Optimization and Engineering
https://doi.org/10.1007/s11081-025-10001-4

RESEARCH ARTICLE

Compound optimal design of experiments – Semidefinite Programming formulations

Belmiro P. M. Duarte[1,2] · Anthony C. Atkinson[3] · Nuno M. C. Oliveira[4]

Received: 20 June 2024 / Revised: 12 March 2025 / Accepted: 16 May 2025

Abstract

An optimal experimental design represents a structured approach to collecting data with the aim of maximizing the information gleaned. Achieving this requires defining an optimality criterion tailored to the specific model under consideration and the purpose of the investigation. However, it is often observed that a design optimized for one criterion may not perform optimally when applied to another. To mitigate this, one strategy involves employing compound designs. These designs balance multiple criteria to create robust experimental plans that are versatile across different applications. In our study, we systematically tackle the challenge of constructing compound approximate optimal experimental designs using Semidefinite Programming. We focus on discretized design spaces, with the objective function being the geometric or the arithmetic mean of design efficiencies relative to individual criteria. We address two combinations of two criteria: concave-concave (illustrated by DE–optimality) and convex-concave (such as DA–optimality). To handle the latter, we reformulate the problem as a bilevel problem. Here, the outer problem is solved using Surrogate Based Optimization, while the inner problem is addressed with a Semidefinite Programming solver. We demonstrate our formulations using both linear and nonlinear models (for the response) of the Beta class, previously linearized to facilitate analysis and comparison.

✉ Belmiro P. M. Duarte
bduarte@isec.pt

Anthony C. Atkinson
A.C.Atkinson@lse.ac.uk

Nuno M. C. Oliveira
nuno@eq.uc.pt

1 Polytechnic Institute of Coimbra, ISEC, Rua Pedro Nunes, Quinta da Nora, 3030-199 Coimbra, Portugal

2 University of Coimbra, INESC Coimbra, Rua Sílvio Lima – Pólo II, 3030-790 Coimbra, Portugal

3 Department of Statistics, London School of Economics and Political Science, London WC2A 2AE, United Kingdom

4 University of Coimbra, CERES, Rua Sílvio Lima – Pólo II, 3030-790 Coimbra, Portugal

Published online: 05 July 2025

Keywords Compound optimal designs · Semidefinite programming · Concave-concave criteria · Convex-concave criteria · Surrogate based optimization

Mathematics Subject Classification 62K05 · 90C47 · 90C22

1 Motivation

The theoretical framework supporting the model-based optimal design of experiments entails defining the parameter and regressor domains, as well as establishing the model form that relates the response and regressors and the distribution of the response. Additionally, it necessitates selecting an optimality criterion to measure the size of the parameters' confidence region (Chernoff, 1953). While a design may be optimal within a given criterion, its performance could be unsatisfactory if assessed using a different criterion. To address this, one strategic approach involves considering multiple criteria simultaneously, leading to the concept of *compound designs*. These designs aim to maximize combined efficiency across combinations of criteria.

Cook and Wong (1994) were pioneers in addressing this issue, demonstrating that every constrained optimal design is also a compound optimal design, and vice versa. Initially focusing on linear models with two objectives, they showed that compound designs emerge from solving the original optimization problem with additional constraints representing lower bounds on efficiency for other criteria. Clyde and Chaloner (1996) extended this equivalence to nonlinear models, particularly in Bayesian optimal design settings involving multiple objectives.

There are three main approaches to constructing compound optimal designs. The first optimizes a primary criterion while enforcing constraints on others, similar to multi-objective reformulations such as the ϵ-constraint method (Miettinen, 1999). The second approach reformulates the design problem as a min-max optimization, which can be particularly challenging when combining criteria from different classes (e.g., convex and concave). The third approach defines a new objective function by combining multiple design criteria, often through weighted averages or products, with the compound optimal design maximizing a convex combination of these criteria. Building on the robustness criteria introduced by Läuter (1974) for competing models, several common weighting schemes arise: (i) the arithmetic mean, leading to a problem of the Archimedean goal programming class; (ii) the minimum of all elements, resulting in a min-max (or max-min) formulation; and (iii) the geometric mean, another frequently used weighting method.

Another noteworthy aspect is the convenience of compound designs when the goal involves: (i) parameter estimation, where all criteria belong to alphabetic classes aimed at minimizing specific components of the parameter confidence ellipsoid (Atkinson et al., 2007); and (ii) parameter estimation and model discrimination, which encompass various criteria for parametrization along with the T-optimality criterion, as outlined by Atkinson (2008) and McGree et al. (2008). Our paper specifically delves into designs tailored for parametrization. Next, we provide an overview of the techniques employed for their computation.

1.1 Numerical algorithms for compound optimal designs

The current body of literature regarding systematic (numerical) approaches for determining compound optimal designs is limited. Initially, algorithms utilized for single criteria were adapted, with the relative importance of each criterion being imposed upfront (Cook and Wong, 1994). However, recent advancements have introduced cocktail algorithms, which integrate both exchange and multiplicative techniques, as proposed by Yu (2010) and Yang et al. (2013). These methods have been effectively applied in the domain of compound designs, as demonstrated by Cheng and Yang (2019).

Over the past two decades, there has been a consistent utilization of general optimization algorithms to tackle the problem of optimal experimental design. Noteworthy examples of approaches for finding approximate designs encompass Linear Programming (Harman and Jurík, 2008), Second Order Conic Programming (Sagnol, 2011), Semidefinite Programming (Vandenberghe and Boyd, 1999; Duarte and Wong, 2015), Semi-Infinite Programming (Duarte et al., 2015), and Nonlinear Programming (Molchanov and Zuyev, 2002). Additionally, stochastic optimization methods have emerged as successful tools for addressing this challenge. These methods, often derivative-free, guide the evolution of candidate solutions by evaluating local performance through a measure of quality.

Among the stochastic optimization techniques, evolutionary strategies stand out, drawing inspiration from natural systems (Heredia-Langner et al., 2004; Qiu et al., 2014), social dynamics (Masoudi et al., 2019), and local surrogate models (Duarte et al., 2023). These diverse methodologies contribute to a comprehensive toolkit for optimizing experimental designs, catering to different problem characteristics and preferences.

In the context of stochastic optimization algorithms for determining compound designs, McGree et al. (2008) exploited a simulated annealing algorithm, derived from Corana et al. (1987). Hu et al. (2010) merged the cross-entropy method with local search techniques employing gradient descent methods. Similarly, Hyun et al. (2018) adopted an algorithm adapted from Yang et al. (2013) optimal weights exchange algorithm. Qiu and Wong (2023) harnessed the power of the Particle Swarm Optimization algorithm proposed by Kennedy and Eberhart (1995) and the Differential Evolution algorithm introduced by Storn and Price (1997).

Deterministic optimization algorithms have seen limited application in the domain of compound designs, although there is a growing body of literature, as referenced above, that has been successfully applied to single-criterion optimal designs. This trend is expanding to encompass compound designs, with notable contributions such as Wong and Zhou (2023) work utilizing Semidefinite Programming (SDP). They tackle two distinct problems: (i) finding optimal designs relative to a single criterion while considering constraints on other criteria, employing the ϵ-constraint method; and (ii) seeking minimax optimal designs relative to multiple criteria, where all criteria are bounded from below, aiming to maximize efficiency across all criteria.

It is noticeable that objective functions based on the arithmetic mean or geometric mean of single-criteria efficiencies remain unexplored. These functions fall within

the robustness criteria classes A and C proposed by Läuter (1974). Designs derived from these criteria may offer advantages in terms of overall efficiency. To date, this problem remains unsolved, and our work endeavors to address this gap. We present systematic formulations for computing compound designs using convex programming, specifically SDP, with the aim of maximizing both arithmetic and geometric means of single-criteria efficiencies within the setup of compound designs.

1.2 Novelty statement and organization

This paper introduces three novel contributions:

1. a computationally automated approach, based on Semidefinite Programming (SDP), for calculating continuous compound experimental designs. The method considers both geometric and arithmetic means of single criteria efficiencies to represent the objective function;
2. formulations tailored for concave-concave and convex-concave pairs of criteria;
3. the reformulation of the problem pertaining to convex-concave criteria as a bilevel problem. Here, the outer problem is addressed using Surrogate Based Optimization (SBO), while the inner problem is tackled via SDP.

The structure of the paper is as follows: Section 2 provides an introduction to the problem. It outlines the notation used to formulate the optimal design of experiments problem, along with fundamental concepts of Semidefinite Programming and Surrogate Based Optimization — two numerical approaches employed in this study. In Section 3, we present formulations for computing optimal designs under concave-concave and convex-concave criteria. We delve into the specificities of these formulations, particularly highlighting the adaptations necessary to address cases involving both geometric and arithmetic means of single criterion efficiencies. The utilization of these algorithms to compute compound optimal designs for pairs of criteria exhibiting concave-concave and convex-concave characteristics is presented in Sect. 4. Finally, Sect. 5 offers a review of the formulations and provides a summary of the results obtained.

2 Notation and background

In our notation, boldface lowercase letters represent vectors, while boldface capital letters denote continuous domains. Blackboard bold capital letters are utilized to indicate discrete domains, and capital letters represent matrices. Finite sets containing ι elements are compactly denoted by $[\![\iota]\!]$, defined as $\{1, \cdots, \iota\}$. The transpose operation of a matrix or vector is denoted by "$^\intercal$", and the trace of a matrix (or vector) is represented by $\mathrm{tr}(\bullet)$.

We begin by introducing the fundamentals of our conceptual framework. Subsequently, in Section 2.1, we present an overview of the framework for model-based optimal design of experiments (ODOE). Proceeding to Section 2.2, we delve into the foundational principles of Semidefinite Programming (SDP), while Section 2.3 is dedicated to the fundamentals of Surrogate Based Optimization (SBO).

We consider a general univariate nonlinear model represented by the equation:

$$y = f(x, \boldsymbol{\theta}) + \epsilon, \tag{1}$$

where y denotes the response, x represents a regression factor, $f(\bullet)$ is a twice-differentiable function, $\boldsymbol{\theta}$ is the vector of parameters to be estimated from experiments, and $\epsilon \approx \mathcal{N}(0, \sigma)$ represents observational noise with zero mean and standard deviation σ. The number of parameters in the model is denoted by n_θ, with each parameter θ_i (where $i \in [\![n_\theta]\!]$) constrained within a compact domain $[\theta_i^L, \theta_i^U]$. The Cartesian domain containing all parameter combinations for the model is denoted as $\Theta \equiv \otimes_{i=1}^{n_\theta} [\theta_i^L, \theta_i^U] \subset \mathbb{R}^{n_\theta}$. To distinguish between the generic vector $\boldsymbol{\theta}$ and a singleton vector representing parameters in Θ, the latter is denoted as $\mathbf{p}$. Furthermore, let $x \in \mathbf{X} \subset \mathbb{R}$, where $\mathbf{X}$ represents the closed domain containing all potential values of the regressors, henceforth referred to as the design space. Model (1) is presented as univariate for simplicity, but it can be extended to the multivariate case. For clarity, this paper focuses on univariate models to simplify the formulation.

Employing SDP to optimize experimental designs requires discretizing the design space and formulating a mathematical program to determine optimal weights for each candidate point. To achieve this, a uniformly spaced grid is utilized for discretization, with Δx being the step size. Consequently, the continuous design space $\mathbf{X}$ for the regressor is approximated by a finite discrete set of candidate points, denoted as $\mathbb{X}^{[\![n_x]\!]}$. Here, $n_x = 1 + \lceil (x^U - x^L)/\Delta x \rceil$, where x^U and x^L respectively denote the upper and lower bounds of $\mathbf{X}$. After discretizing the design space, local Fisher Information Matrices (FIMs) are constructed at each candidate point. In the case of nonlinear models, the first-order approximation of the model used to construct the FIMs is computed at $\mathbf{p}$.

The global Fisher Information Matrix (FIM) corresponding to the model represented by Equation (1), evaluated at a specific singleton parameter vector $\mathbf{p} \in \Theta$, is expressed as:

$$\begin{aligned} \mathcal{M}(\mathbf{x}, \mathbf{p}) &= -\mathbb{E}\left[\frac{\partial}{\partial \boldsymbol{\theta}}\left(\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\intercal}\right)\right] = \sum_{j=1}^{n_x} w_j \cdot M(x_j|\mathbf{p}) = \\ &= \sum_{j=1}^{n_x} w_j \cdot h_i^\intercal(x_j|\mathbf{p}) \cdot h_i(x_j|\mathbf{p}), \end{aligned} \tag{2}$$

where $\mathbf{w}$ represents the vector of weights assigned to the support points in the design, n_x denotes the user-defined number of discrete candidate points, $M(x_j|\mathbf{p})$ represents the elemental Fisher Information Matrix (FIM) at x_j, $\mathbb{E}[\bullet]$ denotes the expectation, $h(x_j|\mathbf{p}) \in \mathbb{R}^{n_\theta}$ denotes the vector of parametric derivatives at $x_j \in \mathbb{X}^{[\![n_x]\!]}$, and $\mathcal{L}(\bullet)$ is the log-likelihood.

In the context of model fitting, the likelihood represents the joint probability of the observed data, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, given the model parameters $\boldsymbol{\theta}$, where n is the number of experimental points used for regression. The log-likelihood is then defined as the natural logarithm of the likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}).$$

When the response variable y follows a normal distribution with mean $f(\mathbf{x}_i, \boldsymbol{\theta})$ and variance σ^2, the log-likelihood simplifies to:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}[y_i - f(\mathbf{x}_i, \boldsymbol{\theta})]^2 .$$

2.1 Optimal design of experiments

There exist two distinct design categories: (i) continuous designs, also referred to as *approximate designs*; and (ii) discrete designs, known as *exact designs*. In both paradigms, the allocation of effort at each experimental point x_j, quantified by the weight (w_j), is defined as $w_j = n_j/N$, where n_j represents the number of replications conducted at the j^{th} experimental point, and N denotes the total number of observations in the experimental plan. For approximate designs, all n_j values are constrained to be non-negative real numbers. Conversely, exact designs necessitate that n_j be non-negative integers across all experimental points.

In both scenarios, the weights constitute a discrete probability measure, satisfying $\sum_{j=1}^{K} w_j = 1$; K is the number of support points of the design. For continuous designs, these weights are confined within the interval [0, 1], while for discrete designs, they are restricted to the set of rational numbers within [0, 1]. Notably, this set is finite for a given N. The optimization problem for finding approximate optimal designs demonstrates convexity (or can be reformulated as such) when the design criterion is a convex (or concave) function of the Fisher Information Matrix (FIM). Consequently, global optimality is guaranteed, as supported by equivalence theorems (Kiefer, 1974; Pukelsheim, 1993), thereby enabling the utilization of tailored optimization algorithms (Vandenberghe and Boyd, 1999). In this work, our focus lies on approximate designs, represented by K-point tuples:

$$\xi = \begin{pmatrix} x_1 & x_2 & \cdots & x_{K-1} & x_K \\ w_1 & w_2 & \cdots & w_{K-1} & w_K \end{pmatrix}.$$

In this representation, the upper row elements are for factor levels, which serve as pivotal support points within our context, while the lower row elements are for the corresponding weights. In practical terms, this corresponds to the set of candidate points where $w_j > \delta,\ \ j \in [\![n_x]\!]$, with δ representing a small positive constant.

For independent, identically, and normally distributed observational errors, the volume of the confidence region for $\hat{\boldsymbol{\theta}}$ is inversely proportional to $\sqrt{\det(\mathcal{M})}$. Therefore, choosing a design that minimizes the determinant of the inverse of the Fisher Information Matrix (or maximizes the determinant of the FIM) yields the most accurate parameter estimates, a criterion known as D–optimality. Other optimization measures for the parametric confidence region include: (i) the sum of the ellipsoid's diagonals, quantified by the inverse of the trace of the FIM, leading to the A–optimality criterion;

(ii) the smallest eigenvalue of the FIM, representing the lower diagonal of the ellipsoid, giving rise to the E–optimality criterion; and (iii) the condition number of the FIM, indicating the ratio between the largest and smallest ellipsoid diagonals, leading to the K-optimality criterion. The respective optimal experimental design problems are formulated as follows:

$$\xi_{\mathrm{D}} = \arg\max_{\xi \in \Xi} \{\det[\mathcal{M}(\mathbf{x}, \mathbf{p})]\}^{1/n_\theta} \tag{3a}$$

$$\xi_{\mathrm{A}} = \arg\min_{\xi \in \Xi} \mathrm{tr}[\mathcal{M}^{-1}(\mathbf{x}, \mathbf{p})] \tag{3b}$$

$$\xi_{\mathrm{E}} = \arg\max_{\xi \in \Xi} \lambda_{\min}[\mathcal{M}(\mathbf{x}, \mathbf{p})] \tag{3c}$$

$$\xi_{\mathrm{K}} = \arg\min_{\xi \in \Xi} \kappa[\mathcal{M}(\mathbf{x}, \mathbf{p})] \tag{3d}$$

Here, Ξ represents the set of feasible designs in $\Theta \times \Sigma$, where Σ is the $(K-1)$-point simplex defined as $\{w_k : w_k \geq 0,\ \sum_{k=1}^{K} w_k = 1,\ k \in [\![K]\!]\}$. Further, $\lambda_{\min}[\bullet]$ denotes the minimum eigenvalue of the argument, and $\kappa[\bullet]$ denotes the condition number.

The expressions (3a-3d) can be formulated as Semidefinite Programs (SDPs), as demonstrated in Vandenberghe and Boyd (1999), Ye and Zhou (2013), and Boyd and Vandenberghe (2004). Specifically, the objective functions can be rewritten as a set of Linear Matrix Inequalities (LMIs), placing them within the framework of SDP-a specialized subclass of convex optimization. A key advantage of SDP is its compatibility with dedicated polynomial-time algorithms, ensuring efficient solutions. For a more in-depth discussion, see Section 2.2. A Linear Matrix Inequality (LMI) is a fundamental constraint in optimization and control theory, extending scalar linear inequalities to matrix variables. LMIs generally take the following form:

$$\max_{\boldsymbol{\zeta}} \left\{ \mathbf{d}^{\intercal}\boldsymbol{\zeta} \;\middle|\; \sum_{i=1}^{m_1} \zeta_i M_i - M_0 \succeq 0 \right\}, \tag{4}$$

where $\boldsymbol{\zeta}$ is the decision variable vector, $\mathbf{d}$ is a given vector, and M_0, M_i are symmetric matrices. The notation $\succeq 0$ signifies that the resulting matrix is positive semidefinite.

In this study, we employ design efficiency as a metric to balance, on a scale of [0, 1], the information yield from various criteria, facilitating their comparison. Following Equation (3), the efficiency of a given design ξ relative to a reference design ξ^*, characterized by the Fisher Information Matrix $\mathcal{M}^*(\mathbf{x}, \mathbf{p})$, is given by:

$$\eta_{\mathrm{D}} = \left\{ \frac{\det[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\det[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} \right\}^{1/n_\theta} \tag{5a}$$

$$\eta_{\mathrm{A}} = \frac{\mathrm{tr}[\mathcal{M}^{*-1}(\mathbf{x}^*, \mathbf{p})]}{\mathrm{tr}[\mathcal{M}^{-1}(\mathbf{x}, \mathbf{p})]} \tag{5b}$$

$$\eta_{\mathrm{E}} = \frac{\lambda_{\min}[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\lambda_{\min}[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} \tag{5c}$$

$$\eta_{\mathrm{K}} = \frac{\kappa[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]}{\kappa[\mathcal{M}(\mathbf{x}, \mathbf{p})]}. \tag{5d}$$

In this context, the subscript of η denotes the specific optimality criterion under consideration. The reference design ξ^* is determined by optimizing a single optimality criterion. The superscript "*" stands for the single-criterion optimal design. The efficiency constitutes a relative measure of information per observation. In practical terms, it enables the comparison of designs across diverse criteria, facilitating the transformation of a multicriteria problem into a single-criterion program, where the objective function comprises a convex combination of the original criteria set (Hwang and Masud, 2012).

2.2 Semidefinite programming

In this Section, we introduce the foundational principles of convex optimization, specifically tailored to tackle optimal design of experiments problems within the discrete design domain $\mathbb{X}^{[\![n_x]\!]}$, which comprises n_x candidate experimental points.

Let $\mathbb{S}_+^{n_\theta}$ represent the space of $n_\theta \times n_\theta$ symmetric positive semidefinite matrices, and $\mathbb{S}^{n_\theta}$ denote the space of $n_\theta \times n_\theta$ symmetric matrices. A convex set $\mathbf{S} \in \mathbb{R}^{n_\theta}$ is said to be semidefinite representable (SDr) if, for every $\boldsymbol{\zeta} \in \mathbf{S}$, the projection $\mathrm{proj}_{\mathbf{S}^{\mathrm{exp}}}(\boldsymbol{\zeta})$ onto a higher-dimensional set $\mathbf{S}^{\mathrm{exp}}$ can be expressed using Linear Matrix Inequalities (LMIs).

A convex (or concave) function $\varphi : \mathbb{R}^{m_1} \mapsto \mathbb{R}$ is considered semidefinite representable (SDr) if and only if its epigraph, denoted by $\{(t, \boldsymbol{\zeta}) : \varphi(\boldsymbol{\zeta}) \leq t\}$, or its hypograph, denoted by $\{(t, \boldsymbol{\zeta}) : \varphi(\boldsymbol{\zeta}) \geq t\}$, respectively, are SDr and can be expressed using Linear Matrix Inequalities (LMIs) (Ben-Tal and Nemirovski, 2001; Boyd and Vandenberghe, 2004). Consequently, the optimal values $\boldsymbol{\zeta}$ of SDr functions are formulated as semidefinite programs with the general form (4); i.e. expressed as LMI.

In our design framework, the vector $\mathbf{d}$ comprises known constants that are specific to the design problem, while the positive semidefinite matrices M_i, where $i \in 0, \cdots, m_1$, encapsulate local Fisher Information Matrices (FIMs) and other matrices derived from the transformation of the functions $\varphi(\boldsymbol{\zeta})$ into LMIs. The decision variables within the vector $\boldsymbol{\zeta}$ correspond to the weights w_i for $i \in [\![n_x]\!]$, representing the optimal design, along with any auxiliary variables necessary. The task of determining a design for a predetermined set of candidate experiments $\mathbb{X}^{[\![n_x]\!]}$, consisting of points $\mathbf{x}_i$ for all $i \in [\![n_x]\!]$, is addressed by employing formulation (4), complemented by the following linear constraints on $\mathbf{w}$: (i) $\mathbf{w} \geq 0$, and (ii) $\mathbf{1}_{n_x}^\intercal \cdot \mathbf{w} = 1$, where $\mathbf{1}_{n_x}^\intercal$ denotes a unit column vector with n_x entries. The problem described by (4) constitutes the classic SDP problem, incorporating LMIs that represent conic constraints.

Ben-Tal and Nemirovski (2001) offer a comprehensive collection of semidefinite representable (SDr) functions, used for tackling continuous optimal design design problems through SDP formulations, as discussed in Boyd and Vandenberghe (2004, §7.3). The A–, D– and E–optimality criteria are special cases of a general family of criteria introduced by Kiefer (1974) that is indexed by a parameter ν. A–optimality corresponds to $\nu = -1$, E–optimality to $\nu \to -\infty$, and D–optimality to $\nu \to 0$.

More recently, Sagnol (2013) demonstrated that each criterion within this class is SDr for all rational values of $\nu \in (-\infty, 0)$, and that Semidefinite Programming formulations can be developed to address them. The determination of optimal approximate experimental designs under common convex (or concave) criteria can be formulated as a Semidefinite Programming (SDP) problem, aligning with the general framework of (4) and incorporating constraints on $\mathbf{w}$, as discussed in Vandenberghe and Boyd (1999). Notably, D– and E–optimality criteria correspond to concave functions, while A–optimality is convex. The K–optimality criterion, being quasi-convex (Agrawal and Boyd, 2020), can be reformulated as a convex problem on an α-sublevel set.

The methodology employed for single criterion designs can seamlessly extend to compound designs, provided that the objective function incorporates all the considered criteria. Both the geometric and arithmetic means of design efficiencies concerning various criteria can be used to measure the performance relative to single criteria, as they also manifest as semidefinite representable (SDr) functions. However, it is important to note that handling combinations of concave-concave criteria necessitates distinct reformulation approaches compared to other combinations of criteria.

2.3 Surrogate based optimizaton

In this Section, we delve into the fundamentals of Surrogate-Based Optimization (SBO). Falling within the realm of polynomial response surface methods, SBO is particularly suitable for tackling problems characterized by complex and black-box functions, denoted as $r(\mathbf{x})$. In such scenarios, the expense associated with fitting and evaluating the surrogate model pales in comparison to that of function evaluation, especially considering the absence of algebraic expressions for either the gradient or the Hessian matrix (Bhosekar and Ierapetritou, 2018; Kim and Boukouvala, 2020).

The SBO methodology unfolds across three consecutive stages: (i) Simulation of the Real Model: This stage involves the emulation of the "real (complex) model", which may or may not be a black box model, using a limited set of judiciously chosen data points; (ii) Construction of an Approximate Model: Here, an "approximate model" with the form of a response surface model is fitted based on the data generated in the preceding stage; (iii) Optimization of the Approximate Model: The constructed surrogate model is then optimized to yield a fresh set of points that faithfully mimic the behavior of the "real model" while significantly expediting computation. This iterative process continues until the response of $f(\mathbf{x})$ converges satisfactorily to a complex (or black box) model $g(\mathbf{x})$ in a closed domain defined by constraints $\mathbf{r}(\mathbf{x})$ at a given point $\mathbf{x}$ (Müller and Woodbury, 2017).

The optimization problems considered by SBO are compactly represented as follows:

$$\min_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}) \tag{6a}$$

$$\text{s.t. } \mathbf{r}(\mathbf{x}) \leq 0 \tag{6b}$$

Here, $f(\bullet)$ represents the computationally inexpensive objective function that serves as an approximation to the more intricate function subject to a set of constraints $r(\mathbf{x})$.

Equation (6b) defines the collection of computationally intensive black-box inequality constraints, while $\mathbf{X}$ denotes the finite domain of decision variables.

The surrogate model is constructed from an initial set of simulations, generated in accordance with a sampling plan. Among the techniques employed for generating these initial sampling points, Latin Hypercube (LHC) designs stand out as particularly common (Müller and Day, 2019). As for the surrogate models, denoted as $f(\bullet)$, interpolating models hold sway, with kriging (Martin and Simpson, 2005) and Radial Basis Functions (RBFs) (Powell, 1992; Buhmann, 2009) emerging as the most prevalent options. These models have found widespread application in optimizing problems featuring computationally intensive objective functions, as demonstrated in the work of Müller et al. (2013).

The algorithm comprises a series of steps wherein a merit function, integrating both the optimality measured by the surrogate function and the distance from existing points, assumes pivotal importance. Essentially, this merit function balances between exploration – filling the voids between existing sample points by exploring various regions within the optimization domain – and exploitation – using available sample points to pinpoint an optimum (Regis and Shoemaker, 2007). Alizadeh et al. (2020) provide a recent review of the application of surrogate models in optimization. Various tools for surrogate optimization are currently available; among them are the works of Müller and Woodbury (2017), Erikson et al. (2019), Le Digabel (2011), Müller (2014), and Müller (2016). In Section 3, we adopt the algorithm proposed by Regis and Shoemaker (2007), which, in turn, employs a cubic Radial Basis Function (RBF) with a linear tail as the surrogate model (Gutmann, 2001).

3 SDP-based formulations for finding compound designs

This Section presents the formulations for computing locally optimal compound designs through Semidefinite Programming. It is worth noting that while SDP guarantees finding the global optimum within a grid of discrete candidate points, it can pose computational challenges with a large number of candidate experiments. The Semidefinite Programming formulations for all single criteria described in Equation (3) follow the general structure depicted by Equation (4). These formulations, considered state–of-the-art and can be found in the works cited in Sect. 2.2.

Utilizing SDP-based formulations to determine optimal experimental designs involves the following initial steps:

1. Discretization of the design space: Initially, the design space is discretized using a uniformly spaced grid, as detailed in Section 2.
2. Construction of local Fisher Information Matrices: Through Equation (2), local Fisher Information Matrices are systematically generated at specified candidate points.
3. Solution of optimal design problems for single criteria: The optimal design problem is solved for each single criterion, and the obtained optima are stored.

4. Combination of optima: The optima obtained in Step 3 are combined to derive a measure of combined efficiency, typically through arithmetic or geometric mean functions.
5. Problem resolution with new objective function: Subsequently, the problem is resolved with the new objective function within the same discretized domain.

The optimal designs derived in Step 3 (as described above) are pivotal in establishing a standardized weighting for each criterion within a composite design formulation, characterized by a convex (or concave) combination of measures denoted as $\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})]$, serving as the objective function. Here, $\Phi_{\text{cr}}[\bullet]$ represents an optimality criterion of interest, with cr $\in$ {D-, A-, E-, K-}, indicating the extent of information captured about the parameters of the specific model, relative to a specified measure of the parametric confidence ellipsoid. Given the diverse range of $\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})]$ values across different criteria, standardization becomes imperative. Reference designs, ξ^*_{cr}, are employed to normalize the information measures. This ensures comparability and uniform expression of the relative importance of each criterion within the objective function, confined to the unitary interval [0, 1]. The metrics assessing the relative information extracted for parametrizing each model are quantified as efficiencies, as expressed in Equation (5).

The objective of this formulation is to identify a set of support points that collectively maximize efficiency across pairs of criteria. This approach ensures that the final design avoids undesirable properties of prediction variances by evenly distributing the loss of efficiency among all criteria. We explore two alternative objective functions for aggregating single-criterion efficiencies in compound design formulations: (i) the geometric mean, referred to as the *Geomt* formulation in subsequent sections; and (ii) the arithmetic mean, referred to as the *Arthm* formulation.

Some criteria are represented by convex functions, while others are concave, yet all are SDr. Their reformulation must adhere to the rules of Disciplined Convex Programming (DCP), which ensure that convex optimization problems remain well-posed and solvable. These rules specify how functions and expressions can be combined to preserve convexity, especially within the set of symmetric positive semidefinite matrices (Nesterov and Nemirovski, 1994; Vandenberghe and Boyd, 1996, 1999; Boyd and Vandenberghe, 2004; Agrawal and Boyd, 2020; MOSEK, 2024). To construct solvable SDP formulations, following DCP is crucial. One key principle is that a convex function expressed as an LMI has a convex epigraph, which allows its maximum to be minimized. Conversely, a concave function has a concave hypograph, allowing its minimum be maximized. Since the treatment of compound designs depends on whether they combine two concave criteria or a mix of convex and concave criteria, we analyze these cases separately. Sect. 3.1 covers concave-concave pairs (e.g., DE–optimality), while Sect. 3.2 examines convex-concave pairs (e.g., DA–optimality).

3.1 Formulation for combinations of concave-concave criteria

Let $\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})]$ represent the combined criteria, where cr is restricted to concave criteria. Following the DCP rules require maximizing the hypograph. Semidefinite

representability allows us to express these criteria using a set of LMIs, forming a convex set known as the hypograph (the set of points below the function). Formally,

$$\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})] \geq \varrho_{\text{cr}}, \tag{7}$$

where ϱ_{cr} is the minimum of the hypograph. To normalize the function, we use the optimal values from previously computed single-criterion designs, $\Phi_{\text{cr}}[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]$. This leads to a normalized efficiency measure bounded from below:

$$\frac{\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\Phi_{\text{cr}}[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} = \eta_{\text{cr}} \geq v_{\text{cr}}, \tag{8}$$

where v_{cr} is the lower bound on cr-optimal design efficiency. Since Equation (8) is SDr, it can be directly incorporated as a constraint in the Semidefinite Program.

We now focus on the *Geomt* formulation, where we aggregate n_{cr} concave criteria. Notably, the geometric mean is an SDr concave function, while the arithmetic mean is affine and requires convexified arguments, specifically the minima of concave functions (Ben-Tal and Nemirovski, 2001). This condition holds for the hypographs v_{cr}, $\forall$cr. To generalize the formulation of the objective function, we introduce the function $J(\mathbf{v})$, which depends on the vector of hypograph values. The function $J(\mathbf{v})$ can take the following forms: (i) *Geometric mean*: $J(\mathbf{v}) = \prod_{\text{cr}\in\mathcal{C}} v_{\text{cr}}^{1/n_{\text{cr}}}$; (ii) *Arithmetic mean*: $J(\mathbf{v}) = \sum_{\text{cr}\in\mathcal{C}} v_{\text{cr}}/n_{\text{cr}}$. Thus, the general formulation for this scenario is as follows:

$$\max_{\mathbf{w},\mathbf{v}} \quad J(\mathbf{v}) \tag{9a}$$

$$\text{s.t.} \quad \mathcal{M}(\mathbf{x}, \mathbf{p}) = \sum_{j=1}^{n_x} w_j \; h^{\intercal}(x_j|\mathbf{p}) \; h_i(x_j|\mathbf{p}) \tag{9b}$$

$$\frac{\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\Phi_{\text{cr}}[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} \geq v_{\text{cr}}, \quad \text{cr} \in [\![n_{\text{cr}}]\!] \tag{9c}$$

$$\sum_{i=1}^{n_x} w_j = 1 \tag{9d}$$

$$w_j \in [0,\ 1], \quad j \in [\![n_x]\!] \tag{9e}$$

$$\mathbf{v} \in [0,\ 1] \tag{9f}$$

Here, Equation (9a) represents the geometric mean of the standardized criteria, Equation (9b) computes the global FIM, Equation (9c) establishes the upper bound of the hypographs of the standardized criteria, Equation (9d) ensures that the weights sum to 1, Equation (9e) imposes their non-negativity, and Equation (9f) sets the domain of $\mathbf{v}$. The reference designs ξ^* have been previously determined, allowing the setting of the standardizing constant $\Phi_{\text{cr}}[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]$. Finally, $\mathcal{C}$ is the set of criteria aggregated in the compound criterion.

The formulation can be extended to handle more than two concave criteria. When $n_{\text{cr}} = 2$ and $\text{cr} \in \{\text{D–}, \text{E–}\}$, it takes the form:

$$\max_{\mathbf{w},\mathbf{v}} \quad (v_{\text{D}} \cdot v_{\text{E}})^{1/2} \tag{10a}$$

$$\text{s.t.} \quad \text{Equations } (9b - 9f) \tag{10b}$$

Here, the ratios

$$\frac{\Phi_{\text{cr}}[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\Phi_{\text{cr}}[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]}$$

are replaced by efficiencies from Equations (5), specifically for the D– and E–optimality criteria. Finally, the objective function is the geometric mean, a concave function under DCP rules, ensuring that we maximize the smallest value in the hypograph set.

When we consider the *Arthm* formulation, the structures resemble those presented in Equation (9) and Equation (10). The main modification concerns the objective function, specifically the form of $J(\mathbf{v})$.

3.2 Formulation for combinations of convex-concave criteria

We now delve into the combinations of pairs of convex-concave criteria, exemplified by the DA-optimality criterion. When employing the geometric mean for aggregation, the objective function for $n_{\text{cr}} = 2$ (comprising one convex and one concave criterion) is expressed as:

$$\prod_{\text{cr}\in\mathcal{C}} \eta_{\text{cr}}^{1/n_{\text{cr}}} = \left\{ \frac{\Phi_1[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\Phi_1[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} \cdot \frac{\Phi_2[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]}{\Phi_2[\mathcal{M}(\mathbf{x}, \mathbf{p})]} \right\}^{1/n_{\text{cr}}}. \tag{11}$$

The second term of the product exhibits a convex hypograph, stemming from the reciprocal of a convex function. Consequently, the arguments within the geometric mean are all convex. To simplify, let

$$\alpha = \frac{\Phi_2[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]}{\Phi_1[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]},$$

be a constant and let $z_1 = \Phi_1[\mathcal{M}(\mathbf{x}, \mathbf{p})]$ and $z_2 = \Phi_2[\mathcal{M}(\mathbf{x}, \mathbf{p})]$ be decision variables. Thus, the optimization problem becomes:

$$\max_{\mathbf{w}} \alpha \cdot z_1(\mathbf{w}) \cdot \frac{1}{z_2(\mathbf{w})}. \tag{12}$$

However, despite the semidefinite representability of z_2, its reciprocal is not SDr. Therefore, an alternative approach must be adopted to handle combinations of convex-concave criteria.

To tackle this problem, we consider the Dinkelbach (1967) transformation. This transformation reformulates the single-ratio problem (12) into the following form:

$$\max_{\mathbf{w},r} \alpha \cdot z_1(\mathbf{w}) - r \cdot z_2(\mathbf{w}), \tag{13}$$

where r is a newly introduced variable representing $\alpha \cdot z_1 \cdot (1/z_2)$, capturing the combined efficiency. Notice that problem (13) is nonlinear due to the product of variables in the second term. Consequently, it cannot be formulated as a convex problem under the DCP rules. To address this, we reformulate the problem as a bilevel program and adopt an algorithm inspired by Crouzeix et al. (2008) to solve it. The outer problem maximizes the combined efficiency, r, while the inner problem solves a semidefinite program (SDP) for a fixed value of r. To circumvent the need for computing gradient and Hessian information, the outer solver is designed as a stochastic optimization procedure, ensuring a systematic search for the optimal solution. It can be demonstrated that convergence is assured by iteratively updating r and solving the nonlinear problem (13). In particular, when the single-ratio problem (12) is a concave-convex combination of functions, optimizing $\mathbf{z}$ in (13) while keeping r fixed transforms the problem into a convex one. Consequently, the iterative algorithm converges to the global optimum solution of (12).

The reformulated problem (13) is as follows:

$$\begin{aligned}
\max_{r} \quad & r && && \text{(14a)}\\
\text{s.t.} \quad & \max_{\mathbf{w}} && \alpha \cdot z_1(\mathbf{w}) - r \cdot z_2(\mathbf{w}) && \text{(14b)}\\
& r \in [0, 1]. && && \text{(14c)}
\end{aligned}$$

Here, (14a) represents the objective function of the outer level problem, (14b) denotes the reformulated convex problem solved for fixed r values, and (14c) establishes the domain of r. To systematically address problem (14), we employ a SBO-based solver (see Section 2.3) to handle the outer level program, iteratively converging r, and a SDP-based solver (see Section 2.2) for the inner program. The SBO solver iteratively adjusts r from various initial solutions to construct a response surface. For each iteration, the corresponding SDP problem is solved. Then, the surface of r values is exploited and the optimum found. This architecture is well-suited for SBO, given the complexity of the inner program's objective and the challenges associated with computing its gradient and Hessian matrices. It's worth noting that the value of r at convergence represents the combined efficiency.

We now apply the bilevel reformulation (14) to a general case involving two criteria, one concave (Criterion 1) and one convex (Criterion 2). The problem transforms into:

$$\begin{aligned}
\max_{r} \quad & r && && \text{(15a)}\\
\text{s.t.} \quad & \max_{\mathbf{w},\mathbf{v}} && J(\mathbf{v}, r) && \text{(15b)}
\end{aligned}$$

$$\text{s.t.} \quad \mathcal{M}(\mathbf{x}, \mathbf{p}) = \sum_{j=1}^{n_x} w_j \; h^{\intercal}(x_j|\mathbf{p}) \; h_i(x_j|\mathbf{p}) \tag{15c}$$

$$\frac{\Phi_1[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\Phi_1[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} \geq v_1 \tag{15d}$$

$$\frac{\Phi_2[\mathcal{M}(\mathbf{x}, \mathbf{p})]}{\Phi_2[\mathcal{M}^*(\mathbf{x}^*, \mathbf{p})]} \leq v_2 \tag{15e}$$

$$\sum_{i=1}^{n_x} w_j = 1 \tag{15f}$$

$$w_j \in [0,\ 1], \quad j \in [\![n_x]\!] \tag{15g}$$

$$v_1 \in [0,\ 1],\ v_2 \geq 1 \tag{15h}$$

$$r \in [0, 1]. \tag{15i}$$

Equations (15c-15g) resemble those in (9). Equation (15a) sets the objective of the outer program, (15b) represents the reformulated objective function of the inner problem, (15h) bounds the vector $\mathbf{v}$, and (15i) bounds the domain of r. For the *Geomt* formulation, the objective function is given by $J(\mathbf{v}, r) = v_1 - r \cdot v_2$, where r is a fixed parameter in the inner problem, but is treated as a variable to be optimized in the outer problem. In the *Arthm* formulation, the objective is $J(\mathbf{v}, r, t) = v_1 + t - r$, where $t \in [0, 1]$ is a new decision variable from the semidefinite constraint:

$$\begin{pmatrix} v_2 & 1 \\ 1 & t \end{pmatrix} \succeq 0,$$

with $\succeq$ indicating positive semidefiniteness.

In our study, we tackled SDP problems using the `cvx` environment coupled with the `Mosek` solver, renowned for its efficient Interior Point algorithm (Ye, 1997). To ensure precision in our computations, we set both the relative and absolute tolerances for solving the SDP problem to 1×10^{-5}.

For the SBO solver, we initiated our search with an initial sample comprising $\max(20,\ 2^K)$ points, generated using a Latin Hypercube (LHC) sampling algorithm within the domain of interest. Our numerical solution incorporated two stopping criteria: (i) reaching the maximum function evaluations, fixed at 100 for all problems addressed; and (ii) meeting the tolerance of the objective function. Here, we set the absolute and relative tolerances to 1×10^{-6} and 1×10^{-7}, respectively, with a requirement of 50 consecutive function evaluations to trigger termination.

All computations presented in this paper were performed on a 64-bit Windows 10 operating system, utilizing an Intel Core i7 processor clocked at 3.40 GHz.

4 Numerical results

In this Section, we apply the formulations outlined in Section 3 to determine compound designs falling under the categories of concave-concave and convex-concave

Table 1 Models considered in the analysis. $B(\theta_1, \theta_2) = (\theta_1 + \theta_2)^{\theta_1+\theta_2}/(\theta_1^{\theta_1}\ \theta_2^{\theta_2})$

Model	Regression function ($\mathbb{E}(y)$)	n_θ	**X**	**p**
1	$\theta_1 + \theta_2\, x + \theta_3\, x^2$	3	$[-1, +1]$	$(1.0,\ 0.4,\ -1.39)$
2	$\theta_1 + \theta_2 B(\theta_2, \theta_3) \left(\frac{x}{\theta_5}\right)^{\theta_3} \left(1 - \frac{x}{\theta_5}\right)^{\theta_4}$	5	$[0, 15]$	$(0.0,\ 0.4,\ 1.39,\ 1.39,\ 10.0)$

criteria. Firstly, in Section 4.1, we construct single-criterion optimal designs, crucial for computing efficiencies. Subsequently, in Section 4.2, we obtain compound optimal designs for a pair of concave criteria, the example being DE–optimality. Finally, in Section 4.3, we employ the algorithm to compute optimal designs for a pair of concave-convex criteria where DA–optimality is our example of interest.

Table 1 outlines the two models utilized for demonstration purposes. Model 1 is linear concerning its parameters, enabling the determination of globally optimal designs. Conversely, Model 2 is a 5-parameter non-linear representational form of the Beta class, and we compute locally optimal designs. The third column of Table 1 denotes the number of parameters in each model (n_θ), while the values of these parameters used for computing locally optimal designs (**p**) are specified in the fifth column. The design space is detailed in the fourth column (**X**). Model 1 has undergone extensive study, leading to the evaluation of the effect of criteria on the weights assigned to each support point, which are independent of the criteria. In contrast, for Model 2, optimal designs may simultaneously have distinct support points and weights.

In all calculations the tolerance parameter δ in Section 2.1, which ensures a minimum weight for all support points of the design, is set equal to 1×10^{-4}.

4.1 Reference optimal designs

This Section presents the optimal designs obtained for each individual optimality criterion corresponding to the models listed in Table 1. The single criterion SDP problems were solved with the computational framework described at the conclusion of Section 3. In both cases we use $n_x = 401$. Consequently, for Model 1, we set $\Delta x = 0.005$, while for Model 2, Δx was set to 0.0375 (see the design spaces in Table 1). The results are summarized in Table 2. The upper panel is for Model 1, while the lower panel relates to Model 2. Optimal designs are presented in the second column (as a tuple), accompanied by the corresponding optimum values in the third column and the CPU time in the fourth column.

As expected, for Model 1, the support points remain consistent across all criteria, with only the weights varying. Each design comprises three support points. Conversely, designs for Model 2 incorporate either five or six support points, with changes in both the location of the support points and their corresponding weights. In practice, if a finer grid is employed, contiguous support points of ξ_D and ξ_K could be collapsed into a single point. However, this consolidation is likely to have minimal impact on the optimum and would not significantly affect the design efficiency. Across all cases, the computational requirements remain modest.

Table 2 Optimal designs for models in Table 1 ($n_x = 401$)

Model 1			
Criterion	Optimal design	Optimum	CPU (s)
D-	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.3334 & 0.3333 & 0.3334 \end{pmatrix}$	0.5291	1.81
A-	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.2500 & 0.5000 & 0.2500 \end{pmatrix}$	8.0000	0.98
E-	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.2000 & 0.6000 & 0.2000 \end{pmatrix}$	0.2000	1.12
K-	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.1666 & 0.6667 & 0.1667 \end{pmatrix}$	5.8242	0.98
Model 2			
Criterion	Optimal design	Optimum	CPU (s)
D-	$\begin{pmatrix} 0.0000 & 1.5250 & 4.6250 & 8.0500 & 8.0750 & 10.0000 \\ 0.2000 & 0.1999 & 0.2000 & 0.1455 & 0.0545 & 0.2000 \end{pmatrix}$	4.185×10^{-3}	2.92
A-	$\begin{pmatrix} 0.0000 & 1.2500 & 4.6500 & 8.3750 & 10.0000 \\ 0.0884 & 0.1899 & 0.2101 & 0.3097 & 0.2019 \end{pmatrix}$	$7.211 \times 10^{+5}$	1.21
E-	$\begin{pmatrix} 0.0000 & 1.2500 & 4.6500 & 8.3750 & 10.0000 \\ 0.0877 & 0.1899 & 0.2103 & 0.3102 & 0.2019 \end{pmatrix}$	1.393×10^{-6}	0.91
K-	$\begin{pmatrix} 0.0000 & 1.0500 & 1.0750 & 4.3000 & 8.3000 & 10.0000 \\ 0.1540 & 0.0584 & 0.2123 & 0.1842 & 0.2308 & 0.1602 \end{pmatrix}$	$1.021 \times 10^{+6}$	1.22

Table 3 presents the efficiency matrices of optimal designs obtained for specific criteria (D–, A–, E–, and K–) when applied to different criteria. The reference designs are those outlined in Table 1, with Equation (5) employed for computation. Specifically, Table 3a corresponds to Model 1, while Table 3b corresponds to Model 2. In virtually all cases, as would be expected, a decrease in efficiency is observed when an optimal design is utilized for a criterion different from the one it was originally designed for. Mitigating these losses is the primary objective of compound optimal designs in the next sections. The exception is the efficiency of the design for A–optimality, ξ_{A}, when evaluated for E–optimality. Table 2 shows that, for Model 2, ξ_{A} and ξ_{E} are virtually indistinguishable.

4.2 Compound optimal designs for concave-concave pairs of criteria.

In this Section, we use the formulations outlined in Section 3.1 to compute compound optimal designs for concave-concave pairs of criteria. To illustrate this, we focus on the DE–optimality criterion and tackle the design problem using both the *Geomt* and *Arthm* formulations. To maintain consistency in our comparison, we utilize the same grid of candidate points. The results of our analyses are summarized in Table 4, structured similarly to preceding tables. While the optimal designs obtained for both formulations exhibit slight differences, they demonstrate efficient performance, requiring minimal CPU time.

Table 3 Efficiency of the optimal designs obtained with an optimality criterion when used for another criterion: ((a)) Model 1; ((b)) Model 2

(a) Model 1.

	Optimal designs (Table 2, upper panel)			
Crit.	ξ_D	ξ_A	ξ_E	ξ_K
D-	1.0000	0.8888	0.7306	0.5601
A-	0.9449	1.0000	0.9549	0.8504
E-	0.8654	0.9601	1.0000	0.9714
K-	0.7937	0.8889	0.9763	1.0000

(a) Model 2.

	Optimal designs (Table 2, lower panel)			
Crit.	ξ_D	ξ_A	ξ_E	ξ_K
D-	1.0000	0.8267	0.8252	0.8535
A-	0.9409	1.0000	0.9999	0.7689
E-	0.9400	0.9999	1.0000	0.7687
K-	0.9433	0.9404	0.9398	1.0000

In reference to Model 1, our analysis shows that the *Geomt* and *Arthm* formulations yield comparable efficiencies-0.9509 and 0.9531, respectively. The effectiveness of compound designs is demonstrated in Table 3, which compares the efficiency of single-criterion designs from Table 2 when applied to an incorrect criterion. For the *Geomt* formulation, the D-optimal design's overall efficiency, computed as the geometric mean of D- and E-optimality efficiencies, is $\sqrt{1.0000 \times 0.7306} = 0.8548$, lower than the reference value of 0.9509. Similarly, the E-optimal design achieves $\sqrt{1.0000 \times 0.8654} = 0.9303$, as detailed in Table 3a, both significantly lower than the compound designs' efficiency. For the *Arthm* formulation, the objective function is the average of two efficiencies: 0.8653 and 0.9327, closely aligned with the *Geomt* results. As anticipated, the compound design, which integrates both criteria, outperforms both single-criterion designs when evaluated under the other criterion, yielding a combined efficiency of 0.9531. This superiority of compound designs is further confirmed by the efficiency analysis in Table 5, where compound designs, when applied to single-criterion frameworks, consistently outperform their single-criterion counterparts in Table 2. This trend extends across Model 2 as well.

4.3 Compound optimal designs for convex-concave pairs of criteria

We now focus on the design of compound designs involving convex-concave criterion pairs, as discussed in Section 3.2. To illustrate this approach, we consider the DA-optimal criterion, with detailed results provided in Table 6. A significant increase in computational cost is observed, with CPU processing time approximately 100 times higher than for concave-concave criteria. The increase in computational cost is due to the SBO-based solver iteratively solving the SDP problem for different values of r, with 100 iterations imposed (see Section 3.2) to enhance the likelihood of obtaining a high-quality solution.

Table 4 DE–optimal designs for models in Table 1 ($n_x = 401$)

Model 1			
Formulation	Optimal design	Optimum	CPU (s)
Geomt	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.2408 & 0.5184 & 0.2408 \end{pmatrix}$	0.9509	1.41
Arthm	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.2396 & 0.5208 & 0.2396 \end{pmatrix}$	0.9531	1.00
Model 2			
Formulation	Optimal design	Optimum	CPU (s)
Geomt	$\begin{pmatrix} 0.0000 & 1.2500 & 1.2750 & 4.5500 & 8.2750 & 10.0000 \\ 0.1312 & 0.1247 & 0.0742 & 0.2113 & 0.2663 & 0.1924 \end{pmatrix}$	0.9711	2.80
Arthm	$\begin{pmatrix} 0.0000 & 1.2500 & 1.2750 & 4.5500 & 8.2750 & 10.0000 \\ 0.1309 & 0.1449 & 0.0540 & 0.2113 & 0.2666 & 0.1923 \end{pmatrix}$	0.9715	2.84

Table 5 Efficiency of the DE–optimal designs in Table 4 when used for the individual criteria

Model	Form. *Geomt*		Form. *Arthm*	
	Eff_D	Eff_E	Eff_D	Eff_E
1	0.9328	0.9694	0.9311	0.9711
2	0.9629	0.9793	0.9625	0.9798

Table 6 DA–optimal designs for models in Table 1 ($n_x = 401$)

Model 1			
Formulation	Optimal design	Optimum	CPU (s)
Geomt	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.2820 & 0.4361 & 0.2820 \end{pmatrix}$	0.9624	164.02
Arthm	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.2657 & 0.4687 & 0.2657 \end{pmatrix}$	0.9812	163.28
Model 2			
Formulation	Optimal design	Optimum	CPU (s)
Geomt	$\begin{pmatrix} 0.0000 & 1.2750 & 4.5250 & 4.5500 & 8.2750 & 10.0000 \\ 0.1366 & 0.1984 & 0.0031 & 0.2074 & 0.2617 & 0.1929 \end{pmatrix}$	0.9437	160.26
Arthm	$\begin{pmatrix} 0.0000 & 1.2500 & 4.6000 & 8.3250 & 10.0000 \\ 0.1069 & 0.1956 & 0.2116 & 0.2898 & 0.1962 \end{pmatrix}$	0.9696	153.45

Table 6 also demonstrates (modest) enhancements in the overall efficiency of the compound designs. Their efficiency relative to each criterion is depicted in Table 7, emphasizing the advantageous nature of compound designs in optimally distributing losses across various criteria.

Table 7 Efficiency of the DA–optimal designs in Table 6 when used for the individual criteria

Model	Form. *Geomt*		Form. *Arthm*	
	Eff_D	Eff_A	Eff_D	Eff_A
1	0.9782	0.9837	0.9630	0.9961
2	0.9676	0.9749	0.9382	0.9958

5 Conclusions

We have addressed the challenge of systematically computing approximate compound optimal experimental designs. Our proposed approach utilizes Semidefinite Programming to exploit the currently available convex programming algorithms, which notably operate in polynomial time (i.e., are of P-type). Our strategy involves constructing new objective functions that aggregate measures of design efficiency relative to single criteria. Among the aggregation forms we consider are the geometric and arithmetic mean operators, which align with the robustness criteria of Läuter (1974) proposed for competing model setups.

Recognizing the necessity of Semidefinite representability of the objective functions in the construction of consistent formulations, we have identified three distinct cases:

1. concave-concave pairs of criteria. These give rise to (convex) hypographs, which combine into a new convex set, thus having a simple SDP representation aimed at finding its minimum.
2. convex-concave pairs of criteria. Here, we combine a hypograph resulting from the concave criterion with an epygraph arising from the convex criterion. To address this, we employ the Dinkelbach (1967) transformation, converting it into a nonlinear (conic) problem, subsequently reformulated as a bilevel program with the inner problem being SDr. The outer level is handled through a Surrogate Based Optimization solver, optimizing the combined efficiency of the designs, while the inner level is tackled with an SDP solver optimizing a proxy function of the overall efficiency.
3. convex-convex pairs of criteria. This problem gives rise to two epygraphs that may require Nonlinear Programming techniques beyond the scope of this paper but are potential areas to exploit in future research endeavors.

We assessed the proposed formulations with both linear and nonlinear models, resulting in globally optimal designs and locally optimal designs, respectively. Our findings indicate a slight improvement in combined efficiency relative to scenarios where a design optimized for one criterion is applied to another. Our formulations minimize (and distribute) the efficiency loss. Importantly, our formulations enable systematic resolution of problems in the optimal design of experiments for compound criteria with manageable computational complexity, by using available convex programming algorithms. In practical terms, this represents a significant step towards generalizing these approaches to address various problem scenarios effectively.

Not all problems in the numerical construction of compound optimal designs can be solved by the method presented here. One statistically important and interesting case is the class of DT optimal designs for model discrimination and parameter estimation, in

which the component designs can be very different. However the DT compound design cannot be obtained using SDP since the T-optimality criterion is not Semidefinite representable (SDr). Duarte et al. (2015) provide numerical methods for the construction of T-optimal designs. On the other hand, the K-optimality criterion is SDr, and the K-optimality criterion is quasiconvex. It could have been used as an example of a convex-concave combination of criteria. We have however chosen to illustrate this class of compound criteria using DA–optimality, which is a member of the same class of problems.

We follow Läuter (1974) in the robustness criteria considered. Use of the arithmetic and geometric means implies that we are implicitly equidistributing the loss of information in the chosen scale. In some cases it might be that there is more interest in the estimation of some models than of others; a set of polynomial models is an instance. Then, for example, Equation (9a) could be altered to give the required unequal weighting.

Acknowledgements Not applicable.

Author Contributions All the authors contributed equally to the study. All authors read and approved the final manuscript.

Funding Open access funding provided by FCT|FCCN (b-on). The paper received no funding.

Data Availability The datasets during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests The authors declare that they have no competing interests.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

References

Agrawal A, Boyd S (2020) Disciplined quasiconvex programming. Optimization Letters 14(7):1643–1657. https://doi.org/10.1007/s11590-020-01561-8

Alizadeh R, Allen JK, Mistree F (2020) Managing computational complexity using surrogate models: a critical review. Research in Engineering Design 31:275–298

Atkinson AC (2008) DT-optimum designs for model discrimination and parameter estimation. Journal of Statistical planning and Inference 138(1):56–64

Atkinson AC, Donev AN, Tobias RD (2007) Optimum Experimental Designs, with SAS. Oxford University Press, Oxford

Ben-Tal A, Nemirovski AS (2001) Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. Society for Industrial and Applied Mathematics, Philadelphia

Bhosekar A, Ierapetritou M (2018) Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Computers & Chemical Engineering 108:250–267

Boyd S, Vandenberghe L (2004) Convex Optimization. University Press, Cambridge

Buhmann MD (2009) Radial Basis Functions - Theory and Implementations, vol 12. Cambridge University Press

Cheng Q, Yang M (2019) On multiple-objective optimal designs. Journal of Statistical Planning and Inference 200:87–101. https://doi.org/10.1016/j.jspi.2018.09.007

Chernoff H (1953) Locally optimal designs for estimating parameters. The Annals of Mathematical Statistics 24:586–602

Clyde M, Chaloner K (1996) The equivalence of constrained and weighted designs in multiple objective design problems. Journal of the American Statistical Association 91(435):1236–1244

Cook RD, Wong WK (1994) On the equivalence of constrained and compound optimal designs. Journal of the American Statistical Association 89(426):687–692

Corana A, Marchesi M, Martini C, Ridella S (1987) Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm - corrigenda for this article is available here. ACM Trans. Math. Softw. 13(3):262–280. https://doi.org/10.1145/29380.29864

Crouzeix JP, Ferland JA, Van Nguyen H (2008) Revisiting Dinkelbach-type algorithms for generalized fractional programs. Opsearch 45(2):97–110

Dinkelbach W (1967) On nonlinear fractional programming. Management Science 13(7):492–498

Duarte BPM, Atkinson AC, Singh SP, Reis MS (2023) Optimal design of experiments for hypothesis testing on ordered treatments via Intersection-Union Tests. Statistical Papers 64:587–615. https://doi.org/10.1007/s00362-022-01334-8

Duarte BPM, Wong WK (2015) Finding Bayesian optimal designs for nonlinear models: A semidefinite programming-based approach. International Statistical Review 83(2):239–262

Duarte BPM, Wong WK, Atkinson AC (2015) A semi-infinite programming based algorithm for determining T-optimum designs for model discrimination. Journal of Multivariate Analysis 135:11–24

Erikson D, Bindel D, Shoemaker CA (2019) `pySOT` and `POAP`: An event-driven asynchronous framework for surrogate optimization

Gutmann HM (2001) A radial basis function method for global optimization. Journal of Global Optimization 19:201–227

Harman R, Jurík T (2008) Computing $c-$optimal experimental designs using the Simplex method of linear programming. Comput. Stat. Data Anal. 53(2):247–254

Heredia-Langner A, Montgomery DC, Carlyle WM, Borror CM (2004) Model-robust optimal designs: A Genetic Algorithm approach. Journal of Quality Technology 36:263–279

Hu J, Zhu W, Su Y, Wong WK (2010) Controlled optimal design program for the logit dose response model. Journal of Statistical Software 35(6):1–17. https://doi.org/10.18637/jss.v035.i06

Hwang CL, Masud ASM (2012) Multiple objective decision making-methods and applications: a state-of-the-art survey, vol 164. Springer Science & Business Media

Hyun SW, Wong WK, Yang Y (2018) VNM: An R package for finding multiple-objective optimal designs for the 4-parameter logistic model. Journal of Statistical Software 83(5):1–19. https://doi.org/10.18637/jss.v083.i05

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968

Kiefer J (1974) General equivalence theory for optimum design (approximate theory). Annals of Statistics 2:849–879

Kim SH, Boukouvala F (2020) Surrogate-based optimization for mixed-integer nonlinear problems. Computers & Chemical Engineering 140:106,847

Läuter E (1974) Experimental design in a class of models. Mathematische Operationsforschung und Statistik 5(4–5):379–398. https://doi.org/10.1080/02331887408801175

Le Digabel S (2011) Algorithm 909: `NOMAD`: Nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. 37(4):1–44

Martin JD, Simpson TW (2005) Use of kriging models to approximate deterministic computer models. AIAA Journal 43(4):853–863
Masoudi E, Holling H, Duarte BPM, Wong WK (2019) A metaheuristic adaptive cubature based algorithm to find bayesian optimal designs for nonlinear models. Journal of Computational and Graphical Statistics 28(4):861–876
McGree JM, Eccleston JA, Duffull SB (2008) Compound optimal design criteria for nonlinear models. Journal of Biopharmaceutical Statistics 18(4):646–661
Miettinen K (1999) Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston
Molchanov I, Zuyev S (2002) Steepest descent algorithm in a space of measures. Statistics and Computing 12:115–123
MOSEK, A.: MOSEK Modeling Cookbook – Release 3.3.0. https://docs.mosek.com/MOSEKModelingCookbook-a4paper.pdf **7**, 25 (2024)
Müller J (2014) `MATSuMoTo`: The Matlab surrogate model toolbox for computationally expensive black-box global optimization problems
Müller J (2016) `MISO`: mixed-integer surrogate optimization framework. Optimization and Engineering 17:177–203
Müller J, Day M (2019) Surrogate optimization of computationally expensive black-box problems with hidden constraints. INFORMS Journal on Computing 31(4):689–702
Müller J, Shoemaker CA, Piché R (2013) `SO-MI`: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Computers & Operations Research 40(5):1383–1400
Müller J, Woodbury JD (2017) `GOSAC`: global optimization with surrogate approximation of constraints. Journal of Global Optimization 69:117–136
Nesterov YE, Nemirovski AS (1994) Interior-Point Polynomial Algorithms in Convex Programming (Studies in Applied and Numerical Mathematics). Society for Industrial Mathematics
Powell MJD (1992) The theory of radial basis function approximation in 1990. In: Light WA (ed) Advances in Numerical Analysis II: Wavelets, Subdivision, and Radial Functions. Oxford University Press, Oxford, pp 105–210
Pukelsheim F (1993) Optimal Design of Experiments. SIAM, Philadelphia
Qiu J, Chen RB, Wang W, Wong WK (2014) Using animal instincts to design efficient biomedical studies via particle swarm optimization. Swarm and evolutionary computation 18:1–10
Qiu J, Wong WK (2023) Nature-inspired metaheuristics for finding optimal designs for the continuation-ratio models. The New England Journal of Statistics in Data Science pp. 1–15. https://doi.org/10.51387/23-NEJSDS44
Regis RG, Shoemaker CA (2007) A stochastic radial basis function method for the global optimization of expensive functions. INFORMS Journal on Computing 19(4):497–509
Sagnol G (2011) Computing optimal designs of multiresponse experiments reduces to second-order cone programming. Journal of Statistical Planning and Inference 141(5):1684–1708
Sagnol G (2013) On the semidefinite representation of real functions applied to symmetric matrices. Linear Algebra and its Applications 439(10):2829–2843
Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11:341–359
Vandenberghe L, Boyd S (1996) Semidefinite programming. SIAM Review 8:49–95
Vandenberghe L, Boyd S (1999) Applications of semidefinite programming. Applied Numerical Mathematics 29:283–299
Wong WK, Zhou J (2023) Using CVX to construct optimal designs for biomedical studies with multiple objectives. Journal of Computational and Graphical Statistics 32(2):744–753. https://doi.org/10.1080/10618600.2022.2104858
Yang M, Biedermann S, Tang E (2013) On optimal designs for nonlinear models: A general and efficient algorithm. Journal of the American Statistical Association 108(504):1411–1420
Ye JJ, Zhou J (2013) Minimizing the condition number to construct design points for polynomial regression models. SIAM Journal on Optimization 23(1):666–686
Ye Y (1997) Interior Point Algorithms: Theory and Analysis. John Wiley & Sons, New York
Yu Y (2010) D-optimal designs via a cocktail algorithm. Statistics and Computing 21(4):475–481