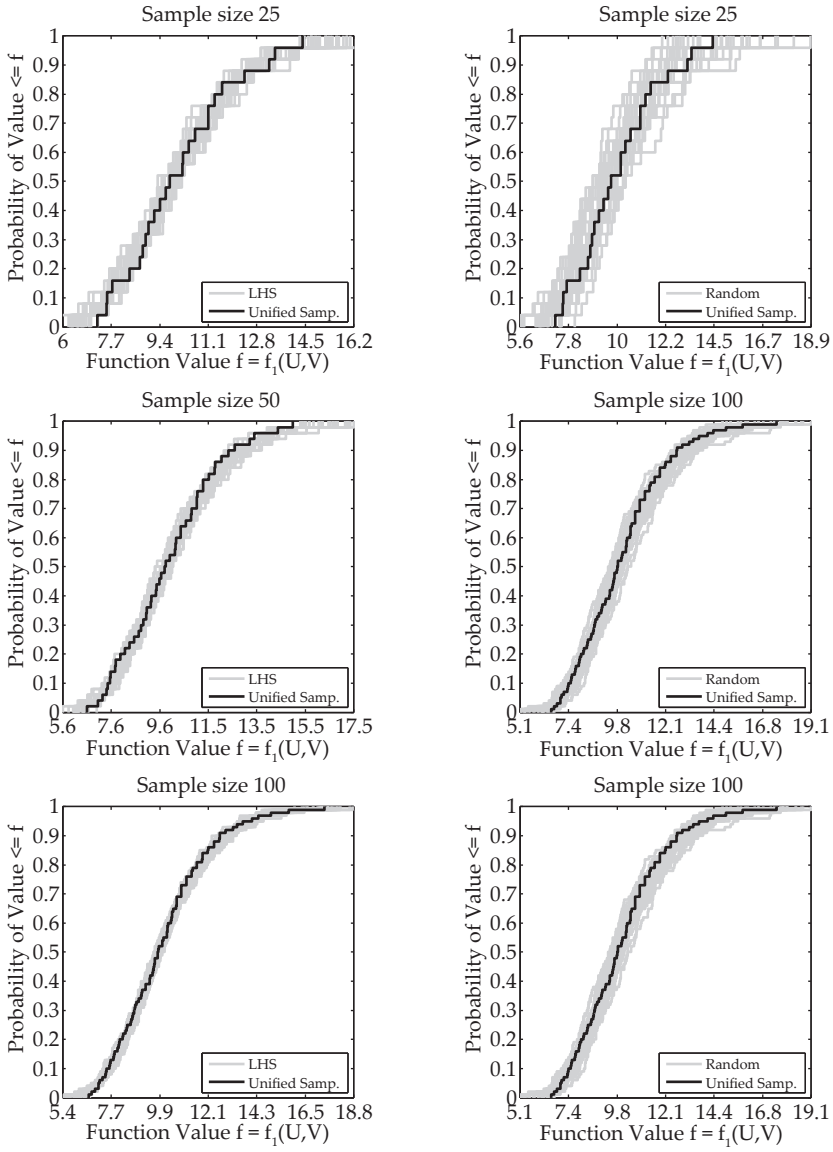


Figure 3.33 Comparison of estimated CDFs for Eq. 3.47 with normal distribution of the parameters obtained with 30 replicated samples of size 25,50, and 100 using Latin Hypercube, Random, and unified sampling.

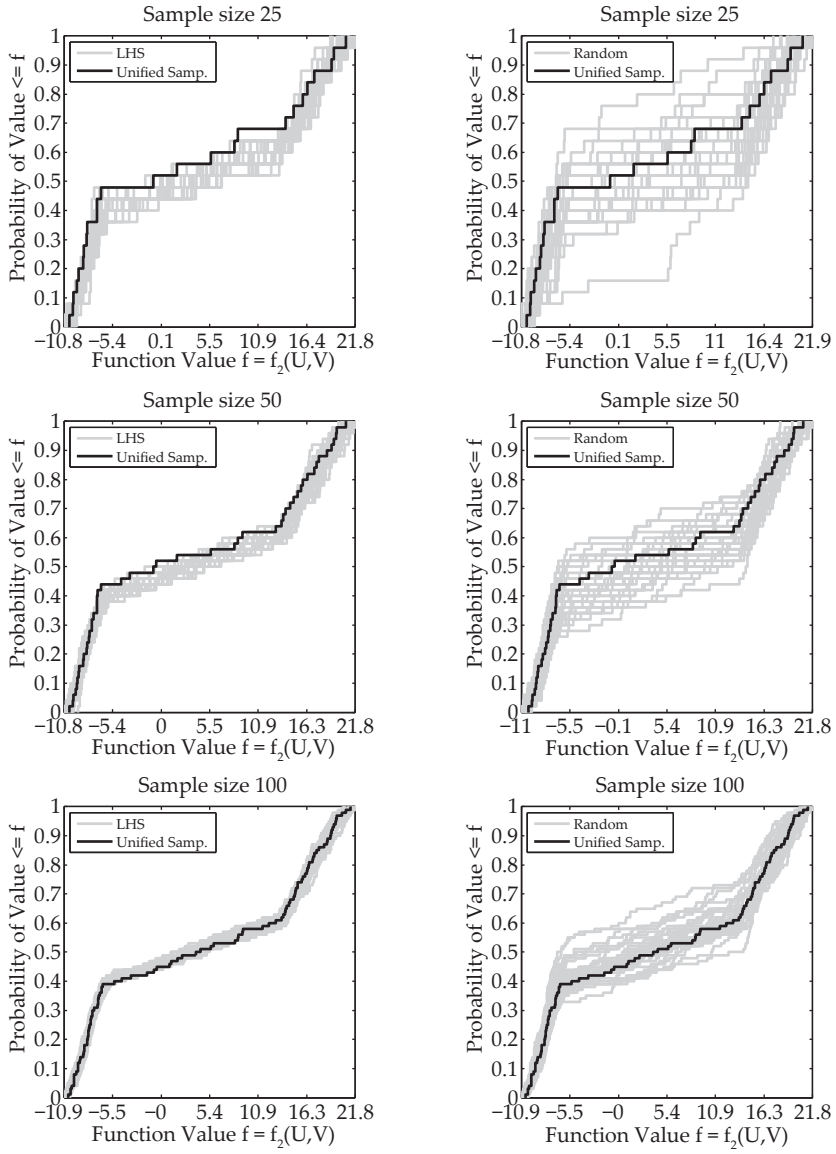


Figure 3.34 Comparison of estimated CDFs for Eq. 3.48 with uniform distribution of the parameters obtained with 30 replicated samples of size 25,50, and 100 using Latin Hypercube, Random, and unified sampling.

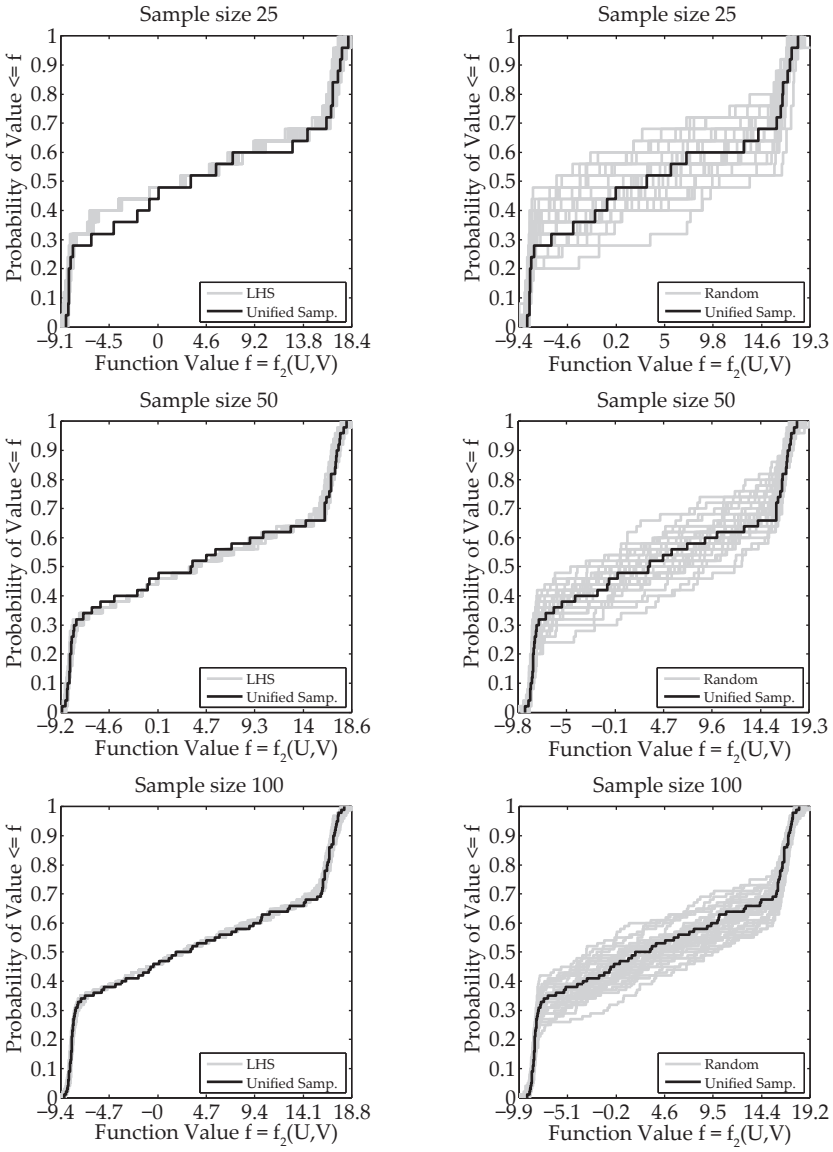


Figure 3.35 Comparison of estimated CDFs for Eq. 3.48 with normal distribution of the parameters obtained with 30 replicated samples of size 25,50, and 100 using Latin Hypercube, Random, and unified sampling.

3.4.2 Test case: satellite system for Earth-observation, uncertainty analysis

In the traditional systems engineering process, design margins are used to account for technical budget uncertainties, *e.g.,* typically for cost, mass and power. A certain percentage of the baseline's performance is added to account for both uncertainties in the model and uncertainties about eventual assumptions made at a preliminary phase that will likely be modified in advanced phases, due to an increased level of detail and knowledge. For instance, the results presented in section 3.3.3 were obtained with a 15% margin on the total satellite mass, total power consumption and propellant stored on board. The results without margins would be different. In particular, the satellite mass would be equal to 1048 kg, the power consumption equal to 830 W and with a cost saving of 15 $M\$(FY2010)$. The unified sampling method allows the engineering team to obtain more insight in the uncertainty structure of the solution by focussing on every single source of uncertainty. This will enable a more informed decision-making process on the allocation of the budgets to each subsystem and each element.

In the case of the Earth-observation mission we considered the uncertain parameters and the uncertainty structure presented in Table 3.19. A mix of normal, log-normal and epistemic distributions has been considered. The normal and the log-normal uncertain variables are centered around the values needed to obtain the results presented before. The epistemic uncertain intervals and BPAs are determined in such a way that the value of the factors needed to obtain the previous results is at the center of the first epistemic interval. Using the unified sampling method, with 200 sample points (*i.e.,* 200 model evaluations) we obtained the results shown in Figure 3.36. In Figure 3.36(a,b,c) the probability density estimates of the satellite cost, mass, and power consumption respectively, are presented. The histograms are plotted with an adjusted scale, so to obtain a total area of the bars equal to 1. The probability density function estimates are obtained using Maximum Likelihood Estimation (MLE). It is a standard approach in statistics to do parameter estimation and inference. In this case we use MLE to estimate the parameters of a probability density function that best fits the data obtained from the 200 simulations.

In Figure 3.36 the black and gray arrows are positioned in correspondence to the values of the performance computed for the analysis in Section 3.3.3 with and without margins, respectively.

The *margins approach* is largely used in conceptual design for several reasons. It provides

Uncertain Variables		Intervals		Distribution
		Min	Max	
Margin δV	[%]	0	0.25	Epistemic[a]
Specific Impulse	[s]	280	320	Normal[d]
Thrusters inert mass fraction	[%]	0.2	0.4	Epistemic[b]
ADCS sens. mass	[kg]	58	70	Log-normal[e]
ADCS sens. power	[W]	33	45	Log-normal[e]
Antenna mass density	[kg/m^2]	9	11.5	Normal[d]
Solar cells η	[%]	0.17	0.23	Normal[d]
Solar array power dens.	[W/kg]	90	110	Normal[d]
Batteries energy dens.	[W-h/kg]	25	75	Normal[d]
PCU mass	[kg]	27	50	Log-normal[e]
Regulators mass	[kg]	33	55	Log-normal[e]
Thermal subs. mass	[kg]	20	50	Log-normal[e]
Struct. mass margin	[%]	0	1	Epistemic[c]

Table 3.19 Settings of the design variables.[a]Intervals $[0, 0.04, 0.1, 0.17, 0.25]$, BPA $[0.4, 0.3, 0.2, 0.1]$. [b]Intervals $[0.2, 0.25, 0.3, 0.4]$, BPA $[0.4, 0.35, 0.25]$. [c]Intervals $[0, 0.25, 0.5, 0.75, 1]$, BPA $[0.4, 0.3, 0.2, 0.1]$.[d]$\mu = 0$ $\sigma = 1$, Min and Max are the 0.01 and 0.99 percentile respectively.[e]$\sigma = 1$, Max is the 0.99 percentile, Min corresponds to $X = 0$.

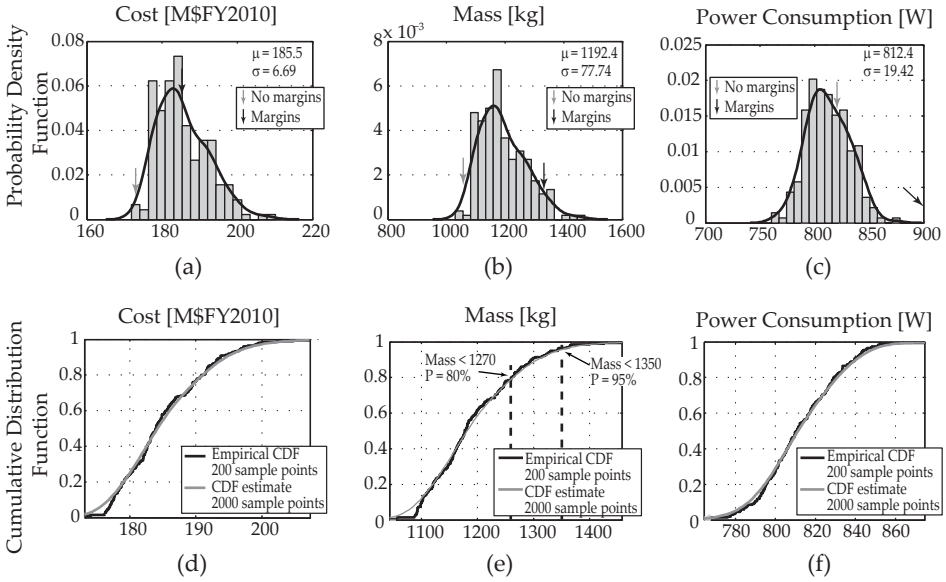


Figure 3.36 Uncertainty-analysis results of the Satellite system for Earth observation test case.

a means to provide precautionary estimates on the performance due to unknowns at the moment of designing and due to performance changes (*e.g.*, mass increase) during manufacturing of the system. Most of the times these margins are based on past experience of design and production of systems of similar nature. Having margins on the performance of the system means that the designers provide an overall estimation of the *worst case scenario* without relating it to the actual probability of the performance reaching that specific level. The margins approach does not give the same insight provided by the PDFs and the histograms on the performances of the system under uncertain input factors. When the uncertainty of the input factors can be estimated (with stochastic or epistemic distributions) with the PDF trends as shown, for instance, in Figure 3.36 allows the engineering team to better understand the behavior of the system under the effect of these uncertainties. Propagating the uncertainty into the model brings two main advantages. First, the uncertainty can be allocated to single subsystems and single elements more effectively. Second, the final performance can be precisely assessed according to the desired confidence level. Further, having a precise distribution of the performances allows for more effective budget-allocation management for subsequent phases of the design process. In Figure 3.36(d,e,f) the empirical cumulative distribution functions of the performances are presented. The CDF estimate, computed with 2000 sample points using a random sampling method, is also represented. In this figure we read, for instance, that given the uncertainties in Table 3.19 the mass of the satellite will not exceed 1270 kg with 80% probability, and that it will not exceed 1350 kg with 95% probability. The fact that the empirical CDF (computed with the 200 sample points from the unified sampling method) and the CDF estimate are very close to each other demonstrates that the unified sampling method is able to provide accurate results with a limited computational effort, also in the presence of uncertain factors.

Uncertainty is an ingredient of conceptual design. Design margins are a typical example of how uncertainty for unknowns at the design stage are taken into account. In this section we have demonstrated that when uncertainties can be determined with a stochastic distribution, the resulting PDFs and CDFs are much more informative to the engineering team. Further, also when uncertainties are *not known*, epistemic estimated distributions can better capture the knowledge of the engineering team. This concept is further analysed in Chapter 6, where we

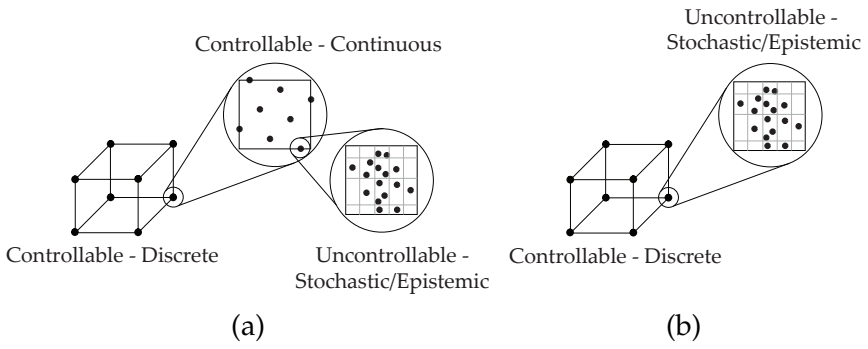


Figure 3.37 Augmented Mixed Hypercube sampling procedure for robust design.

use uncertainty analysis for the mass-budget management of a cubesat mission designed in the ESA Concurrent Design Facility.

3.4.3 Robust design and the Augmented Mixed Hypercube approach

Robustness is a concept that can be seen from two different perspectives, at least according to the discussion so far. One can define robustness of the system with respect to the effect of uncontrollable factors (aleatory and/or epistemic) and, if interested in obtaining a robust design, one can select that combination of controllable design-factor values that minimizes the variance while optimizing the performance. This concept was already expressed in the previous section, and it is the most common way of thinking of robust design. However, robustness can also be defined as the insensitivity of a certain design baseline to modification of the design variables in subsequent phases of the design process, thus providing an intrinsic design-baseline robustness figure. The modification of the levels of the design variables is likely to happen, especially when the baseline is at an early stage of the design process (phase 0/A). In this sense, robustness can be linked to the programmatic risk encountered when modifying a set of design parameters at later stages of the design process (**?**). In the first case, instead, robustness is more related to the operational-life risk of the system (if the uncertainties derive from the operational environment, for instance).

In this section we introduce the Augmented Mixed Hypercube (AMH) as a mixed sampling techniques that takes into account continuous and discrete variables, where continuous variables can be deterministic (*i.e.*, controllable) or probabilistic (*i.e.*, uncontrollable). Discrete design factors are always considered deterministic in this thesis. For system design, discrete variables describe *architectures* of the system. Systems architectures are fully controllable during design.

The AMH is presented in Figure 3.37 as an extension of the mixed hypercube shown at the beginning of this chapter, in Figure 3.6. In the AMH we take into account all types of design factors mentioned in this chapter. When the purpose of the analysis is to study the settings of controllable factors that are able to cope with the uncertainties introduced by the uncontrollable factors (stochastic and epistemic) then the AMH of Figure 3.37(a) shall be used. There, for each combination of the levels of the controllable design variables, an uncertainty analysis can be executed using the unified sampling method to obtain the performance of the system, and the relative statistics, due to uncertain factors. When the purpose is only to propagate uncertainty into the model, then the AMH in the form presented in Figure 3.37(b) shall be used, instead. The AMH in Figure 3.37(b) was used for the analysis presented in Section 3.4.2.

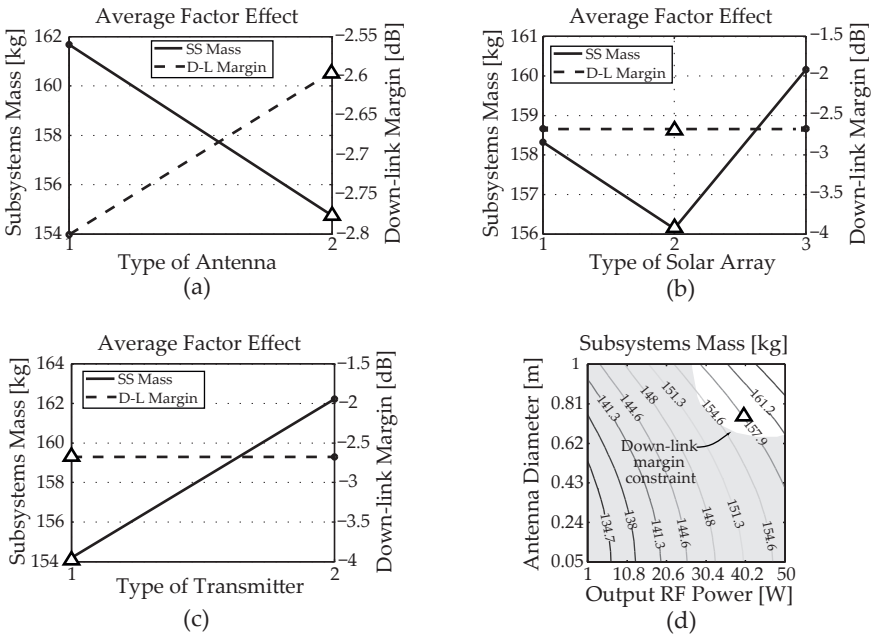


Figure 3.38 Main results of the Communication and Power subsystems analysis. Δ is a tentative selected baseline. The light-gray area of (d) represents the down-link margin constraint-violation conditions.

In the following subsection we present the utilization of the AMH for robust design, applied to the *communication and power subsystems* mathematical model.

3.4.4 Test case: the communication and power subsystems, robust design

In this subsection the robust design of the communication and power subsystems, using the Augmented Mixed Hypercube approach, is discussed. The results obtained with the RBSA in Section 3.2.4 suggest that the linear graphs and contour plots that retain most of the variability of the performances are those presented in Figure 3.38. As shown in Figure 3.38, the trends corroborates the initial insight in the problem gained with the sensitivity analysis.

With these settings of the design variables, a confirmation experiment was performed on the model. The simulation provided a mass of the coupled subsystems of 160.2 kg and a down-link margin of 4.96 dB. The reason for performing a confirmation experiment is that the design point selected from the contour plot may not be very precise eventually due to the presence of lack-of-fit in the regression model. To get the results without the bias caused by the lack-of-fit, a confirmation experiment is needed.

The purpose of the analysis presented in this subsection is to make some conclusions on the robustness to controllable and uncontrollable factors variations of the various architectures, using the AMH approach. A tabular representation of the AMH used for the analysis is presented in Table 3.20. The two continuous design variables are considered with a certain degree of uncertainty with respect to their baseline value. The other uncontrollable factors in Table 3.20 encompass many aspects related to the design and the operative life of the satellite for which there is uncertainty on one side, and the impossibility of controlling them directly on the other side. The results of the robust design on the Communication and Power subsystems, presented in Figure 3.39, are computed using the AMH sampling procedure as shown in Figure 3.37(b).

In Figure 3.39(a,b) the most robust and least robust configurations of the architectural variables are presented. In this case, the optimal configuration selected as a tentative baseline is

Uncertain Variables		Code	Intervals		Distribution
			Min	Max	
Output RF power	[W]	A	35	45	Uniform
Antenna diameter	[m]	B	0.75	0.85	Uniform
Satellite pointing error	[deg]	C	1	4	Normal[d]
Implementation loss	[dB]	D	1	4	Epistemic[a]
Satellite antenna efficiency	[-]	E	0.45	0.55	Normal[d]
Antenna mass density	[Kg/m^2]	F	9	11.5	Log-Normal[e]
Ground antenna efficiency	[-]	G	0.45	0.55	Normal[d]
Ground antenna pointing error	[deg]	H	0.1	1	Log-Normal[e]
Transmission efficiency - Sunlight	[-]	I	0.6	0.8	Epistemic[b]
Transmission efficiency - Eclipse	[-]	J	0.6	0.8	Epistemic[c]
Solar cells η	[%]	K	Nominal[f] - 10%	Nominal[f] + 10%	Log-Normal[e]
Solar array power dens.	[W/kg]	L	Nominal[f] - 10%	Nominal[f] + 10%	Log-Normal[e]
Batteries energy dens.	[W-h/kg]	M	25	75	Log-Normal[e]
Circular orbit altitude	[km]	N	990	1100	Normal[d]
Type of Antenna	[-]		1	2	2 levels
Type of Solar Array	[-]		1	3	3 levels
Type of Transmitter	[-]		1	2	2 levels

Table 3.20 Settings of the design variables. [a]Intervals $[1, 1.75, 2.5, 3.25, 4]$, BPA $[0.4, 0.25, 0.2, 0.15]$. [b]Intervals $[0.6, 0.667, 0.773, 0.8]$, BPA $[0.25, 0.4, 0.35]$. [c]Intervals $[0.6, 0.667, 0.773, 0.8]$, BPA $[0.25, 0.4, 0.35]$. [d]$\mu = 0$ $\sigma = 1$, Min and Max are the 0.01 and 0.99 percentile respectively. [e]$\sigma = 1$, Max is the 0.99 percentile, Min corresponds to $X = 0$. [f] See nominal values in Table A.3.

also the most robust one (see the black PDF). The least robust configuration, the one with the largest variance, is instead represented by the one having the *horn* antenna, the *triple junction* type of solar cell, and the *SSPA* type of transmitter. The sensitivity analysis presented in Figure 3.39(c,d) reports the uncertain-factors contribution to these results. The *transmitter output-power* and the *transmission efficiencies* are the factors that influence most the sensitivity of the subsystem mass to the uncertainties (design and environmental). This means that the transmitter output power shall be carefully controlled in subsequent phases of the design process to maintain the *as-designed* performances. This also means that the margin that shall be applied to the subsystem mass is strongly dependent on the uncertainties that the engineering team has on the efficiencies with which the power is transmitted on board. Further, other sources of uncertainty will not affect the design much from the mass point of view. In Figure 3.39(a), the black vertical arrow represents the 20% margin applied to the mean (nominal) value of the subsystems mass. A classical margins approach just providing the margin with respect to the mean value, will not convey any other kind of knowledge on the uncertainty structure and on the sensitivity with respect to the uncertain factors.

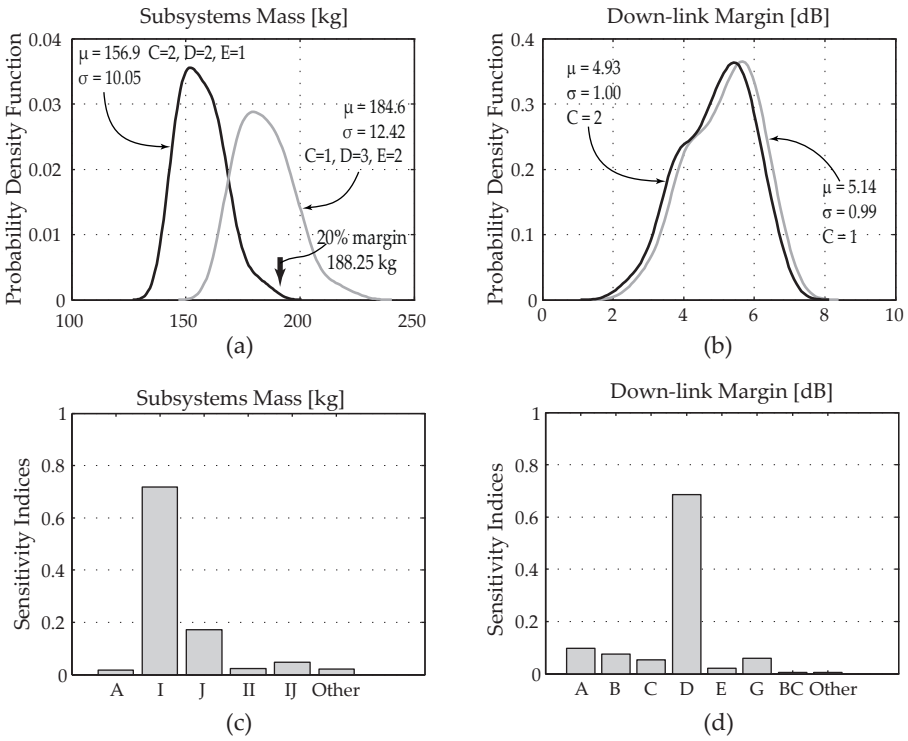


Figure 3.39 Communication and power subsystems robust design and uncertainty analysis. Probability density function of the most robust (black lines) and least robust (gray lines) configuration on the a) Subsystems mass, b) Down-link margin. c) Sensitivity analysis of the subsystems mass to the uncertain factors. d) Sensitivity analysis of the down-link margin to the uncertain factors.

3.5 Summary

Design-space exploration is the fundamental activity with which the model of a complex system is analyzed to understand the effect of the design choices on the performance(s) and to set the values of the variables in such a way that the final product will perform as required by the customer(s). This activity often involves many stakeholders, with many objectives to be balanced, many constraints and many design variables, thus posing the problem to be extremely difficult to solve with a non-structured approach. The purpose of this chapter was to discuss subsequent analysis steps and synthesis methodologies that could serve as a guideline for exploring the design space of complex models in a standardized and possibly more efficient way. The following common design questions can be answered by using the methods presented in this chapter:

Amongst all the design factors of the system model, what are those actually influencing the performance of interest? To what extent do these factors influence the performance?

In case of uncertainties in the factors influencing the performance of the system, how do they propagate through the model? And what are the factors that are mostly responsible for performance uncertainty?

What is the shape of the design-space? And what are the best parameter settings to optimize the objectives and meeting the constraints?

How robust is(are) the baseline(s)?

The AMH is slightly more elaborated than other conventional sampling techniques but it allows the engineering team to gain a great deal of insight in the problem at hand with continuous and discrete, controllable and uncontrollable design variables with one unified method. The final baseline of the Earth-observing satellite, for instance, was selected according to a non-conventional mission architecture for an observation satellite, *i.e.,* quite a high orbit altitude. This choice was mostly driven by the need to balance the coverage requirement and the resolution performance, while keeping the cost down. The *risk* of obtaining conventional design baselines is concrete when non-structured, expert-judgment driven approaches are implemented. However, very often, especially in preliminary design phases, expert judgment is a fundamental ingredient to a good system baseline. In fact, the AMH also allows to take expert-judgment into account with a unified epistemic-stochastic sampling approach.

The RBSA presented in this chapter, coupled with the AMH, demonstrated the characteristics of providing very precise quantitative information on the importance of the factors at a reduced computational effort in the case of linear and non-linear problems, even with a large number of variables. Further, it also demonstrates the possibility of obtaining quantitative indices also of the single effects involving the design variables, information that is not available with other sensitivity analysis methods. In case of highly non-linear and highly non-monotonic problems, the RBSA is able to provide at least a qualitative indication on the importance of the factors and their ranking, even when other qualitative screening methods fail. In the case of the design of a complex system, composed of many interacting elements and sub-elements with many variables to be taken into account, the RBSA can help in supporting the engineering team by lowering the computational cost and time to obtain quantitative results.

One last remark regards the possibility to use the AMH for a wider search. The analysis performed with the AMH, as presented in this chapter, is restricted to the portion of the design space delimited by the variability ranges of the design variables. Sometimes a single hypercube is sufficient to entirely cover the design space, sometimes instead a narrower hypercube might be needed to avoid major lack-of-fit conditions. In this case more than one hypercube may be implemented to study different regions of the design space as different alternative baselines of the system. In this case, the methodologies presented in this chapter will not only support the engineering team in selecting the best configuration for each single baseline, but will also allow to compare and trade between the baselines based on their performances, constraint-violation conditions and robustness.

Chapter 4
Global Design Approach

The local approach based on the Augmented Mixed Hypercube (AMH) discussed in the previous chapter, provides a structured framework for the engineering team to explore the design space in the neighborhood of a specific point, *e.g.*, a design baseline. The AMH has some limitations. One of them is that when the design region of interest becomes larger a polynomial representation of the system (the fulcrum of the AMH approach) may not be accurate enough. This aspect limits the dimensions of the AMH, but on the other hand when the AMH is too small, large parts of the design space may be left unexplored. This is an undesired side-effect of the proper utilization of the AMH approach. For a well-informed decision-making process, the design space shall be explored to a large extent, instead. In this way, the risk of neglecting potentially optimal solutions is reduced and eventually eliminated.

Multiple augmented mixed hypercubes generated in different regions of the design space may help the engineering team to better explore the design space. There is the risk, however, that computational resources are invested in portions of the design space that will yield sub-optimal solutions. There are more efficient methods that can be implemented when the purpose is to find optimal solutions to a problem characterized by having many objectives and constraints, with continuous and discrete variables covering a large design region. Heuristic Multi-Objective Optimization (MOO) algorithms, for instance, seem to be the most suitable approach (Pardalos and Romeijn, 2002; Holland, 1975; Goldberg, 1989; Kennedy *et al.*, 2001). These algorithms provide a set of global-optimal solutions with respect to all the objectives and constraints at the same time.

Even though it is empirically proven that excellent results can be obtained using MOO algorithms, it is also true that some optima could be the result of a particular combination of design variables that will exhibit a steep drop in performance when the levels of the design variables are only slightly modified. For this reason, in this chapter we introduce a novel global design approach for the design and optimization of complex systems based on a synergistic utilization of the global MOO and the local AMH approach. It is called PROA, Pareto Robust Optimization Algorithm, and it brings benefits to the engineering team in understanding the quality of the optimal solutions provided by standard optimizers.

In Section 4.1 of this chapter we describe the main characteristics of global multi-objective optimization. Some global optimizers are compared and an applicative example of the *Lunar space-station* test case is provided to show the benefits of having global optimization already at conceptual-design level. In Section 4.2 we discuss the Pareto Robust Optimization Algorithm, providing some validation examples. Further, we apply PROA to the Satellite system for *localized* Earth-observation test case.

4.1 Global multi-objective optimization

The problem of designing and optimizing a space system, considering its operative environment and the mission it will accomplish, is highly constrained and characterized by having multiple objectives, with continuous and discrete (*e.g.,* architectural) variables. A generalized mathematical formulation is shown hereafter:

$$\begin{aligned} Minimize \quad & f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})]^T \in \mathcal{F} \\ where \quad & \mathbf{x} = [x_1, x_2, \ldots, x_N]^T \in \mathcal{X} \\ & g_j(\mathbf{x}) \geq 0 \qquad j = 1, \ldots, J \\ subject\ to \quad & h_k(\mathbf{x}) = 0 \qquad k = 1, \ldots, K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} \qquad i = 1, \ldots, N \end{aligned} \tag{4.1}$$

The $N \times 1$ vector $\mathbf{x}$ is the design vector (*i.e.,* , the design-variable set) and the $M \times 1$ vector $f(\mathbf{x})$ contains the objective functions. The problem has J inequality constraints (g), and K equality constraints (h). $\mathcal{X}$ represents the design space, while $\mathcal{F}$ is the objective space. Every design variable may assume values between a minimum ($x_i^{(L)}$) and a maximum ($x_i^{(U)}$). The intersection of all the intervals $\left[x_i^{(L)}, x_i^{(U)}\right]$ forms the design-region of interest or design search-space, which is a subset of $\mathcal{X}$. A minimization problem can be transformed into a maximization one by multiplying with -1 the objectives.

In multi-objective optimization problems the *optimum* is treated differently compared to single-objective optimization problems. The former aims at finding a set of good compromises, *i.e.,* trade-offs, rather than a single optimal solution, by optimizing all the objectives simultaneously. This set of solutions is found using the Pareto-optimality concept. A solution is defined to be Pareto-optimal or *non-dominated* if there is no feasible solution for which one cannot improve a single objective without causing a degradation of at least one other objective. According to the Pareto-optimality concept, a vector $\mathbf{a} \in X$ is said to dominate another vector $\mathbf{b} \in X$ in a minimization problem, also written as $\mathbf{a} \prec \mathbf{b}$, if and only if the following relationship holds:

$$\forall i \in \{1, \ldots, N\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \wedge \exists i \in \{1, \ldots, N\} : f_i(\mathbf{a}) < f_i(\mathbf{b})$$

The set of *non-dominated* vectors, plotted in the objective space, is defined as the Pareto front, schematically shown in Figure 4.1. The determination of the true Pareto front, PF_{true} (*i.e.,* the theoretical obtainable Pareto front), depends on many aspects such as the complexity of the problem of interest, the number of design variables and objectives, the nature of the front itself (concave/convex, continuous/discrete) and the number of function evaluations executed.

An optimization problem posed in the form of Eq. (4.1) can be solved following different approaches. However, not all of them are applicable and some of them are more effective than others, especially for design spaces of high dimensionality, with both continuous and discrete variables. In the following paragraphs we will briefly provide an overview of existing methods in the literature that may potentially be used to solve the MOO problem.

Local optimization

The class of local gradient-based techniques mathematically guarantees that an optimal solution is reached. Local gradient-based methods require continuity in the search space and in the space of the objective functions and constraints, and their first derivatives. Especially when

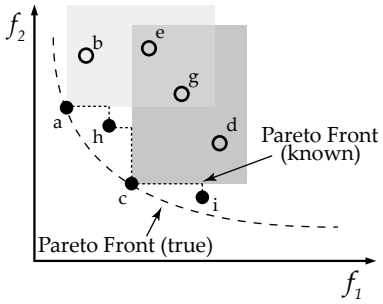


Figure 4.1 Schematic representation of the concept of the Pareto-dominance. $a \prec b, e, g$ and $c \prec e, g, d$. The solutions h and i are not dominated by any other current solution.

dealing with architectures of a space system, the design variables are not always continuous, *e.g.,* the choice of a particular launcher. When also databases are included in the design process, continuity does not even apply. Further, such methods are characterized by finding local optima, locked in the impossibility to overcome this limitation. Global optimization methods, on the other hand, may cover a large portion of the design space while searching for the optimum, and provide mechanisms for avoiding local optima, *e.g.,* random mutation in the case of genetic algorithms (Goldberg, 1989). It is for this reason that global optimization methods are considered in this chapter as a tool in support of the engineering team for the exploration of the design space.

Global optimization - deterministic methods

Deterministic methods like branch-and-bound algorithms (Back *et al.*, 2000; Mitten, 1970), relaxation strategies, enumerative methods (Pardalos and Romeijn, 2002), and interval-analysis methods show poor convergence in some cases, and a rapid increase of computational effort when the dimensions of the search space increase (Alefeld and Mayer, 2000). Dynamic Programming (DP) is a combinatorial optimization technique, which demonstrated to reach exact solutions for problems with specific formulations (also multi-objective as demonstrated by Abo-Sinna and Hussein (1995)), involving the solution of subproblems of similar nature to build the global optimal solution (Bellman and Dreyfus, 1962). The modification of the problem structure to be solvable by a DP algorithm is not always possible, especially in collaborative, possibly distributed, design environments. Classical methods for the generation of the Pareto front like the Normal Boundary Intersection (Das and Dennis, 1998), the Adaptive Weighted-Sum (Kim and de Weck, 2005), the Direct Search Domain (Erfani and Utyuzhnikov, 2010), and the Normal Constraint method (Messac and Mattson, 2004), to mention a few, have shown a good performance in finding Pareto-optimal solutions for multi-objective, constrained, continuous and discontinuous, optimization problems. A good overview of these methods and a comparison of the performance of a few of them is presented in Shukla and Deb (2007).

Global optimization - heuristic methods

The non-classical heuristic methods like evolutionary strategies (Back *et al.*, 2000), simulated annealing (Sanguthevar, 2000), and tabu-search (Tan *et al.*, 2003; Glover, 1989, 1990), proved to be particularly flexible, and applicable to continuous and discontinuous problems with one or more objectives and constraints (Deb, 2001; Coello Coello *et al.*, 2007). Also other approaches exist that exploit alternative formulations of the multi-objective problem. The Iso-performance method, for instance, allows for obtaining optimal solutions amongst those that were previously determined to meet the performance requirements with sufficient margins (de Weck

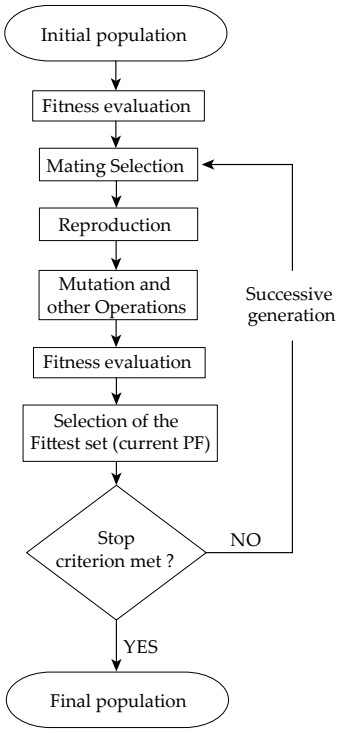


Figure 4.2 Schematic of a general implementation of a heuristic algorithm for multi-objective optimization.

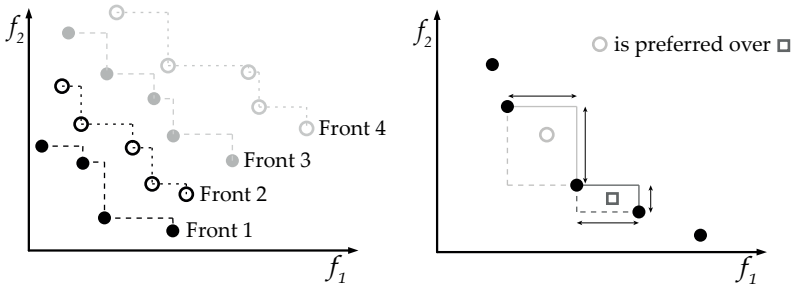
et al., 2002). Physical Programming, instead, demonstrated the possibility to generate Pareto fronts in multi-objective problems considering experts judgment already during the optimization (Messac and Mattson, 2002). Heuristic methods will be considered in this thesis as global optimization methods. The reason is that they are flexible, scalable, easy to use, and problem-independent.

4.1.1 Popular heuristic multi-objective optimization approaches

During the last twenty years, many multi-objective optimization techniques and algorithms have been developed and implemented to solve ad-hoc mathematical problems (Fonseca and Fleming, 1995; Van Veldhuizen and Lamont, 1998). Possible distinctions are made between stochastic and deterministic, or between algorithms that use scalarization of the multi-objective problem and others that do not decompose the problem to solve it. Some of the most widely used multi-objective optimization methods have been mentioned at the beginning of this section, therefore for the details readers are encouraged to read the original studies.

Stochastic MOO algorithms demonstrated to be very flexible and their formulation does not require knowledge about the model to be optimized. They are easy to use, and easy to implement with general problems (continuous and discrete). For these reasons, in this thesis we consider a specific class of stochastic MOO algorithms as Pareto-generating techniques, namely the *evolutionary* algorithms. This class of algorithms is called *evolutionary* because it mimics the evolutionary behavior of living species, *e.g.,* transmission of genetic material (in this case represented by the design parameters) from a generation to the next.

In the following subsection a brief description of three popular MOO evolutionary algorithms is provided, with a comparison on constrained and unconstrained test problems. These algorithms all work according to the general scheme presented in Figure 4.2. However, each of them has a particular characteristic that differentiates it from the others in terms of selec-



(a) Non-domination sorting (b) Crowding distance

Figure 4.3 Selection and diversity preservation principles of the NSGAII algorithm.

tion of the individuals, mating amongst the individuals, and maintaining diversity within the population.

Once the initial population is generated (selecting the design-variable levels) and evaluated (executing the model simulations), a new population is created by using the characteristics of the fittest individuals in the population. The new population is then evaluated, and the process starts again with the best set of individuals. To the best of our knowledge, the most popularly used evolutionary algorithms are the Non-dominated Sorting Genetic Algorithm II (NSGAII), the Multi-Objective Particle Swarm Optimization (MOPSO), and the Multi-Objective Evolutionary Algorithm with Decomposition (MOEA/D).

Non-dominated Sorting Genetic Algorithm II

The Non-dominated Sorting Genetic Algorithm II proposed by Deb *et al.* (2002), was developed as an improved version of the NSGA introduced earlier (Srinivas and Deb, 1995). As in the original concept of genetic algorithms (developed for single-objective implementations), NSGAII uses techniques inspired by natural evolutions, *e.g.,* crossover and mutation for the reproduction phase, as described already by Back *et al.* (2000). The selection of the individuals is based, instead, on the non-domination principle. At every step of the optimization process, once the model is evaluated the solutions are sorted according to their fitness in the Pareto sense, *i.e.,* non-domination sorting. Thus, the individuals of the new population are selected amongst the best non-dominated individuals of the old population and its offspring, obtained during the reproduction phase.

There are several approaches that could be used to sort the individuals, see, for instance, the discussion in Deb *et al.* (2002). The result is that the population is subdivided into several fronts, ranked in order of *optimality*. In Figure 4.3(a) the solutions on the objective space have been assigned to four different fronts. The population of the successive iteration of the optimization process is selected starting from the individuals on the first front, until the maximum number of individuals is reached. The mechanism of the crowding distance, schematically represented in Figure 4.3(b), is one of the mechanisms discussed in the literature to maintain diversity within the population. In this particular case, for instance, the solution represented by the gray circle is preferred over the one represented by the square. This is due to the larger distance between the gray circle and its neighbor solutions on the Pareto front if compared to the distance between the square and its neighbor solutions on the Pareto front.

Figure 4.4 A flock of birds searching for food, Leiden, The Netherlands. Credit: Guido Ridolfi.

Multi-Objective Particle Swarm Optimization

The Multi-Objective Particle Swarm Optimization proposed by Coello Coello *et al.* (2004), is an extension of the Particle Swarm Optimization (PSO) approach developed by Kennedy and Eberhart (1995). The PSO is a distributed behavioral algorithm, also classified as an agent-based algorithm, inspired by the social dynamics of groups of individuals (*e.g.,* flocks of birds or fish schoolings) searching for resources, see Figure 4.4. Birds, for instance, are able to communicate between each other the position where food has been found. Further, they are able to remember the geographical position where they found food themselves. In searching for food, their exploration is driven by these two indications. Inheriting this principle, the velocity of the particles (which are the vectors with the design-variable values) in the design space and their position, at each step of the optimization process, are determined by the following equations (Kennedy and Eberhart, 1995):

$$\mathbf{v}_{i+1} = w \cdot \mathbf{v}_i + c_1 \cdot r() \cdot (\mathbf{x}_i^* - \mathbf{x}_i) + c_2 \cdot r() \cdot (\mathbf{x}_G^* - \mathbf{x}_i) \tag{4.2}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i \tag{4.3}$$

The term $w \cdot \mathbf{v}_i$ is an inertial contribution to the current movement of the particle in the search space. The term $c_1 \cdot r() \cdot (\mathbf{x}_i^* - \mathbf{x}_i)$ influences the velocity, $\mathbf{v}$, proportionally to the distance between the current position and the best position found by the particle itself during its *journey* in the search space. The term $c_2 \cdot r() \cdot (\mathbf{x}_G^* - \mathbf{x}_i)$ is proportional, instead, to the distance between the current position and the best position of the whole population in the search space. The parameters w, c_1 and c_2 need to be tuned properly, depending on the problem at hand. The parameter $r()$ is a randomly-generated number between 0 and 1. The extension of this concept to a multi-objective case, using the concept of Pareto-dominance to evaluate the *goodness* of the particles, was introduced by Coello Coello *et al.* (2004). Reyes Sierra and Coello Coello (2005) propose an improvement of the method by incorporating the concept of crowding distance discussed before.

Multi-Objective Evolutionary Algorithm with Decomposition

The Multi-Objective Evolutionary Algorithm with Decomposition proposed by Zhang and Li (2007) is an alternative method for computing the Pareto front of problems with multiple objectives. The MOEA/D method is based on the decomposition of the problem into a number of scalar sub-problems and on their simultaneous optimization. Consider, for instance, a two-objective (f_1 and f_2) problem. The transformed scalar optimization problem can be formulated as the optimization of the functional $F = \lambda_1 f_1(\mathbf{x}) + \lambda_2 f_2(\mathbf{x})$, where the λ_i are coefficients subject to $\sum \lambda_i = 1$, and $\mathbf{x}$ is the vector of the variables. This weighted-sum approach allows for generating a set of N different Pareto-optimal vectors by using N different combinations of weights. In correspondence with each combination of weights, an optimal solution is found

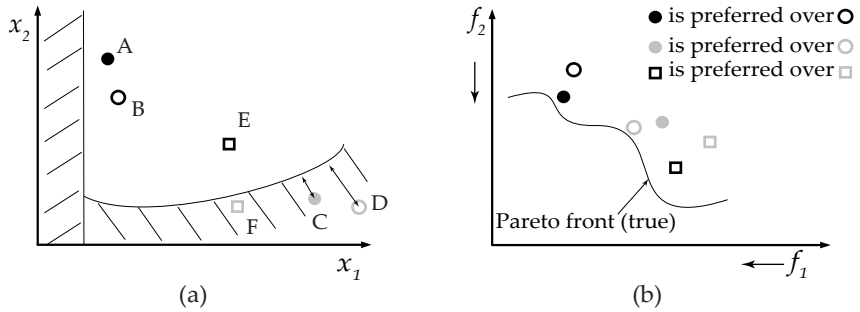


Figure 4.5 Dominance principle in presence of constraints.

for the functional F. These represent the Pareto-optimal solutions. The Tchebycheff approach and the boundary-intersection approach for the decomposition of the multi-objective problem are discussed and compared in Zhang and Li (2007), to which the reader is referred for more detailed information on the MOEA/D method.

4.1.2 Dealing with constraints

The evolutionary algorithms considered for the global optimization processes presented in this thesis, all have the characteristics for which pairs of individuals (*i.e.,* solutions) are compared to select the most suited ones for the evolutionary process in common. The individuals are compared on the basis of the Pareto-dominance principle and eventually also considering the crowding distance. When the multi-objective problems have constraints, these shall also be considered in the comparison between individuals. The general principle that we adopted is that a feasible solution shall always be preferred over an infeasible solution, and that between two infeasible solutions the *less infeasible* shall be preferred. When both solutions do not violate the constraints, then they are compared as in the unconstrained case, *i.e.,* based on the Pareto dominance and crowding distance. A schematic representation of the dominance principle in presence of constraints can be found in Figure 4.5. Clearly, solution A is preferred over solution B because it is better in the Pareto sense. Solution E is preferred over solution F because solution F is infeasible. Solution C is preferred over solution D because, even if they are both infeasible, solution C is closer to the constraint (*i.e.,* less infeasible).

4.1.3 Comparison of algorithms for multi-objective optimization.

For comparison purposes we tested the performances of NSGAII, MOPSO, and MOEA/D on many of the test problems proposed by Zitzler *et al.* (2000), Deb *et al.* (2005), Deb (2001), and Coello Coello *et al.* (2007).

Deb (1999) identified several characteristics that may prevent an MOO algorithm to converge to the true Pareto front and to maintain diversity of the solutions. The Pareto front is harder to reach in the case of multimodal and/or deceptive problems, and in problems with isolated optima. Non-convexity and non-uniformity of the Pareto front, or discreetness of the problem may lead to conditions for which diversity in the solutions is hard to maintain. In Tables 4.1 and 4.2 the test functions used for comparing the three algorithms are presented. In general two objectives are considered sufficient to reflect essential aspects of multi-objective optimization (Coello Coello *et al.*, 2007). Moreover, we only consider minimization problems, since maximization or mixed maximization/minimization would not be more informative.

The results presented in the following figures were obtained using a population size of 100 individuals, with a maximum of 100 generations allowed, for all the algorithms compared. The number of Pareto individuals is also a measure of the quality of the algorithms. In Figure

Problem	n	Variable Bounds	Objective Functions	Comments
ZDT1	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i/(n-1)$	Convex
ZDT2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^2\right]$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i/(n-1)$	Non-convex
ZDT3	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1 + 9\sum_{i=2}^{n} x_i/(n-1)$	Convex Disconnected
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2 \dots n$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right]$	Non-Convex
ZDT5[a]	11	$x_1 \in [0, 30]$ $x_i \in [0, 5]$ $i = 2 \dots n$	$f_1(x) = 1 + u(x_1)$ $f_2(x) = \frac{\sum_{i=2}^{n} v(u(x_1))}{f_1(x)}$ $v(u(x_1)) = \begin{bmatrix} 2 + u(x_1) & if\ u(x_1) < 5 \\ 1 & if\ u(x_1) = 5 \end{bmatrix}$	Deceptive Discrete
ZDT6	10	$[0, 1]$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)\left[1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right]$ $g(x) = 1 + 9\left[\sum_{i=2}^{n} x_i/(n-1)\right]^{0.25}$	Non-convex Non uniformly spaced
DTLZ1	3	$[0, 1]$	$f_1(x) = 0.5(1 + g(x)) \cdot x_1 \cdot x_2$ $f_2(x) = 0.5(1 + g(x)) \cdot x_1 \cdot (1 - x_2)$ $f_3(x) = 0.5(1 + g(x)) \cdot (1 - x_1)$ $g(x) = 100\left[1 + (x_3 - 0.5)^2 - \cos(20\pi(x_3 - 0.5))\right]$	$(11^n - 1)$ local Pareto fronts
DTLZ2	3	$[0, 1]$	$f_1(x) = [1 + g(x)]\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)$ $f_2(x) = [1 + g(x)]\cos\left(x_1\frac{\pi}{2}\right)\sin\left(x_2\frac{\pi}{2}\right)$ $f_3(x) = [1 + g(x)]\sin\left(x_1\frac{\pi}{2}\right)$ $g(x) = (x_3 - 0.5)^2$	$(11^n - 1)$ local Pareto fronts

Table 4.1 Unconstrained test functions. Adapted from (Zitzler *et al.*, 2000) and Deb *et al.* (2005). [a] $u(x_i)$ gives the number of ones in the bit vector x_i(unitation).

4.6 the Pareto fronts obtained using the optimization algorithms on the first six problems of Table 4.1 are presented. In all the cases MOEA/D reached the true Pareto front with evenly distributed solutions. In most cases both NSGAII and MOPSO did not provide the true Pareto front, remaining stuck to suboptimal solutions. In the ZTD4 and ZTD6 problems both NSGAII and MOPSO provided extremely unsatisfactory results.

The DTLZ1 and DTLZ2 problems, despite presenting three objectives, are relatively easy problems to solve. The three algorithms reached the true Pareto front in both cases. The true Pareto front for the DTLZ1 problem is a plane passing through the points with coordinates $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. The true Pareto front for the DTLZ2 problem is a $\frac{1}{8}$ of sphere, in the positive octant, centered in 0 with radius equal to 1. However, as demonstrated in Figures 4.7 and 4.8 MOPSO and MOEA/D provided better results in terms of diversity of the solutions on the Pareto front. Indeed, NSGAII reached clusters of solutions localized in some regions of the Pareto front, which is an unwanted behavior for an MOA. In all cases the three algorithm reached the true Pareto front.

Finally, in Figure 4.9 we present the results obtained on the constrained optimization problems of Table 4.2. The constraint-handling mechanism described in Section 4.1.2 allowed the algorithms to cope with the proposed constrained problems. Indeed, NSGA-II and MOEA/D obtained the correct constrained solutions in all cases. MOPSO did not perform as well as the other two algorithms that we implemented. In the TNK problem both NSGA-II and MOEA/D

Problem	n	Variable Bounds	Objective Functions	Comments
TNK[a]	2	$[0, \pi]$	$f_1(x) = x_1$ $f_2(x) = x_2$ Subject to $-x_1^2 - x_2^2 + 1 + a \cdot \cos(b \cdot \arctan(x_1/x_2)) < 0$ $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 < 0$	Disconnected Non-Linear constraints
DTLZ9[b]	-	$[0, 1]$	$f_j(x) = \sum_{i=\lfloor (j-1)\frac{n}{2} \rfloor}^{\lfloor j \frac{n}{2} \rfloor} x_i^{0.1} \quad j = 1, 2$ Subject to $f_2^2(x) + f_1^2(x) - 1 \geq 0$	Constrained surface

Table 4.2 Constrained test functions. Adapted from Coello Coello *et al.* (2004), and Deb *et al.* (2005). [a] TNK problem $a = 0.1, b = 32$; TNK-II problem $a = 0.1\left(x_1^2 + x_2^2 + 5x_1x_2\right), b = 32$; TNK-III problem $a = 0.1\left(x_1^2 + x_2^2 + 5x_1x_2\right), b = 8\left(x_1^2 + x_2^2\right)$. [b] Increasing number of variables, *i.e.*, 20, 30, and 60, see Figure 4.9.

reached the correct solution. In Figure 4.9, for the three TNK problems, the inside area delimited by the black lines is the feasible area. For all the other test problems MOEA/D performed better, especially with a large number of design variables. The arrows represent points in the objective space where MOEA/D provided a solution while NSGA-II and/or MOPSO did not. With an increasing number of variables, in the case of problem DTLZ9, MOEA/D was able to provide a stable solution to the multi-objective problem, while NSGA-II suffered, remaining stuck to one of the many local Pareto fronts. In these two last graphs the results obtained with the MOPSO algorithms are not shown.

As a result of the testing process, we can conclude that MOEA/D reaches the Pareto front quickly when compared to MOPSO and NSGAII, and it is more accurate in the determination of the Pareto front, maintaining a higher level of diversity of the solutions. This means that using MOEA/D there are high chances of getting solutions close to the true Pareto front and that the solutions presented at the end of the process are *more diverse* from each other, giving more degrees of freedom for the final decision process. Due to its capability to deal with constrained multi-objective problems, in the presence of different types of Pareto fronts, and due to its convergence speed, accuracy, and solution diversity characteristics, we decided to use the MOEA/D algorithm for the optimization processes discussed in this chapter. In this subsection we have often mentioned the true Pareto front. It is the Pareto front that can be computed analytically from the problem definition, *i.e.*, it is the region in the design space where the dominance principle, discussed at the beginning of the chapter, is valid. Later in this chapter we will also consider the *known* Pareto front defined as the best approximation of the Pareto front obtained with a MOO algorithm. For more details on the methodologies briefly introduced in this section, and for more information regarding the test problems, their formulation, and the analytical Pareto fronts (*i.e.*, the mathematical expressions of the true Pareto fronts), readers are encouraged to refer to the original studies.

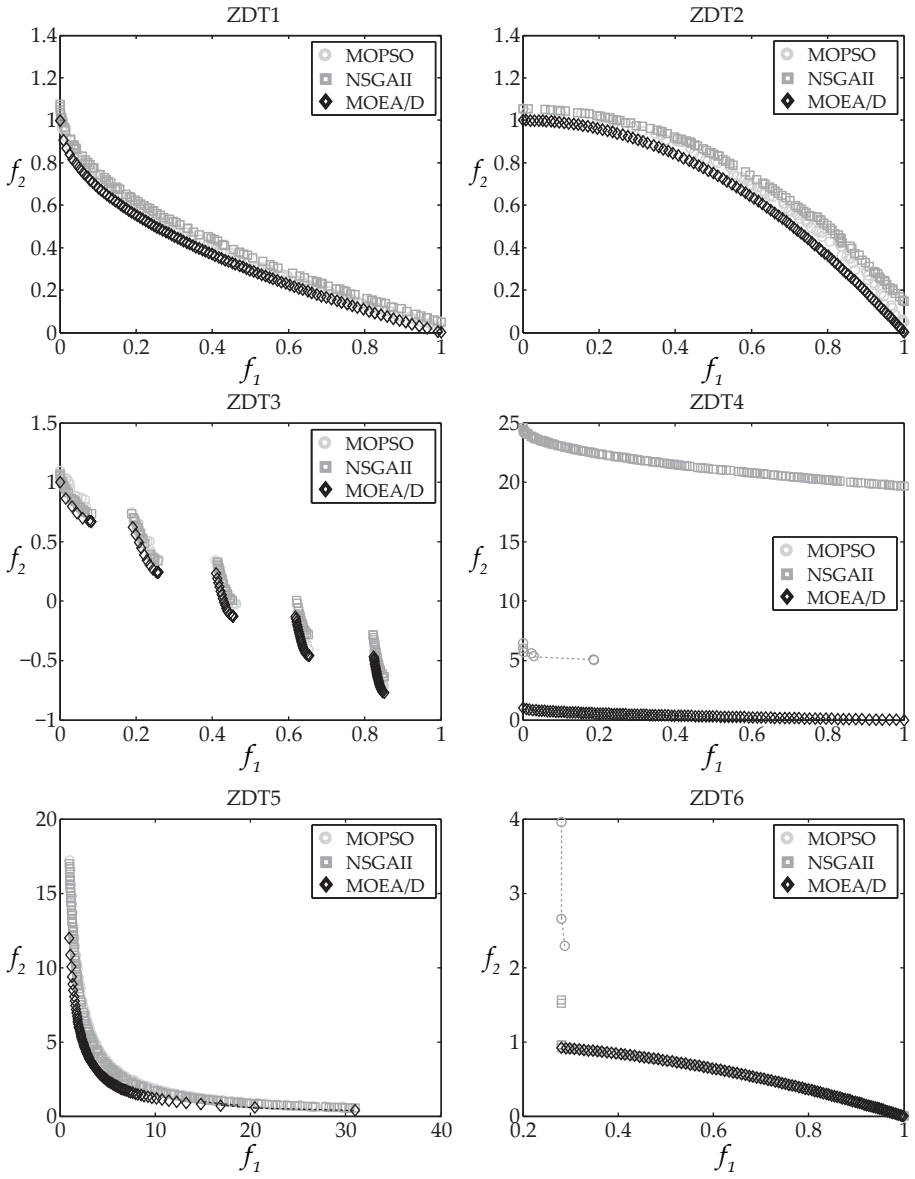


Figure 4.6 Comparison of the MOA on six unconstrained problems.

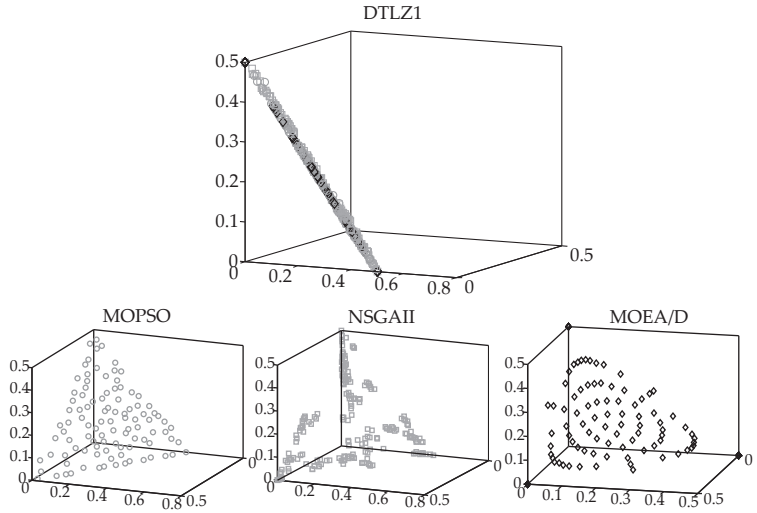


Figure 4.7 Comparison of the MOAs on the DTLZ1 problem. On the top graph a side visualization of the Pareto front obtained with the three algorithms is shown. On the graphs at the bottom, the Pareto fronts are shown separately as obtained by each algorithm.

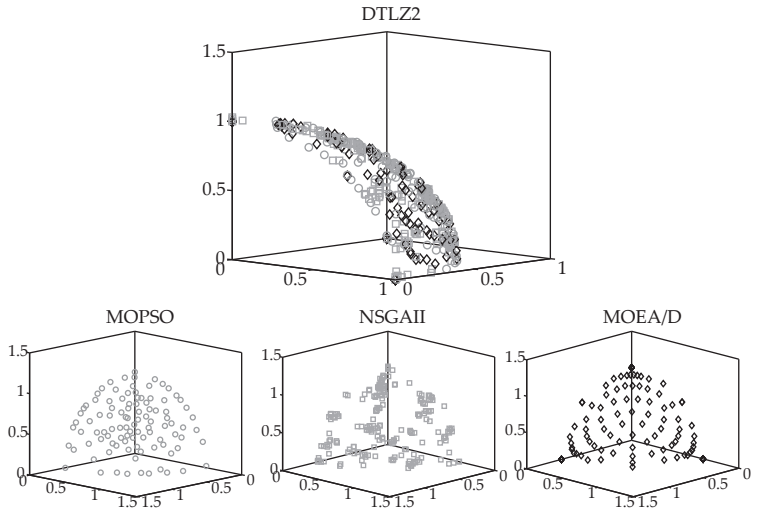


Figure 4.8 Comparison of the MOAs on the DTLZ2 problem. On the top graph a side visualization of the Pareto front obtained with the three algorithms is shown. On the graphs at the bottom, the Pareto fronts are shown separately as obtained by each algorithm.

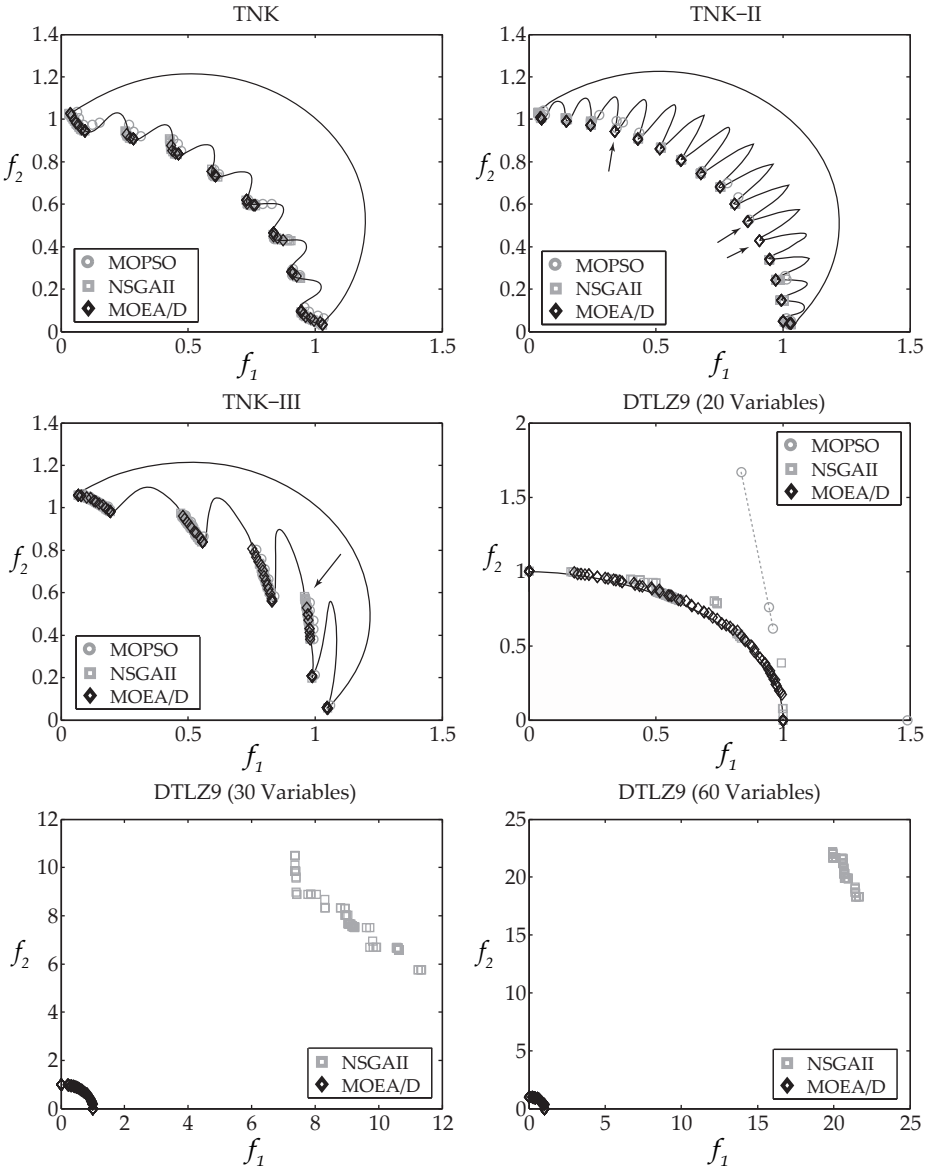


Figure 4.9 Comparison of the MOA on four constrained problems. The arrows represent points in the objective space where MOEA/D provided a solution while NSGA-II and/or MOPSO did not. The solid lines represent the constraints.