A Subspace-Inclusive Sampling Method for the Computational Design of Compositionally Graded Alloys

Marshall Allen
Design Systems Laboratory,
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: marshallen16@tamu.edu

Tanner Kirk
Design Systems Laboratory,
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: tannerkirk@tamu.edu

Richard Malak[1]
Design Systems Laboratory,
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: rmalak@tamu.edu

Raymundo Arroyave
Computational Materials Science Laboratory,
Materials Science and Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: rarroyave@tamu.edu

Compositionally graded alloys, a subclass of functionally graded materials (FGMs), utilize localized variations in composition with a single metal part to achieve higher performance than traditional single material parts. In previous work [Kirk, T., Galvan, E., Malak, R., and Arroyave, R., 2018, "Computational Design of Gradient Paths in Additively Manufactured Functionally Graded Materials," J. Mech. Des., 140, p. 111410. 10.1115/1.4040816], the authors presented a computational design methodology that avoids common issues which limit a gradient alloy's feasibility, such as deleterious phases, and optimizes for performance objectives. However, the previous methodology only samples the interior of a composition space, meaning designed gradients must include all elements in the space throughout the gradient. Because even small amounts of additional alloying elements can introduce new deleterious phases, this characteristic often neglects potentially simpler solutions to otherwise unsolvable problems and, consequently, discourages the addition of new elements to the state space. The present work improves upon the previous methodology by introducing a sampling method that includes subspaces with fewer elements in the design search. The new method samples within an artificially expanded form of the state space and projects samples outside the true region to the nearest true subspace. This method is evaluated first by observing the sample distribution in each subspace of a 3D, 4D, and 5D state space. Next, a parametric study in a synthetic 3D problem compares the performance of the new sampling scheme to the previous methodology. Lastly, the updated methodology is applied to design a gradient from stainless steel to equiatomic NiTi that has practical uses such as embedded shape memory actuation and for which the previous methodology fails to find a feasible path. [DOI: 10.1115/1.4053629]

Keywords: design automation, path planning, additive manufacturing, functionally graded alloys, phase diagram, CALPHAD, *3D printing*

1 Introduction

Functionally graded materials (FGMs) are a class of advanced materials that utilize spatial property gradients to satisfy competing performance requirements in different regions of a single part. Typical methods for achieving these property gradients include using a single material and varying its spatial microstructure [1,2] or varying the location and composition of multiple materials within a part [3–6]. Recently, increased adoption and advancements in additive manufacturing (AM) have drawn more attention to the study of FGMs since AM is well suited for customizing both microstructure and composition throughout a monolithic part [7–9].

Compositionally graded alloys are a specific category of multimaterial FGMs which utilize spatial variations in alloy composition to tailor property gradients to the intended use of a specific part. In the past, this has been accomplished through methods such as localized heat treatments or surface engineering [10]. Compared to these traditional methods, much finer control is possible through modern additive manufacturing techniques which provide a straightforward way to locally vary composition in metal parts. Directed energy deposition (DED) in particular has opened the door for manufacturing these materials due to the ability to easily change composition on a layer-by-layer basis [11].

A significant obstacle in printing compositionally graded alloys, however, is planning the gradient path between the alloys of interest. Although some have been successful, many recent attempts to manufacture compositionally graded alloys use linear gradients between the composition path endpoints which resulted in the formation of deleterious phases that ruined the viability of the part [5,6,12–16].

A solution proposed by Hofmann et al. [10] outlined the use of phase diagrams as maps to avoid undesirable phases. This strategy was still limited in that one can only plan paths up to the degree that the diagrams are able to be visualized. Also, these diagrams do not account for variations in temperature during the printing process, and they are also separated from property considerations, limiting design freedom or capability. Further additions to this strategy include the use of Scheil ternary projections [17], which take into consideration the rapid solidification process during additive manufacturing to more accurately predict phase regions.

Kirk et al. [18] presented a novel design method, which synthesizes the use of phase diagrams as maps with a robotic path planning algorithm (rapidly exploring random tree fixed nodes (RRT)*FN [19,20]) to design a path between a prescribed start and goal point while both avoiding deleterious phase regions that have been designated as obstacles and optimizing for a given cost function. The cost function can be encoded to achieve goals like minimizing path length or maximizing obstacle clearance [18]. This method solves many of the issues with the aforementioned approaches. It can plan gradient paths through arbitrarily large composition spaces and include processing variables as constraints or as explicit design variables. Recent improvements by the authors have

[1] Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received June 30, 2021; final manuscript received January 11, 2022; published online February 15, 2022. Assoc. Editor: Pingfeng Wang.

enabled the methodology to design paths with monotonic property profiles which can be used as building blocks to create arbitrary property profiles [21]. Furthermore, this methodology has been used to design compositionally graded alloys which were successfully manufactured using DED [22].

Despite the many advances made in FGM design to successfully tackle the complex problems posed by compositionally graded alloys, the previously developed methodology [18] cannot plan gradient paths through sub-domains because the sampling method only searches the interior of the composition space. This is not typically an issue for robotic path planning algorithms operating in 2D or 3D physical domains, as robots occupy space and therefore cannot move directly into the walls or corners represented by the subspaces. As such, subspace planning in sampling-based planning algorithms is not addressed by current motion planning or path planning literature. In materials design, unlike robotic path planning, sub-domains often represent superior solutions using simpler alloys. For example, spaces with fewer alloys have often been explored more thoroughly through experimentation and are thus less uncertain, providing more robust solutions. Furthermore, in some systems, deleterious phase regions often lie in close proximity to boundaries, where the smallest inclusion of certain elements results in deleterious phases. Thus, adding the search of boundary paths to the current design methodology would result in a more complete and effective search of the design space. This addition is critical, as it would open up the opportunity for solutions through narrow boundary passages and more robust alloy compositions.

In this work, a new sampling method is proposed that includes the boundaries in the path planning search domain. The proposed method utilizes an artificial expansion of the state space, an n-element unit simplex Δ^n, by designating an expanded n-element simplex with the same centroid as the state space and a binary side length scaled by a user-defined multiplier K_s where $K_s \geq 1$. An algorithm from the literature [23] is used to take uniform random samples in the expanded simplex. If any samples lie outside the boundaries of the true state space, they are projected to the nearest location on the state space boundary using a modified method from the literature [24]. This methodology is evaluated and demonstrated through several case studies. First, a breakdown of the sampling distribution throughout all subspaces is investigated for a three-, four-, and five-component alloy system. Then, the new algorithm is applied to a 3D (three-element) synthetic parametric case study where an artificial obstacle is placed along a binary edge within a ternary state space. The parameter of interest in this study is the side length multiplier K_s of the expanded simplex space used for sampling. Lastly, the new method is applied to a real gradient design problem, investigating a gradient from stainless steel to equiatomic NiTi, to demonstrate the effectiveness of the proposed method on a real design problem that is known to require boundary path segments. This specific gradient path has a variety of practical applications such as embedded shape memory actuation. Ultimately, the proposed sampling method is shown to enable the path planning algorithm to locate boundary solutions without incurring any significant computational penalty for finding solutions in the interior.

2 Methods

2.1 Computational Functionally Graded Material Design Methodology. The computational methodology presented in Ref. [18] solves general gradient design problems using a combination of machine learning surrogate models and a robotic path planning algorithm. A gradient design problem is first defined by its state space, Z, which consists of all possible material states within the desired gradient material. A point, z, in this state space represents a unique material composition, though other important processing variables, like temperature, can also be included to concurrently design the processing of the gradient. A continuous path, σ, through this state space represents a potential gradient. Such a path is a continuous function σ: $[0, 1] \rightarrow z$ that relates a path index, $\alpha \in [0, 1]$, to a point in the state space (e.g., a unique material composition).

The state space of a gradient design problem can be divided into an obstacle space, Z_{obs}, and a free space, $Z \setminus Z_{\text{obs}} = Z_{\text{free}}$. The obstacle space is composed of states that should be avoided by the designed path, such as compositions that produce deleterious phases, while the free space is the complement of the obstacle space. A path is *collision-free* if $\sigma(\alpha) \in Z_{\text{free}} \forall \alpha \in [0, 1]$. A path is *feasible* if it is collision-free, $\sigma(0) = z_{\text{init}}$, and $\sigma(1) = z_{\text{goal}}$ where z_{init} and z_{goal} are the initial and goal states, respectively.

The goal of the optimal gradient design problem is to find the feasible gradient path, σ_{best}, that minimizes a user-defined cost function, $c(\sigma)$. Cost functions can simply be path length, but more intricate cost functions can be used to promote design preferences like property monotonicity [21]. The problem formulation for the general gradient design problem is summarized as follows:

$$\begin{aligned} \textbf{Find} \quad & \sigma_{\text{best}} = \underset{\sigma}{\operatorname{argmin}}\ c(\sigma) \\ \textbf{S.t.} \quad & \sigma(\alpha) \in Z_{\text{free}}\ \forall\, \alpha \in [0, 1] \\ & \sigma(0) = z_{\text{init}} \\ & \sigma(1) = z_{\text{goal}} \end{aligned}$$

The computational methodology presented in Ref. [18] solves the aforementioned problem formulation using a combination of materials modeling software, machine learning, and computational path planning, as summarized in Fig. 1. In short, CALculation of PHAse Diagram (CALPHAD) software is used to model relevant thermodynamic data in the state space: namely, deleterious phase fractions at a given composition and temperature. CALPHAD models are generally too computationally expensive to sample directly with sampling-based path planners. As such, CALPHAD data are gathered across the state space and states are labeled as belonging to the obstacle or free region based on deleterious phase fraction or other constrained criteria. The labeled data are then used to train a machine learning classifier that is much cheaper to evaluate. The path planning algorithm then samples the state space, evaluating this obstacle model and the user-defined cost function to check for collision and optimize paths, respectively. If the cost function depends on property information or other quantities that are expensive to evaluate, a machine learning regressor might be necessary to efficiently evaluate path costs [21].

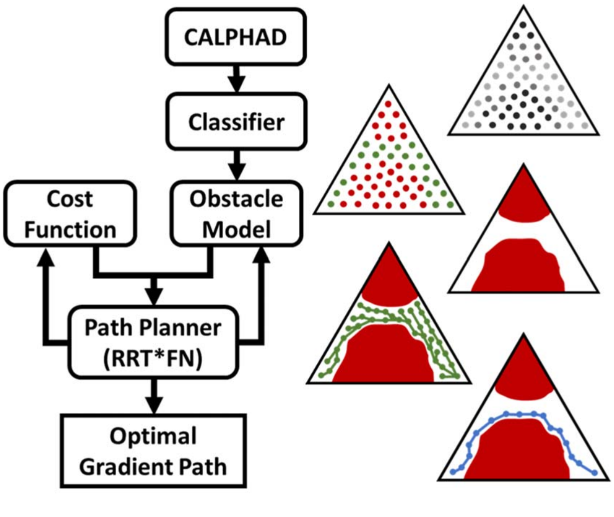

Fig. 1 Computational FGM design methodology presented in Ref. [18]. CALPHAD data are used to train machine learning classifiers that model the obstacles in the state space. These obstacle models and a user-defined cost function are sampled by the path planner (RRT*FN) to find the optimal gradient path through the state space.

The path planning algorithm used by the methodology is a fixed-nodes implementation of the optimal rapidly exploring random tree (RRT). The optimal RRT (RRT*) was developed by Karaman and Frazzoli [20] and provides guarantees of asymptotic optimality and probabilistic completeness. The fixed-nodes implementation (RRT*FN) was developed by Adiyatov and Varol [19] and allows users to limit the memory usage of the algorithm via the introduction of a parameter for the maximum number of nodes.

Put succinctly, RRT*FN is a sampling-based planner that randomly samples the state space to construct a collision-free tree, beginning with the initial state, z_{init}. An iteration of the algorithm begins by randomly sampling the state space. If the new sample lies outside a specified distance from the tree, it is first projected towards the closest node in the tree via a *steering* process. Neighboring nodes within a specified radius of the new sample are then evaluated to determine which potential connection creates the lowest cost path to the new sample while also being collision-free. If a collision-free connection exists, the minimum cost node becomes the new sample's parent node. An additional *rewiring* step then examines the other neighboring nodes and sets the new sample as its parent node if doing so lowers the path cost to that node. Lastly, a *forced removal* step carefully removes suboptimal nodes from the tree if the specified maximum number of nodes is exceeded.

There are several reasons the sampling-based method RRT*FN is preferred for this design methodology instead of graph-based methods such as using a graph search method like Dijkstra's shortest path algorithm to find the optimal path through a pre-computed mesh. Due to the physical nature of the problem, the spatial resolution of the gradient designs is a central concern. Graph search algorithms require a pre-computed fixed graph that defines the cost between adjacent points. In contrast, RRT*FN builds the path as it samples, constructing a path tree that becomes denser overtime as needed to find the optimal solution if it exists [18,19]. The RRT*-based approach produces denser nodes around the optimal path whereas graph-based algorithms must rely on a broad sampling of the entire domain, thereby facing a steeper trade-off in graph-building cost versus resolution. Considering the additional graph-construction time for Dijkstra's algorithm and the superior sampling resolution of RRT*, the scales tip further in support of using an RRT*-based method, which motivated the authors to implement RRT*FN in their methodology [18,25]. In practice, the authors have used RRT*FN to design gradients that were successfully printed using DED [22,25]. The authors have also leveraged the cost-function formulation in RRT*FN to enable the design of compositionally graded alloys for arbitrary property profiles [21]. Thus, not only has this method been thoughtfully selected from modern motion-planning techniques but also, in practice, it has been implemented with a demonstrable record of success [18,21,22,25].

The random sampling procedure that begins each iteration in the RRT*FN process is relatively simple. In typical motion-planning applications (e.g., robotic path planning), a random sample of the state space (e.g., location in space) is simply generated via a random number generator. However, to satisfy the requirement that the composition of all elements in a composition space must sum to unity, the sampling scheme in the gradient methodology transforms randomly generated composition samples to lie within a simplex. This is done by exploiting the fact that uniformly distributed samples on a unit n-element simplex follow an n-dimensional Dirichlet or multivariate beta distribution [26] and that such a distribution can be obtained from a uniform random distribution, U, via the inverse transformation $-\log U$. The sampling procedure of the previous methodology, also discussed by Otis [27], is described as follows:

(1) Let n represent the number of elements in the composition space.
(2) Generate an n-dimensional sample, x^n, from a uniform pseudo-random technique.
(3) Transform each element in the sample to be exponentially distributed: $x_{i,\exp} = -\ln(x_i)$.
(4) Divide each element in the exponentially distributed sample by the sample's sum:

$$x_{i,\text{simplex}} = \frac{x_{i,\exp}}{\sum_{i=1}^{n} x_{i,\exp}}$$

(5) Drop the balance element component from the sample, resulting in a sample of dimension $(n-1)$.

The sampling procedure described above excludes compositional subspaces. For example, if a quaternary (four-element) composition space is sampled with the above procedure, no samples will be generated in the four constituent ternary subspaces or six constituent binaries. This necessitates that these spaces be excluded by potential gradient paths and consequently reduces the space of possible gradient paths that the methodology can design. Consequently, the previous methodology obviously excludes potentially simpler solutions that contain less elements, but can also potentially exclude all feasible paths, resulting in an unsolvable problem, because even small amounts of certain elements can produce deleterious phases.

2.2 Subspace-Inclusive Sampling.

When investigating various methods to include subspace sampling in the RRT*FN algorithm, two potential solutions emerged. The first was to generate a grid on the subspaces of the true state space and to sample from the grid at a given probability. The second method was to sample outside of the true state space and to use some method to project a sample taken outside of the true space onto the subspace boundaries. While the grid method allowed for more precise control over which compositions are included in the sampling, the projection method was more flexible and better aligned with the condition of uniform random sampling necessary to satisfy the guarantees of probabilistic completeness and asymptotic optimality provided by the RRT* algorithm [20]. Thus, the authors ultimately chose to investigate a projection-based method.

The first consideration when developing a projection-based approach for a subspace-inclusive sampling method was defining the position and size of the expanded domain for sampling in order to project samples to the subspace boundaries of the true composition space. The parameter for defining the size of the expanded space (and thereby, the bias of samples towards the subspace boundaries versus the interior) is the binary side length multiplier, hereafter referred to as K_s.

This was accomplished by defining an expanded (n)-dimensional simplex such that its centroid is the same as that of the true unit n-element simplex state space. This *centroid* method is based on the principles that the centroid G of an (n)-dimensional simplex is a function of its vertices given by the set $v_1, \ldots, v_n$, the coordinates of all points in a simplex sum to unity, and scaling the maximum value of the vertex coordinates by the constant K_s results in the side lengths being scaled by the same amount. Thus, the following procedure defines the vertices of an expanded n-element simplex with the same centroid as the unit simplex:

(1) Input $c_{\max} = K_s$ such that $K_s \geq 1$
(2) Compute

$$c_{\min} = \frac{1 - K_s}{n - 1}$$

(3) Define the n vertices of the expanded n-simplex:

$$\begin{aligned} v_1 &= (c_{\max}, c_{\min}, c_{\min}, \ldots, c_{\min}) \\ v_2 &= (c_{\min}, c_{\max}, c_{\min}, \ldots, c_{\min}) \\ &\vdots \\ v_n &= (c_{\min}, c_{\min}, c_{\min}, \ldots, c_{\max}) \end{aligned}$$

Figure 2 visualizes the 3D expanded space and a sample projection example using a 2D visualization where the balance element is

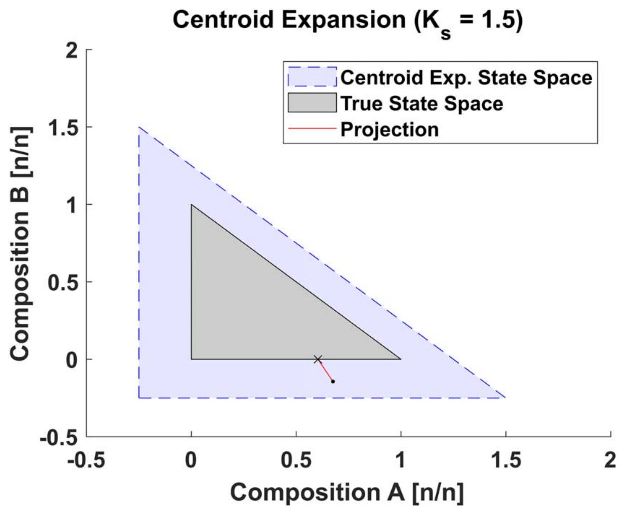

Fig. 2 Centroid-expanded three-element simplex shown for $K_s = 1.5$ in comparison to the true simplex state space with an example projected data point

dropped and therefore located at the origin. The binary side lengths of the expanded simplex are $K_s = 1.5$ times larger than those of the unit simplex. Due to the 2D visualization of the 3D expanded space in Fig. 2, only the hypotenuses perfectly indicate this scaling. By utilizing an equilateral balance-element-inclusive sampling with this centroid expansion, there are no sampling biases among subspaces of a given dimension. Additionally, while other expansion methods were considered, they resulted in subspace biases and were less elegant than the *centroid* method presented in this work.

Having established this definition of the expanded state space, a revised sampling approach is necessary in order to acquire uniform random samples within an arbitrary simplex, as the method previously used was only valid in the true composition space given by the unit n-element simplex. By calculating the vertices of the expanded simplex using the aforementioned procedure, one can apply the method outlined by Grimme for uniform random sampling in an arbitrary simplex [23]. Using barycentric coordinates, in which coordinates are defined in reference to a given simplex, Grimme's method expands a basic method for generating random points in triangles into (n)-dimensions [23,28]. With an arbitrary simplex of dimension n that is defined by its n vertices, v_i, with $i = 1, \ldots, n$, one can generate a random point x using Eq. (1), where $\lambda_j \in (0, 1)$, $j = 0, \ldots, n$, is a list of random weights, and both λ_0 and λ_n are defined as shown [23].

$$x = \sum_{i=1}^{n}\left((1-\lambda_i)\prod_{j=0}^{i-1}\lambda_j\right)v_{\text{i}} \text{ with } \lambda_0 := 1 \text{ and } \lambda_n := 0 \tag{1}$$

With λ_0 and λ_n defined, each remaining weight $\lambda_j \in \lambda_1, \ldots, \lambda_{(n-1)}$ is found from a uniformly distributed random number $z_j \in [0, 1]$ as shown in Eq. (2) [23].

$$\lambda_j = \sqrt{[k]}z_j \text{ with } k := n - j \tag{2}$$

Now that an expanded space and sampling method have been established; the remaining step is to project points outside the true n-element simplex onto the subspace boundaries. Chen and Ye have proposed a simple algorithm for computing the projection onto the unit n-element simplex Δ^n [24]. The addition of both a step eliminating negative coordinates and a subsequent test step to determine if projection is necessary rounds out the algorithm so that it efficiently projects all points outside of Δ^n. The steps of Chen and Ye's algorithm [24], including the added steps, are outlined as follows:

(1) Input point to be projected: $x = (x_1, \ldots x_{(n)})^T \in \mathbb{R}^{(n)}$

(2) (***Added***) For all $(x_i < 0) \in x$, where $i = 1, \ldots, (n)$, set $x_i = 0$

(3) (***Added***) If $(\sum_j^n x_{(j)}) > 1$, proceed to Step 4, otherwise, $x_{\text{proj}} = x$ returns the projection of x onto Δ^n

(4) Sort x in ascending order $x_{(1)} \leq \cdots \leq x_{(n)}$, and set $i = n - 1$

(5) Compute

$$t_i = \frac{\sum_{j=i+1}^{n} x_{(j)} - 1}{(n-i)}$$

(6) If $t_i \geq x_{(i)}$, then set $\hat{t} = t_i$ and proceed to Step 8, otherwise set $i = i - 1$. If $i \geq 1$, then revisit Step 5, otherwise, if $i = 0$, proceed to Step 7

(7) Set

$$\hat{t} = \frac{\sum_{j=1}^{n} x_{(j)} - 1}{n}$$

(8) $x_{\text{proj}} = (x - \hat{t})_+$ returns the projection of x onto Δ^n

With this projection method, an algorithm for subspace sampling has been completed, which can be implemented into the existing RRT*FN algorithm and thus the compositionally graded alloy design methodology. This addition allows for more effective and exhaustive searches of the design space, enabling potentially superior solutions through lower-dimensional regions and even lower-dimensional solutions in problems where a solution does not exist in higher dimensions.

3 Case Studies: Results and Discussion

3.1 Case Study: Sampling Bias. Using the boundary sampling method outlined in the previous section, a sampling bias case study was configured to examine the sampling distribution throughout 3D, 4D, and 5D state spaces and their respective subspaces. If bias exists between subspaces of the same dimension, we would expect to observe a notable difference in the proportion of samples in different subspaces of the same dimension. For this study, the subspaces that contain the balance element were separated from the others to investigate the existence of any sampling bias. In total, 10^6 samples were collected, and the percentage of samples located in each individual subspace were calculated. This process was completed for $K_s = 1.1$ and $K_s = 1.25$ with 10 random runs each and the means and standard deviations were recorded. The results, shown in Table 1, indicate that the chosen sampling method is not biased towards certain subspaces of a given dimension.

As shown in the table, the difference in the proportion of samples between subspaces with the balance element and those without is negligible. This unbiased distribution of samples among subspaces of the same dimension is desirable for path planning, as it is equally probable for a random sample to be taken on any of the subspaces of a given dimension. However, the ideal distribution of samples across subspaces, determined by the value of K_s, for a path planning problem with a state space of dimension n is unknown and will have to be found heuristically.

3.2 Parametric Case Study: Synthetic Ternary. Since the sampling method was shown to successfully capture samples in each subspace, a synthetic case study was constructed to test the effectiveness of the new method when integrated with the path planning algorithm. The goal was to explore the trade-off, in relation to the parameter K_s, between the algorithm's ability to find optimal paths near the subspaces or the interior of the state space. Since the sampling case study revealed that there are no sampling biases among subspaces of the same dimension, this case study was only posed along one of the binaries.

In order to accomplish the goal and gather the necessary observations, the state space was defined by a synthetic ternary (represented by a 3D unit simplex) and an equilateral obstacle with a binary side length $\sqrt{0.5}$. When investigating the algorithm's ability to plan paths

Table 1 Sampling bias case study: subspace sample distribution

			Number of elements in subspace				
			1	2	3	4	5
K_s	n	Subspace contains balance element?	(Number of subspaces) percent of samples in each subspace (mean ± st. dev.)				
1.1	3	Yes	(1) 0.56 ± 0.01	(2) 7.57 ± 0.03	(1) 75.59 ± 0.04		
		No	(2) 0.57 ± 0.01	(1) 7.56 ± 0.02			
	4	Yes	(1) 0.041 ± 0.003	(3) 0.61 ± 0.01	(3) 6.86 ± 0.02	(1) 68.69 ± 0.05	
		No	(3) 0.040 ± 0.002	(3) 0.61 ± 0.01	(1) 6.88 ± 0.03		
	5	Yes	(1) 0.0031 ± 0.0003	(4) 0.049 ± 0.002	(6) 0.59 ± 0.01	(4) 6.25 ± 0.02	(1) 62.41 ± 0.04
		No	(4) 0.0030 ± 0.0004	(6) 0.049 ± 0.002	(4) 0.59 ± 0.01	(1) 6.24 ± 0.03	
1.25	3	Yes	(1) 2.48 ± 0.01	(2) 13.24 ± 0.03	(1) 52.86 ± 0.05		
		No	(2) 2.48 ± 0.02	(1) 13.24 ± 0.03			
	4	Yes	(1) 0.388 ± 0.005	(3) 2.35 ± 0.01	(3) 10.54 ± 0.03	(1) 42.18 ± 0.06	
		No	(3) 0.39 ± 0.01	(3) 2.35 ± 0.02	(1) 10.57 ± 0.03		
	5	Yes	(1) 0.063 ± 0.003	(4) 0.412 ± 0.008	(6) 1.97 ± 0.01	(4) 8.43 ± 0.03	(1) 33.68 ± 0.05
		No	(4) 0.063 ± 0.002	(6) 0.412 ± 0.006	(4) 1.98 ± 0.01	(1) 8.43 ± 0.03	

through a given binary subspace, the obstacle was placed at a distance of 10^{-6} away from the binary to create a narrow passage. This arrangement is representative of a common occurrence in compositionally graded alloy design problems, where, in certain material systems, deleterious phases become present with the slightest addition of a new element, while adjacent subspaces without the new element are completely viable. In order to assess the new algorithm holistically, additional trials were completed where the obstacle was placed so that it blocked the binary subspaces. This forced the algorithm to find a path through the interior so that its performance could be fully compared with the original method which does not sample the subspaces at all. It is worth noting that the original method is identical to the new method when $K_s = 1$ because there is no expansion beyond the true state space, so no boundary projections occur. The prescribed path end points in this study were offset from the binary edges in order to prevent an unfair arrangement benefiting the new method. By offsetting the end points, the problem required the algorithm to plan around the corners of the obstacle to generate the optimal path, rather than the optimal path traveling straight along the boundary. Using this problem formulation, the algorithm was evaluated, with the cost function set to minimize path length, until it reached the goal point (which was given a tolerance radius of 0.05), and then optimized until the cost of the best path was less than or equal to the known optimal path. A 2D projection of this formulation is illustrated, along with the optimal interior and boundary paths, in Fig. 3.

Using this problem formulation, 30 random trials were run for each parameter in the set $K_s \in \{1.05, 1.10, 1.15, 1.20, 1.25, 1.5, 2\}$. The iterations required to find the optimal path, within a goal point tolerance radius of 0.05, were recorded. For the interior problem, the paths were compared to the previous sampling method by adding the results of 30 trials for $K_s = 1$. The previous method was not included in the boundary cases because, as previously noted, it is completely unable to find boundary paths. The results of these trials are shown in the box plot in Fig. 4. The whiskers in this plot were extended to the minimum and maximum instances to indicate the entire spread of the data.

Figure 4 shows the performance trade-off of increasing the expanded binary side length multiplier K_s in finding the optimal path through the interior versus the boundary. As samples become more biased towards the subspace boundaries, the algorithm requires more iterations to find the interior points needed to optimize the interior path, and vice versa for the boundary path. From these results, it is evident that for a ternary state space, K_s values from 1.1 to 1.25 provide a notable increase in boundary path performance while maintaining comparable interior performance to the previous sampling method indicated by $K_s = 1$. It is also worth noting that the distributions shown in the box plot have long whiskers in comparison to the inter-quartile range indicated by the boxes. This comparison illustrates the high variance in the data as a result of the narrow passageway and sharp obstacle corners in the cases of the boundary and interior paths, respectively. Since the path planning algorithm is a sampling-based technique, it must randomly sample within the narrow passage way and close to the sharp corners in order to achieve a cost within tolerance of the optimal path, thus resulting in the high variance evidenced by the large whiskers.

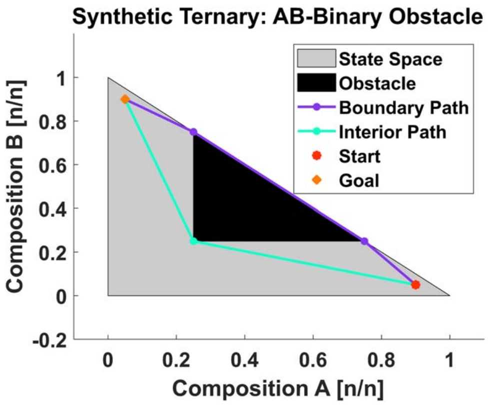

Fig. 3 Problem formulation with the obstacle along the AB-binary-optimal paths for the *interior* path problem (obstacle blocking binary edge) and *boundary* path problem (obstacle 10^{-6} away from binary edge) is shown

3.3 Materials Case Study: Stainless Steel to NiTi.

The new sampling scheme was also tested in a real materials problem to design a compositionally graded alloy between stainless steel (approximated as $Fe_{72}Ni_{12}Cr_{16}$ (at%)) and equiatomic NiTi ($Ni_{50}Ti_{50}$ (at%)), a common shape memory alloy. A gradient between these two materials could enable the creation of monolithic morphing structures where NiTi joints actuate structural transformations and transition to steel plates or members that provide rigidity and support. A continuous gradient would eliminate the need for joining parts like brackets and bolts and enable the direct printing of intricate morphing geometries via additive manufacturing for applications such as extreme-environment solid-state actuators [29].

Unfortunately, the linear gradient between these two materials is rife with deleterious phases. Figure 5 displays this linear gradient and the phases predicted to form at the manufacturing temperature, 1100 °C, along the gradient, as calculated by Thermo-Calc's High Entropy Alloy (TCHEA4) database [30,31]. This value approximates the temperature of the additive manufacturing process and,

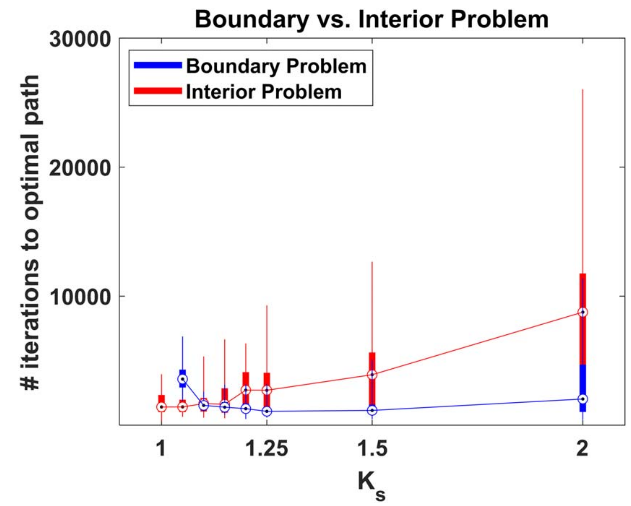

Fig. 4 Number of iterations to reach the known optimal path length within the goal point tolerance of 0.05 for both the *boundary* path problem with the obstacle located 10^{-6} away from the AB-binary and the *interior* path problem with the obstacle blocking the AB-binary.

as such, Fig. 5(*b*) approximates the phases that form during manufacturing. For the linear gradient, these phases include Ni_3Ti and the laves phases at maximums of over 0.35 and 0.6 mole fraction, respectively. Such large fractions of these brittle phases are likely to induce cracking during the build process and will also significantly compromise the mechanical integrity of any potential gradient part. The center of this gradient is also predicted to melt to nearly 50% liquid at this temperature, which would also negatively impact manufacturing. Liquid was considered deleterious when looking at the fixed manufacturing temperature of 1100 °C because it is unknown what compositions it could form upon cooling.

The path planning methodology developed by Kirk et al. [18] was employed to design a new gradient path between these materials without the deleterious phases present in the linear gradient. The Fe–Ni–Cr–Ti composition space formed the state space for this problem, where the Fe composition was left as balance, as shown in Eq. (3) which depicts a point in this state space.

$$z = \left\{ x_{\mathrm{Ni}}, x_{\mathrm{Cr}}, x_{\mathrm{Ti}} : x_{\mathrm{Fe}} = 1 - \sum_{i=1}^{3} x_i \text{ and } x_i \geq 0 \ \forall\, i \right\} \tag{3}$$

The goal of this design problem was to find the path, σ_{best}, that minimizes the defined cost function, $c(\sigma)$, while also satisfying constraints. The chief constraint imposed was to avoid ≥0.01 mole fraction of all deleterious phases ($f_{\mathrm{deleterious}}$) at 1100 °C. Deleterious phases were defined as the *sigma*, *laves*, and *liquid* phases as well as all *intermetallic compounds*. The region of composition space that avoids these phases is called the *free space*, Z_{free}, or the space available for feasible paths. The other constraints ensure feasible paths start at the initial composition, z_{init}, of stainless steel and reach the goal composition, z_{goal}, of NiTi. The cost function was chosen to simply the path length, computed as the sum of every individual segment length, l_k, in a given path. Minimizing this cost function, the path length ultimately minimizes the number of layers required to manufacture the gradient. The gradient design problem formulation is summarized as follows:

$$\begin{aligned}
\textbf{Find} \quad & \sigma_{\mathrm{best}} = \underset{\sigma}{\operatorname{argmin}} \left[c(\sigma) = \sum_{k=1}^{n} l_k \right] \\
\textbf{S.t.} \quad & \sigma(\alpha) \in Z_{\mathrm{free}} \ \forall\, \alpha \in [0, 1] \\
& Z_{\mathrm{free}} = \left\{ z : f_{\mathrm{deleterious}}(z, T) < 0.01, \quad T = 1100\,^{\circ}\mathrm{C} \right\} \\
& \sigma(0) = z_{\mathrm{init}} = \left\{ x_{\mathrm{Fe}} = 0.72,\ x_{\mathrm{Ni}} = 0.12,\ x_{\mathrm{Cr}} = 0.16 \right\} \\
& \sigma(1) = z_{\mathrm{goal}} = \left\{ x_{\mathrm{Ni}} = 0.5,\ x_{\mathrm{Ti}} = 0.5 \right\}
\end{aligned}$$

Two approaches were attempted to solve the aforementioned problem formulation: the original path planning methodology presented in Ref. [18] and the same methodology with the proposed subspace sampling method. In both cases, Halton sequence sampling of the composition space was used to generate compositions for which equilibrium phase formation was predicted by Thermo-Calc's TCHEA4 CALPHAD database at a range of temperatures from 300 to 1800 K [30,31]. However, compositions explicitly from the ternary and binary subspaces were only considered in the second case. CALPHAD information was used to train k-nearest neighbors classifiers ($k = 3$) as surrogate phase models for the total quaternary space and each subspace individually to avoid the influence of data outside a given space on predictions in that space. These models were trained for each of 37 present deleterious phases and combined to create a single obstacle classifier to increase efficiency. All simple cubic, face-centered cubic, and body-centered cubic phases were considered acceptable with all other phases considered as deleterious. Once the manufacturing temperature was known, 5×10^d composition samples were taken from the obstacle classifier at 1100 °C for each subspace using Halton sequence sampling, where d is the number of elements in the subspace (e.g., 50

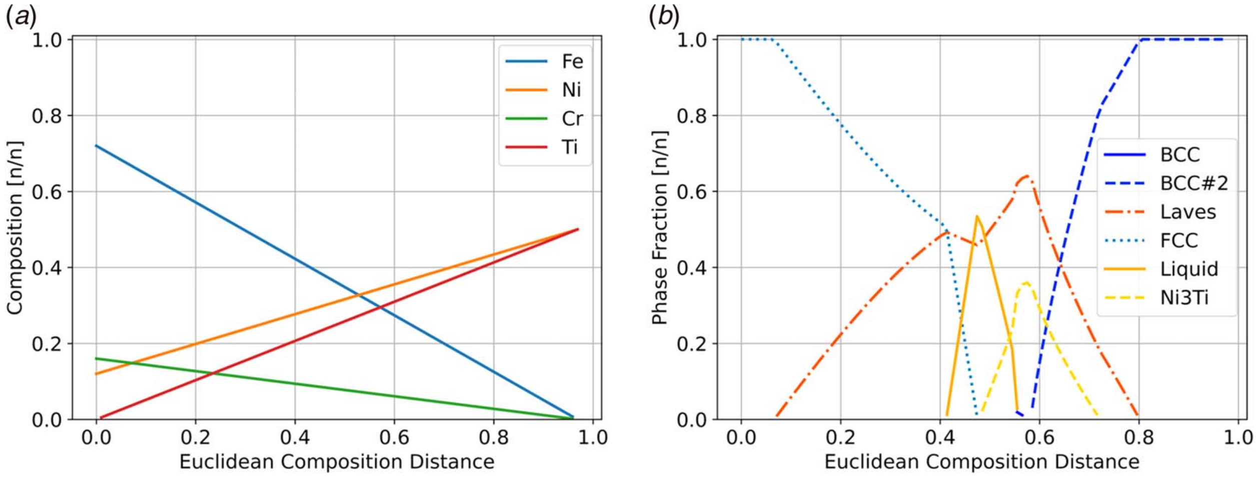

Fig. 5 (*a*) Linear gradient path between stainless steel $Fe_{72}Ni_{12}Cr_{16}$ (at%) and $Ni_{50}Ti_{50}$ (at.%) and(*b*) The mole fraction of deleterious phases predicted to form along this gradient at 1100 °C.

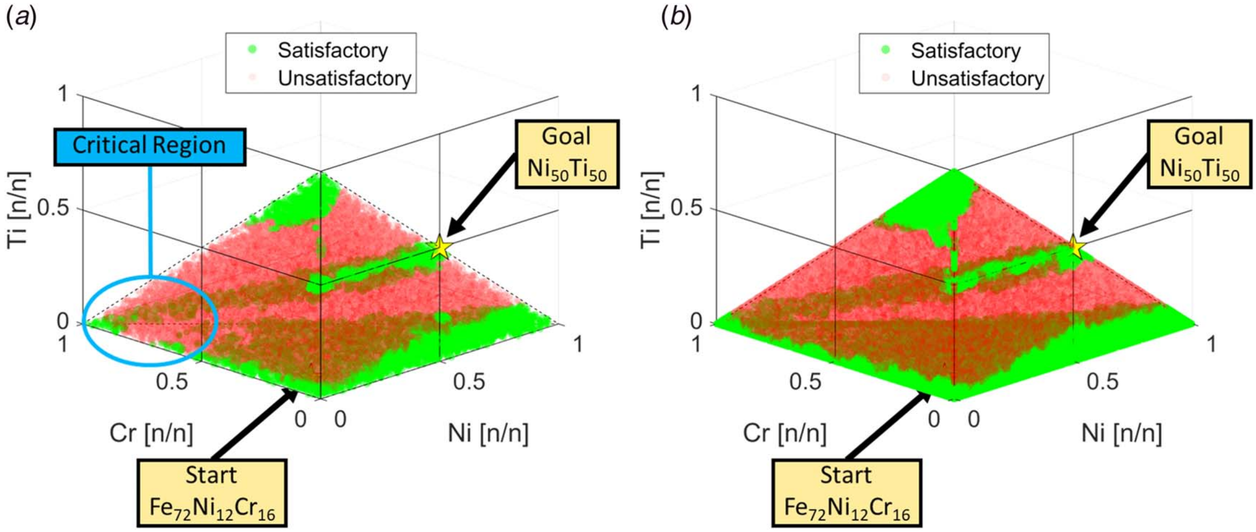

Fig. 6 The Fe–Ni–Cr–Ti state space divided into compositions that contain ≥0.01 mole fraction of deleterious phases at 1100 °C (unsatisfactory) and those that do not (satisfactory) visualize with (*a*) only quaternary data (without subspaces) and (*b*) including ternary and binary compositions. The region of satisfactory compositions forms the free space or the region available for feasible paths. There is no feasible path without considering the subspaces as the free space is disjoint in the critical region shown in (*a*).

for a pure element, 500 for a binary, etc.). These data were used to train a single obstacle classifier at 1100 °C using the same process as the variable-temperature model. Classifier accuracy was estimated with 10-fold cross validation. Only two classifiers produced cross-validated misclassification rates higher than 2%: the Ni–Cr–Ti model at 3.24% and the Fe–Ni–Cr–Ti model at 2.54%. These error rates were considered acceptable for the purpose of modeling gradient path obstacles.

With the trained surrogate model for deleterious phase regions in the quaternary space, the original methodology, without subspace sampling, failed to reach the goal after 1,000,000 iterations of the path planning algorithm. To explain the inability of the original methodology to find a feasible path, the free region and obstacle region of the state space were visualized at the manufacturing temperature of 1100 °C, as shown in Fig. 6. If only the compositions strictly within the Fe–Ni–Cr–Ti quaternary space are considered, there is no feasible path through the free region, as shown in Fig. 6(*a*). The free region (i.e., those compositions without deleterious phases) is not contiguous between the start and goal compositions. The closest potential point of connection is within the critical region shown in Fig. 6(*a*) near pure Cr. In this region, two disjoint regions of free space (one with the start composition and one with the goal composition) come close to connecting, but do not connect due to small amounts of Ti which lead to intermetallic formation. If instead, as shown in Fig. 6(*b*), all ternary and binary subspaces (i.e., faces and edges) are considered, these regions are connected and feasible paths do exist. This is due to the Fe–Ni–Cr ternary (i.e., the bottom face in Fig. 6), which contains almost no deleterious phases at this temperature. This region was ignored by the previous methodology but can now be exploited to find feasible paths with the proposed subspace sampling scheme.

Using obstacle surrogate models informed from the quaternary data and data from all the relevant subsystems, the path planning algorithm with the proposed subspace sampling scheme (expanded side length, K_s, of 1.15) was run for 50,000 iterations. In this time, the algorithm was able to find a feasible path. The best path found by the algorithm after 50,000 iterations is shown in Fig. 7(*a*). This path immediately exploits a subspace as the first half of the gradient is contained entirely within the Fe–Ni–Cr ternary. As mentioned previously, this ternary is entirely devoid of deleterious phases at

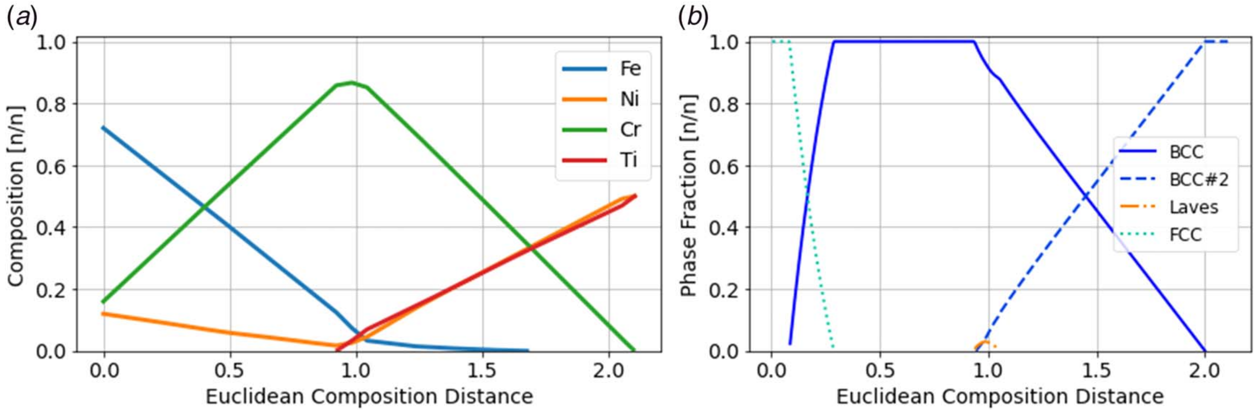

Fig. 7 (*a*) The best gradient path, σ_{best}, obtained by the path planning algorithm with the proposed subspace sampling scheme after 50,000 iterations. Note: Ni and Ti content are near equal in the second half of the gradient and (*b*) The mole fraction of equilibrium phases predicted to form along this gradient at 1100 °C.

Table 2 Effect of binary side length multiplier parameter on path planning efficiency in materials case study

Side length mult., K_s	Iter. to first feasible path (mean ± st. dev.)	Iter. to optimal path (mean ± st. dev.)	CPU time to optimal solution (s) (mean ± st. dev.)
1 (no subspaces)	N/A	N/A	N/A
1.1	15,282 ± 5970	19,792 ± 6833	33.8 ± 82.9
1.2	10,077 ± 2981	20,030 ± 9304	151.4 ± 282.6
1.25	9683 ± 3940	15,831 ± 4390	82.4 ± 91.1
1.3	9636 ± 4508	21,239 ± 10,777	127.8 ± 121.2
1.4	11,063 ± 6030	23,008 ± 14,424	244.7 ± 165.9
1.5	14,912 ± 5132	25,571 ± 13,411	270.6 ± 286.5

Notes: The Iter. to first feasible path column shows the mean and standard deviation of the iterations until the endpoints were connected by a continuous path within the goal point tolerance radius of 0.05. The Iter. to optimal path column shows the mean and standard deviation of the iterations until the path length was optimized to the global minimum within the goal point tolerance radius of 0.05. The time reported in the last column is the mean and standard deviation of the CPU time to achieve the optimal path on an AMD Ryzen 7 2700 Eight-Core Processor at 3.20 GHz with 16.0 GB of RAM.

1100 °C. The gradient path moves along this ternary from stainless steel towards pure Cr, reaching a maximum of nearly 87 at% Cr before adding Ti. In the second half of the gradient, Cr is traded in near equal proportion with equiatomic NiTi to traverse the Cr–Ni–Ti ternary towards the goal point, with the exception of trace amounts of Fe.

The equilibrium phase fractions at 1100 °C along the designed gradient, as predicted directly from Thermo-Calc, are shown in Fig. 7(*b*). The designed gradient is largely free of deleterious phases with the slight exception of a maximum of 0.03 mole fraction of Laves near the center of the gradient. This small amount of a deleterious phase is a consequence of the problem formulation and small errors in the constraint surrogate modeling. By minimizing path length, the problem formulation necessitates that the best path lies along the constraint boundary at some point if any constrained regions lie in the linear path between the initial and goal compositions. Because the constraint modeling technique relies on a surrogate model, there is some error in the representation of the constraint boundary so minimum length paths can sometimes cross these boundaries. Fortunately, the nature of phase diagrams requires that phase fractions decrease to zero at phase boundaries so the fractions of deleterious phases at constraint boundaries are nearly always small. Furthermore, deleterious phases can nearly always be avoided entirely by lengthening the path in the affected area (i.e., increasing the maximum Cr content in this case).

After verifying the success of the proposed sampling scheme, a parametric study was conducted to determine the effect of the expanded side length parameter, K_s, on the efficiency of the path planning process. Ten random runs of the path planning algorithm were conducted for each parameter in the set $K_s \in \{1.1, 1.2, 1.25, 1.3, 1.4, 1.5\}$. The number of iterations until the algorithm obtained a feasible path that reached the goal point (within 5 at%) and the number of iterations until the algorithm reached the solution were recorded. The cost (i.e., length) of the best path shown in Fig. 7 was compared to the cost of the best feasible path at every iteration to determine when the algorithm reached the solution. The results of this parametric study are shown in Table 2 as the mean and standard deviation of both efficiency metrics. Here in Table 2, we see that K_s values from 1.1 to 1.25 reach the true solution in less iterations on average compared to the K_s values from 1.3 to 1.5, with $K_s = 1.25$ being the best on average with a mean of 15,831 iterations to reach the solution. This corresponds with the findings from the synthetic ternary case study, where K_s values from 1.1 to 1.25 indicated a balance between providing boundary solution capability and slowing the algorithm by oversampling the boundaries. However, in this case, we also measured the CPU time to account for the varying amount of function evaluations that occur per iteration. From these results, we see that the best three parameter values in order from fastest to slowest on average are $K_s = 1.1$, 1.25, and 1.3. The standard deviations of iterations to goal, iterations to solution, and CPU time are all quite high relative to the respective mean values in each case. This is due to the narrow passageway in free space near pure Cr and the Cr–Ni–Ti ternary that must be traversed to reach the goal point, as visualized in Fig. 6(*b*). As a sampling-based technique, the path planning algorithm must randomly sample points within this narrow passageway to find a feasible path. The exact iteration at which enough points are sampled in the narrow region to construct a feasible path is therefore subject to high variance. This behavior is further supported by the fact that the iterations to solution is only slightly higher than the iterations to the goal point in all cases. It is worth repeating that without the subspace sampling scheme (when $K_s = 1$), the path planning algorithm failed to reach the goal after even 1,000,000 iterations because no feasible paths are possible, as evidenced by Fig. 6(*a*).

4 Conclusions

In this work, the FGM design methodology previously developed by the authors [18] is improved with the addition of subspace sampling to the path planning algorithm's (RRT*FN) [19,20] sampling process, thus including subspaces in the design search. This development enables the methodology to consider the whole design space: all possible continuous paths containing any combination of constituent elements. It also enables the methodology to provide solutions that are more straightforward, more optimal, or more robust and even enables lower-dimensional solutions in spaces where higher-dimensional solutions might not exist. With the new sampling scheme, new elements can also be added to the state space without the fear that the deleterious phases they might produce could prohibit feasible paths entirely. The addition of subspace sampling could also be useful in other applications of sampling-based path planning algorithms where the subspaces contain potential solutions.

While the new sampling method is limited by the fact that the ideal distribution of samples across subspaces is problem-specific, parametric case studies revealed that the design algorithm's performance is not particularly sensitive to the value of the binary side length multiplier K_s, which controls the sampling distribution. In a ternary, K_s values from 1.1 to 1.25 were shown to be viable parameter settings that provided vast improvement over the past methodology in finding boundary paths, without notably penalizing the algorithm's ability to find paths through the interior of the state space. Further application of the new sampling method indicated that it successfully outperformed the previous algorithm in planning a path from stainless steel to equiatomic NiTi, a compositional gradient with promising applications in embedded actuation. In this instance, the previous methodology was unable to find a feasible path, while the new sampling scheme found optimal paths with relative efficiency. Out of several K_s parameters tested in this quaternary case, the algorithm also ran most efficiently with K_s values from 1.1 to 1.3, indicating that a similar range of values performs well for both ternaries and quaternaries. The algorithm found the optimal solution quickly for this set of K_s values, with the average time to find the optimal path ranging from 33.8 s to 151.4 s. Overall, the algorithm is able to find boundary solutions in a matter of minutes for any $K_s > 1$, the primary difference among the values tested is a minor variation in the time required to obtain an optimal path. While the optimal K_s is problem-specific, optimal paths themselves can be obtained for a wide range of K_s values in a small amount of time.

Acknowledgment

M. A. acknowledges the support of NSF through the project NRT-DESE: Data-Enabled Discovery and Design of Energy Materials (D3EM), NSF-DGSE-1545403.

T. K., R. M., and R. A. acknowledge the support of the DOE through the STTR project Computational Materials Design of Functionally Graded Structures for Enhanced Cooling of Plasma Facing Components via Additive Manufacturing, DE-SC0020032.

Conflict of Interest

There are no conflicts of interest.

Nomenclature

Mathematical Symbols

c = cost function that promotes design preferences

d = number of elements in a subspace

n = number of elements in the total state space

x = sample from an n-element simplex

z = point in state space of the path planning problem

G = centroid of a simplex

T = temperature

U = a uniform random distribution

Z = state space of the path planning problem

$f_{\text{deleterious}}$ = phase fraction of deleterious phases

l_k = length of arbitrary segment k in path

v_i = vertex of an n-element simplex

x_i = composition of element i

z_{init} = initial point in path

z_{goal} = goal point in path

K_s = binary side length multiplier for an expanded simplex

Z_{free} = obstacle-free space

Z_{obs} = obstacle space

S.t. = subject to

α = path index of a point in the state space

Δ^n = n-element unit simplex of dimension n

λ_j = weight for calculating a sample in an arbitrary simplex

σ = continuous path in the state space

σ_{best} = feasible path that minimizes the cost function

References

[1] Aremu, A. O., Brennan-Craddock, J. P. J., Panesar, A., Ashcroft, I. A., Hague, R. J. M., Wildman, R. D., and Tuck, C., 2017, “A Voxel-Based Method of Constructing and Skinning Conformal and Functionally Graded Lattice Structures Suitable for Additive Manufacturing,” Addit. Manuf., **13**(1), pp. 1–13.

[2] Attarilar, S., Salehi, M. T., Al-Fadhalah, K. J., Djavanroodi, F., and Mozafari, M., 2019, “Functionally Graded Titanium Implants: Characteristic Enhancement Induced by Combined Severe Plastic Deformation,” PLoS One, **14**(8), p. e0221491.

[3] Cross, S. R., Woollam, R., Shademan, S., and Schuh, C. A., 2013, “Computational Design and Optimization of Multilayered and Functionally Graded Corrosion Coatings,” Corros. Sci., **77**(12), pp. 297–307.

[4] Simonenko, E. P., Simonenko, N. P., Nikolaev, V. A., Papynov, E. K., Shichalin, O. O., Gridasova, E. A., Maiorov, V. Y., Grishin, A. V., Sevastyanov, V. G., and Kuznetsov, N. T., 2019, “Sol–Gel Synthesis of Functionally Graded SiC–TiC Ceramic Material,” Russ. J. Inorg. Chem., **64**(11), pp. 1456–1463.

[5] Hofmann, D. C., Roberts, S., Otis, R., Kolodziejska, J., Dillon, R. P., Suh, J.-o., Shapiro, A. A., Liu, Z.-K., and Borgonia, J.-P., 2014, “Developing Gradient Metal Alloys Through Radial Deposition Additive Manufacturing,” Sci. Rep., **4**(6), p. 5357.

[6] Bobbio, L. D., Otis, R. A., Borgonia, J. P., Dillon, R. P., Shapiro, A. A., Liu, Z. -K., and Beese, A. M., 2017, “Additive Manufacturing of a Functionally Graded Material From Ti–6Al–4V to Invar: Experimental Characterization and Thermodynamic Calculations,” Acta Mater., **127**(4), pp. 133–142.

[7] Loh, G. H., Pei, E., Harrison, D., and Monzón, M. D., 2018, “An Overview of Functionally Graded Additive Manufacturing,” Addit. Manuf., **23**(10), pp. 34–44.

[8] Vaezi, M., Chianrabutra, S., Mellor, B., and Yang, S., 2013, “Multiple Material Additive Manufacturing—Part 1: A Review,” Virtual Phys. Protot., **8**(1), pp. 19–50.

[9] Oxman, N., 2011, “Variable Property Rapid Prototyping,” Virtual Phys. Protot., **6**(1), pp. 3–31.

[10] Hofmann, D. C., Kolodziejska, J., Roberts, S., Otis, R., Dillon, R. P., Suh, J.-O., Liu, Z.-K., and Borgonia, J.-P., 2014, “Compositionally Graded Metals: A New Frontier of Additive Manufacturing,” J. Mater. Res., **29**(17), pp. 1899–1910.

[11] Schwendner, K. I., Banerjee, R., Collins, P. C., Brice, C. A., and Fraser, H. L., 2001, “Direct Laser Deposition of Alloys From Elemental Powder Blends,” Scr. Mater., **45**(10), pp. 1123–1129.

[12] Reichardt, A., Dillon, R. P., Borgonia, J. P., Shapiro, A. A., McEnerney, B. W., Momose, T., and Hosemann, P., 2016, “Development and Characterization of Ti–6Al–4V to 304L Stainless Steel Gradient Components Fabricated With Laser Deposition Additive Manufacturing,” Mater. Des., **104**(8), pp. 404–413.

[13] Meng, W., Xiaohui, Y., Zhang, W., Junfei, F., Lijie, G., Qunshuang, M., and Bing, C., 2020, “Additive Manufacturing of a Functionally Graded Material From Inconel625 to Ti6Al4V by Laser Synchronous Preheating,” J. Mater. Process. Technol., **275**(1), p. 116368.

[14] Chen, B., Su, Y., Xie, Z., Tan, C., and Feng, J., 2020, “Development and Characterization of 316L/Inconel625 Functionally Graded Material Fabricated by Laser Direct Metal Deposition,” Optics Laser Technol., **123**(3), p. 105916.

[15] Carroll, B. E., Otis, R. A., Borgonia, J. P., Suh, J.-o., Dillon, R. P., Shapiro, A. A., Hofmann, D. C., Liu, Z.-K., and Beese, A. M., 2016, “Functionally Graded Material of 304L Stainless Steel and Inconel 625 Fabricated by Directed Energy Deposition: Characterization and Thermodynamic Modeling,” Acta Mater., **108**(4), pp. 46–54.

[16] Ansari, M., Jabari, E., and Toyserkani, E., 2021, “Opportunities and Challenges in Additive Manufacturing of Functionally Graded Metallic Materials Via Powder-Fed Laser Directed Energy Deposition: A Review,” J. Mater. Process. Technol., **294**(8), p. 117117.

[17] Moustafa, A. R., Durga, A., Lindwall, G., and Cordero, Z. C., 2020, “Scheil Ternary Projection (STeP) Diagrams for Designing Additively Manufactured Functionally Graded Metals,” Addit. Manuf., **32**(3), p. 101008.

[18] Kirk, T., Galvan, E., Malak, R., and Arroyave, R., 2018, “Computational Design of Gradient Paths in Additively Manufactured Functionally Graded Materials,” ASME J. Mech. Des., **140**(11), p. 111410.

[19] Adiyatov, O., and Varol, H. A., 2013, “Rapidly-Exploring Random Tree Based Memory Efficient Motion Planning,” International Conference on Mechatronics and Automation, Takamatsu, Japan, Aug. 4–7, IEEE, pp. 354–359.

[20] Karaman, S., and Frazzoli, E., 2011, “Sampling-Based Algorithms for Optimal Motion Planning,” Int. J. Rob. Res., **30**(7), pp. 846–894.

[21] Kirk, T., Malak, R., and Arroyave, R., 2020, “Computational Design of Compositionally Graded Alloys for Property Monotonicity,” ASME J. Mech. Des., **143**(11), p. 031704.

[22] Eliseeva, O. V., Kirk, T., Samimi, P., Malak, R., Arróyave, R., Elwany, A., and Karaman, I., 2019, “Functionally Graded Materials Through Robotics-Inspired Path Planning,” Mater. Des., **182**(11), p. 107975.

[23] Grimme, Christian, 2015, “Picking a Uniformly Random Point from an Arbitrary Simplex,” ResearchGate.

[24] Chen, Y., and Ye, X., 2011, “Projection Onto a Simplex,” arXiv.

[25] Kirk, T. Q., 2020, “Computational Design of Compositionally Graded Alloys,” Doctoral dissertation, Texas A&M University, Department of Mechanical Engineering, College Station, TX.

[26] Woronow, A., 1993, “Generating Random Numbers on a Simplex,” Comput. Geosci., **19**(1), pp. 81–88.

[27] Otis, R. A., 2016, “Software Architecture for CALPHAD Modeling of Phase Stability and Transformations in Alloy Additive Manufacturing Processes,” Doctoral dissertation, Pennsylvania State University, Department of Materials Science and Engineering, University Park, PA.

[28] Turk, G., 1990, “Generating Random Points in Triangles,” *Graphics Gems*, A. S. Glassner, ed., Academic Press, San Diego, CA, pp. 24–28.

[29] NASA, 2021, “Space Technology Research Grants Program, Early Career Faculty Appendix to NASA Research Announcement (NRA): Space Technology—Research, Development, Demonstration, and Infusion 2021 (SpaceTech–REDDI–2021),” 80HQTR21NOA01.

[30] Thermo-Calc Software High Entropy Alloys Database Version 4, Accessed February 2021.

[31] Mao, H., Chen, H.-L., and Chen, Q., 2017, “TCHEA1: A Thermodynamic Database Not Limited for High Entropy Alloys,” J. Phase Equilib. Diffus., **38**(4), pp. 353–368.