6-2023

# Gradient-Based Trade-Off Design for Engineering Applications

Lena A. Royster

Gene Hou

*designs*

*Article*

# Gradient-Based Trade-Off Design for Engineering Applications

**Lena A. Royster and Gene Hou ***

Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, VA 23529, USA; lena.a.royster@uscg.mil
* Correspondence: ghou@odu.edu

**Abstract:** The goal of the trade-off design method presented in this study is to achieve newly targeted performance requirements by modifying the current values of the design variables. The trade-off design problem is formulated in the framework of Sequential Quadratic Programming. The method is computationally efficient as it is gradient-based, which, however, requires the performance functions to be differentiable. A new equation to calculate the scale factor to control the size of the design variables is introduced in this study, which can ensure the new design achieves the targeted performance objective. Three formal approaches are developed in this study for trade-off design to handle various design scenarios, which include one that can handle cases with linearly dependent constraints and with more constraints than the number of design variables. Three engineering design problems are presented as examples to validate and demonstrate the use of these trade-off approaches to find the best way to adjust the design variables to meet the revised performance requirements.

**Keywords:** gradient-based; trade-off design; sequential quadratic programming; scalar factor and search directions

check for updates

**Citation:** Royster, L.A.; Hou, G. Gradient-Based Trade-Off Design for Engineering Applications. *Designs* **2023**, *7*, 81. https://doi.org/10.3390/designs7040081

Academic Editor: Robert Ian Whitfield

Received: 25 May 2023
Revised: 16 June 2023
Accepted: 23 June 2023
Published: 24 June 2023



## 1. Introduction

Engineering design is an iterative process. The initial design usually reveals insufficiency in the existing design problem formulation, which includes design variables, options, objectives, and constraints based on the demands and limits on the resources. A new set of design variables, design performances, and requirements may be considered and taken for comparison with existing ones through trade-offs. This may lead to the revision of the existing problem formulation for the next phase of the design study. This review and revision process of an existing design problem is called the trade-off design in this study. Particularly, a trade-off can be defined as a balance or compromise between two or more desirable but incompatible features. There has been some progress made in the development of proper design optimization formulations for trade-off design. Multi-objective design optimization is a common choice for trade-off design as it produces Pareto fronts, which are the collection of the most favorable compromised designs. For example, multi-objective design optimization was used by Tan et al. [1] for trade-offs between multiple design performance measurements. The objectives are the design performance requirements, while the constraints are the limits on component performance. The paper indicated that the optimal design under tight constraint limits might fail due to modeling errors and operational uncertainties. Re-design was conducted after the multi-objective design optimization finding a feasible design among the Pareto fronts that can meet the newly targeted performance requirements and relaxed constraint limits. Otto and Antonsson [2] proposed two different design optimization formulations to produce the most preferable design by trading off different design goals. In their study, the objective was a vector of preference ratings associated with each individual design variable and design performance function. One approach, called the conservative design, produces the overall preference by maximizing the design option with a minimal preference rating. The other, called the aggressive design, identifies the optimal design that maximizes the geometric mean of the vector of preference ratings.

Most of the applications of trade-offs are concerned with the performance outcome of a design. Particularly, Silveria and Slack [3] suggested measuring the significance of trade-offs by their levels of importance and sensitivity of their impact on decision-making. Nassar and Austin [4] addressed the importance of constructing consistent evaluation criteria to measure the importance of every performance requirement in a multi-objective formulation for trade-offs, regardless of whether it is a system-level option or a component-level one. The trade-off considered by Bae et al. in [5] is the impact of design variable selection on the final optimal solutions. In their design problem formulation, the probabilities of failures of the components and the system were included as constraints. A Monte Carlo simulation was employed in their study to find the probabilities of failures of components in the system. The objective was formulated as a linear combination of terms associated with the performance requirement of each component and the number of samples used for reliability analysis. The number of samples was included in the objective, as they affected the accuracy of variance estimation in the Monte Carlo simulation. In addition to using the weighting coefficients to combine all performance alternatives together to form a single performance aggregation function, Wang and Terpenny [6] applied fuzzy set theory to count the uncertainty and added penalty coefficients to enforce the constraints that are imposed upon each of the performance alternatives. Furthermore, a power index was applied to all performance alternatives to represent the level of compensation. The goal of the trade-off in [6] was to match the fuzzy-based performance aggregation function to the targeted design output by adjusting the weighting coefficients and the compensation factor based on a root-mean-square minimization strategy. The targeted design output was determined by the designer's decision, which can be quickly updated. Rojas et al. [7] set up a utility function to measure the performance values of all technical components required to build a wireless power transfer device and applied the analytic hierarchy process (AHP) to determine the weighting factors of criteria for selection. The trade-off analysis used the weighted sum of these utility functions to select the proper technical components that could be assembled to produce the most suitable implanted medical device.

The common goal of the trade-off design mentioned in the most publication cited above was to select the most suitable design among the existing design alternatives by trading off different performance requirements. However, the goal of the current research is to trade off the current values of the selected design variables to achieve newly targeted performance function values. Particularly, the current trade-off design aims to modify the existing values of the selected design variables to minimize the difference between the targeted and the current values of the performance functions, regardless of whether they are assigned as objectives or constraints. To this end, a gradient-based method for trade-off design is developed in this study to investigate the impact of changes in the design requirements on the values of the current design variables. This trade-off design method is formulated based on the framework of Sequential Quadratic Programming (SQP) [8–12] to find the minimal changes required in the design variables to achieve the targeted changes to all specified performance requirements. The proposed method is computationally efficient, as it is gradient-based. Consequently, it requires that all performance functions are differentiable with respect to the selected design variables.

The rest of the paper is organized into three sections. Three different optimization formulations will be presented in Section 2 for trade-offs in different design scenarios. All performance measurements and limits are treated as equally important in the formulations. The output of these formulations is the required changes in the selected design variables to achieve the targeted values of the desirable performances and limits. Three examples are presented in Section 3 to validate the proposed formulations and demonstrate their applications for trade-offs. The first example is the design of a cubic box, the second is to adjust the stiffness matrix of a vibration problem to achieve three targeted frequencies, and the last is an I-beam design problem. The weight, the deflection, the bending stress, and the frequencies of the I-beam are considered as performance requirements. Concluding remarks are provided in the final section.

## 2. Materials and Methods

Section 2 is made of three subsections, each of which investigates one mathematical formulation to handle a specific trade-off scenario. All trade-off designs are formulated here in the framework of the SQP but with different objectives and constraints in different subsections. Section 2.1, named the Single Objective Approach (SOA), formulates the trade-off problem by selecting only one performance function as the objective and the rest as the constraints. Section 2.2, named the Constraint Only Approach (COA), counts all performance functions equally as the constraints in the trade-off formulation. Both formulations produce identical results, and both require the gradients of all involved functions to be linearly independent of each other. Section 2.3, named the Multiple Objective Approach (MOA), introduces a new formulation to resolve the problem with the linearly dependent gradients as well as the over-constrained problem. The latter is the case when the number of the performance function requirements is greater than that of the design variables. Different equations are proposed in different subsections to compute scalar factors to support the specific needs of trade-off designs. The scalar factor is an essential element in the trade-off design investigated here, which enables a search direction to accurately achieve the targeted change in the objective function.

Note that in all notations, matrices are denoted by upper case letters, vectors by bold letters, and scalars by lower case letters.

### *2.1. Single Objective Approach (SOA)*

The goal of the trade-off design is to find the change of the design variables, $\boldsymbol{s}$, so that the revised design $\boldsymbol{x}$, $\boldsymbol{x} = \boldsymbol{x_0} + \boldsymbol{s}$, about the current design, $\boldsymbol{x_0}$, can achieve the targeted changes in the objective and constraint functions. Only one objective is considered in this Single Objective Approach (SOA). The trade-off design is formulated in the framework of SQP. Traditionally, the SQP plays a key role in supporting direct search design optimization [8–13]. It aims to find the least change in the design variables that can reduce the objective function, $f(\boldsymbol{x_0})$, and in the meantime, achieve the required corrections in the current values of the inequality constraints, $\boldsymbol{g}(\boldsymbol{x_0})$, and the equality ones, $\boldsymbol{h}(\boldsymbol{x_0})$. Its formulation can be stated below.

$$\min_{\boldsymbol{s} \in R^n} (\nabla f(\boldsymbol{x_0}))^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T W \boldsymbol{s} \tag{1}$$

Subject to:

$$(\nabla \boldsymbol{g}(\boldsymbol{x_0}))^T \boldsymbol{s} + \widetilde{\boldsymbol{g}} \leq 0$$

$$(\nabla \boldsymbol{h}(\boldsymbol{x_0}))^T \boldsymbol{s} + \widetilde{\boldsymbol{h}} = 0$$

where $\widetilde{\boldsymbol{g}}$ and $\widetilde{\boldsymbol{h}}$ are the gaps between the targeted and the current function values, defined as $\widetilde{\boldsymbol{g}} \equiv \boldsymbol{g}(\boldsymbol{x_0}) - \boldsymbol{g_T}$ and $\widetilde{\boldsymbol{h}} \equiv \boldsymbol{h}(\boldsymbol{x_0}) - \boldsymbol{h_T}$, where $\boldsymbol{g_T}$ and $\boldsymbol{h_T}$ represent the targeted values of the performance requirements. The optimal solution, $\boldsymbol{s}$, of Equation (1) is called the search direction, which represents the most effective way to reduce the current objective and correct the current constraint violations. A quadratic term of $\boldsymbol{s}$ is added to the objective to control the size of $\boldsymbol{s}$ to ensure the validity of the first-order approximation. The diagonal matrix $W$ with positive diagonal terms is added to scale the design variables. The gradients in the above formulation, $\nabla f$, $\nabla \boldsymbol{g}$ and $\nabla \boldsymbol{h}$, are also evaluated at the current design variables, $\boldsymbol{x_0}$. The row numbers of all gradients are equal to the number of design variables, while the column numbers of $\nabla \boldsymbol{g}$ and $\nabla h$ are equal to the respective numbers of the inequality and equality constraints. Since only the violated or active inequality constraints, which yield $\boldsymbol{g}_i(x_0) + \varepsilon > 0$, are considered in the solution process, the above problem can be recast as one with equality constraints as stated below.

$$\min_{\boldsymbol{s} \in R^n} (\nabla f)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T W \boldsymbol{s}$$

Subject to:

$$(\nabla \boldsymbol{g})^T \boldsymbol{s} + \widetilde{\boldsymbol{g}} = 0$$

$$(\nabla \boldsymbol{h})^T \boldsymbol{s} + \widetilde{\boldsymbol{h}} = 0$$

where the number of columns of $\nabla \boldsymbol{g}$ is now equal to the number of active constraints. Note that the total number of constraints must not be greater than the number of the design variables. Furthermore, it is assumed that the gradients of the constraints are linearly independent of each other. Since the inequality and equality constraints are all in the same form in the formulation, only one is kept in the following derivation for simplicity. Thus, the problem is simplified as

$$\min_{\boldsymbol{s} \in R^n} (\nabla f)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T W \boldsymbol{s} \tag{2}$$

subject to:

$$(\nabla \boldsymbol{g})^T \boldsymbol{s} + \widetilde{\boldsymbol{g}} = 0$$

The Lagrange function of this problem is stated as

$$L = (\nabla f)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T W \boldsymbol{s} + \boldsymbol{\lambda}^T \left( (\nabla \boldsymbol{g})^T \boldsymbol{s} + \widetilde{\boldsymbol{g}} \right)$$

The Kuhn–Tucker necessary condition yields the following equation,

$$\nabla L = \nabla f + (\nabla \boldsymbol{g})\boldsymbol{\lambda} + W\boldsymbol{s} = 0$$

which can be solved to obtain the optimal solution of Equation (2) in terms of the Lagrange multipliers, $\boldsymbol{\lambda}$, as

$$\boldsymbol{s} = -W^{-1}\nabla f - W^{-1}(\nabla \boldsymbol{g})\boldsymbol{\lambda} \tag{3}$$

Pre-multiplying $(\nabla \boldsymbol{g})^T$ to Equation (3), one has

$$(\nabla \boldsymbol{g})^T \boldsymbol{s} = -(\nabla \boldsymbol{g})^T W^{-1}\nabla f - (\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g})\boldsymbol{\lambda}$$

Note that $(\nabla \boldsymbol{g})^T \boldsymbol{s} = -\widetilde{\boldsymbol{g}}$ as required by the equality constraint statement in Equation (2), the above equation becomes,

$$\widetilde{\boldsymbol{g}} = (\nabla \boldsymbol{g})^T W^{-1}\nabla f + (\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g})\boldsymbol{\lambda}$$

which yields an equation of $\boldsymbol{\lambda}$ as

$$(\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g})\boldsymbol{\lambda} = -(\nabla \boldsymbol{g})^t W^{-1}\nabla f + \widetilde{\boldsymbol{g}} \tag{4}$$

Under the assumption that the columns of $\nabla \boldsymbol{g}$ are linearly independent of each other, Equation (4) can be solved uniquely for $\boldsymbol{\lambda}$ as

$$\boldsymbol{\lambda} = -\left\{ \left[ (\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g}) \right]^{-\mathbf{1}} (\nabla \boldsymbol{g})^t W^{-1} \right\} \nabla f + \left[ (\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g}) \right]^{-\mathbf{1}} \widetilde{\boldsymbol{g}}$$

Substituting $\boldsymbol{\lambda}$ back to Equation (3) results in the search direction, $\boldsymbol{s}$, expressed in terms of $\nabla f$ and $\nabla \boldsymbol{g}$ as

$$\boldsymbol{s} = -W^{-1}\left[ I - (\nabla \boldsymbol{g})\left( (\nabla \boldsymbol{g}) W^{-1} (\nabla \boldsymbol{g})^T \right)^{-\mathbf{1}} (\nabla \boldsymbol{g})^T W^{-1} \right] \nabla f - W^{-1}(\nabla \boldsymbol{g})\left[ (\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g}) \right]^{-\mathbf{1}} \widetilde{\boldsymbol{g}}$$

Set the matrices $P$ and $Q$ as

$$P \equiv I - (\nabla \boldsymbol{g})\left[ (\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g}) \right]^{-1} (\nabla \boldsymbol{g})^T W^{-1} \tag{5}$$

and

$$Q \equiv W^{-1}(\nabla \boldsymbol{g})\left[(\nabla \boldsymbol{g})^T W^{-1}(\nabla \boldsymbol{g})\right]^{-1} \tag{6}$$

The search direction expression derived above can now be simplified as

$$\boldsymbol{s} = -W^{-1}P\,\nabla f - Q\widetilde{\boldsymbol{g}} \tag{7}$$

It allows the search direction to be separated into two parts,

$$\boldsymbol{s} = -\boldsymbol{s}_1 + \boldsymbol{s}_2$$

where $\boldsymbol{s}_1$ is related to the objective reduction, $\nabla f$, while $\boldsymbol{s}_2$ corrects the constraint violation,

$$\boldsymbol{s}_1 \equiv W^{-1}P\,\nabla f \tag{8}$$

$$\boldsymbol{s}_2 \equiv -Q\widetilde{\boldsymbol{g}} \tag{9}$$

The matrix $P$ is called the projection matrix, as it can be proved that $PP = P$. Moreover, it can be shown that $P^T Q = 0$, which implies that $\boldsymbol{s}_1$ is orthogonal to $\boldsymbol{s}_2$ with respect to the weighting matrix, $W$. That is,

$$\boldsymbol{s}_1^T W \boldsymbol{s}_2 = \left(W^{-1}P\,\nabla f\right)^T W Q\widetilde{\boldsymbol{g}} = (\nabla f)^T\left(P^T Q\right)\widetilde{\boldsymbol{g}} = \boldsymbol{0}$$

It can also prove the following relations,

$$-(\nabla f)^T \boldsymbol{s}_1 \leq 0$$

$$(\nabla \boldsymbol{g})^T \boldsymbol{s}_1 = 0$$

$$(\nabla \boldsymbol{g})^T \boldsymbol{s}_2 = -\widetilde{\boldsymbol{g}}$$

These relations indicate that $-\boldsymbol{s}_1$ is the part of the search direction that can reduce the objective function without changing the values of the constraints. On the other hand, $\boldsymbol{s}_2$ is the only part of the search direction that is responsible to reduce the constraint violations. However, $\boldsymbol{s}_2$ may affect the value of the objective function as

$$(\nabla f)^T \boldsymbol{s}_2 \neq 0$$

A scalar factor $\alpha$ is introduced in the search direction to produce a search direction that can achieve the targeted change in the objective function. That is,

$$\boldsymbol{s} = -\alpha \boldsymbol{s}_1 + \boldsymbol{s}_2 \tag{10}$$

Assume that the desired correction, $\Delta f$, in the objective is represented by the difference between the current objective, $f_C = f(\boldsymbol{x}_0)$, and the targeted objective, $f_T$; i.e., $\Delta f \equiv f_C - f_T$. It is expected that the improved design, $\boldsymbol{x}_0 + \boldsymbol{s}$, can meet the targeted objective. Thus, the goal is to achieve $f(\boldsymbol{x}_0 + \boldsymbol{s}) = f_T$. The desired correction, $\Delta f$, can then be approximated by the first-order expansion as

$$0 = f(\boldsymbol{x}_0 + \boldsymbol{s}) - f_T = (\nabla f(\boldsymbol{x}_0))^T \boldsymbol{s} + f(\boldsymbol{x}_0) - f_T = (\nabla f)^T(-\alpha \boldsymbol{s}_1 + \boldsymbol{s}_2) + \Delta f \tag{11}$$

Solving for $\alpha$ yields

$$\alpha = \frac{\Delta f + \nabla f^T \boldsymbol{s}_2}{(\nabla f^T \boldsymbol{s}_1)} \tag{12}$$

Note that the equation of the scalar factor is valid only if the gradient of the objective, $\nabla f$, is linearly independent of the columns of $\nabla \boldsymbol{g}$. Otherwise, $\nabla f^T \boldsymbol{s_1} = 0$. Its proof is presented in Appendix A.

The SQP formulations like the one reported in Equation (2) have been used by many optimization algorithms recursively to find the search direction $\boldsymbol{s}$ that can locate an improved design about the current design. The uniqueness of the current approach is the use of the scalar factor. The scalar factor is called the step size, which is selected to reduce the objective in the commonly used optimization algorithms rather than to achieve the required changes in the objective in this study. In some published work [8,11], the scalar factor, $\alpha$, is calculated to reduce the objective function alone as

$$\alpha = \frac{\Delta f}{(\nabla f^T \boldsymbol{s_1})}$$

The others set $\alpha$ up to adjust the entire search direction as

$$\boldsymbol{s} = \alpha(-\boldsymbol{s}_1 + \boldsymbol{s}_2)$$

In this case, the $\alpha$ is sought to minimize a user-defined merit function which is the weighted combination of the objective function and the maximal violation [8–10].

Two special applications of the SOA are discussed hereafter that can handle certain circumstances more effectively. In the first case, the goal is to find the search direction that can reduce the objective by a certain amount without changing the constraints. In this case, the constraints remain unchanged, $\widetilde{\boldsymbol{g}} = 0$, $\boldsymbol{s}_2 = 0$. Consequently, the search direction becomes.

$$\boldsymbol{s} = -\alpha \boldsymbol{s}_1 = -\left(\frac{\Delta f}{(\nabla f^T \boldsymbol{s_1})}\right) \boldsymbol{s}_1$$

In the second case, the goal is to correct the violation without changing the objective function. In this case, $\Delta f = 0$ and the search direction defined by Equation (10) is revised as

$$\boldsymbol{s} = -\left(\frac{\nabla f^T \boldsymbol{s_2}}{\nabla f^T \boldsymbol{s_1}}\right) \boldsymbol{s}_1 + \boldsymbol{s}_2$$

*2.2. Constraint-Only Approach (COA)*

The objection function in this approach is treated as part of the constraint set with the desired amount of reduction, $\Delta f$. Thus, the constraint set is expanded to include $f$ as

$$\nabla \boldsymbol{g} = \begin{pmatrix} \nabla f & \nabla \overline{\boldsymbol{g}} \end{pmatrix} \text{ and } \widetilde{\boldsymbol{g}} = \begin{Bmatrix} \Delta f \\ \overline{\boldsymbol{g}} \end{Bmatrix} \tag{13}$$

where $\overline{\boldsymbol{g}}$ represents the initial constraint set. The amount, $\Delta f$, presented in Equation (13) is the same as that described in Equation (11). Particularly, $\Delta f$ is the adjustment of the current objective to meet the goal,

$$\Delta f = f_C - f_T$$

Note that in this case, the total number of constraints must be less than the number of design variables. Additionally, gradients of all functions involved, including $f$ and $\overline{\boldsymbol{g}}$, must be linearly independent of each other. Once $f$ is removed from Equation (2), the sequential quadratic programming problem becomes

$$\min_{s \,\in\, R^n} \frac{1}{2} \boldsymbol{s}^T W \boldsymbol{s} \tag{14}$$

subject to:

$$(\nabla \boldsymbol{g})^T \boldsymbol{s} + \widetilde{\boldsymbol{g}} = 0$$

where the change of the objective, $\Delta f$, is now part of $\tilde{g}$. Equation (14) is referred to as the constraint correction algorithm in optimization [12], which also has been used for correcting kinematic constraints in multibody dynamics [14–16].

The solution of Equation (14) is now found to be:

$$s = s_2 = -Q\tilde{g} = -W^{-1}\nabla g\left[(\nabla g)^T W^{-1}(\nabla g)\right]^{-1}\tilde{g} \tag{15}$$

Set the vector $p$ to be the solution of the following matrix equation,

$$\left[(\nabla g)^T W^{-1}(\nabla g)\right]p = \tilde{g} \tag{16}$$

The search direction, $s$, in Equation (15) can be recast as

$$s = s_2 = -Q\tilde{g} = -W^{-1}(\nabla g)p$$

To investigate the relation between the search directions described by Equation (15) and Equation (10) of the SOA, one may expand Equation (16) in detail. This can be done by decomposing Equation (16) into two parts to separate the objective reduction, $\Delta f$ from the rest of the constraint correction, $\overline{g}$ as

$$\left[\begin{Bmatrix} \nabla f^T \\ \nabla \overline{g}^T \end{Bmatrix} W^{-1}\begin{pmatrix} \nabla f & \nabla \overline{g} \end{pmatrix}\right]\begin{Bmatrix} p_f \\ p_{\overline{g}} \end{Bmatrix} = \begin{Bmatrix} \Delta f \\ \overline{g} \end{Bmatrix}$$

or more specifically,

$$\begin{bmatrix} \nabla f^T W^{-1}\nabla f & \nabla f^T W^{-1}\nabla \overline{g} \\ \nabla \overline{g}^T W^{-1}\nabla f & \nabla^T \overline{g} W^{-1}\nabla \overline{g} \end{bmatrix}\begin{Bmatrix} p_f \\ p_{\overline{g}} \end{Bmatrix} = \begin{Bmatrix} \Delta f \\ \overline{g} \end{Bmatrix}$$

The above matrix equation can then be solved separately in two steps. The second row of the above equation yields the solution as

$$\overline{p}_{\overline{g}} = -\left(\nabla^T \overline{g} W^{-1}\nabla \overline{g}\right)^{-1}\left(\nabla \overline{g}^T W^{-1}\nabla f\right)p_f + \left(\nabla^T \overline{g} W^{-1}\nabla \overline{g}\right)^{-1}\overline{g}$$

while the first row produces the solution,

$$p_f = -\left(\frac{\Delta f + \nabla f^T \overline{s}_2}{\nabla f^T \overline{s}_1}\right)s_1 + s_2$$

Consequently, the search direction obtained from Equation (15) can be shown to be,

$$\begin{aligned} s = s_2 = -Q\tilde{g} = W^{-1}(\nabla g)\left[(\nabla g)^T W^{-1}(\nabla g)\right]^{-1}\tilde{g} = -W^{-1}(\nabla g)p \\ = \left[-W^{-1}\overline{P}\nabla f p_f - \overline{Q}\overline{g}\right] = -p_f\overline{s}_1 + \overline{s}_2 \end{aligned} \tag{17}$$

where $\overline{P}$, $\overline{Q}$, $\overline{s}_1$, and $\overline{s}_2$ are the same as those defined by Equations (5)–(9) in terms of the original constraint set, which does not include the objective function, $f$. Also, note that $p_f$ is identical to the scalar factor, $\alpha$, derived earlier in Equation (12). Equation (17) states that the search direction, $s$, in the COA, is the same as Equation (10) derived for the SOA. The COA provides an alternative to formulate and solve the trade-off problem defined by Equation (2). Once the amount of the reduction in the objective is known and included as part of constraint correction, the design change can be obtained alone by $s_2$ of Equation (17) in terms of the newly defined $\tilde{g}$ in Equation (13). The detailed proof of Equation (17) can be found in Appendix A.

*2.3. Multiple-Objective Approach (MOA)*

The formulation presented by either Equation (2) or (14) for trade-off design cannot handle the cases when the number of functions is greater than the number of the design variables nor some of the functions whose gradients are linearly dependent on each other. In this case, one may set up an objective function that involves those linearly dependent or over-constrained functions as

$$\eta(\boldsymbol{s}) \equiv \frac{1}{2}\left((\nabla \boldsymbol{f})^T \boldsymbol{s} + \Delta \boldsymbol{f}\right)^T \left((\nabla \boldsymbol{f})^T \boldsymbol{s} + \Delta \boldsymbol{f}\right) \tag{18}$$

The scalar objective function, $\eta(\boldsymbol{s})$, in Equation (18) represents the magnitude of adjustment of the objective function vector, $\boldsymbol{f}$ as,

$$\boldsymbol{f}(\boldsymbol{x_0} + \boldsymbol{s}\ ) - \boldsymbol{f}_T = (\nabla \boldsymbol{f})^T \boldsymbol{s} + \boldsymbol{f}(\boldsymbol{x_0}) - \boldsymbol{f}_T = (\nabla \boldsymbol{f})^T \boldsymbol{s} + \Delta \boldsymbol{f}$$

Thus, the intention of Equation (18) is to measure the gap between the revised design, $\boldsymbol{f}(\boldsymbol{x_0} + \boldsymbol{s}\ )$ and the targeted $\boldsymbol{f_T}$. The gradient of $\eta(\boldsymbol{s})$ is found to be,

$$\nabla \eta = -\nabla \boldsymbol{f} \Delta \boldsymbol{f} \tag{19}$$

Note that the desirable changes in functions are now defined in a vector form,

$$\Delta \boldsymbol{f} \equiv \boldsymbol{f}(\boldsymbol{x_0}\ ) - \boldsymbol{f_T}$$

and the Jacobian $\nabla \boldsymbol{f}$ is now a matrix. Its row number is equal to the number of design variables, and its column number is the same as the number of functions involved in Equation (19).

The SQP of Equation (2) can then be conveniently extended to include the new objective, $\eta(\boldsymbol{s})$ for the case with multiple objective functions as,

$$\min_{\boldsymbol{s}\ \in\ R^n} (\nabla \eta)^T \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}^T W \boldsymbol{s}$$

subject to:

$$(\nabla \boldsymbol{g})^T \boldsymbol{s} + \widetilde{\boldsymbol{g}} = 0$$

where $\boldsymbol{g}$ is the collection of the rest of linearly independent constraints. The search direction is now given by,

$$\boldsymbol{s} = -\alpha \boldsymbol{s}_1 + \boldsymbol{s}_2 = -\alpha W^{-1} P\, \nabla \eta - Q \widetilde{\boldsymbol{g}} = -\alpha W^{-1} P\, \nabla \boldsymbol{f} \Delta \boldsymbol{f} - Q \widetilde{\boldsymbol{g}}$$

The scalar factor $\alpha$ is now determined by minimizing the gap between the revised design and the targeted objectives, $\boldsymbol{f}(\boldsymbol{x_0} + \boldsymbol{s}\ ) - \boldsymbol{f_T}$. To this end, the value of $\alpha$ is found to minimize the error measurement defined in Equation (18),

$$\underset{\alpha}{min} \quad \eta(\alpha) = \left((\nabla \boldsymbol{f})^T \boldsymbol{s} + \Delta \boldsymbol{f}\right)^T \left((\nabla \boldsymbol{f})^T \boldsymbol{s} + \Delta \boldsymbol{f}\right)$$

where $\boldsymbol{s} = -\alpha \boldsymbol{s}_1 + \boldsymbol{s}_2$. The scalar factor $\alpha$ is then found to be

$$\alpha = \frac{\boldsymbol{a}^T \boldsymbol{b}}{\boldsymbol{a}^T \boldsymbol{a}} \tag{20}$$

where $\boldsymbol{a} = (\nabla \boldsymbol{f})^T \boldsymbol{s_1}$ and $\boldsymbol{b} = (\nabla \boldsymbol{f})^T \boldsymbol{s_2} + \Delta \boldsymbol{f}$.

## 3. Results

Three examples are presented in this section to demonstrate the use of methodologies described in Section 2 for the trade-off. The first example trades off three design variables of a cubic box to match two different sets of performance requirements. The second example uses the COA to adjust three design variables of an eigenvalue problem to match tightly three targeted requirements. The third example uses trade-off design recursively to adjust the dimensions of a cantilever beam to achieve different design requirements. The example starts with an initial optimization that aims to reduce the weight of the beam as much as possible while still satisfying constraints imposed upon the deformation, yielding stress, shear stress, and geometry. New performance requirements are then added to problem formulation in sequence. The MOA is used in the last part of Example 3 to handle the cases with linearly dependent function gradients and overloaded constraints.

### *3.1. Example 1: Formulation and Design of a Cubic Box*

The lengths of three edges of a box are assigned to be the design variables, $x_1$, $x_2$, and $x_3$. The performance functions considered here involve the volume, surface area, and weld length of this box, which are defined by,

$$V = x_1 x_2 x_3$$

$$A = 2(x_1 x_2 + x_2 x_3 + x_3 x_1)$$

$$W = 4x_1 + 4x_2 + 4x_3$$

The design starts with the given values as, $x = \{3.0 \quad 2.0 \quad 1.0\}$, which results in a volume of 6, a surface area of 22, and a weld length of 24. Two scenarios will be investigated here: one with two performance functions; volume and surface area, and the other with three; volume, surface area, and weld length.

#### 3.1.1. Two Performance Requirements: Volume and Surface Area

The goal here is to update the design variables that can reduce the current volume from 6 to 4 and the surface area from 22 to 18. The SOP will be used in the first attempt in which the volume is assigned as the objective and the surface area the constraint. Consequently, the values of $\Delta f$ and $\widetilde{g}$ are 2 and 4, respectively. The search direction for the objective correction is given by,

$$\boldsymbol{s}_1 \equiv P\nabla f = \begin{pmatrix}-0.88 & -0.84 & 1.20\end{pmatrix}^T$$

while the search direction for constraint correction is given by

$$\boldsymbol{s}_2 \equiv -Q\widetilde{\boldsymbol{g}} = \begin{pmatrix}-0.12 & -0.16 & -0.20\end{pmatrix}^T$$

On the other hand, the scalar factor $\alpha$ calculated by Equation (12) is found to be,

$$\alpha = \frac{-\Delta f + \nabla f^T \boldsymbol{s}_2}{\nabla f^T \boldsymbol{s}_1} = 0.0274$$

Therefore, the required changes in the design variables are found to be,

$$\boldsymbol{s} = -\alpha \boldsymbol{s_1} + \boldsymbol{s_2} = \begin{pmatrix}-0.0959 & -0.1370 & -0.2329\end{pmatrix}^T$$

which produces the revised design as

$$\boldsymbol{x} = \boldsymbol{x_0} + \boldsymbol{s} = \begin{pmatrix}2.904 & 1.863 & 0.767\end{pmatrix}^T$$

As a result, the volume and the surface area of the revised design are obtained as 4.15 and 18.13, respectively, which are close to the target values of 4 and 20.

The COA is used in the second attempt to solve the same design requirement. However, in this case, both volume and surface area are counted as constraints, and their required changes are set to be,

$$\widetilde{g} = \begin{Bmatrix} \Delta f \\ \widetilde{g} \end{Bmatrix} = \begin{Bmatrix} 2 \\ 4 \end{Bmatrix}$$

This approach produces the same revised design as the SOA.

Note that the trade-off design proposed here is based upon the linearized approximation of all functions of concern. Therefore, it is expected that a better trade-off design can be achieved with less amount of adjustment in the performance measurements. To demonstrate this matter, an exercise was performed with the amount of adjustment in volume being reduced in sequence from 2 to 1 and finally to 0.5, and the amount of adjustment in the surface area was reduced from 4 to 2 and to 1. The results are summarized in Table 1.

**Table 1.** Fewer changes in function adjustments produce better results.

| Cases | $x_1$ | $x_2$ | $x_3$ | Volume/Target | Surface Area/Target |
|---|---|---|---|---|---|
| Initial | 3 | 2 | 1 | 6 | 22 |
| Reduction (−2, −4) | 2.904 | 1.863 | 0.769 | 4.150/4.0 | 18.134/18 |
| Reduction (−1, −2) | 2.952 | 1.932 | 0.884 | 5.038/5.0 | 20.034/20 |
| Reduction (−0.5, −1) | 2.976 | 1.966 | 0.941 | 5.509/5.5 | 21.008/21 |

### 3.1.2. Three Performance Requirements: Volume, Surface Area, and Weld Length

A new performance is added to the constraint set in this example, which requires the weld length to be 20. The special feature of this new example is that it is subjected to three equality constraints which are equal to the number of the design variables. Therefore, the constraint set can be directly employed to solve for three design variables. To this end, a root-finding function, *fsolve*, in Matlab (R2023a) is applied here but failed to generate a solution.

The COA is then employed to reach a revised design,

$$\boldsymbol{x} = \boldsymbol{x_0} + \boldsymbol{s_2} = \begin{pmatrix} 0.5 & 4.0 & 0.5 \end{pmatrix}^T \tag{21}$$

The new design produces a cube with a volume, a surface area, and a weld length of 1.0, 8.5, and 20. The new design satisfies the linear constraint in the weld length but fails to match the required volume and surface area. A second attempt is made to use the above solution as the initial design to restart the new design revision process. The COA failed as the Jacobian of the constraints is singular at this initial design stated in Equation (21). This is because the gradient of the volume is now linearly dependent on the gradients of the surface area and the weld length at the current design point. Specifically, the linear dependence of gradients can be described as

$$\nabla V = 0.25 \times \nabla A - 0.625 \times \nabla W$$

The MOA was then used to find the revised design with linearly dependent constraint gradients. The trade-off design starts with the design specified in Equation (21). The performance requirements of the volume and the surface area will be included in the composite function defined in Equation (18), while the weld length is kept as an equality constraint. Since the initial design has the weld length matched with the targeted value, it starts with $\widetilde{g} = 0$ in this case, which leads to $\boldsymbol{s_2} = \boldsymbol{0}$. Consequently, the computation of the search direction is simplified as $\boldsymbol{s} = -\alpha \boldsymbol{s}_1$, so does the scalar factor, which is stated in Equation (21) with $\boldsymbol{a} = (\nabla \boldsymbol{f})^T \boldsymbol{s_1}$ and $\boldsymbol{b} = \Delta \boldsymbol{f}$. It takes three recursive runs to reach a satisfactory solution. The largest gap between the final and the targeted performances was found in the surface area.

A Matlab built-in function, *fgoalattain* [17], which is a multi-objective goal attainment program, is also applied here to solve the same problem. The volume and the surface area are given as the objectives, with the goals being 2 and 4, respectively. The limit on the weld length is provided as an equality constraint. It takes 10 iterations to reach the solution. The results of these exercises are listed in the last row of Table 2 for comparison. The goal attainment program produces a result with equal design variables. It performs better in surface area but worse in volume in comparison with the results of the MOA.

**Table 2.** Multi-objective approach for dependent constraint gradients.

| Cases | $x_1$ | $x_2$ | $x_3$ | Volume/Target | Surface Area/Target | Weld Length/Target |
|---|---|---|---|---|---|---|
| Initial | 0.5 | 4.0 | 0.5 | 1.0/4.0 | 8.5/18.0 | 20.0/20.0 |
| Run 1 | 1.1891 | 2.6218 | 1.1891 | 3.7070 | 15.2981 | 20.0 |
| Run 2 | 1.5599 | 1.8803 | 1.5595 | 4.5753 | 16.5989 | 20.0004 |
| Run 3 | 2.0220 | 0.9562 | 2.0220 | 3.9092 | 15.9101 | 20.0004 |
| Goal Att. | 1.6667 | 1.6667 | 1.6667 | 4.6296 | 16.6667 | 20.0 |

### *3.2. Example 2: Control Problem with Three Targeted Eigenvalues*

The goal of this example is to modify the design to achieve multiple targeted changes in the performance requirements. This is done with the recursive use of the Constraint Only Approach (COA).

The performance of concern is the eigenvalues of a $3 \times 3$ matrix equation,

$$D\boldsymbol{y} = \lambda \boldsymbol{y}$$

where matrix $D$ is a matrix of 4 design variables, $x_1$ , $x_2$, $x_3$ and $x_4$, as specified below,

$$D = \begin{bmatrix} -0.5 + x_1 & 0 & x_2 \\ -2x_1 + 2x_3 & -2 & 10 - 2x_2 + 2x_4 \\ x_3 & 1 & -2 + x_4 \end{bmatrix}$$

With initial design values, $x_1 = x_3 = x_4 = -4$, and $x_2 = -0.2564$, the matrix $D$ is given by,

$$D = \begin{bmatrix} -4.5 & 0 & -0.2564 \\ 0 & -2 & 2.5128 \\ -4.0 & 1 & -6 \end{bmatrix}$$

and the eigenvalue equation yields three eigenvalues, $\lambda_1 = -6.9314$, $\lambda_2 = -4.1587$ and $\lambda_3 = -1.4099$. The goal now is to modify $D$ such that eigenvalues can match with the targeted values; $\lambda_1^* = -5.0$, $\lambda_2^* = -3.0$ and $\lambda_3^* = -1.0$. To this end, the performance requirements can be expressed as a set of three equality constraints according to Equation (14):

$$g_1 \equiv \lambda_1(\boldsymbol{x} + \boldsymbol{s}) - \lambda_1^* = 0$$

$$g_2 \equiv \lambda_2(\boldsymbol{x} + \boldsymbol{s}) - \lambda_2^* = 0$$

$$g_3 \equiv \lambda_3(\boldsymbol{x} + \boldsymbol{s}) - \lambda_3^* = 0$$

The required change in this initial design is given by $\tilde{\boldsymbol{g}} = -\begin{pmatrix} 1.931 & 1.159 & 0.410 \end{pmatrix}^T$. The COA is employed here recursively to find the revised design. After four iterations, the revised design can produce eigenvalues close to the targeted values with errors less than $10^{-4}$. The results of this trade-off design process are summarized in Table 3.

**Table 3.** Recursive Trade-off Design History to Match Three Eigenvalues.

| Study Case | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ |
|---|---|---|---|---|---|---|---|
| Initial | −4 | −0.2564 | −4 | −4 | −6.9314 | −4.1587 | −1.4099 |
| Run 1 | −1.9544 | 0.2879 | −3.9360 | −2.8956 | −5.6370 | −2.2574 | −1.4556 |
| Run 2 | −2.4562 | 0.5175 | −3.8645 | −2.0438 | −4.9435 | −3.0723 | −0.9842 |
| Run 3 | −2.4196 | 0.4990 | −3.8695 | −2.0804 | −4.9990 | −3.0019 | −0.9998 |
| Run 4 | −2.1486 | 0.4984 | −3.8696 | −2.0814 | −4.9999 | −3.0001 | −1.0000 |

The goal attainment program, *fgoalattain*, is again employed here to resolve the above problem. The object in its formulation is to have all three eigenvalues matched with the targeted values. The results are found to be $x = \begin{pmatrix} -1.5954 & 1.2040 & -0.4201 & -2.9046 \end{pmatrix}^T$, which is different from that presented in Table 3. The solution to the problem studied here is not unique.

### *3.3. Example 3: Design Problem of an I-Beam*

A design problem of an I-beam is used as a platform to demonstrate the trade-offs between the values of design variables and the design performance requirements. The cantilever I-beam, shown in Figures 1 and 2, is required to be as light as possible and be able to support a uniformly distributed load, $P$, without failure.

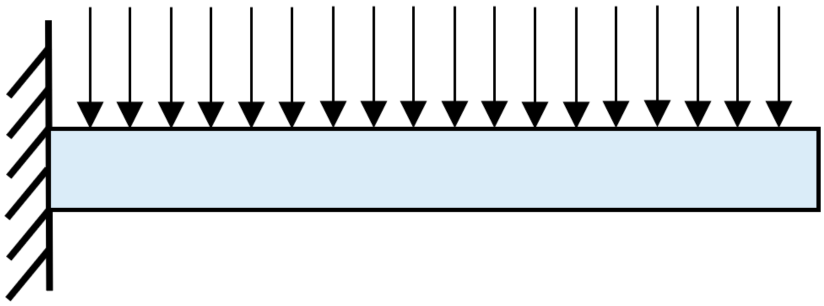

**Figure 1.** Cantilever Beam.

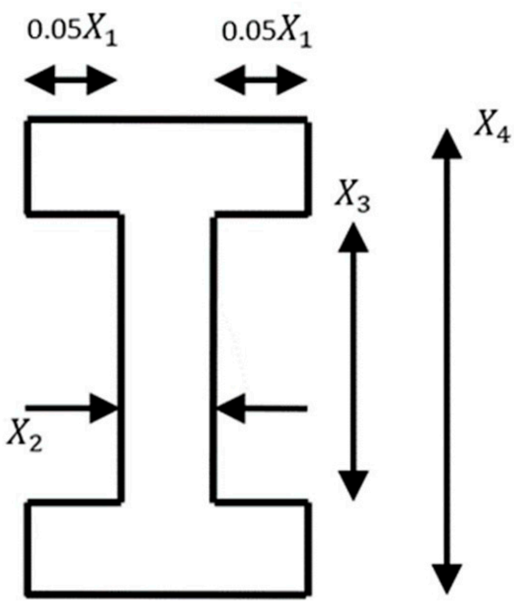


**Figure 2.** I Beam cross section.

The design variables are those specifying the cross-sectional dimensions of the beam. They are defined as $x_1$, the width of the beam minus the web thickness, $x_2$, the web thickness, $x_3$, the height minus the thickness of both flanges and $x_4$, the total height of the beam. Based on the definition of the variables, the cross sections area, the moment of inertia $I$, and the area of the moment, $Q$, are expressed as:

$$A = x_1x_4 + x_2x_4 - x_1x_3$$

$$I = \frac{x_1x_4^3 + x_2x_4^3 - x_1x_3^3}{12}$$

and

$$Q = \frac{(x_1 + x_2)(x_4^2 - x_3^2) + x_2x_3^2}{8}$$

The total distributed load $\omega$ on the beam is equal to the summation of the externally distributed load and the weight density per unit length. Thus, the total distributed load is given by

$$\omega = P + \rho A$$

where $\rho$ is the density of the beam per unit volume, and $A$ is the cross-sectional area. The performance requirements in this I-beam design problem include the total weight, the maximal deflection at the free end, $\delta_{max}$, the maximal bending stress, $\sigma_{max}$, and the maximal shear stress at the fixed end, $\tau_{max}$, and the two fundamental frequencies, $f_1$ and $f_2$.

For this example, the length of the beam $\ell$ is set to be 40 inches, and the uniformly distributed load $P$ is set at 25 lbf/in. The beam is made of steel with Young's modulus $E$ of $30 \times 10^6$ psi and a density $\rho$ of 29 lbf/in$^3$. The maximum allowable yielding stress $\sigma_{yd}$ is equal to 12 kpsi, the maximal deflections $\delta_o$, is 0.1 inches, and maximum shear stess $\tau_o$, is 1500 lbf/in$^2$. These performance measurements are expressed as

$$W_t = \rho\ell A$$

$$\delta_{max} = \frac{\omega\ell^4}{8EI}$$

$$\sigma_{max} = \frac{\omega\ell^2x_4}{4I}$$

$$\tau_{max} = \frac{\omega\ell\big((x_1 + x_2)(x_4^2 - x_3^2) + x_2x_3^2\big)}{8\ x_2I}$$

The fundamental frequencies are calculated from the eigenvalues of the I-beam. To this end, the I-beam is discretized into two Euler beam elements, based upon which an eigenvalue matrix equation is built as follows

$$K\boldsymbol{y}_i = \boldsymbol{\lambda}_i M\boldsymbol{y}_i \tag{22}$$

where the inputs are $K$ and $M$, the stiffness and the mass matrices, and the output is $\lambda_i$ and $\boldsymbol{y}_i$ are the pair of the $i$th eigenvalue and eigenvector. Here the eigenvector is normalized. That is $\boldsymbol{y}_i^T M\boldsymbol{y}_i = 1$. The equations of the first and second fundamental frequencies can then be defined as,

$$f_1 = \frac{\sqrt{\lambda_1}}{2\pi}$$

$$f_2 = \frac{\sqrt{\lambda_2}}{2\pi}$$

The gradients of the frequencies with respect to the design variables can then be found by differentiating Equation (22) as

$$\frac{\partial f_i}{\partial x_k} = \frac{\sqrt{\lambda_i}}{4\pi}\left(\frac{1}{I}\frac{\partial I}{\partial x_k} - \frac{1}{A}\frac{\partial A}{\partial x_k}\right) \tag{23}$$

By careful examination, the above equation reveals that the gradient vectors of distinct eigenvalues of the I-beam problem are linearly independent of each other.

Three trade-off design examples are reported separately in Sections 3.3.2–3.3.4. They are carried out using the result of an optimal design run as the initial design point. The optimal design run is reported in Section 3.3.1. Example 1 in Section 3.3.2 deals with three targeted constraint functions. Example 2 in Section 3.3.3 deals with four constraints, among which two constraints exhibit linearly dependent gradients. Example 3 in Section 3.3.4 works with five constraints. The challenge in Example 3 is that the number of the targeted constraints is greater than the number of the design variables. To demonstrate its effectiveness, the MOA will be used to solve Examples 2 and 3.

### 3.3.1. Initial Design Optimization

The design process starts with an optimization problem to minimize the weight subjected to constraints on deflection, normal stress, shear stresses, and geometry. The geometry constraint ensures that the thickness of the flange is greater than 0.25 inches and the bounds of the dimensions are in the range between 0.5 and 5 inches. The mathematical formulation of the design optimization problem is expressed below

$$\min_{X_1,X_2,X_3,X_4} W = \rho\ell A = \rho\ell(x_1x_4 + x_2x_4 - x_1x_3)$$

subject to:

$$g_1 = \frac{\omega\ell^4}{8EI} - \delta_o \le 0$$

$$g_2 = \frac{\omega\ell^2 X_4}{4I} - \sigma_{yd} \le 0$$

$$g_3 = \frac{\omega\ell\left((x_1+x_2)\left(x_4^2 - x_3^2\right) + x_2x_3^2\right)}{8\,x_2 I} - \tau_o \;\le 0$$

and the geometry constraint and the bounds

$$g_4 = x_3 - x_4 + 0.5 \le 0$$

$$5 \ge x_1, x_2, x_3, x_4 \ge 0.5$$

All performance constraints are weighted equally in the optimization process, which is normalized with respect to their upper limits.

The initial optimization was solved using the Matlab built-in function, *fmincon*, with the initial guess (2, 0.25, 1, 2). The optimal design was found to be (0.7469, 0.5, 3.1583, 3.6583). This result produces an I-beam with a weight of 25.55 lbs, a deformation of 0.0875 inches, a yielding stress of 12 kpsi, and a shear stress of 757.5 lbf/in$^2$. More specifically, the yielding stress of the final design hits the upper bound, the design variable, $x_2$, which is the thickness of the web, hits the lower bound, and the geometry constraint, or the difference between $x_4$ and $x_3$, hits the bound. Thus, the thickness of the flange also reaches the lower bound. In short, the four design variables of the I-beam problem are subjected to three tight constraints, $\sigma_{max} = \sigma_{yd}$, $x_3 - x_4 + 0.5 = 0$, and $x_2 = 5$, at the optimal solution.

### 3.3.2. Example 1: Trade-Offs with Three Performance Functions

The goal of this example is to modify the optimal design to have three selected performance functions matched exactly with the targeted values. The first case will consider the maximal deflection, the maximal yielding stress, and the geometry constraint, while the second case will replace the geometry constraint with the constraint on the first natural frequency.

The first case was solved by the SOA, in which the deformation is set as the objective function, while the yielding stress and the difference between $X_4$ and $X_3$, which is referred to as the geometry constraint, were set as the targeted equality constraints. The initial design is set at (0.7469, 0.5000, 3.1583, 3.6583), which is the optimal design result obtained from Section 3.3.1. The SOA takes four iterations to produce a revised design, (0.81341, 0.68104, 2.7, 3.2), which results in a deformation of 0.1 inches, yielding stress of 12 Kpsi, and the difference between $x_3$ and $x_4$ of 0.5 inches. Therefore, it accurately meets all the desired requirements. The same problem is solved by the COA, which produces the same result.

The second case will continue the previous study, with its starting design point setting as (0.81341, 0.68104, 2.7, 3.2). In this case, though, the existing geometry constraint in the existing constraint set will be replaced by the constraint imposed upon the first fundamental frequency, which is required to be 80 Hz. The SOA took four iterations to match closely to the targeted constraints. The final design is (1.0545, 0.3869, 2.3932, 3.2), which results in a first fundamental frequency of 80 hertz, a deformation of 0.1 inches, a yielding stress of 12 Kpsi, and a difference between $x_3$ and $x_4$ of 0.807 inches. It is important to note that the geometry constraint is higher than the previous targeted value; however, the geometry constraint is not being imposed in this case. Therefore, it accurately meets all the desired performance requirements.

This case is also solved by using the COA. In this approach, the first fundamental frequency, deformation, and yielding stress are all set as constraints. Like Example 1, the results produced by the COA are the same as those by the SOA. The results of these two cases are summarized in Table 4.

**Table 4.** Recursive trade-off design history for Example 3.3.

| Cases | $x_1$ | $x_2$ | $x_3$ | $x_4$ | Weight | Defl. | Stress | $x_4-x_3$ | Freq 1 | Freq 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| Opt | 0.747 | 0.5 | 3.158 | 3.658 | 25.55 | 0.0875 | 12,000 | 0.5 | 83.349 | 526.52 |
| Ex. 1 Case 1 | 0.813 | 0.681 | 2.7 | 3.2 | 29.99 | 0.1 | 12,000 | 0.5 | 72.098 | 455.44 |
| Ex. 1 Case 2 | 1.055 | 0.387 | 2.393 | 3.2 | 24.23 | 0.0999 | 11,999 | 0.8068 | 79.999 | 505.36 |
| Ex. 2 | 1.024 | 0.414 | 2.409 | 3.2 | 24.75 | 0.1 | 12,000 | 0.7905 | 79.172 | 500.13 |
| Goal att. | 1.028 | 0.411 | 2.407 | 3.2 | 24.69 | 0.1 | 12,000 | 0.7926 | 79.267 | 500.73 |
| Ex. 3 | 1.374 | 0.452 | 2.7 | 3.2 | 24.75 | 0.1 | 12,000 | 0.5 | 79.172 | 500.13 |
| Goal att. | 1.380 | 0.449 | 2.7 | 3.2 | 24.69 | 0.1 | 12,000 | 0.5 | 79.267 | 500.73 |

### 3.3.3. Example 2: Trade-Offs with Four Performance Functions

The goal for this example is to demonstrate the use of MOA to handle situations where the gradients of involved functions are not all linearly independent of each other. Neither the SOA nor the COA can work in these cases. This example will consider four performance functions, including the first, the second fundamental frequencies, the maximal deflection, and the maximal stress constraints. Note that the gradient of the second fundamental frequency, which is newly added to the problem formulation, is parallel to the gradient of the first fundamental frequency, as indicated by Equation (23).

The MOA will be employed in this case, in which the objective function is made of both first and second fundamental frequencies, and the constraints include the maximal deflection and the maximal stress. The approach starts with an initial design at (1.0545, 0.3869, 2.3932, 3.2), which is the result obtained at the end of Example 1. After two iterations, the approach reaches a design with an acceptable result. The design is $(1.0242,\ 0.41375,\ 2.4095,\ 3.2)$, which results in a first fundamental frequency of 79.172 Hz and a second fundamental frequency of 500.1311 Hz and achieves the targeted constraints: the maximal deflection, 0.1 inches, and the maximal stress, 12 Kpsi.

For comparison, this problem is also solved using the goal attainment function in Matlab, *fgoalattain*. The goal is to enforce the two fundamental frequencies to be the same as the respective targeted values. To this end, the *EqualityGoalCount* option is selected as an input parameter for this run. The optimization was terminated after 20 iterations because the maximal constraint violation was less than $10^{-6}$. The results are listed in Table 4, which are close to those obtained by the MOA.

#### 3.3.4. Example 3: Trade-Off with Five Performance Functions

Example 2 is repeated here with the same objective and the same initial design. However, one more equality constraint on geometry, $x_4 - x_3 = 0.5$, is added back to the constraint set. Therefore, the number of the targeted functions is now 5, which is higher than the number of the design variables. The problem will be solved by the MOA as well as the goal attainment program, *fgoalattain*. The targeted objectives will be the fundamental frequencies 1 and 2.

The MOA takes three iterations to reach the design $(1.3739,\ 0.45209,\ 2.7,\ 3.2)$, which results in a first fundamental frequency of 79.172 Hz, a second fundamental frequency of 500.131 Hz and achieves the targeted constraints; the maximal deflection, 0.1 inches, the maximal stress, 12 Kpsi, and the newly added constraint, $x_4 - x_3,\ 0.5$. On the other hand, the goal-attainment program produces the final design at $(1.3804,\ 0.44944,\ 2.7,\ 3.2)$ after seven iterations. The new design satisfies the targeted constraint values and produces the two fundamental frequencies of 79.26 Hz and 500.73 Hz.

The results of all seven examples investigated in this I-beam problem are summarized in Table 4. The four design variables of the I-beam problem are related to the sectional geometry, while the performance requirements involve the weight, the maximal deflection, the maximal stress, the thickness of the flange, and the first two fundamental frequencies. The active performance constraints considered in each case are indicated by the bold numbers in Table 4. The results show that adding the deflection and the first frequency requirements in the constraint set will not affect the value of the design variable $x_4$, which is the height of the beam. Frequency 2 can be added as a new performance requirement without causing too many changes in the design variables. On the other hand, adding the geometry constraint as a new requirement will increase about 30% the values of the design variables $x_1$ and $x_3$, which are related to the width of the flange and the depth of the web. The weight of the I-beam remains stable throughout this trade-off study, which is not considered here as a constraint.

## 4. Concluding Remarks

The goal of a trade-off methodology is to find a design balance between demands and supplies. In common practice, the demands are the performance requirements, while the supplies are the available design alternatives. The trade-off presented in this paper, however, offers a different design problem formulation. The new trade-off process starts with an existing design. The goal now is not to find an optimal design but rather to find the most effective way to modify the current design variables to achieve the newly revised or added performance requirements. To this end, three different trade-off formulations and solution procedures were developed in this study, the Single Objective Approach (SOA), the Constraint Only Approach (COA), and the Multi-objective Approach (MOA). All these developments are solved by the traditional quadratic programming techniques to produce a new design that can achieve the revised performance requirements with minimal modification of the current design variables. An equation is developed in this study to compute a scale factor that can adjust the size of the new design variables to accurately achieve the required changes imposed upon the performance functions, regardless of whether they are formulated as objectives or constraints. This scale factor is the most critical element in these proposed trade-off methods. Furthermore, two critical issues that often prevent the use of gradient-based optimization algorithms are addressed by the MOA approach in this study. One is when the gradients of some function requirements are linearly dependent on each other, and the other is when the number of functions of concern is greater than the number of design variables.

Three examples are documented in this study. The first example designs a cubic box with concerns about its volume, surface area, and weld length. This example shows that the SOA produces the same results as the COA. This result not only proves that the scalar factor equation derived for objective function correction is accurate but also provide users with more choices in formulating trade-off design for different engineering applications. The same example also demonstrates that the trade-off methods developed here can produce a reasonable design while the root-finding algorithm cannot in a case when the number of the design variables is equal to that of the constraints.

The second example demonstrates the use of the COA to match three eigenvalues of an eigenvalue equation by adjusting four design variables. The approach is compared well to the goal-attainment algorithm in terms of accuracy and efficiency. The third example demonstrates the success of using the MOA to handle the case with linearly dependent gradients and oversaturated constraints.

These examples demonstrate the effectiveness of using the proposed methods to find the revised design variables for a broad range of trade-off engineering applications. The proposed methods can easily handle the addition or removal of the design variables and the performance requirements. They are computationally efficient and can be easily implemented, as the proposed methods are gradient-based, which require first-order derivatives with respect to the design variables. Consequently, all performance functions involved in the proposed methods must be continuous and differentiable. It will be a challenge for future research to extend the proposed methodology to much broader applications with non-differentiable functions.

**Author Contributions:** Conceptualization, G.H. and L.A.R.; methodology, G.H.; validation, L.A.R. and G.H.; writing—original draft preparation, L.A.R.; writing—review and editing, G.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All of the data is provided within the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

1. $\nabla f^T \boldsymbol{s_1} = 0$ for $\nabla f$ being linearly dependent on $\nabla \boldsymbol{g}$

Assume that the gradient of the objective, $\nabla f$, is linearly dependent of the columns of $\nabla \boldsymbol{g}$. That is, $\nabla f$ can be expanded as a linear combination of the columns of $\nabla \boldsymbol{g}$ or $\nabla f = (\nabla \boldsymbol{g})\boldsymbol{c}$ with a nonzero vector $\boldsymbol{c}$. One can then proceed to prove that $\nabla f^T \boldsymbol{s_1} = 0$ as follows.

$$
\begin{aligned}
&\nabla f^T \boldsymbol{s_1} = \nabla f^T \left(W^{-1} P \nabla f\right) \equiv \boldsymbol{c^T} (\nabla \boldsymbol{g})^T W^{-1} P (\nabla \boldsymbol{g}) \boldsymbol{c} \\
&= \boldsymbol{c^T} (\nabla \boldsymbol{g})^T W^{-1} \left\{ I - (\nabla \boldsymbol{g}) \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right]^{-1} (\nabla \boldsymbol{g})^T W^{-1} \right\} (\nabla \boldsymbol{g}) \boldsymbol{c} \\
&= \boldsymbol{c^T} \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right] \boldsymbol{c} - \boldsymbol{c^T} \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right] \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right]^{-1} \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right] \boldsymbol{c} \\
&= \boldsymbol{c^T} \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right] \boldsymbol{c} - \boldsymbol{c^T} \left[ (\nabla \boldsymbol{g})^T W^{-1} (\nabla \boldsymbol{g}) \right] \boldsymbol{c} = 0
\end{aligned}
$$

2. Scalar Factor and Search Direction of the Constraint Only Approach

The scalar factor, $p_f$, and the search direction, $\boldsymbol{s_2}$, can be derived based upon Equation (16), which can be expanded as

$$\begin{bmatrix} \nabla f^T W^{-1} \nabla f & \nabla f^T W^{-1} \nabla \overline{g} \\ \nabla \overline{g}^{\mathbf{T}} W^{-1} \nabla f & \nabla^T \overline{g} W^{-1} \nabla \overline{g} \end{bmatrix} \begin{Bmatrix} p_f \\ \overline{\boldsymbol{p}}_{\overline{g}} \end{Bmatrix} = \begin{Bmatrix} \Delta f \\ \overline{g} \end{Bmatrix} \tag{A1}$$

The first row of Equation (A1) is found to be

$$\left( \nabla f^T W^{-1} \nabla f \right) p_f + \left( \nabla f^T W^{-1} \nabla \overline{g} \right) \overline{\boldsymbol{p}}_{\overline{g}} = \Delta f \tag{A2}$$

The solution of the second row of Equation (A1) is found to be

$$\overline{\boldsymbol{p}}_{\overline{g}} = -\left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} \left( \nabla \overline{g}^{\boldsymbol{T}} W^{-1} \nabla f \right) p_f + \left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} \overline{g} \tag{A3}$$

Substituting Equation (A2) for $\overline{\boldsymbol{p}}_{\overline{g}}$ in the above equation yields a single equation of $p_f$ as

$$\begin{aligned} &[(\nabla f^T W^{-1} \nabla f) - (\nabla f^T W^{-1} \nabla \overline{g})(\nabla^T \overline{g} W^{-1} \nabla \overline{g})^{-1} (\nabla \overline{g}^T W^{-1} \nabla f)] p_f \\ &= \Delta f - (\nabla f^T W^{-1} \nabla \overline{g})(\nabla^T \overline{g} W^{-1} \nabla \overline{g})^{-1} \overline{g} \end{aligned}$$

which can be expanded to obtain the value of $p_f$ explicitly as

$$\begin{aligned} &p_f = [(\nabla f^T W^{-1} \nabla f) - (\nabla f^T W^{-1} \nabla \overline{g})(\nabla^T \overline{g} W^{-1} \nabla \overline{g})^{-1} (\nabla \overline{g}^{\boldsymbol{T}} W^{-1} \nabla f)]^{-1} \\ &\times [\Delta f - (\nabla f^T W^{-1} \nabla \overline{g})(\nabla^T \overline{g} W^{-1} \nabla \overline{g})^{-1} \overline{g}] \\ &= [\nabla f^T W^{-1} (I - \nabla \overline{g} (\nabla^T \overline{g} W^{-1} \nabla \overline{g})^{-1} \nabla \overline{g}^T W^{-1}) \nabla f]^{-1} [\Delta f - \nabla f^T \overline{Q} \overline{g}] \\ &= [\nabla f^T W^{-1} \widetilde{P} \nabla f]^{-1} (\Delta f - \nabla f^T \overline{Q} \overline{g}) \\ &= \frac{\Delta f - \nabla f^T \overline{Q} \overline{g}}{\nabla f^T \overline{\boldsymbol{s}}_{\boldsymbol{1}}} \\ &= \frac{\Delta f + \nabla f^T \overline{\boldsymbol{s}}_{\boldsymbol{2}}}{\nabla f^T \overline{\boldsymbol{s}}_{\boldsymbol{1}}} \end{aligned}$$

where $\overline{P}$, $\overline{Q}$, $\overline{\boldsymbol{s}}_{\boldsymbol{1}}$, and $\overline{\boldsymbol{s}}_{\boldsymbol{2}}$ are those defined by Equations (5)–(9) in terms of the original constraint set, $\overline{g}$, which does not include the objective function, $f$. Consequently, $p_f$ derived here is identical to the scalar factor, $\alpha$, derived earlier in Equation (12). The search direction, $\boldsymbol{s}_2$, defined in Equation (15), can now be expanded as,

$$\begin{aligned} &\boldsymbol{s} = \boldsymbol{s}_2 = -Q\widetilde{g} = W^{-1}(\nabla g)\left[(\nabla g)^T W^{-1} (\nabla g)\right]^{-1} \widetilde{g} = -W^{-1}(\nabla g)\boldsymbol{p} \\ &= -W^{-1}\{\ \nabla f \quad \nabla \overline{g}\ \} \begin{Bmatrix} p_f \\ -\left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} (\nabla \overline{g}^T W^{-1} \nabla f) p_f + \left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} \overline{g} \end{Bmatrix} \\ &= -W^{-1}\left[ (\nabla f) p_f - \nabla \overline{g} \left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} (\nabla \overline{g}^T W^{-1} \nabla f) p_f + \nabla \overline{g} \left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} \overline{g} \right] \\ &= -\left[ W^{-1} \left\{ \left( I - \nabla \overline{g} \left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} \right) \nabla \overline{g}^T W^{-1} \right\} \nabla f p_f + \left\{ W^{-1} \nabla \overline{g} \left( \nabla^T \overline{g} W^{-1} \nabla \overline{g} \right)^{-1} \right\} \overline{g} \right] \\ &= \left[ -W^{-1} \overline{P} \nabla f p_f - \overline{Q} \overline{g} \right] = -p_f \overline{\boldsymbol{s}}_{\boldsymbol{1}} + \overline{\boldsymbol{s}}_{\boldsymbol{2}} = -\alpha \overline{\boldsymbol{s}}_{\boldsymbol{1}} + \overline{\boldsymbol{s}}_{\boldsymbol{2}} \end{aligned}$$

which provides the proof of Equation (17).

## References


1. Tan, J.; Otto, K.; Wood, K. Concept Design Trade-Offs Considering Performance Margins. In *Proceedings of Nord Design, Throndheim, Norway, 10–12 August 2016*; The Design Society: Glasgow, UK, 2016; Volume 1, pp. 421–429. Available online: https://www.researchgate.net/publication/312023209_Concept_Design_Trade-Offs_Considering_Performance_Margins (accessed on 24 May 2023).
2. Otto, K.N.; Antonsson, E.K. Trade-off strategies in engineering design. *Res. Eng. Des.* **1991**, *3*, 87–104. [CrossRef]
3. Silveira, G.D.; Slack, N. Exploring the Trade-Off Concept. *Int. J. Oper. Prod. Manag.* **2001**, *21*, 949–964. [CrossRef]

4. Nassar, N.; Austin, M. Model-based systems engineering design and trade-off analysis with RDF graphs. *Procedia Comput. Sci.* **2013**, *16*, 216–225. [CrossRef]
5. Bae, S.; Kim, N.H.; Jang, S. System reliability-based design optimization under tradeoff between reduction of sampling uncertainty and design shift. *J. Mech. Des. Trans. ASME* **2019**, *141*, 4. [CrossRef]
6. Wang, J.; Terpenny, J.P. Learning and Adapting Fuzzy Set-Based Trade-Off Strategy in Engineering Design Synthesis. *Int. J. Smart Eng. Syst. Des.* **2003**, *5*, 177–186. [CrossRef]
7. Rojas, J.A.M.; Fernández, J.L.; Sánchez Montero, R.; Espí, P.L.L.; Diez-Jimenez, E. Model-Based Systems Engineering Applied to Trade-Off Analysis of Wireless Power Transfer Technologies for Implanted, Biomedical Microdevices. *Sensors* **2021**, *21*, 3201. [CrossRef] [PubMed]
8. Haug, E.J.; Arora, J.S. *Applied Optimal Design: Mechanical and Structural Systems*; John Wiley & Sons: Hoboken, NJ, USA, 1979.
9. Gero, J.S. (Ed.) Design Optimization. In *Notes and Reports in Mathematics in Science and Engineering*; Academic Press: Cambridge, MA, UK, 1985; Volume 1.
10. Haftka, R.T.; Gurdal, Z. *Elements of Structural Optimization*; Kluwer Academic Publisher: Boston, MA, USA, 1992.
11. Arora, J.S. *Introduction to Optimum Design*; Academic Press: Waltham, MA, USA, 2012.
12. Belegundu, A.D.; Chandrupatla, T.R. *Optimization Concepts and Applications in Engineering*; Cambridge University Press: Cambridge, UK, 2019.
13. Choi, K.K.; Haug, E.J.; Hou, J.W.; Sohoni, V.N. Pshenichny's linearization method for mechanical system optimization. *J. Mech. Transm. Autom. Des.* **1983**, *105*, 97–103. [CrossRef]
14. Heaney, P.S.; Hou, G. Projection method with minimal correction procedure for numerical simulation of constrained dynamics. In Proceedings of the ASME 2017 Dynamic System and Control Conference, Tysons, VA, USA, 11–17 October 2017; Paper No. DSCC2017-5212. [CrossRef]
15. Haug, E.J. *Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume II: Modern Methods*, 2nd ed.; Pearson College Div: New York, NY, USA, 2021. Available online: https://www.researchgate.net/publication/341385504_Computer-Aided_Kinematics_and_Dynamics_of_Mechanical_Systems_Vol_II_Modern_Methods (accessed on 24 May 2023).
16. Bauchau, O.A.; Laulusa, A. Review of contemporary approaches for constraint enforcement in multibody systems. *J. Comput. Nonlin. Dyn.* **2008**, *3*, 011005. [CrossRef]
17. Fgoalattain: Solving Multiobjective Goal Attainment Problems. Available online: https://www.mathworks.com/help/optim/ug/fgoalattain.html (accessed on 24 May 2023).