Res Eng Design
DOI 10.1007/s00163-011-0115-8

ORIGINAL PAPER

An information-passing strategy for achieving Pareto optimality in the design of complex systems

Francesco Ciucci · Tomonori Honda ·
Maria C. Yang

Received: 13 October 2010 / Revised: 22 April 2011 / Accepted: 17 June 2011
© Springer-Verlag London Limited 2011

Abstract As engineering systems grow in complexity, it becomes more challenging to achieve system-level designs that effectively balance the trade-offs among subsystems. Lewis and others have developed a well-known, traditional game-theoretic approach for formally modeling complex systems that can locate a Nash equilibrium design with a minimum of information sharing in the form of a point design. This paper builds on Lewis' work by proposing algorithms that are capable of converging to Pareto-optimal system-level designs by increasing cooperation among subsystems through additional passed information. This paper investigates several forms for this additional passed information, including both quadratic and eigen-based formulations. Such forms offer guidance to designers on how they should change parameter values to better suit the overall system by providing information on directionality and curvature. Strategies for representing passed information are examined in three case studies of 2- and 3-player scenarios that cover a range of system complexity. Depending on the scenario, findings suggest that passing more information generally leads to convergence to a Pareto-optimal set. However, more iterations may be required to reach the Pareto set than if using a traditional game-theoretic approach.

Keywords Complex system design · Distributed design · Design optimization · Pareto optimality

1 Introduction

In recent years, engineering systems have become steadily more complex, resulting in an increasing number of disparate subsystems that must be designed, developed, and integrated. Each subsystem can be designed separately by groups of experts from different fields who are in principle geographically dispersed. This makes information sharing and communication between these subsystem experts logistically challenging. In organizations, it has been shown that even modest geographical distribution can decrease information sharing dramatically (Allen 1984). Olesen and Myers (1999) demonstrated that even careful use of new information communication technologies, it is difficult to overcome the geographical information barrier. Subsystem designers may also further compete with each other for a limited set of system-level resources. Under such circumstances, it becomes critical to understand the trade-offs that must be made between subsystems in order to reach a final design. An example of a complex engineering design process can be found in NASA's Space Mission Design program that focuses on highly specialized subsystems such as propulsion, power, and attitude control

Francesco Ciucci and Tomonori Honda equally contributed to this paper.

F. Ciucci
Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences, Universität Heidelberg, Heidelberg, Germany
e-mail: francesco@alumni.caltech.edu

T. Honda (✉)
Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA
e-mail: tomonori@mit.edu

M. C. Yang
Department of Mechanical Engineering and Engineering System Division, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA
e-mail: mcyang@mit.edu

Published online: 12 July 2011

systems. One of NASA Jet Propulsion Laboratory's organizational approaches to addressing trade-offs is the Advanced Projects Team (Team X) (Jet Propulsion 2010; Mark 2002; Smith et al. 1999). In Team X, the entire team is geographically co-located to facilitate rapid communication between subsystem designers and a system-level facilitator. This arrangement helps ensure that design trade-offs are understood at both the subsystem and system level (Avnet 2009). Balancing trade-offs is not only an issue for space missions, but for other large systems such as aircraft, automobiles, ships, and buildings. However, many complex systems are designed by geographically distributed teams, which makes sharing information and balancing trade-offs even more challenging.

One formal approach to addressing complex engineering systems is through the integration of multiple subsystems using techniques such as game-theoretic models of design (Franssen and Bucciarelli 2004; Keeney 2009; Lewis 1996; Lewis and Mistree 1997) and multidisciplinary design optimization (MDO) (Park 2007; Sobieszczanski-Sobieski 1988). This paper proposes and evaluates algorithms for these types of system-level design models that can improve the way information is passed among its subsystems so that designers are more fully informed and can make better decisions. In a traditional game-theoretic approach, complex engineering systems are represented as a set of distinct subsystems that pass information among each other. Traditional game-theoretic approaches generally converge to a Nash equilibrium, though that equilibrium may not be an optimal solution (Verbeeck et al. 2002). Furthermore, traditional game-theoretic models pass information in the form of a "point" design rather than a set of multiple solutions as found in a Pareto frontier. In traditional MDO, a complex system is modeled as a set of subsystems orchestrated by an overall system facilitator, rather than permitting subsystems to pass information to each other directly. The information passed in traditional MDO takes the form of a gradient. The designer can use the gradient's magnitude and direction to determine whether to increase or decrease the value of a particular parameter. This type of information sharing results in a local optimum solution.

This paper investigates the value of increasing the amount of information that is passed between subsystems and builds on Lewis's traditional game-theoretic approach. It is posited that this additional information can result in a system design that is Pareto-optimal for the cases described in this paper. In a Pareto-optimal set, improvement in any design objective requires a sacrifice in at least one of the other design objectives. The resulting solution is also globally optimal assuming that the priorities for the objectives are appropriate (Scott 1999; Scott and Antonsson 2000). In the proposed approach, information is passed in the form of a Hessian and can be compressed in size using eigenvalue decomposition. Like a gradient, the Hessian provides directionality information, but also offers information on the curvature. In effect, the proposed framework makes the traditional game-theoretic approach partially cooperative. The goal of this approach is to provide teams who design complex systems a framework by which to arrive at desirable optimal designs through the use of slightly more information sharing than in traditional game-theoretic approaches.

2 Background

This work draws on several perspectives on integrating multiple subsystems in design, including game theory and design optimization. The approach employed by these strategies involves decomposing a system into subsystems that must interact with each other according to a specific set of protocols or guidelines. This section presents related literature on traditional game theory and optimization, along with the gap that this paper seeks to fill.

2.1 Traditional game-theoretic approach

Traditional game-theoretic design attempts to identify the most rational design under a specific set of conditions by allowing only limited information to be passed between designers. The resulting designs may differ depending on the type and quantity of information exchanged. Though an individual subsystem designer may make rational design choices, the resulting design will not necessarily be optimal.

The use of game theory as a methodology for design was first proposed by Vincent (1983) and further developed by Lewis, Mistree, and others (Lewis 1996; Whitfield et al. 2002). These traditional game-theoretic approaches have further been combined with decision-based design (Hazelrigg 1998) and adopted in a broad range of design research (Chanron and Lewis 2004, 2005; Chanron et al. 2005a, b; Fernandez et al. 2005; Xiao et al. 2005) to become a prominent framework for the study of design optimization problems (Lewis et al. 2006). In decision-based design, the main activity of a designer is to make decisions that are determined by optimizing individual objectives within a group using game theory tools. The final design depends on how much information is communicated among the subsystems. Lewis and Mistree represent this information sharing using game theory (Lewis and Mistree 1999, 2001) as described for a two-designer scenario outlined in Table 1.

Mathematically, Table 1 states that the purpose of the design (or game) is to minimize the objective functions F_i for each subsystem under given constraints. Here, $\mathbf{x}_i$

Table 1 Lewis and Mistree's distributed design formulation (Lewis and Mistree 1999, 2001)

Designer 1	Designer 2
Minimize	Minimize
$\mathbf{F}_1(\mathbf{x}_1, \mathbf{x}_{2c}) = \{F_1^1, F_1^2, \ldots, F_1^p\}$	$\mathbf{F}_2(\mathbf{x}_{1c}, \mathbf{x}_2) = \{F_2^1, F_2^2, \ldots, F_2^q\}$
subject to	subject to
$g_{j_1}^1(\mathbf{x}_1, \mathbf{x}_{2c}) \leq 0, \quad j_1 = 1, \ldots, m_1$	$g_{j_2}^2(\mathbf{x}_{1c}, \mathbf{x}_2) \leq 0, \quad j_2 = 1, \ldots, m_2$
$g_{h_1}^1(\mathbf{x}_1, \mathbf{x}_{2c}) = 0, \quad h_1 = 1, \ldots, l_1$	$g_{h_2}^2(\mathbf{x}_{1c}, \mathbf{x}_2) = 0, \quad h_2 = 1, \ldots, l_2$
$x_{1L} \leq x_1 \leq x_{1U}$	$x_{2L} \leq x_2 \leq x_{2U}$

represents design variables controlled by designer i and $\mathbf{x}_{ic}$ describes a nonlocal variable that is determined by the other designer (subscript $(\cdot)_c$ denotes a constrained variable). In this game theory representation of distributed design, several information-sharing protocols can be used in order to reach an agreement, each resulting in a different rational/optimal set of designs. The following subsections describe protocols of information sharing among designers.

2.1.1 Cooperative protocol

In a completely cooperative protocol, all information about a subsystem is shared among all designers. Each designer has sufficient information about all subsystems to work together with others to find a set of Pareto-optimal solutions. Thus, this protocol allows a design team to select among the best possible design performances of all the protocols. Lewis mentions that such a *systems thinking* approach allows members in a design team to focus on cooperation through a common vision (Lewis and Mistree 1999; Senge 1990). Even though this protocol is ideal for an organization, it is rarely feasible in a geographically distributed environment in part because of communication and logistical challenges. It becomes even more impractical to share all information when subsystems are technologically complex or disparate.

2.1.2 Noncooperative protocol

A noncooperative protocol occurs when there is imperfect information sharing between subsystems. This means that different amounts of information are shared between subsystems. In a game-theoretic construct of noncooperative protocol, there is a special case in which two individuals with competing objectives converge to a single Nash equilibrium (Chanron and Lewis 2005). A Nash Equilibrium represents a point in design space in which every designer cannot improve a design without breaking constraints given by other subsystems. The constraints imposed by other subsystems represent the limitation of information sharing. As more information is shared between subsystems, there will be fewer constraints imposed by one subsystem on the other subsystems. A Nash Equilibrium solution may not necessarily be in a Pareto set, but it is the most rational solution given the lack of information between subsystems.[1]

2.1.3 Leader/follower protocol

In a leader/follower protocol, a system is designed sequentially. The first designer designs his subsystem completely, then passes the subsystem design to the next subsystem designer. The final solution of a leader/follower protocol is called a Stackelberg solution (Rao et al. 1997; Simaan and Cruz 1973) and generally differs from a Nash solution. This approach is challenging because it requires the first designer to make reasonable assumptions about other subsystem designs, which is rare in large, complex systems.

2.2 Optimization perspective

Design in a distributed environment can also be addressed from an optimization point of view. Mathematically, engineering design may be formalized as a global multi-objective optimization problem (de Weck 2004). In this formulation, optimization will not necessarily lead to a single "best" design, but rather a full set of optimal design choices that lie on a Pareto frontier. Unlike results found using decision theory, this set should depend only on the design objectives and design constraints, not on the information shared or on the rationality of the decision-makers.

One common approach to accomplishing this is Multidisciplinary Design Optimization (MDO) (Agte et al. 2010; Cramer et al. 1993; Park 2007; Yi et al. 2008). MDO was first formulated by Sobieszczanski-Sobieski as Concurrent Subspace Optimization (CSSO) (Sobieszczanski-Sobieski 1988). Other classes of MDO developed later include Collaborative Optimization (CO) (Tappeta and Renaud 1997), Bi-Level Integrated System Synthesis (BLISS) (Sobieszczanski-Sobieski et al. 1998), Multiple-Discipline-Feasible (MDF), Individual-Discipline-Feasible (IDF), All-at-once (AAO), Multidisciplinary Optimization based on Independent Subspaces (MDOIS) (Park 2007; Shin and Park 2005) and Multi-Objective Pareto Concurrent Subspace Optimization (Huang et al. 2007). These can be generally classified into hierarchical and nonhierarchical methods depending on the role of the system and subsystems.

There is a range of new research on MDO as well as possible improvements (Agte et al. 2010). Zhao and Cui

[1] This is a rational solution rather than an optimal solution because there could be a design that is better than a rational solution for every design objective. However, if there is too little information shared between one subsystem and another, there will be no deterministic method to find these optimal solutions.

have developed a new MDO formulation called BLISCO by combining BLISS and CO (Zhao and Cui 2011). Tao et al. have increased the computational efficiency of CSSO and BLISS (Tao et al. 2010). Tosserams, et al. have developed a new, more sophisticated benchmark problem for comparing MDO coordination and optimization algorithms (Tosserams et al. 2010). Allison, et al have improved the partitioning required for a hierarchical formulation by considering coordination simultaneously. (Allision et al. 2009). Finally, Shaja and Sudhakar have formulated a method to optimize the sequence of analysis for multidisciplinary design (Shaja and Sudhakar 2010).

The main difference between MDO and the approach described in this paper is that this strategy does not require a system facilitator who combines design from multiple disciplines into a coherent design through optimization (Park 2007). Lu and Kim (2010) have created a set of new MDO approaches called MDO-CC that converge to game-theoretic equilibrium such as a Stackelberg solution or a Nash equilibrium. Note that this formulation still requires centralization of information. It is often the case that subsystems interact with each other without the guidance of a system designer, especially in the case of highly coupled systems. Thus, this approach can represent the design process for highly coupled system.

The approaches to integrating multiple subsystems described in this paper tend to focus on strategies and protocols for sharing information between subsystems or facilitators. An area that has had limited focus is the format of the information that is passed between entities. Generally speaking, game theory assumes that information is passed in the form of a single point design and does not include information about whether a design parameter should increase or decrease, or what value the parameter should take on. MDO assumes that passed information takes the form of a gradient that provides a sense of the directionality (increasing or decreasing) that a parameter should take. This paper seeks to fill that gap by introducing a partially cooperative protocol that enables the sharing of a quadratic function between each designer. This will provide additional directionality and curvature information that can assist in the formation of more successful designs by increasing the amount of information shared. Therefore, the proposed game-theoretic approach follows a more cooperative protocol.

3 Mathematical formulation of a distributed design problem

This paper focuses on the class of distributed design problems posed by Lewis (1996) as shown in Table 1 and equivalently rewritten in Table 2. These problems were chosen because they were well scoped for further study and would illustrate the approach appropriately. The analysis of the design problem will be initially limited to two players who are optimizing two different objective functions F_1 and F_2, depending upon the variables $\mathbf{x}$. Note that each subsystem can be formally defined as $\tilde{f}_i(x_1, x_2, x_{i,l})$ where $x_{i,l}$ is a local variable for a subsystem that other subsystems do not have access to. The main factor that influences the classification of the variables are the global sensitivity (Homma and Saltelli 1996) of $\tilde{f}_i$ with respect to the variables. Variables with the highest (relative) global sensitivity should be considered critical variables and should be shared, while variable with lower sensitivity can remain local. After shared and hidden variables are decided, the subsystem function can be redefined as:

$$F_i(x_1, x_2) = \min_{x_{i,l}} \tilde{f}_i(x_1, x_2, x_{i,l}) \tag{1}$$

Table 2 Proposed distributed design formulation

	Subsystem Designer 1	Subsystem Designer 2
Obj. function	$\min\limits_{x=(\mathbf{x}_1,\mathbf{x}_2)} F_1(\mathbf{x}_1, \mathbf{x}_2)$	$\min\limits_{\mathbf{x}=(\mathbf{x}_1,\mathbf{x}_2)} F_2(\mathbf{x}_1, \mathbf{x}_2)$
Constraints	$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{U}_1$	$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{U}_2$

Each subsystem is allowed to modify the full set of design variables. At the same time, it needs to satisfy its own local constraints

These objectives represent different design goals given by performance, utility, or preference variables. Furthermore, each player is constrained to be in some set $\mathcal{U}_i$. Because each player passes sets of constraints, this approach is also similar to Ward's set-based concurrent engineering (Ward 1989; Ward et al. 1990) in which sets of intervals for feasible designs are passed iteratively between design groups in order to converge toward a "good" design. In this case, approximations of objective functions and constraints are exchanged as constraints rather than as feasible intervals (see Table 3), thus increasing the amount of passed information.

The procedure necessary to solve the above optimization problem will be determined by modifying the algorithm based on the ε-constraint method (Haimes 1973; Marglin 1967) and the method of equality constraints (Das and Dennis 1997; Lin 1976). The approach is illustrated in

Table 3 The forms of passed information examined in this paper are listed below in order of decreasing amount of information

Problem information type	Shared information form
Objective function	Exact surface
	Quadratic approx. of the surface
	Eigen-based quadratic approx. of the surface
Constraints	Exact constraint
	Linear approx. of the constraint

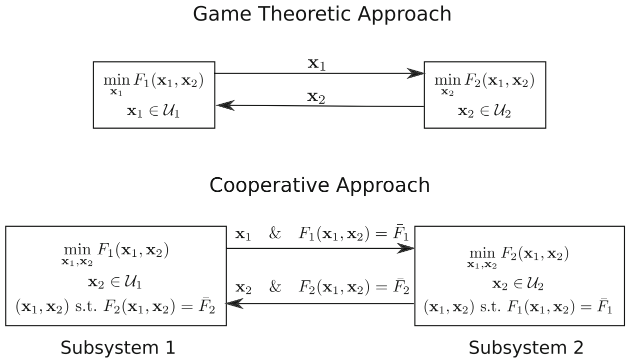
Game Theoretic Approach
$\min_{\mathbf{x}_1} F_1(\mathbf{x}_1, \mathbf{x}_2)$
$\mathbf{x}_1 \in \mathcal{U}_1$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\min_{\mathbf{x}_2} F_2(\mathbf{x}_1, \mathbf{x}_2)$
$\mathbf{x}_2 \in \mathcal{U}_2$
Cooperative Approach
$\min_{\mathbf{x}_1,\mathbf{x}_2} F_1(\mathbf{x}_1, \mathbf{x}_2)$
$\mathbf{x}_2 \in \mathcal{U}_1$
$(\mathbf{x}_1, \mathbf{x}_2)$ s.t. $F_2(\mathbf{x}_1, \mathbf{x}_2) = \bar{F}_2$
$\mathbf{x}_1$ & $F_1(\mathbf{x}_1, \mathbf{x}_2) = \bar{F}_1$
$\mathbf{x}_2$ & $F_2(\mathbf{x}_1, \mathbf{x}_2) = \bar{F}_2$
$\min_{\mathbf{x}_1,\mathbf{x}_2} F_2(\mathbf{x}_1, \mathbf{x}_2)$
$\mathbf{x}_2 \in \mathcal{U}_2$
$(\mathbf{x}_1, \mathbf{x}_2)$ s.t. $F_1(\mathbf{x}_1, \mathbf{x}_2) = \bar{F}_1$
Subsystem 1
Subsystem 2

Fig. 1 Flowchart illustrating information passing between subsystems in a two-player scenario for traditional game-theoretic (*top*) and proposed cooperative approach (*bottom*). Each *block* represents a subsystem. In the *bottom panel*, the values of the constants are given by the design point at the previous step, i.e., $\bar{F}_1 = F_1\left(\mathbf{x}_1^{(k,1)}, \mathbf{x}_2^{(k,1)}\right)$ and $\bar{F}_2 = F_2\left(\mathbf{x}_1^{(k,2)}, \mathbf{x}_2^{(k,2)}\right)$

Fig. 1. Subsystem Designer 1 starts from a point in $(\mathbf{x}_1^{(1,1)}, \mathbf{x}_2^{(1,1)})$ in design space and minimizes his objective function $F_1(\mathbf{x}_1, \mathbf{x}_2)$ and is constrained to the intersection of the sets $\mathcal{U}_1$ and $\mathcal{A}_1^{(1)}$ where $\mathcal{A}_1^{(1)} = \{\mathbf{x} | F_2(\mathbf{x}_1, \mathbf{x}_2) = F_2\left(\mathbf{x}_1^{(1,1)}, \mathbf{x}_2^{(1,1)}\right) = \bar{F}_2\}$. He will choose a minimization point $\left(\mathbf{x}_1^{(1,2)}, \mathbf{x}_2^{(1,2)}\right)$ and pass it to the subsystem Designer 2 along with the $\mathcal{A}_2^{(1)} = \left\{\mathbf{x} | F_1(\mathbf{x}_1, \mathbf{x}_2) = F_1\left(\mathbf{x}_1^{(1,2)}, \mathbf{x}_2^{(1,2)}\right) = \bar{F}_1\right\}$. Subsystem Designer 2 will then minimize the objective function $F_2(\mathbf{x}_1, \mathbf{x}_2)$ constrained to the intersection of the sets $\mathcal{U}_2$ and $\mathcal{A}_2^{(1)}$. Subsystem Designer 2 will choose a minimization point $\left(\mathbf{x}_1^{(2,1)}, \mathbf{x}_2^{(2,1)}\right)$ and pass it to the subsystem Designer 1 along with the $\mathcal{A}_1^{(2)}$. Subsystem Designers 1 and 2 will then iterate until they converge on a design. This will be repeated at the k-th steps: Designer 2 will pass to Designer 1 $\left(\mathbf{x}_1^{(k,1)}, \mathbf{x}_2^{(k,1)}\right)$ along with the $\mathcal{A}_1^{(k)}$ and Designer 1 will share with Designer 2 $\left(\mathbf{x}_1^{(k,2)}, \mathbf{x}_2^{(k,2)}\right)$ along with the $\mathcal{A}_2^{(k)}$. It is important to note that the addition of the surface constraint restricts the design space.

3.1 Approximation of the information passed

This section of the paper describes how the equality constraint method can be modified in a way that reduces information flow between subsystems and yet allows convergence to a set of a Pareto-optimal designs. In principle, passing a full surface[2] requires less information sharing than fully cooperative protocols. However, exchanging surface data requires significant time and effort due to the potential complexity of the surface. To reduce the quantity of information shared, the surface $\mathcal{A} = \{\mathbf{x} | F(\mathbf{x}) = F(\mathbf{x}_0)\}$ can be approximated quadratically as:

$$\mathcal{A} \approx \{\mathbf{x} \,|\, \nabla F(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) + (\mathbf{x} - \mathbf{x}_0) \cdot \mathrm{He}_F(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) = 0\} \tag{2}$$

where the Hessian $\mathrm{He}_F\ (\mathbf{x_0})$ is the matrix of second derivatives defined as $(\mathrm{He}_F(\mathbf{x_0}))_{i,j} = \frac{\partial^2 F}{\partial x_i \partial x_j}$. Both the gradient and Hessian can be numerically computed and exchanged between subsystems much more efficiently than a complete surface. Also, the approximation of the surface near the Pareto set improves as a design converges.

Passing information in gradient and Hessian form requires information transfer of order n^2 (n^2 numbers will be exchanged), where n is the dimension of design space. By comparison, MDO frameworks pass a local objective function from subsystems to the overall system facilitator using gradients of order n. Thus, in engineering design problems, it is considered reasonable to assume that information of order n is exchanged. When the dimension of design space increases, this difference between n^2 and n may become significant, but the quadratic approximation can be further reduced using the Eckart–Young theorem (Eckart and Young 1936; Stewart 1993). In order to minimize the approximation error (in Matrix norm) between any matrix M and its reduced rank approximation $\tilde{M}$, one needs to retain the desired number of largest singular values. Because the Hessian is symmetric and real, there is always an orthonormal basis of eigenvectors. Hence, to approximate $Q_F = (\mathbf{x} - \mathbf{x}_0)^T \mathrm{He}_F\,(\mathbf{x} - \mathbf{x}_0)$, it is sufficient to compute the set of eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ and eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of He_F satisfying $\mathrm{He}_F\ \mathbf{u}_i = \lambda_i\ \mathbf{u}_i$ where $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$, and truncate $Q_F = (\mathbf{x} - \mathbf{x}_0)^T (U\,\mathrm{diag}(\lambda) U^T)(\mathbf{x} - \mathbf{x}_0)$ at the desired order $K \approx (\mathbf{x} - \mathbf{x}_0)^T \left(\sum_i^K \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T\right)(\mathbf{x} - \mathbf{x}_0)$. Furthermore, the singular values are the absolute values of eigenvalues. Thus, the Hessian can be approximated using a subset of the eigenvalues and corresponding eigenvectors. In this paper, the Hessian is approximated using only one eigenvalue with the largest absolute value and its corresponding eigenvector.

Additionally, because a quadratic approximation is used for the surface, this approach is similar to sequential quadratic constrained quadratic programming (SQCQP) (Jian et al. 2007) for each optimization step. In a mathematically rigorous way, SQCQP shows that a quadratic approximation of the nonlinear constraint can achieve local optimality. Here, in discordance with SQCQP, the objective function does not need to be approximated as a quadratic function. Each subsystem designer considers the objective functions of other subsystem designers as quadratic constraints. There are other key issues to consider, such as the

[2] In this case, a surface represents a contour of the desired objective value in design variable space.

existence of the Pareto frontier for constrained games, necessary regularity conditions (Ding et al. 2003; Ding 2000; Yu 2003), algorithmic and numerical issues (Fliege and Svaiter 2000; GarcÌa-Palomares et al. 2008; Laumanns et al. 2006; Wang and Zhang 2008), which are explained in the vast body of operations research literature (Chinchuluun and Pardalos 2007; Dauer and Stadler 1986; Marler and Arora 2004).

Finally, in previous work (Honda et al. 2007), it was assumed that all constraints are shared. In this paper, this assumption is relaxed in order to allow the subsystem designers to pass a local linear approximation of unshared constraints. This approach is applied in the last two case studies. Note that if a linear approximation does not suffice, it would also be possible to switch from linear to eigen-based quadratic approximation only for active constraints.

In Lewis and Mistree's formulation, the convergence and stability of the underlying algorithm is not guaranteed (Chanron and Lewis K 2004). This paper builds on their approach by expressing convergence criteria more generally. A bounded, monotonic sequence (a contraction) always has a unique point of the convergence; thus, when subsystem designers exchange a complete surface with one another as an equality constraint, each subsystem's design space (objective with constraints) will always be contracted, guaranteeing the convergence in the full surface case. If the information each subsystem designer passes is approximated by a constraining surface, an approximation error will be introduced. It is clear that if the erroneous increase in the objective value caused by this approximation error is smaller compared to each objective's improvement for every iteration, then the sequence will also be monotonically decreasing. Thus, the key ingredient for guaranteeing convergence is within the accuracy of the approximation (see "Appendix").

4 Three case studies

4.1 Case study 1: mathematical example

This first example examines a purely academic problem. Each subsystem designer's design objective and constraints are given as follows:

Design problem for Designer 1
Minimize:
$F_1(x_1, x_2) = (x_1^2 - 8x_1 + 4x_1x_2) - 0.1x_2\sin(2x_1)$
Design variables: x_1 and x_2
Constraints: $x_1 \in [0, 10]$ and $x_2 \in [0, 10]$

Design problem for Designer 2
Minimize: $F_2(x_1, x_2) = x_2^2 - 8x_1x_2 - 0.1x_1\cos(x_2)$
Design variables: x_1 and x_2
Constraints: $x_1 \in [0, 10]$ and $x_2 \in [0, 10]$

Figure 2 shows the Nash and Pareto sets for baseline comparison purposes. The asterisks on the figures represent the Nash solutions using a traditional game-theoretic approach. The Nash solution is not a stable equilibrium in this case, but it converges to a limit cycle that loops between the two points. Interestingly, the region of attraction for this limit cycle is the whole domain. In other words, a traditional game-theoretic approach converges to the same limit cycle independent of the starting point. As can be seen from the figure, the Pareto frontier is non-convex in design space. Finally, these results demonstrate that the Nash Equilibrium may not necessarily be a Pareto-optimal Solution.

Next, the proposed algorithms were implemented for an initial point chosen from uniform random distribution over the whole feasible domain, i.e., $x_1^0 = 10\ U(0,1)$ and $x_2^0 = 10\ U(0,1)$ (see Fig. 3). Table 4 shows that even though all of the proposed algorithms converge to a Pareto set, the number of iterations necessary for convergence depends strongly on the accuracy of the approximation. One parameter that describes the convergence is the average number of iterations $\overline{\text{iter}}$, which is defined as $\overline{\text{iter}} = \sum_{k=1}^{N} \text{iter}(k)$. In the latter, iter($k$) is the number of iterations necessary to converge starting at point x_k, and $\{x_1, \ldots, x_N\}$ is the set of randomly chosen initial points. Similarly, the standard deviation of the iterations is defined as $\sigma(\text{iter}) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\text{iter}(k) - \overline{\text{iter}}\right)^2}$. Note that the number of iterations for the game-theoretic approach is important for comparison purposes, but is not listed because the approach does not necessarily converge for the given initial conditions. Unexpectedly, even though information sharing does decrease using an eigen-based approximation, the number of iterations required for convergence increases on average by 6.6 times. These results suggest that there is a trade-off between the number of iterations and the amount of information shared. These computational results also show that passing the quadratic approximation (including the Eckart–Young approximation of the Hessian) for each subsystem designer's objective function is sufficient to obtain Pareto-optimal design solutions. Finally, Table 4 lists the standard deviation with respect to the number of iterations. By increasing the approximation error, the average number of iterations increases and so does the variability on the number of iterations necessary to achieve convergence.

Note that this algorithm will only converge to a single design on the Pareto frontier for each initial point. It will require design teams to re-run this algorithm with many different initial designs to obtain the Pareto set. However, our assumption is that for many design teams, the goal is not to determine the entire

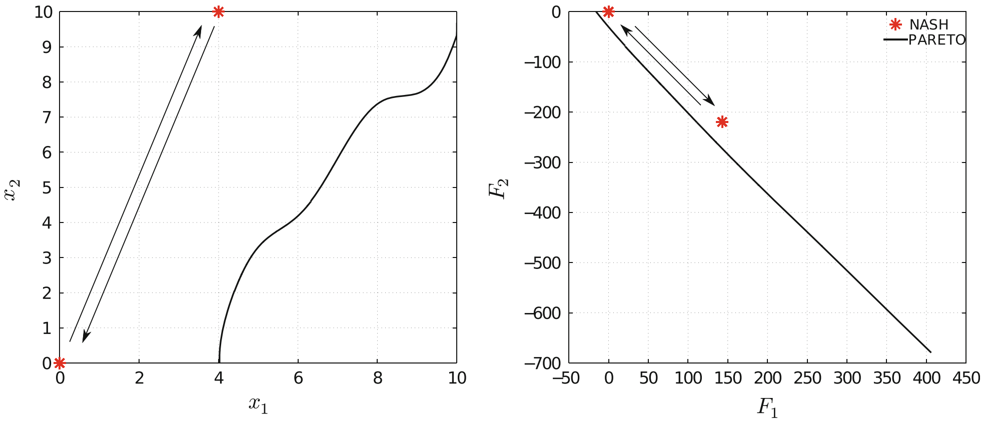

Fig. 2 Pareto frontier in design (*left*) and performance (*right*) space for Case study 1. The traditional game-theoretic protocol converges to a limit cycle and loops between the two asterisks, while the Pareto set is indicated by *dots*. Note that the Nash solution *does not converge to an equilibrium point*, and the Pareto frontier is neither convex not concave in design space

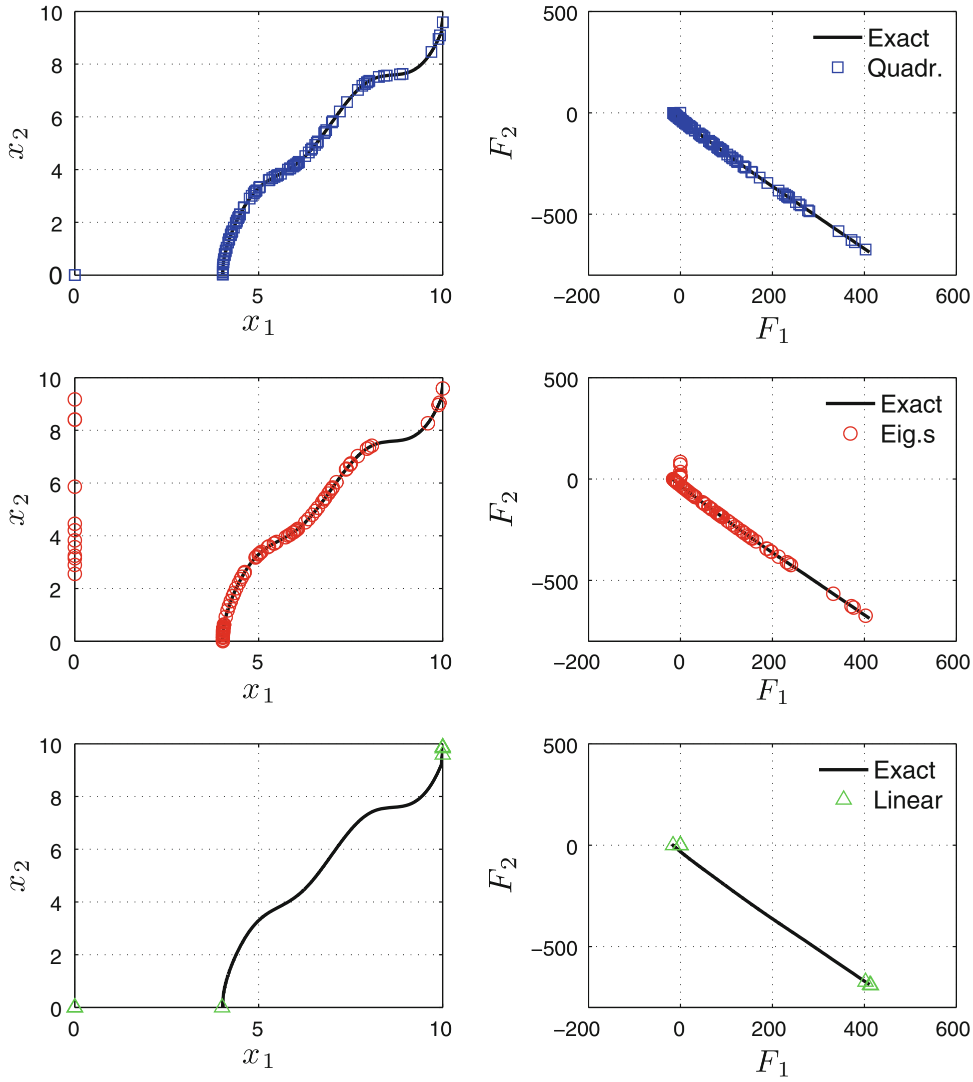

Fig. 3 Pareto frontier in design (*left*) and performance space (*right*) for Case study 1. These plots shows the result of several algorithms approximating the shared equality constraint for passed information. The "Exact" strategy consists in passing the exact form of the complete surface, and the quadratic ("Quad") approximation consists in sharing the quadratic approximation of the complete surface. "Eig.s" corresponds to the eigen-based quadratic approximation and "Linear" to the linear approximation

Pareto frontier, but rather to select one "good" design. Thus, the intent of our algorithm is to help the design team in converging to a single design on the Pareto frontier. The design team should only iterate by re-running different initial designs if they final design is not "good" enough.

Table 4 Average number of iterations necessary for convergence in Case study 1 along with standard deviations

Method	Average number of iterations	Standard deviation of the number of iterations
Exact surface	1.00	0.00
Quad approx.	5.9690	2.2435
Eigen decomp. approx.	6.6410	4.4793

4.2 Case study 2: aircraft design

This second example is based on an aircraft example given in the papers of Lewis and Mistree (1997) and in the NASA report of Mistree et al. (1988). There are two subsystems: one optimizes the weight of the aircraft and the other optimizes its aerodynamics.[3] The key equations are the same as described in Lewis' previous work, but one constant b_t and weighting for objectives have been modified to enable more convenient analysis. In particular, the thrust-specific fuel consumption constant b_t has been set to 9.7222×10^{-5} $lb/(lbf\ sec)$ as in a typical airliner (Honda et al. 2007).

Multiple algorithms for approximating the objective function for the new, hybrid formulation were explored. It was expected that there is a trade-off between the amount of information that an algorithm could pass and the number of iterations required to reach convergence to the Pareto set. These algorithms are listed in descending order with respect to the amount of information passed:

1. Complete surface with full constraints: No approximation of the objective function with no approximation of constraints.
2. Quadratic approximation with full constraints: A second-order approximation of objective function with no approximation of constraints.
3. Eigen-based decomposition with full constraints: A quasi-second-order approximation of objective function with no approximation of constraints.
4. Eigen-based decomposition with linearized constraints: A quasi-second-order approximation of objective function with first-order approximation of constraints.

Table 5 Comparison of the number of iterations for convergence for Case study 2 of an aircraft design for traditional game theory, an exact algorithm, and three approximating algorithms. These algorithms are listed in decreasing order of information passed between subsystems

Method	Average number of iterations	Standard deviation of the number of iterations
Exact surface	1.00	0.00
Quad approx.	1.00	0.00
Eigen decomp. approx.	1.00	0.00
Eigen decomp. approx. and linear constraint	1.00	0.00
Traditional game theory	11.00	6.28

In the case of eigen-based quadratic approximation, note that while the game theory approach assumes that the nonlinear inequality constraints are not shared between subsystems, the inequalities are shared between subsystems in linearized form while maintaining an eigen-based approximation for the objective function. Results (Fig. 4) show that the Pareto-optimal set is consistently reached using each one of these approximations. Thus, for this example, if each subsystem designer can pass a quadratic approximation of his or her objective functions, it is sufficient for a system to reach Pareto-optimality.

Table 5 shows that, for this example, all of the above methods converge in just one iteration, and in fewer iterations than in a Lewis' game-theoretic approach, although it does require more information. Given this result, a rational strategy for approximation is that the design team should start the design process using eigen-based approximation first and then improve the approximation if the design does not converge. In this case, because the majority of inequality constraints are inactive, the linearized constraint is usually sufficient for convergence.

4.3 Case study 3: speed reducer

This last example is a case study of a speed reducer for a mechanical transmission. The example is taken from the works of Golinski (1970) and Azarm et al. (1999), Kurpati et al. (2002), and Li and Azarm (2007). In this scenario, there are three subsystem designers. One subsystem designer minimizes the total volume of the speed reducer, and the other two minimize the stresses on their assigned shafts.

The multiplayer algorithm is depicted in Fig. 5. The design iterations start with one designer optimizing his subsystem using a given starting design point ($\mathbf{x}_1^0$, $\mathbf{x}_2^0$, $\mathbf{x}_3^0$) and information from other designers in the form of the exact or approximate equalities (e.g. $\bar{F}_2 = F_2(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ and $\bar{F}_3 = F_3(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ and the inequality constraints. This

[3] Note that our goal is to optimize and balance the aero designers' and weight designers' overall objective rather than individual weight and aerodynamic subsystem objectives. We assume that each subsystem develops its own overall objectives using techniques common in the literature (Cross 2000). This example can be extended to trading each subsystem's goal separately, but this was not within the scope of this example. This example instead shows how to trade-off overall subsystem objectives.

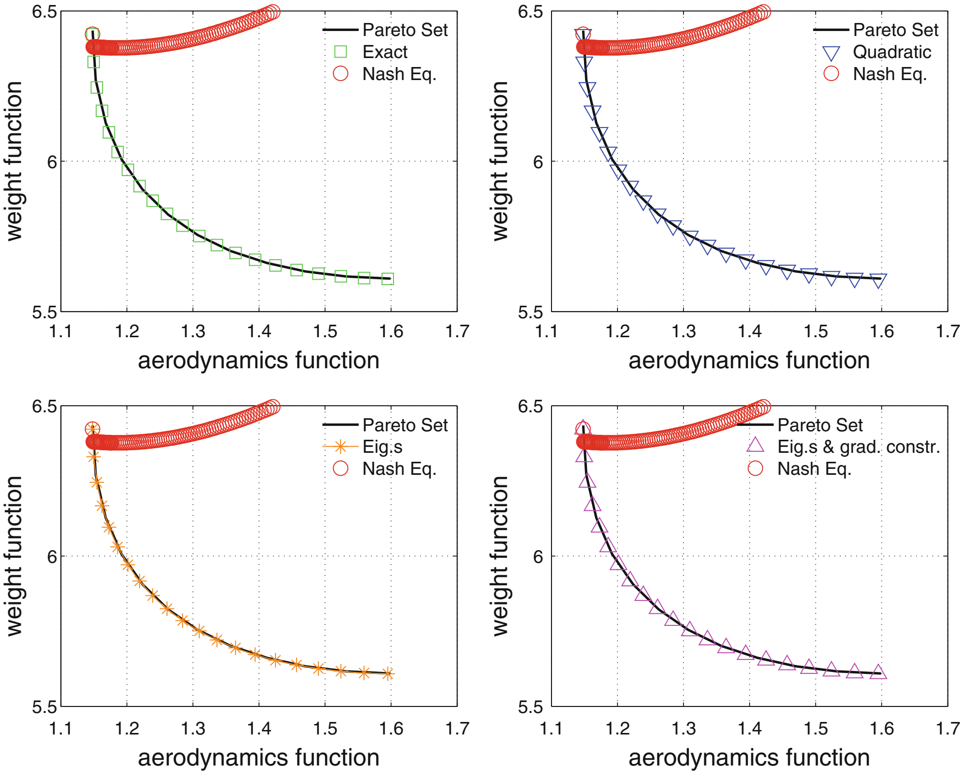

Fig. 4 Pareto frontier in performance space for Case study 2 of an aircraft design and comparison with Nash points, the exact algorithm, and three algorithms of varying levels of approximation. The exact algorithm and the approximating algorithms all converge to the Pareto set, while the Nash points depart significantly from the Pareto set

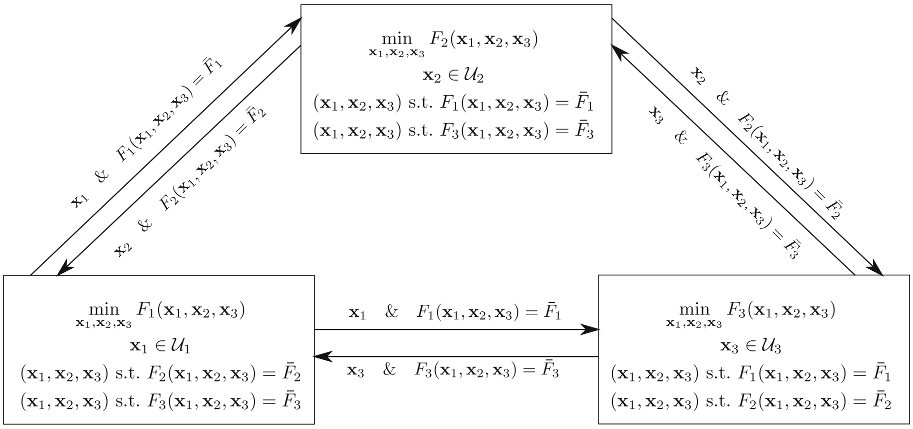
$\min_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3} F_2(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$
$\mathbf{x}_2 \in \mathcal{U}_2$
$(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$ s.t. $F_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_1$
$(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$ s.t. $F_3(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_3$
$\mathbf{x}_1$ & $F_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_1$
$\mathbf{x}_2$ & $F_2(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_2$
$\mathbf{x}_2$ & $F_2(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_2$
$\mathbf{x}_3$ & $F_3(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_3$
$\min_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3} F_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$
$\mathbf{x}_1 \in \mathcal{U}_1$
$(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$ s.t. $F_2(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_2$
$(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$ s.t. $F_3(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_3$
$\mathbf{x}_1$ & $F_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_1$
$\mathbf{x}_3$ & $F_3(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_3$
$\min_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3} F_3(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$
$\mathbf{x}_3 \in \mathcal{U}_3$
$(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$ s.t. $F_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_1$
$(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$ s.t. $F_2(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \bar{F}_2$

Fig. 5 Information flow for Case 3 of a three-player speed reducer design. The three designers, each represented by a block, work sequentially to optimize the performance of their respective subsystems and iterate until there is convergence of the overall system

new design will be passed to Designer 2 along with updated equality and inequality constraints. Designer 2 will undertake a similar minimization procedure. Subsequently, the data will be passed to Designer 3 who will repeat similar steps. The procedure will be repeated until convergence is reached.

As in the previous test case, different algorithms were compared: complete surfaces (no approximation on the objective function), a quadratic approximation, and an eigen-based approximation. The results are presented in Fig. 6. The nonlinear inequality constraints were shared perfectly for the first two protocols and approximated linearly in the third case. Results show that the Pareto-optimal set can be obtained using each of these approximations and are in excellent agreement with the results of Li and Azarm.

Table 6 shows the number of iterations necessary for convergence. Note that while the average is high, so is the standard deviation. It should also be stressed that the exact case converges in a single iteration for the vast majority of initial conditions (89% of the time), and it converges in 2 iterations 4% of the time. The quadratic approximation

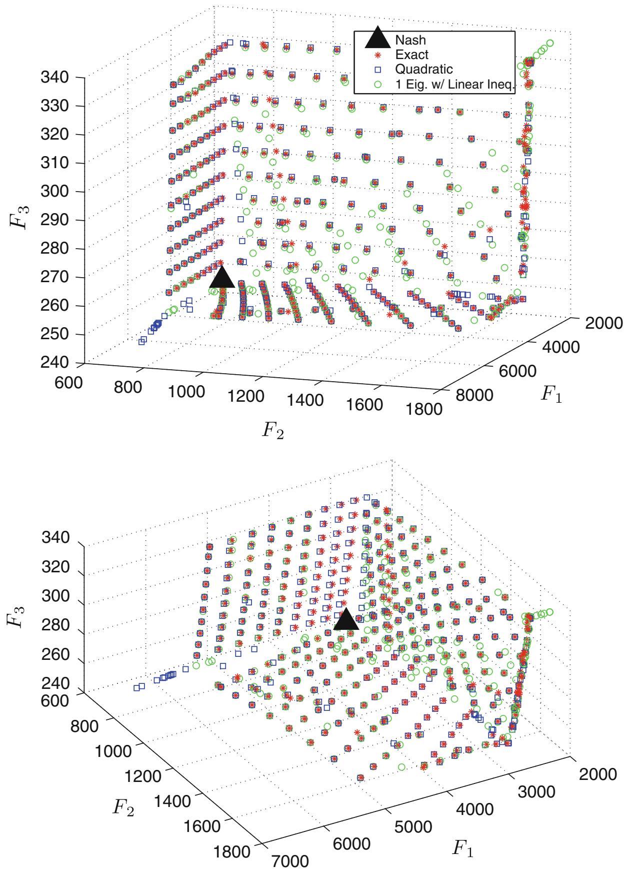

Fig. 6 The Pareto frontier in performance space for Case study 3 of a three-player speed reducer design and comparison with an exact surface, a quadratic approximation, and an eigen-based approximation with linearized constraints. The exact and approximate algorithm converge to the Pareto set. The Pareto Surface is not shown because it is 3D. The Nash point is close enough to the Pareto set that it can approximate a point in Pareto set

Table 6 Number of Iterations for convergence for Case study 3 of a speed reducer in a three-player game

Method	Average number of iterations	Standard deviation of the number of iterations
Exact surface	2.7510	7.4409
Quad approx.	4.2247	7.6838
Eigen decomp. approx. and linear constraint	3.5582	4.9876
Traditional game theory	2.2143	0.4133

Strategies include exact surface (no approximation, full information), quadratic approximation, eigen-based decomposition with linear constraints, and traditional game theory. Algorithms are listed in descending order of quantity of information passed between subsystems

converges in 2 steps 80% of the time. The eigen approximation converges in less than 2 steps in 70% of cases. In this case, the game-theoretic convergence rate is on average more efficient than the proposed algorithm, but in spite of that it is more likely that the cooperative approach converges in fewer iterations than the noncooperative approach. It is also worth noting that the traditional game-theoretic approach is restricted to only one point.

Lastly, it should be pointed out that, because the convergence is usually fast for the proposed algorithms, the location in the Pareto set will strongly depend not only on the initial condition but also on the choices made by the first designer. For example, if Designer 3 starts, the solution will likely converge at the bottom portion of Fig. 6. Starting with Designer 1 will likely lead to the right portion of the plot, and starting with Designer 2 will likely result in the left portion of the plot.

5 Conclusions

In a decentralized environment, Lewis and others have developed an effective, widely used game-theoretic approach based on decision theory, but there exist few other formal design techniques for obtaining rational solutions. In the experiments conducted for this paper, Lewis' approach was found to be reliable in that it generally determined a Nash equilibrium using minimum information sharing. The main question explored in this paper is as follows: if a subsystem is able to share more information than in a noncooperative environment, but less than in a fully cooperative environment, is there an approach that can obtain a Pareto-optimal set? In this paper, subsystem designers were permitted to share several forms of quadratic approximation, including an eigen-based approximation of a Hessian, for each subsystem's objective function at each iteration. In all three case examples, the Pareto sets were obtained by passing along these new approximations. These results also show that linear approximation in subsystem-specific nonlinear inequality constraints is sufficient for convergence in the aircraft and speed reducer examples. Thus, this paper demonstrates two cases in which the Pareto-optimal set of design points can be found in a decentralized design environment by sharing a quadratic approximation to the objective functions as long as each subsystem's objective function is sufficiently regular.

Acknowledgments The work described in this paper was supported in part by the National Science Foundation under Award CMMI-0830134. The opinions, findings, conclusions, and recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsors.

Appendix: Remarks on the algorithm and its convergence

Definition 1 (*Problem statement*) Determine the Pareto set for the problem.

$$\min_{\mathbf{x}\in\mathcal{U}} F_k \quad \text{for } 1 \le k \le N$$

Definition 2 (*Pareto-optimal points*) $\mathbf{x}^* \in \mathcal{U}$ is Pareto-optimal, if it does not exist an $\mathbf{y} \in \mathcal{U}$ such that $F_j(\mathbf{y}) \le F_j(\mathbf{x}^*)$ for all j such that $1 \le j \le N$ and $F_k(\mathbf{y}) \ne F_k(\mathbf{x}^*)$ for some k such that $1 \le k \le N$.

Definition 3 (*Pareto set*) The Pareto set is defined as the set of points $\mathbf{x} \in \mathcal{U}$ that are Pareto-optimal.

We observe that in practical terms, a point is Pareto-optimal if there exists no other point that decreases some objective function without causing a simultaneous increase in at least one objective function.

Definition 4 (*EXACT algorithm*)

1. All designers compute their respective minima $\mathbf{x}_k = \arg\min_{\mathbf{x}\in\mathcal{U}} F_k$;
2. The designers select a starting point $\mathbf{x}_0$ s.t. it is in the convex hull of the Pareto minima;
3. The distributed design process starts from designer p, all other designers $k \ne p$ pass to p their objective function at that point, i.e., $\bar{F}_k = F_k(\mathbf{x}_0)$;
4. The designer p computes $\mathbf{x}_{new} = \arg\min_{\mathbf{x}\in\mathcal{U}} F_p(\mathbf{x})$ such that $F_k(\mathbf{x}_{new}) = \bar{F}_k$ for all $k \ne p$;
5. The design process restarts from $p' \ne p$, goes back to point 3 and continues until convergence is reached.

Remark 1 (Dominated inequalities) The next point found by the algorithm is never worse than the previous. It is sufficient to note that by construction $F_k^{(n)} \le F_k^{(n-1)} \le \cdots \le F_k^{(0)}$ for all k, where the upper index indicates the iteration number.

Proposition 1 *If the Pareto set is convex and of dimension* $N-1$, *then the algorithm is feasible and it converges to the Pareto Set in at most* 1 *iteration.*

Proof Since the Pareto set exists then minima and maxima for the F_j exist in $\mathcal{U}$ and are in the Pareto set. Since the Pareto set is convex then $\mathbf{x}$ such that $\min_{\mathcal{U}} F_j \le F_j(\mathbf{x}) \le \max_{\mathcal{U}} F_j$ for all j. If there exists one point in the Pareto set such that $\bar{F}_j = F_j(\mathbf{x})$ for $j \ne k$, then that point in the Pareto set is found by taking $\mathbf{x} = \arg\min F_k$ with $\bar{F}_j = F_j(\mathbf{x})$. □

Proposition 2 *Suppose the Pareto exists and is of dimension* $N-1$ *and there exists a point in the Pareto such that* $F_k = F_k(\mathbf{x}_0)$ *for all* $1 \le k \le N$*'s but one, then the algorithm is feasible and it converges to the Pareto Set in at most N trials.*

Proof In order to show convergence, it is sufficient to apply the argument used in the earlier proof for all $1 \le p \le N$. □

Remark 2 On the Propositions

- The nature of the hypothesis for the two propositions is shown in Fig. 7. In the convex case, Fig. 7a, the projection of the starting point onto the Pareto set is guaranteed to occur. Convexity is not necessary as shown in Fig. 7b. In the 2 objective functions case, in order to achieve convergence in one iteration, it is sufficient that there exist two points in the Pareto set such that $F_1(x'_{Pareto}) = F_1(\mathbf{x}_0)$ and $F_2(\mathbf{x}''_{Pareto}) = F_2(\mathbf{x}_0)$. A similar argument can be used in the case that the Pareto set is not connected (Fig. 7c). In this case, the algorithm will take at most 2 trials to converge.
- *Realization of equality constraints* We observe that enforcing equality constraints is costly, especially in terms of shared information. Throughout the article, we compared the exact algorithm with the approximation of the shared objective function. In particular, we took that $F_k(\mathbf{x}) - \bar{F}_k \approx \text{approx}(F_k) - \bar{F}_k$, where approx indicates an approximation operator. We note that $F_k(\mathbf{x}) = \text{approx}(F_k) + \text{error approx.}$, where the error can be quantified.
- *Estimates of the approximation errors* In the article, we utilized a polynomial approximation operator. For this type of approximation strategy, we can provide the error estimate on the basis of the generalized mean value theorem. In particular, if we assume that $F_k \in$

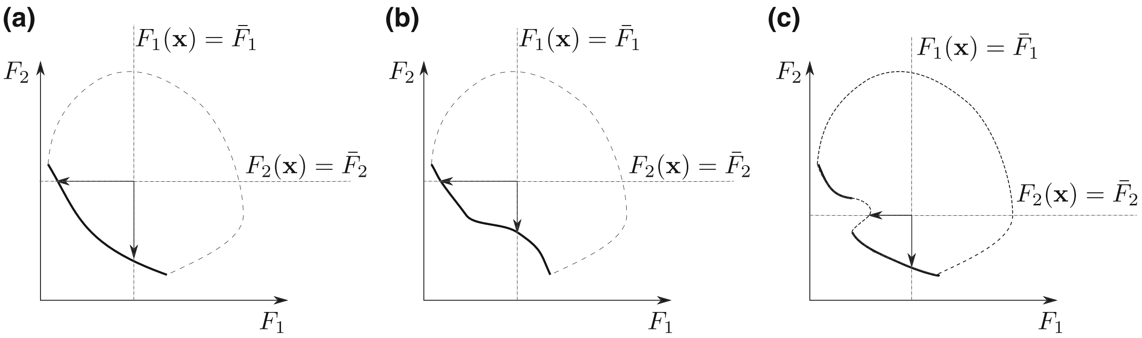
(a)
$F_1(\mathbf{x}) = \bar{F}_1$
F_2
$F_2(\mathbf{x}) = \bar{F}_2$
F_1
(b)
$F_1(\mathbf{x}) = \bar{F}_1$
F_2
$F_2(\mathbf{x}) = \bar{F}_2$
F_1
(c)
$F_1(\mathbf{x}) = \bar{F}_1$
F_2
$F_2(\mathbf{x}) = \bar{F}_2$
F_1

Fig. 7 Visual explanation of Propositions 1 and 2

C^{m+1} then the error at a point $\mathbf{x}$ due to truncation at order m will be at most $\frac{1}{(m+1)!}\left[\sup_{\mathbf{x}\in\mathcal{U}} F_k^{(m+1)}(x)\right]$ $(\mathbf{x}-\mathbf{x}_0)^{(m+1)}$.

- *Relaxation of shared constraints* As shown in the previous remark, the shared constraint can be approximated numerically by means of polynomials. However, it is clear that feasibility of the design cannot always be guaranteed unless the approximation error is small.

References

Agte J, de Weck O, Sobieski J, Arendson P, Morris A, Spieck M (2010) MDO: assessment and direction for advancement—an opinion of one international group. Struct Multidiscip Optim 40(1–6):17–33

Allen TJ (1984) Managing the flow of technology: technology transfer and the dissemination of technological information. MIT, Cambridge

Allision JT, Kokkolara M, Papalambros PY (2009) Optimal partitioning and coordination decisions in decomposition-based design optimization. J Mech Des 1(8):081008–1–081008–8

Avnet MS (2009) Socio-cognitive analysis of engineering systems design: shared knowledge, process, and product. PhD thesis, Massachusetts Institute of Technology

Azarm S, Tits A, Fan MKH (1999) Tradeoff driven optimization-based design of mechanical systems. In: 4-th AIAA/USAF/NASA/OAI symposium on multidisciplinary analysis and optimization, Cleveland, Ohio, USA, AIAA. AIAA-92-4758-CP

Chanron V, Lewis K (2004) Convergence and stability in distributed design of large systems. In: ASME design automation conference, Salt Lake City, Utah

Chanron V, Lewis K (2005) A study of convergence in decentralized design processes. Res Eng Design 16(3):133–145

Chanron V, Lewis K, Murase Y, Izui K, Nishiwaki S, Yoshimura M (2005) Handling multiple objectives in decentralized design. In: ASME design automation conference, Long Beach, CA

Chanron V, Singh T, Lewis K (2005) Equilibrium stability in decentralized design systems. Int J Syst Sci 36(10):651–662

Chinchuluun A, Pardalos P (2007) A survey of recent developments in multiobjective optimization. Ann Oper Res 154(1):29–50

Cramer EJ, Dennis JE, Frank PD, Lewis RM, Shubin GR (1993) Problem formulation for multidisciplinary optimization. Technical report, Center for Research on Parallel Computation, Rice University

Cross N (2000) Engineering design methods: strategies for product design. Wiley, New York

Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. Struct Multidiscip Optim 14(1):63–69

Dauer JP, Stadler W (1986) A survey of vector optimization in infinite-dimensional spaces, part 2. J Optim Theory Appl 51(2):205–241

Ding XP, Park JY, Jung IH (2000) Existence of pareto equilibria for constrained multiobjective games in h-space. Comput Math Appl 39(9–10):125–134

Ding XP, Park JY, Jung IH (2003) Pareto equilibria for constrained multiobjective games in locally l-convex spaces. Comput Math Appl 46(10–11):1589–1599

Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1:211–218

Fernandez MG, Panchal JH, Allen JK, Mistree F (2005) An interactions protocol for collaborative decision making—concise interactions and effective management of shared design spaces. In: ASME design engineering technical conferences, Long Beach, CA

Fliege J, Svaiter BF (2000) Steepest descent methods for multicriteria optimization. Math Methods Oper Res 51(3):479–494

Franssen M, Bucciarelli LL (2004) On rationality in engineering design. J Mech Des 126(6):945–949

GarcÌa-Palomares UM, Burguillo-Rial JC, Gonzalez-Castano FJ (2008) Explicit gradient information in multiobjective optimization. Oper Res Lett 36(6):722–725

Golinski J (1970) Optimal synthesis problems solved by means of nonlinear programming and random methods. J Mech 5

Haimes Y (1973) Integrated system identification and optimization. Control Dyn Syst Adv Theory Appl 10:435–518

Hazelrigg GA (1998) A framework for decision-based engineering design. J Mech Des 120:653–658

Homma T, Saltelli A (1996) Importance measures in global sensitivity analysis of nonlinear models. Reliab Eng Syst Saf 52:1–17

Honda T, Ciucci F, Yang MC (August 2007) Achieving pareto optimality in a decentralized design environment. In: 17th international conference on engineering design (ICED). The Design Society

Huang CH, Galuski J, Bloebaum CL (2007) Multi-objective pareto concurrent subspace optimization for multidisciplinary design. AIAA J 45(8):1894–1906

Jet Propulsion Laboratory (2010) Team X. ([http://jplteamx.jpl.nasa.gov/])

Jian JB, Ju QJ, Tang CM, Zheng HY (2007) A sequential quadratically constrained quadratic programming method of feasible directions. Appl Math Optim 56(56):343–363

Keeney RL (2009) The foundations of collaborative group decisions. Int J Collab Eng 1(1–2):4–18

Kurpati A, Azarm S, Wu J (2002) Constraint handling improvements for multiobjective genetic algorithms. Struct Multidiscip Optim 23(3):204–213

Laumanns M, Thiele L, Zitzler E (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. Eur J Oper Res 169(3):932–942

Lewis K (1996) An algorithm for integrated subsystem embodiment and system synthesis. PhD thesis, Georgia Institute of Technology

Lewis K, Mistree F (1997) Modeling the interactions in multidisciplinary design: a game theoretic approach. AIAA J Aircraft 35:1387–1397

Lewis K, Mistree F (1999) Collaborative, sequential, and isolated decisions in design. ASME J Mech Des 120:643–652

Lewis K, Mistree F (2001) Modeling subsystem interactions: a game theoretic approach. J Des Manuf Autom 1:17–36

Lewis, KE, Chen, W, Schmidt, LC (eds) (2006) Decision making in engineering design. American Society of Mechanical Engineers, New York

Li M, Azarm S (2007) Multiobjective collaborative robust optimization (mcro) with interval uncertainty and interdisciplinary uncertainty propagation. In: Proceedings of the ASME 2007 international design engineering technical conferences & computers and information in engineering conference IDETC/CIE 2007, Las Vegas, Nevada, USA, ASME. DETC2007-34818

Lin JI (1976) Multiple-objective problems: pareto-optimal solutions by method of proper equality constrains. IEEE Trans Autom Control 21(5):641–650

Lu S, Kim HM (2010) Optimized sequencing of analysis components in multidisciplinary systems. J Mech Des 132(4):041005-1–041005-12

Marglin S (1967) Public investment criteria. MIT, Cambridge

Mark G (2002) Extreme collaboration. Commun ACM 45(6):89–93

Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26(6):369–395

Mistree F, Marinopoulos S, Jackson DM, Shupe JA (1988) The design of aircraft using the decision support problem technique. Technical report NAS 1.26:4134, NASA-CR-4134, NASA

Olesen K, Myers MD (1999) Trying to improve communication and collaboration with information technology. an action research project which failed. Inf Technol People 12(4):317–332

Park GJ (2007) Analytic methods in design practice. Springer, Berlin

Rao JRJ, Badhrinath K, Pakala R, Mistree F (1997) A study of optimal design under conflict using models of multi-player games. Eng Optim 28:63–94

RV Tappeta, JE Renaud (1997) Multiobjective collaborative optimization. J Mech Des 119:403–411

Scott MJ (1999) Formalizing negotiation in engineering design. PhD thesis, California Institute of Technology, Pasadena, CA

Scott MJ, Antonsson EK (2000) Arrow's theorem and engineering design decision making. Res Eng Design 11(4):218–228

Senge PM (1990) The fifth discipline: The art & practice of the learning organization. Crown Business, New York

Shaja AS, Sudhakar K (2010) Optimized sequencing of analysis components in multidisciplinary systems. Res Eng Design 21(3):173–187

Shin MK, Park GJ (2005) Multidisciplinary design optimization based on independent subspaces. Int J Numer Methods Eng 64:599–617

Simaan M, Cruz JB (1973) On the stackelberg strategy in nonzero-sum games. J Optim Theory Appl 11(5):533–555

Smith J, Koenig L, Wall SD (1999) Team efficiencies within a model-driven design process. In: INCOSE symposium, Brighton, England

Sobieszczanski-Sobieski J (1988) Optimization by decomposition: A step from hierarchic to non-hierarchic systems. Technical report, NASA

Sobieszczanski-Sobieski J, Agte JS, Sandusky RR (1998) Bi-level integrated system synthesis (bliss). In: AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, volume AIAA-98-4916. AIAA, pp 1543–1557

Stewart GW (1993) On the early history of the singular value decomposition. SIAM Rev 35:551–566

Tao YR, Han X, Jiang C, Guan FJ (2010) A method to improve computational efficiency for csso and bliss. Struct Multidiscip Optim. Nov.(online):1–6

Tosserams S, Etman LFP, Rooda JE (2010) A micro-accelerometer mdo benchmark problem. Struct Multidiscip Optim 41(2): 255–275

Verbeeck K, Nowe A, Lenaerts T, Parent J (2002) Learning to reach the pareto optimal nash equilibrium as a team. Adv Artif Intell 2557:407–418

Vincent TL (1983) Game theory as a design tool. J Mech Transm Autom Des 105:165–170

Wang F, Zhang K (2008) A hybrid algorithm for nonlinear minimax problems. Ann Oper Res 164(1):167–191

Ward AC (1989) A theory of quantitative inference for artifact sets, applied to a mechanical design compiler. PhD thesis, MIT

Ward AC, Lozano-Pérez T, Seering WP (1990) Extending the constraint propagation of intervals. Artif Intell Eng Des Manuf 4(1):47–54

de Weck O (2004) Multiobjective optimization: History and promise. In: The third China-Japan-Korea joint symposium on optimization of structural and mechanical systems, Kanazawa, Japan, 2004. Invited Keynote Paper, GL2-2

Whitfield RI, Duffy AHB, Coates G, Hills B (2002) Distributed design coordination. Res Eng Design 13:243–252

Xiao A, Zheng S, Allen JK, Rosen DW, Mistree F (2005) Collaborative multidisciplinary decision making using game theory and design capability indices. Res Eng Design 16(1–2):57–72

Yi SI, Shin JK, Park GJ (2008) Comparison of mdo methods with mathematical examples. Struct Multidiscip Optim 35:391–402

Yu H (2003) Weak pareto equilibria for multiobjective constrained games. Appl Math Lett 16(5):773–776

Zhao M, Cui W (2011) On the development of bi-level integrated system collaborative optimization. Struct Multidiscip Optim 43(1):73–84