ABSTRACT

Title of Document: APPROXIMATION ASSISTED MULTIOBJECTIVE AND COLLABORATIVE ROBUST OPTIMIZATION UNDER INTERVAL UNCERTAINTY

Weiwei Hu, PhD, 2012

Directed By: Professor Shapour Azarm
Department of Mechanical Engineering

Optimization of engineering systems under uncertainty often involves problems that have multiple objectives, constraints and subsystems. The main goal in these problems is to obtain solutions that are optimum and relatively insensitive to uncertainty. Such solutions are called robust optimum solutions. Two classes of such problems are considered in this dissertation. The first class involves Multi-Objective Robust Optimization (MORO) problems under interval uncertainty. In this class, an entire system optimization problem, which has multiple nonlinear objectives and constraints, is solved by a multiobjective optimizer at one level while robustness of trial alternatives generated by the optimizer is evaluated at the other level. This bi-level (or nested) MORO approach can become computationally prohibitive as the size of the problem grows. To address this difficulty, a new and improved MORO approach under interval uncertainty is developed. Unlike the previously reported bi-level MORO methods, the improved MORO performs robustness evaluation only for optimum solutions and uses this information to iteratively shrink the feasible domain and find the location of robust optimum solutions. Compared to the previous bi-level

approach, the improved MORO significantly reduces the number of function calls needed to arrive at the solutions. To further improve the computational cost, the improved MORO is combined with an online approximation approach. This new approach is called Approximation-Assisted MORO or AA-MORO.

The second class involves Multiobjective collaborative Robust Optimization (McRO) problems. In this class, an entire system optimization problem is decomposed hierarchically along user-defined domain specific boundaries into system optimization problem and several subsystem optimization subproblems. The dissertation presents a new Approximation-Assisted McRO (AA-McRO) approach under interval uncertainty. AA-McRO uses a single-objective optimization problem to coordinate all system and subsystem optimization problems in a Collaborative Optimization (CO) framework. The approach converts the consistency constraints of CO into penalty terms which are integrated into the subsystem objective functions. In this way, AA-McRO is able to explore the design space and obtain optimum design solutions more efficiently compared to a previously reported McRO.

Both AA-MORO and AA-McRO approaches are demonstrated with a variety of numerical and engineering optimization examples. It is found that the solutions from both approaches compare well with the previously reported approaches but require a significantly less computational cost. Finally, the AA-MORO has been used in the development of a decision support system for a refinery case study in order to facilitate the integration of engineering and business decisions using an agent-based approach.

APPROXIMATION ASSISTED MULTIOBJECTIVE AND COLLABORATIVE ROBUST OPTIMIZATION UNDER INTERVAL UNCERTAINTY

by

Weiwei Hu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Dr. Shapour Azarm, Advisor and Chair
Dr. Ali Almansoori (The Petroleum Institute, Abu Dhabi, UAE)
Dr. Nikhil Chopra
Dr. Jeffrey W. Herrmann
Dr. P. K. Kannan (Dean's Representative)

ACKNOWLEDGEMENTS

I would like to express my foremost appreciation to Professor Shapour Azarm for his guidance and continuous support throughout my Ph.D. study at the University of Maryland (UMD), College Park. His energy and passion for doing ethical, methodical and careful research and his attention to details have made me grow academically, personally and professionally.

I also would like to thank the following faculty for agreeing to serve on my dissertation committee. Special thanks to Professor P.K. Kannan for generously providing his time and expertise in helping me better model the business aspects. I am indebted to Professor Ali Almansoori from The Petroleum Institute (PI), UAE, for providing his expert insights on chemical engineering aspects of my research. I am grateful that he is coming to Maryland for my defense! I am also thankful to other members of my committee: Professor Jeffrey Herrmann and Professor Nikhil Chopra for providing comments for improving my dissertation.

During my study at UMD, I have been fortunate to know and have friendships with many great people. I would like to specially thank Mr. Khaled Saleh for discussing various research problems and having other interesting conversations. I also would like to thank Zhichao Wang, Dr. Josh Hamel, Amir Mortazavi, Adeel Butt, Phillippe Kamaha for sharing their research expertise with me. Special thanks to Dr. Mian Li who provided some inspiring ideas in my initial years in Maryland that helped me with my Ph.D. research.

I am most grateful to my parents. I specially want to thank my father for giving me the aspiration to pursue my education in engineering, and my mother for believing in me. Last but not least, I would like to thank my girlfriend Sally for her consistent love and support.

I would like to acknowledge the Department of Mechanical Engineering at UMD for providing the right environment for doing my graduate study and research. The work

presented in this dissertation was supported in part by The Petroleum Institute, Abu Dhabi, UAE, as part of the Education and Energy Research Collaboration (EERC) agreement between the PI and UMD. This support is gratefully acknowledged.

LIST OF ABBREVIATIONS

AA-MORO	Approximation Assisted MultiObjective Robust Optimization
AA-McRO	Approximation Assisted Multiobjective collaborative Robust Optimization
AAO	Approximation Assisted Optimization
AIO	All-In-One
AOVR	Acceptable Objective Variation Range
ATC	Analytical Target Cascading
CO	Collaborative Optimization
DOE	Design of Experiments
DSS	Decision Support System
GA	Genetic Algorithm
HD	Hyperarea Difference
LHS	Latin Hypercube Sampling
KPI	Key Performance Indicator
MAE	Mean Absolute Error
MED	Maximum Entropy Design
MSE	Mean Square Error
MCO	Multiobjective Collaborative Optimization
MDO	Multi-disciplinary Optimization
MOGA	MultiObjective Genetic Algorithm
MOP	Multiobjective Optimization Problem
MORO	MultiObjective Robust Optimization
McRO	Multiobjective collaborative Robust Optimization
NDS	Non-Dominated Sorting
OS	Overall Spread

NOMENCLATURE

c	Iteration counter for non-regret learning
c_{cru}	Unit cost of crude oil
c_{pro}	Unit price of end product
C_{res}	Other cost including capital, operating, utility, labor cost, etc.
Cov	Covariance
$\mathbf{f}$	Vector of objective functions
$\mathbf{f}_i$	Objective functions in subsystem i
$\mathbf{g}$	Vector of constraint functions
$\mathbf{g}_i$	Constraint functions in subsystem i
i	Index for subsystems
I	Number of subsystems
k	Index for constraint functions
k_i	Index for constraint functions in subsystem i
l	Lower bound
m	Index for objective functions
nc	Number of iterations in non-regret learning
nf	Number of objective functions
nf_i	Number of objective functions in subsystem i
nit	Number of iteration in the improved MORO
ng	Number of constraint functions
ng_i	Number of constraint functions in subsystem i
ngs	Number of generations in a Genetic Algorithm
nos	Number of optimum solutions in a Genetic Algorithm
np_1	Number of $\Delta\mathbf{p}$ values in S_f
np_2	Number of $\Delta\mathbf{p}$ values in S_g
np	Number of Pareto points
npr	Number of parameters
nps	Population size in a Genetic Algorithm
ns	Number of non-dominated ranks
nx	Number of design variables
$\mathbf{p}$	Parameters

$\mathbf{p}_i$	Parameters in subsystem *i*
$\mathbf{p}_{sh}$	Shared parameters
$\mathbf{P}$	Combination of local and shared parameters
$\mathrm{P}_{\mathrm{good}}$	Good point for calculating quality metric
$\mathrm{P}_{\mathrm{bad}}$	Bad point for calculating quality metric
q_{cru}	Quantity of end-product sale
q_{pro}	Crude oil feed flow rate
$R(\mathbf{x}_1, \mathbf{x}_2)$	Gaussian correlation function between $\mathbf{x}_1$ and $\mathbf{x}_2$
$R^c(\mathbf{x})$	Regret function calculated at the c^{th} iteration
S_f	Set of $\Delta\mathbf{p}$ values for the objective function
S_g	Set of $\Delta\mathbf{p}$ values for the constraint function
S_1	Set of different types of end products
S_2	Set of different types of crude oil
S_3	Set of resources for production
$\mathbf{t}_{ij}$	Target variables between subsystem *i* and subsystem *j*
u	Upper bound
$\mathbf{x}$	Vector of design variables
$\mathbf{x}_{\mathrm{I}}$	Vector of first-stage decision variables in the DSS
$\mathbf{x}_{\mathrm{II}}$	Vector of second-stage decision variables in the DSS
$\mathbf{x}_e$	An existing sample point in online approximation
$\mathbf{x}_i$	Vector of design variables in subsystem *i*
$\mathbf{x}_n$	A new sample point in online approximation
$\mathbf{x}_{sh}$	Vector of shared design variables
$\mathbf{x}^l$	Lower bound for vector of design variables
$\mathbf{x}^u$	Upper bound for vector of design variables
$\mathbf{X}$	Combination of local and shared vector of design variables
$\mathbf{y}_{ij}$	Vector of coupling variables between subsystem *i* and subsystem *j*
$Z(x)$	Gaussian random process function
ε	Threshold value
σ	Variance of Gaussian model
η_f	Acceptable limit for objective functions

η_y	Acceptable limit for coupling variables
$\Delta\mathbf{f}$	Vector of variation in objective functions
$\Delta\mathbf{p}$	Vector of uncertainty in parameters
$\Delta\mathbf{p}^l$	Lower bound of vector of uncertainty in parameters
$\Delta\mathbf{p}^u$	Upper bound of vector of uncertainty in parameters
$\Delta\mathbf{y}$	Vector of variation in coupling variables
Ω	Feasible domain
Ψ, Φ	Non-dominated set

Table of Contents

Chapter 1: Introduction and Research Thrusts

Many engineering optimization problems are multiobjective, constrained, and subject to various sources of uncertainty in their system inputs. For example, in designing a power tool, it is desirable to minimize its weight while maximizing the output power subject to the uncertainty in environment, loading conditions, part dimensions, and so on. Note that uncertainty in the system inputs can be transmitted through the system and produce large variations in the system outputs. Indeed, this input uncertainty could make the performance (objective and/or constraint functions) of an optimum solution too sensitive and therefore undesirable. Furthermore, current engineering system design problems are becoming increasingly complex and difficult to model and solve by an "all-in-one" formulation and therefore multiobjective multi-disciplinary optimization approaches are needed to be explored particularly when there is uncertainty in the inputs. One significant limitation of previous approaches in this area is that they require a large number of function calls in order to arrive at a solution. Therefore, the main focus of this dissertation is in developing new methods in robust optimization combined with an online approximation to improve the computational effort for obtaining solution for these problems when compared with previous methods.

The overall objective of this dissertation is to develop an approximation assisted multiobjective robust optimization methods for single- and multi-disciplinary optimization problems under uncertainty.

More specifically, this dissertation develops (i) a new approximation assisted MultiObjective Robust Optimization (MORO) approach; and (ii) a new approach for

the solution of multiobjective multi-disciplinary optimization problems. Furthermore, a new Decision Support System (DSS) in the context of a refinery case study has been developed that combines MORO with an agent-based approach to facilitate decision making under uncertainty when both business and engineering decisions are considered.

To develop these models and methods, three research thrusts are identified and explored.

1.1 Research Thrusts

A brief overview of the main thrusts and corresponding research in each thrust are presented in the following subsections.

Thrust 1: Approximation Assisted MultiObjective Robust Optimization (AA-MORO) under Interval Uncertainty

This first thrust is focused on developing a new online approximation assisted MORO approach that is able to significantly reduce the number of function call, compared to existing and comparable MORO techniques.

First, a new MORO approach is developed to reduce the number of robustness evaluations compared to a previous MORO approach. In a previous bi-level MORO (Li et al., 2006), candidate design points are identified and iteratively improved in an upper-level problem, while in a lower-level subproblem, the robustness of the candidate points are evaluated. However the new MORO is a sequential approach in which multiobjective optimal solutions are first obtained and subsequently the robustness of each optimal solution is evaluated. The sequential MORO requires

significantly fewer robustness evaluations and thus more efficient than the bi-level MORO.

Second, an online approximation technique is developed and integrated with the sequential MORO approach to further improve the computational efficiency. The online approximation can replace a computationally expensive objective/constraint function with an inexpensive metamodel. Furthermore, the metamodel is updated using the optimum solution points obtained as the approach iteratively proceeds towards the solution. In this way, the predictive capabilities of the metamodel is progressively improved in the area where the optimum is expected to be, as more and more sample points are evaluated and added to the sample set.

Thrust 2: Approximation Assisted Multiobjective collaborative Robust Optimization (AA-McRO) Under Interval Uncertainty

The second research thrust is focused on developing a new AA-McRO approach to improve the computational efficiency of a previous approach (Li and Azarm, 2008).

The new AA-McRO approach converts an upper-level system problem in McRO into two-level subproblems. Under this framework, the upper-level subproblem is responsible for coordinating the value of shared and coupling variables and guiding the optimization and robustness in lower-level subproblems. Because the upper-level subproblem in AA-McRO only focuses on coordination, AA-McRO is able to obtain optimum design solutions more efficiently when compared to a previous approach.

The new AA-McRO approach converts “consistency constraints” in the subsystem optimization problems into penalty terms which are integrated into the objective function of the subsystem optimization subproblems. These penalty terms allow the

system and subsystem optimization subproblems to explore the design space better. As the optimization proceeds, the penalized value is minimized so that eventually the consistency constraints are satisfied.

Finally, AA-McRO employs an online approximation technique to reduce the number of function calls. An online verification of the estimated optimum solution is integrated such that the absolute error of the objective and constraint functions can be kept within a user specified threshold. In this way, AA-McRO can significantly reduce the computational effort compared to the previous McRO while obtaining reasonably accurate optimum solutions.

Thrust 3: Integration of Business and Engineering Decisions in an Oil Refinery

The third thrust is focused on developing a framework for integrating engineering and business decisions in an oil refinery using AA-MORO. Several research questions are explored in this thrust in the context of an oil refinery application. These include: (i) how to develop a business model that includes strategic management decisions and at the same time accounts for engineering objectives and constraints; (ii) how to determine the relative importance and effects of uncertain system and/or subsystem input parameters on subsystem and/or system outputs (e.g., system performance); and (iii) how to develop different case study scenarios to demonstrate the integrated decision support framework and understand the impact of business and engineering decisions on the refinery key performance indicators.

1.2 Outline of the Dissertation

The dissertation is organized as follows. The background and terminology are introduced in Chapter 2, followed by the three research thrusts in Chapter 3 to

Chapter 5. The conclusions, contributions and suggested future directions are presented in Chapter 6.

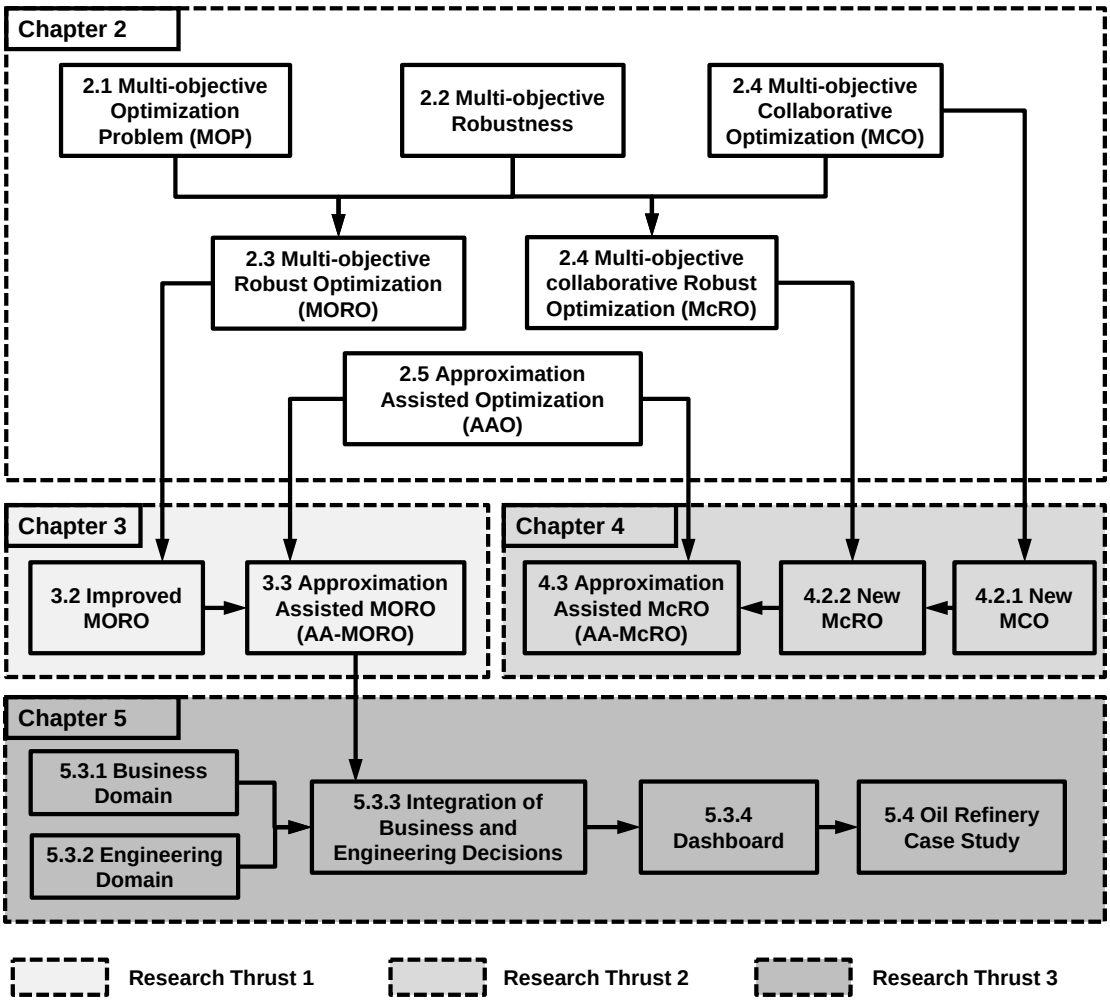


Fig. 1.1 The organization of Chapter 2-5 of this dissertation

As shown in Fig 1.1, following an overview of a Multiobjective Optimization Problem (MOP) in Section 2.1, Chapter 2 includes an introduction of multiobjective robustness in Section 2.2. Combining a MOP with the robustness evaluation, Section 2.3 reviews Multiobjective Robust Optimization approach. Multiobjective Collaborative Optimization (MCO) and Multiobjective collaborative Robust

Optimization (McRO) are reviewed in Section 2.4. Section 2.5 presents a review of Approximation Assisted Optimization (AAO).

The first research thrust, i.e., Approximation Assisted MORO (AA-MORO) is presented in Chapter 3. First an improved MORO approach is developed in Section 3.2. Based on the improved MORO and the AAO, the AA-MORO approach is presented in Section 3.3. Chapter 4 focuses on the second research thrust in which the new MCO and new McRO approaches are presented in Section 4.2. Based on the new McRO approach and online approximation, the AA-McRO framework is presented in Section 4.3. In the third research thrust, the AA-MORO is used to integrate business and engineering decisions by way of a decision support system. The integration framework is presented in Section 5.3. An oil refinery case study is developed in Section 5.4 to demonstrate the decision support system with a dashboard user interface. Finally, in Chapter 6, the conclusions, contributions and suggested future research directions are provided.

Chapter 2: Background

This chapter provides the technical background and terminology related to the main research thrusts of this dissertation.

An overview of a Multiobjective Optimization Problem (MOP) is provided in Section 2.1, including a definition for a MOP and the concept of non-dominated points for multiobjective optimization in subsection 2.1.1. Next, a brief introduction for MultiObjective Genetic Algorithm (MOGA) is presented in subsection 2.1.2. In Section 2.2, the basic idea of objective and feasibility robustness is provided. First a definition of robustness for a single-objective optimization problem is presented, followed by the definition of multiobjective robustness. Section 2.3 reviews the previous bi-level MultiObjective Robust Optimization (MORO) approach (Li et al. 2006) and the steps in the previous MORO approach. A review of the previous Multiobjective Collaborative Optimization (MCO) and Multiobjective collaborative Robust Optimization (McRO) is provided in Section 2.4. In Section 2.5 an overview of a generic Approximation Assisted Optimization (AAO) technique is presented, which includes design of experiment, metamodeling and verification. Finally, a summary of the chapter is provided in Section 2.6.

2.1 Multiobjective Optimization Problem (MOP)

2.1.1 Definition

A general formulation for a MOP is given in Eq. (2.1).

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{f}(\mathbf{x},\mathbf{p}) \\ \text{s.t.} \ & \mathbf{g}(\mathbf{x},\mathbf{p}) \le 0 \\ & \mathbf{x} \in [\mathbf{x}^l, \mathbf{x}^u] \end{aligned} \tag{2.1}$$

where **x** and **p** represent the variables and parameters, respectively. Variables can be changed by an optimizer, while parameters are fixed during an optimization run. A bold letter such as **x** denotes a row vector, i.e., $\mathbf{x} = (x_1, x_2, \ldots, x_{nx})$ where *nx* is the total number of design variables. **f** and **g** represents the real-valued objective functions and constraint functions, respectively. In Eq. (2.1), the superscripts *l* and *u* in the variable **x** represent the lower and upper bounds, respectively. However, **p** (and even **x**) can have uncertainty -- more on this in Section 2.2. It is assumed that variables, parameters, objective and constraint functions are all real-valued.

The feasible domain in Eq. (2.1), denoted by Ω, consists of a set of points that satisfy all constraints. For a MOP where the objective functions are to be minimized and at least partly conflicting, as in Eq. (2.1), a point $\mathbf{x}_1$ is said to multiobjectively dominate $\mathbf{x}_2$, if $\mathbf{f}(\mathbf{x}_1) \le \mathbf{f}(\mathbf{x}_2)$ for all objective functions with strict inequality holding for at least one objective function (Deb, 2001). A solution point $\mathbf{x} \in \Omega$ is non-dominated if there does not exist another solution point $\mathbf{y} \in \Omega$ that dominates it. A non-dominated set Ψ, or Pareto set/frontier, is defined by the set: {$\mathbf{x} \in \Psi$ | there does not exist $\mathbf{y} \in \Omega$ such that **y** dominates **x**}.

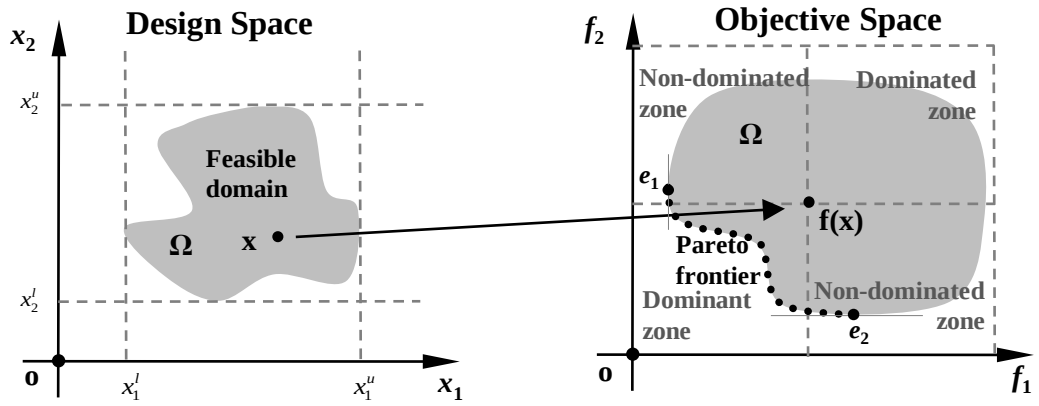


Fig. 2.1 Pareto dominance in MOP with two design variables

Fig. 2.1 shows the basic idea of Pareto dominance in a MOP with two design variables. The feasible domain is determined based on the constraints and the lower/upper bounds of design variables. This is illustrated in the design space on the left of Fig. 2.1. The feasible domain in design space is mapped to the objective space on the right of Fig. 2.1, where the set of points within the closed curve (grey area) represent the feasible domain. With respect to a design point **x**, the objective space is divided into three zones namely, the dominated zone, the dominant zone and non-dominated zone. It can be seen that all points inside the dominated zone are dominated by point **x** and all point inside the dominant zone dominate point **x**. However, any point that falls inside the non-dominated zone is said to be non-dominated to point **x**. Fig. 2.1 also shows the Pareto frontier (which is also a non-dominated set) in the objective space, where e_1 and e_2 represent the two end points of the Pareto frontier.

One way to measure the relative goodness of Pareto frontier is by using the quality metrics (Wu and Azarm, 2001). The Hyperarea Difference (HD) and Overall Spread (OS) are the two quality metrics calculated based on a set of non-dominated points.

As shown in Fig. 2.2, HD is represented by the shaded area based on a definition of a good point (P_{good}) and a bad point (P_{bad}) in the objective space. Because HD measures the closeness of the non-dominated points to a good point, the smaller the HD value the better. On the other hand, OS is defined as the ratio between the rectangle area bounded by the two extreme points of the non-dominated points {a-f-e-m} to the rectangle area bounded by the good and bad points. Since OS measures the spread of the set of non-dominated points, the larger the value of OS is the better the spread of the non-dominated points.

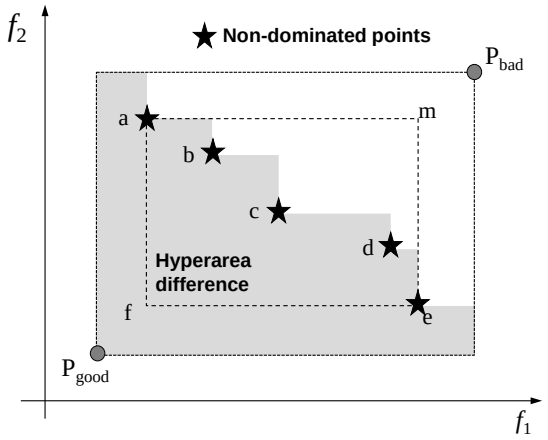


Fig. 2.2 Quality metrics for a set of non-dominated points

To obtain the Pareto frontier for a MOP, many methods are reported in the literature (e.g., Miettinen 1999). In many of these methods, the solutions for a MOP are obtained based on the idea of dominance which distinguishes between dominated and non-dominated solutions. One such method is a MultiObjective Genetic Algorithm (MOGA) (Deb, 2001) which is an evolutionary (meta-heuristic) approach, as overviewed next.

2.1.2 MultiObjective Genetic Algorithm (MOGA)

MOGA is basically a Genetic Algorithm (GA) (Holland, 1975), which is a meta-heuristic with provisions for multiobjective dominance. It operates on a population of design points (or chromosomes). The fitness of each point in MOGA is a measure of performance of that point as defined by the objective and constraint functions. MOGA basically consists of three parts: (1) generating a population of design points; (2) evaluating the fitness of each design point based on multiple objectives (and constraints); and (3) applying genetic operators to generate the next generation of design points. Among these, the first and third parts in MOGA are essentially the same as those used in a single-objective GA, see, e.g., Goldberg (1989) and Fonseca and Fleming (1993). The fitness of each design point in MOGA is evaluated by performing a sorting algorithm based on the value of the objective functions.

A commonly used sorting algorithm is Non-Dominated Sorting (NDS) (Deb, 2001). NDS is based on a population of design points and works as follows for an unconstrained MOP. First, a non-dominated set Φ_1 for the population is determined based on their objective function values. All design points in Φ_1 are assigned to have the first (or highest) rank. From the remaining design points, the set of non-dominated points Φ_2 is determined and its members are assigned to the second rank. This procedure is repeated, until the entire population is divided into partitions (or sets) $\Phi_1, \Phi_2, \ldots, \Phi_{ns}$. Members (or points) in each of these sets are assigned to ranks 1, 2, . . . *ns*. Obviously, there can be more than one element in each set. In order to establish a distinctive ranking among the elements of a particular set, crowding distance sorting can be used (Coello et al., 2002). Using a crowding distance sorting,

individuals that contribute more to the diversity are assigned higher ranks. In addition to NDS, other sorting schemes (e.g., Fonseca and Fleming, 1993; Knowles and Corne, 1999) can be used for fitness assignment in MOGA.

Finally, a simple penalty method can be integrated with MOGA to handle constraints. Using the penalty method, infeasible points are penalized by considering a large positive value (for constraints in Eq. (2.1)) to their original fitness values. Typically, the penalty value is proportional to a constraint function value. Since the constraint values for infeasible points are positive, as in Eq. (2.1), a highly infeasible point which results in a higher constraint value is penalized more than a less infeasible point. Other constraint handling techniques (e.g., Kurpati et al., 2002; Qu and Suganthan, 2011) can also be used. In general, these constraint handling approaches may consider constraint function values, number of constraint violations and other appropriate measures during the fitness assignment stage of MOGA.

Next, the definition of multiobjective robustness is provided in Section 2.2 including the basic ideal of robustness with interval uncertainty for a single objective and single constraint function (Section 2.2.1), followed by a description of the concept of multiobjective and feasibility robustness (Section 2.2.2).

2.2 Multiobjective Robustness

In this section, the main concepts and approaches behind robustness for a MOP under interval uncertainty are reviewed and discussed.

2.2.1 Robustness with Interval Uncertainty: Basic Idea

It is assumed that, at a candidate design point **x**, the parameter **p** (i.e., each of its elements) has a nominal value. Let $\Delta\mathbf{p}$ represents the interval uncertainty around the nominal value of **p**. The uncertainty range is presumed to be known as: $\Delta\mathbf{p} \in [\Delta\mathbf{p}^l, \Delta\mathbf{p}^u]$. Note that this definition of uncertainty in the parameters can be easily extended to the design variable **x**. The uncertainty in the parameters is transmitted to the objective and constraint functions, i.e., **f**(**x**, **p**) and **g**(**x**, **p**), which are then varied from their nominal value due to the transmitted uncertainty. This variation can be undesirable. For example, the optimum value of objective functions may be degraded significantly due to the variation in **f**(**x**, **p**). Or an optimum design can become infeasible due to the transmitted uncertainty in **g**(**x**, **p**). To address this, the concept of robust design is considered and integrated in the optimization procedure (Taguchi, 1987). The basic idea is for the optimizer to search the design space and identify design solutions which are relatively insensitive (or robust) to uncertainty, in terms of change in the objective function and feasibility of the design. To better understand the concept of robust design, consider the plots for a one-variable objective function *f* and constraint *g* as shown in Fig. 2.3. The goal is to minimize *f* subject to $g \leq 0$ with interval uncertainty, as shown by the grey area along the *x* axis. In Fig. 2.3(a), points A, B and C represent the three local optimum points. The objective value at point A is rather sensitive to the uncertainty in the design variable, as represented by the variation or change of the function along the *f* axis. On the other hand, compared to point A, point C is less sensitive or relatively insensitive (objectively robust) to the interval uncertainty and that the variation of objective function for point C is the

smallest among the three points. In terms of constraint function values, points A and B are immune to infeasibility (or are feasibly robust) because the constraint function value at both points are still negative when there is uncertainty in *x*. However, point C may become infeasible when a positive variation occurs in *x*, as shown in Fig. 2.3(b). Therefore, considering the objective/feasibility robustness and optimality, point B is preferred (optimum and robust) among these points. Note that while in this illustrative example the robust optimum solution is nominally (locally) optimum, in general, this may not be the case.

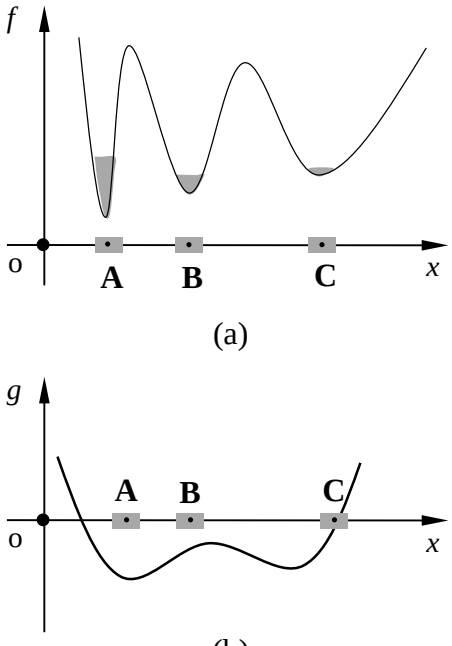


Fig. 2.3 Basic ideal of robustness (a) objective robustness (b) feasibility robustness

2.2.2 Multiobjective and Feasibility Robustness

In a MOP problem, both multiobjective robustness and feasibility robustness are considered. Similar to the single objective example discussed in Section 2.2.1, multiobjective robustness can be defined such that variation or uncertainty in each

objective function, as a result of input uncertainty, does not exceed an Acceptable Objective Variation Range (AOVR). Also, feasibility robustness can be defined in which the variation in constraint functions maintains the feasibility of a design point. Because the objective functions in a MOP are minimized as defined in Eq. (2.1), it is undesirable that the variation in any objective function increases its value beyond AOVR. Let the vector $\Delta\mathbf{f}^+$ denotes the amount of increase (with superscript + for an increase) in the value of the objective functions, which can be formulated as:

$$\begin{aligned}\Delta\mathbf{f}^+ &= [\mathbf{f}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) - \mathbf{f}(\mathbf{x},\mathbf{p})]^+ \\ &= \begin{cases}\mathbf{0}, \text{ if } \mathbf{f}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) \le \mathbf{f}(\mathbf{x},\mathbf{p}) \\ \mathbf{f}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) - \mathbf{f}(\mathbf{x},\mathbf{p}), \text{ otherwise}\end{cases}\end{aligned} \tag{2.2}$$

where $\mathbf{f}(\mathbf{x}, \mathbf{p}+\Delta\mathbf{p})$ and $\mathbf{f}(\mathbf{x}, \mathbf{p})$ represent the actual (due to uncertainty from $\mathbf{p}$) and nominal values for the objective functions, respectively. $\Delta\mathbf{f}^+$ is called the objective variation vector whose elements represent the increase (if any) in each objective function as a result of input uncertainty. To measure the variation in all objectives with a scalar, the Euclidean norm $\|\Delta\mathbf{f}^+\|$ is used. In solving a multiobjective optimization problem under interval uncertainty, it is important to obtain solutions which are not only optimal but also have acceptable increases in all of their objectives as a result of uncertainty. It is presumed that the decision maker can specify a positive scalar value of η_f as an AOVR for the objective variation value, such that the maximum of the norm of the objective variation vector is smaller than or equal to η_f:

$$\max_{\Delta\mathbf{p}} \|\Delta\mathbf{f}^+\| \le \eta_f \tag{2.3}$$

In Eq. (2.3), $\|\Delta\mathbf{f}^+\| = (\sum_{m=1}^{nf} \Delta f_m^2)^{1/2}$ where m and nf denote the index and the total number of objective functions, respectively. Since $\Delta\mathbf{f}^+$ represents an increase in multiple

objectives, the inequality constraint in Eq. (2.3) is referred to as a multiobjective robustness constraint. Any design point that satisfies Eq. (2.3) is considered to be a multiobjectively robust design point. A two dimensional example of multiobjective robustness with interval uncertainty is shown in Fig. 2.4. The grey area on the left represents the uncertainty variation region. Any point inside this variation region corresponds to a realization of uncertainty. For example, point o in the left figure is a nominal point (x, p), and points u and v each represents a realization of an uncertain **p** value. The uncertainty variation region can be mapped to the objective space where the set of points inside the closed curve (grey shaded) area represents the objective variation region.

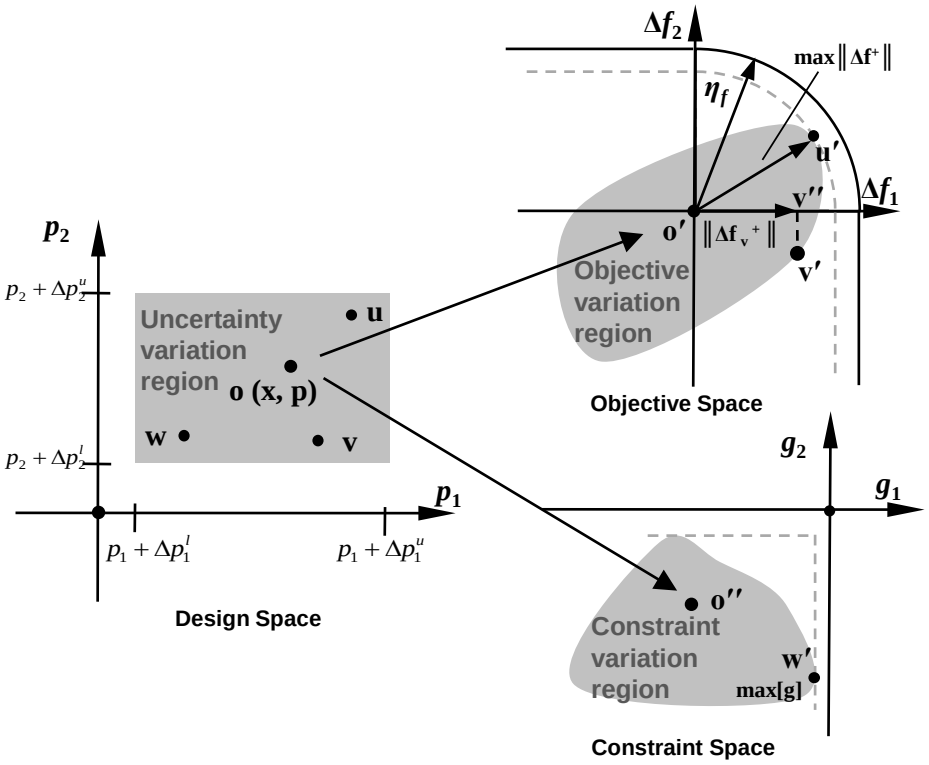


Fig. 2.4 Multiobjective robustness hypothetical case

In the objective space, the point o′ represents the nominal value for the objective functions, i.e., **f**(**x**, **p**). It can be seen that the Euclidean norm of the objective variation vector $\|\Delta\mathbf{f}^+\|$ is the distance from the nominal point o′ to a point in the objective space. Since the only concern is with an increase in the objective function values, any point in the third quadrant in the mapped objective variation region is acceptable from an objective robustness point of view. Therefore, the third quadrant points are ignored when calculating the Euclidean norm of the objective variation. However, to calculate the Euclidean norm of the objective variation for points in the second and forth quadrant in the objective space, those points are to be projected on to the positive axis. For example, to calculate the objective variation for point v, it is first projected from the design space to the objective space, as represented by point v′, which is in the fourth quadrant. Then, point v' is again projected to the positive axis in the objective space to point v″. The Euclidean norm of the objective variation for point v, i.e., $\|\Delta\mathbf{f}_v^+\|$, is represented by the horizontal line segment along the Δf_1 axis.

With the projection of the objective variation region in the second and forth quadrant, the maximum Euclidean norm of the objective variation vector $\max\|\Delta\mathbf{f}^+\|$ is the distance from the nominal point o′ to the furthest point u′ in the first quadrant (including the projected line segments on the positive axis). This value is essentially the radius of the smallest quarter-circle (shown in dashed line) that encloses the objective variation region in the first quadrant. According to the multiobjective robustness requirement in Eq. (2.3), this maximum value must not exceed the acceptable limit η_f. The geometrical interpretation is that the dash-lined quarter-circle must be enclosed within the solid-lined quarter-circle whose radius is equal to η_f.

Fig. 2.4 shows the case in which the furthest point on the projected objective variation region is located in the first quadrant in the objective space. Since the dashed circle is within the solid circle and $\max \left\| \Delta \mathbf{f}^{+} \right\| \leq \eta_f$ is satisfied, the candidate design point is said to be multiobjectively robust.

Similarly, the feasibility robustness is defined with the following inequality constraint as:

$$\max_{\Delta \mathbf{p}} [\max_{k} \mathbf{g}(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p})] \leq 0 \tag{2.4}$$

The first maximization in Eq. (2.4) is conducted with respect to $\Delta \mathbf{p}$, and the second maximization inside the square bracket is with respect to *k*, where *k* represents the index for the constraint function number. Note that $\mathbf{g} = (g_1, g_2, \ldots, g_{ng})$ where *ng* represents the total number of constraint functions.

For simplicity, from this point on the term "max[**g**]" is used to refer to the left-hand side of Eq. (2.4). Note that for feasibility robustness, the decision maker is concerned with the feasibility of a point (i.e., $\mathbf{g} \leq \mathbf{0}$) due to uncertainty in the design variables and parameters. As shown by the feasibility robustness constraint, Eq. (2.4), max[**g**] represents the worst-case constraint value, which should be less than or equal to zero in order to ensure feasibility. Consequently, any candidate design point that satisfies Eq. (2.4) is called a feasibly robust design point.

The geometric interpretation of Eq. (2.4) can be explained by Fig. 2.4. As in multiobjective robustness, the uncertainty variation region is mapped to the constraint space on the right-hand side where the grey shaded area represents the constraint variation region. Point o″ is the projected point for the nominal design point o. And point w′ corresponds to max[**g**] which is obtained by searching for the closest point

on the constraint variation region to any constraint axis. The feasibility robustness as required by Eq. (2.4) indicates that the entire constraint variation region must be located in the third quadrant of the constraint function space.

It must be noted that the definition of multiobjective and feasibility robustness are derived from the previous work (Gunawan and Azarm, 2005; Li et al., 2006). However, these earlier works focus on both positive and negative variations around a nominal design point for objective robustness. The one-sided objective robustness presented here is more applicable to engineering design applications where the designer is only concerned with the variation which degrades the optimum design (increasing objective function value). For example, a downside variation (or decrease) in cost (a typical objective) implies a cost reduction which is desirable.

2.3 MultiObjective Robust Optimization (MORO)

MORO is formulated as a bi-level optimization problem with an upper-level problem and two lower-level subproblems (Li et al., 2006). In the upper-level problem, the optimizer searches the design space for candidate feasible design points which optimize the objectives; while in the lower level, a single-objective global optimizer (such as genetic algorithm) evaluates multiobjective and feasibility robustness of each candidate design point. The formulation for the upper-level problem and lower-level subproblems is given in Eqs. (2.5) and (2.6) as follows:

Upper-level problem:

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{f}(\mathbf{x},\mathbf{p}) \\ &\text{s.t.} \ \ \max\left\|\Delta\mathbf{f}^{+}\right\| \le \eta_f \\ &\qquad \max\left[\mathbf{g}\right] \le 0 \\ &\qquad \mathbf{x} \in [\mathbf{x}^{l}, \mathbf{x}^{u}] \end{aligned} \tag{2.5}$$

Lower-level subproblems:

$$\max \left\| \Delta \mathbf{f}^{+} \right\| = \max_{\Delta \mathbf{p}} \left\| [\mathbf{f}(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p}) - \mathbf{f}(\mathbf{x}, \mathbf{p})]^{+} \right\| \quad \text{s.t. } \Delta \mathbf{p} \in [\Delta \mathbf{p}^{l}, \Delta \mathbf{p}^{u}] \tag{2.6-1}$$

$$\max[\mathbf{g}] = \max_{\Delta \mathbf{p}} [\max_{k} \mathbf{g}(\mathbf{x}, \mathbf{p} + \Delta \mathbf{p})] \quad \text{s.t. } \Delta \mathbf{p} \in [\Delta \mathbf{p}^{l}, \Delta \mathbf{p}^{u}] \tag{2.6-2}$$

The upper-level problem is formulated as a multiobjective optimization problem as in Eq. (2.1), except that the multiobjective and feasibility robustness constraints as defined in Section 2.2 are added. Notice that in Eq. (2.5) the left-hand side of the inequalities of multiobjective and feasibility robustness constraints, i.e., $\max \left\| \Delta \mathbf{f}^{+} \right\|$ and max[**g**] must be evaluated in the lower-level subproblems. As shown in Eqs. (2.6-1) and (2.6-2), the lower-level includes two single-objective optimization subproblems where the value for the nominal design, denoted by **x**, is fixed. Essentially, the first optimization subproblem obtains the maximum Euclidean norm of increase in the objective vector, i.e., $\max \left\| \Delta \mathbf{f}^{+} \right\|$, and the second optimization subproblem obtains the worst-case constraint value, i.e., max[**g**].

Solving a MORO problem, Eq. (2.5) and (2.6), using a population based approach, such as MOGA for the upper-level and GA for the lower-level, works as follows (see, also, Li et al., 2006):

First a set of candidate design points are generated in the upper-level problem. In order to assign fitness to each point, the objective and constraint functions of each point are calculated. Since the multiobjective robustness and feasibility robustness constraint in Eq. (2.5) needs to be evaluated in the lower-level subproblems, the nominal value of a current point (from the population), as denoted by a vector **x**, is passed on to the lower-level. Once the lower-level subproblems receive **x**, the single-objective optimization problems in Eqs. (2.6-1) and (2.6-2) are solved with respect to

$\Delta\mathbf{p}$. The optimal value from the two subproblems, i.e., $\max \left\| \Delta \mathbf{f}^{+} \right\|$ and $\max[\mathbf{g}]$ are obtained and returned to the upper-level problem to complete constraint evaluation for the current point **x**. The same procedure (solving the two lower-level subproblems) is repeated for all other candidate design points. Next, all current candidate design points are ranked based on their values of objective and constraint functions (including multiobjective and feasibility robustness constraints). Consequently the fitness of each candidate point is determined based on which the non-dominated points are determined. Depending on the multiobjective optimization approach being used, the previous procedure typically is repeated until a set of non-dominated solutions (which are multiobjectively and feasibly robust and optimum) are obtained.

As can be seen from the procedure above for MORO, the two lower-level subproblems are essentially "nested" within the upper-level problem. As the upper-level problem considers each candidate design point, the two lower-level subproblems have to be solved to evaluate the maximum Euclidean norm of increase in the function and the worst-case constraint value. This bi-level procedure requires considerable computational effort. Using a population-based approach such as MOGA and GA, the computational effort needed by MORO to arrive at a solution grows exponentially as the size (number of variables) of the problem and therefore the number of points in the population increases. An improved MORO approach is presented in Chapter 3 in this dissertation which is more efficient than the previous MORO approach (Li et al., 2006).

2.4 Multiobjective Collaborative Optimization (MCO) and Multiobjective collaborative and Robust Optimization (McRO)

The optimization models presented earlier in this chapter including those in Section 2.1 and Section 2.3 focused on a single discipline all-in-one formulation. In this section, two multiobjective multi-disciplinary optimization approaches are presented. Section 2.4.1 reviews a previous deterministic MCO approach (Aute and Azarm, 2006). Then, a brief review of the McRO approach (Li and Azarm, 2008) is presented in Section 2.4.2.

2.4.1 MCO Formulation

Fig. 2.5 shows the schematic of a MCO framework where both system and subsystem problems are characterized by a MOP. The system problem at the upper level is to achieve multiobjectively optimum system design solutions. At the same time the system problem also coordinates the shared variables ($\mathbf{x}_{sh}$) and coupling variables ($\mathbf{y}_{ij}$) to guide the subsystem problems at the lower level.

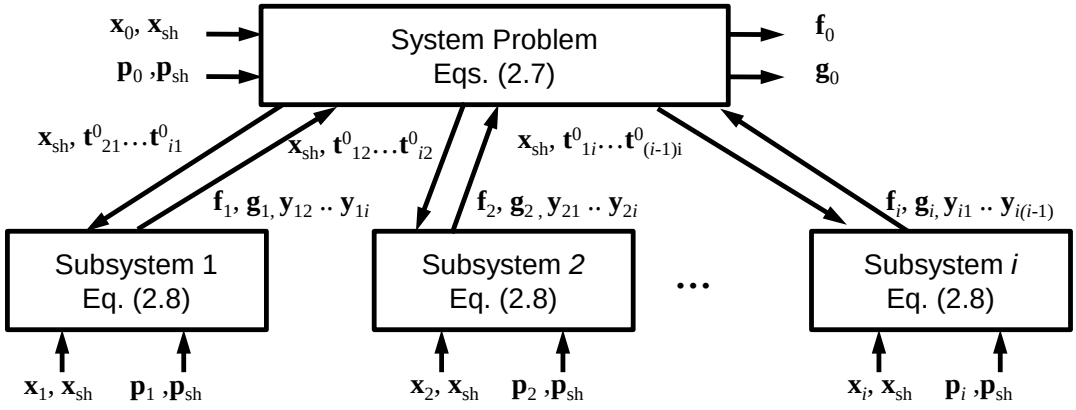


Fig. 2.5 Schematic of bi-level MCO framework

A general MCO problem with a two-level formulation is shown in Eqs. (2.7) and (2.8). The system optimization problem in Eq. (2.7) minimizes the system design objectives subject to system constraints, while a series of I subsystem-level optimization problems in Eq. (2.8) each minimizes the subsystem local design objectives subject to subsystem constraints.

System problem (upper level):

$$\begin{aligned} \min_{\mathbf{X}_0} \quad & \mathbf{f}_0(\mathbf{X}_0, \mathbf{P}_0, \mathbf{f}_i) \quad i=1,...I \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{X}_0, \mathbf{P}_0, \mathbf{g}_i) \le 0 \quad i=1,...I \\ & \left\| \mathbf{y}_{ij} - \mathbf{t}_{ij}^0 \right\| \le \varepsilon_i; \;\; i, j=1,...I; j \ne i \\ & \mathbf{X}_0 \in [\mathbf{X}_0^l, \mathbf{X}_0^u] \\ & \mathbf{X}_0 \equiv [\mathbf{x}_0, \mathbf{x}_{\text{sh}}^0, \mathbf{t}_{ij}^0]; \; \mathbf{P}_0 \equiv [\mathbf{p}_0, \mathbf{p}_{\text{sh}}] \end{aligned} \tag{2.7}$$

Subsystem problems (lower level) i=1,…,I :

$$\begin{aligned} \min_{\mathbf{X}_i} \quad & \mathbf{f}_i(\mathbf{X}_i, \mathbf{P}_i) \\ \text{s.t.} \quad & \mathbf{g}_i(\mathbf{X}_i, \mathbf{P}_i) \le 0 \\ & \left\| \mathbf{x}_{\text{sh}}^i - \mathbf{x}_{\text{sh}}^0 \right\| \le \varepsilon \\ & \left\| \mathbf{y}_{ij} - \mathbf{t}_{ij}^0 \right\| \le \varepsilon_i; i, j=1,...I; j \ne i \\ & \left\| \mathbf{t}_{ji}^i - \mathbf{t}_{ji}^0 \right\| \le \varepsilon_j; \; i, j=1,...I; j \ne i \\ & \mathbf{y}_{ij} = Y_{ij}(\mathbf{X}_i, \mathbf{P}_i) \\ & \mathbf{X}_i \in [\mathbf{X}_i^l, \mathbf{X}_i^u] \\ & \mathbf{X}_{\text{i}} \equiv [\mathbf{x}_i, \mathbf{x}_{\text{sh}}^i, \mathbf{t}_{ji}^i]; \; \mathbf{P}_i \equiv [\mathbf{p}_i, \mathbf{p}_{\text{sh}}] \end{aligned} \tag{2.8}$$

where $\mathbf{f}_0$ and $\mathbf{g}_0$ represent the system design objectives and constraints, respectively. In Eq. (2.7), The system problem's design variables $\mathbf{X}_0$ include system variables $\mathbf{x}_0$, shared variables $\mathbf{x}^0{}_{\text{sh}}$ and target variables $\mathbf{t}^0{}_{ij}$, with a superscript "0" referring to the system. $\mathbf{P}_0$ includes system parameters $\mathbf{p}_0$, and shared parameters $\mathbf{p}_{\text{sh}}$. Likewise, in the i^{th} subsystem problem (i = 1,…, I), $\mathbf{f}_i$ and $\mathbf{g}_i$ represent the subsystem objectives and constraints, respectively. $\mathbf{X}_i$ includes subsystem variables $\mathbf{x}_{\text{i}}$, shared variables $\mathbf{x}^i{}_{\text{sh}}$ and target variables $\mathbf{t}^i{}_{ji}$. $\mathbf{P}_i$ includes subsystem's parameters $\mathbf{p}_i$ and shared parameters $\mathbf{p}_{\text{sh}}$. Note that both the system target variables $\mathbf{t}^0{}_{ij}$ and subsystem target variables $\mathbf{t}^j{}_{ij}$

are specified to match the coupling variables $\mathbf{y}_{ij}$. The subscript ij of a coupling variable $\mathbf{y}_{ij}$ indicates the coupling variable $\mathbf{y}$ is computed as an output from subsystem i and then used as input in subsystem j. The subscript of a target variable has the same definitions.

In the i^{th} subsystem, the coupling variable $\mathbf{y}_{ij}$ is a function (represented by Y_{ij}) of the subsystem variables and parameters as shown in Eq. (2.8). Because of the couplings between subsystems, there is a consistency constraint in Eq. (2.8) to ensure that a coupling variable matches a target variable. There, the deviation of coupling variable $\mathbf{y}_{ij}$ from the system-level target variables $\mathbf{t}_{ij}$ (represented by the Euclidean norm: $\|\mathbf{y}_{ij} - \mathbf{t}_{ij}\|$) is constrained and will not exceed a pre-specified tolerance ε_i. A second consistency constraint in Eq. (2.8) is used to ensure the deviation of subsystem target variable $\mathbf{t}^i_{ji}$ from the system-level target variables $\mathbf{t}_{ji}$ (represented by $\|\mathbf{t}^i_{ji} - \mathbf{t}^0_{ji}\|$) does not exceed a tolerance ε_j.

The goal in solving Eq. (2.7) is to obtain a set of solutions for the system optimization problem while simultaneously coordinating the optimization of the subsystem optimization problems in Eq. (2.8).

2.4.2 Multiobjective Collaborative Robustness

A parameter in a MCO problem, as shown in Eqs. (2.7) and (2.8), can be characterized by a nominal value $\mathbf{P}$ and an interval uncertainty $\Delta\mathbf{P}$ around nominal, i.e., $\mathbf{P}+\Delta\mathbf{P}$. It is assumed that the upper and lower bounds of the interval uncertainty is known a priori such that $\Delta\mathbf{P} \in [\Delta\mathbf{P}^l, \Delta\mathbf{P}^u]$. This definition of uncertainty can be extended to design variables $\mathbf{X}$ by adding their corresponding interval uncertainties

$\Delta\mathbf{X}$ to their nominal. Similar to the definition in Section 2.2, let $\Delta\mathbf{f}^+$ denotes an increase in the value of objective function in subsystem *i*, it can be formulated as:

$$\begin{aligned}\Delta\mathbf{f}_i^+ &= [\mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)-\mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i)]^+ \\ &= \begin{cases}\mathbf{0}, & \text{if } \mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)\le\mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i)\\ \mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)-\mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i), & \text{otherwise}\end{cases}\end{aligned} \tag{2.9}$$

where $\mathbf{f}_i(\mathbf{X}_i,\ \mathbf{P}_i+\Delta\mathbf{P}_i)$ and $\mathbf{f}_i(\mathbf{X}_i,\ \mathbf{P}_i)$ represent the actual (due to uncertainty) and nominal values for the objective functions in subsystem *i*. To measure the variations in all objectives in the *i*th subsystem using a scalar value, a Euclidean norm $\|\Delta\mathbf{f}_i^+\|$ is used, where $\left\|\Delta\mathbf{f}_i^+\right\|_2=(\sum_{m=1}^{nf_i}\Delta f_{i,m}^2)^{1/2}$ and nf_i represent the total number of objective functions in subsystem *i*. As described below, to quantify the variation in the objective functions considering all subsystems, the maximum value is selected. Similarly, the maximum variations for the coupling variables is calculated except that both positive and negative variations in a coupling variable are considered, i.e., $\Delta\mathbf{y}_i=\mathbf{y}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)-\mathbf{y}_i(\mathbf{X}_i,\mathbf{P}_i)$. And finally the maximum value of constraint functions is calculated as defined earlier in Section 2.2. These are shown in Eqs. (2.10-1)-(2.10-3):

Robustness evaluation problems (lower level)

$$\begin{aligned}&\max\left\|\Delta\mathbf{f}^+\right\|=\max_{\Delta\mathbf{P}_i}[\max_i\left\|[\mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)-\mathbf{f}_i(\mathbf{X}_i,\mathbf{P}_i)]^+\right\|],i=1,...,I\\ &\text{s.t.}\quad \Delta\mathbf{P}_i\in[\Delta\mathbf{P}_i^l,\Delta\mathbf{P}_i^u]\end{aligned} \tag{2.10-1}$$

$$\begin{aligned}&\max\left\|\Delta\mathbf{y}\right\|=\max_{\Delta\mathbf{P}_i}[\max_i\left\|\mathbf{y}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)-\mathbf{y}_i(\mathbf{X}_i,\mathbf{P}_i)\right\|],i=1,...,I\\ &\text{s.t.}\quad \Delta\mathbf{P}_i\in[\Delta\mathbf{P}_i^l,\Delta\mathbf{P}_i^u]\end{aligned} \tag{2.10-2}$$

$$\begin{aligned}&\max[\mathbf{g}]=\max_{\Delta\mathbf{P}_i}\{\max_i[\max_{k_i}\mathbf{g}_i(\mathbf{X}_i,\mathbf{P}_i+\Delta\mathbf{P}_i)]\},i=1,...,I\\ &\text{s.t.}\quad \Delta\mathbf{P}_i\in[\Delta\mathbf{P}_i^l,\Delta\mathbf{P}_i^u]\\ &\mathbf{X}_i=[\mathbf{x}_i,\mathbf{x}_{\text{sh}},\mathbf{t}_{ji}^i];\quad \mathbf{P}_i=[\mathbf{p}_i,\mathbf{p}_{\text{sh}},\mathbf{t}_{ji}^i]\end{aligned} \tag{2.10-3}$$

Eqs. (2.10-1)-(2.10-3) are collectively called the "robustness evaluation problems" for a MCO problem under uncertainty. Eq. (2.10-1) represents the maximum variation in the objective functions, where the first max is performed over all realizations of $\Delta\mathbf{P}_i$ in the interval $[\Delta\mathbf{P}^l_i, \Delta\mathbf{P}^u_i]$ and the second max is performed with respect to different subsystems. Similarly, the maximum variations in the coupling variables and the maximum constraint function values are calculated using Eqs. (2.10-2) and (2.10-3), respectively. Note that in Eq. (2.10-3), the third maximization is done with respect to k_i where k_i represents the constraint function number index in subsystem i.

Based on the maximum variation values as calculated in Eq. (2.10), the definition of robustness for MCO problem is similar to that in MOP. For example, objective robustness in MCO requires $\max\left\|\Delta\mathbf{f}^+\right\| \leq \eta_f$, in which η_f is a user specified positive scalar. Similarly, the feasibility robustness requires $\max[\mathbf{g}] \leq 0$. However, a new measure of robustness of coupling variables, called collaborative robustness is defined in MCO. Collaborative robustness requires that the amount of variation in the value of coupling variables due to interval uncertainty should remain within an acceptable limit η_y. Notice that because a coupling variable could be an input to another subsystem, allowing a variation range for a coupling variable creates an input uncertainty to another subsystem. Li and Azarm (2008) discuss this issue in detail and develop a measure for the propagation of interdisciplinary uncertainty which is adopted in this dissertation.

Based on the optimum value in the robustness evaluation problems, that is, the value of $\max\left\|\Delta\mathbf{f}^+\right\|$, $\max\left\|\Delta\mathbf{y}\right\|$, and $\max[\mathbf{g}]$, the robustness of a candidate design alternative can be determined using the following robustness conditions:

$$\max\left\|\Delta\mathbf{f}^{+}\right\| \le \eta_f ; \max\left\|\Delta\mathbf{y}\right\| \le \eta_y ; \max[\mathbf{g}] \le 0 \qquad (2.11)$$

A candidate design alternative must satisfy the inequality constraints in Eq. (2.11) to be considered as a robust design. Therefore, Eq. (2.11) is added as constraint functions in the system problem in Eq. (2.7).

2.4.3 Multiobjective collaborative Robust Optimization (McRO)

With the optimization formulation for system, subsystem and robustness evaluation problems defined above, the framework for McRO (Li and Azarm, 2008) under uncertainty is shown in Fig. 2.5. Based on the McRO approach, system optimization as given in Eq. (2.7) is performed at the upper level while the subsystem optimization and robustness evaluations, Eqs. (2.11) and (2.10-1)-(2.10-3), are performed at the lower level. McRO uses MOGA (Deb 2001) to solve both system and subsystem problems Eqs. (2.7) and (2.8). Next, the solutions steps for McRO are briefly presented so that the difficulties in the previous McRO can be highlighted (see Section 2.4.4).

The approach starts with multiobjective optimization at the system-level where each candidate design alternative, as denoted by $\mathbf{x}^0{}_{\text{sh}}$ and $\mathbf{t}^0{}_{ij}$ are sent as parameters to the subsystems. Next, multiobjective optimization is performed in each subsystem as defined in Eq. (2.8), for a given $\mathbf{x}^0{}_{\text{sh}}$ and $\mathbf{t}^0{}_{ij}$ from the system level. Next, the i^{th} subsystem optimizer obtains a set of multiobjective optimal solutions. Then, one optimal solution is selected from each subsystem and returned to the system problem. The system (optimization) problem uses the returned solutions, one from each subsystem, to calculate the objective and constraint function values at the system-level. Next, the system problem forwards the current candidate design alternative,

including system variables, the selected subsystem optimal solution from each subsystem to the robustness evaluation block as shown in Fig. 2.5. Based on Eqs. (2.10-1) to (2.10-3), robustness for the current design alternative is assessed and the results are returned to the system problem. Finally, the system problem assigns a fitness value to the current design alternative based on its system objectives, constraint function values and robustness evaluation results. This completes function evaluation for one candidate design alternative. The approach repeats the above procedure to evaluate a large number of design alternatives until convergence to an optimum Pareto frontier at the system problem.

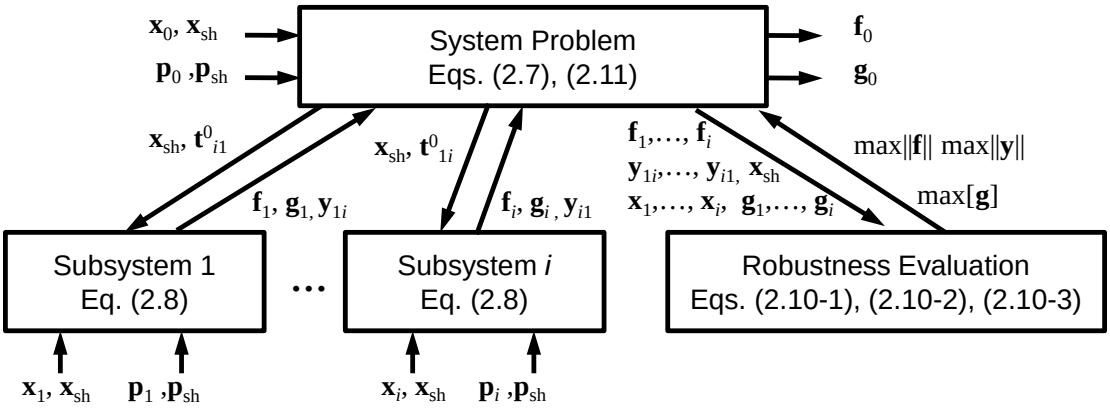


Fig. 2.5 McRO under interval uncertainty

In an optimization problem with a computationally costly analyzer, Approximation Assisted Optimization (AAO) techniques can be used to replace the objective and/or constraint functions with a more compact (or closed form) metamodel (Wang and Shan 2007, Simpson et al., 2001b). In the following sections, the three main components of AAO are reviewed. These are Design of Experiments (DOE) as presented in Section 2.5.1, followed by the Kriging metamodeling technique in

Section 2.5.2. Section 2.5.3 discusses the error calculation formula used for a metamodel verification.

2.5 Approximation Assisted Optimization (AAO)

A typical AAO technique consists of DOE, metamodeling, and verification stages. In DOE, a set of sample points in the design space are generated. These points are then used to construct metamodels for the objective and constraint functions of an optimization problem which is called metamodeling. Based on these metamodels, the optimizer obtains a set of approximated optimum solutions, which should be verified. If the accuracy of these optimum solutions is not acceptable, the metamodels should be refined by adding additional sample points.

The details of the three component of AAO are presented next.

2.5.1 Design of Experiments (DOE)

DOE is aimed at identifying the locations of a set of sample points in the design space. These sample points are then "observed", which means their actual objective/constraint function values are computed. The observed sample points are used later for constructing the metamodels. Most DOE methods for AAO can be categorized as either offline and online. The main difference between these two is that sample points in an offline DOE method are not updated during AAO while they are updated in an online DOE. Examples of offline DOE method can be found in e.g., Lian and Liou (2004) and Li et al., (2010a); while online DOE approach can be referred in e.g., Karakasis et al., (2001) and Hu et al., (2012d). The detail in offline and online DOE is presented next.

In an offline DOE method, the sample points are determined based only on the information from the design space. For example, factorial design is typically used as a DOE method. Although factorial design covers well the boundary of the design space, it fails to locate enough sample points in the middle of the design space. Therefore, space-filling sampling becomes more widely used as DOE method in AAO. One popular space-filling sampling technique is Latin Hypercube Sampling (LHS) proposed by McKay et al. (1979). In LHS, the design space is divided into equal levels along each input variable dimension and only one sample point is placed at each level. Locations of the sample points are randomly determined.

On the other hand, an online DOE method usually consists of two stages. In the first stage, a small portion of the sample points are determined based on an offline DOE method. In the second stage, there is a feedback loop from the optimizer to choose additional sample points. The second stage in an online DOE method is iterative based on the information from the optimizer. In this way, and compared to an offline technique, one significant advantage of online DOE is that the sample points obtained from online DOE is able to progressively improve the predictive capabilities of the metamodels, as more and more sample points are evaluated and added to the sample set.

Because offline DOE does not update sample points, it can be computationally expensive as it may require more sample points to build a globally accurate metamodel for AAO. However, online DOE may require fewer sample points. But one limitation of online DOE is that, with limited number of sample points in its first

stage, the predictive capability of the metamodel is poor. This can mislead the optimization process into sub-optimum or infeasible regions in the design space.

2.5.2 Metamodeling: Kriging

Many existing technique can be used in constructing a metamodel in AAO, such as response surface (Myers and Montgomery, 1995) neural network (Hong et al., 2003), support vector machine (Nakayama et al., 2003), as well as Gaussian-based methods (Buche et al., 2005), Kriging is a widely used metamodeling technique for a black-box engineering function where the response is assumed to be the sum of both global and local models (Koehler and Owen, 1996). Kriging is based on an interpolation of the sample points. This property of Kriging is especially attractive in engineering applications because the response from Kriging on existing sample point is exactly accurate. Furthermore, Kriging is able to provide a confidence interval based on an estimated mean squared error for a predicted response function value. Because of these advantages, Kriging is used as a metamodeling technique in this dissertation.

Let $y(\mathbf{x})$ represents an objective or constraint function that should be approximated; the Kriging model is presented as:

$$y(\mathbf{x}) = \delta + Z(\mathbf{x}) \tag{2.12}$$

where $\mathbf{x}$ is a point (vector) in the sample space, δ captures the overall (global) trend of the Kriging model and $Z(\mathbf{x})$ is used to represent a localized deviation from the global function. $Z(\mathbf{x})$ is defined as the realization of a Gaussian random process with mean zero and variance σ^2. Typically, the sample points are interpolated with the Gaussian random function to estimate the stochastic process. Given a total number of n sample

points, the covariance of $Z(\mathbf{x})$ at two sample points $\mathbf{x}_1$ and $\mathbf{x}_2$ can be expressed as in Eq. (2.13):

$$Cov[Z(\mathbf{x}_1), Z(\mathbf{x}_2)] = \sigma^2 R(\mathbf{x}_1, \mathbf{x}_2) \tag{2.13}$$

where $R(\mathbf{x}_1, \mathbf{x}_2)$ is the Gaussian correlation function between sample points $\mathbf{x}_1$ and $\mathbf{x}_2$. The Kriging predictor is given in Eq. (2.14):

$$\hat{y}(\mathbf{x}) = \hat{\delta} + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\delta}) \tag{2.14}$$

where $\hat{y}$ is the predicted (metamodel) value (predictor) of y and $\hat{\delta}$ is the predicted value of δ which is the expected value of the posterior process. $\mathbf{R}$ is a $n \times n$ matrix whose (i, j) element is Cov[$Z(\mathbf{x}_i)$, $Z(\mathbf{x}_j)$], $\mathbf{r}$ (with prime superscript for transpose) is the vector whose i^{th} element expressed as:

$$r_i(\mathbf{x}) = Cov[Z(\mathbf{x}), Z(\mathbf{x}^i)] \tag{2.15}$$

Based on the Kriging model, the Mean Square Error (*mse*) at an unobserved design point $\mathbf{x}^*$ is given in Eq. (2.16):

$$mse(\mathbf{x}^*) = \sigma^2 (1 + c\mathbf{R}c' - 2cr) \tag{2.16}$$

where c is a row vector of Kriging coefficients, r is a vector of correlations between $\mathbf{x}^*$ and observed sample points. Since r represents the degree of correction between $\mathbf{x}^*$ and the existing points, the value of $2cr$ should be large when an unobserved design point is close to the observed sample points. Therefore the closer $\mathbf{x}^*$ is to the observed sample point, the smaller *mse* ($\mathbf{x}^*$) is.

In the later chapters of this dissertation, Kriging is the primary metamodeling techniques for constructing online approximation for expensive objective and constraint functions in multiobjective optimization.

2.5.3 Verification

Because the optimum solutions from AAO essentially estimate the location of optimum points, the mean squared error as defined in Eq. (2.16), can be used as a measure to verify the accuracy of these estimated solutions. In addition, the accuracy of the Pareto optimum solutions can also be validated independently based on the actual objective/constraint function evaluations of the estimated optimum solutions (e.g., Aute, 2007). For example, the Maximum Absolute Error (MAE) is defined in Eq. (2-17):

$$\text{MAE} = \max(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, ..., |y_{np} - \hat{y}_{np}|) | \tag{2.17}$$

where np represents the total number of Pareto points obtained. y_1, y_2, …, y_{np}, represent the observed values at the Pareto points obtained using the actual function evaluations, and $\hat{y}_1$, $\hat{y}_2$, …, $\hat{y}_{np}$ is the predicted values of the corresponding objective or constraint function based on the metamodels. Generally, smaller error values indicate better accuracy at the optimum solutions from AAO.

2.6 Summary

This chapter gives the background necessary for understanding the remaining chapters of this thesis.

In Section 2.1, a general formulation and important definitions for a Multiobjective Optimization Problem (MOP) is provided. For solving a MOP, a MultiObjective

Genetic Algorithm (MOGA) is presented and the advantage of a MOGA compare to other approaches is briefly discussed. Section 2.2 provides a definition of objective and feasibility robustness in both single- and multiobjective optimization problems. Several important concepts related to the multiobjective robustness are explained with an illustrative figure. A review of the previous bi-level MultiObjective Robust Optimization (MORO) approach and MORO steps is presented in Section 2.3. The computational difficulty with the bi-level MORO approach is discussed. Section 2.4 reviewed the Multiobjective Collaborative Optimization (MCO) and Multiobjective collaborative Robust Optimization (McRO) approaches in the previous literature. The solution steps for both MCO and McRO approaches are provided.

A review of approximation assisted optimization (AAO) is presented in Section 2.5, which include the three main components namely Design of Experiment (DOE), metamodeling and verification. The advantage of limitation of online and offline DOE methods for AAO is discussed. A brief review of Kriging metamodeling technique and an error measurement method are presented.

In the next chapter, the approach for a new approximation assisted MORO is detailed.

Chapter 3: Approximation Assisted MultiObjective Robust Optimization under Interval Uncertainty (AA-MORO)

In this chapter, an Approximation Assisted MultiObjective Robust Optimization (AA-MORO) approach is presented. In this approach, MORO has been improved by Hu et al. (2009)[1] to overcome the computational difficulty experienced in a "previous MORO" approach (Li et. al., 2006). The AA-MORO of this chapter was also presented before in Hu et al. (2011)[2] and Hu et al. (2012c)[3] wherein an approximation technique is used to significantly reduce the number of function calls compared to a previous MORO.

As reviewed in Section 2.3, the previous MORO (Li et. al., 2006) is a bi-level (nested) approach in that candidate design points are identified and iteratively improved in an upper-level problem, while in a lower-level subproblem the robustness of all intermediate points are evaluated. Due to this bi-level structure, the previous MORO requires a large number of function calls for robustness evaluations and as a result the computational cost can become prohibitive. On the other hand, the AA-MORO approach is based on an improved sequential MORO approach, hereinafter called "improved MORO", where multiobjective optimal solutions are first obtained and

---

[1] Hu, W., Li, M., Azarm, S., Al Hashimi, S., Almansoori, A., and Al-Qasas, N., 2009, "Improving Multi-Objective Robust Optimization under Interval Uncertainty Using Worst Possible Point Constraint Cuts," *Proceedings of the ASME International Design Engineering Technical Conferences*, San Diego, CA.

[2] Hu, W., Li, M., Azarm, S., and Almansoori, A., 2011, "Improving Multi-Objective Robust Optimization Under Interval Uncertainty using Online Approximation and Constraint Cuts," *Journal of Mechanical Design*, 133(6), pp. 061002-1 to 061002-9.

[3] Hu, W., Butt, A., Azarm, S., Almansoori, A., and Elkamel, A., 2012c, "*Robust Multi-Objective Genetic Algorithm under Interval Uncertainty, in Multi-Objective Optimization: Developments and Prospects for Chemical Engineering*," Wiley, New York (Accepted).

subsequently the robustness of each optimal solution is evaluated and iteratively used to shrink the feasible domain and arrive at robust optimum solutions. AA-MORO involves an online approximation method. The online approximation can be used to replace a computationally expensive objective/constraint function with an inexpensive metamodel.

Compared to the previous MORO, the contributions of AA-MORO are: (i) an improved MORO approach which significantly reduces the number of robustness evaluations, and (ii) an online approximation technique which is integrated with the improved MORO to further improve the computational efficiency. Several numerical and engineering examples are used to compare AA-MORO with the previous approaches. It is shown that the optimum solutions from AA-MORO are consistent with the previous approaches. Further, it is found that AA-MORO is able to approach the robust optimum solution using a relatively small number of function calls.

In the following sections, first in Section 3.1, a literature review of related work is provided. Section 3.2 presents an improved MORO approach. Section 3.3 the AA-MORO approach. Several numerical and engineering examples are used to demonstrate the AA-MORO and compare it with the related previous approaches in Section 3.4. Finally, a summary of the chapter is provided in Section 3.5.

3.1 Literature Review

The literature on robust optimization reports on numerous methods for obtaining a solution that is optimum and relatively insensitive (or robust) to uncertainty (Park et al., 2006, Beyer and Sendhoff, 2007). A significant number of these previous techniques can be classified as bi-level (nested) or sequential. In a bi-level approach,

an upper-level master problem searches for candidate design alternatives to optimize the objective functions subject to feasibility, while a lower-level subproblem evaluates robustness for each candidate design considered at the upper level. Examples of bi-level methods have been used in reliability based optimization (Youn et. al., 2003), and design under interval uncertainty using an "anti-optimization" concept (Elishakoff and Haftka, 1993). In a sequential approach, the search for the robust optimum design is performed through a series of iterations with each iteration consisting of two steps. The first step is a deterministic optimization and the second step is the robustness evaluation. These two steps are alternated (Du and Chen 2004). Related approaches report using a simple approximation to estimate robustness of design points (e.g., Zou and Mahadevan, 2006, Liang et al., 2007), and reducing computational cost in robustness evaluation based on the linearity and convexity of the functions in optimization problems (e.g., Li et al. 2010b; Siddiqui et al., 2010).

The majority of the previous robust optimization approaches focus on single objective optimization problems. Recent interests in multiobjective optimization approaches have attracted significant research efforts in developing multiobjective optimization approaches under uncertainty (Limbourg, 2005, Ferreira et al., 2008). A limited number of previous works considers robustness of candidate points in the context of multiobjective optimization (Messac and Ismail-Yahaya, 2002, Ray, 2002). The importance of evaluating uncertainty combined with non-dominated sorting of multiobjective solution points and MOGA are also considered (Deb and Gupta, 2006; Gaspar-Cunha and Covas, 2008). In addition, a bi-level MORO is reported in which uncertainty in both objectives and constraints are accounted for (Li et al., 2006).

One well-known limitation of previous robust optimization approaches is that they require significantly more function calls than their deterministic counterpart. To address the computational issue, Approximation Assisted Optimization (AAO) is used in the literature for both single-objective optimization problems e.g., Lian and Liou (2004), Lee and Park (2006), as well as in multi-objective optimization problems (Ray et al., 2009, Voutchkov and Keane, 2010). Combining multiple surrogates in design optimization is also considered (Viana and Haftka, 2008). In addition, the literature reports on combining approximation technique with a single-objective robust optimization approach (Jin et al. 2003, Lee and Park 2006). However, integration of multiobjective robust optimization with approximation has rarely been considered (Hu et al. 2009).

3.2 Improved MORO

In this section, the details for the improved MORO approach are provided.

3.2.1 Improved MORO Formulation

The improved MORO (see for details: Hu et al., 2011) is a sequential MORO approach developed to improve the computational efficiency of the previous MORO (Li et al. 2006). The improved MORO approach is iterative with each iteration involving two steps. In the first-step, a deterministic multiobjective optimization problem is solved to obtain a set of optimal solutions; while in the second step, the robustness is evaluated for each optimal solution obtained from the first step. These two steps are alternated iteratively to obtain the robust optimum solutions. The

formulation for the optimization problems in the two steps is given in the equations below:

First step (Deterministic optimization):

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{f}(\mathbf{x},\mathbf{p}) \\ \text{s.t.} \ & \left\| [\mathbf{f}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) - \mathbf{f}(\mathbf{x},\mathbf{p})]^+ \right\| \le \eta_f, \forall \Delta\mathbf{p} \in S_f \\ & \mathbf{g}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) \le 0, \forall \Delta\mathbf{p} \in S_g \\ & \mathbf{x} \in [\mathbf{x}^l, \mathbf{x}^u] \end{aligned} \tag{3.1}$$

Second step (Robustness evaluation):

$$\max_{\Delta\mathbf{p}} \left\| [\mathbf{f}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) - \mathbf{f}(\mathbf{x},\mathbf{p})]^+ \right\| \le \eta_f, \ \forall \Delta\mathbf{p} \in [\Delta\mathbf{p}^l, \Delta\mathbf{p}^u] \tag{3.2-1}$$

$$\max_{\Delta\mathbf{p}} [\max_k \mathbf{g}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p})] \le 0, \ \forall \Delta\mathbf{p} \in [\Delta\mathbf{p}^l, \Delta\mathbf{p}^u] \tag{3.2-2}$$

Suppose the deterministic optimization problem of Eq. (3.1) obtains *np* number of Pareto optimum solutions. After robustness evaluation is performed for each Pareto optimum solution, there will be *np* number of $\Delta\mathbf{p}$ values obtained from Eqs. (3.2-1). These $\Delta\mathbf{p}$ values are returned to the deterministic problem in Eq. (3.1) and inserted in the set S_f. Likewise, *np* number of $\Delta\mathbf{p}$ values are inserted in the set S_g. These are defined as: $S_f = \{0, \Delta\mathbf{p}_1^f, ..., \Delta\mathbf{p}_{np1}^f\}$ and $S_g = \{0, \Delta\mathbf{p}_1^g, ..., \Delta\mathbf{p}_{np2}^g\}$, where $\Delta\mathbf{p}^f$ and $\Delta\mathbf{p}^g$ each represents a fixed value of $\Delta\mathbf{p}$, *np*1 and *np*2 represent the total number of $\Delta\mathbf{p}$ values in S_f and S_g, respectively. Because the improved MORO repeats the first and second step for a number of iterations and the $\Delta\mathbf{p}$ values in either S_f or S_g are accumulated, both *np*1and *np*2 can be larger than *np*. In this way, the number of constraint functions defined by $\left\| [\mathbf{f}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) - \mathbf{f}(\mathbf{x},\mathbf{p})]^+ \right\| \le \eta_f, \forall \Delta\mathbf{p} \in C_f$ is *nf*×(*np*1+1) and the number of constraints defined by $\mathbf{g}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}) \le 0, \forall \Delta\mathbf{p} \in C_g$ is *ng*×(*np*2+1) where *nf* and *ng* represent the number of objective and constraint functions, respectively. Notice that the robustness evaluation in Eqs. (3.2-1) and (3.2-2) is

similar to the lower-level subproblems in the previous MORO, as defined in Section 2.3.

3.2.2 Improved MORO Iterations

The iterative process for the improved MORO is as follows:

First Iteration: At the beginning, $S_f = S_g = \{0\}$, which means that in the first step, Eq. (3.1) reduces to the original MOP problem in Eq. (2.1). The Pareto optimal solutions from Eq. (3.1) are obtained. In the second step, the robustness for the Pareto optimum solutions from the first step is evaluated. This robustness evaluation is performed by solving Eqs. (3.2-1) and (3.2-2) for each obtained Pareto optimum solution. Solving each of the maximization problems in Eqs. (3.2-1) and (3.2-2) globally obtains an optimum value of $\Delta\mathbf{p}$. This essentially is the worst value of $\Delta\mathbf{p}$ considering the variation in the objective and constraint functions. The two worst values of $\Delta\mathbf{p}$, one from Eqs. (3.2-1) and the other from (3.2-2), are inserted in S_f and S_g, respectively. The robustness evaluation is performed for the remaining Pareto optimum solutions one by one. By the end of the second step for all Pareto optimum points in the first iteration, there are an equal number of worst values of $\Delta\mathbf{p}$ in S_f and S_g. Finally, based on the robustness evaluation, i.e., whether the inequality in Eq. (3.2-1) and Eq. (3.2-1) are satisfied, the robust solutions are identified while the non-robust ones are discarded. This completes a single iteration in the improved MORO.

Second Iteration: MORO repeats the previous steps in the first iteration except that both S_f and S_g now contain the worst values of $\Delta\mathbf{p}$, from the previous iteration. In this way, the problem in Eq. (3.1) has more constraints and becomes more restricted compared to that in the first iteration. As a result, the Pareto optimal solutions from

Eq. (3.1) may be different from those obtained in the first iteration. Again, the robustness for each Pareto optimum solution obtained in this iteration is evaluated, and additional worst values of $\Delta\mathbf{p}$ are added to S_f and S_g. The robust Pareto solutions obtained from the second iteration are combined with those from the first iteration, and this completes the second iteration in the improved MORO.

Remaining Iterations: The same procedure as in the above iterations is repeated for a number of iterations until the following stopping criteria are satisfied: (i) a maximum number of function calls is reached; (ii) no improvement in the Pareto solutions from one iteration to the next is obtained.

3.2.3 Discussions of the Sets S_f and S_g

In Section 3.2.1, it is stated that there are an equal number of worst values of $\Delta\mathbf{p}$ in S_f and S_g during the iterations of the improved MORO approach. However, in practice the number of worst values of $\Delta\mathbf{p}$ are different in S_f and S_g. The reason for this is explained in the following.

Ideally the number of worst values of $\Delta\mathbf{p}$ in S_f and S_g should be the same. This is because in the second step, Eqs. (3.2-1) and (3.2-2) are used to evaluate multiobjective and feasibility robustness constraints respectively for the same set of Pareto optimum solutions. Let *np* be the number of Pareto optimum solutions, then the total number of optimum solutions obtained from either Eq. (3.2-1) or Eq. (3.2-2) should be equal to *np*. Although the number of $\Delta\mathbf{p}$ in S_f and S_g can be equal, the value of $\Delta\mathbf{p}$ obtained from Eq. (3.2-1) is not the same as the values of $\Delta\mathbf{p}$ from Eq. (3.2-2). Therefore, the set S_f must be different from S_g. Furthermore, some values of $\Delta\mathbf{p}$ in either S_f or S_g can be redundant. Therefore, duplicate copies of $\Delta\mathbf{p}$ values are

eliminated by the end of each iteration, and thus the total number of $\Delta\mathbf{p}$ values, as represented respectively by *np*1 and *np*2, are also different.

3.2.4 Previous versus Improved MORO

There are several differences between the previous MORO (Li, et al., 2006) and the improved MORO (Hu et al., 2009):

(i) Both S_f or S_g are already known in the optimization problem in Eq. (3.1). This allows the improved MORO to solve Eq. (3.1) as deterministic optimization in the first step without further calls to the robustness evaluation in each iteration. However, S_f or S_g are updated by using the worst values of $\Delta\mathbf{p}$ from the robustness evaluation in the second step. Indeed, when the upper-level problem in the previous MORO is solved, the constraint values in Eq. (2.5), i.e., $\max\left\|\Delta\mathbf{f}^{+}\right\|$ and $\max[\mathbf{g}]$ must be evaluated by calling the robustness evaluation in the lower-level subproblems, which is referred as the bi-level approach. As a result, the upper-level problem in Eq. (2.5) requires significantly more computational cost than Eq. (3.1).

(ii) In the previous MORO, the upper-level problem in Eq. (2.5) is solved only once during the entire procedure. Because the upper-level problem evaluates multiobjective and feasibility robustness constraints for all intermediate points, the solutions from the upper-level problem include only robust solutions. In the improved MORO, the first-step problem in Eq. (3.1) is solved to produce some candidate optimal solutions. Because the multiobjective and feasibility robustness constraints are not evaluated in the first step, the candidate optimal solutions from the first step are not necessarily robust. Therefore, the robustness for these candidate optimal solutions is evaluated using Eqs. (3.2-1) and (3.2-2) during the second step. The two

steps in the improved MORO are repeated for a few iterations (e.g., about five iterations based on the empirical observations).

(iii) In the previous MORO, the lower-level subproblems defined in Eqs. (2.6-1) and (2.6-2) do not evaluate multiobjective and feasibility robustness constraints, they just provide $\max \| \Delta \mathbf{f}^{+} \|$ and max[**g**] values for the upper-level problem. On the other hand, the second-step problems in Eqs. (3.2-1) and (3.2-2) in the improved MORO evaluate multiobjective and feasibility robustness constraints. Due to the nonlinear behavior of the objective functions in Eqs. (2.6-1), (2.6-2), (3.2-1) and (3.2-2), a global optimizer such as genetic algorithm must be used for the robustness evaluation in both approaches.

Because of the differences between the two MORO approaches, as discussed above, the improved approach requires considerably less computational effort and can be more efficient than the previous approach. On the other hand, and in general, the improved MORO may not be able to obtain all robust solutions that can be obtained by the previous MORO.

One can estimate and compare the number of function calls by the previous and improved MORO as follows:

Suppose all optimization problems in both the previous and improved MORO are solved either using a multiobjective or single-objective genetic algorithm (GA) with the same number of generations and population size. Let *ngs* be the number of generations and *nps* be the population size. In the improved MORO, let *nit* represents the number of iterations and *nos* represent the average number of optimal solutions obtained from the first-step subproblems. Then the total number of function calls for

the previous MORO is of the order $\Theta(ngs^2 \times nps^2)$ while the total number of function calls in the improved MORO is of the order $\Theta[nit \times (ngs \times nps + nos \times ngs \times nps)]$ which has the same order of magnitude as $\Theta(nit \times nos \times ngs \times nps)$. The number of iterations in the improved MORO is much smaller than the number of generations of GA, e.g., as will be shown in the case study section, the number of iterations for the improved MORO is 5 while the number of GA generations is 50. Also the average number of optimal solutions must be smaller than the population size. Since $nit < ngs$ and $nos < nps$, it follows that $\Theta(nit \times nos \times ngs \times nps) < \Theta(ngs^2 \times nps^2)$. Therefore, the number of function calls by the improved MORO can be significantly less than that by the previous MORO.

Nevertheless, both the previous and improved MORO can become computationally expensive. To reduce their computation effort, an online approximation approach is developed and combined with both the previous and improved MORO, as described next.

3.3 Approximation Assisted MORO (AA-MORO)

The proposed AA-MORO approach combines the improved MORO with approximation as presented in Section 2.5. Online approximation is used to update metamodels for objective and constraint functions as new sample points are determined during AA-MORO iterations. One important goal in AA-MORO is to locate and observe a limited number of sample points while satisfying an accuracy threshold for the metamodel of all objective and constraint functions. The proposed online approximation should help improve the accuracy of the metamodels in the

neighborhood of where multiobjective optimal solutions are expected to be. As AA-MORO iteratively progresses, a better approximation of the objective and constraint functions can be expected.

In the following, the online sampling, metamodeling and verification in AA-MORO in discussed in Section 3.3.1 followed by the sample selection and filtering strategy in in AA-MORO in Section 3.5.2. Finally, the AA-MORO solution steps are presented in Section 3.3.3.

3.3.1 Online Sampling, Metamodeling and Verification

In AA-MORO, the offline samples are generated initially based on a Latin Hypercube Sampling (LHS) (Koehler and Owen, 1996). These sample points are used to construct a metamodel for each objective and constraint function. Note that each sample point needs to be observed once for all functions. Using the metamodels of the objective and constraint functions (see Section 3.3.2), AA-MORO obtains a set of estimated optimal solutions. From these estimated optimal solutions, a few are selected (the selection scheme is presented in Section 3.5.2) and observed, which are designated as the online samples. Both online and (previously obtained) offline samples are combined and used to reconstruct/update the metamodels for the objective and constraint functions. Once the metamodels are updated, online sampling is repeated until AA-MORO approaches the Pareto optimum solutions.

One motivation for using the estimated optimum solutions for online sampling is because they are potentially located close to the true optimum solutions. By observing the online sample points, the accuracy for the metamodel based objective and

constraint functions in the nearby region (close to optimum) is expected to be significantly improved.

In AA-MORO, Kriging is used as the metamodeling technique for all the objective and constraint functions. The mean squared error (*mse*) defined in Chapter 2 accounts for an estimated correlation between an unobserved point and the sample (observed) points. Since the correlation between points decreases as the distance between them increases, an unobserved point with a large *mse* indicates a poor correlation with (e.g., located in a distance from) the existing sample points. In this way, the predicted function values for a distant point (from an observed point) can become inaccurate. In AA-MORO, the *mse* at the estimated optimal solutions are calculated using Eq. (2.16). If this error is larger than a user-specified tolerance value, additional sample points are considered and observed in order to increase the accuracy of the metamodel.

3.3.2 Sample Selection and Filtering

In AA-MORO, it is unnecessary to observe all the estimated optimum solution points, instead only a subset of the intermediate optimum solution points are selected and observed. For selection, the optimum solution points are ranked by using the *mse* which can be calculated from Eq. (2.16). From a sampling point of view, it is more desirable to observe a sample with relatively larger *mse* in order to improve the overall accuracy of the approximation. On the other hand, since there are multiple functions (objectives and constraints) that need to be approximated through Kriging at each unobserved sample point, it is easier to define a single scalar as a measure of the overall accuracy for the estimated functions as shown in Eq. (3.3)

$$error = (\sum mse_f^2 + \sum mse_g^2)^{1/2} \tag{3.3}$$

where mse_f and mse_g represent the Kriging *mse*'s of the objective and constraint functions respectively. The values of the Kriging mdeta-model of objective and constraint functions must be normalized so that their Kriging *mse* calculated from Eq. (2.16) are in the same scale. For example, the values of objective and constraint functions can be normalized using the largest absolute values for corresponding functions. Based on the calculation from Eq. (3.3), an estimated optimum solution with the largest error is ranked first followed by the one with the second largest error, and so on. Using this error based ranking, one or a number of solution points are selected and observed, i.e,, their actual objective/constraint function values are computed, in AA-MORO.

In addition to the scheme in selecting the estimated optimum solutions, a sample filtering is also considered in AA-MORO to prevent clustering of sampled points. When the distance (measured in the design variable space) between a new sample and a previous sample point is less than a threshold value, the new sample point is eliminated. In AA-MORO, the following filtering constraint is used: $\|\mathbf{x}_n - \mathbf{x}_e\| \geq \varepsilon$, where $\mathbf{x}_n$ refers to a new sample point and $\mathbf{x}_e$ refers to any existing sample points. $\|\cdot\|$ denotes the Euclidean norm (distance) between two vectors in the sample space, and ε is a user defined threshold value specifying the minimum acceptable distance between two sample points. This constraint requires that the distance between a new sample point and any existing sample points must be large than the minimum acceptable distance. As a general rule, the threshold is selected such that it is at least larger than one half of the shortest distance among the existing sample points (pair-

wisely measured). Additionally, the computational costs of the optimization models must also be considered in selecting the threshold ε. When the objectives and constraint functions are computationally expensive to compute, the value of ε needs to be increased in order to reduce the number of online samples. After additional sample points are determined and the actual simulations are evaluated to obtain the response (or observed) values, these sample points are added to the current set of sample points. Finally, the updated sample points are used to update the metamodels.

3.3.3 AA-MORO Solution Steps

The steps of the AA-MORO approach are summarized below:

Step 1: Initialize S_f or S_g, i.e., $S_f = S_g = \{0\}$. An initial set of sample points based on LHS is created.

Step 2: Calculate response values for all sample points by calculating the true value of objective and constraint functions. The Kriging metamodels are created for each objective and constraint function based on the current sample points.

Step 3: Obtain Pareto optimal points by solving the optimization problem in Eq. (3.1) using the metamodels.

Step 4: Evaluate robustness for each Pareto optimal points from Step 3 based on the metamodels.

Step 5: Return the worst values of $\Delta\mathbf{p}$ from Step 4 to Eq. (3.1) and insert them into S_f and S_g .

Step 6: Stop if one of the two stopping criteria stated next is satisfied; otherwise, continue to Step 7. [The stopping criteria are: (i) AA-MORO should stop if a pre-specified maximum number of iterations is reached. An iteration in AA-MORO is

defined as a single pass from Step 2 through Step 5. (ii) AA-MORO should stop if no change to S_f and S_g in Step 5 is observed after a pre-specified number of consecutive iterations.]

Step 7: The Pareto optimal points and worst values of $\Delta\mathbf{p}$ are added to the existing sample set and the control returns to Step 2.

3.4 Numerical and Engineering Examples

In this section the results for the application of AA-MORO to several numerical and engineering examples are presented. Based on empirical observations, the maximum number of iterations for AA-MORO is set to five (5) to allow a sufficient number of robust solutions to be obtained. The multiobjective genetic algorithm of MATLAB™ "Global Optimization Toolbox" version 2010a (Mathwork, 2010) is used as the optimizer. Parameter settings for the genetic algorithm in all examples are shown in Table 1. The DACE toolbox (Lophaven et al. 2002) is used for constructing Kriging metamodels in all examples. A second order polynomial function and a Gaussian function is used for Kriging metamodeling. The initial offline samples are generated using a LHS technique with $(nx+1)\times(nx+2)/2$ number of samples, which is the minimum number of sample points required for constructing a Kriging metamodel.

Table 3.1 Genetic algorithm parameter settings

Parameter	Upper level/First step	Lower level/Second step
Population size	$15\cdot nx$	$15\cdot npr$
Maximum generation	50	50
Elite number	1($nx < 5$) and 2($nx > 5$)	1
Crossover probability	0.9	0.9
Mutation probability	0.1	0.1

* *nx* and *npr* are the number of variables and uncertain parameters, respectively

3.4.1 Illustrative Example

The first example, well known in the literature (Deb, 2001) as TNK, is a deterministic bi-objective optimization problem that has been revised here with some added uncertainty. The deterministic problem formulation is:

$$\begin{aligned} &\min \quad f_1 = x_1 \\ &\min \quad f_2 = x_2 \\ &\text{s.t.} \quad g_1 = 1 + 0.1\cos(16\arctan\frac{x_1}{x_2}) + \\ &\qquad\qquad 0.2\sin(p_1)\cos(p_2) - x_1^2 - x_2^2 \le 0 \\ &\qquad g_2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \le 0.5 \\ &\qquad 0 < x_1, x_2 \le \pi \end{aligned} \tag{3.4}$$

The feasible domain is defined by the non-convex area within two inequality constraints g_1 and g_2. The optimum solutions to the problem in Eq. (3.4) are obtained using a multiobjective genetic algorithm and shown as "MOGA (Deterministic)" in Fig. 3.1. Notice that these solutions are located along the boundary of constraint g_1. Due to the non-convexity of g_1, the Pareto frontier consists of three discontinuous sections as shown in Fig. 3.1.

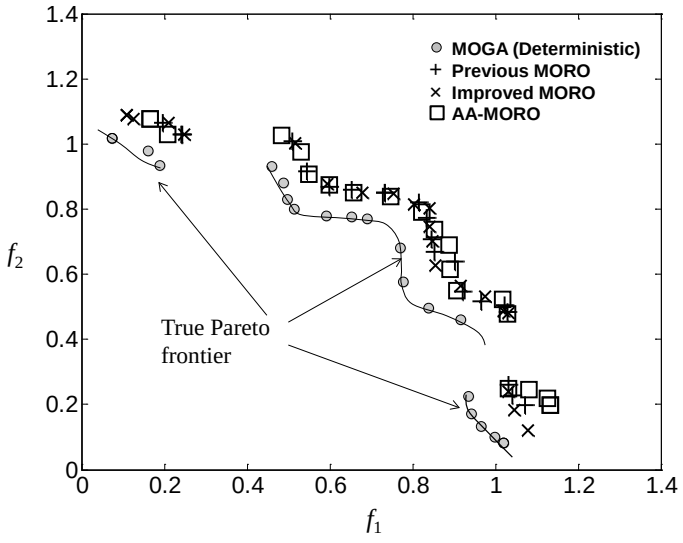


Fig. 3.1 Optimum solutions in numerical example (TNK)

By considering interval uncertainty in the parameters, the optimization problem in Eq. (3.4) can be formulated using the previous MORO with an upper-level problem and a lower-level subproblem as in Eq. (3.5) and Eq. (3.6), respectively:

$$\begin{aligned}
&\min \quad f_1 = x_1 \\
&\min \quad f_2 = x_2 \\
&\text{s.t.} \quad g_1 = 1 + 0.1\cos(16\arctan\frac{x_1}{x_2}) + \\
&\qquad 0.2\sin(p_1)\cos(p_2) - x_1^2 - x_2^2 \le 0 \\
&\qquad g_2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \le 0.5 \\
&\qquad \max[\mathbf{g}] \le 0 \\
&\qquad 0 < x_1, x_2 \le \pi
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
&\max \left\|\mathbf{g}\right\| = \max_{\Delta\mathbf{p}} \quad 1 + 0.1\cos(16\arctan\frac{x_1}{x_2}) + \\
&\qquad 0.2\sin(p_1 + \Delta p_1)\cos(p_2 + \Delta p_2) - x_1^2 - x_2^2 \\
&\text{s.t. } \Delta p_1, \Delta p_2 \in [-2,\ 2]
\end{aligned} \tag{3.6}$$

This optimization problem has two design variables and two uncertain parameters. The nominal values for both parameters are: $p_1 = p_2 = 1$. Δp_1 and Δp_2 represent the uncertainty in the parameters and their uncertainty ranges are as specified in Eq. (3.6). Notice that the upper-level problem in Eq. (3.5) reduces to the original formulation in Eq. (3.4) when $\Delta p_1 = \Delta p_2 = 0$. Because there is no uncertainty in the objective functions and constraint g_2, only feasibility robustness for constraint g_1 is considered in this example. The lower-level subproblem in Eq. (3.6) is essentially a single-objective maximization of g_1. On the other hand, the optimization problem in Eq. (3.4) can be formulated using the improved MORO as shown in Eq. (3.7) and (3.8) respectively:

Deterministic Optimization

$$\begin{aligned}
&\min \quad f_1 = x_1 \\
&\min \quad f_2 = x_2 \\
&\text{s.t.} \quad g_1 = 1 + 0.1\cos(16\arctan\frac{x_1}{x_2}) + 0.2\sin(p_1 + \Delta p_1) \\
&\qquad \cos(p_2 + \Delta p_2) - x_1^2 - x_2^2 \le 0, \forall \Delta p_1, \Delta p_2 \in S_g \\
&\qquad g_2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \le 0.5 \\
&\qquad 0 < x_1, x_2 \le \pi
\end{aligned} \tag{3.7}$$

Robustness evaluation

$$\begin{aligned}
&\max_{\Delta \mathbf{p}} \quad 1 + 0.1\cos(16\arctan\frac{x_1}{x_2}) + \\
&\qquad 0.2\sin(p_1 + \Delta p_1)\cos(p_2 + \Delta p_2) - x_1^2 - x_2^2 \le 0 \\
&\text{s.t. } \Delta p_1, \Delta p_2 \in [-2,\ 2]
\end{aligned} \tag{3.8}$$

where S_g represent a set of Δp_1 and Δp_2 values in the uncertain interval. As mentioned earlier, S_g is determined after robustness evaluation in Eq. (3.8).

Next, the numerical example is solved with AA-MORO and compared with the previous MORO (defined in Eqs. (3.5) and (3.6)), improved MORO (defined in Eqs. (3.7) and (3.8)). Notice that in this example, the metamodels are developed only for the constraint functions but not for the objective functions due to the simplicity of the objective functions. The number of initial (offline) sample points is 15. To account for the randomness in GA, all MORO approaches were repeated for 10 times, among which the best solutions are selected based on the Hyperarea Difference (HD) value (a quality metric developed by Wu and Azarm, 2001) and plotted in Fig. 3.1. Notice that the "Previous MORO" and "Improved MORO" in Fig. 3.1 refer to the approaches without approximation.

According to Fig. 3.1, the optimal solutions from both MORO approaches are inferior to the deterministic solutions, which are expected because the robust solutions are typically more conservative than the deterministic ones. On the other hand, the

optimal solutions from both previous and improved MORO approaches, and AA-MORO are generally consistent in the objective space. Table 3.2 compare the obtained mean value and standard deviation information for different MORO approaches. The previous MORO typically requires a large number of function calls; while the improved MORO requires considerably (about two orders of magnitude) less number of function calls.

Table 3.2 Number of function calls and quality metrics (HD and OS) for the illustrative, numerical and oil refinery (case study) examples

Examples	Approach	Average Num. function call	HD Mean (std.)	OS Mean (std.)
Illustrative	Previous MORO	1.9×10^{6}	0.60 (0.04)	0.27 (0.09
	Improved MORO	1.1×10^{5}	0.53 (0.01)	0.64 (0.27
	AA-MORO	26*	0.58 (0.04)	0.34 (0.08)
Example 1	Previous MORO	3.4×10^{6}	0.33 (0.01)	0.28 (0.08)
	Improved MORO	2.2×10^{5}	0.33 (0.01)	0.27 (0.07)
	AA-MORO	466*	0.34 (0.02)	0.31 (0.13)
Example 2	Previous MORO	3.5×10^{6}	0.42 (0.11)	0.37 (0.15)
	Improved MORO	1.9×10^{5}	0.41 (0.08)	0.36 (0.17)
	AA-MORO	657*	0.49 (0.12)	0.37 (0.17)
Example 3	Previous MORO	4.1×10^{6}	0.57 (0.11)	0.22 (0.13)
	Improved MORO	1.9×10^{5}	0.51 (0.08)	0.18 (0.05)
	AA-MORO	571*	0.58 (0.05)	0.14 (0.14)
Example 4	Previous MORO	4.3×10^{6}	0.69 (0.05)	0.22 (0.08)
	Improved MORO	1.9×10^{5}	0.65 (0.06)	0.18 (0.13)
	AA-MORO	614*	0.66 (0.04)	0.08 (0.05)
Oil refinery	AA-MORO	28*	0.66 (0.01)	0.22 (0.05)

*The number of function calls for AA-MORO refers to the number of sample points

The quality metrics (Wu and Azarm, 2001): Hyperarea Difference (HD) and Overall Pareto Spread (OS) are calculated to measure the goodness of the Pareto solution sets. It is found that the optimal solutions obtained from the improved MORO are slightly better than the solutions obtained from the previous MORO. For AA-MORO, the number of sample points is treated as the number of function calls. It is observed that

AA-MORO significantly reduces the number of calls for the evaluation of the constraint functions.

The mean square errors calculated from the Kriging model (recall Eq. (2.16)) for the final optimal solution in AA-MORO are reasonably small. Furthermore, the optimum solutions obtained from different MORO approaches (including the previous and improved MORO and AA-MORO) are validated to ensure feasibility robustness for the optimal solutions are satisfied. For robustness validation, a Monte Carlo simulation method is used with a large number (10^4) of randomly points is generated within the uncertainty variation range around each optimal solution points. The probability of violation is calculated based on the number of points which violates the objective/feasibility robustness constraints. It is found that robustness of the solutions from different MORO approaches are validated with a probability of violation that is less than a 0.01% threshold value.

3.4.2 AA-MORO vs. Previous and Improved MORO

In this section, four numerical examples adapted from the literature, which are originally in deterministic form (Deb, 2001), are used to compare AA-MORO with the previous and improved MORO approaches. The problem formulations are summarized below:

Example 1

$$
\begin{aligned}
&\min\ f_1 = x_1 \\
&\min\ f_2 = g(1-\sqrt{f_1 / g}) \\
&\text{where } g = 1+\Delta p+9\sum_{i=2}^{5}(x_i - x_1)^2 / 4 \\
&\forall \Delta p \in [-0.2, 0.2]; \Delta f_1 = 0.1, \Delta f_2 = 0.15 \\
&0 \le x_i \le 1, i = 1,...,5
\end{aligned}
\tag{3.9}
$$

Example 2

$$
\begin{aligned}
&\min\ f_1 = x_1 \\
&\min\ f_2 = g(1 - f_1 / g^2) \\
&\text{where}\ \ g = 1 + \Delta p + 9(\sum_{i=2}^{5} (x_i - x_1)^2) / 4 \\
&\forall \Delta p \in [-0.2, 0.2]; \Delta f_1 = 0.1, \Delta f_2 = 0.3 \\
&0 \le x_i \le 1, i = 1, ..., 5
\end{aligned}
\tag{3.10}
$$

Example 3

$$
\begin{aligned}
&\min\ f_1 = x_1 + (1 + \Delta p) - 1 \\
&\min\ f_2 = g(1 - \sqrt{f_1 / g}) \\
&\text{where}\ \ g = 1 + 9[\sum_{i=2}^{5} (x_i^2 - x_1)^2] / 4 \\
&\forall \Delta p \in [-0.1, 0.1]; \Delta f_1 = 0.3, \Delta f_2 = 0.15 \\
&0 \le x_i \le 1, i = 1, ..., 5
\end{aligned}
\tag{3.11}
$$

Example 4

$$
\begin{aligned}
&\min\ f_1 = \sqrt{x_1 + 1 + \Delta p} - 1 \\
&\min\ f_2 = g[(1 - (f_1 / g)^2] \\
&\text{where}\ \ g = 1 + 9(\sum_{i=2}^{5} (x_i^2 - x_1)^2) / 4 \\
&\forall \Delta p \in [-0.2, 0.2]; \Delta f_1 = 0.3, \Delta f_2 = 0.15 \\
&0 \le x_i \le 1, i = 1, ..., 5
\end{aligned}
\tag{3.12}
$$

The optimal solutions from AA-MORO are compared with the previous and improved MOROS approaches. As in the illustrative example, the best solutions from each approach are selected based on the HD value and plotted in Fig. 3.2. The deterministic solutions from MOGA are also obtained as a baseline for the comparison. As shown in Fig. 3.2, in general, the solutions obtained from AA-MORO are comparable with the solutions from the two MORO approaches. However, in some cases as in Example 3, AA-MORO may not be able to identify the complete robust optimal frontier. Another observation is with the relative goodness of the robust optimal solutions in comparison with the deterministic optimal solutions. It can be seen that a portion of the deterministic optimal designs could by themselves be

robust as shown in Examples 1 and 2. But in Examples 3 and 4, the robust solutions are dominated by the deterministic solutions.

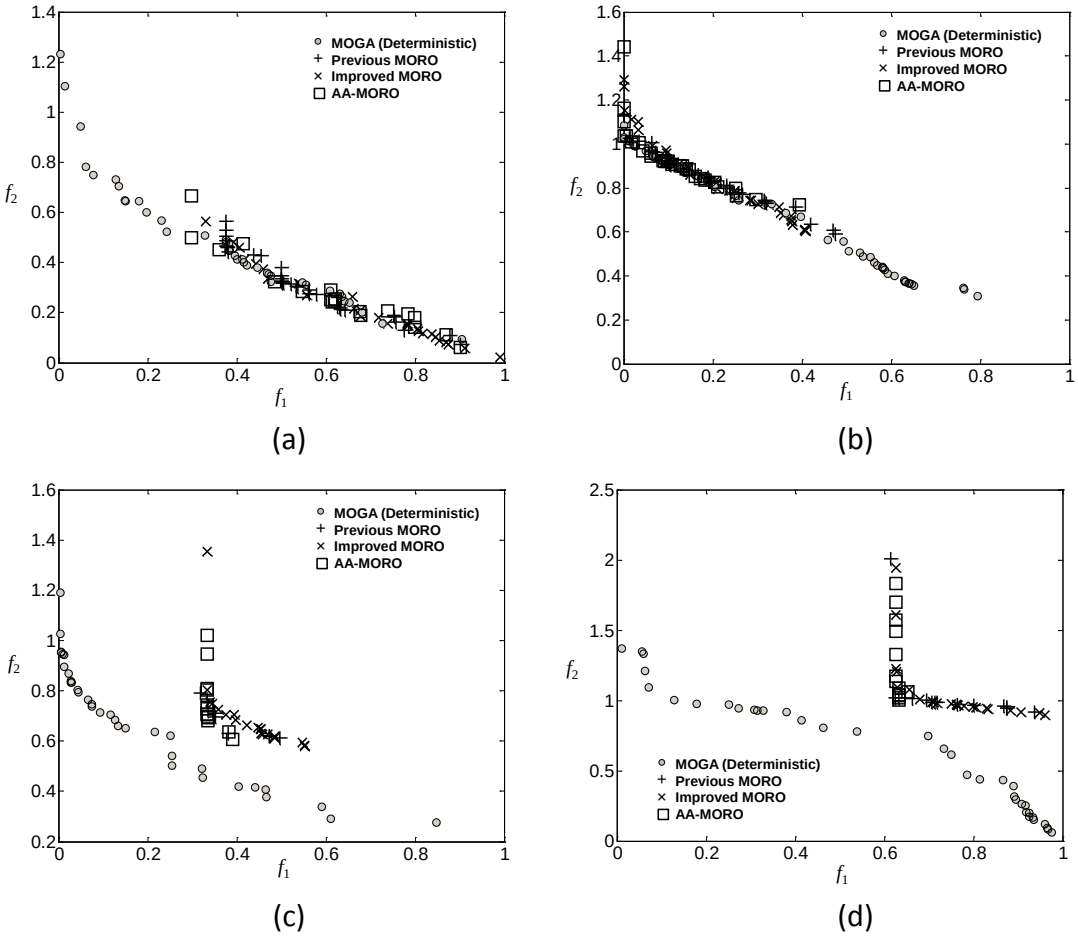


Fig. 3.2 Comparison of AA-MORO with previous and improved MORO approaches: (a)-(d) numerical example 1-4

The mean value and standard deviation of the number of function calls and the quality metrics with the numerical examples are summarized in Table 3.2. Again the optimum solutions from AA-MORO are generally consistent with the other two MORO approaches. Finally, the AA-MORO solutions are validated based on the mean square error calculated using Eq. (2.16) from the Kriging model. Like in the numerical examples, for robustness validation, a Monte Carlo simulation method is used with 10^4 of random points with the probability of violation less than 0.01%.

3.4.3 Oil Refinery Case Study

In this case study, a typical crude oil refinery is considered (Hu et al., 2012a), and the AA-MORO approach is employed for optimization. The refinery consists of common unit process/operations and nonlinear correlations are used to predict the yields and properties of the products of each unit. The units in this refinery case study are: 1) Crude distillation unit; 2) Delayed coker; 3) Hydrocracker for heavy vacuum gas oils; 4) Hydrotreater for light vacuum gas oils; 5) Fluid catalytic cracking unit (FCCU); 6) Hydrotreater for heavy straight run naphtha: 7) Catalytic reformer; 8) Light naphtha hydrotreater; 9) Isomerization unit.

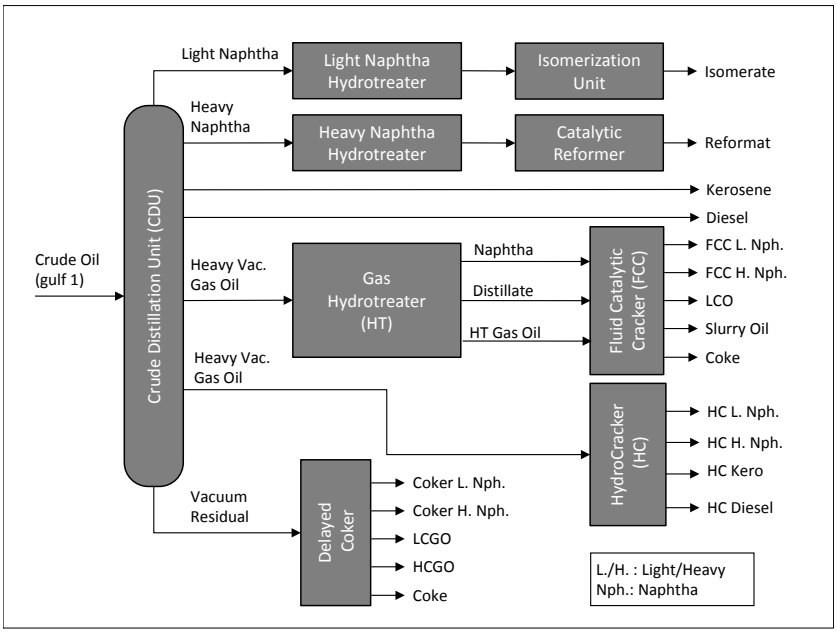


Fig. 3.3 Schematic of refinery model

The schematic of the oil refinery is shown in Fig. 3.8. The flow diagram depicts various unit processes and flows of intermediate product streams. The products out of the crude distillation unit are lower straight run (LSR) naphtha, higher straight run (HSR) naphtha, straight run diesel (SRD), kerosene, light vacuum gas oil (LVGO),

heavy vacuum gas oil (HVGO) and vacuum residue (VacResid). The Vacuum residue is further processed in the delayed coker to get the lighter fractions. The heavy vacuum gas oils are hydrocracked in the hydrocracker to get light naphtha and heavy naphtha fractions. The LVGO, HSR and LSR are hydrotreated to reduce the sulfur contents and further treated in FCCU, Catalytic reformer and isomerization unit respectively to get the products of interest. All naphtha is sent to the blending pool to get the gasoline for the required grade.

The simulation of the described refinery is done through Matlab (Mathwork, 2010) and simple non-linear correlations are used. The flow rate of crude oil to the crude distillation unit is assumed to be fixed with a value of 100,000 BPD. For simplicity, the schematic in Fig. 3.8 does not include the utility units such as steam, cooling water and electricity. Also, the storage facilities such as crude oil and intermediate product storage tanks are not shown.

The refinery model is formulated as a MOO problem as described in Eq. (3.13). The two objectives are to maximize the product flow rate f_1 and to minimize the cost f_2. Both objectives can be evaluated from the refinery simulation model for a given set of design variables. The variables considered for optimization are the six cut temperatures (t_1, t_2,…, t_6) in the crude distillation unit. The lower and upper bounds for the cut temperatures are given in Eq. (3.13). It is assumed that t_2 and t_3 are uncertain and the uncertainties are represented by Δt_i, j = 1, 2, and the range of uncertainties are between ±10% of their nominal cut temperature values.

$$
\begin{aligned}
&\max\ f_1(t) = \text{flow rate of light naphtha (bbl/day)} \\
&\min\ f_2(t) = \text{total cost(\$/day)} \\
&\text{s.t. } \left\| [\, f_m(t,\Delta t) - f_m(t)]^+ \right\| \le \eta_f, m = 1,2 \\
&\quad \forall \Delta t_1^l \le \Delta t_1 \le \Delta t_1^u, \Delta t_2^l \le \Delta t_2 \le \Delta t_2^u \\
&\quad 162 \le t_1 \le 198; \\
&\quad 360 \le t_2 \le 440; \\
&\quad 477 \le t_3 \le 583; \\
&\quad 585 \le t_4 \le 715; \\
&\quad 810 \le t_5 \le 950; \\
&\quad 950 \le t_6 \le 1155;
\end{aligned}
\tag{3.13}
$$

In the refinery example, the absolute value for the two objective functions are not in the same scale, e.g., the flow rate of light naphtha and total cost are in the order of 10^4 and 10^6 respectively. As such, the original value of the flow rate and total cost are first normalized to a value of unity using normalizing factors 10^5 and 10^7, respectively. The advantage of normalization is that using a Euclidean norm to restrict the objective variation as shown by the inequality constraint in Eq. (3.13), will give equal importance for both objectives. The acceptable variation limit η_f is specified as 0.1. The optimization problem given in Eq. (3.13) is solved using AA-MORO and compared with the deterministic approach (MOGA). Due to the randomness in AA-MORO, a total of 10 runs are performed and the best sets of optimal solutions based on the HD value are plotted in the objective functions space in Fig. 3.4. It can be seen that the optimal solutions for the refinery example from AA-MORO are much inferior to the MOGA solutions. For example, the deterministic optimal solutions are better than AA-MORO approaches in achieving maximum flow rate of light naphtha, while the uncertainty in the cut temperature seems to have little effect on the daily total cost.

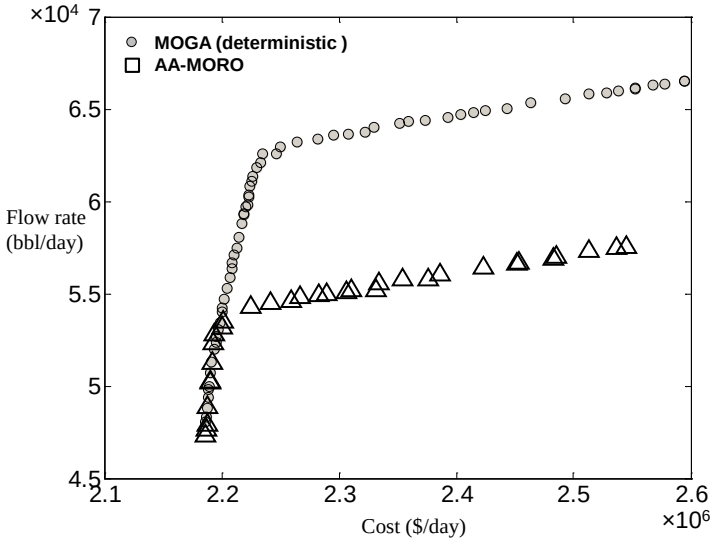


Fig. 3.4 The optimal solutions for the refinery example

The average value and standard deviation of the optimal solutions based on the ten runs for AA-MORO are shown in the last row of Table 3.2. The average number of iterations for AA-MORO in the refinery example is 3. The number of total samples in AA-MORO for all 10 runs is 28. Based on the MSE error calculated using Eq. (2.16) in the approximated objective functions (less than 0.001), Kriging provides good accuracy in both approaches. This is possibly due to the good characterization of the polynomial relationship between the input and output variables in the refinery model.

3.5 Summary

In this chapter, an AA-MORO approach is presented to address the associated computational cost challenge in MORO. AA-MORO evaluates robustness of solutions in terms of their objective and constraint functions by using a worst-case analysis. AA-MORO integrates an online approximation method in the improved MORO approach where in an iterative fashion a deterministic optimization problem

is solved in the first step to obtain multiobjective optimum solutions. The solutions are then passed on to the second step for a robustness evaluation. It is shown that the improved MORO can be more efficient than the previous MORO approach. However, the computational cost for applying the improved MORO can be intractable. To overcome this difficulty, an online approximation method is integrated with the improved MORO.

Several numerical examples and an oil refinery test examples are solved and their results are compared with the previous methods. In the majority of the test examples, the solutions from AA-MORO are compared well to those obtained from the previous MORO approaches. The results from all numerical examples and case study also indicate that typically AA-MORO requires considerably fewer number of function calls than previous approaches.

In the next chapter, the development of an improved Multiobjective collaborative Robust Optimization (McRO) approach is detailed.

Chapter 4: Approximation Assisted Multiobjective collaborative Robust Optimization under Interval Uncertainty (AA-McRO)

This chapter provides the details of a new approach for multiobjective collaborative robust optimization with approximation and interval uncertainty considerations. The material of this chapter is essentially the same as that given in the paper by Hu et al. (2012b)[4] with some slight modifications.

Existing collaborative optimization techniques with multiple coupled subsystems are predominantly focused on single-objective deterministic optimization. However, many engineering optimization problems have system and subsystems optimization problems that can each be multiobjective, constrained under uncertainty. The literature reports on a few deterministic Multiobjective Multi-Disciplinary Optimization (MMDO) techniques. However, these techniques in general require a large number of function calls and their computational cost can be exacerbated when uncertainty is present. In this chapter, a new Approximation-assisted Multiobjective Collaborative Robust Optimization (AA-McRO) under interval uncertainty is presented. This new AA-McRO approach uses a single-objective optimization problem to coordinate all system and subsystem problems in a Collaborative Optimization (CO) framework. The approach also converts the consistency constraints of CO into penalty terms which are integrated into the subsystem

[4] Hu, W., Azarm, S., and Almansoori, A., 2012b, "New Approximation Assisted Multi-Objective Collaborative Robust Optimization (New AA-McRO) Under Interval Uncertainty," *Structure and Multidisciplinary Optimization* (To be appeared).

objective functions. The new AA-McRO is able to explore the design space and obtain optimum design solutions more efficiently. The new AA-McRO approach obtains an estimate of Pareto optimum solutions for MMDO problems whose system-level objective and constraint function are relatively insensitive (or robust) to input uncertainties. Another characteristic of AA-McRO is the use of online approximation for objective and constraint functions to perform system robustness evaluation and subsystem-level optimization. Based on the results from a numerical and an engineering example, it is concluded that AA-McRO performs better than previously reported MMDO methods.

Compared to the previous McRO approach (Li and Azarm, 2008) presented in Chapter 2, the proposed AA-McRO approach has made the following contributions: (i) AA-McRO converts an upper-level system problem in McRO into a single-objective coordination problem at the upper level and a multiobjective optimization problem at the lower level. Under this framework, the upper-level problem is responsible for coordinating the shared and coupling variables and guiding the lower-level problems, while the system problem in the lower-level is responsible for achieve the optimum design solutions. Because the upper-level problem in AA-McRO only focuses on coordination, it is able to reach convergence and obtain optimum design solutions more efficiently. (ii) AA-McRO converts the consistency constraints in the lower-level optimization problems into penalty terms which are integrated into the objective function at the lower level. These penalty terms allow the system and subsystem optimization to explore the design space better. However as the optimization proceeds, the penalized value is minimized so that eventually the consistency constraints are

satisfied. (iii) AA-McRO improves the optimal solution selection strategy at the lower level problems. This improvement enhances the consistency among the shared and coupling variables and further improves system convergence. (iv) AA-McRO employs an online approximation technique to reduce the number of function calls. An online verification of the estimated optimum solution is integrated such that the absolute error of the objective and constraint functions can be kept within a user specified threshold. In this way, AA-McRO can significantly reduce the computational effort compared to McRO and obtain reasonably accurate optimum solutions. A numerical and an engineering problem are tested with the AA-McRO approach. The test results show that with a limited number of sample points, AA-McRO is able to obtain a good set of optimal solutions with significantly less computational cost for a MMDO problem under interval uncertainty than the McRO approach of (Li and Azarm, 2008).

Section 4.1 provides a review of related work in the literature and presents the limitation of the McRO approach. A description of the new MCO approach is presented in Section 4.2.1, and the new McRO approach in presented in Section 4.2.2, In Section 4.3, the new AA-McRO approach is presented, including the online approximation approach in Section 4.3.1, and the steps in the new AA-McRO approach in Section 4.3.2. To illustrate the new AA-McRO approach, one numerical and one engineering example are solved and discussed in Section 4.4. Section 4.5 summarizes the new AA-McRO approach with some concluding remarks.

4.1 Literature Review

Multi-Disciplinary Optimization (MDO) refers to a class optimization methods for solving system optimization problems that can be decomposed into multiple coupled subsystem optimization subproblems (e.g., Sobieszczanski-Sobieski, 1982; Azarm and Li, 1988; Renaud and Gabriele, 1993; Balling and Sobieszczanski-Sobieski, 1996, Allison et al., 2009). Examples of MDO include methods like the all-at-once (Cramer et al., 1994), concurrent subspace optimization (Sobieszczanski-Sobieski, 1998), Collaborative Optimization (CO) (Braun and Kroo, 1996; Kroo and Manning, 2000), bi-level integrated system synthesis (BLISS) (Sobieszczanski-Sobieski and Agte, 2000), Analytical Target Cascading (ATC) (Kim et al., 2003) and so on. Among these methods, the CO approach requires less information exchange among the subsystems and allows more flexibility at subsystem optimization (Kroo and Manning, 2000). This characteristic of CO has attracted considerable interests among researchers in improving the CO technique (Roth and Kroo 2008) and applying it to engineering optimization problems (e.g., Gu, et al., 2006).

The original CO approach was developed for a single-objective optimization problem at the system level while the subsystems did not have any design objective (Braun and Kroo, 1996). Related work also proposed formulating CO with the use of penalty methods (DeMiguel and Murray, 2006). However, these CO formulations in general are not suitable for MDO with multiple objectives at both system- and subsystem-level. Some previous methods have reported extending single-objective CO to multiobjective CO. For example, Tappeta and Renaud (1997) proposed a multiobjective CO where a system-level objective function was formulated as a

weighted sum of subsystem level objectives. However, their formulation has sthe shortcomings of the weighting method, which is not capable of capturing non-convex portion of the Pareto frontier (Deb, 2001). Gunawan et al. (2003) developed a multidisciplinary multiobjective genetic algorithm. However, their formulation used quality metrics at the system level instead of system design objectives and they did not consider any interdisciplinary couplings. Other examples include physical programming (McAllister et al., 2004) and genetic algorithms (Aute and Azarm, 2006) in multiobjective CO (MCO).

Recently, some MDO methodologies have been extended to account for uncertainty. Gu and Renaud (2006) considered an implicit uncertainty propagation technique and developed a robust CO framework. Kokkolaras et al. (2006) developed a probabilistic version of the ATC for hierarchically decomposable systems under uncertainty. Ahn and Kwon (2006) proposed a "ProBLISS" approach by embedding a single-level reliability-based design scheme with a BLISS framework. Li and Azarm (2008) extended the MCO framework of Aute and Azarm (2006) and developed a Multiobjective Collaborative Robust Optimization (McRO) approach under interval uncertainty. Among these approaches, the McRO technique allows for both system and subsystem levels to have multiple design objectives and also considers uncertainties in input parameters with a multidisciplinary uncertainty propagation approach.

One of the most pressing limitations of the bi-level CO formulation (Braun and Kroo, 1996) is that a large number of iterations is required between the system- and subsystem-level optimization problems. As a result, a significant number of function

calls is required to find optimum solutions and that the optimization of even a relatively simple engineering system could become intractable. To overcome this difficulty, Roth and Kroo (2008) proposed an enhanced CO formulation to improve convergence. They suggested solving the system- and subsystem-level optimization problems sequentially which reduced the computational cost. An overview of the bi-level and sequential CO formulations can be found in Tosserams et al. (2009). In addition, a variety of methods are proposed to use approximation in CO. Notably, Sobieski et al. (1998) developed a local response surface method to approximate a subsystem optimization output in CO. Simpson et al. (2001a) applied Kriging to construct global approximation in an MDO framework. Jang et al. (2005) combined neural network and Kriging metamodeling of subsystems to classify feasibility of a system design vector and employed an "adaptive approximation" to maintain accuracy. More recently, Zadeh et al. (2008) proposed an approximation-based CO in which multi-fidelity and global approximation is used at the subsystem and system levels, respectively. The aforementioned methods mainly focus on deterministic and single-objective MDO.

4.1.1 Limitations of the McRO Approach of (Li and Azarm, 2008)

As presented in Chapter 2, McRO provides a general approach for solving robust MDO problems under interval uncertainty. However, there are several difficulties associated with the McRO approach, as elaborated in the following:

(1) The system optimization problem in McRO fulfills two goals. First, a multiobjective optimization problem is solved at the upper level to determine system design variables while optimizing the system design objectives. Second, the system

problem coordinates subsystem optimization problems by selecting the shared and target variables in order to satisfy the subsystem consistency constraints. In this way, the system optimization problem in McRO becomes rather restricted and typically includes a large number of variables. This presents considerable challenge for the system optimizer convergence to solutions in an efficient manner.

(2) Both system and subsystem optimization problems in McRO use consistency constraints to ensure that the coupling variables match among the subsystems. These consistency constraints are strict, as specified by a small tolerance value on their right-hand side, see Li and Azarm (2008). As a result, it is difficult to obtain feasible solutions for these optimization problems and the system optimization in McRO may require many iterations before converging to final solutions.

(3) As mentioned earlier in this section, the subsystem optimization problem in McRO obtains a set of optimum solutions for each system candidate design alternative. Therefore a strategy must be developed to select the optimal solutions produced from the subsystem optimization problems. These solutions are then returned to the system optimization problem. There are a few strategies reported in the literature which can be used for this purpose (Aute and Azarm, 2006), including selecting a single solution from each subsystem or combining different optimal sets from all subsystems. For example, one can devise a scheme that selects the “best” solution with the minimal value for one of the objectives which is considered the most important one (Aute and Azarm, 2006; Li and Azarm, 2008). However, such previous selection strategies in general are rather arbitrary. Moreover, the criteria of

the selection are based solely on the design objectives in the subsystem but ignore issues such as convergence at the system optimization problem.

(4) Finally, solving a MDO problem with McRO can be computationally expensive because of a large number of iterations required between the system and subsystem optimization problems. With the integration of uncertainty and robustness evaluation in the lower level (Fig. 2.5), the computational cost of McRO increases further.

The objective of the current chapter is to enhance McRO in order to address the above mentioned difficulties.

4.2 New Collaborative Optimization Approach

In this section, the newly developed AA-McRO approach is presented. Compared to McRO, the AA-McRO has two major improvements. These are: (i) a new McRO formulation to improve convergence, and (ii), online approximation approach to significantly reduce the computational effort.

The formulations for the new MCO and new McRO are presented in Section 4.2.1 and 4.2.2. Section 4.2.3 discusses the online approximation in the new AA-McRO and finally the solution steps of the proposed approach are given in Section 4.2.4.

4.2.1 New MCO Approach

The basic idea in the new MCO approach is to divide the system optimization problem into two subproblems. One of the subproblems is placed at the upper level for coordinating the system and all subsystem optimization problems. The second subproblem is placed at the lower level for optimization of the system design objectives subject to its constraints. Essentially this new system optimization problem

is solved and treated in the same way as is a lower level subsystem optimization problem. The formulations for the coordination problem, the new system and subsystem problems are given as:

Coordination problem (upper level)

$$\begin{aligned} \min_{\mathbf{X}_0} \quad & \sum_{i=0}^{I} \left\| \mathbf{x}_{\text{sh}}^{i} - \mathbf{x}_{\text{sh}} \right\|_2 + \sum_{i=1}^{I} \sum_{j \neq i}^{I} \left\| \mathbf{y}_{ij} - \mathbf{t}_{ij} \right\|_2 + \sum_{i=1}^{I} \sum_{j \neq i}^{I} \left\| \mathbf{t}_{ij}^{i} - \mathbf{t}_{ij} \right\|_2 \\ \text{s.t.} \quad & \mathbf{X}_0 \in [\mathbf{X}_0^l, \mathbf{X}_0^u]; \ \mathbf{X}_0 \equiv [\mathbf{x}_{\text{sh}}, \mathbf{t}_{ij}] \end{aligned} \tag{4.1}$$

New system problem (lower level)

$$\begin{aligned} \min_{\mathbf{X}_0} \quad & \mathbf{f}_0(\mathbf{X}_0, \mathbf{P}_0) + \left\| \mathbf{x}_{\text{sh}}^{0} - \mathbf{x}_{\text{sh}} \right\|_2 + \sum_{i=1}^{I} \left\| \mathbf{t}_{i0}^{0} - \mathbf{t}_{i0} \right\|_2 \\ \text{s.t.} \quad & \mathbf{g}_0(\mathbf{X}_0, \mathbf{P}_0) \leq 0 \\ & \mathbf{X}_0 \in [\mathbf{X}_0^l, \mathbf{X}_0^u] \\ & \mathbf{X}_0 \equiv [\mathbf{x}_0, \mathbf{x}_{\text{sh}}^{0}, \mathbf{t}_{i0}^{0}]; \ \mathbf{P}_0 \equiv [\mathbf{p}_0, \mathbf{p}_{\text{sh}}] \end{aligned} \tag{4.2}$$

New subsystem problems (lower level) i=1,…,I

$$\begin{aligned} \min_{\mathbf{X}_i} \quad & \mathbf{f}_i(\mathbf{X}_i, \mathbf{P}_i) + \left\| \mathbf{x}_{\text{sh}}^{i} - \mathbf{x}_{\text{sh}} \right\|_2 + \sum_{j \neq i}^{I} \left\| \mathbf{y}_{ij} - \mathbf{t}_{ij} \right\|_2 + \sum_{j \neq i}^{I} \left\| \mathbf{t}_{ji}^{i} - \mathbf{t}_{ji} \right\|_2 \\ \text{s.t.} \quad & \mathbf{g}_i(\mathbf{X}_i, \mathbf{P}_i) \leq 0 \\ & \mathbf{y}_{ij} = Y_i(\mathbf{X}_i, \mathbf{P}_i) \\ & \mathbf{X}_i \in [\mathbf{X}_i^l, \mathbf{X}_i^u] \\ & \mathbf{X}_{\text{i}} \equiv [\mathbf{x}_i, \mathbf{x}_{\text{sh}}^{i}, \mathbf{t}_{ji}^{i}]; \ \mathbf{P}_i \equiv [\mathbf{p}_i, \mathbf{p}_{\text{sh}}] \end{aligned} \tag{4.3}$$

where in Eq. (4.1) the objective is to minimize the differences between the coordination problem target variables and coupling variables, and the differences between the coordination problem target and all system/subsystem target variables. This objective is intended to maximize the compatibility between all system and subsystem optimization problems. Note that the coordination target variable does not have a superscript. The new system problem defined in Eq. (4.2) characterizes the original design objectives and constraints in the system. Because the objective and constraint functions in the original system problem uses the subsystem objective and constraint value as input, in the new formulation in Eq. (4.2), the subsystem objective and constraint values are included in the target variables. However, for problems with

a large number of subsystems, this transformation may incur a larger number of target variables for the coordination problem and new system problem. One way to overcome this is to simply treat these subsystem objective and constraint values as parameters. In another word, the subsystem objective and constraint values are simply passed from the subsystem optimization to the system problem, without setting target variables to match them. Also in the new system problem in Eq. (4.2), the consistency constraints are included as a series of penalty terms in the objective functions. Similarly, the subsystem problem is revised in Eq. (4.3) to enforce the consistency constraints as penalty in the objective functions.

One significant difference between the new MCO and MCO by Aute and Azarm, (2006) is that the upper-level problem in the new MCO is a single-objective optimization problem. In addition, a new system optimization problem as defined in Eq. (4.2) is added to the lower level, representing the design objectives in the original system optimization problem. Because no design objectives are included in the coordination problem, it becomes easier for the upper level in the new MCO to ensure the value of shared and target variables among the system/subsystems are eventually matched.

4.2.2 New McRO Approach

Here, the new MCO formulation is an extension of the McRO formulation. Eq. (4.4) is the coordination problem in the new McRO.

Coordination problem (upper level)

$$\begin{aligned} \min_{\mathbf{X}_0} \quad & \sum_{i=0}^{I} \left\| \mathbf{x}_{\text{sh}}^{i} - \mathbf{x}_{\text{sh}} \right\|_2 + \sum_{i=1}^{I} \sum_{j \neq i}^{I} \left\| \mathbf{y}_{ij} - \mathbf{t}_{ij} \right\|_2 + \sum_{i=1}^{I} \sum_{j \neq i}^{I} \left\| \mathbf{t}_{ij}^{i} - \mathbf{t}_{ij} \right\|_2 \\ \text{s.t.} \quad & \max \left\| \Delta \mathbf{f}^{+} \right\| \leq \eta_f ; \max \left\| \Delta \mathbf{y} \right\| \leq \eta_y ; \max[\mathbf{g}] \leq 0 \\ & \mathbf{X}_0 \in [\mathbf{X}_0^{l}, \mathbf{X}_0^{u}]; \ \mathbf{X}_0 \equiv [\mathbf{x}_{\text{sh}}, \mathbf{t}_{ij}] \end{aligned} \tag{4.4}$$

where the robustness conditions as defined earlier in Chapter 2 are now integrated in the coordination problem in the new McRO. To determine these robustness conditions, the robustness evaluation problems Eqs. (2.10-1)-(2.10-3) defined earlier in Chapter 2 are also included below for completeness:

Robustness evaluation problems (lower level)

$$\begin{aligned} & \max \left\| \Delta \mathbf{f}^{+} \right\| = \max_{\Delta \mathbf{P}_i} [\max_{i} \left\| [\mathbf{f}_i(\mathbf{X}_i, \mathbf{P}_i + \Delta \mathbf{P}_i) - \mathbf{f}_i(\mathbf{X}_i, \mathbf{P}_i)]^{+} \right\|_2], i = 1, \ldots, I \\ & \text{s.t.} \quad \Delta \mathbf{P}_i \in [\Delta \mathbf{P}_i^{l}, \Delta \mathbf{P}_i^{u}] \end{aligned} \tag{4.4-1}$$

$$\begin{aligned} & \max \left\| \Delta \mathbf{y} \right\| = \max_{\Delta \mathbf{P}_i} [\max_{i} \left\| \mathbf{y}_i(\mathbf{X}_i, \mathbf{P}_i + \Delta \mathbf{P}_i) - \mathbf{y}_i(\mathbf{X}_i, \mathbf{P}_i) \right\|_2], i = 1, \ldots, I \\ & \text{s.t.} \quad \Delta \mathbf{P}_i \in [\Delta \mathbf{P}_i^{l}, \Delta \mathbf{P}_i^{u}] \end{aligned} \tag{4.4-2}$$

$$\begin{aligned} & \max[\mathbf{g}] = \max_{\Delta \mathbf{P}_i} \{\max_{i} [\max_{k_i} \mathbf{g}_i(\mathbf{X}_i, \mathbf{P}_i + \Delta \mathbf{P}_i)]\}, i = 1, \ldots, I \\ & \text{s.t.} \quad \Delta \mathbf{P}_i \in [\Delta \mathbf{P}_i^{l}, \Delta \mathbf{P}_i^{u}] \end{aligned} \tag{4.4-3}$$

$$\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_{\text{sh}}, \mathbf{t}_{ji}^{i}]; \quad \mathbf{P}_i = [\mathbf{p}_i, \mathbf{p}_{\text{sh}}, \mathbf{t}_{ji}^{i}]$$

Fig. 4.2 shows the schematic of the new McRO approach. While the coordination problem in Eq. (4.1) is solved at the upper level, the system and subsystem problems as defined in Eqs. (4.2) and (4.3) are solved at the lower level. Notice that solving the system problem in the new McRO is essentially the same as solving all the other subsystem optimization problems. The procedure of applying new McRO is similar to McRO. It starts with solving the coordination problem where each candidate design alternative (including shared and target variables) is forwarded as a parameter to the system/subsystem problems in the lower level. After these optimization problems are

solved in the lower level, robustness of the candidate design alternative is evaluated in the lower level. Based on the system/subsystem response and robustness evaluations, the objective function value in the coordination problem (upper level) is determined.

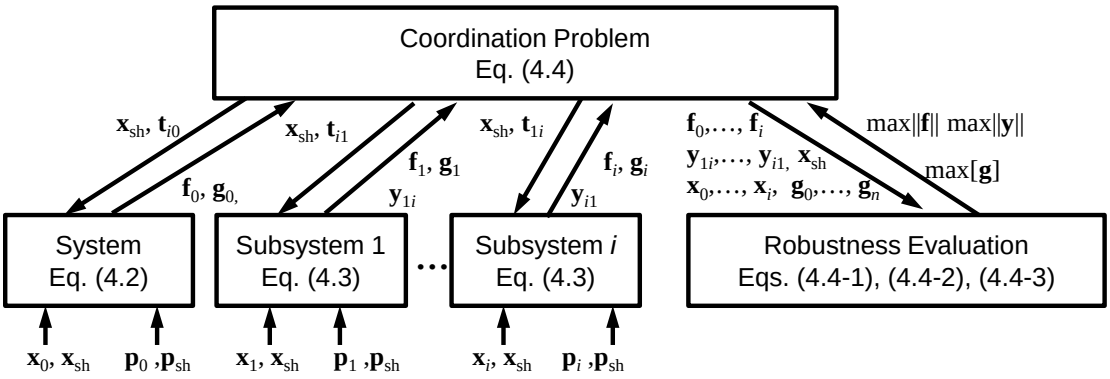


Fig. 4.1 New McRO

To resolve the issue of an arbitrary choice of a subsystem optimum solution in McRO, the selection of a system and subsystem optimal solution in the new McRO is improved to take into consideration the compatibility of the system/subsystem optimal solutions. At the conclusion of system and subsystem optimization in the lower level, the penalty terms as defined, i.e., $\left\|\mathbf{x}_{\text{sh}}^{i}-\mathbf{x}_{\text{sh}}\right\|_2+\sum_{j\neq i}^{I}\left\|\mathbf{y}_{ij}-\mathbf{t}_{ij}\right\|_2+\sum_{j\neq i}^{I}\left\|\mathbf{t}_{ji}^{i}-\mathbf{t}_{ji}\right\|_2$ is calculated for all the optimal solutions. The solutions with the smallest value of the penalty give the best consistency among an obtained set of optimal solutions. Selecting the best compatible solutions facilitate the convergence of the coordination problem at the upper level.

As presented above, the new McRO approach has three major improvements compared to McRO, namely, transformation of the system optimization problem, conversion of consistency constraints to penalty terms, and compatibility-based

selection. As it will be shown in the examples, these improvements help obtain optimal design solutions much more efficiently.

4.2.3 Discussions

From an implementation perspective, the optimum system design solutions in the new MCO and McRO approaches are obtained from the system optimization problem in the lower level. This is different from the previous approaches where the system design solutions are obtained from the system optimization problem in the upper level. Since the coordination problem is posed as a single-objective optimization problem in the new approaches, it will converge to a single design point. However, some problems may have multiple optimum solutions or converging points. Each of such points corresponds to a consistent solution for the shared and coupling variables among the subsystems. When solving such a problem, it is recommended to use a global optimization approach, e.g., genetic algorithm (Goldberg 1989), for the coordination problem so that multiple optimum points can be explored. In this case, the system optimization problem at the lower level may generate different sets of Pareto optimal solutions. These Pareto optimal solutions should then be combined and sorted, in order to obtain a final set of optimum system design solutions.

By comparing McRO in Fig. 2.5 and new McRO in Fig. 4.1, it may appear that the new McRO is more complex than McRO. Indeed, the total number of function calls required by the lower level in the new McRO is slightly more than McRO. However, the total number of function calls for the new McRO equals to the multiplication of the upper-level and lower-level function calls. Because the number of function calls for the added subproblem is of the same order of magnitude as the other subsystem

problems, the system optimization does not increase the order of magnitude of the lower-level function calls in the new McRO. In this way, the computational cost for McRO and new McRO is essentially comparable. One interesting observation from the results of the examples is that by applying the new McRO, the upper-level (coordination) problem requires fewer number of iterations to converge than it does with McRO. This is because the coordination problem in the new McRO is more efficient in obtaining optimum solutions as discussed earlier. More discussions on the comparison of the number of function calls between MCO and new MCO are presented with the examples in Section 4.4.

4.3 AA-McRO Approach

To address the computational difficulty associated with McRO, an online approximation method, similar to the one used in AA-MORO in Chapter 3, is used in the new McRO. However, the online approximation has to be extended to fit the framework of the new McRO with multiple subsystems. As shown in Fig. 4.2, the new AA-McRO approach includes an online Design of Experiment (DOE) and metamodeling, in which the metamodels of subsystem analysis models (which might be computationally expensive) are developed. These metamodels are forwarded to the subsystem optimization and robustness evaluation problems. Meanwhile, the online DOE locates sample points based on the optimum solutions from subsystem optimization and robustness evaluation. These sample points are collected and used to iteratively update metamodels for the design objectives and constraint functions. In the subsystem optimization problem, multiobjective optimization is performed by each subsystem optimizer based on the subsystem metamodels. The multiobjective

optimal solutions are returned back to the online DOE as additional sample points. During robustness evaluation, any function evaluation is accomplished based on the subsystem metamodels. Finally, the robustness evaluation as defined in Eq. (4.4-1)-(4.4-3) returns optimal solutions for additional sampling as well.

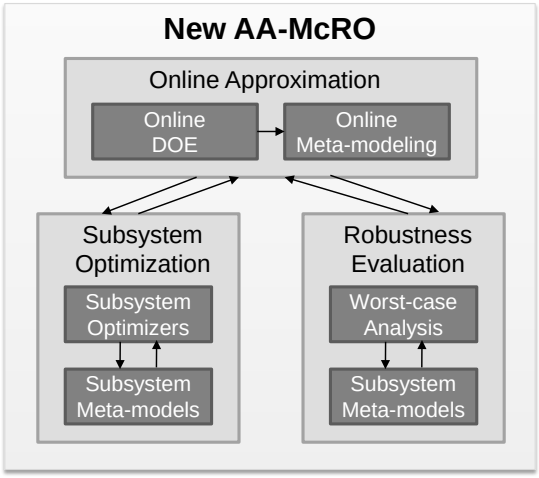


Fig. 4.2 Schematic of online approximation in the proposed approach

4.3.1 Online Approximation

The online approximation approach applied for AA-McRO is similar to the approximation approach in AA-MORO presented in Chapter 3. The objective of online DOE is to select points in the sample space so that they are as close to the optimum solutions as possible. The majority of the sample points are collected from the optimal solutions obtained in the subsystem optimization during online approximation. The online approximation starts with an initial set of sample points which are created using Maximum Entropy Design (MED) (Koehler and Owen, 1996). These sample points are then forwarded to each subsystem analysis model where their actual design objectives and constraint functions are evaluated. Next, the sample points are used to build metamodels for each design objectives and constraint

functions in the subsystems. Additional online sample points are determined based on optimal solution points estimated from the approximation-assisted optimization. For the sample points, the true design objectives and constraint functions are evaluated and used to update the subsystem metamodels. This is repeated in all subsystems.

A subsystem sample space in AA-McRO is defined over the entire variables and parameters space. For all "online" sample points, the shared variables are determined by the system level optimizer, which passes their values to the subsystems. During the online DOE, sample points are generated in two steps from the subsystem optimization and robustness evaluation. The first set of sample points is determined after subsystem optimization is completed, i.e., the value of the subsystem variables and target variables for these sample points are the optimum solutions in each subsystem. Note that the nominal value of uncertain parameters is used for these sample points. The second set of online sample points are determined during the robustness evaluation, i.e., the optimized uncertain parameter $\Delta\mathbf{P}_i^*$ is used as the values of the uncertain parameter. For these sample points, their subsystem variables and target variables is the same as the selected optimum solution.

Since the new McRO involves design iterations, there can be many optimal solutions generated from the subsystem optimization problems. Using the online sampling scheme above, some sample points may overlap with existing sample points. Therefore, a filtering scheme as discussed in Chapter 3 is applied in AA-McRO. After additional sample points are determined and the actual simulations are evaluated to obtain the response values, these sample points are added to the current set of sample points. Consequently, the new sample set is used to update the metamodels.

A metamodel can be developed based on one of the many existing techniques. As highlighted in Chapter 2, AA-McRO uses a Kriging technique (Koehler and Owen, 1996). Kriging estimates the Mean Squared Error (*mse*) when interpolating a response at an unobserved design point. This error is considered and discussed next.

In AA-McRO, a design point with a smaller error is preferred than a design point with a large error, when their estimated function values are the same. This is because the proposed online approximation is intended to focus on sampling more points around the optimal solutions. The $mse(\mathbf{x}^*)$ defined in Eq. (2.16) is used in adjusting the estimated objective and/or constraint function values so that a design point with a large error is not considered and automatically eliminated by the optimizer. In doing so, an estimate of design objective or constraint functions is adjusted by its *mse* as follows:

$$\begin{aligned} f_{\text{adjust}}(\mathbf{x}^*) &= f_{\text{estimate}}(\mathbf{x}^*) + mse(\mathbf{x}^*) \\ g_{\text{adjust}}(\mathbf{x}^*) &= g_{\text{estimate}}(\mathbf{x}^*) + mse(\mathbf{x}^*) \le 0 \end{aligned} \tag{4.5}$$

where f_{estimate} and g_{estimate} represents the estimated design objectives and constraint functions from Kriging, f_{adjust} and g_{adjust} represents the adjusted design objectives and constraint functions, respectively. According to Eq. (4.5), a design point with a large *mse* leads to larger adjusted design objectives and constraint functions since the term *mse* ($\mathbf{x}^*$) is always positive according to its definition in Eq. (2.16). As new McRO attempts to minimize design objectives subject to the feasibility of the constraint functions, the adjustment in Eq. (4.5) will always give more preference to a design alternative with a smaller error. Therefore, the design point with a large *mse* is less likely to be selected and automatically eliminated by the optimizer. In

implementation, the estimated function values and $mse(\mathbf{x}^*)$ should be normalized so that all terms on the right hand side of Eq. (4.5) are in scale. In the examples of this chapter, the maximum estimated function values and $mse(\mathbf{x}^*)$ are used for normalization of the estimated design objectives, constraint functions and $mse(\mathbf{x}^*)$ in Eq. (4.5).

Since the new AA-McRO applies online approximation-assisted optimization, its optimum solutions are essentially "estimated" values. To ensure the new AA-McRO solutions are optimum and feasible with respect to the actual objective and constraint functions, these solutions should be verified. In the new AA-McRO, this validation step is integrated as a part of the stopping criteria. Basically, when the new AA-McRO has satisfied all stopping criteria (e.g., total number of samples are exhausted, as discussed later in Section 4.4.2), the current set of optimum solutions is observed and the maximum absolute error (MAE) for each objective and constraint functions are calculated. The MAE is calculated using the following equation: $\mathrm{MAE} = \max(|f_1 - f_{1,estimate}|,...,|f_{nf} - f_{nf,estimate}|,|g_1 - g_{1,estimate}|...,|g_{ng} - g_{ng,estimate}|)$ where m and n represent the number of objective and constraint functions, respectively. If the error is larger than a user specified threshold, e.g., 5% of the nominal value, additional iterations (samples) are added to refine the metamodel. This validation step effectively improves the accuracy of the new AA-McRO.

4.3.2 New AA-McRO Solution Steps

The new AA-McRO begins with a number of sample points generated using MED for each subsystem. Based on these samples, metamodels for design objectives and constraints of each subsystem are constructed in the design variable and uncertain

parameter space. These metamodels are used in lieu of the actual subsystem analysis models to obtain robust optimum solutions and to evaluate robustness of candidate solutions. Notice that a metamodel can also be constructed for a coupling variable if it is defined as a computationally expensive function of subsystem variables. In the following iterations, additional new samples are placed based on the optimum solutions obtained in each iteration, combined with a filtering strategy to eliminate overlapping of sample points as discussed earlier. The metamodels from previous iterations are updated using the new samples. In this way, as the system iteratively converges to optimum solutions, the sample points converge to the optimum design region and produce better prediction of objective and constraint function values. The steps in the new AA-McRO are as follows:

Step 1: Generate an initial set of sample points in each subsystem. Use the subsystems analysis models to calculate the design objectives and constraint functions for the sample points.

Step 2: Build metamodels for each subsystem's design objectives and constraint functions using the sample points from Step 1.

Step 3: Perform single-objective optimization for the coordination problem in Eq. (4.1). For each candidate design alternative, send the value of the shared and target variables $\mathbf{x}_{\text{sh}}$ and $\mathbf{t}_{ij}$ as parameters to the system and subsystem problems in the lower level.

Step 4: For each upper-level candidate design alternative, perform multiobjective optimization in system and subsystem problems based on the metamodels.

Step 5: Evaluate the robustness for the selected system and subsystem design solutions based on the metamodels. Repeat steps 3-5 until all candidate design alternatives are considered.

Step 6: Collect the system and subsystem optimal design solutions and the optimized $\Delta\mathbf{P}_i$ value, or $\Delta\mathbf{P}_i^*$, in the robustness evaluation. Perform online DOE, add the new samples and update the metamodels.

Step 7: Check the following stopping criteria (i) and (ii). If the stopping criterion (i) is not satisfied, return to Step 3. If the stopping criterion (i) is satisfied but (ii) is not, then increase the maximum number of iterations or sample points and return to Step 3. Otherwise, stop and report the obtained optimum solutions.

The stopping criteria are: (i) a pre-specified maximum number of iterations or sample points have been reached, (ii) the absolute error for the solutions are smaller than a user defined threshold.

4.4 Examples

To demonstrate the new AA-McRO approach, two examples are presented in this section. The first example is a numerical problem with two subsystems, which is used to illustrate how the new AA-McRO approach works. The second example is an engineering problem which contains three subsystems and is more complex than the numerical example. In both examples, the real-coded Genetic Algorithm (GA) of MATLAB^{TM}, version 2010a (Mathwork, 2010) is used as the optimizer for both system and subsystem optimization problems. The default setting in the MATLAB's genetic algorithm is used. The population size of the genetic algorithm is set equal to 15×*nx* (*nx* denotes the number of variables) and the maximum number of generations

is 50. The elite number in GA is set equal to 1 when the number of variables is less than five and 2 otherwise. The probability for crossover and mutation is 0.9 and 0.1, respectively. In both examples, several related approaches, including MCO (Aute and Azarm, 2006), McRO, AA-McRO (Hu et al., 2012b), new MCO, new McRO, and new AA-McRO are compared. For each example, the deterministic optimization approaches, i.e., MCO and new MCO are first compared. Then, the comparison among four robust optimization approaches including McRO, AA-McRO, new McRO, and new AA-McRO are conducted. These comparisons are intended to show the performance and efficiency of the new approaches in view of the previous methods. Due to the stochastic nature of GA, all approaches are repeated for ten times. Because the space limitation, the optimum solutions from an average run are shown in subsections 4.5.1 and 4.5.2. It should be mentioned that for the metamodeling in the AA-McRO approaches, Kriging is used with a second order polynomial function to build the regression model and a Gaussian function used for the correlation model. For a fair comparison, the same number of total sample points is used in both AA-McRO and new AA-McRO approaches. This is done so that it can be observed which method gives a better estimate of solution for the same number of function calls. In addition, for the verification of the results, an acceptable threshold for the absolute errors on the optimum solutions is set equal to 5% of the nominal value for both examples.

4.4.1 Illustrative Example

The first example was used as a test case in a previous work (Aute and Azarm, 2006). In the "All-at-once" formulation, Eq. (4.6), the example has six design variables with two objectives and six inequality constraints:

All-at-once formulation

$$\begin{aligned}
&\min\ f_1 = -[25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 + (x_5-1)^2] \\
&\min\ f_2 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\
&\text{s.t. } g_1 = 2 - x_1 - x_2 \le 0 \\
&\quad g_2 = x_1 + x_2 - 6 \le 0 \\
&\quad g_3 = x_2 - x_1 - 2 \le 0 \\
&\quad g_4 = x_1 - 3x_2 - 2 \le 0 \\
&\quad g_5 = (x_3-3)^2 + x_4 - 4 \le 0 \\
&\quad g_6 = 4 - (x_5-3)^2 + x_6 \le 0 \\
&\quad 0 \le x_1, x_2, x_6 \le 10, 1 \le x_3, x_5 \le 5, 0 \le x_4 \le 6
\end{aligned} \tag{4.6}$$

The above problem can be divided into two subproblems and formulated based on a new McRO with an upper-level coordination problem, one system problem and two subsystem and robustness problems, as in the following:

Coordination problem

$$\begin{aligned}
&\min\ f = \sum_{i=0}^{2} \left\| \mathbf{x}_{\text{sh}}^i - \mathbf{x}_{\text{sh}} \right\|_2 + \sum_{i=0}^{2} \sum_{j=1}^{2} \left\| \mathbf{t}_{ji}^i - \mathbf{t}_{ji} \right\|_2 + \left\| y_{21} - t_{21} \right\|_2 \\
&\text{s.t. } \max \left\| \Delta \mathbf{f}^+ \right\| \le \eta_f ; \max[\mathbf{g}] \le 0 \\
&\quad \mathbf{x}_{\text{sh}} \equiv [x_1, x_2], t_{21} \equiv x_5, \mathbf{t}_{10} \equiv [f_{11}, f_{12}], \mathbf{t}_{20} \equiv [f_{21}, f_{22}] \\
&\quad 0 \le x_1, x_2 \le 10, 1 \le x_5 \le 5
\end{aligned} \tag{4.7}$$

System problem

$$\begin{aligned}
&\min\ f_1 = f_{11} + f_{21} + \left\| \mathbf{x}_{\text{sh}}^0 - \mathbf{x}_{\text{sh}} \right\|_2 + \left\| \mathbf{t}_{10}^0 - \mathbf{t}_{10} \right\|_2 + \left\| \mathbf{t}_{20}^0 - \mathbf{t}_{20} \right\|_2 \\
&\min\ f_2 = f_{12} + f_{22} + \left\| \mathbf{x}_{\text{sh}}^0 - \mathbf{x}_{\text{sh}} \right\|_2 + \left\| \mathbf{t}_{10}^0 - \mathbf{t}_{10} \right\|_2 + \left\| \mathbf{t}_{20}^0 - \mathbf{t}_{20} \right\|_2 \\
&\text{s.t. } g_1 = 2 - x_1 - x_2 \le 0 \\
&\quad g_2 = x_1 + x_2 - 6 \le 0 \\
&\quad g_3 = x_2 - x_1 - 2 \le 0 \\
&\quad g_4 = x_1 - 3x_2 - 2 \le 0 \\
&\quad \mathbf{x}_{\text{sh}}^0 \equiv [x_1, x_2], \mathbf{t}_{10}^0 \equiv [f_{11}, f_{12}], \mathbf{t}_{20}^0 \equiv [f_{21}, f_{22}] \\
&\quad 0 \le x_1, x_2 \le 10
\end{aligned} \tag{4.8}$$

Subsystem 1 problem

$$
\begin{aligned}
&\min\ f_{11} = -[12.5(x_1-2)^2 + 0.5(x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2] \\
&\qquad + \left\|\mathbf{x}_{\text{sh}}^1 - \mathbf{x}_{\text{sh}}\right\|_2 + \left\|t_{21}^1 - t_{21}\right\|_2 + \left\|\mathbf{y}_{10}^1 - \mathbf{t}_{10}\right\|_2 \\
&\min\ f_{12} = 0.5(x_1^2 + x_2^2) + x_3^2 + x_4^2 + x_5^2 \\
&\qquad + \left\|\mathbf{x}_{\text{sh}}^1 - \mathbf{x}_{\text{sh}}\right\|_2 + \left\|t_{21}^1 - t_{21}\right\|_2 + \left\|\mathbf{y}_{10}^1 - \mathbf{t}_{10}\right\|_2 \\
&\text{s.t. } g_5 = (x_3-3)^2 + x_4 - 4 \le 0 \\
&\quad \mathbf{x}_{\text{sh}}^1 \equiv [x_1, x_2], t_{21}^1 \equiv x_5, \mathbf{y}_{10}^1 \equiv [f_{11}, f_{12}], \\
&\quad 0 \le x_1, x_2 \le 10, 1 \le x_3, x_5 \le 5, 0 \le x_4 \le 6
\end{aligned}
\tag{4.9}
$$

Subsystem 2 problem

$$
\begin{aligned}
&\min\ f_{21} = -[12.5(x_1-2)^2 + 0.5(x_2-2)^2 + (x_5-1)^2] \\
&\qquad + \left\|\mathbf{x}_{\text{sh}}^2 - \mathbf{x}_{\text{sh}}\right\|_2 + \left\|y_{21} - t_{21}\right\|_2 + \left\|\mathbf{y}_{20} - \mathbf{t}_{20}\right\|_2 \\
&\min\ f_{22} = 0.5(x_1^2 + x_2^2) + x_6^2 \\
&\qquad + \left\|\mathbf{x}_{\text{sh}}^2 - \mathbf{x}_{\text{sh}}\right\|_2 + \left\|y_{21} - t_{21}\right\|_2 + \left\|\mathbf{y}_{20} - \mathbf{t}_{20}\right\|_2 \\
&\text{s.t. } g_6 = 4 - (x_5-3)^2 + x_6 \le 0 \\
&\quad \mathbf{x}_{\text{sh}}^2 \equiv [x_1, x_2], y_{21} = x_5, \mathbf{y}_{20} \equiv [f_{21}, f_{22}] \\
&\quad 0 \le x_1, x_2, x_6 \le 10, 1 \le x_5 \le 5
\end{aligned}
\tag{4.10}
$$

Robustness evaluation problems

$$
\begin{aligned}
&\max \left\|\Delta\mathbf{f}^+\right\|_2 = \max_{\Delta\mathbf{x}}[\max_{i,j} \left\|[f_{ij}(\mathbf{x}+\Delta\mathbf{x}) - f_{ij}(\mathbf{x})]^+\right\|_2], i, j = 1,2 \\
&\text{s.t. } \Delta\mathbf{x} \in [-10\%\mathbf{x}, 10\%\mathbf{x}]; \Delta\mathbf{x} \equiv [\Delta x_3, \Delta x_6] \\
&\quad f_{11}(\mathbf{x}) = -[12.5(x_1-2)^2 + 0.5(x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2] \\
&\quad f_{12}(\mathbf{x}) = 0.5(x_1^2 + x_2^2) + x_3^2 + x_4^2 + x_5^2 \\
&\quad f_{21}(\mathbf{x}) = -[12.5(x_1-2)^2 + 0.5(x_2-2)^2 + (x_5-1)^2] \\
&\quad f_{22}(\mathbf{x}) = 0.5(x_1^2 + x_2^2) + x_6^2
\end{aligned}
\tag{4.11-1}
$$

$$
\begin{aligned}
&\max[\mathbf{g}] = \max_{\Delta\mathbf{x}}[\max_k g_k(\mathbf{x} + \Delta\mathbf{x})], k = 1,\ldots,6 \\
&\text{s.t. } \Delta\mathbf{x} \in [-10\%\mathbf{x}, 10\%\mathbf{x}]; \Delta\mathbf{x} \equiv [\Delta x_3, \Delta x_6] \\
&\quad g_1(\mathbf{x}) = 2 - x_1 - x_2 \le 0 \\
&\quad g_2(\mathbf{x}) = x_1 + x_2 - 6 \le 0 \\
&\quad g_3(\mathbf{x}) = x_2 - x_1 - 2 \le 0 \\
&\quad g_4(\mathbf{x}) = x_1 - 3x_2 - 2 \le 0 \\
&\quad g_5(\mathbf{x}) = (x_3-3)^2 + x_4 - 4 \le 0 \\
&\quad g_6(\mathbf{x}) = 4 - (x_5-3)^2 + x_6 \le 0
\end{aligned}
\tag{4.11-2}
$$

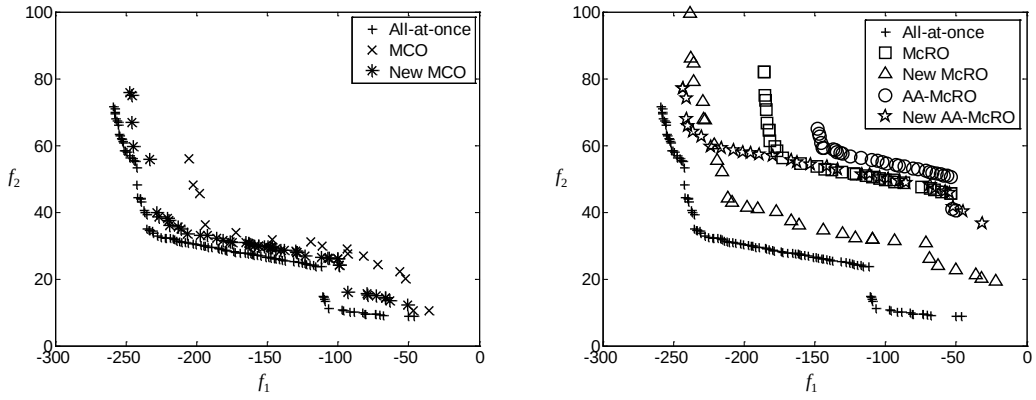
The hierarchical relationship among the above problems follows Fig. 2. In other words, the coordination problem in Eq. (4.7) is at the upper level. The system problem in Eq. (4.8) has four constraints from the original problem in Eq. (4.6) and its objective function is formulated as the summation of subsystem 1 and subsystem 2 objective values. Each of the two subsystems in Eqs. (4.9) and (4.10) has one constraint from the original problem in Eq. (4.6) and their objective functions are part of the original objective function as well. The variables x_1 and x_2 are the two shared variables in this example, and x_5 is a coupling variable which goes from subsystem 2 to subsystem 1. Additionally, all subsystem objective function such as f_{11}, f_{12}, f_{21} and f_{22} are treated as coupling variables under the new collaborative optimization framework. Also notice that in the system and subsystem problems, the compatibility of shared and coupling variables with the coordination level variables are enforced by using penalty terms. As mentioned earlier, these penalty terms allow the system and subsystem optimization to search the design space better. However as the optimization proceeds, these penalty term must be minimized so essentially the compatibility requirement (e.g., $\left\|\mathbf{x}_{\text{sh}}^0 - \mathbf{x}_{\text{sh}}\right\|_2 = 0; \left\|\mathbf{t}_{10}^0 - \mathbf{t}_{10}\right\|_2 = 0; \left\|\mathbf{t}_{20}^0 - \mathbf{t}_{20}\right\|_2 = 0$ for system problem) can be satisfied when the problem converges. The robustness evaluation problems, Eqs. (4.11-1) and (4.11-2), consider the interval uncertainty in the two design variables x_3, x_6. It is assumed that the range of uncertainty is within +/-10% of their nominal values. The acceptable variation limit η_f for the objective functions is 20% of the nominal value. Notice that objective robustness of Eq. (4.11-1) is only considered for the two subsystems. This is because in this example, the system objective is a simple summation of the subsystem objectives and there is no need to

evaluate the objective robustness for the system problem separately. Also because the coupling variable x_5 in subsystem 2 does not have any uncertainty, the robustness on the coupling variable is not considered.

The above problem is first solved using the MCO and also the new MCO. The formulation for the problem with the new MCO approach is given in Eqs. (4.7)-(4.10), except that the robustness conditions in Eq. (4.8) should be ignored. The formulation with the MCO approach can be found in Aute and Azarm (2006) and is omitted here. The system Pareto optimum solutions are plotted in the objective space as in Fig. 4.3 (a) for one of the ten runs. The all-at-once problem as defined by Eq. (4.6) is also solved using a multiobjective genetic algorithm. Because the all-at-once formulation is a single-disciplinary optimization and less restricted (relaxed compared to MCO with multiple subsystem), its solutions are typically better than (or in the worst case the same as) any decomposition-based (or MCO) optimization approaches. The all-at-once solutions are used as a baseline for the comparison. As shown in Fig. 4.3 (a), the new MCO solutions are slightly closer to the baseline solutions in the objective space than the MCO solutions. The spread of the new MCO solutions seems to be better than the MCO solutions as well.

Table 4.1 summarizes number of function calls, the mean and standard deviation of two quality metrics for the MCO and new MCO solutions. The quality metrics considered for the comparison are the Hyperarea Difference (HD) and Overall Spread (OS) (Wu and Azarm, 2001). While HD measures the closeness (the smaller the better) of the non-dominated points to the ideal (or good) solutions, the quality metric OS characterizes the spread (the larger the better) of the set of non-dominated points

in the objective space. Note that a function call here refers to evaluating a combined set of objective and/or constraint functions for the given input variables. For example, one function call in the subsystem 1 problem of Eq. (4.9) implies a single evaluation of that subsystem's objective functions f_{11} and f_{12} and constraint g_5 altogether. The function calls for all subsystems are added up to obtain the overall number of function calls. As shown in Table 1, the average number (based on ten runs) of function calls are approximately 1.5×10^7 and 1.2×10^7 for MCO and new MCO approaches, respectively. As mentioned earlier in Section 3.3, the lower-level problems in the new MCO require slightly more function calls because of the addition of the system problem at the lower level. However, the upper-level (coordination) problem in the new MCO requires fewer number of iterations than the upper-level (system) problem in MCO. As a result the total number of function calls in the new MCO is actually less than MCO.



(a) (b)

Fig. 4.3 Pareto optimum solutions for numerical example
(a) deterministic solutions (b) robust solutions

Next, the robust collaborative optimization approaches, including McRO, new McRO, AA-McRO and new AA-McRO are applied to the numerical example. The McRO

formulation uses the MCO formulation (Aute and Azarm, 2006) with an addition of a robustness evaluation as in Eq. (4.7) in the lower level. The formulation for the new McRO is given in Eq. (4.7)-(4.11), while the new AA-McRO is formulated the same way as the new McRO except that all objective and constraint function evaluations are approximated following the steps given in Section 3.4. The system Pareto optimum solutions from all these different approaches are shown in the objective space and compared against the baseline solutions in Fig. 4.3 (b). In terms of closeness to the baseline solution, the performance of the new McRO is significantly better than McRO. Most importantly, the Pareto optimum solutions from the new AA-McRO are better (in terms of closeness to the baseline deterministic solution) than the solutions from AA-McRO for the same number of sample points. The spread of the new McRO and AA-McRO solutions are also clearly better than the McRO and AA-McRO solutions. Similar observations are found with other runs and Table 1 summarizes the average value of the closeness (HD) and spread (OS) for the ten runs using different approaches. In terms of computational cost, the numbers of function calls by McRO and new McRO are approximately 1.9×10^7 and 1.3×10^7, while the total number of sample points (one sample can be regarded as one function call) for both AA-McRO approaches are 360. Note that both AA-McRO and new AA-McRO use the same number of sample points for metamodeling and thus their function calls are the same. However, by comparing the optimum solutions in Fig. 4.3, it can be seen clearly that new AA-McRO performs better than AA-McRO.

It should be mentioned that since the new AA-McRO and AA-McRO use approximation-assisted optimization, the optimum solutions are essentially estimated

from the metamodeling. As discussed earlier, in the new AA-McRO the MAE for the objective and constraint functions of these optimum solutions are verified so that they are within a 5% limit of the nominal objective and constraint function values. Notice that in AA-McRO the final optimum solutions are also verified except that this step is performed after the optimum solutions are obtained. For the numerical example, the error of the optimum solution from AA-McRO is within 5% of the nominal values.

Table 4.1 Number of function calls and quality metrics for numerical and engineering examples

	Approach	Average number Function call	HD mean (Std.)	OS mean (Std.)
Illustrative example	MCO	1.5×10^7	0.45 (0.02)	0.34 (0.10)
	New MCO	1.2×10^7	0.41 (0.02)	0.42 (0.20)
	McRO	1.9×10^7	0.73 (0.10)	0.12 (0.03)
	New McRO	1.3×10^7	0.46 (0.03)	0.41 (0.20)
	AA-McRO	360*	0.82 (0.20)	0.09 (0.02)
	New AA-McRO	360*	0.50 (0.03)	0.38 (0.20)
Angel grinder example	MCO	6.5×10^7	0.38 (0.23)	0.19 (0.01)
	New MCO	2.5×10^7	0.29 (0.06)	0.38 (0.10)
	McRO	9.0×10^7	0.55 (0.10)	0.17 (0.06)
	New McRO	3.3×10^7	0.42 (0.08)	0.28 (0.08)
	AA-McRO	1350*	0.64 (0.16)	0.11 (0.05)
	New AA-McRO	1350*	0.48 (0.08)	0.42 (0.10)

*The number of function calls for AA-McRO and new AA-McRO refers to the number of sample points

Finally, in order to verify the robustness of the optimum solutions, a Monte Carlo simulation method is applied by checking the system responses (including objective, constraint functions) at a large number of random points around each of the optimum solution points. The total number of randomly generated points used for robustness verification is 10^4. Only five out of the 10^4 random points are allowed to violate the robustness conditions as specified in Eq. (4.7). However, it is found that the optimum

solutions from McRO, AA-McRO, new McRO and new AA-McRO approaches as shown in Fig. 4.3 (b) satisfy these robustness conditions.

4.4.2 Angle Grinder Example

To demonstrate the applicability of the new AA-McRO approach to an engineering design problem, a cordless angle grinder is considered in the second example (Li et al., 2010c). Fig. 4.4 shows the three major subsystems of an angle grinder model considered in this example. The details of the system and subsystem optimization specifics, including the number of functions and variables, are shown in Table 4.2.

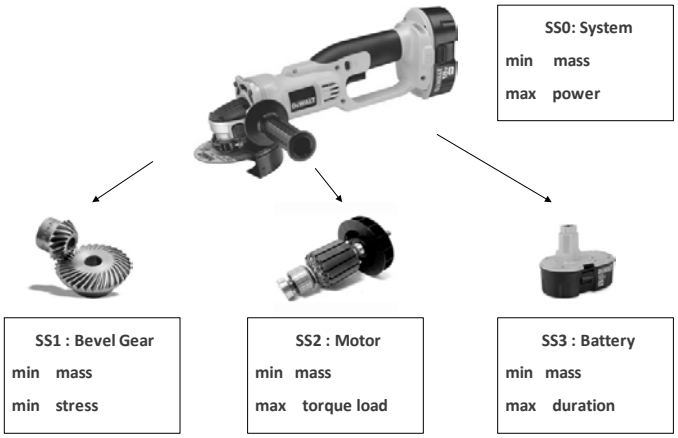


Fig. 4.4 System decomposition of angle grinder

The three subsystems are bevel gear, motor and battery pack. The system design objective is to minimum total mass and maximum output power. While each of the subsystems has its own design variables, four variables are shared among the system and subsystems namely, current (I), the number of battery cells (n_{cell}), motor-gear shaft diameter (d_{s}) and motor-gear shaft length (l_{s}). The first two design variables (I and n_{cell}) are shared by the system (SS0), motor (SS2) and battery (SS3); and the last two design variables (d_{s} and l_{s}) are shared by the bevel gear (SS1) and motor (SS2).

There is also a coupling variable σ_{load} (motor output torque at loaded condition) which is an output from the motor subsystem and input to the bevel gear subsystem. The details of the system and subsystem design variables can be found in Li et al. (2010c). Notice that in this example each subsystem also has its own design objectives.

In the angle grinder example, a large number of parameters exists and many of them are uncertain. These uncertain parameters are shown in Table 4.3. Notice that a range of [-10%, 10%] for the uncertainty interval is specified. The acceptable variation range for the objective functions and coupling variables are each 5% from their nominal values.

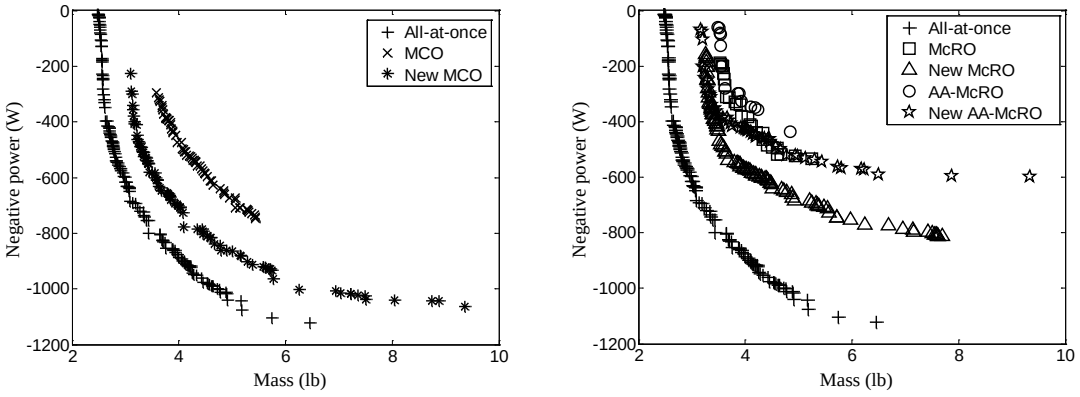
Table 4.2 System and subsystem optimization problem specifics

	Subsystem0 (system)	**Subsystem1** (gear)	**Subsystem2** (motor)	**Subsystem3** (battery)
# of objective functions	2	2	2	2
# of constraint functions	2	3	10	6
# of design variables	5	2	6	5
# of uncertain parameters	1	1	3	4
# of target variables	1	1	0	0

Table 4.3 Uncertain parameters in angle grinder example

Parameter	**Desciprtion**	**Nominal value**	**Uncertain interval**
V_{cell}	nominal cell voltage (V)	1.2	[-10%, 10%]
$r_{cellmax}$	max battery cell radius(mm)	100	[-10%, 10%]
$h_{cellmax}$	max battery cell height(mm)	200	[-10%, 10%]
R_w	20 awg copper wire resistivity (ohms/m)	0.036	[-10%, 10%]
A_w	20 awg copper wire cross-sectional area (mm^2)	0.504	[-10%, 10%]
M_{ext}	mass of misc components (kg)	1.355	[-10%, 10%]
α	brush loss factor	2	[-10%, 10%]
w_f	face width in meters (m)	0.008	[-10%, 10%]
t_{cell}	battery cell wall thickness (mm)	0.2	[-10%, 10%]

As in the first numerical example, two deterministic optimization approaches are first applied to the angle grinder model. The optimal solutions from new MCO are compared with those from MCO as shown in Fig. 4.5 (a). In terms of closeness to the baseline (all-at-once) optimal solutions and the spread in the objective space, the new MCO solutions are better than the MCO solutions. The same trend can be observed from Table 4.1 in which the average values for closeness (HD) and spread (OS) from the new MCO are better than MCO. The average number (based on ten runs) of function calls for MCO and new MCO approaches are approximately 6.5×10^7 and 2.5×10^7 respectively.



(a) (b)

Fig. 4.5 System Pareto optimum solutions for angle grinder example
(a) deterministic solutions (b) robust solutions

Next, the robust optimization approaches are applied to the angle grinder example with the uncertainty specified in Table 4.3. The robust optimum solutions from McRO, AA-McRO, new McRO and new AA-McRO are compared with the baseline solutions in Fig. 4.5 (b). It is observed that the solutions from new McRO slightly dominate the solutions from McRO, and the optimum solutions from the new AA-McRO are better than AA-McRO solutions in terms of the closeness to the baseline

solution. Moreover, the optimum solutions from the new McRO and AA-McRO cover a wider range than the previous approaches. The number of function calls for different McRO approaches can be found in Table 4.1. Again, as in the numerical example, the solutions are verified with a 5% error threshold. The average (mean) values of the quality metrics from all test runs with different approaches are again compared and shown in Table 4.1. The values of quality metrics are consistent with the results in Fig. 4.5. Finally, the robustness for the optimum solutions from all solution approaches is verified as in the numerical example.

4.5 Summary

A new approximation-assisted McRO (new AA-McRO) approach is developed in this Chapter. The new AA-McRO significantly improves the convergence (based on numerical evidence) and overcomes computational difficulties in a previously developed McRO (Li and Azarm, 2008) and AA-McRO (Hu et al., 2012b). The new AA-McRO enhances the convergence by transforming the multiobjective system problem at the upper level into a single-objective upper-level coordination problem and a multiobjective lower-level optimization problem. This transformation allows the new AA-McRO achieve convergence more efficiently than the previous approaches. To mitigate the computational cost, the new AA-McRO replaces subsystem analysis outputs which are used for calculating design objectives and constraint functions with approximations. The new AA-McRO also verifies the final optimum solutions until the absolute error falls under a user-specified threshold. Two examples are used to demonstrate the applicability of the proposed new AA-McRO and compare it with several related approaches. It is found that the new MCO, new

McRO and new AA-McRO converge faster than the previous MCO, McRO and AA-McRO approaches. This is attributed to the single-objective problem dedicated to coordinating the shared and coupling variables. Both the AA-McRO and new AA-McRO require significantly fewer number of function calls and thus more efficient than the other CO approaches that do not use approximation. In both examples, the total number of AA-McRO sample points is several orders of magnitude smaller than the number of function calls required by the McRO and new McRO approach. However, the optimal solutions from the new AA-McRO are better than the AA-McRO solutions in terms of closeness to the optimum Pareto frontier and spread in the objective space. The robustness of final solutions in McRO, AA-McRO, new McRO and new AA-McRO was verified using a Monte Carlo simulation and found to be acceptable. Although the performance of the new AA-McRO is attractive for the examples, the approach requires additional coupling variables compared to the previous approaches because the original system optimization is moved to the lower level. In the case in which a problem has many coupling variables between the subsystems and system problem, this move could create many target variables in the coordination problem which can increase the problem size compared to AA-McRO. On the other hand, although the online approximation in AA-McRO provides a good estimate of the predicted functions according to the results in both examples, the approximation accuracy is dependent upon the approximation method used.

In the next chapter, a two-stage decision support system (DSS) is developed to integrated business and engineering decisions under uncertainty for an oil refinery case study based on the AA-MORO approach of Chapter 3.

Chapter 5 Integration of Engineering and Business Decisions in Oil Refineries Using AA-MORO and Agent-Based Approaches

The material of this chapter is essentially the same as that given in the paper by Hu et al. (2012a)[5] with some slight modifications.

It is generally very challenging for an oil refinery to make integrated decisions encompassing multiple functions based on a traditional Decision Support System (DSS), given the complexity and interactions of various decisions. To overcome this limitation, we propose an integrated DSS framework by combining both business and engineering systems with a dashboard. The dashboard serves as a human-computer interface and allows a decision maker to adjust decision variables and exchange information with the DSS. The proposed framework provides a two-stage decision making mechanism based on optimization and agent-based models. Under the proposed DSS, the decision maker decides on the values of a subset of decision variables. These values, or the first-stage decision, are forwarded through the dashboard to the DSS. For the given set of first-stage decision variables, a multi-objective robust optimization problem, based on an integrated business and engineering simulation model, is solved to obtain the values for a set of second-stage decision variables. The two-stage decision making process iterates until a convergence is achieved. A simple oil refinery case study with an example dashboard demonstrates the applicability of the integrated DSS.

---

[5] Hu, W., Almansoori, A., Kannan, P.K., and Azarm, S., 2012a, "Corporate Dashboards for Integrated Business and Engineering Decisions in Oil Refineries: an Agent-Based Approach," *Journal of Decision Support Systems*, 52(3), pp. 729-741.

5.1 Introduction

An oil refinery is a complex and continuous processing system with a series of highly nonlinear and strongly coupled subsystems (Pinto et al., 2000). Such a system presents considerable difficulties for enterprise management, operational optimization and process control, especially in uncertain environments (Khor et al., 2008). Managerial decisions for an oil refinery need to take into account capital investments, production, sales, material supply, product transportation, inventory, product developments and improvements, financial markets and market risks (Chryssolouris et al., 2005, Grossmann, 2005, Pongsakdi, et al., 2006). It is crucial that decision and information flow at different hierarchical levels of the company be considered as a whole to account for uncertainties in demand, raw material procurements, product quality and other market changes while achieving effective integration of business and engineering decisions (Koutsoukis et al., 1999).

Managing the inherent tradeoffs in decisions in business and engineering processes are most essential to an oil refinery's success and profitability. A typical oil refinery business process consists mainly of crude procurement, sales, inventory, transportation (delivery), and others (Lee et al., 1996). While business decisions are made at the upper level of the overall refinery operations, the lower-level engineering decisions are focused on transforming crude oil into various intermediate and end products in an energy-efficient manner (Kondili et al., 1993) while meeting the specifications demanded by the upper-level business processes. Several commercial decision support tools in the context of oil refinery operations are available, e.g., GRTMPS by Haverly Systems (2003), RPMS by Honeywell (2006) and PIMS by

Aspen Technology (2009). However, these commercial tools are predominantly focused either on business process and supply chain management, e.g., GRTMPS, or on engineering and process control, e.g., RPMS and PIMS, while taking little consideration of the interactions and integration of business and engineering processes. Consequently, a significant gap exists between the upper-level business and the lower-level engineering decision processes, while the problems of adaptability of an engineering department in response to market fluctuations have become increasingly prominent (Wang, 2005). In order to improve operational efficiency and enterprise profitability, it is necessary to achieve integration between the business and engineering decisions by making full use of the information flow between them.

In recent years, with an increasingly competitive global market, decisions in oil refinery business and engineering processes are frequently influenced by the market fluctuations and uncertainties. Matching demand and output of a refinery is a delicate balance, and a significant mismatch can be the difference between profit and loss. The management of this delicate balance has often led to comments such as “Oil production creates wealth, but oil refining has often destroyed it” (Mouaward, 2009). The commercially available decision support tools are typically developed with deterministic models and could suggest decisions which are sensitive to the uncertainty. Under such circumstances, it is desirable for business and engineering decisions to take into account the uncertain factors and obtain robust (or insensitive) decisions in midst of the fluctuating global markets and uncertain environments. The proposed DSS in this chapter is aimed at obtaining optimally robust decisions for

products and/or processes, that is, solutions that are optimum and relatively insensitive to uncertainty.

A variety of DSS methodologies and frameworks have been developed with real-world applications (Raghunathan, 1996, Power and Sharda, 2007). Kim et al. (2002) evaluated the enterprise information portal systems in the context of knowledge management activities. Their framework can be used to improve knowledge integration and information flow and facilitate efficient operations in large scale enterprises. The related literature also reports on an active intelligent DSS to support complex system decision making (Rao et al., 1994). Various information structures for team decision making are also considered for business decisions (Rao et al., 1995). While many of these developments are common to oil refinery systems, oil refinery problems are characterized by volatile input and market conditions that make it particularly challenging for DSS development.

Focusing on business decisions in an oil and petrochemical system, Chryssoloouris et al. (2005) presented a simulation-based approach to tackle short-term refinery scheduling problem. Their approach is able to handle discrete decision variables in a short decision-making time frame thus handling the uncertainties using a shorter planning horizon. Paolucci et al. (2002) considered the problem of allocating the crude oil loads of tanker ships to port and refinery tanks. Pitty et al. (2008) and Koo et al. (2008) used Matlab (Mathwork, 2010) to model the integrated refinery supply chain taking into consideration the activities of each component of the chain. They were able to model various business decisions and policies and to monitor the impact on the company's business performance. Clark (2005) showed the possibility for a

refinery company to monitor its supply chain in real-time or near real-time using advanced forecasting, planning and scheduling tools. Pinto et al. (2000) investigated optimization of a multi-product plant and proposed modeling of multi-product plant assuming that the fluctuations in market demand characteristics provide opportunity to define new operating points that increases the production of more valuable products. Gattu et al. (2003) identified integration of yield accounting with SAP (2010) for inventory management and order fulfillment and allocation. Their approach is based on an online (real time) optimization of the whole refinery. Jackson et al. (2003) used nonlinear optimization in the planning of multi-plant production site, where nonlinear models are used at the plant level to determine monthly production and inventory levels to meet demand forecast and maximize profit. Zhang and Zhu (2000) proposed a two-level decomposition approach for optimizing a large-scale refinery plant. The main advantage of their technique is the flexibility to adapt different optimizers for different subsystems.

While the majority of literature focus on refinery business decisions as presented above, optimization models for engineering decisions have also been studied (Al-Sharrah et al., 2001). For example, Gadalla et al. (2003) focused on optimization of an existing distillation process by changing key engineering variables. Micheletto et al. (2008) developed a mix-integer mathematical model for operational variables to minimize refinery utility costs. However, none of the previous work has considered both business and engineering decisions in a larger enterprise such as an oil refinery.

This chapter proposes to integrate business and engineering decisions with a dashboard based on an agent-based approach (Shoham et al., 2007) and a two-stage

decision-making process. Under the proposed decision support framework, dashboard serves as a human-computer interface which allows a decision maker to adjust decision variables and exchange information with the DSS. During the decision-making process, the first-stage decision variables are determined by the decision maker and forwarded through the dashboard to the DSS. For a given set of first-stage decision variables, the DSS simulates the business and engineering performances of the refinery as a function of the second-stage decision variables. Essentially, the second-stage decision-making process is posed as a multi-objective (both business and engineering objectives are considered) optimization problem which is solved to obtain a set of optimum solutions from which a preferred one is selected by the decision maker. Upon observing the selected solution in the oil refinery and its performance, the decision maker is able to refine and adjust the first-stage values of decision variables in order to achieve certain goals. Finally, the first-stage decision variables are updated through the dashboard and optimization of the second-stage decision variables is repeated. With the help of dashboard, the decision maker is able to interact with the DSS until a desired refinery performance is achieved. To demonstrate the proposed integration framework, a simple oil refinery case study is developed, in which the decision maker is modeled as an intelligent agent. The values of the first-stage decision variables are generated from a distribution profile function and updated using a no-regret learning algorithm (Hart and Mas-Colell, 2000) according to the profit. The oil refinery simulation model was developed to simulate the business and engineering performances using an agent-based simulation tool NetLogo (1999) and a commercial simulation software HYSYS (2009), respectively.

In the case study, a multi-objective robust optimization approach is applied to solve the integrated business and engineering optimization problem. As we show in this chapter, the integrated DSS framework considerably improves efficiency and effectiveness of decision support and information-processing capability for oil refinery decision making under uncertainty.

In the next section, a background on multi-objective robust optimization and the agent based approach is provided. In Section 5.3, a general framework for an integrated DSS is proposed. A case study which presents the specifics of the system constructed on the basis of the proposed framework is presented in Section 5.4. Section 5.5 concludes this chapter by providing the advantages of the integrated framework.

5.2 Problem Definition and Background

This section first presents the problem definition and then presents the background on agent-based approach for the proposed framework.

5.2.1 Problem Definition

Consider an enterprise such as an oil refinery company where the values for a set of business and engineering decisions need to be determined in order to achieve certain goals, e.g., to maximize profit and to maximize product quality. The decision variables are divided into two subsets, each of which contains both business and engineering decisions. The first set of decisions, as represented by $\mathbf{x}_I$ (a vector), consists of the values of decision variables which are set by the decision makers. A decision maker can be the manager or an expert in the company who makes critical

and strategic decisions and can set such values based on his/her expertise and experience. The second set of decisions, as represented by $\mathbf{x}_{II}$ (also a vector), includes decision variables considered for optimization. Categorization of decision variables to either $\mathbf{x}_I$ or $\mathbf{x}_{II}$ is based on the following rules: (1) the decision space (number of decision variables) for $\mathbf{x}_I$ is limited because it is difficult for a human decision maker to consider too many decisions; decisions on $\mathbf{x}_I$ typically consist of variables that decision maker has expertise/intuition and experience in setting; and (2) there is almost no limit on the number of decision variables in $\mathbf{x}_{II}$ unless restricted by the size of optimization problem and computation costs.

In this study, an oil refinery is characterized by a series of models which defines the functional relationships between the inputs and the outputs. The inputs to the oil refinery include a set of decision variables which are controlled by the decision maker and some parameters which are fixed at their nominal values. For example, a parameter such as the price of phthalic anhydride (an end product) is fixed at its nominal value of 1,200 $/ton. The outputs include intermediate and end product flows, utility costs, performance and characteristics of process units and so on. The outputs from the refinery are used to calculate Key Performance Indicators (KPIs) as well as the objective and constraint functions in the optimization problem. It is assumed that the lower and upper bounds of all decision variables are known a priori. Further, both decision variables and parameters can have interval uncertainty whose lower and upper limits are assumed to be known. Before introducing the proposed DSS framework, we present a brief introduction of two techniques used in the DSS system next.

5.2.2 Agent-Based Approach

An agent is an entity (software, model or individual) that performs a specific task without intervention of users or other agents (Julka et al., 2002a, 2002b). The most essential characteristic of an agent lies in its capability of making independent decisions. An agent can be responsive to and learn from the environment which is usually referred to as the adaptive behavior in an agent-based approach. In the proposed DSS framework, the agent-based approach is used to model the business process and the interaction between decision maker and dashboard, as elaborated next.

In simulating the refinery business model, the refinery, its inventory and the customers are each modeled as an agent. During a simulation cycle, each customer agent determines the quantity of product it is willing to purchase. This could be based on a realization from the distribution of the demand and sometimes updated based on a customer modifying its preferences based on other customers' preferences (learning from others). When a customer agent places its order, the refinery and inventory agents will respond and decide how to fulfill the order based on a predefined protocol. For instance, an end product is always delivered to a customer agent directly from the refinery whenever the production meets the demand; also the remaining products by the end of each simulation cycle are stored in the inventory. An inventory agent incurs a holding cost based on the amount of stock in inventory. There is also a "stock-out" penalty cost when stock in the inventory runs out or is not enough to fulfill an order. Since the rate of penalty cost is typically higher than the holding cost, the inventory agent should strike a balance between keeping too much

stock, which runs up inventory holding-costs, and too little stock, which brings a greater risk of running out of stock and incurring excessive penalty cost.

In general, the decision maker uses the dashboard to determine the values of some decision variables ($\mathbf{x}_{II}$) and then selects the values of the rest of the decision variables ($\mathbf{x}_{I}$) and vice versa. On the other hand, by taking the values of the decision variables $\mathbf{x}_{I}$ from the decision maker, the dashboard uses the optimization model to obtain a new set of decision alternatives $\mathbf{x}_{II}$ for the decision maker. This action-reaction process between the decision maker and the dashboard would eventually align with an improvement in the corporate profit such as profit. In the case study, the decision maker and a multi-objective optimization assisted dashboard are each modeled as an agent. The decision making agent is responsible for deciding on $\mathbf{x}_{I}$, while the dashboard agent generates a set of values for $\mathbf{x}_{II}$. These two agents interact by observing decisions made by each other and gradually learn to improve the decisions. In particular, a “no-regret learning” algorithm is used to model a “simulated decision maker” in the case study. The detail of the interaction between the decision maker and dashboard is discussed in Section 5.3.4 and the “no-regret learning” algorithm in Section 5.4.4.

5.3 An Integrated Decision Support System

The main components in the business and engineering domains are presented first. An integration framework is presented next which considers both business and engineering domains. Finally, a dashboard is developed as a decision support tool for the proposed DSS framework.

5.3.1 The Business Domain

An oil refinery's business domain is characterized by a network of retailers, distributors, transporters, storage facilities, and suppliers that participate in the sale, delivery, and production of a series of fuel and petrochemical products. The business domain in a typical oil refinery has the following components:

Procurement: request and track crude oil supply, and maintain crude supply records.

Demand planning: create an overall demand forecast for the oil refinery.

Capacity planning: evaluate the long-term and short-term capacity of the refinery to meet customers' demand.

Material requirements planning: determine crude quality requirements to support the production plan.

Inventory management: develop inventory policies and decisions based on the primary inventory cost.

Distribution planning: select the most cost-effective route and inventory movements based on customers' demand, transportation and inventory costs.

The oil refinery business domain serves to collect and process data concerning customers, orders, market fluctuations, distributors and services. An important function of oil refinery business is to determine market demands with respect to actual customer orders and estimate market trends by applying forecasting techniques. Based on the market information, the business decision variables such as how much crude oil to be purchased, what type of end products to produce and the quality requirements of these end products, are made to maximize profit for a given time period. The formulation of profit is given in Eq. (5.1):

$$profit = \sum_{pro \in S_1} q_{pro} c_{pro} - \sum_{cru \in S_2} q_{cru} c_{cru} - \sum_{res \in S_3} C_{res} \tag{5.1}$$

where c_{pro} is the unit price of products and c_{cru} is the unit cost of crude oil. q_{pro} and q_{cru} are the quantities of product sales and crude oil feed flow rate respectively. C_{res} represents other costs such as the capital cost, operating and utility costs, labor cost, inventory cost and so on. S_1 denotes different types of products and S_2 represents different types of crude oil. S_3 includes resources that the production requires, e.g., human resources, utility and so on. Notice that the quantity of feed flow rate F_k is a business decisions variable in Eq. (5.1). However, there are other business decision variables in a company that are not explicitly expressed in Eq. (5.1). For example, the quantity of an intermediate product used for production of an end-product is a short-term business decision not given in Eq. (5.1). Such a decision could affect the output capacity. On the other hand, end-product quantity and quality also depend on engineering decisions such as the operational settings. Therefore, the quantity on product sales q_{pro} is a result of both business decisions and engineering decisions. On the other hand, if difference between the internal and external markets is considered, product sales can be divided into internal market sales and external market sales as in Eq.(5.2):

$$\sum_{pro \in S_1} q_{pro} c_{pro} = \sum_{pro \in S_{1,I}} q_{pro,I} c_{pro,I} + \sum_{pro \in S_{1,E}} F_{pro,E} c_{pro,E} \tag{5.2}$$

where $q_{pro,I}$ and $q_{pro,E}$ represent quantity of products sold to the internal and external markets respectively. $c_{pro,I}$ and $c_{pro,E}$ represent the unit price of products in the internal and external markets respectively. In both Eq. (5.1) and Eq. (5.2), the unit price/cost of product and material such as c_{pro} (including $c_{pro,I}$ and $c_{pro,E}$), c_{cru} and

C_{res} are business parameters whose values can be assumed known. However, these parameters might have uncertainties. The uncertainties in business parameters are considered in the proposed DSS framework using the AA-MORO approach, while the detail will be discussed later in this paper. In the next section, the components of the engineering domain are presented.

5.3.2 The Engineering Domain

The oil refinery's engineering domain starts from the supply of crude oil. Crude oil is separated in the Crude Distillation Unit (CDU) into kerosene, naphtha, gas oil, petroleum gases, and others. These intermediate products are further processed and blended into fuel (e.g., gasoline, kerosene) and other petrochemical products. The objectives of an engineering department are to maximize the purity of the products that they produce while keeping the utility (energy, electricity, cooling water and so on) cost at a minimum level. Engineering is also responsible for collecting, accessing, and analyzing production data, and forwarding the critical information to the management. In order to meet business goals and comply with an oil company's overall plan, decision maker in an engineering department need to consider a few critical factors as in the following:

Operation setting: adjust operation parameters in the process equipment, for example feed flow rate, reflux ratio, boil-up ratio, and utility in CDU.

Production scheduling: develop a feasible production schedule for a product, given demand, production plan, capacity, and material availability.

Specifications: determine the quality specification of end products according to the law, standards, and regulations, taking into account customer requirements.

Quality management: monitor variations in the intermediate and end product quality through the enforcement of quality control criteria

Maintenance: optimize equipment operations to reduce utility cost, malfunctioning and equipment maintenance fee.

Typically, engineering decisions are focused on the operational variables in the process equipments and thus calculating product purity for a set of engineering decision variables involves solving a series of complicated nonlinear equations. In practical applications, this can be accomplished with the help of chemical process simulation software such as Aspen HYSYS (2009).

In a refinery, all facilities need to operate in an equilibrium state defined by the nominal values of the engineering process variables. However, uncertainty in the operating environment and variance in crude oil composition and properties tends to cause fluctuations in the engineering process, which could affect the quality and specification of the products. In the proposed DSS framework, AA-MORO is used to consider such uncertainty in engineering parameters. On the other hand, the engineering departments are at the lower echelons of decision making in an oil refinery and are rarely involved in the upper-level business/management decision-making process. If the engineering department can only make adjustment on the process variables based on the decisions made solely by the business process, the overall performance of the oil refinery can suffer as a result of the disconnect between business and engineering. This limitation is expected to be overcome by integration of business and engineering decisions, as presented in the next section.

5.3.3 Integration of Business and Engineering Decisions

The roadmap for integration of business and engineering decisions is shown in Fig. 5.1. It contains two primary flows: decisions flow mainly from the top decision maker to the business and engineering simulation models, and information flows in the opposite direction. Both decision and information flows pass through the dashboard which assists decision maker in implementing decisions and visualizing information for decision making. At the top of the roadmap, decision maker has certain goals to achieve during the decision-making process. These goals can include but not limited to maximizing profit, complying with market laws, regulations and etc. At the bottom of the roadmap, a robust optimization problem is formulated based on an integrated business and engineering simulation model.

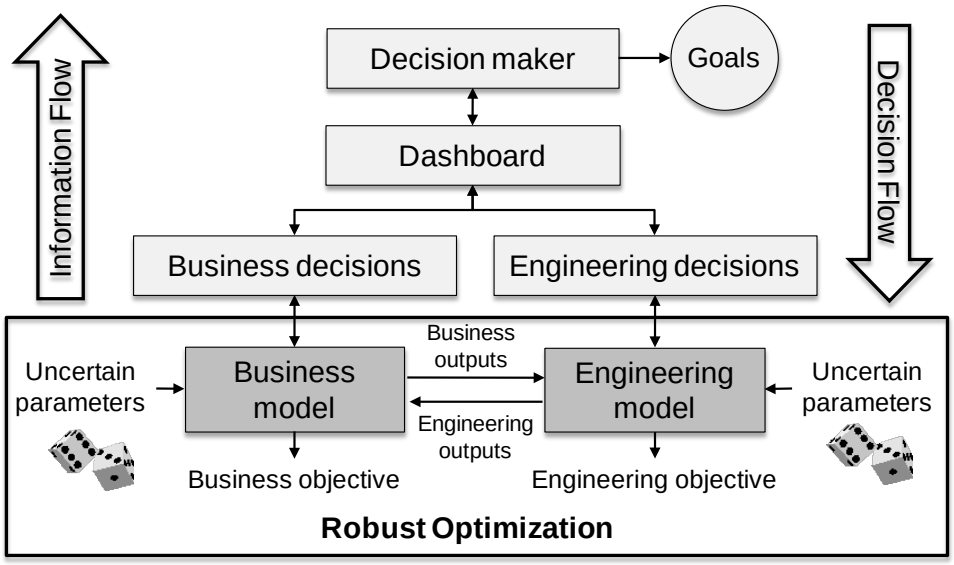


Fig. 5.1 Roadmap for integration of business and engineering decisions

The business and engineering models each is capable of predicting respective business and engineering performances for a given set of decision variables and parameters. In this integrated model, the business decision focus on the long-term

strategic decisions which is made at the higher hierarchy while the engineering decisions focus on the short-term operational decisions which is formulated at the lower hierarchy. Furthermore, the business and engineering model are coupled through the coupling variables. The coupling variables are represented by the business/engineering outputs between the two models. To achieve the optimal decision, it is desirable to connect business and engineering models, taking into account the coupling variables between them. An example of such couplings is characterized by the feed flow rate which is determined by a business department will be used as an input by the engineering department. The engineering department, on the other hand, returns the operating (utility) cost and product flow rate to the business department for calculating profit. The values of these coupling variables must be agreed upon by both business and engineering departments. Identifying the coupling variables to reach a mutual agreement without the support of a DSS system is a delicate task which typically involves many trials and errors. In the proposed DSS framework, however, a consistency constraint is enforced for each coupling variable such that if there are any discrepancies on a coupling variable, the differences will lead to a violation of the corresponding consistency constraint. Consequently, the optimizer tries to minimize the inconsistency as much as possible to retain model feasibility. When the optimal decision are obtained, the approach will guarantee the business and engineering analysis models to agree upon each other while achieving the optimal objectives.

Based on the integrated business and engineering model and considering both business and engineering objective and constraint functions, the optimization of

business and engineering decision variables is formulated as a multi-objective problem as follows:

$$
\begin{aligned}
&\min_{\mathbf{x}_{\mathrm{II}}} \; f_B(\mathbf{x}_{\mathrm{II}}, \mathbf{p}_B + \Delta\mathbf{p}_B) \\
&\min_{\mathbf{x}_{\mathrm{II}}} \; f_E(\mathbf{x}_{\mathrm{II}}, \mathbf{p}_E + \Delta\mathbf{p}_E) \\
&s.t.: \;\; \mathbf{g}_{B,k}(\mathbf{x}_{\mathrm{II}}, \mathbf{p}_B + \Delta\mathbf{p}_B) \le 0, \\
&\qquad \mathbf{g}_{E,k}(\mathbf{x}_{\mathrm{II}}, \mathbf{p}_E + \Delta\mathbf{p}_E) \le 0, \\
&\qquad \mathbf{x}_{\mathrm{II}}^l \le \mathbf{x}_{\mathrm{II}} \le \mathbf{x}_{\mathrm{II}}^u \\
&\qquad \forall \Delta\mathbf{p}_B \in [\Delta\mathbf{p}_B^l, \Delta p_B^u]; \forall \Delta\mathbf{p}_E \in [\Delta\mathbf{p}_E^l, \Delta\mathbf{p}_E^u]
\end{aligned}
\tag{5.3}
$$

where f_B and $g_{B,j}$ represent business objective and business constraints, respectively, such as inventory capacity, limitation on the type and volume of products that can be produced, and so on. f_E and $g_{E,j}$ represent engineering objective and engineering constraints, respectively, considering equipment processing capacity, maximum allowable vessel pressure and temperature for safety and other restrictions. $\mathbf{x}_{\mathrm{II}}$ is a vector consisting of business and engineering decision variables in optimization, as defined earlier in Section 5.2.1. $\mathbf{p}_B$ and $\mathbf{p}_E$ represent uncertain business and engineering parameters while $\Delta\mathbf{p}_B$ and $\Delta\mathbf{p}_E$ represent the variation in those parameters.

Note that in Eq. (5.3), evaluation of each objective and constraint function such as f_B, f_E, $\mathbf{g}_B$, $\mathbf{g}_E$, requires a simulation run of either the business or engineering model. Therefore, it could result in a large number of function calls and present computational difficulties. However, in the proposed DSS framework, the AA-MORO approach is employed to efficiently solve Eq. (5.3). On the other hand, in practice, the optimization problem in Eq. (5.3) may include many decision variables from both business and engineering domain, which poses another challenge to the decision maker of an oil refinery to observe and make decisions. To improve the

efficiency of decision-making process and quality of the decisions, an interactive user interface, or dashboard is constructed in the proposed DSS framework. The main function and role of dashboard is presented in the next section.

5.3.4 Dashboard: Management Decision Support System

Dashboard is a human-computer interface. In the proposed framework, dashboard connects the decision maker and the integrated business and engineering simulation and optimization model to facilitate presentation of information to top-level decision maker of the refinery.

The layout of a conceptual dashboard for oil refinery performance management is shown in Fig. 5.2. One important capability of dashboard is to visualize KPIs. For example, profit from sales of an end product is an indicator of how efficient the company is in turning investment into net income and which products are driving profits. Similarly, stock-out cost and production informs the decision maker how well the oil refinery's production capacity can meet market demands. With the current and historical KPIs presented on dashboard, the decision maker is able to observe oil refinery's performance first hand. In addition, the dashboard allows decision makers across various departments in an oil refinery to coordinate and implement decisions. When there is a significant deviation of KPIs from their normal value, the decision maker can take actions by changing the decision variables through the sliders on dashboard. Because an oil refinery may involve many decision variables, it is typically difficult for a decision maker to control all decision variables manually on the dashboard. Therefore, the proposed DSS framework also integrates multi-objective optimization to obtain optimum decision variables as shown in Fig. 5.2.

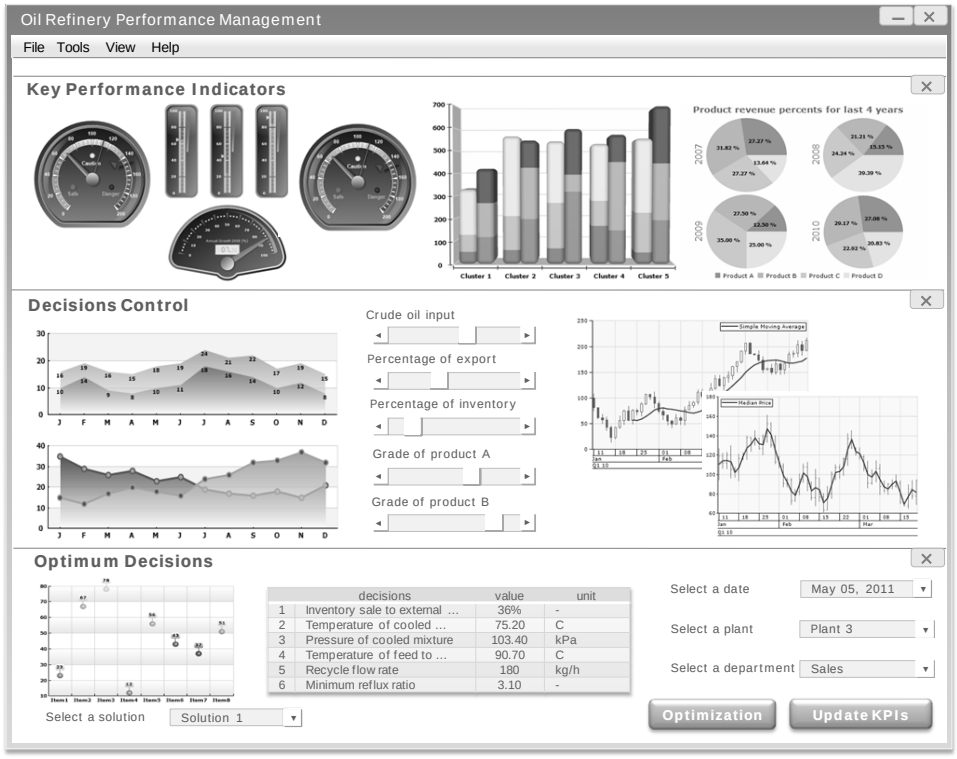


Fig. 5.2 Layout of a conceptual dashboard for oil refinery performance management

Fig. 5.3 shows decision support role of a dashboard where $\mathbf{x}_{\mathrm{I}}$ and $\mathbf{x}_{\mathrm{II}}$ represent decision variables controlled by the decision maker and optimizer, respectively. Particularly, the values of $\mathbf{x}_{\mathrm{I}}$ are determined according to decision maker's expertise and previous experience while the value of $\mathbf{x}_{\mathrm{II}}$ is selected based on the optimum solutions obtained from AA-MORO. When making decisions, decision maker needs to evaluate the information shown on the dashboard. The decision maker has certain goals to achieve such as maintain all the KPIs in the oil refinery at their normal level and ensure that the refinery is profitable. It should be noted, however, that any decision from the decision maker must comply with market regulations and other constraints. On the other hand, the objective functions in AA-MORO need to be consistent with the goal of decision maker. For example, maximizing profit from

sales and maximizing the purity of an end-product are the two primary goals in the case study presented in Section 5.4.

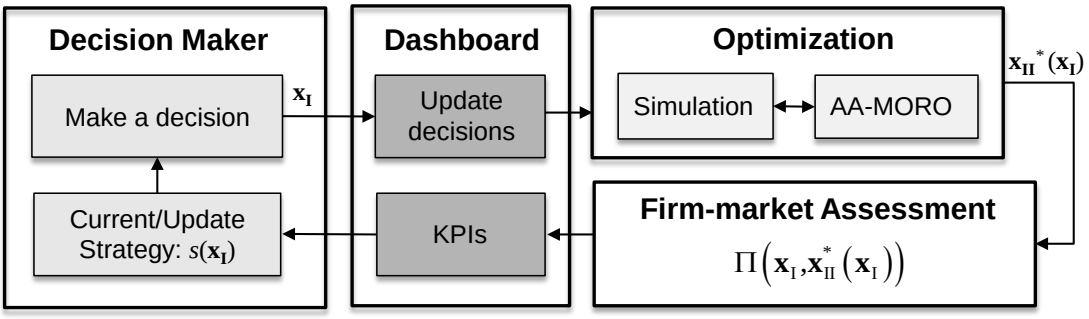


Fig. 5.3 The decision support role of dashboard

With the help of dashboard, the decision-making process starts with an initial set of decision variables ($\mathbf{x}_I$) by the decision maker. These decision variables ($\mathbf{x}_I$) are then reflected (update previous decision variables) on the dashboard and then passed on to the integrated business and engineering simulation in the 'Optimization' block. Next, AA-MORO searches for the robust optimum solutions using the simulation model for the given set of decision variables ($\mathbf{x}_I$) made by the decision variable maker. The optimum decision variables, represented by $\mathbf{x}_{II}^*(\mathbf{x}_I)$, are forwarded to the oil refinery where $\prod(\mathbf{x}_I, \mathbf{x}_{II}^*(\mathbf{x}_I))$ represents firm-market assessment (for example, profit) function. Based on the optimum decision variables $\mathbf{x}_{II}^*(\mathbf{x}_I)$ and current values of decision variables $\mathbf{x}_I$, the KPIs of the oil refinery are observed and sent to the dashboard. By observing the KPIs on dashboard, decision maker needs to update the decision strategy function $s(\mathbf{x}_I)$. According to the updated strategy function, decision maker makes a new decision on $\mathbf{x}_I$ and updates the decision variables on the dashboard. The integrated simulation-based AA-MORO is then repeated. After the new set of optimum decision variables are obtained and implemented in the refinery, it may impact the current KPIs according to the firm-market assessment. As a result,

these KPIs may be changed and are shown on the dashboard again, through which the decision maker continue to update decision variables until plant performance reaches an equilibrium state with desired values of KPIs.

In the next section, we use a case study to demonstrate how the above mentioned approach works for an example dashboard.

5.4 An Oil Refinery Case Study

In this case study, the focus is on (1) identifying the KPIs that help better represent the interactions among the market forces, the management policies and business and engineering decisions, (2) designing the measurement schemes across various business and engineering departments that will encompass the areas of marketing metrics, financial measures, and key engineering performance measures, and (3) designing simulation studies to test these measures under different product market (ranging from products being substitutes to products being complements among the firms) scenarios, and the sensitivity of these measures to policy changes and actions. Based on the KPIs, the measurement schemes and simulations, a dashboard that integrates data of the market, the company, and those KPI's is devised.

The schematic of the supply, production, and marketing activities involved in the case study is shown in Fig. 5.4. In the figure, "Murban" refers to a particular type of crude oil. The internal market consists of local customers, while the external market is composed of other customers not considered in the internal market, e.g., foreign customers. The schematic starts with the crude oil (Murban) extraction in the oil field where the oil refinery purchases Murban from the oil extraction company. The majority of the purchased Murban is transported to the oil refinery plant for

producing fuel and petrochemical products, which are sold in both internal and external market. In addition to supply internal and external market for petroleum product demand, the oil refinery in the case study can also sell some portion of crude oil directly in the external crude oil market, as shown in Fig. 5.4.

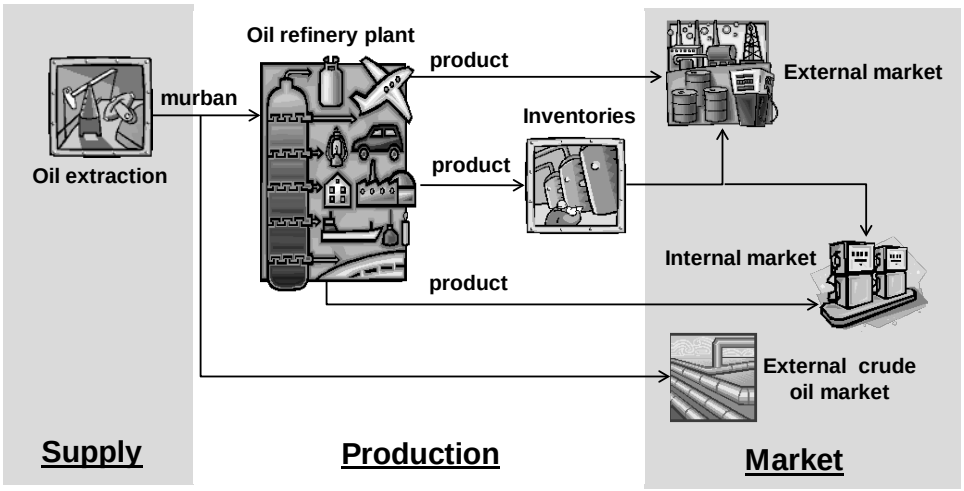


Fig. 5.4 Schematic of case study model

Inside the oil refinery plant in Fig. 5.4, Murban is first processed in the Crude Distillation Unit (CDU), where the output of CDU includes among others some naphtha. Naphtha is then used to produce o-xylene and processed in a reactor-distillation unit. One output product from the reactor-distillation unit is phthalic anhydride, an industrial chemical for production of plasticizers for plastics. In our case study, it is assumed that phthalic anhydride is the end product sold in the market.

In the case study, we make various other assumptions. One assumption is that phthalic anhydrides can be sold directly to the internal and external customers if the production from the refinery is equal or greater than the quantity of demands. Note in reality, the products from refinery are first stored in a short-term inventory whenever they are produced. The products that are stored in the short-term inventory are then

delivered to customers, depending on their demands. In the case study, however, the short-term inventory is not considered and the transportation and short-term inventory costs are ignored. After satisfying customer demands, the remaining (excessive) products are forwarded and stored in the designated long-term inventory to meet future customer demands. Long-term inventory costs are considered in the case study. In case the production from the refinery is less than demand, the product previously stored in the long-term inventory is used to fulfill the demand. However, if a combination of the refinery production and inventory still fails to meet the demand, a stock-out penalty cost has to be assessed.

The decision variables in the case study are summarized in Table 5.1. Among these decision variables, the amount of daily crude oil purchase (variable $\mathbf{x}_{\mathrm{I}}$) is determined by the decision maker and a few selected engineering and business decision variables ($x_{\mathrm{II},1}$, $x_{\mathrm{II},2}$,… $x_{\mathrm{II},5}$) are determined by AA-MORO (optimization).

Table 5.1 Decision variables in the oil refinery case study

Description	Variable	Unit	Lower bound	Upper bound
Daily crude oil purchase	$\mathbf{x}_{\mathrm{I}}$	bbl/day	9.0×10^{4}	10.0×10^{4}
Percentage of crude oil sold to external market	$x_{\mathrm{II},1}$	N/A	0%	100%
Percentage of storage sold to external market	$x_{\mathrm{II},2}$	N/A	0%	100%
Mass flow rate of feed air	$x_{\mathrm{II},3}$	kg/s	24	27
Pressure of cooled mixture	$x_{\mathrm{II},4}$	kPa	100	104
Phthalic column reflux ratio	$x_{\mathrm{II},5}$	N/A	0.3	2.0

In general, both $\mathbf{x}_{\mathrm{I}}$ and $\mathbf{x}_{\mathrm{II}}$ are defined as vectors and include engineering as well as business decision variables. In the case study $\mathbf{x}_{\mathrm{I}}$ is a scalar and contains only one

business decision variable for simplicity but $\mathbf{x}_{II}$ includes two business decision variables ($x_{II,1}$ and $x_{II,2}$) and three engineering decision variables ($x_{II,3}$, $x_{II,4}$, $x_{II,5}$) as shown in Table 5.1.

In the case study, it is also assumed that both engineering and business parameters can have interval uncertainties. For example, the temperature of feed stream to phthalic distillation column is an uncertain engineering parameter and the selling price of phthalic anhydride in external market is an uncertain business parameter. The nominal values of uncertain parameters and their lower and upper limits of uncertainties are shown in Table 5.2.

Table 5.2 Uncertain parameters in the oil refinery case study

Description	Nominal	Unit	Lower limit	Upper limit
Price of phthalic anhydride in external market	1,200	$/ton	-5%	5%
Temperature of feed stream to phthalic distillation column	75.27	°C	-5%	5%

The objective of AA-MORO is to maximize profit that the refinery generates through the sale of phthalic anhydride and maximize the purity of phthalic anhydride. The optimization problem needs to satisfy certain constraints such as capacity of inventory, limitation of pressure and temperature in the oil refining process and so on. It is subject to the lower and upper bounds on the decision variables as specified in Table 5.2. In order to obtain robust optimal solutions, an acceptable variation range for each objective is defined. In the case study, the acceptable range for each objective is assumed to be ±5% of the nominal objective function value. By solving the bi-objective optimization problem, AA-MORO obtains a set of Pareto optimum

solutions from which the decision maker can choose. The selected values for optimum decision variables ($\mathbf{x}_I$) are assumed to be used or implemented in the oil refinery and accordingly the decision maker obtains the KPIs from the dashboard (as presented later) that will help him/her to adjust decision variables on $\mathbf{x}_I$.

5.4.1 Engineering and Business Simulation

The engineering model focuses on the reactor-distillation process for producing phthalic anhydride from naphtha. The process is simulated in Aspen HYSYS (2009), as shown in Fig. 5.5.

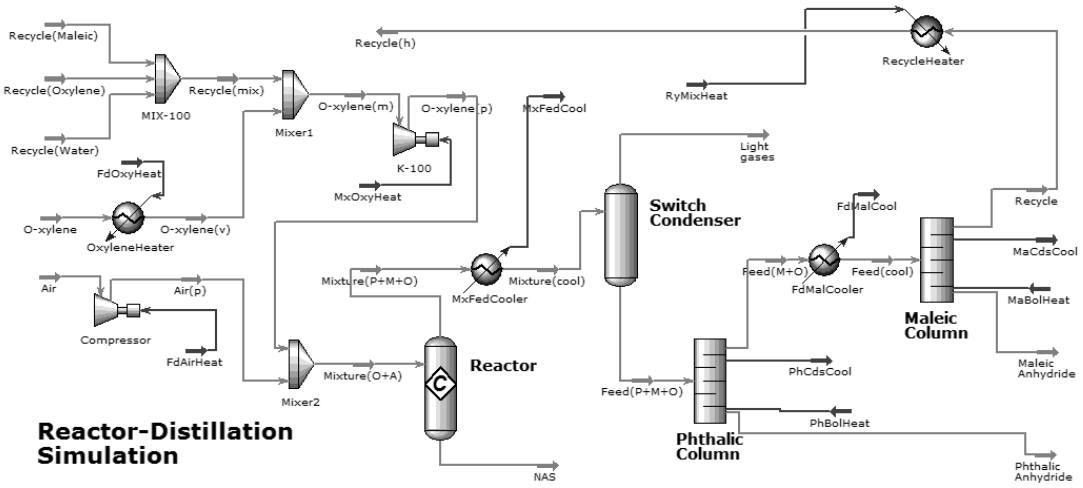


Fig. 5.5 Process flow diagram of engineering simulation

In the simulated reactor-distillation process, the raw materials are air and o-xylene. The vaporized o-xylene and hot air are first combined and then fed to the reactor. In the reactor, o-xylene is oxidized to form phthalic anhydride but some maleic anhydride may also be formed. The reactor effluent enters the switch condenser to remove light gases and water. From the switch condenser, the remaining anhydride and unreacted o-xylene are fed to a series of two distillation columns to separate and

obtain phthalic anhydride. The phthalic distillation column separates phthalic anhydride from the feed stream and the maleic distillation column separate maleic anhydride from the remaining components. The top stream from the maleic column contains mostly unreacted o-xylene with a small amount of maleic anhydride and water. In the simulation, the unreacted o-xylene is recycled and combined with the o-xylene as feed material.

The business model is simulated using the agent-based software NetLogo (1999). The business model characterizes the crude oil and end-product markets by simulating oil refinery supply and the customer demand. The customers, including the internal and external customers are modeled using the customer agents which can be distributors or downstream chemical companies. The internal and external markets for phthalic anhydride each have five customer agents. The demand by each customer agent is assumed to be normally distributed. The mean and standard deviation of demand for internal and external customer agents are summarized in Table 5.3. Notice that the probabilities for customers in both internal and external markets making purchases are presumed to follow truncated normal distribution as per empirical observations. The mean and standard deviation of the probability are also shown in Table 5.3. Particularly, the external crude oil market is characterized by one customer agent with a 100% probability of purchasing. That is, no matter how much the company decides to sell to the crude oil external market, all quantity will be purchased by the crude oil market customer agent.

Table 5.3 Parameter of customer agents in the oil refinery case study

Description	Number of agents	Demand (kg/day)		Probability	
		mean	std.	mean	std.
Internal market	5	30,000	500	100%	10%
External market	5	25,000	1000	100%	10%
External crude oil market	1	-	-	100%	-

Fig. 5.6 shows the simulation window of the business model in NetLogo. The business parameter and their descriptions in the Netlogo simulation are defined in Table 5.4, where the nominal values of business parameters are fixed. The two circles shown in the center of the simulation window represent the refinery agent and inventory agent. The black and white agents in the simulation window represent the internal customer agents and external customer agents, respectively. According to the interactions between the markets and the oil refinery, the profit from end-product sales can be obtained as an output from the business model.

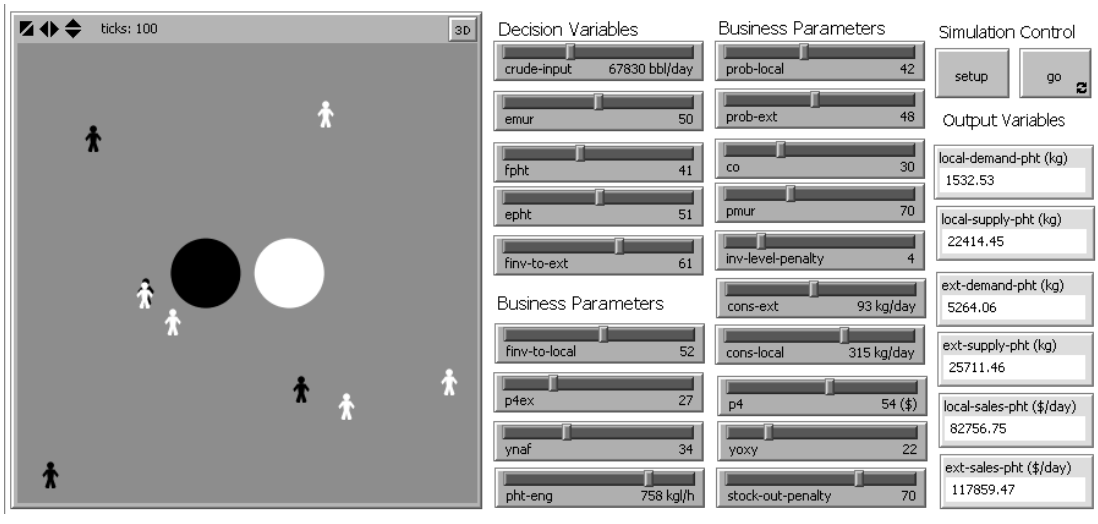


Fig. 5.6 Agent-based business simulation window in Netlogo

Table 5.4 Descriptions of business parameters in Netlogo

Description	Parameter	Nominal Value	Unit
Price of crude oil in external market	*pmur*	70	$/bbl
Percentage of inventory storage released to internal market	*finv-to-local*	52%	N/A
Price of phthalic anhydride in internal market	*p4*	900	$/ton
Inventory storage expense of phthalic anhydride	*inv-level-penalty*	4	$/ton/day
Inventory stock-out penalty of phthalic anhydride	*stock-out-penalty*	400	$/ton
Yield of naphtha from crude distillation	*ynaf*	34%	N/A
Yield of oxylene from naphtha	*yoxy*	22%	N/A

The input to the engineering simulation includes three decision variables (Table 5.1) such as mass flow rate of feed air, pressure of cooled mixture, phthalic column reflux ratio, one uncertain parameter such as temperature of feed stream to phthalic distillation column (Table 5.2) and the feed flow rate of o-xylene (obtained as a output from the business simulation). The output to the engineering simulation includes the purity and flow rate of phthalic anhydride. The input to the business simulation includes three decision variables (Table 5.1) such as daily crude oil purchase, percentage of crude oil sold to external market, percentage of inventory storage sold to external market, one uncertain parameter: price of phthalic anhydride in external market (Table 5.2) and the flow rate of phthalic anhydride (obtained as a output from the engineering simulation). The output of the business simulation includes profit and the feed flow rate of o-xylene. In the case study, the engineering and business simulations are connected through an interface program developed in

Matlab, which is used to run both simulations programmatically and exchange information between Netlogo and Aspen HYSYS.

5.4.2 Formulation of Multi-Objective Robust Optimization

Based on the engineering and business simulations, a multi-objective optimization problem is formulated to maximize profit (business objective) and maximize purity of phthalic anhydride (engineering objective), as defined in the following:

$$\begin{aligned}
&\underset{x_{2,i}}{\text{Maximize:}} && \text{Profit (objective 1)} \\
&\underset{x_{2,i}}{\text{Maximize:}} && \text{Purity of phthalic anhydride (objective 2)} \\
&\text{Subject to:} && \text{Business (e.g. inventory) constraints} \\
& && \text{Engineering (e.g. pressure) constraints} \\
& && \text{Uncertainty in parameters (e.g. price, temperature)} \\
& && \mathbf{x}_{\text{II}}^{l} \le \mathbf{x}_{\text{II}} \le \mathbf{x}_{\text{II}}^{u}
\end{aligned} \tag{5.4}$$

Using the AA-MORO approach, $x_{\text{II},1}$, $x_{\text{II},2}$,… $x_{\text{II},5}$ (including both business and engineering decision variables in Table 5.1) is robustly optimized.

5.4.3 Dashboard

Dashboard in the case study is developed using the Graphical User Interface capability in Matlab. It includes three main functional panels: “Key Performance Indicator”, “Decision Control” and “Optimum Decision”, as shown in Fig. 5.7. The daily crude oil purchase ($\mathbf{x}_{\text{I}}$) is controlled by the decision maker and a set of five engineering and business decision variables ($x_{\text{II},1}$, $x_{\text{II},2}$,… $x_{\text{II},5}$) are considered for optimization by AA-MORO. Profit is used as a key performance indicator in the case study, as shown in “Key Performance Indicator” panel. The decision maker can observe current KPI as well as the data in the previous iterations. Based on the

observation, decision maker then updates his/her decisions in the “Decision Control” panel when necessary. The history (simulation steps) of decisions made by the decision maker are recorded and shown as well. On the right-hand side, the “Optimum Decision” panel presents the multi-objective (Pareto) optimum solutions to the decision maker. The decision maker can select a preferred solution from the Pareto solutions. The table in the lower half of the “Optimum Decision” panel indicates the optimum values of selected solution.

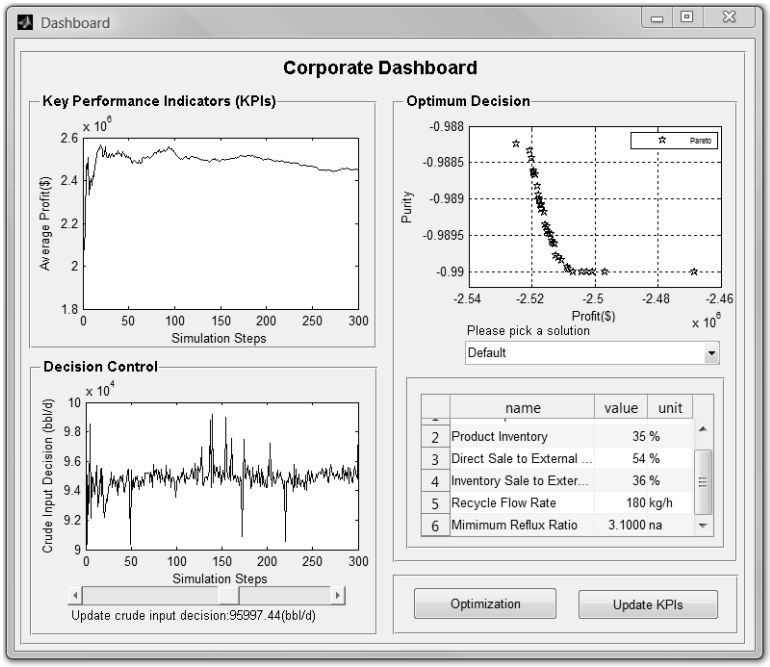


Fig. 5.7 A Matlab GUI based dashboard in the case study

Decision making based on dashboard is an iterative process. In the following paragraph, we briefly explain how dashboard facilitates such a process:

Initially, the daily crude oil purchase ($\mathbf{x}_I$) is determined according to previous settings in the oil refinery. Decision maker adjusts the slider bar in the “Decision Control” panel for an initial value of $\mathbf{x}_I$. Dashboard forwards the value of $\mathbf{x}_I$ to the integrated business and engineering simulation model. When decision maker engages the

"optimization" button on dashboard, AA-MORO start running multi-objective robust optimization based on the integrated engineering and business simulations for a fixed value of $\mathbf{x}_{\mathrm{I}}$ (as determined by the decision maker earlier). After optimization completes, AA-MORO obtains a set of multi-objective optimum solutions. These solutions are presented in the "Optimum Decision" panel, as shown in Fig. 5.7. From these optimum solutions, the decision maker is required to select one solution per his/her preference. The optimum values of the decision variables ($x_{\mathrm{II},1}$, $x_{\mathrm{II},2}$,… $x_{\mathrm{II},5}$) for the selected solution are shown in the table in the lower half of the 'Optimum Decision' panel. These optimum values of $\mathbf{x}_{\mathrm{II}}$, along with $\mathbf{x}_{\mathrm{I}}$ are then implemented in the oil refinery. According to engineering operation and the firm-market interaction, the actual values of refinery's performance are obtained. Afterwards, decision maker engages the "Update KPIs" button on dashboard to refresh the current value of KPI in the "Key Performance Indicators" panel.

By observing the KPI, the decision maker can adjust the previous values of decision variables on dashboard. The adjustment on $\mathbf{x}_{\mathrm{I}}$ is forwarded to the integrated simulations by dashboard. Once again, decision maker engages the "Optimization" button on dashboard to run AA-MORO and select an optimum solution for implementation in the oil refinery. Finally, the current KPIs may be changed and reflected on the dashboard after decision maker engages "Update KPIs". Consequently, the decision maker updates $\mathbf{x}_{\mathrm{I}}$ after observing new data of KPI, and the procedure is repeated until a desired reading of KPI is achieved.

5.4.4 No-regret Learning

In the case study, it is assumed that the decision maker has a decision strategy function, as represented by $s(\mathbf{x}_I)$. A decision maker's strategy function is essentially a probability distribution profile (or probability density function) of the decision variable $\mathbf{x}_I$. The decision-making process is comparable to drawing a sample from the strategy function. A change of the strategy function reflects a change of belief of the decision maker about a decision made previously. In simulating the decision-making process by the decision maker, a no-regret learning algorithm (Hart and Mas-Colell, 2000) is used in the case study. We assume that the decision maker exhibits learning behavior, i.e., updating his/her decision strategy by iteratively making decisions and observing payoffs (Miller, 2007). We define "action" as the decision, i.e., $\mathbf{x}_I^c$, made in the *c'th* iteration. Additionally, we define "strategy", i.e., $s(\mathbf{x}_I)$, as a probability density function representing the likelihood that the decision maker chooses an action $\mathbf{x}_I$. By letting the decision maker exhibit learning behavior, we account for the fact that the decision maker may deviate from making optimal decision by anticipating the future (Montgomery et al., 2005).

The no-regret learning algorithm is adapted to simulate the process that the decision maker uses to gradually develop his/her decision strategy by interacting with the dashboard. The no-regret learning algorithm was previously applied to represent a dynamic procedure of action-reactions among multiple players (Wang et al., 2011). In the case study, we consider the decision maker and the optimizer (presenting the optimization results in the dashboard) who make decisions collectively to affect the firm's profit. The decision maker's decision is made by learning from the past

whereas the optimizer (AA-MORO) searches and obtains decisions to optimize its objectives.

Let $\prod(\mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{II}})$ denote the profit function for the firm. The decision maker's payoff function is set to be identical to profit. In every iteration, the decision maker first computes a regret function *R* defined as (*c* is the iteration counter):

$$R^c(\mathbf{x}_{\mathrm{I}}) = \frac{1}{nc}\sum_{c=1}^{nc}\left(\Pi(\mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{II}}^c) - \Pi(\mathbf{x}_{\mathrm{I}}^c, \mathbf{x}_{\mathrm{II}}^c)\right) \tag{5.5}$$

The regret function reflects the average increase in profit if an action has been always played in previous iterations. The strategy function, i.e., the probability of playing $\mathbf{x}_{\mathrm{I}}$ in the following iteration, is proportional to the regret:

$$s^c(\mathbf{x}_{\mathrm{I}}) = \frac{\left[R^c(\mathbf{x}_{\mathrm{I}})\right]^+}{\sum_{\mathbf{x}_{\mathrm{I}} \in X_{\mathrm{I}}}\left[R^c(\mathbf{x}_{\mathrm{I}})\right]^+} \tag{5.7}$$

in which:

$$\left[R^c(\mathbf{x}_{\mathrm{I}})\right]^+ = \begin{cases} 0, & \text{if } R^c(\mathbf{x}_{\mathrm{I}}) \le 0 \\ R^c(\mathbf{x}_{\mathrm{I}}), & \text{if } R^c(\mathbf{x}_{\mathrm{I}}) > 0 \end{cases} \tag{5.8}$$

where a positive superscript implies that a non-negative value of regret function is used. In case the calculated regret function value is negative, a zero value is used instead. The above equations assume that the decision variable $\mathbf{x}_{\mathrm{I}}$ is discrete. In case $\mathbf{x}_{\mathrm{I}}$ is continuous, the summation in Eq. (5.7) is replaced with an integral. When the decision maker makes a decision, i.e., playing an action, it essentially draws a sample from the updated strategy function in Eq. (5.7).

5.4.5 Dashboard Demo

Two case study scenarios are considered. In both scenarios, the range for crude oil input ($\mathbf{x}_{\mathrm{I}}$) is pre-specified based on unit capacity as $9.0\times10^4 \le \mathbf{x}_{\mathrm{I}} \le 1.0\times10^5$ (bbl/day).

In the first scenario, the decision maker controls the dashboard manually based on his/her experience. In the second scenario, the decision-making process and the control of dashboard is automated by using a no-regret learning algorithm, which is essentially used to simulate a decision maker.

Scenario 1: $\mathbf{x}_I$ *is determined by decision maker*

In scenario 1, the value of daily crude oil input ($\mathbf{x}_I$) is determined by decision maker. For demonstration, it is assumed that decision maker randomly selects four discrete values for $\mathbf{x}_I$: 9.2×10^4, 9.4×10^4, 9.6×10^4 and 9.8×10^4 bbl/day. The decisions on daily crude oil that are selected by the decision maker are used to run the simulation. In each iteration, the decision maker first adjusts the decision control bar on dashboard. The value of daily crude oil input ($\mathbf{x}_I$) is sent by dashboard as fixed value to the integrated engineering and business simulation. Next, decision maker engages the "optimization" level on dashboard to initiate AA-MORO to obtain the multi-objective robust optimal solutions. Finally, decision maker is required to select one desirable solution from a set of optimum solutions based on its objective values. In addition, the optimum values for the decision variables corresponding to the optimum solution are implemented in the oil refinery by the dashboard.

Fig. 5.8 shows the optimum solutions for each iteration in the objective function space. In Fig. 5.8, it can be seen that the two objective functions are conflicting and therefore as profit increases (its negative value decreases), the purity of phthalic anhydride decreases (its negative value increases). Comparing different iterations with different values of daily crude oil input, the range on product purity is similar.

However the largest profit is achieve in iteration 3 when the amount of daily crude oil input is 9.6×10^4 bbl/day.

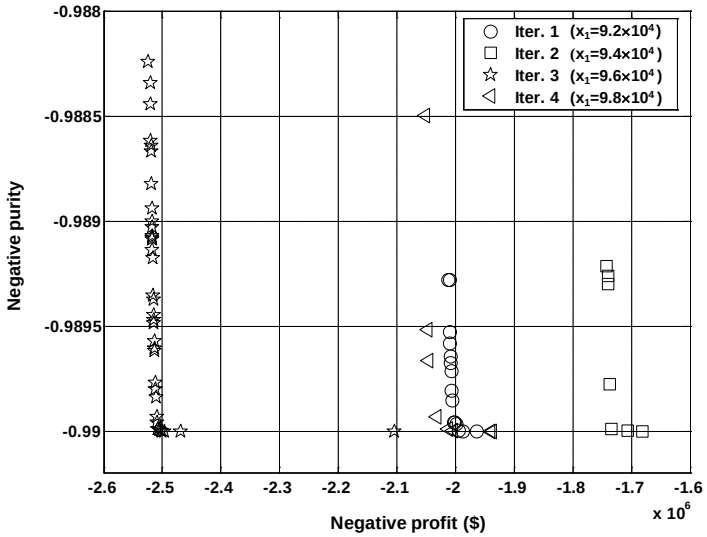


Fig. 5.8 Optimum design solution for case study scenario 1

Scenario 2: $\mathbf{x}_I$ *is determined by no-regret learning*

In scenario 2, instead of the decision maker manually determining the value of daily crude oil input (decision on $\mathbf{x}_I$), a no-regret learning algorithm is used to simulate (mimic) the decision making process. Based on the no-regret learning, a total of 300 iterations are simulated. In each iteration, a sample is first drawn from the strategy function $s(\mathbf{x}_I)$ which is characterized by a distribution profile (PDF). The procedure of drawing a sample is comparable to the decision maker making a new decision. Dashboard is informed of the new decision and forwards it to the integrated simulation model. Similar to Scenario 1, AA-MORO obtains robust optimum solutions and again the decision maker selects a solution with maximum profit. Notice that a selection strategy can also be derived using a utility function (e.g., a weighed summation) of the two objective functions. For demonstration, the solution

with maximum profit is selected which essentially assumes the utility (weight) of product purity is zero. When the selected solution is implemented, the outcome profit is used to update the strategy function from which another sample (decision) is drawn and the iteration continues.

The simulated distribution profile (PDFs) for $s(\mathbf{x}_I)$ in iteration 10, 50, 150 and 300 are shown in Fig. 5.9. It can be seen that initially (i.e., in iteration 10), the distribution profile is diffusive because the decision maker has no information about the past. By iteratively making decisions through interacting with the dashboard and observing the outcome (profit), the decision maker gradually shift his/her decision to a more profitable position, as represented by the shift of distribution of the PDFs. Furthermore, the shrinkage of the distribution profile reflects that the decision maker strengthens her/her belief on the more profitable decisions.

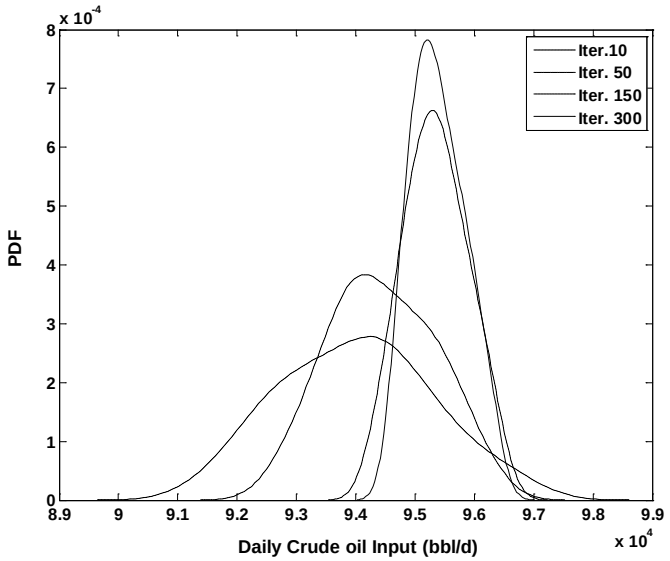


Fig. 5.9 Simulated distribution profile (PDFs) of daily crude oil input

In Scenario 2, it is noticed that the shape of the distribution profile and the observed profit remain unchanged (converged) after 300 iterations. As the no-regret learning

algorithm is driven by profit, the value of profit is maximized when the simulation is converged. A closer look at Fig. 5.9 reveals that the maximum profit in Scenario 2 corresponds to the peak of the PDF where the amount of daily crude oil input is between 9.5×10^4 and 9.6×10^4 (bbl/day). This observation appears consistent with the result from Scenario 1 where profit is maximum at $\mathbf{x}_I = 9.6\times10^4$ (bbl/day) among four discrete choices.

It should be noted that a variety of other learning algorithms are also applicable to modeling the decision-making process. However, the simulation in Scenario 2 is primarily aimed at demonstrating how a human decision maker and an automated decision agent (i.e., AA-MORO) can interact and collectively improve a firm's performances (e.g., profit). We leave a comparison and appropriateness of learning algorithms to a future study.

5. Summary

The traditional oil refinery DSS is built around a single decision maker and based on either business or engineering decisions, but not on both. Under the traditional DSS, business and engineering decision makers make decisions only considering their own domains, and there is no connection/discussion between them. A significant limitation of the traditional DSS scheme is that both business and engineering decisions could drive to optimize their own local functional objective. The decisions made in this way could be conflicting or suboptimal. In this study, we show that when the business and engineering decisions are combined, an integrated DSS can be more effective in supporting the management to make critical decisions. This integrated DSS is based on an "all-at-once" multi-objective robust optimization scheme where business and

engineering analysis models are integrated and considered as a whole. Using the proposed integration framework, an oil refinery is able to obtain a global optimal solution and better decisions than otherwise.

Complexities and problems of non-linearity that are inherent to any refinery supply chain optimization, and to which multi-objective optimizers have been proposed, can be efficiently combined as a backbone of the dashboard with the help of simulation software like NetLogo. The power of NetLogo resides in the ability to represent every single customer, their interactions, and the element of the supply chain and its interactions, and model these appropriately. In the context of stochastic demand data, by varying decision variables, we were able to monitor the company's performance and ultimately tune those decisions to meet our objective of maximum profit. Also visible through the simulation were the changes to the elements that would constitute the dashboard, and the set of key performance indicators critical to the health of the company. We use a simple case study to show the decision support role of dashboard and how it can be used to coordinate across various departments in decision making. It is observed that a maximum simulated profit can be achieved in the two case study scenarios. This study and the integration framework are in the context of oil and petrochemical industries. The proposed framework is applicable to many firms with similar business and engineering sectors and it can also accommodate other market variables such as interest rate and exchange rate fluctuations.

In the next Chapter, the conclusion of this dissertation is presented. The chapter highlights the contributions and limitation of three main research thrusts of this dissertation, followed by the recommendations for future work.

Chapter 6: Conclusion

In this dissertation, a new Approximation Assisted MultiObjective Robust Optimization (AA-MORO) approach and a new Approximation Assisted Multiobjective collaborative Robust Optimization (AA-McRO) approach under interval uncertainty are developed. Some of the highlights of these approaches are as follows:

- Both AA-MORO and AA-McRO use online approximation that can be used to replace a computationally expensive objective and/or constraint function with an inexpensive metamodel, and significantly reduce the computational cost.
- AA-MORO is developed based on an improved MORO framework. This improved MORO is sequential and significantly reduces the number of robustness evaluations compared to a previous MORO approach.
- AA-McRO is based on a newly formulated bi-level collaborative optimization framework where a single-objective optimization problem in the upper level is used to coordinate the lower-level optimization problems and enhances the convergence (numerical evidence) of system solutions.
- A two-stage robust decision support system is developed with AA-MORO to integrate business and engineering decisions in the context of an oil refinery. The proposed decision framework allows an oil refinery manager to make informed decisions more efficiently by way of a dashboard.

Section 6.1 provides concluding remarks, followed by the contributions in Section 6.2. The limitations of the proposed approaches are summarized in Section 6.3. Finally, the recommendations for the future work are provided in Section 6.4.

6.1 Concluding Remarks

In Chapter 3, a new MORO approach under interval uncertainty is developed. The new MORO approach performs robustness evaluation of solutions with respect to their objective and constraint functions based on a worst-case analysis. The robustness evaluation is only performed for the optimum solutions and this information is used to iteratively refine the feasible domain to locate robust optimum solutions. Compared to the bi-level approach, the new MORO approach significantly reduces the number of robustness evaluation needed to obtain robust optimum solutions. To further improve the computational cost, the new MORO approach is combined with an online approximation. The resulting approach is called Approximation-Assisted MORO or AA-MORO. Several numerical examples and an oil refinery engineering example are solved and their results are compared for AA-MORO and other MORO approaches. The comparison results indicate that typically AA-MORO requires considerably fewer number of function calls than the previous approaches.

In Chapter 4, an Approximation Assisted Multiobjective collaborative Robust Optimization (AA-McRO) approach under interval uncertainty is presented. AA-McRO uses a single-objective optimization problem to coordinate all system and subsystem optimization problems in a CO framework. AA-McRO is based on a new framework in which the consistency constraints in a CO are converted into penalty terms. These penalty terms are then integrated into the subsystem objective functions. By using this new framework, AA-McRO is able to explore the design space and obtain optimum solutions more efficiently compared to a previously reported McRO.

AA-McRO also uses online approximation for objective and constraint functions to perform system robustness evaluation and subsystem-level optimization. The optimum solutions from AA-McRO are verified online and considered acceptable only when the absolute error falls under a user-specified threshold. A numerical example and an engineering example are used to demonstrate the applicability of AA-McRO. The results from AA-McRO and several related approaches including the previous and the new MCO and McRO are compared. It is found that the new MCO, new McRO and new AA-McRO converge faster (numerical evidence) than the previous MCO, McRO and AA-McRO approaches. This is attributed to the single-objective problem that is dedicated to coordinate the shared and coupling variables. Both the AA-McRO and new AA-McRO require significantly fewer number of function calls and thus more efficient than the other CO approaches that do not use approximation.

In Chapter 5, a two-stage decision support system (DSS) is developed in the context of an oil refinery to integrate engineering and business decisions under uncertainty using the AA-MORO approach. This integrated DSS is based on a scheme where business and engineering analysis models are integrated and considered together. The two-stage DSS allows the user to focus on the most critical decision variables in an oil refinery, while AA-MORO is used to optimize the rest of the decision variables. To demonstrate how the integrated DSS framework can be used in a real-world application, a simple oil refinery case study is developed with dashboard. The case study employs two scenarios to show that the two-stage DSS iteratively guide the user in the decision making process in achieving the desired refinery performance.

6.2 Contributions

The main contributions of this dissertation are as described next.

1. Development of a new MORO approach

- A new sequential MORO approach is developed which does not require robustness evaluation for each candidate design point. Instead, the sequential MORO performs robustness evaluation only for the optimum solution points, and is more efficiency compared to the previously reported bi-level MORO approach.
- The new MORO approach obtains robust optimum solutions by iteratively restricting the feasibility domain of the optimization problem. The approach accumulates the worst values of $\Delta\mathbf{p}$ during robustness evaluation and subsequently uses that information to form constraint cuts to refine the optimization problem and obtain robust optimum solutions.

2. Development of a New MCO and a New McRO approach

- A new bi-level collaborative optimization framework is developed for MCO. The new MCO approach uses a single-objective optimization problem at the upper level to coordinate the optimization problems in each subsystem at the lower level, which allows MCO to achieve convergence (numerical evidence) much faster compared to a previous approach.
- A new McRO approach is developed with the same new bi-level collaborative optimization framework as in the new MCO. To consider interval uncertainty in the system and subsystem optimization problems, the new McRO integrates robustness evaluation in its lower level.

- In the new MCO and McRO framework, the consistency constraints are converted into penalty terms which are integrated into the objective function in both system and subsystem optimization. Using the penalty method in the system and subsystem optimization problems allows the optimizer to explore the design space better than using the consistency constraints in these problems.
- The optimal solution selection strategy in the new MCO and new McRO are also improved. Instead of selecting a random solution, as is done a previously reported approach, the new selection strategy devised such that it achieves better consistency in the coupling variables among the subsystems.

3. Integration of Online Approximation with MORO and McRO

- A new online approximation approach is proposed in which the metamodel is updated using the optimum solution points. In this way, the predictive capabilities of the metamodel is progressively improved in the area where the optimum is expected to be, as more and more sample points are evaluated and added to the sample set.
- A new online verification method is developed and implemented in the proposed AAO to quantify the accuracy of the estimated optimum solution. Using the online verification, the accuracy of the objective and constraint functions at the optimum solution points has to reach a user specified threshold before the solutions are declared.
- Both AA-MORO and AA-McRO combine online approximation with the improved MORO and new McRO to reduce computational cost.

4. An Integrated Decision Support System (DSS)

- A new two-stage decision support paradigm is developed in which the decision maker and AA-MORO each determines a subset of the decision variables in an oil refinery. The values of decision variables determined by the decision maker depend on the values of decision variables optimized using an AA-MORO approach, and vice versa. These two stages are iterated. In this way, the decision maker is able to progressively refine the value of decision variables until a desired oil refinery performance is achieved.

6.3 Limitations

The limitations of the proposed approaches are summarized in the following:

- The new MORO approach obtains robust design solutions by iteratively restricting the feasible domain of a deterministic optimization problem. The success of obtaining robust solutions from the new MORO depends on accumulation of the worst values of $\Delta\mathbf{p}$ during the robustness evaluation, which are used to form constraint cuts. In some numerical examples, the new MORO approach may not obtain as many robust solutions as the previous MORO approach given a limited number of iterations.
- Online approximation in the proposed approaches, i.e., AA-MORO and AA-McRO, provides good estimated values for the predicted functions according to the results in the numerical and engineering examples. However, the accuracy of the estimate is dependent upon the approximation (metamodeling) techniques and the parameters used.

- The new AA-McRO requires additional coupling variables compared to the previous McRO approaches. This is because the original system optimization is moved to the lower level. In the case in which a problem has many coupling variables between the subsystems and system problem, this move could create a large number of target variables in the coordination problem which can increase the problem size in AA-McRO.
- The integrated DSS focuses on a small scale model of an oil refinery. The size of the problem can grow significantly considering the interaction and relationship between business and engineering decisions for real-world and larger scale oil refinery firms. This may create computational difficulty with the optimization and agent based DSS framework, as presented in this dissertation.

The above limitations could be addressed as part of the future work, as discussed next.

6.4 Recommendations for Future Work

Some recommendations for future work are presented as follows.

1. Quantification of reducible uncertainty in AA-MORO and AA-McRO

A key assumption in this dissertation is that the uncertainty is defined by a known and fixed interval where the lower and upper bounds are known a priori. This assumption can be relaxed by considering interval uncertainty to be reducible. The concept of reducible interval uncertainty has been reported in the previous literature. Combining the concept of reducible uncertainty with the proposed AA-MORO and AA-McRO approach can be a natural extension to this dissertation. Several research questions need to be explored with reducible uncertainty in AA-MORO and AA-McRO. These are: (i) How to quantify the trade-off between cost of reducing the uncertainty and the

adjustments on the lower and upper bounds of uncertainty. (ii) How to combine reducible uncertainty in the existing AA-MORO and AA-McRO frameworks to provide design flexibility while not significantly increasing computational cost. (iii) How to develop an approximation approach to consider the adjustment of the uncertain interval and improve the accuracy of metamodeling in the robustness evaluation stage.

2. Improvement to AA-McRO

The current AA-McRO uses a bi-level formulation, where the evaluation of each system candidate design requires a call to the subsystem optimization problems, and a large number of system design points are considered. This bi-level formulation can require significant amount of computational effort, particularly when the size of the problem grows. To address this limitation, a sequential formulation for collaborative optimization (e.g., Roth and Brian 2008, Tosserams et al. 2009) may be considered for the AA-McRO approach. Using a sequential approach, the coordination problem and subsystem optimization problems are iterated to arrive at the system optimum solution. This can be more efficient than the bi-level approach. However, reformulating AA-McRO with a sequential approach still presents considerable technical challenge because robustness evaluation is not considered in the existing sequential collaborative optimization approaches. As such, the investigation and research on the development of a sequential AA-McRO approach should be explored further. On the other hand, the dissertation has not provided any proof of existence of solutions with the AA-McRO approach in its current formulation. Further research is required to show a formal convergence proof for the AA-McRO approach.

3. Improvement to online approximation

One of the most critical strategies in online approximation is the use of optimum solution points in (DOE) sampling for the purpose of focusing on the expected optimum regions. However, it may be important to perform sampling with respect to the constraint functions that are used to form the optimum solutions as well. Furthermore, research is needed on refining the metamodel accuracy for robustness evaluation. This is because for robustness evaluation the metamodel should focus on a smaller uncertain interval around each design point. On the other hand, developing a good metamodel is typically problem dependent. Specially, other types of metamodeling techniques and fine-tuning of the parameters used in construction of the metamodel need to be further investigation.

4. Extending the current DSS for large-scale and/or multi-plant oil refinery

The context within which the DSS framework in Chapter 5 is developed is based on a small-scale and single-plant oil refinery. However, a large oil refinery may include a number of plants with close interactions among those plants. This requires the development of a DSS framework which is able to support and consider a multi-plant oil refinery. In achieving this objective, one important challenge lies in the computational cost when the size of the optimization problem increases, considering the coupling variables between different plants. With multi-plant oil refinery, the AA-McRO approach can be applied to solve the optimization problem in a decomposed manner. Finally, the DSS framework can also be extended further to consider competition among a number of oil refineries. All these research recommendations need to be considered in a future work.

References

1. Ahn J., and Kwon J.H., 2006, “An Efficient Strategy for Reliability-Based Multidisciplinary Design Optimization Using BLISS,” Structural and Multidisciplinary Optimization, 31(5) pp. 363-372.

2. Allison J.T., Kokkolaras M., and Papalambros P.Y., 2009, “Optimal Partitioning and Coordination Decisions in Decomposition-Based Design Optimization,” *Journal of Mechanical Design*, 131(8), pp. 081008-1 to 081008-8.

3. Al-Sharrah, G. K., Alatiqi, I., Elkamel, A, and Alper, E., 2001, “Planning an Integrated Petrochemical Industry with an Environmental Objective,” *Industrial & Engineering Chemistry Research*, *40*(9), pp. 2103-2111.

4. Aspen HYSYS and Aspen PIMS, 2009, http://www.aspentech.com/core, Aspen Technology (Last time accessed: May 30, 2012).

5. Aute V., and Azarm S., 2006, “A Genetic Algorithms Based Approach for Multidisciplinary Multiobjective Collaborative Optimization,” *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA.

6. Azarm S., and Li W.C., 1988, “A Two Level Decomposition Method for Design Optimization,” *Engineering Optimization*, 13 pp. 211-224.

7. Balling R.J., and Sobieszczanski-Sobieski J., 1996, “Optimization of Coupled Systems: A Critical Overview of Approaches,” *AIAA Journal*, 34(1), pp. 6-17.

8. Beyer, H.G., and Sendhoff, B., 2007, "Robust Optimization - A Comprehensive Survey," *Computer Methods in Applied Mechanics and Engineering*, 196(33-34), pp. 3190-3218.

9. Braun R., and Kroo I., 1996, "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization: State-of-the-Art*, SIAM, pp. 98-116.

10. Buche, D., Schraudolph, N. N., and Koumoutsakos, P., 2005, "Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, 35(2) pp. 183–194.

11. Chen, W., Wiecek, M. M., and Zhang, J., 1999, "Quality Utility---a Compromise Programming Approach to Robust Design," *Journal of Mechanical Design*, 121(2), pp. 179-187.

12. Chryssolouris, G., Papakostas, N., and Mourtzis, D., 2005, "Refinery Short-Term Scheduling with Tank Farm, Inventory and Distillation Management: An Integrated Simulation-Based Approach," *European Journal of Operational Research*, 166(3), pp. 812-827.

13. Clark, S., 2005, "Supply Chain Logistics Challenges," *Refining*, pp. 1-4.

14. Coello, C. A., Veldhuizen D. A., and Lamont, G. B., 2002, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second edition, Springer, New York.

15. Cramer E., Dennis J., Frank P., Lewis R., and Shubin G., 1994, "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, 4, pp. 754–776.

16. Deb, K., 2001, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, New York.

17. Deb, K., and Gupta, H., 2006, "Introducing Robustness in Multi-Objective Optimization," *Evolutionary Computation*, 14(4), pp. 463-494.

18. Demiguel A,, and Murray W., 2006, "A Local Convergence Analysis of Bilevel Decomposition Algorithms," *Optimization and Engineering*, 7(2), pp. 99–133.

19. Du, X., and Chen, W., 2000, "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design," *Journal of Mechanical Design*, 122(4), pp. 385-394.

20. Du, X., and Chen, W., 2004, "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," *Journal of Mechanical Design*, 126(2), pp. 225-233.

21. Elishakoff I., Haftka R.T., Fang J., 1994, "Structural Design under Bounded Uncertainty—Optimization with Anti-Optimization," *Computers & Structures*, 53(6), pp.1401–1405.

22. Ferreira, J., Fonseca, C. M., and Covas, J. A., 2008, "Evolutionary Multi-Objective Robust Optimization," *Advances in Evolutionary Algorithm*, pp. 261-278.

23. Fonseca, C. M., and Fleming, P. J., 1993, "Genetic Algorithms for Multiobjective Optimization : Formulation, Discussion and Generalization,"

Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423.

24. Gadalla, M., Jobson, M., and Smith, R., 2003, "Optimization of existing heat-integrated refinery distillation systems," *Chemical Engineering Research and Design*, 81(1), pp. 147-152.

25. Gaspar-Cunha, A., and Covas, J., 2008, "Robustness in Multi-Objective Optimization Using Evolutionary Algorithms," *Computational Optimization and Applications*, 39(1), pp. 75-96.

26. Gattu, G., Palavajjhala, S., and Robertson, D.B., 2003, "Are Oil Refineries Ready For Non-Linear Control and Optimization?" *International Symposium on Process Systems Engineering and Control*, Mumbai, India.

27. Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA.

28. Grossmann, I.E., 2005, "Enterprise-wide optimization: A new frontier in process systems engineering," *AIChE Journal*, 51(7), pp. 1846-1857.

29. Gu X., Renaud J.E., and Penninger C.L., 2006, "Implicit Uncertainty Propagation for Robust Collaborative Optimization," *Journal of Mechanical Design*, 128(4), pp. 1001-1013.

30. Gunawan S., and Azarm S., 2003, "Quality-Assisted Multi-Objective Multidisciplinary Genetic Algorithms," *AIAA Journal*, 41(9), pp.1752-1762.

31. Gunawan, S., and Azarm, S., 2005, "Multi-Objective Robust Optimization Using a Sensitivity Region Concept," *Structural and Multidisciplinary Optimization*, 29(1), pp. 50-60.

32. Hart, S., and Mas-Colell, A., 2000, “A Simple Adaptive Procedure Leading to Correlated Equilibrium,” *Econometrica*, 68(5) pp. 1127-1150.

33. Haveerly GRTMPS, 2003, http://www.haverly.com/grtmps.htm, Haverly Systems (Last time accessed: May 30, 2012).

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

35. Honeywell Refinery and Petrochemical Modeling System (RPMS), 2006, http://hpsweb.honeywell.com/RPMS/default.htm, Honeywell (Last time accessed: August 1, 2010).

36. Hong, Y. S., Lee, H., and Tahk, M. J., 2003, “Acceleration of the Convergence Speed of Evolutionary Algorithms Using Multilayer Neural Networks,” *Engineering Optimization*, 35(1), pp. 91–102.

37. Hu, W., Li, M., Azarm, S., Al Hashimi, S., Almansoori, A., and Al-Qasas, N., 2009, “Improving Multi-Objective Robust Optimization under Interval Uncertainty Using Worst Possible Point Constraint Cuts,” *Proceedings of the ASME International Design Engineering Technical Conferences*, San Diego, CA.

38. Hu, W., Azarm, S., and Almansoori, A., 2010, “Approximation Assisted Multi-objective collaborative Robust Optimization (AA-McRO) under Interval Uncertainty,” *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Fort Worth, TX.

39. Hu, W., Li, M., Azarm, S., and Almansoori, A., 2011, “Improving Multi-Objective Robust Optimization Under Interval Uncertainty using Online

Approximation and Constraint Cuts," *Journal of Mechanical Design*, 133(6), pp. 061002-1 to 061002-9.

40. Hu, W., Almansoori, A., Kannan, P.K., and Azarm, S., 2012a, "Corporate Dashboards for Integrated Business and Engineering Decisions in Oil Refineries: an Agent-Based Approach," *Journal of Decision Support Systems*, 52(3), pp. 729-741.

41. Hu, W., Azarm, S., and Almansoori, A., 2012b, "New Approximation Assisted Multi-Objective Collaborative Robust Optimization (New AA-McRO) Under Interval Uncertainty," *Structure and Multidisciplinary Optimization* (To appear).

42. Hu, W., Azarm, S., Almansoori, A., and Elkamel, A., 2012c, "*Robust Multi-Objective Genetic Algorithm under Interval Uncertainty, in Multi-Objective Optimization: Developments and Prospects for Chemical Engineering*," Wiley, New York (Accepted).

43. Hu, W., Saleh, K.H., and Azarm, S., 2012d, "Approximation Assisted Multiobjective Optimization with Combined Global and Local Metamodeling," *Proceedings of the ASME International Design Engineering Technical Conferences*, Chicago, IL (To appear).

44. Jackson, J.R., and Grossmann, I.E., 2003, "Temporal Decomposition Scheme for Nonlinear Multisite Production Planning and Distribution Models," *Industrial & Engineering Chemistry Research*, 42(13), pp. 3045-3055.

45. Jang, B.S., Yang, Y.S., Jung H.S., and Yeun, Y.S., 2005, "Managing Approximation Models in Collaborative Optimization," *Structural and Multidisciplinary Optimization*, 30(1), pp. 11-26.

46. Jin, R., Du, X., and Chen, W., 2003, “The Use of Metamodeling Techniques for Optimization under Uncertainty,” *Structural and Multidisciplinary Optimization*, 25(2), pp. 99-116.

47. Julka, N., Srinivasan, R., and Karimi, I., 2002a, “Agent-based Supply Chain Management--1: Framework,” Computers & Chemical Engineering, 26(12), pp. 1755-1769.

48. Julka, N., Karimi, I., and Srinivasan, R., 2002b, “Agent-based Supply Chain Management--2: a Refinery Application,” *Computers & Chemical Engineering*, 26(12) (2002) 1771-1781.

49. Karakasis, M.K., Giotis, A. and Giannakoglou, K. C., 2001, “Efficient Genetic Optimization using Inexact Information and Sensitivity Analysis. Application in Shape Optimization Problems,” *ECCOMAS Computational Dynamics Conference*, Swansea, Wales, UK.

50. Khor, C.S., Elkamel, A., Ponnambalam, K., and Douglas, P.L., 2008, “Two-stage Stochastic Programming with Fixed Recourse Via Scenario Planning with Economic and Operational Risk Management for Petroleum Refinery Planning under Uncertainty,” *Chemical Engineering and Processing: Process Intensification*, 47(9-10), pp. 1744-1764.

51. Kim, H.M., Michelena, N.F., Papalambros, P.Y., Jiang, T., 2003, “Target Cascading in Optimal System Design,” *Journal of Mechanical Design*, 125(3), pp. 474-480.

52. Kim, Y.J., Chaudhury, A., and Rao, H.R., 2002, "A Knowledge Management Perspective to Evaluation of Enterprise Information Portals," *Knowledge and Process Management*, 9(2), pp. 57-71.

53. Knowles, J., and Corne, D., 1999, "The Pareto Archived Evolution Strategy: a New Baseline Algorithm for Pareto Multiobjective Optimisation," *IEEE Proceedings of the 1999 Congress on Evolutionary Computation*, pp. 98-105.

54. Koehler, J.R., and Owen, A.B., 1996, *Computer Experiments, Handbook of Statistics*, Elsevier Science, New York, pp. 261-308.

55. Kokkolaras, M., Mourelatos, Z.P., Papalambros, P.Y., 2006, "Design Optimization of Hierarchically Decomposed Multilevel Systems Under Uncertainty," *Journal of Mechanical Design*, 128(2), pp. 503-508.

56. Kondili, E., Shah, N., and Pantelides, C.C., 1993, "Production Planning for The Rational Use of Energy in Multiproduct Continuous Plants," *Computers & Chemical Engineering*, 17(1), pp. 123-128.

57. Koo, L.Y., Adhitya, A., Srinivasan, R., and Karimi, I.A., 2008, "Decision Support for Integrated Refinery Supply Chains: Part 2. Design and Operation," *Computers & Chemical Engineering*, 32(11), pp. 2787-2800.

58. Koutsoukis, N.-S., Mitra, G., and Lucas, C., 1999, "Adapting On-Line Analytical Processing for Decision Modeling: The Interaction of Information and Decision Technologies," *Decision Support Systems*, 26(1), pp. 1-30.

59. Kroo, I., and Manning, V., 2000, "Collaborative Optimization: Status and Directions," *Proceedings of the 8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA.

60. Kurpati, A., Azarm, S.. and Wu, J., 2002, “Constraint Handling Improvements for Multi-Objective Genetic Algorithms,” *Structural and Multidisciplinary Optimization*, 23(3), pp. 204-213.

61. Lee, H., Pinto, J.M., Grossmann, I.E., and Park, S., 1996, “Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management, Industrial & Engineering Chemistry Research,” 35(5), pp. 1630-1641.

62. Lee, K. H., and Park, G. J., 2006, “A Global Robust Optimization Using Kriging Based Approximation Model,” *Mechanical Systems, Machine Elements and Manufacturing*, 49(3), pp. 779-788.

63. Li, M., Azarm, S., and Boyars, A., 2006, “A New Deterministic Approach using Sensitivity Region Measures for Multi-Objective and Feasibility Robust Design Optimization,” *Journal of Mechanical Design*, 128(4), pp. 874-883.

64. Li, M., Azarm, S., 2008, “Multiobjective Collaborative Robust Optimization with Interval Uncertainty and Interdisciplinary Uncertainty Propagation,” *Journal of Mechanical Design*, 130(8), pp. 081402-1 to 081402-11.

65. Li, G., Aute, V., and Azarm, S., 2010a, “An Accumulative Error Based Adaptive Design of Experiments for Offline Metamodeling,” *Structural and Multidisciplinary Optimization*, 40, pp. 137-155.

66. Li, M., Gabriel, S. A., Shim, Y., and Azarm, S., 2010b, “Interval Uncertainty-Based Robust Optimization for Convex and Non-Convex Quadratic Programs with Applications in Network Infrastructure Planning,” *Networks & Spatial Economics* 11, pp. 159-191.

67. Li, M., Hamel, J., and Azarm, S., 2010c, “Optimal Uncertainty Reduction for Multi-Disciplinary Multi-Output Systems Using Sensitivity Analysis,” *Structural and Multidisciplinary Optimization*, 40, pp. 77-96.

68. Lian, Y., and Liou, M., 2004, “Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithm,” *10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Albany, NY, 1, pp. 1316-1325.

69. Liang, J., Mourelatos, Z. P., and Nikolaidis, E., 2007, “A Single-Loop Approach for System Reliability-Based Design Optimization,” *Journal of Mechanical Design*, 129(12), pp. 1215-1224.

70. Limbourg, P., 2005, “Multi-Objective Optimization of Problems with Epistemic Uncertainty,” *Evolutionary Multi-Criterion Optimization.* (Springer, Berlin), pp. 413–427.

71. Lombardi, M., and Haftka, R. T., 1998, “Anti-Optimization Technique for Structural Design under Load Uncertainties,” *Computer Methods in Applied Mechanics and Engineering*, 157(1-2), pp. 19-31.

72. MATLAB, Version 2010a, 2010, *MATLAB and Simulink for Technical Computing*, Mathworks.

73. McAllister, C.D., Simpson, T.W., Hacker, K., Lewis, K., and Messac, A., 2005, “Integrating Linear Physical Programming Within Collaborative Optimization for Multiobjective Multidisciplinary Design Optimization,” *Structural and Multidisciplinary Optimization*, 29(3), pp. 178-189.

74. McKay, M.D., Beckman, R.J., and Conover, W.J., 1979, "A Comparison of Three Methods for Selecting Values of Input Variables from a Computer Code," *Technometrics*, 21, pp. 239-245.

75. Messac, A., and Ismail-Yahaya, A., 2002, "Multiobjective Robust Design Using Physical Programming," *Structural and Multidisciplinary Optimization*, 23(5), pp. 357-371.

76. Micheletto, S.R., Carvalho, M.C.A., and Pinto, J.M., 2008, "Operational Optimization of the Utility System of An Oil Refinery," *Computers & Chemical Engineering*, 32(1-2), pp. 170-185.

77. Miettinen, K. M., 1999, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, MA.

78. Miller, J. H., 2007, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, Princeton University, Princeton, NJ.

79. Mouaward, J., 2009, "Chilly Climate for Oil Refiners," *The New York Times*.

80. Montgomery, D. B., Moore M. C., and Urbany, J. E., 2005, "Reasoning about Competitive Reactions: Evidence from Executives," *Marketing Science*, 24(1), pp.138-149.

81. Myers, R. H., and Montgomery, D. C., 1995, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley & Sons, New York.

82. Nakayama, H., Arakawa, M., and Washino, K., 2003, "Using Support Vector Machines in Optimization for Black-box Objective Functions," *Proceedings of the International Joint Conference on Neural Networks*, 2, pp. 1617-1622.

83. Netlogo, U. Wilensky, NetLogo, http://ccl.northwestern.edu/netlogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999).

84. Paolucci, M., Sacile, R., and Boccalatte, A., 2002, "Allocating Crude Oil Supply to Port And Refinery Tanks: A Simulation-Based Decision Support System," *Decision Support Systems*, 33(1), pp. 39-54.

85. Park, G.J., Lee, T.H., Lee, K. H., and Hwang, K.H., 2006, "Robust Design: An Overview," *AIAA Journal*, 44(1), pp. 181-191.

86. Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R.T. and Kim, N.H., 2010, "Adaptive Designs of Experiments for Accurate Approximation of a Target Region," *Journal of Mechanical Design*, 132(7), pp. 071008-071009.

87. Pinto, J.M., Joly, M., and Moro, L.F.L., 2000, "Planning and Scheduling Models For Refinery Operations," *Computers & Chemical Engineering*, 24(9-10), pp. 2259-2276.

88. Pitty, S.S., Li, W., Adhitya, A., Srinivasan, R., and Karimi, I.A., 2008, "Decision Support for Integrated Refinery Supply Chains: Part 1. Dynamic Simulation," *Computers & Chemical Engineering*, 32(11), pp. 2767-2786.

89. Pongsakdi, P., Rangsunvigit, K., Siemanond, M.J., and Bagajewicz, 2006, "Financial Risk Management in The Planning Of Refinery Operations," *International Journal of Production Economics*, 103(1), pp. 64-86.

90. Power, D.J., and Sharda, R., 2007, "Model-driven Decision Support Systems: Concepts and Research Directions, *Decision Support Systems*, 43(3), pp. 1044-1061.

91. Raghunathan, S., 1996, “A Structured Modeling Based Methodology to Design Decision Support Systems, *Decision Support Systems*, 17(4), pp. 299-312.

92. Rangaian, G.P., 2009, *Multi-Objective Optimization Applications in Chemical Engineering*, *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, World Scientific. pp. 27-54.

93. Rao, H.R., Sridhar, R., and Narain, S., 1994, “An Active Intelligent Decision Support System - Architecture And Simulation,” *Decision Support Systems*, 12(1), pp. 79-91.

94. Ray, T., 2002, “Constrained Robust Optimal Design Using a Multiobjective Evolutionary Algorithm,” *Proceedings of the 2002 Congress on Evolutionary Computation*, pp. 419-424.

95. Ray, T., Isaacs, A., and Smith, W., 2009, *Surrogate Assisted Evolutionary Algorithm for Multi-Objective Optimization. Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, World Scientific. pp. 131-149.

96. Renaud, J.E., Gabriele, G.A., 1993, “Improved Coordination in Nonhierarchical System Optimization,” *AIAA Journal*, 31(12), pp. 2367-2373.

97. Roth, B.D., Kroo, I.M., 2008, “Enhanced Collaborative Optimization: A Decomposition-Based Method for Multidisciplinary Design,” *Proceedings of the ASME Design Engineering Technical Conferences*, Brooklyn, NY.

98. Sacks, J., Welch, W.J., Toby, J.M., and Wynn, H.P., 1989, “Design and Analysis of Computer Experiments,” *Statistical Science*, 4(4), pp. 409-423.

99. SAP, 2010, http://www.sap.com/index.epx (2010), SAP (Last time accessed: May 30, 2012).

100. Shoham, Y., Powers, R., and Grenager, T., 2007, "If Multi-Agent Learning Is the Answer, What is the Question?" *Artificial Intelligence*, 171(7), pp. 365–377.

101. Siddiqui, S., Azarm, S., Gabriel, S., 2010, "A Modified Benders Decomposition Method for Efficient Robust Optimization under Interval Uncertainty," *Structural and Multidisciplinary Optimization*, 44(2), pp. 259-275.

102. Simpson, T.W., Korte, J.J., Mauery, T.M., and Mistree, F., 2001a, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, 39(12), pp. 2233-2241.

103. Simpson, T., Poplinski, J., Koch, P., and Allen, J., 2001b, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, 17(2), pp. 129-150.

104. Sobieszczanski-Sobieski, J., 1982, "A Linear Decomposition Method for Large Optimization Problem - Blueprint for Development," *NASA TM 83248*.

105. Sobieski, I.P., Manning, V.M., and Kroo, I.M., 1998, "Response Surface Estimation and Refinement in Collaborative Optimization," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium Multidisciplinary Analysis and Optimization*, St. Louis, MO.

106. Sobieszczanski-Sobieski, J., 1998, "Optimization by Decomposition: A Step From Hierarchic to Non-Hierarchic Systems," *Proceedings of the 2nd NASA/Air force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, Hampton, VA

107. Sobieszczanski-Sobieski, J., Agte, J., and Sandusky, J.R., 2000, "Bi-Level Integrated System Synthesis (BLISS)," *AIAA Journal*, 38(1), pp. 164-172.

108. Taguchi, G., 1987, *Systems of Experimental Design*, Kraus International, New York.

109. Tappeta, R.V., Renaud, J.E., 1997, "Multiobjective Collaborative Optimization," *Journal of Mechanical Design*, 119(3), pp. 403-411

110. Tosserams, S., Etman, L.F.P., Rooda, J.E., 2009, "A Classification of Methods for Distributed System Optimization Based on Formulation Structure," *Structural and Multidisciplinary Optimization*, 39(5), pp. 503–517.

111. Viana, F.A.C., and Haftka, R.T., 2008, "Using Multiple Surrogates for Metamodeling," *7th ASMO-UK/ISSMO International Conference on Engineering Design Optimization*, Bath, UK.

112. Voutchkov, I., and Keane, A., 2010, *Multi-objective Optimization Using Surrogates*, *Computational Intelligence in Optimization*, Springer-Verlag, Berlin Heidelberg, pp. 155-175.

113. Wang, G., and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," Journal *of Mechanical Design*, 129(4), pp. 370-380.

114. Wang, C., 2005, *Enterprise Integration of Management and Automation in a Refinery*, *Knowledge and Skill Chains in Engineering and Manufacturing Information infrastructure in the Era of Global Communications*, Springer, pp. 321-328.

115. Wang, Z., Azarm, S. and Kannan, P.K., 2011, "Strategic Design Decisions for Uncertain Market Systems Using an Agent Based Approach," *Journal of Mechanical Design*, 133(4), pp. 041003-1 to 041003-11.

116. Wu, J., and Azarm, S., 2001, "Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set," *Journal of Mechanical Design*, 123(1), pp. 18-25.

117. Youn, B. D., Choi, K. K., and Park, Y. H., 2003, "Hybrid Analysis Method for Reliability-Based Design Optimization," *Journal of Mechanical Design*, 125(2), pp. 221-232.

118. Zadeh, P., Toropov, V., and Wood, A., 2009, "Metamodel-Based Collaborative Optimization Framework," *Structural and Multidisciplinary Optimization*, 38(2), pp. 103-115.

119. Zhang, N., and Zhu, X.X., 2000, "A novel modelling and decomposition strategy for overall refinery optimisation," *Computers & Chemical Engineering*, 24(2-7), pp. 1543-1548.

120. Zou, T., and Mahadevan, S., 2006, "A Direct Decoupling Approach for Efficient Reliability-Based Design Optimization," *Structural and Multidisciplinary Optimization*, 31(3), pp.190-200.