ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign

Roshan Suresh Kumar[1]
Department of Aerospace Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: roshan94@tamu.edu

Srikar Srivatsa
Sibley School of Mechanical and Aerospace Engineering,
Cornell University,
Ithaca, NY 14850
e-mail: sms722@cornell.edu

Emilie Baker
Sibley School of Mechanical and Aerospace Engineering,
Cornell University,
Ithaca, NY 14850
e-mail: erb259@cornell.edu

Meredith Silberstein
Sibley School of Mechanical and Aerospace Engineering,
Cornell University,
Ithaca, NY 14850
e-mail: meredith.silberstein@cornell.edu

Daniel Selva
Department of Aerospace Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: dselva@tamu.edu

Identifying and Leveraging Promising Design Heuristics for Multi-Objective Combinatorial Design Optimization

Design heuristics are traditionally used as qualitative principles to guide the design process, but they have also been used to improve the efficiency of design optimization. Using design heuristics as soft constraints or search operators has been shown for some problems to reduce the number of function evaluations needed to achieve a certain level of convergence. However, in other cases, enforcing heuristics can reduce diversity and slow down convergence. This paper studies the question of when and how a given set of design heuristics represented in different forms (soft constraints, repair operators, and biased sampling) can be utilized in an automated way to improve efficiency for a given design problem. An approach is presented for identifying promising heuristics for a given problem by estimating the overall impact of a heuristic based on an exploratory screening study. Two impact indices are formulated: weighted influence index and hypervolume difference index. Using this approach, the promising heuristics for four design problems are identified and the efficacy of selectively enforcing only these promising heuristics over both enforcement of all available heuristics and not enforcing any heuristics is benchmarked. In all problems, it is found that enforcing only the promising heuristics as repair operators enables finding good designs faster than by enforcing all available heuristics or not enforcing any heuristics. Enforcing heuristics as soft constraints or biased sampling functions results in improvements in efficiency for some of the problems. Based on these results, guidelines for designers to leverage heuristics effectively in design optimization are presented. [DOI: 10.1115/1.4063238]

Keywords: design methodology, design of engineered materials system, design optimization, multi-objective optimization

1 Introduction

This paper explores approaches to leverage design heuristics in design optimization. Design optimization is a widely used method in engineering design, particularly to support trade studies and detailed design. However, most of the literature studying design heuristics has focused on other design processes such as design ideation and concept exploration [1,2]. In general, the design optimization process consists of a problem formulation step, a search step in which the tool iterates between generating and evaluating designs, and a data analysis step that leads to decision-making, i.e., the down-selection of a (few) design(s) to develop in more detail. As shown in Fig. 1, expert knowledge can be incorporated in different ways in design, including into the different steps of design optimization. We classified these ways into direct input (human-in-the-loop), explicit approaches, and implicit approaches. As examples of direct input, the engineer directly uses their expert knowledge to develop a problem formulation and refine it based on the results. They can also be in the loop during the optimization, playing different roles such as generating or modifying designs, evaluating designs, or steering the search towards promising regions of the design space [3]. The user can also implicitly provide expert knowledge, e.g., by providing an initial dataset of good or feasible designs [4]. Alternatively, the user or developer of the tool can explicitly incorporate their knowledge in different ways, such as in expert systems [5,6], using ontologies and knowledge graphs [7,8], or using design heuristics [9,10], which is the focus of this paper.

A design heuristic (as defined in Fu et al. [11]) is a context-dependent directive, based on intuition, tacit knowledge, or experiential understanding, which provides design process direction to increase the chance of reaching a satisfactory but not necessarily optimal solution. In other words, it is a rule of thumb that can guide the design search toward improved but not necessarily optimal designs. As a result, heuristics can be leveraged to improve the efficiency of the design optimization process, i.e., to reach satisfactory designs using fewer function evaluations compared to conventional data-driven approaches [12–14].

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received January 7, 2023; final manuscript received August 11, 2023; published online September 11, 2023. Assoc. Editor: Katherine Fu.

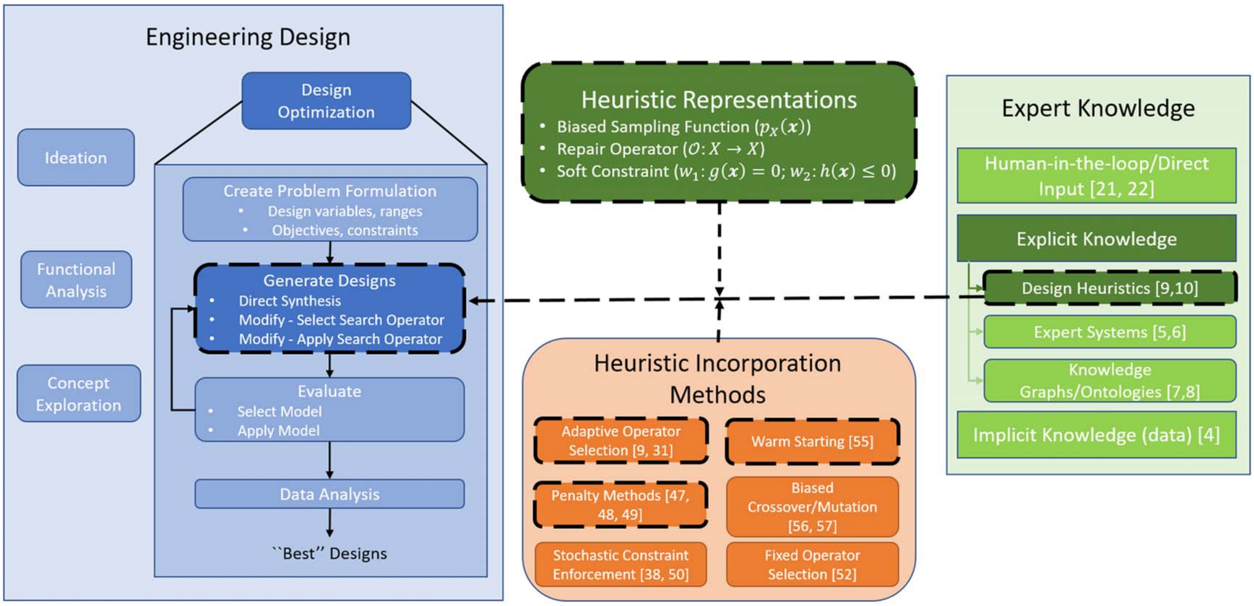

Fig. 1 The scope of the paper within the broad area of incorporating expert knowledge in design processes is shown with thick dashed lines. The focus of this work is leveraging expert knowledge as design heuristics in the design search step of design optimization. For this purpose, design heuristics can be represented in different ways and incorporated into the search process using different methods.

Extraction of design heuristics can be done using interviews [13,15], examination of high-quality products [16], or through data-driven methods [17,18]. Fueled by the digital engineering revolution, design heuristics are increasingly being captured in computer-friendly forms (e.g., SysML models), conducive to direct incorporation into computational design tools. Together with the advances in large language models, this provides an opportunity since organizations may soon have databases of design heuristics that could be leveraged to improve their design processes. To achieve this goal, new methods must be developed to automatically extract design heuristics, identify which ones are relevant for a given design problem and task, and incorporate them into that task in a way that improves efficiency. A recent survey of industrial and academic professionals showed that, along with an easy-to-understand representation of design heuristics, the ability to identify good heuristics and test their applicability and validity for a given design problem is needed [19].

Design heuristics are used throughout the product/system development process. For example, heuristics are widely used in design ideation [20,21]. This paper, however, deals with applying heuristics to design optimization, where there is less literature, despite the fact that heuristics have been used in practice to solve complex optimization problems for decades. Human-in-the-loop design optimization has been found to improve efficiency thanks to designers using their expert knowledge and common sense. However, having a human in the loop is costly [22–24], thus motivating the study of methods to automate the identification and incorporation of key chunks of expert knowledge into the optimization. Alternatively, expert knowledge can also be implicitly represented as databases of good or feasible designs, which can be used to warm start classical optimization approaches or to train generative design or reinforcement learning agents for design optimization (e.g., Ref. [4]). Other methods to accelerate convergence of design optimization algorithms, such as surrogate-assisted optimization [25,26] and Bayesian optimization [27,28] utilize function evaluations to train computationally cheap surrogate models. Another method to incorporate expert knowledge is an ad-hoc generative design algorithm, but this is not always available or viable to develop [29,30]. Theoretically, the use of expert knowledge as design heuristics does not require any additional function evaluations, although adaptive methods do use function evaluations to learn the applicability of the heuristics to the design problem [31]. Therefore, the question arises of when it is worthwhile in practice to attempt to incorporate these domain-specific heuristics into optimization algorithms, given the net improvement in computational performance, and overall increase in complexity of the algorithm.

Several papers have devised design heuristics for specific design problems and shown that adding them to a search algorithm, either as constraints or as repair operators, can improve search performance [32,33]. Binder and Paredis [34] presented a design method for the problem of designing a pressure vessel in which a heuristic function based on an inbuilt factor of safety is considered in the computation of the wall thickness. They found that the heuristic-enabled design method outperformed an expected-utility-maximization approach that explicitly accounted for uncertainty in certain regions of the design space. Calvo et al. [10] introduced a novel heuristic-enabled multi-objective optimization approach to tackle a protein structure prediction problem. Problem-specific heuristics related to the side-chain torsion angles in the protein structure were incorporated as a library of dependencies between these angles and used to appropriately modify protein structure designs. This approach was shown to be faster than and have comparable performance to the benchmark algorithm [35] that iteratively assembles protein structures through guided fragment rearrangements. Hitomi and Selva [36] developed heuristics to design Earth Observation Satellite systems (EOSS). They observed that even though all design heuristics were "good" (they improved the aspect of designs that they are developed to address), not all heuristics were useful for a given design problem in the sense that they improved search efficiency when incorporated as repair operators or soft constraints. For example, a design heuristic may address a problem goal that is very easy to solve by random search for the given problem formulation, or it may improve an objective but degrade a conflicting objective or constraint. Additionally, Hitomi and Selva [9] compared various ways of leveraging those heuristics based on repair operators and soft constraints. They showed that for the constellation design problem, leveraging the design heuristics as repair operators within an adaptive operator selection (AOS) [31] strategy was superior to using constraint-based methods such as adaptive constraint handling (a combination of Refs. [37] and [38]). Thus, in addition to identifying the subset of available heuristics to leverage, the selection of the method to leverage those heuristics is also important [39,40]. Of note, AOS may require many function evaluations to learn the promising heuristics to leverage if the candidate set is large.

The main contribution of this paper is a method to identify promising design heuristics from a set of candidate heuristics for a given design optimization problem. Three forms of heuristics are considered: soft constraints, repair operators, and biased sampling functions. Two novel heuristic impact indices are proposed to identify the promising heuristics for a given design optimization problem. The estimation of these indices is done in an "offline" screening study preceding the optimization, to maximize exploration. The approach is applied to four test problems: two constrained metamaterial design problems and two unconstrained Earth Observation Satellite constellation design problems. We validate the method by comparing the search performance and efficiency in the four test problems of three approaches: no heuristics, all heuristics, and promising heuristics only. In addition, seven guidelines are identified based on our findings to help tool users and developers leverage design heuristics for design optimization. This paper extends the work of Kumar et al. [41] to generalize the method of identifying promising heuristics to different design problems and different heuristic representations. Unlike previous work, this paper considers constrained optimization problems, which adds new challenges when enforcing heuristics that help with constraint satisfaction but hurt other problem goals. The scope of the paper is multi-objective evolutionary optimization with discrete (combinatorial) design decisions, but conceptually the method can be applied to continuous design decisions and other optimization schemes.

The paper is structured as follows: Sec. 2 presents and discusses related work on methods to incorporate design heuristics into design optimization; Sec. 3 presents the two metamaterial design problems and two satellite constellation case studies; Sec. 4 presents and motivates the methods used to ascertain the set of promising heuristics for a given design problem and validating their efficacy; Sec. 5 details the results of the determination of promising heuristics and the verification of their effectiveness for the design problems; Sec. 6 provides a detailed discussion on the results from Sec. 5 and presents some guidelines for designers to incorporate design heuristics into their design problems; and Sec. 7 closes the paper with the main conclusions, limitations, and future work.

2 Leveraging Design Heuristics for Design Optimization

A general constrained multi-objective design optimization problem can be formulated as follows:

$$\begin{aligned} \boldsymbol{x}^* &= \arg\min_{x\in X} \boldsymbol{f}(\boldsymbol{x}) \\ \text{s.t.} \quad \boldsymbol{g}(\boldsymbol{x}) &= \boldsymbol{0} \\ \boldsymbol{b}(\boldsymbol{x}) &\leq \boldsymbol{0} \end{aligned} \tag{1}$$

where $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{b}(\boldsymbol{x})$ are the equality and inequality constraints on $\boldsymbol{x}$. Solving such a design optimization problem implies finding a set of non-dominated designs ($\boldsymbol{x}^*$) that approximates the true Pareto Front (PF). This problem is especially challenging when design decisions are discrete and objective functions and constraints are nonlinear and computationally expensive, which is the focus of this paper. Meta-heuristic algorithms such as evolutionary algorithms are often employed to solve these problems [42]. Therefore, they are used as the overarching optimization framework in this paper.

In this paper, we define the penalized objectives as

$$f_{pen}(\boldsymbol{x}) = \boldsymbol{f}(\mathbf{x}) + \mathbf{1}_t \odot \left(\sum_{j=1}^{q} g_j(\mathbf{x}) + \sum_{k=1}^{r} b_k(\mathbf{x}) \right) \tag{2}$$

Here q and r are the number of equality and inequality constraints respectively, $\mathbf{1_t}$ is an $t \times 1$ size vector of ones, and $\odot$ is the element-wise product. Assuming that the objectives and constraints are normalized, satisfaction of each constraint is considered as important as Pareto optimality. As such, the weight of the constraint violation penalties would be equal to the total number of constraints. We refer to the "problem goals" (noted p_i) as satisfying each of the constraints and finding non-dominated designs.

There are many types of multi-objective evolutionary algorithms including dominance-based algorithms such as NSGA-II [43] or ϵ-MOEA [44], decomposition-based algorithms such as MOEA/D [45], and indicator-based algorithms such as IBEA [46]. The evolutionary algorithm used in this work is ϵ-MOEA. ϵ-MOEA is a steady-state algorithm that maintains an archive of the best designs at different stages of the optimization process and uses an ϵ-box approach to prevent duplicates in the archive and promote design diversity in the PF.

Three different types of heuristic representations can be used: (1) soft constraint; (2) repair operator; and (3) biased sampling function. These representations come from the literature search and try to generalize some previous work by the authors on heuristics as repair operators (e.g., Ref. [9]) by systematically considering the functions involved in design optimization and how heuristics or expert knowledge in general could be applied to each of those.

Some design heuristics are most naturally represented as a soft constraint, i.e., a function $a : X \to \mathbb{R}$, where $x \in X$ represents a design from the design space X and $a(x)$ represents the degree of violation of the heuristic by design x. From the definition of a heuristic, it is generally suboptimal to incorporate heuristics as hard constraints, and a soft constraint approach is more appropriate. Methods for handling soft constraints include using a weighted-objective penalty function [47,48], or a constraint selection strategy, which can be deterministic [49] or stochastic [38,50].

Other heuristics are most naturally represented as a repair operator representation that encodes the prescriptive action or directive represented by the heuristic in the form of a *move* in design space, i.e., a function $O : X \to X$. This form is consistent with the representation of heuristics considered in Filingim et al. [13], where heuristics prescribe an action toward improving a design. The operator form acts upon a design to produce a new design that adheres to the directives of the heuristic to a greater extent. Just as hard constraint handling mechanisms should not be used to enforce heuristics, heuristic-based operators [32,51] are often used in conjunction with other knowledge-independent operators so as not to excessively reduce exploration of the design space [51]. When using the operator form, heuristics are handled with operator selection strategies, which can be either fixed or adaptive and deterministic or stochastic. Fixed strategies continuously apply the same set of knowledge-directed operators in the same way throughout the design search, e.g., assigning a fixed percentage of solutions to each operator at each iteration [52]. On the other hand, AOS strategies assign more solutions to operators that perform better. This requires using a credit assignment strategy to keep track of the performance of the operators, and applying an operator selection strategy that considers the performance of the operators to assign solutions [31]. This approach was used in Ref. [9] to incorporate design heuristics to the satellite design problem. A pool of knowledge-independent (e.g., crossover and mutation) and knowledge-dependent operators is maintained, with operators being selected based on their relative cumulative performance. Different credit assignment strategies (e.g., offspring dominates parent, set improvement, contribution to set) and operator selection strategies (probability matching, adaptive pursuit) have been proposed in the literature, building on multi-armed bandit theory [53,54].

Finally, a heuristic can also be represented as a biased sampling function that produces a set of designs that statistically tend to satisfy the directives of the heuristic. In population-based algorithms, the biased set of designs can be used as the initial population, with the hope that the initial presence of some "good designs" will accelerate convergence [55]. These biased sampling functions can also be used in any other search operator that requires

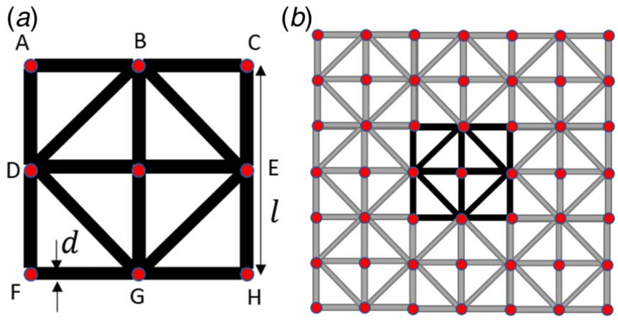

Fig. 2 Example design: (a) the 3 × 3 unit cell and (b) the 3 × 3 repeated lattice. The bars are members that can only submit axial forces. The circles are nodes that are modeled mechanically as pin joints.

random sampling of the design space, such as in mutation, selection, or crossover [56,57]. Note that the same heuristic intent or directive can often be implemented using different representations. More broadly, the details of how the heuristic intent is actually implemented may affect the heuristic's performance, as is discussed in Sec. 6.

3 Test Problems

Four test problems, two metamaterial design problems and two EOSS design problems, are used to demonstrate the proposed method. These problems were chosen because they are representative of realistic combinatorial optimization problems found in engineering design across different areas.

3.1 Metamaterial Design Problems.

Metamaterials are materials whose properties are determined by geometry of a repeated microstructure, in addition to the intrinsic mechanical properties.[2] The particular metamaterial design problems in this paper were selected because they contain hard constraints that are difficult to satisfy, in addition to two conflicting objectives.

3.1.1 Artery Design Problem. The artery design problem is inspired by matching the 2D components of a stiffness tensor of an artery. The properties are taken from a material model fit to rabbit carotid artery data (Table II in Ref. [58]). A 2D 3×3 node grid is considered which represents a single repeat unit cell of the metamaterial (shown in Fig. 2). The design decisions are binary variables that represent the presence or absence of truss members within the 3×3 node grid. Allowing for all possible connections between pairs of nodes, there are 36 possible truss members. However, to account for repetition of lattice units in the two orthogonal directions, the design decisions corresponding to members on opposite edges must be the same (i.e., using Fig. 2(a) as reference $x_{AB} = x_{FG}$, $x_{BC} = x_{GH}$, $x_{AD} = x_{CE}$, and $x_{DF} = x_{EH}$), thus making the design vector of 30 elements long.

The effective 2D (or in-plane) material stiffness tensor for each design is evaluated by modeling the lattice as a truss. Each linear elastic member can only deform axially and connects to other members only at the nodes, which are modeled as pin joints. The stiffness values of individual members are combined based on shared member endpoints to form the global stiffness matrix. The effective material stiffness tensor is then calculated by applying a series of controlled displacements to the boundary of the lattice, calculating force, and normalizing by area to obtain stress components. The volume fraction of the design is found by summing the volumes of all members, subtracting redundant volumes that occur at the intersections of members, and then dividing the corrected lattice volume by a volume of the same side length and thickness of the unit cell. Based on convergence of the moduli, all values were calculated using 3×3 repeats of the 3×3 node grid. The ratio of the radius of each member to the side length of the lattice unit is fixed at 5:200. All material stiffness tensor values are normalized to the constituent material elastic modulus.

An unrestricted design space contains many designs that are not realizable. First, nodes cannot be connected to exactly one member, as this member will then not actually connect with the rest of the material. Second, no members can cross or overlap since they would then occupy the same physical space. These hard requirements are enforced as the connectivity (g_{conn}) and feasibility constraints (g_{feas}), respectively. Constraint functions are defined for each constraint that span from 0 to 1, with 1 representing no violations and 0.1 subtracted for each violation.

The goal is to maximize density-normalized vertical stiffness $\left(\dfrac{(C_{11}/v_f) - 2e5}{1e6}\right)$ while minimizing a deviation term (f_{dev}) that is the average of various normalized stiffnesses and stiffness ratios of the repeated lattice configuration, subject to constraints on design feasibility ($1 - g_{feas}(\boldsymbol{x}) = 0$) and design connectivity ($1 - g_{conn}(\boldsymbol{x}) = 0$). The deviation term f_{dev} is specified as

$$f_{dev} = \frac{\left|\dfrac{C_{22}}{C_{11}} - 0.421\right|}{6} + \left|\frac{C_{21}}{C_{11}} - 0.0745\right| + \left|\frac{C_{12}}{C_{11}} - 0.0745\right| + \frac{\left|\dfrac{C_{66}}{C_{11}} - 5.038\right| - 4.5}{0.5} + \frac{|C_{61}|}{1.5e5} + \frac{|C_{62}|}{1.5e5} + \frac{|C_{16}|}{9e4} + \frac{|C_{26}|}{9.5e4} \quad (3)$$

The normalization constants were adjusted to ensure that each objective is normalized and each deviation term contributes equally to the f_{dev} objective.

3.1.2 Equal Normal Stiffness Metamaterial Problem. This problem is modeled as a constant radius truss design problem that considers the same 2D 3×3 node grid as the artery problem. The same model used in the artery problem to compute the objectives and constraints is utilized here but the goal is different: to maximize vertical stiffness while minimizing volume fraction, subject to constraints above on design feasibility and design connectivity and an additional constraint representing a target stiffness ratio ($g_{stiff}(\boldsymbol{x}) = 0$).

Here, $g_{stiff} = \left|\frac{C_{22}}{C_{11}} - c_{target}\right|$, and we use $c_{target} = 1$.

3.1.3 Candidate Heuristics. Four general heuristics are identified for both metamaterial design problems. All soft constraints are heuristic satisfaction functions (i.e., large-is-better).

(1) Partial collapsibility (PC): Include at least one diagonal member in each half of the unit cell, both vertically and horizontally. This heuristic embodies resistance of the truss design to collapse due to shear loading.
(2) Nodal properties (NP): Create designs where each node has at least three connections unless completely unused and there is at most one unused node, for physically stability. The main goal is to aid in satisfaction of the connectivity constraint.
(3) Orientation (OR): This heuristic is aimed at satisfying the stiffness ratio constraint by instructing designers to create designs that achieve a certain target average orientation of its members (from the horizontal axis), computed from the target stiffness ratio c_{target}. The orientation soft constraint form (a_{or}) varies from 0 to 1 with 1 implying full heuristic satisfaction and calculates a design's orientation as the average orientation of all individual members relative to

[2]The code used for data generation, post-processing, and analysis is available on https://github.com/seakers/KDDMM and data used in this paper are available upon request.

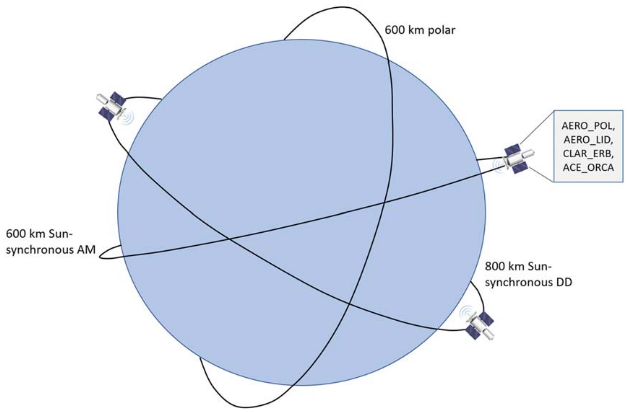

Fig. 3 An example architecture design for the two EOSS problems is shown. The design shows three orbits around the earth, the first with two satellites, the second with one, and the third with no satellites. The instruments in one of the satellites are shown (AERO_POL: aerosol polarimeter, AERO_LID: differential absorption lidar, CLAR_ERB: broadband radiometer, ACE_ORCA: ocean color spectrometer).

the horizontal axis, then assigns a score based on the deviation of the design's orientation from the target orientation (e.g., 45 deg for a target stiffness ratio of 1). The target orientation is determined by assuming the total member stiffness contribution is directly proportional to the average member vector. The heuristic operator adds a member such that the average orientation of the members is closest to the target orientation. This operator is a greedier version of the one used in Kumar et al. [41]. Additionally, the biased prior distribution form generates designs that have an average orientation across all members within a margin of 10 deg from the target orientation as computed using the target stiffness ratio.

(4) Intersection (IS): Aimed at improving satisfaction of the feasibility constraint, this heuristic instructs designers to minimize intersections among members.

Only OR was available in the biased sampling form. The soft constraint and repair operator implementations of the PC, NP, and IS heuristics are provided in the Supplementary Material available in the Supplemental Materials on the ASME Digital Collection.

3.2 Satellite Constellation Design Problems.

The two EOSS problems (assignment and partitioning) are concerned with the design of a satellite constellation system to measure different climate-related parameters such as ocean color, precipitation rate, and atmospheric temperature.[3] The goal is to maximize the scientific benefit while minimizing lifecycle cost. The scientific benefit [59,60] metric quantifies the degree of satisfaction of over 370 measurement requirements provided by the World Meteorological Organization OSCAR database.[4] The Value ASsessment of System Architectures using Rules (VASSAR) tool [61], which uses a rule-based system to compute the degree of satisfaction of these requirements, is used to calculate the scientific benefit for both problems. The lifecycle cost [59,60] metric includes different costs associated with the development, implementation, integration, testing, launch, and operation of the system. It is computed using the spacecraft design algorithm and cost-estimating relationships also available in VASSAR.

In both problems, each satellite in the constellation can have a subset of the 12 instruments and can be present in one of the five orbits as shown in the Supplementary Material available in the Supplemental Materials. Figure 3 shows the candidate orbits for the two EOSS problems. In the EOSS assignment problem, any subset of the 12 instruments can be assigned to each of the five orbits, resulting in a design vector of 60 binary variables. All the instruments in a given orbit are assumed to be in the same satellite. See Ref. [9] for a more complete description of this problem. In the EOSS partitioning problem, each instrument must be assigned to exactly one satellite, but each satellite can be assigned to any of the five orbits. See Ref. [59] for a more complete description of the partitioning problem. Both problems are unconstrained beyond the structural constraints ensured by the formulation of the design vector.

3.2.1 *Candidate Heuristics.*

Seven general heuristics were used for the two EOSS problems. The first six heuristics presented are common to both problems, whereas the last heuristic is used only for the assignment problem. All seven general heuristics were implemented both as a soft constraint and a repair operator. Only the last heuristic was available as a biased sampling function. While the intent of each general heuristic is the same for both problems, the implementation as a repair operator is slightly different for the two problems (e.g., to ensure that valid designs are generated in the partitioning problem). The intents of the general heuristics are listed below. The implementations of each heuristic representation are provided in the Supplementary Material available in the Supplemental Materials. All soft constraints are heuristic violation functions (i.e., small-is-better). The candidate heuristics are as follows:

(1) Instrument duty cycles (DC): Avoid resource-constrained satellites where the duty cycle of instruments must be artificially reduced to save energy or bandwidth. The soft constraint form of this heuristic computes the violation as the average violation by each satellite, where each satellite violation is the difference between the duty cycle of the satellite and a threshold (0.5), or 0 if duty cycle exceeds the threshold. The operator form moves or removes an instrument from a satellite that is resource-constrained.

(2) Instrument-orbit relations (IO): Try to satisfy two instrument-orbit preference relations. First, instruments measuring atmospheric ozone are best assigned to an afternoon SSO to observe peak pollution. Second, passive optical instruments should not be assigned to dawn-dusk orbits to avoid poor illumination conditions.

(3) Instrument interference (IF): Avoid assigning certain types of instruments to the same satellite, since that will lead to increases in spacecraft design complexity and cost [59]. For example, avoid assigning a conically scanning instrument and a sensitive limb sounder to the same satellite, since the former will induce vibrations that will degrade the performance of the latter. Ten such instrument pairs are identified.

(4) Packing efficiency (PE): Try to use the full payload capacity of the launch vehicle. A launch vehicle packing efficiency is computed as the max of: (1) the fraction of the volume of the launch vehicle fairing used by the combined payload volume, and (2) the fraction of the launch vehicle's lifting capacity to the assigned orbit used by the total launch mass) and compared to a threshold of 0.7.

(5) Spacecraft mass (SM): Most civilian Earth Observation satellites weigh less than 3000 kg [62], because beyond this rough threshold, increased complexity may lead to diseconomies of scale. Consequently, the heuristic penalizes or tries to repair designs with satellites that exceed a wet mass of 3000 kg.

(6) Instrument synergies (SYN): Ten pairs of synergistic instrument types are identified, whose data products can be

[3]The code used for data generation, post-processing, and analysis is available on https://github.com/seakers/VASSAR–;exec.git—heuristics branch for the EOSS problems. The raw data are available upon request.

[4]Observing Systems Capability Analysis and Review Tool available at www.wmo-sat.info/oscar

combined when acquired from the same platform to improve each other's accuracy or create new higher-level data products. Thus, the heuristic directive is to incorporate these synergistic instrument pairs into satellites as much as possible.

(7) Instrument count (IC): This heuristic instructs designers to limit the total number of instruments in the constellation (all satellites in all orbits represented by the design) to a reasonable number (15). Keeping the number of instruments per satellite low simplifies platform design and allows to optimize the design for the given payload, which in turn may improve performance and reduce cost.

4 Methods

The overall methodology is as follows. First, heuristic impact indices are calculated to identify the promising heuristics among a set of available heuristics for a given problem. These indices are calculated in an off-line screening study. Second, the promising heuristics are incorporated into the optimization and their efficacy (impact on search performance) is benchmarked against the cases of no heuristics and all heuristics used. This is done for the four test problems with all the heuristics defined earlier.

4.1 Identifying Promising Heuristics. A screening study is used to determine the promising heuristics in an offline manner to promote exploration. Moreover, for the soft constraint representation, this allows determining the impact indices for all heuristics using the same set of designs.

In general, the impact indices are evaluated on a set of randomly generated designs, as this emulates the initial stages of the optimization. Heuristics are most useful when only a modest number of function evaluations are possible, and are hence expected to show improvements in the initial stages of optimization.

4.1.1 Weighted Influence Index. As mentioned above, heuristics can improve a problem goal while degrading conflicting goals. Therefore, the idea of the heuristic impact index is to capture the overall effect of a heuristic across all problem goals. Intuitively, heuristics strongly aligned with more important or harder-to-satisfy objectives and constraints will help the most. This is the idea of the weighted influence index (WII) I_{wi}. I_{wi} estimates the overall impact of a heuristic on the design search as a weighted average of the correlation coefficients $\rho(a, p_i)$ of the heuristic's soft constraint function a with each problem goal p_i. Here, the problem goals are satisfaction of each of the constraint functions $\boldsymbol{g_i}(\boldsymbol{x})$, $\boldsymbol{b_j}(\boldsymbol{x})$, and dominance in objective space. The dominance goal is represented as minimizing the average distance to the PF d_{PF}. d_{PF} for a design x is computed as the distance between x and the closest Pareto design. The WII formulation is shown in Eq. (4).

$$I_{wi}(h) = \frac{1}{m}\sum_{i=1}^{m} w(p_i) k_{h,p_i} \rho(h, p_i) \tag{4}$$

where $m = q + r + 1$ is the number of problem goals and $w(p_i)$ is the relative weight of problem goal p_i. In this paper, we choose $w(p_i)$ to be the mean constraint violation and/or the mean d_{PF} observed in the sample design set. As a result, hard to satisfy constraints will have a greater contribution to I_{wi}. However, the designer may wish to assign importance to problem goals based on other criteria. The weights can be normalized between 0 and 1 if it is desired to compare values of $I_{wi}(h)$ across problems. k_{a,p_i} is the expected sign of the correlation: $k_{a,p_i} = 1$ if increasing the heuristic's soft constraint function is expected to increase the problem goal function (e.g., a is a constraint violation function and the goal is function small-is-better) and $k_{h,p_i} = -1$ if it is expected to decrease it. Thus, a positive index value is always expected for promising heuristics. A positive (negative) value of I_{wi} means that the average effect of the heuristic across all objectives and constraints is positive (negative, respectively). A value of $I_{wi} = 1$ would mean that the heuristic has a perfect linear relationship with all the objectives and constraints and can help improve all of them. This, of course, is extremely unlikely to occur since the different objectives and constraints in real-world problems tend to conflict with one another.

There are several choices for the correlation coefficient. In this paper, both the Pearson [63] (ρ_p) and Spearman [64] (ρ_s) correlation coefficients are used to compute two values of the index which are then averaged. The Pearson's and the Spearman's correlation coefficient measure the degree of linearity and monotonicity respectively between the independent and dependent variable, so averaging the two indices can capture both types of dependencies. This average provides more accurate correlation estimations in case of sparse distributions of heuristic and constraint satisfaction compared to just using either correlation coefficient, which occurs in the case of very hard and very easy to satisfy heuristics and constraints such as the feasibility and connectivity constraints for the metamaterial problems. Beyond correlation coefficients, other metrics could be used to measure the statistical alignment of the heuristics with the problem goals. For example, interestingness measures from the association rule mining literature (e.g., lift [65] or F-score [66]) could also be used, but that would require binary heuristic functions and problem goals and thus subjective thresholding of the heuristic soft functions to determine what is a promising value of the heuristic.

4.1.2 Hypervolume Difference Index. Another approach to assess the impact of a heuristic, inspired by the concept of main effects in design of experiments, is to define impact as the average difference in problem goal satisfaction before and after the heuristic is applied. That is the idea behind the hypervolume difference index (HDI) which assesses the overall effect on the problem's goals of applying the heuristic (its repair operator or biased sampling function) to an initial set of designs. Rather than doing a weighted average of individual impacts on problem goals as in the WII, here we simply combine dominance and constraint satisfaction goals into a single scalar goal, such as the HyperVolume (*HV*) [42] of the penalized objectives. It is important to note that HV captures both convergence and to a certain extent diversity (in the objective space) of the Pareto designs [67]. Note that the weights of the problem goals (e.g., the ones used in WII) can still be incorporated in the penalized objectives. Alternative metrics such as HV of the feasible solutions (which fully satisfy the constraints) can also be used, but would likely yield zero throughout the screening study for problems with very hard-to-satisfy constraints (e.g., the equal stiffness problem). The HDI index is defined as shown in Eq. (5).

$$I_{hd}(h) = HV(\boldsymbol{y_h}) - HV(\boldsymbol{y}) \tag{5}$$

where $\boldsymbol{y}$ and $\boldsymbol{y_h}$ are the penalized objectives of the test designs before and after the operator is applied respectively. A positive value of I_{hd} indicates a promising heuristic. A positive (negative) value of I_{hd} means that the effect of the heuristic on the penalized HV (which considers all objectives and constraints) is positive (negative, respectively). If $I_{hd} = 0$, that means that the non-dominated set after application of the heuristic is the same as before applying it (or at least it covers the same HV), i.e., the heuristic has not improved or worsened the designs globally. If the initial HV was 0 and the best achievable HV was 1, $I_{hd} = 1$ would indicate that when the heuristic is applied to a base set of designs to generate new designs, the newly non-dominated set is the true Pareto Front (i.e., the heuristic is as effective as it could possibly be). This, of course, is unrealistic. Moreover, in the cases presented here, the start HV is not 0 and the max HV is not 1 so the max possible value of I_{hd} is unknown. Beyond HV, other metrics could be used to assess impact, including any of the well-known indicators used in indicator-based multi-objective evolutionary algorithms. Regardless, the computation of an HDI-like index is less efficient than that of WII since in HDI, a set of designs needs to be generated for each heuristic to assess, whereas in WII, given a set of designs, one can compute WII for any number of heuristics.

While both WII and HDI could theoretically be applied to any heuristic representation, the WII is a more natural choice for heuristics represented as soft constraints since one can readily compute the correlation between problem goals and heuristic violation functions. On the other hand, the HDI is a more natural choice for repair operators and biased sampling than the WII, since it is closer to the way we assess performance in multiobjective optimization. For biased sampling, the index compares the HV for a randomly sampled population versus the heuristic-biased population.

4.1.3 Screening Study. The procedure to screen a heuristic of different representations is illustrated in Fig. 4. Positive indices suggest promising heuristics. However, the index is a random variable because the design sampling process and the operator forms of the heuristics are stochastic. Therefore, we compute each index $N = 10$ times with different sets of 300 designs generated by simple random sampling. Other techniques from design of experiments like orthogonal arrays or Latin hypercube sampling could be employed for dataset generation. The number of samples and sampling strategy for a dataset should be such that good coverage of the design space with a diverse range of objectives, constraints, and heuristics is obtained. To identify promising heuristics, rather than just using the average index, we use the empirical probability that the index is positive ($\hat{p}$) (i.e., the fraction of cases where the index is positive). A probability of 70% or more is considered promising in the studies, which is more robust to noisy indices than 50%. The results may change for borderline heuristics depending on the choice of threshold, which would be user-selected based on the user's risk aversion. Other approaches to handle noise in the index without spending too many function evaluations would be possible such as bootstrapping to generate several smaller subsets of the dataset, or checking if the index is greater than a threshold $I_{\min} > 0$ that is a function of the variance seen in the data.

There are some special cases to consider: (1) If a problem goal shows no variability in the dataset, then one cannot compute the corresponding correlation coefficients in WII. This could be because: (a) the goal is too easy to satisfy, but then its weight would be zero so this case is not a problem; (b) the goal is too hard to satisfy, in which case it is suggested to increase the size of the dataset to increase the chances of finding feasible designs; (c) the goal takes the same "intermediate" value all the time, in which case one can simply skip that goal in the computation since the heuristic does not affect that goal. (2) If a heuristic's soft constraint is always fully satisfied and thus shows no variability in the dataset, one can also not compute the corresponding correlation coefficients. In that case, one can conclude that the heuristic is not promising. (3) If a repair operator's IF statement is never satisfied (i.e., there is nothing to repair in any design in the dataset), then the heuristic is never applied, and HDI = 0, which reasonably suggests that the heuristic is not promising.

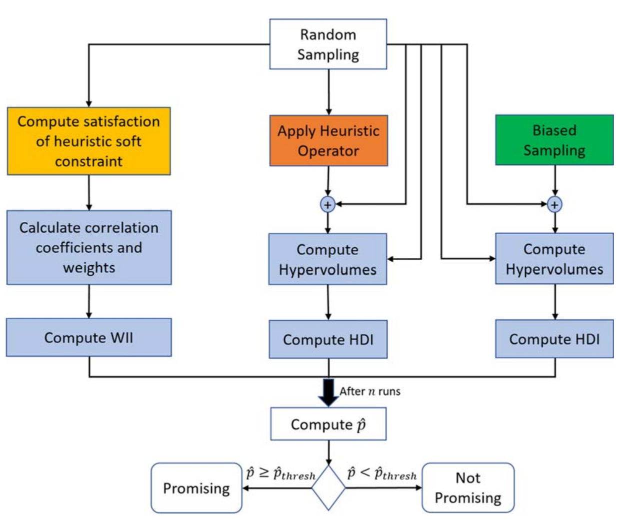

Fig. 4 Flowchart showing the steps involved in the screening of a single heuristic

For example, for the metamaterial soft constraint screening studies, none of the randomly generated designs across the ten datasets fully satisfy $g_{feas} \& g_{stiff}$ (special case 1b above). Therefore, in addition to 100 randomly generated designs, 100 designs from the middle stages of the optimization and the final population from an ϵ-MOEA run with an initial population of 100, ϵ bounds of [0.01, 0.01], and maximum of 6000 function evaluations are added to the datasets to total 300 designs each in order to more accurately capture correlations with the hard-to-satisfy equality constraints. The operator and biased sampling screening studies for the metamaterial use ten datasets of 300 randomly generated designs each.

4.2 Assessing Efficacy of Promising Heuristics.

To compare the performance of the optimization algorithm with and without the enforcement of the promising heuristics identified in the previous subsection, three cases are considered in general: (1) no heuristics enforced; (2) all heuristics enforced; and (3) only promising heuristics enforced. As discussed in Sec. 2, each heuristic representation is leveraged in the design optimization framework using different methods. The chosen leveraging methods for the soft constraint, operator, and biased sampling forms are interior penalty, AOS, and warm starting respectively.

4.2.1 Soft Constraint Forms—Interior Penalty. The interior penalty method [47] adds a penalty term to the design objectives based on the degree of violation of the heuristics. The heuristic-penalized objectives $\boldsymbol{f_h}(\boldsymbol{x})$ are then used for non-dominated sorting in the ϵ-MOEA algorithm. Equation (6) shows the form of the penalized objectives.

$$\boldsymbol{f}_h(\boldsymbol{x}) = \boldsymbol{f}_{pen}(\mathbf{x}) + \boldsymbol{1}_t \odot w \frac{\sum_{i=1}^{u} a_i(\boldsymbol{x})}{u} \tag{6}$$

Here t is the number of objectives, $a_i(\boldsymbol{x})$ is the degree of satisfaction of the ith heuristic by design $\boldsymbol{x}$, and u is the number of enforced heuristics. The weight parameter w represents the relative priority for heuristic satisfaction relative to penalized objective minimization. The optimal value of w is hard to predict and must be learned through multiple trials. This is one of the main shortcomings of fixed weighted optimization. The same heuristic-weighted penalty is applied to all objectives. It is also possible to weigh each heuristic penalty differently and proportionally to the heuristic's impact index. An alternative approach to learn the optimal weights would be to use a co-evolution method [68] in which a population of the heuristic penalty weights is evolved in addition to the main population of designs, and the fitness of the population of weights depends on the fitness of the designs found using the corresponding weights.

4.2.2 Repair Operator Forms—Adaptive Operator Selection. Since the performance of AOS may depend strongly on the choice of credit assignment and operator selection strategy, the three credit assignment strategies and two operator selection strategies benchmarked in Ref. [54] were considered. The credit assignment strategies are: offspring-dominates-parent (where an operator receives +1 credit if offspring dominates parent, 0.5 if non-dominated), set improvement (where an operator receives +1 credit if the offspring enters the archive), and set contribution (where the offspring receives +1 credit for every iteration in which a design generated by it remains in the archive). For operator selection strategies, probability matching and adaptive pursuit were considered. Probability matching [69] (shown in Eq. (7)) maintains a quality metric q_i with a hyperparameter α that determines the relative weight of current versus past credits c_i. The probability of selection of heuristic i pr_i is then updated based on q_i. In general, pr_i is not allowed to go below a $p_{\min}$ hyperparameter to ensure some level of exploration.

$$q_{i,t+1} = (1-\alpha)q_{i,t} + \alpha c_{i,t} \tag{7}$$

$$pr_{i,t+1} = p_{\min} + (1 - |O|p_{\min})\frac{q_{i,t+1}}{\sum_{j=1}^{|O|} q_{j,t+1}} \tag{8}$$

Adaptive Pursuit [70] is a greedier version of probability matching that defines a maximum probability of selection $p_{\max}$ based on the number of operators and the selection probabilities of the operators are updated such that the best performing operator asymptotically pursues $p_{\max}$ whereas the others pursue $p_{\min}$. In the version of AOS used in this work, $p_{\min}$ is reduced by a small amount every set number of iterations until it reaches zero, assuming that if heuristics are effective mostly at the beginning of the search, they should be discouraged in the later stages.

The best performing strategies for each problem were chosen. For the satellite problems, set improvement dominance is used as the credit assignment strategy. Adaptive pursuit is used as the operator selection strategy with $\alpha = 0.8$, $\beta = 0.8$, and $p_{\min} = 0.1$. For the metamaterial design problems, offspring-dominates-parent is used as the credit assignment strategy. Probability matching is used as the operator selection strategy with $\alpha = 0.6$ and $p_{\min} = 0.03$. In all four problems, every 500 function evaluation, $p_{\min}$ is reduced by 0.01 until it reaches zero.

4.2.3 Biased Sampling Forms—Warm Starting. The biased sampling forms of heuristics are used to generate the initial population for the optimization algorithm. The optimization proceeds normally after that.

4.2.4 Efficacy Study. Figure 5 shows the procedure of the efficacy study to determine the effect of incorporating all, promising, or no heuristics in the optimization framework. Thirty runs of each of the three cases (no heuristics, all heuristics, and promising heuristics) are conducted for comparison. The selected constraint handling mechanism for the ϵ-MOEA algorithm is a modified dominance operator for non-dominated sorting where designs are compared by aggregate constraint satisfaction first, and by Pareto dominance in case of ties.

For the EOSS problems, the ϵ-MOEA runs use a population size of 300, with ϵ bounds of [0.01, 0.01], and termination criteria of 5000 function evaluations. One-point crossover with probability of 1 and bit-flip mutation with probability of $\frac{1}{60}$ for the assignment problem and $\frac{1}{24}$ for the partitioning problem are used as the knowledge-independent operators. Specialized crossover and mutation operators are employed for the partitioning problem to ensure that the architectures after operation are feasible. The optimization parameters come from Hitomi et al. [71] which explored the assignment problem.

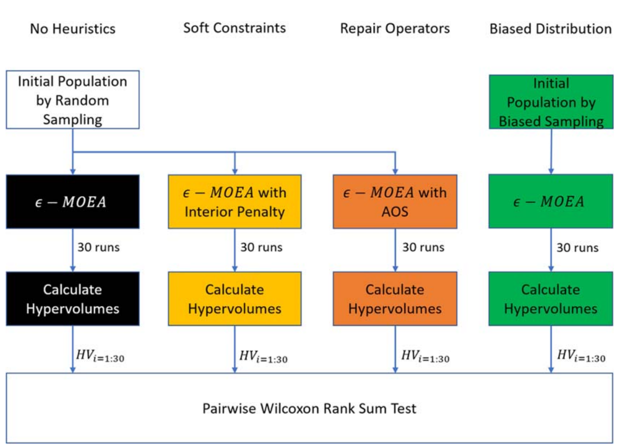

Fig. 5 Flowchart illustrating the various stages in the efficacy study for different heuristic representations

For the metamaterial problems, the ϵ-MOEA runs are conducted with a population size of 100, ϵ bounds of [0.01, 0.01], and termination criteria of 6000 function evaluations. One-point crossover with probability of 1 and bit-flip mutation with probability of $\frac{1}{30}$ are used as the knowledge-independent operators. Similar optimization parameters have been used to explore the equal normal stiffness problem in Kumar et al. [41].

The different bit-flip probabilities arise from the different chromosome lengths for different problems and the common rule of thumb that mutation probability should be about 1/(*chromosome length*).

The single-tailed Wilcoxon rank sum test with $p = 0.05$ is used to compare HVs between cases enforcing no heuristics, all heuristics, and only the promising heuristics. For all problems, HV values are tested at 250 number of function evaluations (NFE) intervals from 0 to 1000 (both included) and at every 500 NFE intervals up to and including the end of the optimization. The HV of designs that fully satisfy all constraints is used for performance testing and visualization.

5 Results

The results of the screening and efficacy studies for the four design problems are presented in this section. Correlation coefficients for heuristic-problem goal pairs and fraction of satisfaction for each constraint and heuristic as well as the efficacy study statistics and additional plots for all four problems are available in the Supplementary Materials available in the Supplemental Materials.

5.1 Screening Study.

Figure 6 shows box plots of the distribution of the impact indices for each heuristic form for the four problems. Results show that not all heuristics are promising, which emphasizes the need for robustness in identifying the promising heuristics. The results for each case study are discussed individually below.

5.1.1 Artery Problem. Soft constraints: Neither the partial collapsibility nor the nodal properties soft constraints are found to be promising ($\hat{p} = 0.0$ for PC and 0.2 for NP) whereas the orientation and intersection soft constraints are both promising ($\hat{p} = 1.0$ for both.) This makes sense since both orientation and intersection are aligned with distance to PF in addition to the constraints they are targeting (see Table in Supplementary Materials available in the Supplemental Materials with correlation coeffs.) On the other hand, both partial collapsibility and nodal properties show negative alignment with distance to PF and feasibility.

Repair Operators: The $\hat{p}$ for partial collapsibility, nodal properties, orientation, and intersection repair operators are 0.4, 0.3, 0.7, and 1.0 respectively. Thus, the orientation and intersection repair operators are promising, as found with the corresponding soft constraints.

Biased Sampling: The $\hat{p}$ for orientation biased sampling is found to be 0.7, which makes it narrowly promising for this problem.

5.1.2 Equal Stiffness Problem. Soft constraints: The partial collapsibility and nodal properties soft constraints are non-promising ($\hat{p} = 0.0$ for both) whereas the orientation and intersection soft constraints are promising ($\hat{p} = 0.9$ and 1.0 respectively), as in the artery problem.

Repair Operators: The partial collapsibility and nodal properties repair operators have a $\hat{p}$ of 0.5 and 0.6 respectively and are thus non-promising. The orientation repair operator has a $\hat{p}$ of 0.6, making it narrowly non-promising, unlike in the

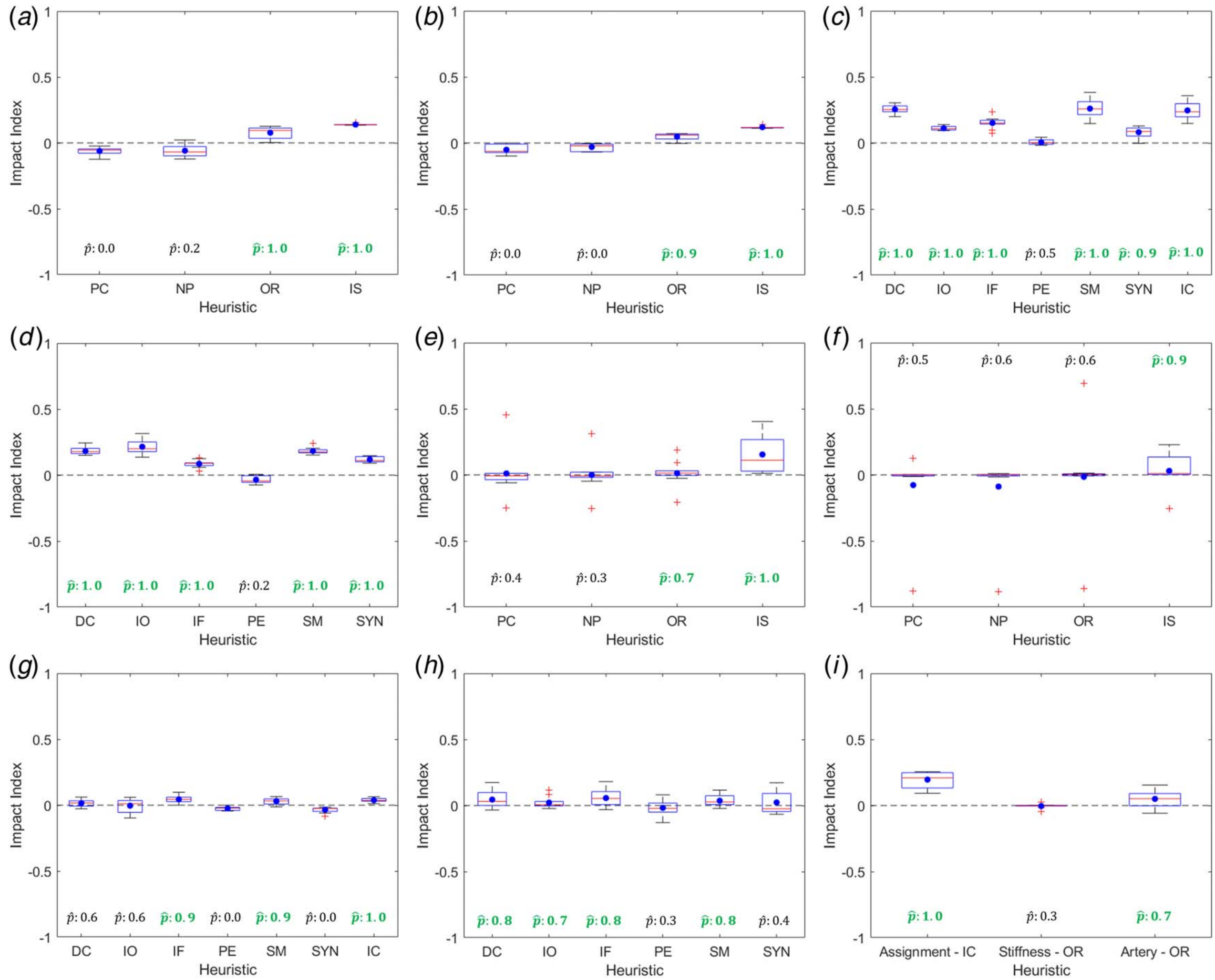

Fig. 6 Distributions of the impact indices for the heuristic forms of the four test problems shown as boxplots with the $\hat{p}$ values and the mean impact index represented as blue dots. The $\hat{p}$ values of the promising heuristics are shown in green and bold: (*a*) artery—soft constraints, (*b*) equal stiffness—soft constraints, (c) assignment—soft constraints, (*d*) partitioning—soft constraints, (*e*) artery—repair operator, (*f*) equal stiffness—repair operator, (*g*) assignment—repair operator, (*h*) partitioning—repair operator, and (*i*) biased sampling.

artery problem. This is likely because a stiffness ratio of 1 is easier to achieve randomly than the arbitrary target of the artery problem. The intersection repair operator is still promising with a $\hat{p}$ of 0.9.

Biased Sampling: The orientation biased sampling heuristic is not promising with a $\hat{p}$ of 0.3, probably also because the 1 target stiffness ratio is easy to obtain randomly.

5.1.3 Assignment Problem. Soft constraints: Duty cycle, instrument-orbit relations, interference, spacecraft mass, and instrument count have positive index values for all ten datasets. Synergy and packing efficiency have probability of positive index of 0.9 and 0.5 respectively. Hence, all heuristic soft constraints except packing efficiency are considered promising for the assignment problem. While improving packing efficiency reduces launch cost, it is possible that it also degrades other aspects of lifecycle cost or science, leading to a net negative effect.

Repair operators: Both the interference and spacecraft mass repair operator forms have a probability of positive index of 0.9. Instrument count repair operator has all positive index values and packing efficiency and synergy repair operators have negative index values for all ten datasets. Both duty cycle and instrument-orbit relations repair operators have probability of positive index of 0.6. Thus, the promising heuristic repair operators are interference, spacecraft mass, and instrument count, although duty cycle and instrument-orbit relations are almost promising. A potential reason why the synergy operator is not promising is that it works by adding a synergistic instrument to the design, which can lead to too low resources.

Biased sampling: The impact index for the instrument count biased sampling heuristic is positive for all ten datasets, implying that it is promising for this problem, as expected.

5.1.4 Partitioning Problem. Soft constraints: All heuristic soft constraints except packing efficiency have all positive index values across the datasets and are thus promising. The packing efficiency soft constraint form is not promising ($\hat{p} = 0.2$), as in the assignment problem.

Repair operators: Duty cycle, instrument-orbit relations, interference, and spacecraft mass repair operators have probability of positive index of 0.8, 0.7, 0.8, and 0.8 respectively whereas packing efficiency and synergy repair operators have probability of positive index of 0.3 and 0.4 respectively, probably because of the same reasons mentioned earlier.

5.2 Efficacy Study. Table 1 shows the results of the efficacy study for the three heuristic representations incorporated into the four case studies. The use of the rejection of the null hypothesis for at least one of the test NFEs to indicate performance

Table 1 Summary of the results of the efficacy studies for the three forms of the heuristics and the four test problems as tested pairs of cases

Design problem	Soft constraints (Int. Pen.)	Repair operators (AOS)	Biased sampling (Bias. Init.)
Artery	**Prom. > All;** **Prom. > ϵ-MOEA** **All > ϵ-MOEA**	**Prom. > All;** **Prom. > ϵ-MOEA;** **All $\not>$ ϵ-MOEA**	Prom. = All $\not>$ ϵ-MOEA
Equal stiffness	Prom. $\not>$ ϵ-MOEA; Prom. $\not>$ All; All $\not>$ ϵ-MOEA	**Prom. > All;** **Prom. > ϵ-MOEA;** **All $\not>$ ϵ-MOEA**	All $\not>$ ϵ-MOEA = Prom.
Assignment	**Prom. > ϵ-MOEA;** **All > ϵ-MOEA;** **Prom. $\not>$ All**	**Prom. > All;** **Prom. > ϵ-MOEA;** **All > ϵ-MOEA**	**Prom, = All > ϵ-MOEA**
Partitioning	Prom. $\not>$ ϵ-MOEA; Prom. $\not>$ All; All $\not>$ ϵ-MOEA	**Prom. > ϵ-MOEA;** **All > ϵ-MOEA;** **Prom. $\not>$ All**	—

Note: Cases with an improvement in performance due to the promising heuristics (rejection of the HV null hypothesis) for at least one of the test NFEs are shown with > and in bold font. Case pairs for which the null hypothesis could not be rejected are shown with .

improvement between different test cases can be justified since in most cases, heuristics are useful in the beginning of the search up to a certain NFE. It is also worth noting that the number of function evaluations one can afford for a given problem depends on available optimization time, computational resources, and infrastructure and the computational cost of models so this approach, although optimistic, seems reasonable. In general, we see that heuristics help in most cases, especially repair operators. The results for each case study are discussed individually in the following.

5.2.1 Artery Problem. For the enforcement of the heuristic soft constraints using interior penalty, out of the trials conducted using penalization weights $w = 0.1$, 1, 10 the best results were obtained for $w = 10$ and those are presented here.

Figure 7 compares the evolution of HV as a function of NFE for the three cases (all, promising, or no heuristics) for both soft constraints (Fig. 7(*a*)) and repair operators (Fig. 7(*b*)). The HV for fully feasible designs as a function of NFE for the biased sampling cases is shown in Fig. 8(*a*).

Generally, it is observed that enforcing only the promising heuristics, in the case of both the soft constraint and repair operator forms, results in a decrease in NFE required to find the first fully feasible design (i.e., faster jump in the HV plot) compared to no heuristics and to all heuristics—i.e., there are some NFE savings for a fixed (low) computational budget.

Soft constraints: Figure 7(*a*) shows the statistics of the HV for 30 runs for the three cases. The lines show the median HV values for each case while the shaded regions show the inter-quartile ranges. The increase in the median HV line and a portion of the inter-quartile range for the promising heuristics case at lower NFE values compared to the other two cases in the embedded plot of Fig. 7(*b*) shows that enforcing only the promising heuristics leads to higher HV values for the same NFE for most runs compared to enforcing all heuristics or enforcing no heuristics. The HV for the promising heuristics case is significantly higher than the ϵ-MOEA HV at NFE = 750 ($U = 263.5$, $p = 0.003$) and 1000 ($U = 247$, $p = 0.001$) and significantly higher than the all heuristics case at NFE = 500 ($U = 304$, $p = 0.01$) and 750 ($U = 265$, $p = 0.003$). Additionally, HV for the all heuristics case is significantly higher than the ϵ-MOEA HV at NFE = 1000 ($U = 331$, $p = 0.04$).

The difference in the median HV jump NFE between enforcing no heuristics (750) and enforcing only promising heuristics (600) is 150, which is slightly less than the 300 designs on average required to determine the set of promising heuristics. Since the feasibility constraint is hard to satisfy, more feasible designs had to be evaluated as well so in this case the method does not improve efficiency overall. However, the NFE at which the initial jump in HV is observed (which corresponds to the discovery of the first design satisfying all constraints) from enforcing all heuristics is the same as enforcing no heuristics.

Repair operators: Similar HV trends to the soft constraints case are observed in Fig. 7(*b*). The HV for promising heuristics is significantly higher than the ϵ-MOEA HV at NFE = 500 ($U = 331$, $p = 0.03$), 750 ($U = 257$, $p = 0.002$), and 1000 ($U = 270.5$, $p = 0.004$). Additionally, the HV for the promising heuristics case is significantly higher than the HV for the all heuristics case at NFE = 500 ($U = 349$, $p = 0.05$) and 750 ($U = 339$, $p = 0.05$). The difference in the median HV jump NFE between enforcing no heuristics (750) and enforcing only promising heuristics (500) is 250, which is slightly less than the 300 random designs used in the impact index computation—i.e., the method does not improve efficiency overall. However, the HV jump NFE by enforcing all heuristics is 700, resulting in net NFE savings of 200.

Biased sampling: The orientation biased sampling initialization did not improve HV and has similar performance to the ϵ-MOEA case. The heuristic enforced case results in a smaller inter-quartile range of HV values as shown in Fig. 8(*a*) compared to the ϵ-MOEA case. The HV for the heuristic enforcement case is not significantly higher than ϵ-MOEA HV at any of the tested NFE.

5.2.2 Equal Stiffness Problem. Multiple trials for the enforcement of heuristics using interior penalty were performed with the penalization weights of 0.005, 0.05, 0.01, 0.1, 0.5, 1, and 10. The best results were obtained for $w = 0.05$ and are presented here.

Figures 7(*c*), 7(*d*), and 8(*b*) show the results for the soft constraint, repair operator, and biased sampling representations respectively.

Soft constraints: Figure 7(*c*) shows that enforcing only the promising heuristics demonstrates an improvement in performance over enforcing all heuristics, seen by the jump in the median HV curve for the promising heuristics case compared to the all heuristics case. However, there is no discernible improvement in performance compared to the ϵ-MOEA case. The Wilcoxon test on the HV values for the three cases also does not show significant performance improvements between any pair of cases at any of the tested NFE.

Repair operators: Figure 7(*d*) illustrates the improvement in performance by enforcing only the promising heuristic repair operators. The promising heuristics case is able to find designs satisfying all constraints faster, as seen in the comparison of the point of jump in the median HV curves. The difference in the NFE at which the jump takes place for the promising heuristics (950) and ϵ-MOEA (3850) cases is 2900, which is much more than the 300 function evaluations required on average to ascertain the set of promising heuristics. The HV for the

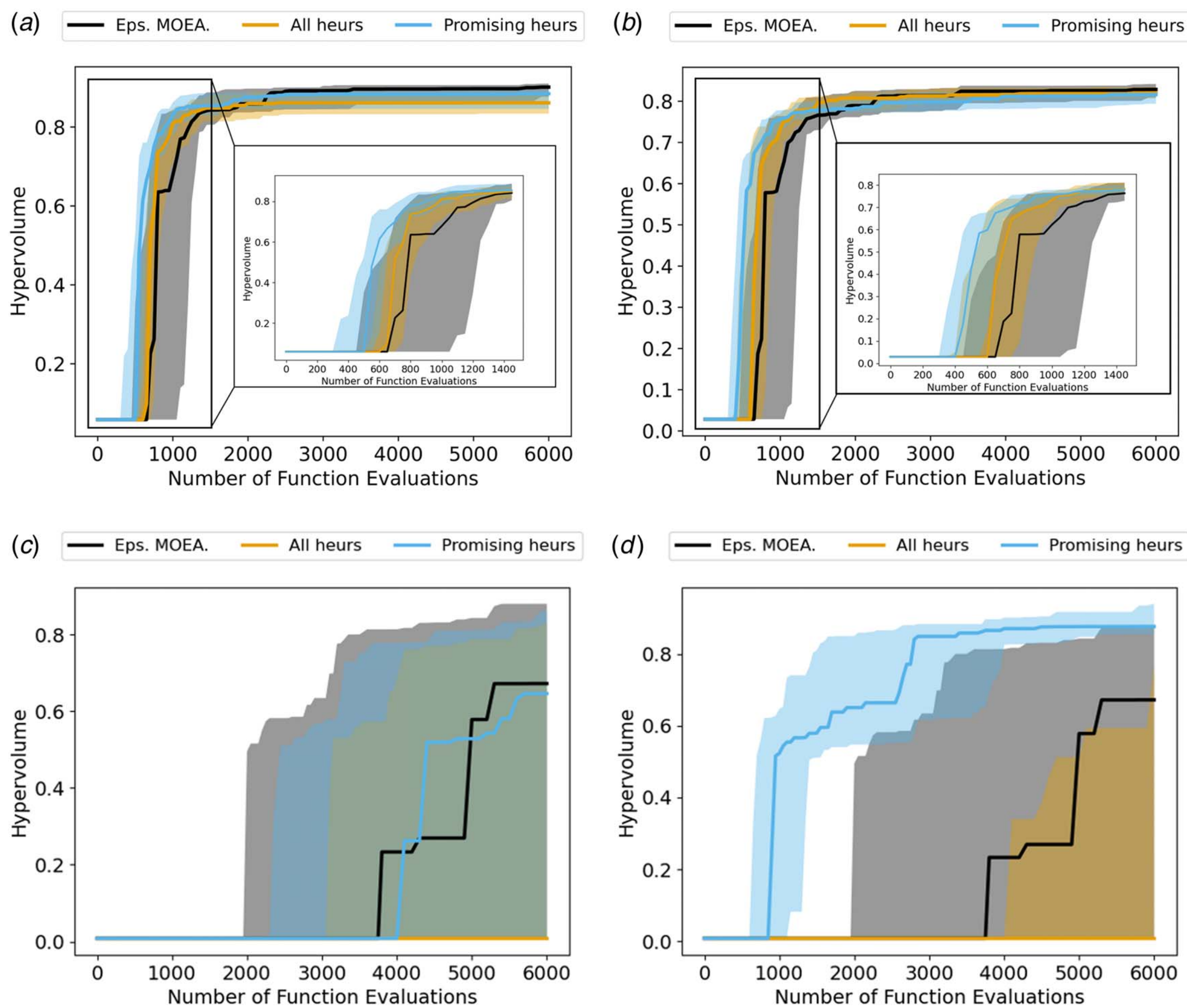

Fig. 7 HV as a function of time (NFE) for the three cases in the metamaterial problems. The lines represent the median values, the shaded regions represent the inter-quartile ranges and only designs satisfying all constraints are considered: (*a*) artery—soft constraints, (*b*) artery—repair operators, (*c*) equal stiffness—soft constraints, and (*d*) equal stiffness—repair operators.

promising heuristics case is significantly higher than for both the all heuristics and the ϵ-MOEA from 500 NFE onwards (all p-values are ≤ 0.01).

Enforcing all the heuristics (as either soft constraints or repair operators) results in more than half of the runs failed to reach fully satisfying designs, thus illustrating the merit of identifying the promising heuristics to enforce.

Biased sampling: The orientation biased initialization strategy did not significantly improve HV and in fact, hurts performance as seen by comparing the mean HV curves for the heuristic enabled and ϵ-MOEA cases. Any advantage of improved orientation satisfaction in the initial population appears to be immediately lost after application of other operators.

5.2.3 Assignment Problem. Penalization weights (w in Eq. (6)) of 1 and 0.1 were tried for the enforcement of the heuristic soft constraints using interior penalty. The best results were obtained for $w = 0.1$ and are presented here.

Figures 9(*a*), 9(*b*), and 8(*c*) show the results for the assignment problem's repair operators+soft constraints and biased sampling respectively.

Soft constraints: Comparing the inter-quartile ranges in Fig. 9(*a*) shows that enforcing only the promising heuristics leads to the design search obtaining better performing designs for most runs compared to enforcing all heuristics and no heuristics. As a result, the HV for the promising heuristics case is significantly higher than the ϵ-MOEA HV for NFE = 1000 ($U = 325$, $p = 0.03$), 1500 ($U = 293$, $p = 0.01$), 2000 ($U = 285$, $p = 0.008$), and 2500 ($U = 337$, $p = 0.05$). Additionally, the HV for the all heuristics case is significantly higher than the ϵ-MOEA HV for NFE = 1500 ($U = 338$, $p = 0.05$), 2000 ($U = 326$, $p = 0.03$), and 2500 ($U = 317$, $p = 0.02$). This can be explained by the fact that since only packing efficiency is not promising, there is not a huge improvement in performance between the promising and all cases. However, removing the non-performing heuristic quickens the improvement in performance over ϵ-MOEA. The NFE savings depend on the level of performance (median HV). To achieve an HV = 0.88 for the all heuristics case 500 fewer NFE are needed, compared to only 300 fewer for the promising heuristics case in comparison with ϵ-MOEA. The promising heuristics case also performs better for HV=0.87. In contrast, the all heuristics case performs better for HV = 0.85 and 0.86. At HV = 0.70, both heuristics enforcement cases show NFE savings of 100 with respect to ϵ-MOEA. In general, both heuristics enforcement cases are evenly matched in terms of NFE savings at most values of HV. But these savings are less than the 300 designs on average used in the screening study, i.e., the method does not improve efficiency overall.

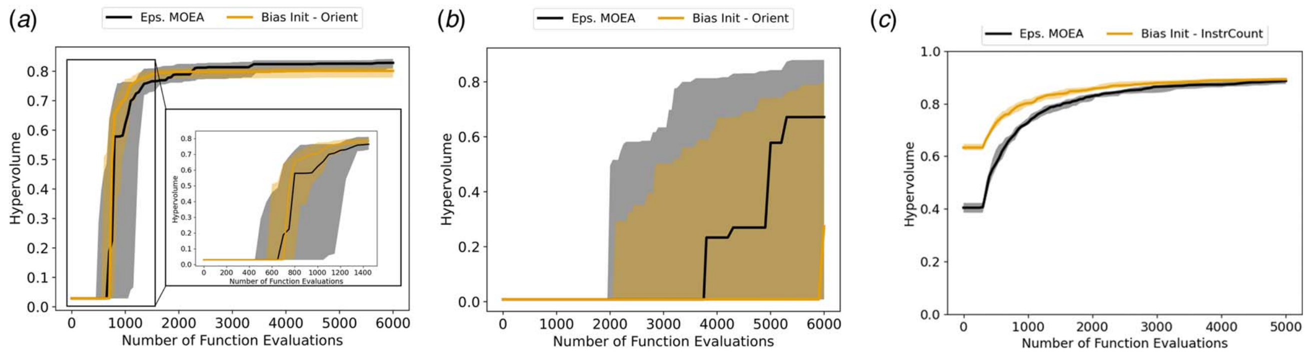

Fig. 8 Performance comparison between cases for the metamaterial design problems with orientation and for the assignment problem with instrument count violation enforced as a biased initial population. The lines represent the median values, the shaded regions represent the inter-quartile ranges, and only designs that fully satisfy the constraints are considered: (*a*) artery problem, (*b*) stiffness problem, and (*c*) assignment problem.

Repair operators: The improvement in performance by enforcing the heuristic repair operators is apparent in the comparison of the inter-quartile ranges of HV for the three heuristics enforcement cases in Fig. 9(*b*). The HV for the all heuristics case is significantly higher than the ϵ-MOEA HV at all tested NFEs except at NFE = 0, 250, and 2000 (all p-values are ≤ 0.04, except at 2500 NFE where $p = 0.05$). The promising heuristics HV is significantly higher than the ϵ-MOEA HV at all tested NFE except at NFE = 0, 250, and 5000 (all p-values are ≤ 0.02 except at 4500 NFE where $p = 0.05$). The promising heuristics HV is significantly higher than the all heuristics HV at NFE = 1000 ($U = 323$, $p = 0.03$), 1500 ($U = 338$, $p = 0.05$), 2000 ($U = 322$, $p = 0.03$), and 2500 ($U = 341$, $p = 0.05$). The maximum NFE savings with respect to ϵ-MOEA for the all heuristics case is 1000 for median HV = 0.89, compared to 800 for the promising heuristics case. In contrast, the maximum NFE savings by enforcing the promising heuristics is 900 for median HV = 0.88 which is greater compared to 700 NFEs saved by enforcing all heuristics. For median HV = 0.70 the promising heuristics case saves 200 NFEs whereas the all heuristics case saves only 150 NFEs. However, these savings are again less than the 300 designs on average used in the screening study.

Biased sampling: The clear separation between the inter-quartile ranges for the two cases in Fig. 8(*c*) demonstrates the clear superiority of the biased initialization case over ϵ-MOEA. The biased sampling case HV is significantly higher than the ϵ-MOEA HV for all tested NFE (all p-values are ≤ 0.01). The median HV difference between the two cases at 0 NFE is 0.2284 and ϵ-MOEA takes more than 5000 NFE to catch up, thus illustrating the high savings in NFE by enforcing the instrument count biased sampling form.

Overall, all methods show significant improvements in efficiency for the assignment problem, although the gains are much larger for biased sampling than for the other two forms.

5.2.4 Partitioning Problem. To enforce the heuristic soft constraint forms, different trials of interior penalty were conducted with penalization weights of 0.005, 0.1, 1, and 10. The best results were obtained for $w = 0.1$ and are shown here.

Figures 9(*c*) and 9(*d*) show the results for the soft constraints and repair operators respectively.

Soft constraints: Comparing the inter-quartile ranges of HV for the three heuristic enforcement cases in Fig. 9(*c*) shows that there is no significant improvement in performance obtained by enforcing the heuristic soft constraints. None of the HV values is significantly higher between any case pairs at all tested NFE.

Repair operators: Comparing the inter-quartile ranges for the three heuristic enforcement cases in Fig. 9(*d*) does not show any noticeable improvement in performance by the enforcement of the repair operators at the beginning of the search. However, the promising heuristics HV is significantly higher than the ϵ-MOEA HV at NFE = 3500 ($U = 329$, $p = 0.04$), 4000 ($U = 297$, $p = 0.01$), 4500 ($U = 304$, $p = 0.02$), and 5000 ($U = 324$, $p = 0.03$). Additionally, the all heuristics HV is significantly higher than the ϵ-MOEA HV at NFE = 4000 ($U = 338$, $p = 0.05$), 4500 ($U = 306$, $p = 0.02$), and 5000 ($U = 297$, $p = 0.01$). The promising heuristics HV is not significantly higher than the all heuristics HV. Interestingly, the performance improvement happens towards the end of the optimization, which is opposite to the usual behavior of heuristics. The maximum NFE savings by enforcing only the promising heuristics (700) is seen for median HV = 0.91, in comparison to only 600 by enforcing all heuristics. Additionally, 300 NFEs are saved by enforcing only the promising heuristics to reach median HV = 0.66, whereas no savings are observed by enforcing all heuristics. However, these savings are not seen for both heuristic enforcement cases for other HV values. Overall, for the partitioning problem, the repair operator heuristics lead to small but significant gains in efficiency after a few thousand NFEs, whereas soft constraints do not seem to improve efficiency at any point.

6 Discussion

6.1 Main Findings. Table 2 summarizes the efficacy study results for all problems and all heuristic representations as a comparison of the NFE savings (if any) for the promising heuristics case compared to the ϵ-MOEA case with the 300 NFE used in the screening study. Nearly all cases except the equal stiffness problem and partitioning problem soft constraint cases show gains in enforcing only the promising heuristics as compared to enforcing no heuristics. The greatest NFE gain is observed in the assignment problem (biased sampling function case) and the equal stiffness problem (repair operator case). Starting with a lower number of instruments on average (as enforced by the instrument count biased sampling form) aided in finding the optimal balance between allocation of instruments to satellites to satisfy the stakeholder requirements and incurring high cost. It is also worth noting that although enforcing the promising heuristics in the Assignment problem soft constraint and repair operator cases shows NFE savings greater than 300 at high HV values (0.88 for both cases), this is contrary to the assumption that heuristics are useful at the beginning of the search up to a certain NFE. The partitioning problem repair operator case also shows similar results, however the promising heuristics show NFE savings of 300 at HV = 0.66.

Overall, it can be concluded that the impact indices are able to identify the promising heuristics which are found to be effective for most cases. That said, the gains are more significant in some problems than in others, and they depend on the computational budget. It must be noted here that although NFE is computed similarly for three efficacy study cases, the incorporation of repair operators through AOS does increase runtime slightly compared

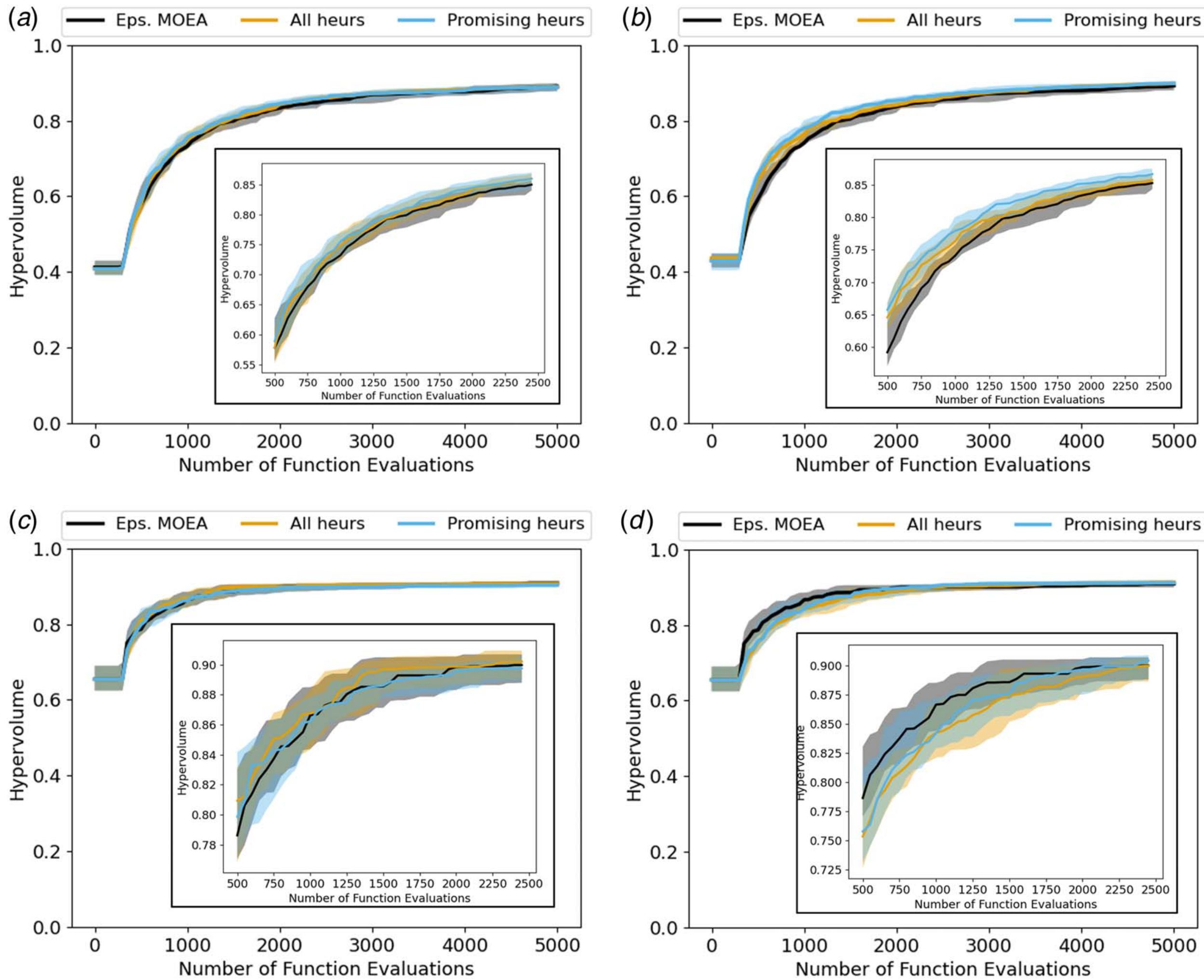

Fig. 9 HV as a function of time (NFE) for the three cases in the EOSS problems. The lines represent the median values, the shaded regions represent the inter-quartile ranges: (*a*) assignment—soft constraints, (*b*) assignment—repair operators, (*c*) partitioning—soft constraints, and (*d*) partitioning—repair operators.

to the ϵ-MOEA case owing to the computational overheads pertaining to the knowledge-dependent design manipulation, operator credit assignment, and selection. In this paper, we assume that the cost of executing a heuristic is negligible compared to the cost of executing the design evaluation model, which we believe is a reasonable assumption for real-world engineering problems. The computational overheads associated with the soft constraint and biased sampling forms are negligible compared to the ϵ-MOEA design search.

Large differences in performance are seen between different heuristic representations for the different problems, with repair operators being the most robust overall. This can be attributed to the details of how the heuristics are implemented as well as how they are incorporated into the optimization framework. Soft constraint forms continuously impact the design search since the heuristic violation penalizations to the objectives are always levied. In comparison, there can be stages in the optimization algorithm wherein the heuristic repair operator is not applied and hence is not directly impacting the design search. Moreover, interior penalty, which is used to enforce the heuristic soft constraints, has the inherent shortcoming that the penalization weight is a hyperparameter that must be fixed using trials.

The results for the partitioning and stiffness problems show that there is some utility in continuing to utilize heuristics that improve both objective minimization and constraint satisfaction throughout the optimization process and not just at the beginning.

6.2 Designer Guidelines. Based on the results of the case studies, seven guidelines for designers to incorporate design heuristics are presented. These guidelines are most useful for whomever is in charge of either developing the search tool or configuring it for a given problem, which in some cases may also be the user of the tool (i.e., the "designer" or "systems engineer") but not necessarily.

(1) Many heuristics "look good" but are actually not useful for a given problem. Thus, employ the screening study to identify the promising heuristics. For example, nodal properties were non-promising for both metamaterial problems even though it was aimed at increasing the satisfaction of the connectivity constraint. Additionally, the screening study is especially useful if many heuristic repair operators are available. Even though AOS is an online adaptive operator enforcement method, it may take a lot of function evaluations for AOS to learn the useful heuristics for a given problem.
(2) Heuristic representation and implementation details matter. Different representations of the same general heuristic may have very different behaviors (see results of the EOSS study). Even the details of the heuristic implementation (e.g., the "move size" or a threshold parameters for a repair operator) may have a large impact on heuristic performance.
(3) Heuristics that help with hard to satisfy problem goals (e.g., constraints) have great potential to be promising for the

Table 2 Summary of the results of the efficacy studies for the three forms of the heuristics and the four test problems as tested pairs of cases

Design problem	Soft constraints (Int. Pen.)	Repair operators (AOS)	Biased sampling (Bias. Init.)
Artery	150	250	×
Equal Stiffness	×	**2900**	×
Assignment	100 (at $HV = 0.70$)	200 (at $HV = 0.70$)	**>5000**
Partitioning	×	**300 (at $HV = 0.66$)**	–

Note: Here the NFE savings by enforcing the promising heuristics compared to enforcing no heuristics is shown for each case (including the test HV as is applicable), with the savings ≥ the 300 NFEs on average required for the screening study in bold font and × implies no performance improvement.

problem. For example, for the metamaterial problems, orientation and intersection in its various representations aided in reaching the fully satisfying designs faster, whereas many ϵ-MOEA runs could not find a single fully feasible design.

(4) In the presence of very-hard-to-satisfy constraints, there can be high variability in the screening study results. Designers are advised to rerun the screening study using different datasets to assess robustness.

(5) In constructing heuristic repair operators, incorporating randomness into the operation of the heuristic will help reduce the exploitation of its inherent knowledge and may help maintain appropriate balance between exploration and diversity.

(6) Repair operators seem to be a more robust way overall of incorporating heuristics compared to the other forms we studied. The implementation as soft constraints has some limitations. First, the fixed-weight interior penalty method we used continuously enforces the heuristics even in design regions where they may not be promising and thus hurt the design search (although that could be alleviated if an adaptive penalty method is used such as Ref. [68]). Second, in cases with very hard to satisfy constraints, the screening study for soft constraints required a large number of NFE to find designs with varying degrees of constraint satisfaction.

(7) Incorporate the promising heuristics using an adaptive approach such as AOS in repair operators. This approach can "shut down" heuristics selectively when they stop being useful for the problem, and restart them if they become useful again (as seen in the partitioning problem).

7 Conclusions and Future Work

This paper is, to the best of our knowledge, the most comprehensive study on the use of design heuristics to improve the efficiency of multi-objective design optimization. A novel approach to identify the promising heuristics from a set of candidate heuristics for a given design problem is introduced, which utilizes two new impact indices that capture the aggregate effect of a heuristic on a given problem's goals (objectives and constraints).

The approach is tested and validated on four multi-objective design problems: two unconstrained EOSS design problems and two constrained metamaterial design problems. For six out of the eight cases (four design problems with heuristic soft constraint and repair operator forms for each), the identified promising heuristics were found to significantly improve the design optimization performance in comparison to enforcing all heuristics and enforcing no heuristics. For the remaining two cases, the performance of the promising heuristics was found to be at least as good as the enforcement of all heuristics. Results for the three cases of heuristic-biased distribution enforcement varied from a performance improvement for the Assignment problem, to a performance decline for the stiffness problem and no significant change in performance for the artery problem. Based on the results, some points of discussion and guidelines for designers were provided.

The study has several limitations. The scope of this work was limited to multi-objective evolutionary algorithms and specifically the baseline algorithm was ϵ-MOEA. Other baseline algorithms and how to incorporate heuristics into other search schemes (e.g., Bayesian optimization) should be explored.

All our findings and guidelines are based on four test problems and more work is to be done to ascertain the generalizability to other problems. In particular, all the problems are combinatorial optimization problems (i.e., they only have discrete design decisions) and problems with continuous variables should be studied.

The study used specific methods to incorporate the heuristics of various forms, namely interior penalty, adaptive operator selection, and warm starting. However, other methods exist to leverage the different design heuristic representations (e.g., a heuristic-biased sampling form can be incorporated into the evolutionary optimization framework as a biased crossover, mutation, or selection operator) and should be explored in the future. Additionally, as shown in Fig. 1, design optimization involves more than just searching: it also involves creating a problem formulation and evaluating designs. One could also consider heuristics for design evaluation, or model selection if multiple models of varying fidelities were available. Naturally, a lot of expert knowledge goes into developing a problem formulation (i.e., choice of design variables, objectives, constraints, etc.). Figure 1 also suggests some different ways to incorporate different heuristic representations into different steps of the optimization process (selecting designs, generating designs). For example, the heuristic-biased sampling function form can also be incorporated as a biased crossover or mutation operator. Finally, heuristics can also be used for selecting search operators—replacing the domain-independent adaptive operator selection strategies used in this work.

The study used a relatively small number of heuristics. Therefore, the results we obtained could be due to our choice and specific implementation of heuristics rather than the proposed methods. More heuristics should be studied, particularly for the biased sampling form. Although only a few heuristics in this work are represented as biased prior distributions, this is still consistent with the main goal of the paper, to demonstrate the efficacy of the methodology in identifying the promisingness of a heuristic. More broadly, the method should be applied to problems with a larger database of available heuristics. To work in practice, knowledge-based mechanisms (e.g., knowledge graph embeddings) may be used to down-select from a larger list of heuristics to a smaller list for the screening study.

The proposed indices evaluate the impact of heuristics one-at-a-time and therefore they do not capture potential interactions between heuristics (e.g., one heuristic "undoing the work" of the other). Future work will explore how to incorporate interactions into the indices.

Despite these limitations, overall, the results of the identification and leveraging of the promising heuristics for the four design problems show that the framework has potential in significantly speeding up the process of finding satisfying designs.

Acknowledgment

This work is supported and funded by the National Science Foundation under CMMI Nos. 1825444 and 1825521.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Yilmaz, S., and Seifert, C. M., 2011, "Creativity Through Design Heuristics: A Case Study of Expert Product Design," Des. Stud., **32**(4), pp. 384–415.

[2] Daly, S. R., Yilmaz, S., Christian, J. L., Seifert, C. M., and Gonzalez, R., 2012, "Design Heuristics in Engineering Concept Generation," J. Eng. Educ., **101**(4), pp. 601–629.

[3] Simpson, T. W., Spencer, D. B., Yukish, M. A., and Stump, G., 2008, "Visual Steering Commands and Test Problems to Support Research in Trade Space Exploration," 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, MAO, Victoria, British Columbia, Canada, Sept. 10–12, MAO, pp. 1–15.

[4] Chen, W., and Ahmed, F., 2021, "MO-PaDGAN: Reparameterizing Engineering Designs for Augmented Multi-Objective Optimization," Appl. Soft. Comput., **113**(Part A), p. 107909.

[5] Price, A. R., Keane, A. J., and Holden, C. M., 2011, "On the Coordination of Multidisciplinary Design Optimization Using Expert Systems," AIAA J., **49**(8), pp. 1778–1794.

[6] Mackerle, J., and Orsborn, K., 1988, "Expert Systems for Finite Element Analysis and Design Optimization-A Review," Eng. Comput., **5**(2), pp. 90–102.

[7] Witherell, P., Krishnamurty, S., and Grosse, I. R., 2007, "Ontologies for Supporting Engineering Design Optimization," ASME J. Comput. Inf. Sci. Eng., **7**(2), pp. 141–150.

[8] Zhou, B., Bao, J., Li, J., Lu, Y., Liu, T., and Zhang, Q., 2021, "A Novel Knowledge Graph-Based Optimization Approach for Resource Allocation in Discrete Manufacturing Workshops," Rob. Comput. Integr. Manuf., **71**, p. 102160.

[9] Hitomi, N., and Selva, D., 2018, "Incorporating Expert Knowledge Into Evolutionary Algorithms With Operators and Constraints to Design Satellite Systems," Appl. Soft Comput. J., **66**, pp. 330–345.

[10] Calvo, J. C., Ortega, J., and Anguita, M., 2011, "PITAGORAS-PSP: Including Domain Knowledge in a Multi-Objective Approach for Protein Structure Prediction," Neurocomputing, **74**(16), pp. 2675–2682.

[11] Fu, K. K., Yang, M. C., and Wood, K. L., 2016, "Design Principles: Literature Review, Analysis, and Future Directions," ASME J. Mech. Des., **138**(10), p. 101103.

[12] Lee, B., Fillingim, K. B., Binder, W. R., Fu, K., and Paredis, C. J. J., 2017, "Design Heuristics: A Conceptual Framework and Preliminary Method for Extraction," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, Aug. 6–9.

[13] Fillingim, K. B., Nwaeri, R. O., Borja, F., Fu, K., and Paredis, C. J. J., 2020, "Design Heuristics: Extraction and Classification Methods With Jet Propulsion Laboratory's Architecture Team," ASME J. Mech. Des., **142**(8), p. 081101.

[14] Moore, R. A., Romero, D. A., and Paredis, C. J., 2014, "Value-Based Global Optimization," ASME J. Mech. Des., **136**(4), p. 041003.

[15] Watson, B. C., Morris, Z. B., Weissburg, M. J., and Bras, B., 2022, "System of System Design for Resilience Heuristics Derived From Forestry Case Study Variants," Reliab. Eng. Syst. Safety, **229**, p. 108807.

[16] Yilmaz, S., Seifert, C., Daly, S. R., and Gonzalez, R., 2016, "Design Heuristics in Innovative Products," ASME J. Mech. Des., **138**(7), p. 071102.

[17] Puentes, L., Cagan, J., and McComb, C., 2021, "Data-Driven Heuristic Induction From Human Design Behavior," ASME J. Comput. Inf. Sci. Eng., **21**(2), p. 024501.

[18] Mehta, P., Malviya, M., McComb, C., Manogharan, G., and Berdanier, C. G. P., 2020, "Mining Design Heuristics for Additive Manufacturing Via Eye-Tracking Methods and Hidden Markov Modeling," ASME J. Mech. Des., **142**(12), p. 124502.

[19] Kr, G., Peters, I., Bingoel, B., and Stark, R., 2022, "Introduction of a Standardized Notation of Design Heuristics for Knowledge Formalization," Proceedings of the 33rd Symposium Design for X (DFX2022), Hamburg, Germany, Sept. 22–23.

[20] Jin, X., Evans, M., Dong, H., and Yao, A., 2021, "Design Heuristics for Artificial Intelligence: Inspirational Design Stimuli for Supporting UX Designers in Generating AI-Powered Ideas," Conference on Human Factors in Computing Systems, Yokohama, Japan, May 8–13.

[21] Fillingim, K. B., Nwaeri, R. O., Paredis, C. J., Rosen, D., and Fu, K., 2020, "Examining the Effect of Design for Additive Manufacturing Rule Presentation on Part Redesign Quality," J. Eng. Des., **31**(8-9), pp. 427–460.

[22] Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., and He, L., 2022, "A Survey of Human-in-the-Loop for Machine Learning," Future Gener. Comput. Syst., **135**, pp. 364–381.

[23] Viros-I-Martin, A., and Selva, D., 2020, "Daphne: A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions," IEEE J. Select. Top. Appl. Earth Observ. Remot. Sens., **13**, pp. 30–48.

[24] Bang, H., and Selva, D., 2016, "iFEED: Interactive Feature Extraction for Engineering Design," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, Aug. 21–24.

[25] Dong, H., Song, B., Wang, P., and Dong, Z., 2018, "Hybrid Surrogate-Based Optimization Using Space Reduction (HSOSR) for Expensive Black-Box Functions," Appl. Soft Comput. J., **64**, pp. 641–655.

[26] Dong, H., Song, B., Wang, P., and Dong, Z., 2018, "Surrogate-Based Optimization With Clustering-Based Space Exploration for Expensive Multimodal Problems," Struct. Multidiscipl. Optim., **57**(4), pp. 1553–1577.

[27] Snoek, J., Larochell, H., and Adams, R. P., 2012, "Practical Bayesian Optimization of Machine Learning Algorithms," Advances in Neural Information Processing Systems, Lake Tahoe, NV, Dec. 3 –6.

[28] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N., 2016, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," Proc. IEEE, **104**(1), pp. 148–175.

[29] Srivatsa, S., Kumar, R. S., Selva, D., and Silberstein, M. N., 2022, "Examining the Impact of Asymmetry in Lattice-Based Mechanical Metamaterials," Mech. Mater., **172**, p. 104386.

[30] Singh, V., and Gu, N., 2012, "Towards an Integrated Generative Design Framework," Design Stud., **33**(2), pp. 185–207.

[31] Fialho, A., 2010, "Adaptive Operator Selection for Optimization," Ph.D. thesis, Universite Paris Sud-Paris XI.

[32] Chabuk, T., Reggia, J., Lohn, J., and Linden, D., 2012, "Causally-Guided Evolutionary Optimization and Its Application to Antenna Array Design," Integr. Comput. Aided Eng., **19**(2), pp. 111–124.

[33] Jahr, R., Calborean, H., Vintan, L., and Ungerer, T., 2012, "Boosting Design Space Explorations With Existing Or Automatically Learned Knowledge," International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance, Bamberg, Germany, Mar. 17–19, Springer, pp. 221–235.

[34] Binder, W. R., and Paredis, C. J. J., 2017, "Optimization Under Uncertainty Versus Algebraic Heuristics: A Research Method for Comparing Computational Design Methods," ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, Aug. 6–9.

[35] Wu, S., Skolnick, J., and Zhang, Y., 2007, "Ab Initio Modeling of Small Proteins by Iterative TASSER Simulations," BMC Biol., **5**(17), pp. 1–10.

[36] Hitomi, N., and Selva, D., 2016, "A Hyperheuristic Approach to Leveraging Domain Knowledge in Multi-Objective Evolutionary Algorithms," ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, Aug. 21–24.

[37] Gaur, A., and Deb, K., 2016, "Adaptive Use of Innovization Principles for a Faster Convergence of Evolutionary Multi-objective Optimization Algorithms," Genetic and Evolutionary Computation Conference 2016, Denver, CO, July 20–24, Vol. 2909019, pp. 75–76.

[38] Woldesenbet, Y. G., Yen, G. G., and Tessema, B. G., 2009, "Constraint Handling in Multiobjective Evolutionary Optimization," IEEE Trans. Evolut. Comput., **13**(3), pp. 514–525.

[39] Bonissone, P. P., Subbu, R., Eklund, N., and Kiehl, T. R., 2006, "Evolutionary Algorithms + Domain Knowledge = Real-World Evolutionary Computation," IEEE Trans. Evolut. Comput., **10**(3), pp. 256–280.

[40] Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Qu, R., 2013, "Hyper-Heuristics: A Survey of the State of the Art," J. Oper. Res. Soc., **64**(12), pp. 1695–1724.

[41] Kumar, R. S., Srivatsa, S., Silberstein, M., and Selva, D., 2021, "Leveraging Design Heuristics for Multi-objective Metamaterial Design Optimization," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, Online, Aug. 17–19, pp. 1–15.

[42] Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V., 2007, "Evolutionary Algorithms for Solving Multi-Objective Problems."

[43] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evolut. Comput., **6**(2), pp. 182–197.

[44] Deb, K., Mohan, M., and Mishra, S., 2005, "Evaluating the ϵ-domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions," Evolut. Comput., **13**(4), pp. 501–525.

[45] Zhang, Q., and Li, H., 2007, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," IEEE Trans. Evolut. Comput., **11**(6), pp. 712–731.

[46] Thiele, L., Miettinen, K., Korhonen, P. J., and Molina, J., 2009, "A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization," Evolut. Comput., **17**(3), pp. 411–436.

[47] Coello, C., and Carlos, A., 1999, "A Survey of Constraint Handling Techniques Used With Evolutionary Algorithms," Lania-RI-99-04, Laboratorio Nacional de, pp. 1–33.

[48] Hoffmeister, F., and Sprave, J., 1996, "Problem-Independent Handling of Constraints by Use of Metric Penalty Functions," Fifth Annual Conference on Evolutionary Programming, San Diego, CA, Feb. 29–Mar. 2, pp. 289–294.

[49] Michalski, R. S., 2000, "Learnable Evolution Model: Evolutionary Processes Guided by Machine Learning," Mach. Learn., **38**(1), pp. 9–40.

[50] Runarsson, T. P., and Yao, X., 2000, "Stochastic Ranking for Constrained Evolutionary Optimization," IEEE Trans. Evolut. Comput., **4**(3), pp. 284–294.

[51] Mahbub, M. S., Wagner, M., and Crema, L., 2016, "Incorporating Domain Knowledge Into the Optimization of Energy Systems," Appl. Soft. Comput., **47**, pp. 483–493.

[52] Selva, D., 2014, "Experiments in Knowledge-Intensive System Architecting: Interactive Architecture Optimization," IEEE Aerospace Conference, Big Sky, MT, Mar. 1–8.
[53] Fialho, A., da Costa, L., Schoenauer, M., and Sebag, M., 2010, "Analyzing Bandit-Based Adaptive Operator Selection Mechanisms," Ann. Math. Artificial Intell., **60**(1), pp. 25–64.
[54] Hitomi, N., and Selva, D., 2017, "A Classification and Comparison of Credit Assignment Strategies in Multiobjective Adaptive Operator Selection," IEEE Trans. Evolut. Comput., **21**(2), pp. 294–314.
[55] Surry, P. D., and Radcliffe, N. J., 1996, "Inoculation to Initialise Evolutionary Search," Evolutionary Computing. AISB EC 1996. Lecture Notes in Computer Science, 1143, pp. 269–285.
[56] Zhou, L., Feng, L., Tan, K. C., Zhong, J., Zhu, Z., Liu, K., and Chen, C., 2021, "Toward Adaptive Knowledge Transfer in Multifactorial Evolutionary Computation," IEEE Trans. Cybernetics, **51**(5), pp. 2563–2576.
[57] Maini, H., Mehrotra, K., Mohan, C., and Ranka, S., 1994, "Knowledge-Based Nonuniform Crossover," IEEE Conference on Evolutionary Computation, Orlando, FL, June 27–29, Vol. 1, pp. 22–27.
[58] Holzapfel, G. A., Gasser, T. C., and Ogden, R. W., 2000, "A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models," J. Elast., **61**(1–3), pp. 1–48.
[59] Selva, D., Cameron, B. G., and Crawley, E. F., 2014, "Rule-Based System Architecting of Earth Observing Systems: Earth Science Decadal Survey," J. Space. Rockets, **51**(5), pp. 1505–1521.
[60] Selva, D., 2014, "Knowledge-Intensive Global Optimization of Earth Observing System Architectures: A Climate-Centric Case Study," SPIE Remote Sensing, Amsterdam, The Netherlands, Sept. 22–25, Vol. 9241, p. 92411S.
[61] Selva, D., and Crawley, E. F., 2013, "VASSAR: Value Assessment of System Architectures Using Rules," IEEE Aerospace Conference, Big Sky, MT, Mar. 2–9, pp. 1–21.
[62] Kramer, H. J., 2002, *Observation of the Earth and Its Environment: Survey of Missions and Sensors*, Springer Berlin, Heidelberg, Germany.
[63] Lee Rodgers, J., and Alan Nice Wander, W., 1988, "Thirteen Ways to Look at the Correlation Coefficient," Am. Statist., **42**(1), pp. 59–66.
[64] Zar, J. H., 2005, "Spearman Rank Correlation," Encyclopedia Biostatist., **7**.
[65] Brin, S., Motwani, R., and Silverstein, C., 1997, "Beyond Market Baskets: Generalizing Association Rules to Correlations," 1997 ACM SIGMOD International Conference on Management of Data – SIGMOD '97, Tucson, AZ, May 11–15.
[66] Derczynski, L., 2016, "Complementarity, F-score, and NLP Evaluation," 10th International Conference on Language Resources and Evaluation, LREC 2016, Portorož, Slovenia, May 23–28, pp. 261–266.
[67] While, L., Hingston, P., Barone, L., and Huband, S., 2006, "A Faster Algorithm for Calculating Hypervolume," IEEE Trans. Evolut. Comput., **10**(1), pp. 29–38.
[68] Coello Coello, C. A., 2000, "Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems," Comput. Ind., **41**(2), pp. 113–127.
[69] Goldberg, D. E., 1990, "Probability Matching, the Magnitude of Reinforcement, and Classifier System Bidding," Mach. Learn., **5**(4), pp. 407–425.
[70] Thierens, D., 2007, "Adaptive Strategies for Operator Allocation," Stud. Comput. Intell., **54**, pp. 77–90.
[71] Hitomi, N., Bang, H., and Selva, D., 2018, "Adaptive Knowledge-Driven Optimization for Architecting a Distributed Satellite System," J. Aerospace Inf. Syst., **15**(8), pp. 485–500.