Res Eng Design (2013) 24:395–409
DOI 10.1007/s00163-013-0158-0

ORIGINAL PAPER

Usage scenarios for design space exploration with a dynamic multiobjective optimization formulation

S. K. Curtis · B. J. Hancock · C. A. Mattson

Received: 9 August 2012 / Revised: 1 July 2013 / Accepted: 3 July 2013 / Published online: 24 July 2013

Abstract In a recent publication, we presented a new strategy for engineering design and optimization, which we termed formulation space exploration. The formulation space for an optimization problem is the union of all variable and design objective spaces identified by the designer as being valid and pragmatic problem formulations. By extending a computational search into this new space, the solution to any optimization problem is no longer predefined by the optimization problem formulation. This method allows a designer to both diverge the design space during conceptual design and converge onto a solution as more information about the design objectives and constraints becomes available. Additionally, we introduced a new way to formulate multiobjective optimization problems, allowing the designer to change and update design objectives, constraints, and variables in a simple, fluid manner that promotes exploration. In this paper, we investigate three usage scenarios where formulation space exploration can be utilized in the early stages of design when it is possible to make the greatest contributions to development projects. Specifically, we look at formulation space boundary exploration, Pareto frontier generation for multiple concepts in the formulation space, and a new way to perform targeted boundary expansion. The benefits of these methods are illustrated with the conceptual design of an impact driver.

Keywords Multiobjective optimization · Pareto frontier · Design space exploration · Dynamic formulation

S. K. Curtis · B. J. Hancock · C. A. Mattson (✉)
Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA
e-mail: mattson@byu.edu

List of symbols

$\mathbf{g}$	Vector of inequality constraints
$\mathbf{h}$	Vector of equality constraints
$\mathbf{p}$	Vector of fixed design parameters
$\mathbf{v}$	Vector of interest in the objective space
$\mathbf{w}$	Diagonal matrix of objective weights
$\mathbf{x}$	Vector of design variables or design objects
$\mathbf{y}$	Vector of independent design objects
$\mathbf{z}$	Vector of dependent design objects
$\boldsymbol{\mu}$	Vector of design objectives
γ	Vector in the objective space

Subscripts and superscripts

$[]_i$	Dummy index
$[]_j$	Dummy index
$[]_q$	Dummy index
$[]_l$	Lower bound
$[]_u$	Upper bound
$[]^{(-)}$	Lower bound in targeted boundary expansion
$[]^{(+)}$	Upper bound in targeted boundary expansion
$[]^{(0)}$	Benchmark
$[]^{(k)}$	Concept or formulation
$\underline{[\,]}$	Formulation space

1 Introduction

Success in engineering design is closely tied to a designer's ability to make rational, informed decisions throughout the product development process. Decisions that typically have the largest impact on a design's outcome occur during early conceptual design, when the least is known about design objectives or constraints (Homan and Thornton 1998; Ishii 1995; Mattson and Messac 2002; Wang 2001). While many ad hoc, heuristic methods (Mulet and Vidal 2008; Olewnik and Lewis 2003; Pahl et al. 2007; Pugh 1996; Ulrich and Eppinger 2004) exist to support conceptual design decision-making, computational search methods are rarely utilized until the later, detailed design stages. Thus, to a large extent, the valuable information that is provided through computational search (i.e., algorithmic optimization) is only available when its impact is the least. Several researchers have pushed to capitalize on the benefits of using computational methods and optimization techniques earlier in the design process (Antonsson and Cagan 2001; Barnum and Mattson 2010; Cagan et al. 2005; Chakrabarti 2002; Dye et al. 2007; Hassan and Crossley 2002; Kurtoglu and Campbell 2009; Lewis et al. 2011; Mattson et al. 2009; Morino et al. 2006; Qazi and Linshu 2005; Shelley et al. 2007). Despite these advances, there are still challenges that currently limit the extent to which designers can use computational search methods to assist in early-stage design decision-making.

One challenge is that much of early-stage, conceptual design is qualitative in nature. Designers use sketches and rough prototypes to explore concept ideas; very little quantitative modeling takes place. To help bridge this gap, some have used sketch recognition software to transform hand-drawn sketches into parametric, computational models (Alvarado and Davis 2007; Davis 2007; Landay and Myers 2001; LaViola 2011; Masry et al. 2005; Zeleznik et al. 2008). With regards to optimization, methods such as interactive genetic algorithms (Brintrup et al. 2007; Brintrup et al. 2008; Gong and Yuan 2011; Takagi 2001) or fuzzy logic systems (Huber et al. 2008; Oduguwa et al. 2007) can help to resolve design conflicts involving qualitative design objectives. We note that when analytical models do exist, they do not necessarily need to be high fidelity to be useful during conceptual design; in fact, computationally inexpensive models are advantageous because they allow the designer to quickly explore a large design space (Kuehmann and Olson 2009). Metamodeling techniques have been widely used to obtain adequate analytical models for use during conceptual design exploration and optimization (Wang and Shan 2007). Therefore, for the purposes of this paper, we will assume that designers have access to preliminary analytical models for use in a computational search during conceptual design.

Another challenge that hinders the use of computational search during conceptual design is the designer's lack of knowledge about the design problem itself. The designer may still be learning about the true needs and limitations of the design project—two critical elements of numerical optimization (Wang and Shan 2007). Traditionally, a designer must define objectives, constraints, and limits before executing an optimization algorithm. However, the results of the search will be less useful to the designer if the problem is not formulated properly to reflect his or her true preferences, which is often the case (Balling 1999; Stump et al. 2009). In recent years, a number of dynamic multiobjective optimization techniques have been developed for handling models that change over time (Guan et al. 2005; Tantar et al. 2011; Farina et al. 2004). These optimization algorithms are equipped to treat models as time-dependent scenarios, rather than static snapshots. However, lacking attention in the literature is a method for creating a dynamic multiobjective formulationan optimization framework that easily changes over time to reflect changes not due to fluctuating operating conditions, but rather due to a designer's evolving needs and preferences as new design needs and objectives arise throughout the design process. In other words, a dynamic multiobjective optimization formulation is needed—one that allows for easily modified problem formulations and does not confine the search to the space defined by the initial parameterization (Agte et al. 2010).

In a previous paper (Curtis et al. 2013), the authors presented a dynamic optimization problem formulation, which allows the designer to explore a new space termed the *formulation space*. The formulation space is the union of all variable and design objective spaces identified by the designer as being valid and pragmatic problem formulations. By extending the search into this new space, the solution to an optimization problem is no longer predefined by the problem formulation. For many practical problems, this predefinition is not a drawback, since numerical optimization is employed to simply carry out the routine computations so that the designer does not have to. For other design problems, not of this nature, the designer is genuinely interested in exploring the design options without having to have formed a concrete understanding of the problem or definition of the formulation. In such cases, which are abundant in early design, formulation space exploration enables the designer to search computationally in both a divergent and convergent manner.

In this paper, we explore how a dynamic optimization problem formulation, and more specifically formulation space exploration, can be used to obtain valuable information during conceptual design. We provide three scenarios for its use: (1) formulation space boundary generation and exploration, (2) Pareto frontier generation for multiple concepts in design concept selection, and (3) targeted boundary expansion. Our goal in presenting these usage scenarios is not to highlight the novelty of these particular scenarios, but rather to demonstrate how exploring the formulation space using a dynamic multiobjective optimization formulation can provide the designer with valuable information in a variety of activities in early-stage design.

The remainder of this paper is organized as follows: We begin in Sect. 2 with technical preliminaries where we present and discuss the standard multiobjective optimization problem formulation, followed by a dynamic multiobjective optimization problem formulation. Then, in Sect. 3, we discuss three uses for the dynamic optimization problem formulation in conceptual design. In Sect. 4, we provide evidence of the benefits of formulation space exploration with a case study involving the conceptual design of an impact driver. Finally, in Sect. 5, we offer concluding remarks.

2 Technical preliminaries

In this section, we present the standard multiobjective optimization formulation. Also, we briefly summarize the developments of Curtis et al. (2013) by presenting a dynamic multiobjective optimization formulation.

2.1 Standard multiobjective optimization formulation

The generic, deterministic multiobjective optimization problem is formulated as Problem 1 ($P1$):

$$\min_{\mathbf{x}} \{\mu_1(\mathbf{x},\mathbf{p}), \mu_2(\mathbf{x},\mathbf{p}), \ldots, \mu_n(\mathbf{x},\mathbf{p})\} \quad (n \geq 2) \tag{1}$$

subject to inequality constraints $g_q(\mathbf{x},\mathbf{p}) \leq 0 \{q = 1, 2, \ldots, n_g\}$, equality constraints $h_j(\mathbf{x},\mathbf{p}) = 0 \{j = 1, 2, \ldots, n_h\}$, and side constraints $x_{l,i} \leq x_i \leq x_{u,i} \{i = 1, \ldots, n_x\}$. The vector $\mathbf{x}$ represents a set of design variables, and $\mathbf{p}$ contains a set of fixed design parameters. In other words, we minimize a set of objective functions, $\boldsymbol{\mu}$, by finding the optimal values for the design variables in $\mathbf{x}$ that satisfy all design constraints. The objectives, equality constraints, and inequality constraints may be linear or nonlinear functions of $\mathbf{x}$ and $\mathbf{p}$.

When design objectives are competing, $P1$ produces a set of optimal solutions called the Pareto frontier. This is shown graphically in Fig. 1, where the feasible design objective space for two minimized objectives (μ_1 and μ_2) is plotted. In the figure, any point residing on or in the shaded region represents a feasible design solution, meaning that the inequality, equality, and side constraints for the design are satisfied. Each solution comprising the frontier (shown as the bolded curve in Fig. 1) is said to be *Pareto optimal*. Pareto optimality indicates that there are no other designs for which *all* objectives are improved. In other words, the Pareto frontier is the set of all nondominated solutions for a particular problem. In multiobjective optimization, designers generally seek Pareto solutions because they indicate that the objectives cannot be improved any more without reducing the performance of other objectives in exchange (Miettinen 1999).

2.2 Dynamic multiobjective optimization formulation

In Sect. 1, we briefly mentioned the idea of formulation space exploration—this is depicted graphically in Fig. 2. In

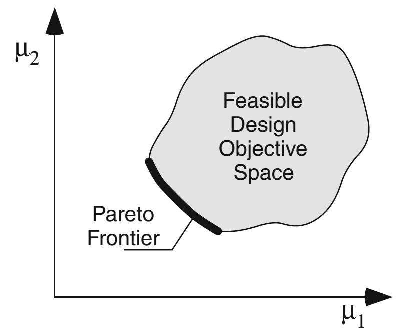

Fig. 1 Feasible design objective space is shown *shaded* and the Pareto frontier is shown as the *bolded curve*

Fig. 2a, we plot the design objective space for two objectives (μ_1 and μ_2), similar to what was done in Fig. 1. In Fig. 2(b), we plot a different design objective space shown as the shaded region, which came from adjusting the optimization problem formulation (e.g., the constraints and/or limits changed). The design space from the previous formulation is represented by the region enclosed by the dashed lines. In Fig. 2c, we plot the design space for one more problem formulation, which resulted in the smaller shaded design space. Again, the previous design spaces are represented with the dashed lines. The aggregate of these spaces, shown in Fig. 2d, is the formulation space. It is clear from the plots that formulation space exploration is divergent in nature, allowing the designer to form the solution as the search progresses. Notice that we have underlined the objectives in this final plot to signify that this is the formulation space.

For the purposes of illustration, we have limited the graphs in Fig. 2 to two-dimensional space; however, formulation space exploration is free to expand into n-dimensional space. In fact, the formulation space will continuously evolve as the designer modifies objectives and variables or as design models are updated. This evolving formulation space can be viewed as belonging to a single problem; however, because the final step of formulation space exploration is generally the selection of a single formulation (and corresponding design space) to pursue further, each unique formulation is labeled to differentiate it from other design spaces for when the time comes to make that selection. By analyzing the gains and tradeoffs of each newly discovered space, the designer is able to use previous formulations to guide his or her search further for the formulation that will best meet his or her design needs.

Unlike many other design optimization methods, the purpose of formulation space exploration is to lead the designer to a desired formulation, rather than a desired solution. This is because, as has been previously mentioned, the designer does not always know at this point in

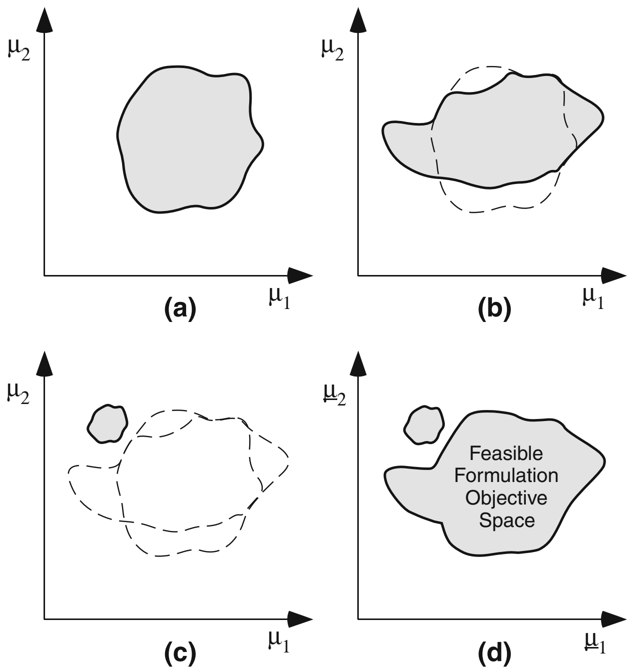

Fig. 2 **a** A traditional design objective space. **b** A different design objective space, obtained by reformulating the optimization, shown as the *shaded region* and overlaid on the design space from the previous plot. **c** A third design space obtained by manipulating the optimization problem formulation. **d** The formulation objective space

the design process all objectives, constraints, and variables for the problem in question. Furthermore, under our stated assumption that relatively inexpensive models are being used, it is understood that more detailed models will be used later on to reach a final solution. Importantly, the human designer is integral to this exploration process and acts as the rational decision-maker while using the computer to execute mundane calculations. In other words, formulation space exploration requires the designer to remain *in the loop*, and the designer benefits from the added computational assistance.

It is beneficial to briefly compare and contrast this approach for design space exploration to other methods. There are many existing approaches for systematically exploring a defined design space. An excellent example of this is the approach called *innovization*, which finds a set of optimal solutions for a problem and examines their commonalities to expose useful and innovative underlying design principles (Deb and Gupta 2006). However, the nature of the formulation space is such that each problem contains a potentially infinite number of possible design spaces. Because any number of design objects can be added, deleted, or changed during the exploration process, there are no defined boundaries within which an algorithm can search to systematically extract information. Instead, formulation space exploration depends on the designer's intuition, keeping him or her in the design loop, as has been mentioned. Instead of searching the infinite space by computer, the user undergoes an intuitive exploration process with computational assistance to improve the efficiency of the search.

Another related method is the multicriteria multiscenario optimization approach (Wiecek et al. 2009). In this approach, the Pareto frontiers of multiple formulations for multiobjective optimization problems are identified and compared with common solutions. A less involved approach is often performed in structural analysis for multiple load cases, wherein an aggregate objective function is formed to arrive at a solution that has been identified to perform satisfactorily for all cases. The primary difference between the method proposed in this paper and these other methods is that they have a number of predefined scenarios or formulations for which they are seeking a single satisfactory solution. Formulation space exploration is meant for a different setting—one in which the user is exploring various design spaces (typically not belonging to a predefined set) in order to select a single formulation that can then be used in a detailed design setting to optimize for the solution that best satisfies that single scenario.

Formulation space exploration is a dynamic process: design variables, parameters, constraints, and objectives change as a designer formulates and reformulates an optimization problem. For example, a design variable in one formulation may be implemented as fixed design parameter in the next, and as an objective in subsequent formulations. Therefore, to preserve clarity and to emphasize the fluid nature of formulation space exploration, we will refer to all optimization components—design variables, parameters, constraints, and objectives—as *design objects*. The behavior of each design object is dictated by how it is implemented in the generic dynamic multiobjective optimization problem, given as Problem 2 (*P*2):

$$\min_{\mathbf{y}}\left\{\mu_1(\mathbf{x}), \mu_2(\mathbf{x}), \ldots, \mu_{n_x}(\mathbf{x})\right\} \quad (n_x \geq 2) \tag{2}$$

subject to the side constraints

$$y_{l,i} \leq y_i \leq y_{u,i} \quad \{i = 1, \ldots, n_y\} \tag{3}$$

$$z_{l,i} \leq z_i \leq z_{u,i} \quad \{i = 1, \ldots, n_z\} \tag{4}$$

where

$$\boldsymbol{\mu} = \mathbf{w} * \mathbf{x} \tag{5}$$

$$\mathbf{w} = \begin{bmatrix} w_{1,1} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & w_{n_x,n_x} \end{bmatrix} \tag{6}$$

$$\mathbf{x} = \left[y_1, y_2, \ldots, y_{n_y}, z(y)_1, z(y)_2, \ldots, z(y)_{n_z}\right]^T \tag{7}$$

where $\mathbf{y}$ is a vector of independent design objects, $\mathbf{z}$ is a vector of dependent design objects, $\mathbf{x}$ is a concatenated vector of all the design objects in $\mathbf{y}$ and $\mathbf{z}$, and $\mathbf{w}$ is a diagonal matrix where each element along the diagonal is a member of the set $\{-1, 0, 1\}$. The number of total design objects, independent design objects, and dependent design objects is denoted n_x, n_y, n_z, respectively.

Again, the behavior of each design object in $\mathbf{x}$ is determined by the implementation of $P2$. If in Eq. (3), $y_{\{1}, i = y_{\{u,i}$, then y_i (which is also x_i) is a fixed design parameter. Otherwise, if $y_{l,i} \neq y_{u,i}$, then y_i is a design variable. Likewise, if in Eq. (4), $z_{l,i} = z_{u,i}$, then z_i (which is also x_{i+n_y}) is an equality constraint. And when $z_{l,i} \neq z_{u,i}$, z_i is an inequality constraint. If the lower or upper bound on z_i is to be ignored, then the value of $z_{l,i}$ or $z_{u,i}$ is set to $-\infty$ or ∞. Objectives are dictated by $\mathbf{w}$ when $w_{i,i} = 0$, x_i is not a design objective. When $w_{i,i} = 1$, x_i is an objective to be minimized; when $w_{i,i} = -1$, x_i is to be maximized. In this manner, design objects can easily transform and mutate as the designer explores the formulation space.

Admittedly, formulation space exploration is possible with $P1$ or $P2$; in fact, both will yield the same Pareto frontier. However, it has been shown that $P2$ requires fewer lines of code to be manipulated after it has been formulated, while requiring roughly the same number of lines of code to initially formulate (Curtis et al. 2013). For example, objectives are turned on and off by simply changing the scalar values in the diagonal of $\mathbf{w}$—no additional programming is necessary. The same is true when changing a design parameter into a design variable—only the limit values in $\mathbf{y}$ and $\mathbf{y}_l$ need to be changed. This reduction in effort to reformulate optimization problems is important for effective formulation space exploration. Designers must be willing to ask "what if" questions and explore tradeoffs, and this is less likely to occur if the cost of reformulation is perceived to be high. Similar trends have been reported in CAD modeling—designers are less willing to modify CAD models if significant effort is required (Robertson and Radcliffe, 2009). The more natural an engineering design tool or process is, the more likely it will be generally accepted (Lopez-Mesa and Bylund 2011). For the full development of $P2$, including computational limitations, we refer the reader to Curtis et al. (2013).

3 Usage scenarios for formulation space exploration in conceptual design

In this section, we present three scenarios for performing formulation space exploration using the dynamic optimization problem formulation presented above. All three scenarios are encountered during conceptual design after at least one design concept has been developed. To avoid confusion, we will adopt the definition of a design concept from Mattson and Messac (2003), where a concept is defined as *an idea that has evolved to the point that there is a parametric model that represents the performance of the family of design alternatives that belong to that concept's definition*. The applicability of each scenario is governed by the amount of information a designer truly knows at that point in the design process. In Sect. 3.1, we look at a situation where the designer knows little about the design objectives and is more interested in divergently exploring the formulation space as a whole rather than any particular Pareto frontier. In Sect. 3.2, we discuss a scenario where the designer has solidified the objectives of the project and is ready to converge on a particular concept. And in Sect. 3.3, we investigate how to divergently explore regions of infeasibility, with the intent of learning more about the design, its tradeoffs, and potential future design possibilities. Because the design process ideally consists of a number of iterations of convergence and divergence, there is no single right way for applying these scenarios or other design activities during exploration. It should be noted that solving a multiobjective optimization problem repeatedly can easily become a time-consuming process. Thus, we reiterate that formulation space exploration is most feasible and useful in applications where the designer has access to relatively computationally inexpensive analytical models with which to work. Such models are often available in early-stage design activities. As a drawback, these models are also generally the least accurate.

3.1 Scenario 1: Formulation space boundary exploration

One scenario where the dynamic formulation allows the engineer to explore and learn more about a product's design space is through formulation space *boundary* exploration. If design objectives and preferences are truly unknown, which is often the case in early design, then finding an s-Pareto frontier for a set of concepts is less meaningful than finding the boundaries of the formulation space, which represent the extreme values of the formulation space with respect to any combination of objectives being maximized or minimized. In other words, understanding the full objective space can be useful in some design scenarios. For example, when designing an automobile, there are many potential design objectives: power, fuel efficiency, size, maximum speed, acceleration, etc. Depending on the design purpose, different regions of the objective space may be more desirable than others. Suppose we have a new composite material that can be implemented in a vehicle. In the highly coupled system of an automobile, the design implications of this composite material may be unknown. Thus, we may desire to explore the formulation space using a basic automobile model to see if the technology is better suited for the high performance vehicle market, where power is maximized and size is less desirable, or in the minivan market, where fuel efficiency and size are to be maximized. The overall shape and size of the formulation space boundaries may also suggest the confidence that the designer can have in the ability of a certain design to achieve particular specifications in practice. However, if investigation of the robustness of a model is the designers primary objective, the authors suggest a more thoroughly examined approach, such as those presented in (Barrico and Antunes 2006; Deb and Gupta 2006).

Formulation space boundary exploration is fully possible with either $P1$ or $P2$; however, we will only explicitly present this procedure with $P2$. Recall that in $P2$, objectives are controlled by the values along the diagonal in **w**. For a two-dimensional problem of a single optimization formulation, such as the one seen in Fig. 3, it is possible to obtain the boundary of the design objective space using four different **w** matrices in $P2$ and the normal boundary intersection method or a modified normal constraint method (Das and Dennis 1998; Messac et al. 2003). For example, in $\mathbf{w}^{(1)}$ in the figure, both objectives are minimized (i.e., the elements of $\mathbf{w}_{i,i}$ corresponding to μ_i for both objectives are equal to 1); this results in an optimization problem that produces the lower left boundary of the design space. The three remaining boundaries can be obtained by toggling the requisite values in **w** between 1 and -1, as shown in Eq. (8). A similar process can be used to find the boundaries of the formulation space. As shown by this example, exploring the various combinations of minimized and maximized objectives for a particular problem is easily performed using a dynamic multiobjective optimization formulation. This is because between each formulation, only the values in the w-matrix require changing, according to the full factorial matrix f. To perform these same calculations using the standard multiobjective optimization formulation would require greater effort to reformulate and therefore decrease the likelihood of such exploration taking place.

The general process for finding the boundaries of the formulation space is shown in Fig. 4. First, the designer chooses the design objects of interest from the vector **x**, the total number of which is n_d, and stores the indices that correspond to **x** in a vector, **d**. Second, the designer generates a 2^{n_d} two-level, full factorial matrix **f** in standard form. For example, if $n_d = 2$, the following matrix would be displayed.

$$\mathbf{f} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} \tag{8}$$

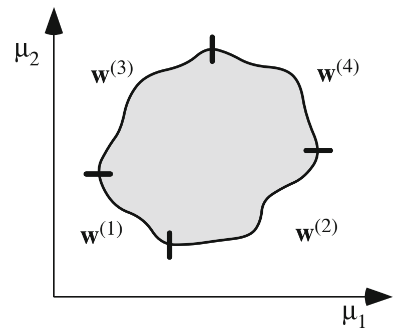

Fig. 3 The results of four different optimization formulations are overlaid on the design objective space

Next, $P2$ is executed 2^{n_d} times in a loop. For every iteration in the loop, $\mathbf{w}^{(i)}$ is generated by setting $w_{l,l}$ to the ith row and the mth column of **f**, where l is the mth entry in **d**. For every loop, the solution is added to the set S and i is incremented. Once i is greater than 2^{n_d}, the process is repeated for any remaining concepts or formulations, the total number of which is n_k. The result is a set of designs outlining the extreme boundaries of the formulation space for the given concepts and/or formulations. A limitation of this scenario is that it may not identify the complete formulation space boundary when the shape of the space has certain unique features. These features include concavities that are nonmonotonic with respect to at least one axis, discontinuities inside the space, or disjointed spaces, as shown in Fig. 5. However, because the predominant purpose of boundary exploration is to simply gain an overall basic understanding of achievable values in various objectives, this limitation does not significantly decrease the utility of this scenario for most cases.

3.2 Scenario 2: s-Pareto generation for multiple formulations

Perhaps the most obvious use for an optimization problem is to converge to an optimal solution. In this section, we combine the dynamic multiobjective optimization approach with an s-Pareto generation and selection strategy presented by Mattson and Messac (2003). Consider the two-formulation spaces shown in Fig. 6. An s-Pareto frontier is defined as the Pareto optimal solutions for a set of concepts. In this case, however, we have shown an s-Pareto frontier in a formulation space, because it contains the Pareto optimal solutions for the set of all formulations and concepts. This is evident in the figure because the bolded line, representing the s-Pareto frontier, spans multiple concepts and formulations. With the s-Pareto frontier defined, concept selection can proceed with a qualitative or

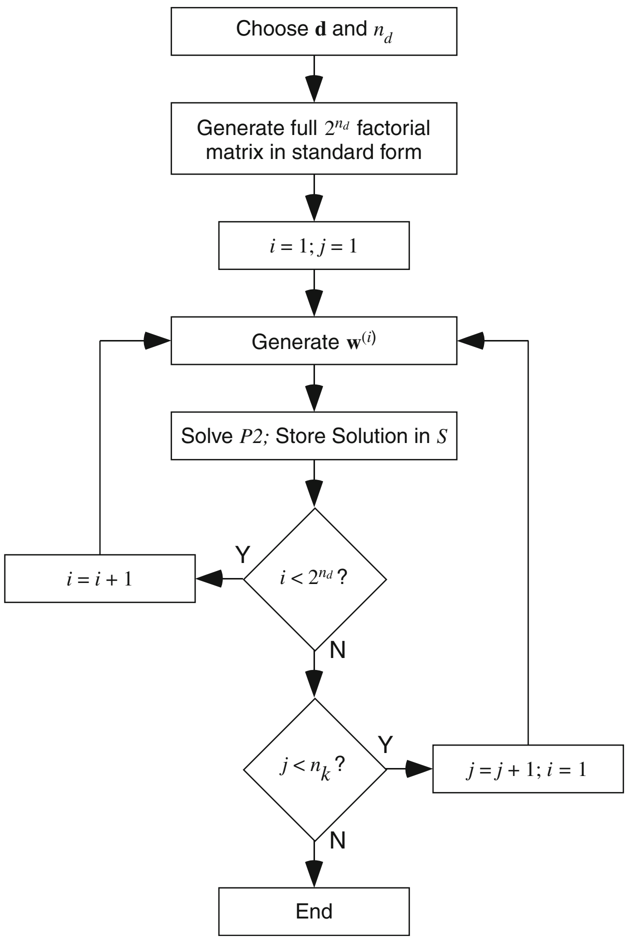

Fig. 4 Flow chart for determining the boundaries of a formulation space using the dynamic optimization formulation

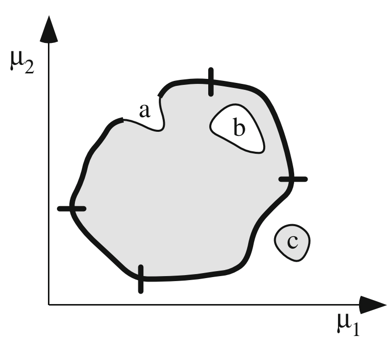

Fig. 5 A formulation space exhibiting features that may not be recognized by formulation space boundary exploration: *a* Nonmonotic concavity, *b* discontinuity inside the space, *c* disjointed space

quantitative analysis of concept goodness as described by Mattson and Messac (2005).

The optimization problem given by $P2$ (Eqs. 2–7) needs to be modified to account for multiple concepts. The generic, dynamic multiobjective optimization problem capable of comparing multiple concepts is given by Problem 3 ($P3$):

$$\min_{k}\left\{\min_{\mathbf{y}^{(k)}}\left\{\mu_1^{(k)}(\mathbf{x}^{(k)}),\mu_2^{(k)}(\mathbf{x}^{(k)}),\ldots,\mu_{n_x^{(k)}}(\mathbf{x}^{(k)})\right\}\right\} \quad (n_x^{(k)}\geq 2) \tag{9}$$

where the superscript $[]^{(k)}$ indicates that [] is associated with formulation or concept k. Equations (3–7) from $P2$ are still valid here for $P3$, although each equation will be specific to the formulation or concept k. Solving $P3$ results in an s-Pareto frontier—one that potentially spans multiple formulations and concepts. Note that while it is generally easier to explore the formulation space of each concept using the dynamic optimization problem rather than the standard optimization problem (see Sect. 2.2), it is not necessarily easier to compare different concepts, since the same initial coding effort is required for each concept model.

The objectives minimized in $P3$ are *set objectives*, meaning that they are comparable across all formulations and concepts. The inputs to the concept models that generate set objectives may be unique. For example, the required variable inputs needed to calculate the mass of a bevel gear and a spur gear may be different; yet, the mass of the two types of gears is comparable. Additionally, it is possible for a formulation or concept to have one or more objectives that are specific to the formulation or concept. To illustrate, consider a design concept that contains a hazardous material. It may be necessary to maximize the safety of this concept, whereas other design concepts generated may not contain the hazardous material, obviating the need to maximize safety. These formulation and concept specific objectives are easily included as constraints in **x**. More information on how to handle formulation or concept specific objectives can be found in Mattson and Messac (2003).

3.3 Scenario 3: Targeted Boundary Expansion

A third scenario for using the dynamic optimization problem formulation is to perform design feasibility

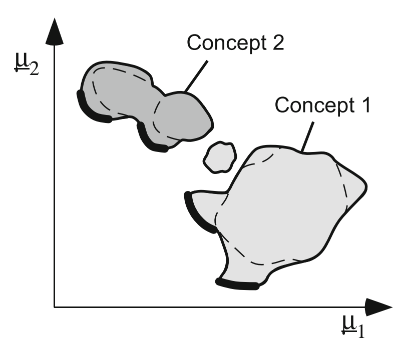

Fig. 6 Design objective formulation spaces for two concepts are shown. The resulting s-Pareto frontier is outlined in *bold* and spans both concepts and multiple formulations

studies, or targeted boundary expansion. Consider the feasible design space shown as the dark-shaded region in Fig. 7. If the designer wants to see designs near the black circle, he or she could easily run an optimization (using *P*1 or *P*2) that minimizes the Euclidean distance to that point in the design objective space (Stump et al. 2009). However, if the designer would like to see designs near the black star in Fig. 7, which is located outside of the feasible design space, then the problem constraints need to change ($\mathbf{y}_l$, $\mathbf{y}_u$, $\mathbf{z}_l$, and $\mathbf{z}_u$ need to be modified) to extend the searchable space toward the star. This scenario could occur if a designer highly desires a particular performance in the product, and is willing and able to compromise some of the constraints of the design. For example, consider a project where a design team has been given a certain budget, which they understand to be a constraint. The team wants to know, however, how much more it would cost to get to a particular performance level that they currently cannot reach, given the monetary constraints of the budget. If the increase in cost is fairly small for a significant increase in performance, this may justify a request for a change in budget, or a change in the constraints that they have been given. In this manner, the *optimization formulation itself becomes a part of the optimization*.

We present an optimization formulation with two objective functions to explore infeasible regions of interest as Problem 4 (*P*4):

$$\min_{\mathbf{x}_l,\mathbf{x}_u}\left\{f_1(\mathbf{x}_l,\mathbf{x}_u,\mathbf{v}),\quad f_2(\mathbf{x}_l,\mathbf{x}_u,\mathbf{x}_l^{(0)},\mathbf{x}_u^{(0)})\right\} \tag{10}$$

subject to the side constraints

$$x_{l,i}^{(-)} \le x_{l,i} \le x_{l,i}^{(+)} \quad \{i = 1,\ldots,n_x\} \tag{11}$$

$$x_{u,i}^{(-)} \le x_{u,i} \le x_{u,i}^{(+)} \quad \{i = 1,\ldots,n_x\} \tag{12}$$

and

$$f_1(\mathbf{x}_l,\mathbf{x}_u,\mathbf{v}) = \min_{\mathbf{y}} \|\gamma(\mathbf{y}) - \mathbf{v}\| \tag{13}$$

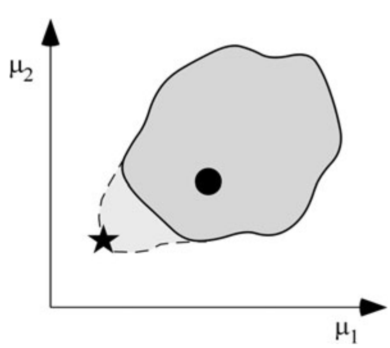

Fig. 7 The original design objective space is *darkly shaded*, with a feasible point shown as a *circle* on the interior of this space. A *star* represents a design point of interest that is not feasible

$$f_2(\mathbf{x}_l,\mathbf{x}_u,\mathbf{x}_l^{(0)},\mathbf{x}_u^{(0)}) = \|(\mathbf{x}_l - \mathbf{x}_l^{(0)})\| + \|(\mathbf{x}_u - \mathbf{x}_u^{(0)})\| \tag{14}$$

where the superscripts $[]^{(-)}$ and $[]^{(+)}$ indicate a lower or upper bound on [], respectively, and the superscript $[]^{(0)}$ indicates that [] is from the original formulation (i.e., the formulation that defines the feasible objective space—the dark-shaded region in Fig. 7). The vector $\mathbf{x}_l$ is the concatenation of $\mathbf{y}_l$ and $\mathbf{z}_l$, or $\mathbf{x}_l = [\mathbf{y}_l; \mathbf{z}_l]$; likewise, $\mathbf{x}_u = [\mathbf{y}_u; \mathbf{z}_u]$. The vector $\mathbf{v}$ represents a point of interest in the current infeasible objective space, or the star in Fig. 7, and $\gamma(\mathbf{y})$ is the set of objectives in $\mathbf{x}$ that correspond to $\mathbf{v}$. According to Eq. (10), the designer attempts to minimize the Euclidean distance between $\gamma(\mathbf{y})$ and $\mathbf{v}$ while also minimizing the changes made to the original optimization formulation. This is similar but not identical to goal programming, which can also be used in multiobjective optimization situations. Goal programming seeks to discover a particular combination of variables (satisfying predefined constraints) that will yield a solution that minimizes the Euclidean distance between itself and a desired point (Charnes and Cooper 1977). Targeted boundary expansion, on the other hand, adjusts both variables and constraints to arrive at a formulation that minimizes the distance between its nearest boundary and the desired point. As with most optimization problems, proper scaling of the design objects will produce better results; this is especially critical when calculating f_2 with Eq. (14), as a relatively small change in one constraint could be large in comparison with another. Suitable methods for scaling have been presented in the following publications (Gill et al. 1981; Kasprzak and Lewis 2001; Nha et al. 1998; Parkinson et al. 1992).

Successfully solving *P*4 will result in a Pareto frontier of solutions, each of which represents an optimization formulation. This frontier represents the whole tradeoff surface between minimizing changes to the problem formulation and reaching the desired point in the objective space. In other words, this is a targeted boundary expansion process, where the designer picks a point of interest, and the optimization formulation that can find that point is returned. Ultimately, the designer will discover (and be able to select from the identified solutions) the minimum cost to obtain a desired performance.

4 Case study: impact driver design

The purpose of this case study is to illustrate how to use the dynamic multiobjective optimization formulation in the three usage scenarios presented in Sect. 3 to produce valuable design information for decision-makers during conceptual design. Although this case study is anecdotal in nature, it illustrates several important points: (1) By searching the formulation space, designers are able to

Table 1 Model inputs and outputs for the five impact driver concepts

	Type	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
Model input						
Drive shaft materials	Discrete	X	X	X	X	X
Drive shaft sizes	Continuous	X	X	X	X	X
Motor type	Discrete	X	X	X	X	X
Motor location	Continuous	X	X	X	X	X
Impact assembly location	Continuous	X	X	X	X	X
Gear material	Discrete	X	X		X	X
Gear type	Discrete	X	X		X	X
Counterweight location	Continuous		X	X		
Counterweight material	Discrete		X	X		
Counterweight size	Continuous		X	X		
Model output						
Drive shaft locations	Continuous	X	X	X	X	X
Shaft stress constraint	Continuous	X	X	X	X	X
Total mass	Continuous	X	X	X	X	X
Center of mass	Continuous	X	X	X	X	X
Total cost	Continuous	X	X	X	X	X
Number of collissions	Continuous	X	X	X	X	X
Torque supplied	Continuous	X	X	X	X	X
Torque difference	Continuous	X	X	X	X	X
Speed supplied	Continuous	X	X	X	X	X
Speed difference	Continuous	X	X	X	X	X
Gear locations	Continuous	X	X		X	X
Gear torque constraint	Continuous	X	X		X	X

search the design space in both a divergent and convergent manner. (2) Formulation space exploration requires the human designer to be intimately involved in the search process, allowing his or her judgement and rational decision-making capabilities to guide the search. (3) Using the dynamic optimization problem formulation promotes design exploration. The focus here is not to defend the practicality of the design resulting from the application of the methods presented in this paper, but rather to show how these methods could be used in the development of a new product.

4.1 Problem description

The case study is based upon a proposed new type of impact driver, which is a specialized tool that applies high torque to fasteners by the means of a hammer mechanism. The novel aspect of this new type of impact driver is depicted in Fig. 8. On the left side of the figure, a backpack holding several batteries is shown. The batteries connect to a power cord which runs from the backpack, down the user's arm and into a special glove with electrical contacts embedded in the palm of the glove (shown on the right in the figure). There are corresponding electrical contacts on the impact driver. Thus, a complete, electrical circuit is made when the user grabs the impact driver with the glove on.

The goals of the design are to (1) reduce arm fatigue for those who use the impact driver for long periods of time, such as outdoor deck fabricators, sheetrock hangers, or general construction workers; (2) increase the battery life between charges (more batteries can fit in a backpack than directly on a typical impact driver); and (3) maintain the mobility of a cordless impact driver. For the remainder of the case study, we will direct our attention to how a designer might develop an impact driver to accompany the backpack and glove—specifically, how to design a DC motor impact driver with *no* battery attachment. Two different groups of engineering graduate students at Brigham Young University (BYU) designed and built functional prototypes of this novel type of impact driver. Both prototypes are shown in Fig. 9. We will compare the results of our exploration process with these designs.

We have generated five impact driver concepts that could potentially fulfill the design specifications of this case study, shown as Concepts 1 through 5 in Fig. 10. In each case, we alter the geometry and product architecture, and add or subtract drive train components to achieve the desired goals. We assume that each concept will use the same impact assembly, which has already been designed

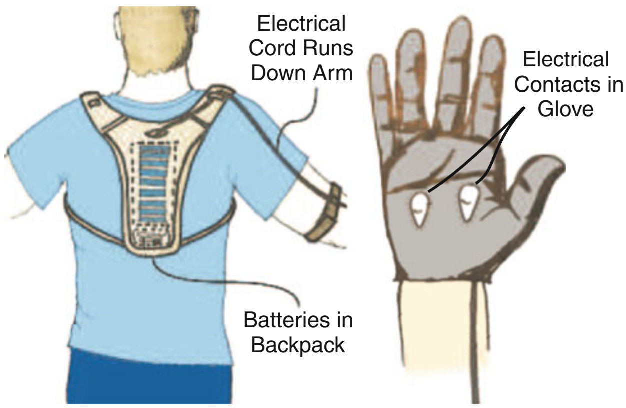

Fig. 8 General idea for new type of impact driver. A backpack holds several batteries, which connect to a special glove via a power cord. The glove has electrical contacts that correspond and connect power to an impact driver

and is the existing impact assembly for a 12V Hitachi (model WH10DFL) impact driver. We now describe each concept in greater detail:

- *Concept 1*—In this concept, we orient the impact assembly (I) horizontally above the trigger assembly (T). Two sets of bevel gears (G) connect the impact driver to the motor (M), which is oriented horizontally and located where batteries are typically found on most commercially available impact drivers.
- *Concept 2*—This concept is similar to Concept 1; however, the motor is oriented vertically. Only one set of bevel gears is needed to connect the motor to the impact assembly. Additionally, a counterweight (W) is added to the design.
- *Concept 3*—In this concept, we directly attach the motor to the impact assembly, obviating the need for any bevel gears. The counterweight from Concept 2 is included to help improve balance.
- *Concept 4*—In this concept, we orient the impact assembly vertically, with a set of bevel gears at the output to allow the user to drive fasteners horizontally. A gear train consisting of four spur gears (S) connects the impact driver to the motor, which is also oriented vertically. The trigger assembly is located directly above the motor.
- *Concept 5*—This concept is similar to Concept 4; however, the motor is located directly below the trigger assembly and the impact assembly. No gear train is needed in this concept as the motor is directly in line with the impact assembly.

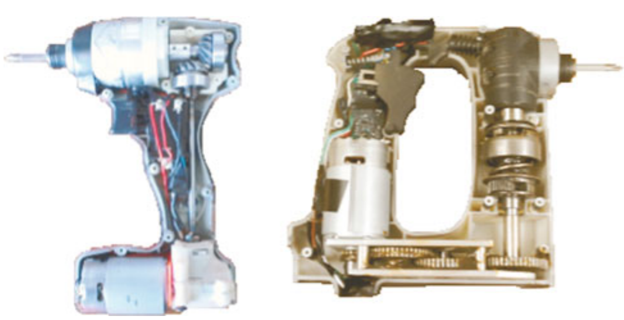

Fig. 9 Two existing functional prototypes for new impact driver. The *left* prototype corresponds to Concept 1 in the case study. The *right* prototype is represented by Concept 4

We developed five separate models to analyze the concepts. The model inputs (**y**) and outputs (**z**) are summarized in Table 1. While the inputs to every model vary, each model includes estimates for the total mass, center of mass, total cost, torque output, speed output, and various other outputs of interest. The total mass and cost in each model are calculated by summing the masses and costs of the individual components comprising each concept in Fig. 10; the outer plastic shell that encases the impact driver is not included. For every model, the center of mass is calculated about the origin, which is defined as the upper corner of the trigger assembly that faces the front of the impact driver (see Fig. 10); this was chosen because we assume the ideal center of mass of the impact driver to be at that point, which is approximately true for the commercially available 12V Hitachi impact driver. The torque and speed outputs are determined with kinematic equations for gear trains, while the stresses on the drive shafts are calculated using accepted strength of materials equations. Specifications for the motors, gears, shafts, and counterweights in the models were obtained from various online catalogs and retailers. Gears are given three discrete options for both material (MC901 nylon, S45C steel, and SUS303 stainless steel) and number of teeth (20, 40, 80). Shafts and counterweights are given discrete options for material (S45C steel, 6061 aluminum, and ASTMB29 lead) and continuous values for length and diameter. The impact assembly is identical throughout, and constraints are placed to maintain approximately identical maximum torque and speed for the drill, so as to optimize comfort without significantly changing performance. With models defined for each concept, we turn our attention to formulation space exploration.

4.2 Boundary exploration

Recall that a main goal of the design is to reduce arm fatigue for those who use the impact driver for long periods of time. Thus, it is reasonable to begin formulation space exploration with the assumption that our overall objectives for this problem are to (1) minimize the total mass of the impact driver and (2) minimize the Euclidean distance

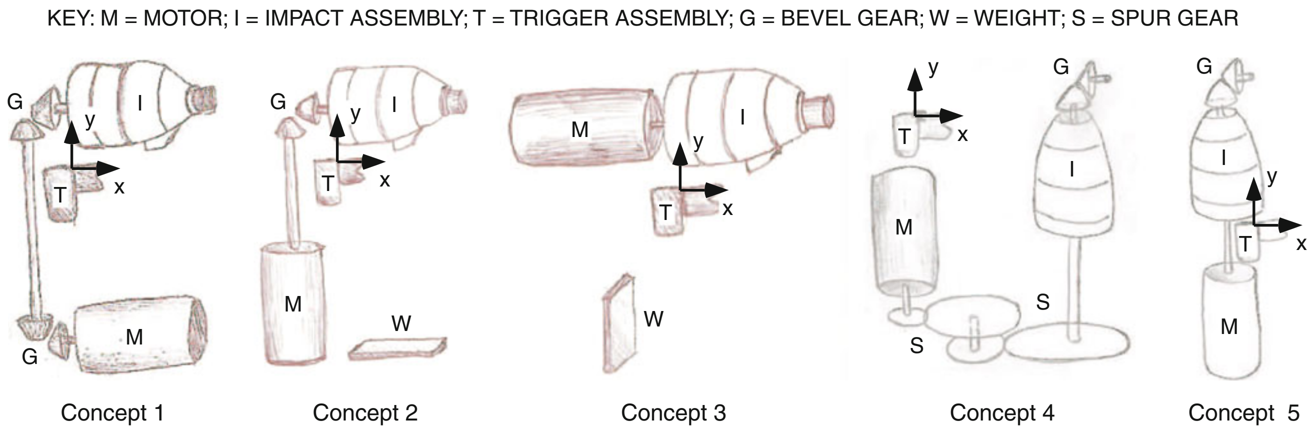

Fig. 10 Five concepts for impact driver with no batteries

between the ideal center of mass (located at the origin of the models) and the actual center of mass of each concept. Nevertheless, other objectives are still unclear at this point; it is in this scenario where boundary exploration is most useful. We will first consider Concept 1. We formulate a preliminary optimization problem, using the process outlined in Fig. 4 (with $n_d = 2$ and $n_k = 1$), and we explore the boundaries for the x and y locations of the center of mass, using the minimization of these locations to the ideal center of mass as temporary objectives for our preliminary problem. The result is shown in Fig. 11. From the plot, we see that the ideal center of mass (0,0) is not possible, given our current optimization formulation. Additionally, we see that the x-location of the center of mass ranges from -30 to 105 mm and the y-location from -90 to -10 mm. Using this data, we can run some worst case scenario experiments with rudimentary, physical prototypes to determine whether the x-location or y-location has a greater effect on arm fatigue and use this information in subsequent optimization formulations.

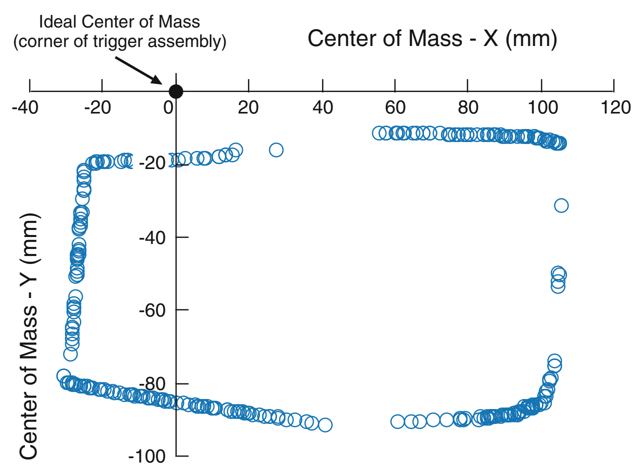

Fig. 11 Boundary exploration for the center of mass of Concept 1

4.3 Formulation modifications

Using the results from the previous section, we formulate a new optimization problem that now includes a third objective: minimize the x-location of the center of mass. The projection of this new 3-dimensional design objective space is plotted in a two-dimensional plane as the dashed lines in Fig. 12 and labeled as $k = 0$ (where each different value of k represents a new formulation or concept). The x-axis in the figure is the total mass of the impact driver in grams, and the y-axis is the distance to the ideal center of mass in millimeters. An architectural layout for one design alternative on the Pareto frontier of this formulation is depicted on the right in the plot; the labels are the same as those in Fig. 10. Notice that the vertical drive shaft of this design is relatively distant from the trigger assembly. From a design usability standpoint, we would hope that the vertical shaft would fit inside the impact driver handle, along with the trigger assembly. In this particular design alternative, it is clear that in order for both components to fit inside the handle, the handle needs to be quite large—too large, in fact, for a hand to grip easily. Moreover, the vertical distance between the impact assembly and the motor is too small. In other words, this space is not pragmatic or valid and by definition does not contribute to the formulation space.

With what we have learned from $k = 0$, we reformulate the optimization problem ($k = 1$) with an added objective: to minimize the distance between the trigger assembly and the vertical shaft. We also update the constraint on the y-location of the motor to allow more vertical space for a hand to grip the impact driver. The resulting design space is shown as the region enclosed by solid lines in Fig. 12. It is worth noting that the revision of the formulation to produce $k = 1$ has apparently decreased the overall quality of the Pareto frontier by moving it further from the origin, even though this new design space represents a more usable solution in practice. This re-emphasizes the need for keeping the designer in the loop during design activities so that adjustments to the formulation can be made dynamically that corresponds to the designer's intuition. A design alternative from this Pareto frontier is depicted on the left—notice that there is no horizontal space between the vertical drive shaft and the trigger assembly, and there is adequate vertical space between the impact assembly and the motor. As seen here, visualization of optimization results is critical to effective formulation space exploration. In this case, the architectural layouts are generated by a concept analytical model and, while low in fidelity, provide adequate information to decision-makers. However, visualization of design alternatives is not always practical, nor is it possible to directly and simultaneously plot formulation spaces that exist in more than three dimensions. Optimization visualization is a topic of ongoing research, and several methods exist that could potentially facilitate formulation space exploration (Blasco et al. 2008; Huang and Bloebaum 2004; Jones 1996; Stump et al. 2009).

4.4 Targeted boundary expansion

Suppose that we want to learn the minimum amount of change to our current optimization formulation ($k = 1$) that would result in an objective space that contains the following point of interest: (mass = 600 g, distance to ideal center of mass = 30 mm). The star in Fig. 13a represents this point of interest. Using *P*4, we allow an optimization algorithm to modify the current lower and upper bounds of the weight of the motor, the length of the motor, the torque output of the motor, and the shaft stresses within new ranges that we define. These ranges are contained in $\mathbf{x}_l^{(-)}, \mathbf{x}_l^{(+)}, \mathbf{x}_u^{(-)}$, and $\mathbf{x}_u^{(+)}$. Solving *P*4 results in a Pareto

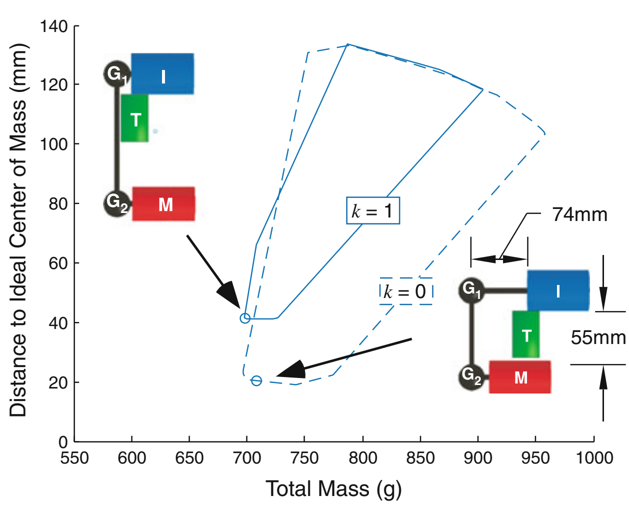

Fig. 12 Visualization of design alternatives for two formulations

frontier of *optimization formulations*, shown in Fig. 13b. The minimum scaled distance to the point of interest is shown on the *x*-axis, and the number of changes to the original optimization formulation is on the y-axis. The optimization formulation identified as $k = 2$ represents the formulation that captured the point of interest with minimal change to the original formulation. Using the lower and upper bounds on **x** from this formulation in *P*2, we plot the design objective space in Fig. 13a with medium solid lines. As shown, this space has the point of interest on its Pareto frontier and contains most of the previous formulation ($k = 1$). Although this region is assumed to be infeasible, we learn that the minimum change to our formulation which would be required in order to obtain the objective values of our point of interest comes from lowering the upper constraint on the motor mass by 95 g and the motor torque by 201 N-mm.

With this information, we can find a different motor for our design that will approximate the results of the targeted boundary expansion. In all previous designs, the motor remained fixed. With a new motor, additional constraints in the formulation are needed to ensure that the torque and speed of the impact driver are appropriate. For example, some gear train combinations would increase the speed at the expense of the torque without preventative constraints. However, the purpose of our optimization is to maintain performance while increasing user comfort, so torque and speed are kept nearly constant. The design space of our new formulation ($k = 3$) is shown in thick solid lines in Fig. 13a. While formulation $k = 3$ does not match the performance of formulation $k = 2$ exactly, it is noticeably better than formulation $k = 1$.

4.5 s-Pareto generation and concept selection

Six more formulations are created for Concept 1 and shown in Fig. 14. As long as the designer finds each explored region to be pragmatic and useful, the union of these regions becomes the formulation space. Using metrics developed in (Curtis et al. 2013), we can quantitate the goodness of this formulation space exploration process in terms of three aspects: novelty, preferred variety, and quality. Novelty is a measure of how expansive our search has been. Preferred variety indicates how well our search has expanded in useful directions. And quality indicates improvement in the "best" design as determined by an aggregate objective function. For the formulation exploration of Concept 1, we get values of 0.35, 0.11, and 0.08 for novelty, preferred variety, and quality, respectively. Each is an indication of improvement over the baseline design space (in this case $k = 1$, since $k = 0$ proved to be infeasible) and provide evidence that our exploration process has added value to our search. As described in that

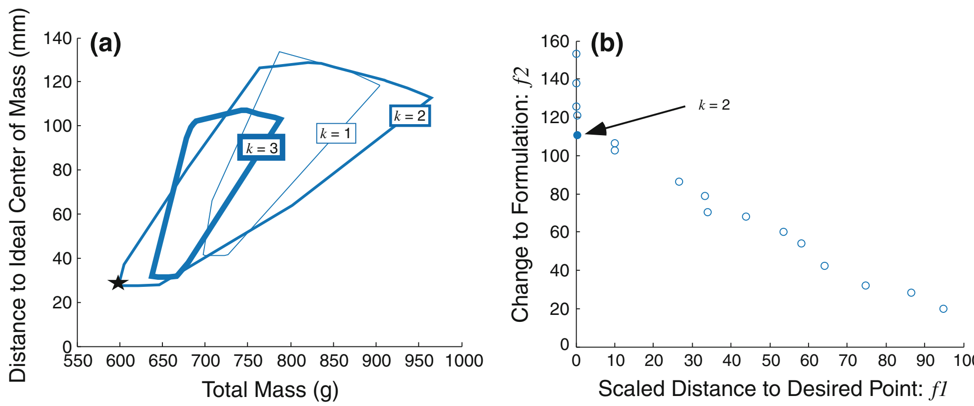

Fig. 13 (a) A point of interest (the *star*) is selected, and an optimization problem is formulated to modify the constraints of the design space in *thin solid lines* ($k = 1$) so that it includes the point of interest. The resulting space is shown with medium *solid lines* ($k = 2$). Another space, shown with *thick solid lines* ($k = 3$), is then formulated to approximate the results of the targeted boundary expansion. **b** This is the pareto frontier of the targeted boundary expansion problem

publication, these metrics can be observed throughout the exploration process to assist the designer in determining when the exploration process is no longer providing sufficient value to be worth continued computational cost.

A similar exploration process is performed for the remaining impact driver concepts. Only one formulation for each concept is displayed and numbered in Fig. 15, for the sake of readability. However, in practice, $P3$ can be used to find the s-Pareto frontier of all the formulations developed for all concepts. Three data points corresponding to the physical prototypes in Fig. 9 and to the Hitachi impact driver are also included in the plot. The asterisk marked with C1 represents the prototype for Concept 1, the asterisk marked with C4 represents the prototype for Concept 4, and the asterisk marked as H represents the 12V Hitachi impact driver. We note that since our models did not include the mass of the plastic casing, we do not include this in the mass of our benchmark designs in the plot. Also, these prototypes use a different motor than the one modeled and therefore fall outside design spaces depicted for these concepts. In the figure, we see that Concept 3 contains the largest portion of the s-Pareto front, and the predicted performance is significantly better than that of the existing prototypes. We hypothesize that if the BYU designers would have had access to this information that was provided by the exploration process, they would have made different decisions and, according the models, would likely have seen better results.

4.6 Limitations

While the results of this new framework and method of design exploration are promising, there are several avenues for improvement that can be made in future research. First,

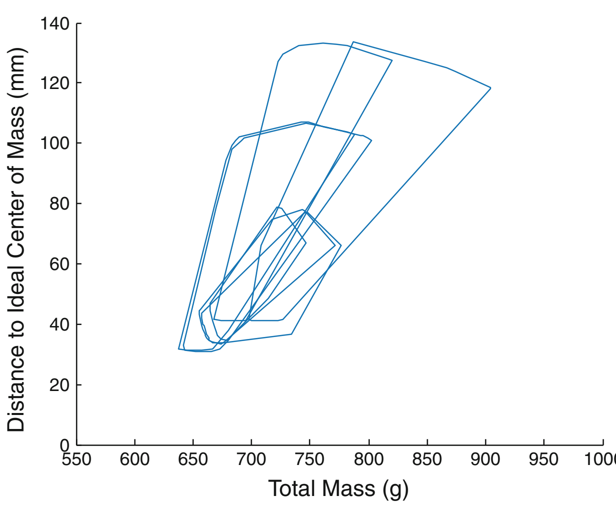

Fig. 14 Formulation space for Concept 1

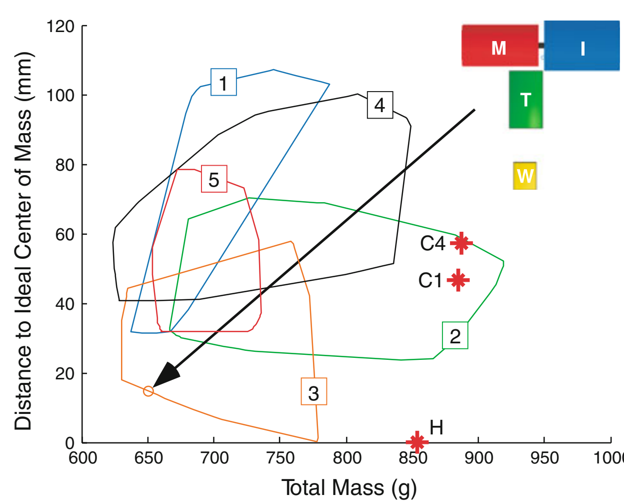

Fig. 15 Comparison of five impact driver concepts. The *asterisk* marked "H" represents the Hitachi WH10DFL, and the *asterisks* marked "C1" and "C4" represent the prototypes from Fig. 9

if the vision for synergistic designer/computer interaction in early-stage design is to be realized, methods for inputting design concepts into the computer and automatically interpreting and parameterizing these concepts need to be developed or improved. Second, the dynamic multiobjective optimization formulation has a few limitations. Using evolutionary algorithms in conjunction with this method of exploration may prove to be computationally prohibitive. Moreover, it requires design objects to be separated into independent and dependent objects, meaning, for example, that a dependent design constraint cannot be implemented directly as a design variable as the formulation currently stands. Finally, as shown in the case study, proper visualization of optimization results can have a significant impact on formulation space exploration. An in-depth study of existing visualization methods and their application to formulation space exploration is warranted.

5 Concluding remarks

We have presented an optimization strategy that facilitates both convergence and divergence during conceptual design. Using this strategy, a computational search is not confined to the search space defined initially by an optimization problem. Instead, a designer may search the formulation space, which we have defined as the set of all feasible design regions identified by the designer as being pragmatic and valid, to form the solution as he or she learns more about the design problem. We have presented three usage scenarios for concept evaluation and selection where a designer could benefit from formulation space exploration. We have shown how to explore formulation space boundaries, generate the s-Pareto frontier in the formulation space, and use targeted boundary expansion to modify existing optimization formulations and expand a search in the direction of infeasible points of interest. We have demonstrated these scenarios on the conceptual design of a novel impact driver. From the results of the case study, we see that the designer can both diverge and converge a design space using the methods presented in this paper and make more informed design decisions with computational assistance earlier in the design process.

Acknowledgements This research was partially supported by National Science Foundation Grant 0954580. The authors also thank the students from the BYU ME 576 class who developed the prototypes in this paper: Morgan Tackett, Brad Smith, Allan Tribe, Prasad Weerakoon, Garrett Bodily, Clark Davis, Sam Wilding, Brad Solomon, Robert Campbell, and David Olligschlager.

References

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

Alvarado CJ, Davis R (2007) Sketchread: a multi-domain sketch recognition engine. In: ACM SIGGRAPH 2007 courses, ACM, New York, NY, USA, SIGGRAPH '07

Antonsson EK, Cagan J (2001) Formal engineering design synthesis. Cambridge University Press, Cambridge

Balling R (1999) Design by shopping: a new paradigm? In: Third world congress of structural and multidisciplinary optimization, vol 1, pp 295–297

Barnum GJ, Mattson CA (2010) A computationally-assisted methodology for preference-guided conceptual design. J Mech Des 132(12):121003

Barrico C, Antunes CH (2006) Robustness analysis in multi-objective optimization using a degree of robustness concept. In: 2006 IEEE congress on evolutionary computation

Blasco X, Herrero JM, Sanchis J, Martínez M (2008) A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. Inf Sci 178(20):3908–3924

Brintrup AM, Ramsden J, Tiwari A (2007) An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. Comput Ind 58:279–291

Brintrup AM, Ramsden J, Takagi H, Tiwari A (2008) Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms. IEEE Trans Evol Comput 12(3):343–354

Cagan J, Campbell MI, Finger S, Tomiyama T (2005) A framework for computational design synthesis: model and applications. J Comput Inf Sci Eng 5:171–181

Chakrabarti A (2002) Engineering design synthesis: understanding, approaches, and tools. Springer, Berlin

Charnes A, Cooper WW (1977) Goal programming and multiple objective optimization. Eur J Oper Res 1:39–54

Curtis SK, Mattson CA, Hancock BJ, Lewis PK (2013) Divergent exploration in design with a dynamic multiobjective optimization formulation. Struct Multidiscip Optim 47(5):645–657

Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657

Davis R (2007) Magic paper: sketch-understanding research. Computer 40(9):34 –41

Deb K, Gupta H (2006) Introducing robustness in multi-objective optimization. Evol Comput 14:463–494

Dye C, Staubach JB, Emmerson D, Jensen CG (2007) CAD-based parametric cross-section designer for gas turbine engine MDO applications. Comput Aided Des Appl 4(1–4):509–518

Farina M, Deb K, Amato P (2004) Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput 8:425–442

Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic Press, New York

Gong D, Yuan J (2011) Large population size IGA with individuals fitness not assigned by user. Appl Soft Comput 11:936–945

Guan S, Chen Q, Mo W (2005) Evolving dynamic multi-objective optimization problems with objective replacement. Artif Intell Rev 23:293

Hassan RA, Crossley RA (2002) Multi-objective optimization of conceptual design of communication satellites with a two-branch tournament genetic algorithm. In: 43rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference

Homan BS, Thornton AC (1998) Precision machine design assistant: A constraint-based tool for the design and evaluation of precision machine tool concepts. Artif Intell Eng Des Anal Manuf 12(5): 419–429

Huang CH, Bloebaum CL (2004) Visualization as a solution aid for multi-objective concurrent subspace optimization in a mulidisciplinary design environment. In: Proceedings of 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY

Huber M, Petersson O, Baier H (2008) Knowledge-based modeling of manufacturing aspects in structural optimization problems. Adv Mater Res 43:111–122

Ishii K (1995) Life-cycle engineering design. J Mech Des 117:42–47

Jones CV (1996) Visualization and optimization. Springer, Berlin

Kasprzak EM, Lewis KE (2001) Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. Struct Multidiscip Optim 22:208–218

Kuehmann CJ, Olson GB (2009) Computational materials design and engineering. Mater Sci Technol 25(4):472–478

Kurtoglu T, Campbell MI (2009) An evaluation scheme for assessing the worth of automatically generated design alternatives. Res Eng Des 20:59–76

Landay JA, Myers BA (2001) Sketching interfaces: toward more human interface design. Computer 34(3):56 –64

LaViola JJ (2011) Mathematical sketching: an approach to making dynamic illustrations. In: Jorge J, Samavati F (eds) Sketch-based interfaces and modeling, Springer, London, pp 81–118

Lewis PK, Murray VR, Mattson CA (2011) A design optimization strategy for creating devices that traverse the Pareto frontier over time. Struct Multidiscip Optim. doi:10.1007/s00158-010-0555-1

Lopez-Mesa B, Bylund N (2011) A study of the use of concept selection methods from inside a company. Res Eng Des 22:7–27

Masry M, Kang DJ, Lipson H (2005) A pen-based freehand sketching interface for progressive construction of 3d objects. J Comput Graph 29:563–575 (Special issue on pen-based user interfaces)

Mattson CA, Messac A (2002) A non-deterministic approach to concept selection using s-Pareto frontiers. In: ASME IDETC/CIE2002, Montreal, Quebec, Canada, DETC2002/DAC-34125

Mattson CA, Messac A (2003) Concept selection using s-Pareto frontiers. AIAA J 41(6):1190–1198

Mattson CA, Messac A (2005) Pareto frontier based concept selection under uncertainty with visualization. Optim Eng 6(1):85–115

Mattson CA, Muller A, Messac A (2009) Case studies in concept exploration and selection with s-Pareto frontiers. Int J Prod Dev 9(1/2/3):32–59 (special issue on space exploration and design optimization

Messac A, Ismail-Yahaya A, Mattson CA (2003) The normalized normal constraint method for generating the Pareto frontier. Struct Multidiscip Optim 25:86–98

Miettinen KM (1999) Nonlinear multiobjective optimization. International Series in Operations Research & Management Science, Kluwer, Dordrecht

Morino L, Bernardini G, Mastroddi F (2006) Multi-disciplinary optimization for the conceptual design of innovative aircraft configurations. Comput Model Eng Sci 13(1):1–18

Mulet E, Vidal R (2008) Heuristic guidelines to support conceptual design. Res Eng Des 19:101–112

Nha CD, Xie YM, Steven GP (1998) An evolutionary structural optimization method for sizing problems with discrete design variables. Comput Struct 68:419–431

Oduguwa V, Roy R, Farrugia D (2007) Development of a soft computing based framework for engineering design optimisation with quantitative and qualitative search spaces. Appl Soft Comput 7(1):166–188

Olewnik AT, Lewis KE (2003) On validating design decision methodologies. In: ASME DETC/DTM 2003, Chicago, ILlli-nois, USA, DETC2003/DTM-48669

Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Engineering design: a systematic approach. Springer, Berlin

Parkinson A, Balling R, Free J, Talbert J, Davidson D, Gritton G, Borup L, Busaker B (1992) OptdesX: a software system for optimal engineering design, user's manual, 1st edn. Design Synthesis, Inc., San Diego

Pugh S (1996) Creating innovative products using total design: the living legacy of Stuart Pugh. Addison-Wesley, Reading, MA

Qazi M, Linshu H (2005) Rapid trajectory optimization using computational intelligence for guidance conceptual design of multistage space launch vehicles. In: AIAA guidance, navigation and control conference

Robertson BF, Radcliffe DF (2009) Impact of cad tools on creative problem solving in engineering design. Comput Aided Des 41(3):136–146

Shelley JK, Giullian NC, Jensen CG (2007) Incorporating computational fluid dynamics into the preliminary design cycle. Comput Aided Des Appl 4(1–4):235–245

Stump G, Lego S, Yukish M, Simpson T, Donndelinger J (2009) Visual steering commands for trade space exploration: User-guided sampling with example. J Comput Inf Sci Eng 9:044501-1–044501-10

Takagi H (2001) Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc IEEE 9(9):1275–1296

Tantar A, Tantar E, Bouvry P (2011) A classification of dynamic multi-objective optimization. In: Proceedings of genetic and evolutionary computation conference (GECCO companion), Dublin, Ireland, 12–16 July 2011, pp 105–106

Ulrich KT, Eppinger SD (2004) Product design and development, 3rd edn. McGraw-Hill/Irwin, New York

Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. J Mech Des 129:370–380

Wang J (2001) Ranking engineering design concepts using a fuzzy outranking preference model. Fuzzy Sets Syst 119:161–170

Wiecek MM, Blouin VY, Fadel GM, Engau A, Hunt BJ, Singh V (2009) Mutli-scenario multi-criteria optimization in engineering design. Multiobject Program Goal Program 618:283–298

Zeleznik R, Miller T, van Dam A, Li C, Tenneson D, Maloney C, LaViola JJ (2008) Applications and issues in pen-centric computing. IEEE Multimed 15:14–21