

Dichte-skaliertes Optimierungskriterium für Sliced Latin Hypercube Designs

Julian Bültemeier^{1*}, Marvin Schöne^{2*}, Martin Kohlhase²,
Christoph-Alexander Holst¹, Volker Lohweg¹, Oliver Nelles³

* gleicher Beitrag

¹inIT – Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe,
Campusallee 6, D-32657 Lemgo, Germany
E-Mail: julian.bueltemeier@th-owl.de

²CfADS – Center for Applied Data Science
Hochschule Bielefeld,
Schulstraße 10, 33330 Gütersloh
E-Mail: marvin.schoene@hsbi.de

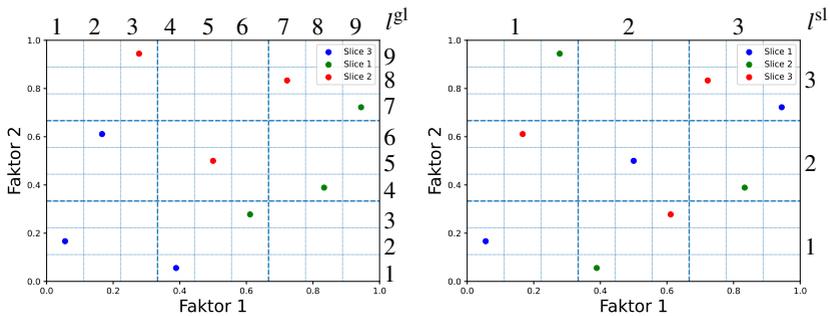
³Universität Siegen,
Paul-Bonatz-Str. 9-11, 57068, Siegen, Germany
E-Mail: oliver.nelles@uni-siegen.de

1 Einleitung

Die Approximation realer Prozesse durch datenbasierte Modelle erfordert in vielen Fällen die Erhebung neuer oder zusätzlicher Daten [1,2]. Dazu eignen sich Methoden des *Design of Experiments* (DOE), die auf eine effiziente Datenerhebung in Form von Versuchsplänen abzielen. Falls für den zu approximierenden Prozess keine Modellannahmen getroffen werden können, sind raumfüllende Versuchspläne vorteilhaft [3]. Etablierte raumfüllende Versuchspläne sind *Latin Hypercube Designs* (LHDs), bei denen der Unterraum jeder der M Eingangsvariablen (Faktor) in gleichverteilte Intervalle (Level) unterteilt und die N Versuchspunkte so platziert werden, dass in jedem Level l^{gl} jeder Dimension genau ein Punkt liegt (vgl. Bild 1a) [4,5].

Damit in mehrdimensionalen LHDs eine angemessene Raumfülle erzielt werden kann, ist eine Optimierung erforderlich, die häufig mit Suchalgorithmen durchgeführt wird. Diese Suchalgorithmen sind auf die Optimierung kontinuierlicher Faktoren beschränkt [6]. Jedoch müssen bei der experimentellen Untersuchung von Prozessen nicht nur kontinuierliche sondern auch kategorische Faktoren berücksichtigt werden. Letztere können beispielsweise durch eine zufällige Zuordnung eines kategorischen Faktors zu jedem Versuchspunkt berücksichtigt werden. Dadurch entsteht für jede Ausprägung des kategorischen Faktors ein sogenannter *Slice*, der eine Untermenge der Versuchspunkte beinhaltet. Bei einer willkürlichen Zuweisung des kategorischen Faktors können jedoch keine raumfüllenden Eigenschaften in den Slices und damit auch nicht für den kategorischen Faktor garantiert werden. Bild 1a veranschaulicht die daraus resultierende Problematik einer ungleichmäßigen Raumabdeckung in den Slices anhand eines zweidimensionalen Datensatzes, bei dem der Einfluss von zwei kontinuierlichen Faktoren für einen kategorischen Faktor mit 3 Ausprägungen experimentell untersucht wird.

Um dieser Problematik zu begegnen, wurden *Sliced Latin Hypercube Designs* (SLHDs) \mathcal{D} entwickelt, mit denen sowohl die Versuchspunkte im Unterraum



- (a) Global optimiertes Design mit zufällig zugewiesenen kategorischen Faktoren mit den N eingezeichneten globalen Level l^{gl} für beide numerischen Faktoren. (b) Design, bei dem sowohl das globale Design als auch die Slices optimiert wurden mit den $K = \frac{N}{T}$ eingezeichneten Slice-Level l^{sl} für beide numerischen Faktoren.

Bild 1: Vergleich von 2 LHDs für die experimentelle Untersuchung des Einflusses von zwei kontinuierlichen Faktoren bei drei verschiedenen kategorischen Ausprägungen bzw. Slices. Jeder Slice in (b) besitzt Versuchspunkte, die sich über den gesamten Eingangsraum erstrecken, wohingegen die Slices in (a) nur einzelne Bereiche abdecken.

der kontinuierlichen Faktoren (globales Design) als auch die Versuchspunkte in den T Slices $\mathcal{D}_1, \dots, \mathcal{D}_T$ optimiert werden können [7]. Hierfür wird der Eingangsraum zusätzlich in $K = \frac{N}{T}$ gleichverteilte Slice-Level l^{sl} unterteilt und die K Versuchspunkte eines jeden Slices \mathcal{D}_t so platziert, dass sich in jedem Slice-Level genau ein Punkt jedes Slices befindet (vgl. Bild 1b). Dies führt dazu, dass die Versuchspunkte sowohl im globalen Design als auch in den einzelnen Slices *Latin Hypercube* (LHC)-Eigenschaften aufweisen, weshalb die Slices in Bild 1b über bessere raumfüllende Eigenschaften als in Bild 1a verfügen.

Bei der Optimierung von SLHDs müssen zwei konkurrierende Optimierungsziele in Form einer Mehrzieloptimierung (auch Pareto-Optimierung genannt) berücksichtigt werden: Die raumfüllenden Eigenschaften des globalen Designs und die raumfüllenden Eigenschaften der Slices. Bei einem Großteil der bestehenden Verfahren zur Optimierung von SLHDs wird jedoch nur das globale Design optimiert [10, 12, 14]. Andere Optimierungsverfahren vereinfachen die aufwendige Pareto-Optimierung, indem sie die beiden konkurrierenden Optimierungsziele additiv zu einem einzigen Optimierungskriterium zusammenfassen. Durch eine unangemessene Gewichtung der beiden Ziele wird jedoch das globale Design in der Optimierung stärker berücksichtigt als die Slices, was zu einem unausgewogenen Pareto-Optimum und damit zu einem suboptimalen Ergebnis für die Slices führen kann [6, 9, 11]. Der Ansatz von CHEN et al. hingegen kann vergleichbare Optima der Slices und des globalen Designs erzielen, ist aber auf SLHDs mit maximal 30 Punkten beschränkt [13]. Ein ausgewogenes Optimum kann mit den bestehenden Verfahren nur erreicht werden, wenn die Gewichtung, wie stark das globale Design und die Slices das Optimierungskriterium beeinflussen sollen, im Rahmen einer aufwendigen Pareto-Optimierung variiert wird.

Um eine Pareto-Optimierung zu vermeiden, wird in diesem Beitrag ein neues dichte-skaliertes Optimierungskriterium vorgestellt, das sowohl das globale Design als auch die Slices gleichermaßen bei einer (Einzel-)Optimierung berücksichtigt. Dieses Kriterium wird mithilfe des *Simulated Annealing* (SA) Algorithmus untersucht, der bei der Optimierung von SLHDs etabliert ist. Im Rahmen der Untersuchung wird nachgewiesen, dass eine Optimierung des dichte-skalierten Kriteriums zu einem ausgewogenen Pareto-Optimum führt.

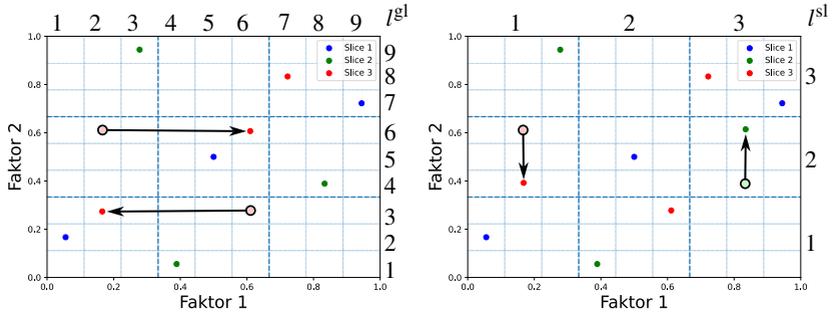
2 Optimierte Sliced Latin Hypercube Designs

Die SLHDs ermöglichen die Erstellung von Versuchsplänen mit kategorischen Faktoren in Slices. Diese Versuchspläne bewahren raumfüllende Eigenschaften und bieten gleichzeitig die Flexibilität, sowohl numerische als auch kategorische Faktoren zu berücksichtigen. Um diese raumfüllenden Eigenschaften zu erreichen, wird eine Optimierung der SLHDs durchgeführt, die typischerweise auf dem Prinzip des Koordinatenaustauschs basiert. Dabei werden die Positionen der Versuchspunkte schrittweise so angepasst, dass ein Kriterium zur Charakterisierung der raumfüllenden Eigenschaften des globalen Designs als auch der verschiedenen Slices optimiert wird.

2.1 Erlaubte Tauschoperationen

Ein Aspekt der SLHD-Optimierung ist der Erhalt der LHC-Eigenschaften innerhalb jedes Slices. Diese Einschränkung führt dazu, dass nur bestimmte Tauschoperationen zulässig sind. Im Rahmen der Optimierung können zwei Arten von Tauschoperationen ausgeführt werden:

1. **Tausch innerhalb eines Slices:** Tauschoperationen zwischen zwei Versuchspunkten $\mathbf{x}(i)$ und $\mathbf{x}(j)$ des selben Slices haben keinen Einfluss auf die LHC-Eigenschaften des jeweiligen Slices. Infolgedessen können beliebige Wechsel von Koordinaten $x_m(i)$ und $x_m(j)$ zweier Versuchspunkte für den Faktor m durchgeführt werden, sofern $\mathbf{x}(i), \mathbf{x}(j) \in \mathcal{D}_l$ gilt. Bild 2a zeigt einen exemplarischen Tausch innerhalb des Slices 3 für den Faktor $m = 1$.
2. **Tausch zwischen zwei Slices:** Damit bei Tauschoperationen zwischen zwei Versuchspunkten unterschiedlicher Slices $\mathbf{x}(i) \in \mathcal{D}_l$ und $\mathbf{x}(j) \in \mathcal{D}_u$ die LHC-Eigenschaften der Slices bestehen bleiben, dürfen nur Koordinaten getauscht werden, die dem selben Slice-Level $l^{\text{sl}} = i$ mit $i \in \{1, \dots, K\}$ zugeordnet sind. In Bild 2b ist ein Tausch zwischen dem Slice 2 und 3 für Faktor $m = 2$ und dem Slice-Level $l^{\text{sl}} = 2$ dargestellt.



(a) Koordinatenaustausch innerhalb des Slices 3 für den Faktor $m = 1$. (b) Koordinatenaustausch zwischen dem Slice 2 und 3 für $m = 2$ und dem Slice-Level $l^{sl} = 2$.

Bild 2: Exemplarische Darstellung eines Koordinatenaustauschs innerhalb eines Slices (a) und zwischen zwei Slices (b). Die schwarzen Pfeile symbolisieren den durchgeführten Tausch.

2.2 Optimierung mit Simulated Annealing

Zur Optimierung von SLHDs wird überwiegend der SA Algorithmus [6, 14] oder eine auf SA basierende Erweiterung [8, 10–13] verwendet. SA ist ein iteratives globales Suchverfahren, bei dem zu Beginn der Suche der Lösungsraum exploriert wird. Mit fortschreitender Optimierung wird die Exploration bis zur Konvergenz der Lösung oder bis zum Erreichen einer maximalen Anzahl von Iterationen reduziert [21]. Die Optimierung erfolgt üblicherweise anhand eines kontinuierlichen distanz-basierten Kriteriums, das die raumfüllenden Eigenschaften des globalen Design und der Slices beschreibt. Algorithmus 1 illustriert einen SA Algorithmus zur iterativen Optimierung von SLHD [6]. Die Optimierung, welche im folgenden näher erläutert wird, basiert auf zufällig ausgewählten eindimensionalen Koordinatenwechselln zwischen zwei Versuchspunkten unter Berücksichtigung der in Kapitel 2.1 aufgeführten Randbedingungen.

Bevor das initiale SLHD optimiert werden kann, muss die Wahrscheinlichkeit eines Koordinatenaustausches innerhalb eines Slices $P_s \in [0, 1]$ und ein Parameter zur Steuerung des Explorationsverhaltens, die Temperatur $\theta \in \mathbb{R}^+$, definiert werden. Außerdem wird in Zeile 1 der optimierte Versuchsplan $\mathcal{D}^* = \mathcal{D}$ initialisiert.

Algorithmus 1.: Simulated Annealing Algorithmus zur Optimierung von SLHDs

Input: Initiales SLHD: \mathcal{D} , Wahrscheinlichkeit innerhalb eines Slice zu tauschen: P_s , Temperatur: θ

Output: Optimiertes SLHD: \mathcal{D}^*

- 1: **Initialisiere:** Optimiertes SLHD: $\mathcal{D}^* \leftarrow \mathcal{D}$
 - 2: **While** Algorithmus ist nicht konvergiert **do**
 - 3: Zufällige Auswahl einer Dimension: $m \leftarrow \text{randSel}(1, \dots, M)$
 - 4: **If** $P_s \leq X \sim U(0, 1)$ **then** \triangleright Koordinatentausch innerhalb eines Slice
 - 5: Zufällige Auswahl eines Slice: $\mathcal{D}_{\text{ex}} \leftarrow \text{randSel}(\mathcal{D}_1, \dots, \mathcal{D}_T)$
 - 6: **Else** \triangleright Koordinatentausch zwischen zwei Slices
 - 7: Zufällige Auswahl eines Slice-Level: $k \leftarrow \text{randSel}(1, \dots, K)$
 - 8: Selektion aller Versuchspunkte innerhalb des Slice-Level:
 $\mathcal{D}_{\text{ex}} \leftarrow \{ \mathbf{x}(n) \in \mathcal{D}^* \mid \frac{T^{(k-1)}}{N} \leq x_m(n) < \frac{T^k}{N} \}$
 - 9: Zufällige Auswahl von zwei Koordinaten: $x_m(i), x_m(j) \leftarrow \text{randSel}(\mathcal{D}_{\text{ex}})$
 - 10: Koordinaten austauschen: $\mathcal{D}_{\text{test}} \leftarrow \text{exchange}(x_m(i), x_m(j))$
 - 11: Berechnung der Wahrscheinlichkeit, ein schlechteres Design zu akzeptieren:
 $P \leftarrow \exp\left(\frac{\phi_p(\mathcal{D}^*) - \phi_p(\mathcal{D}_{\text{test}})}{\theta}\right)$
 - 12: **If** $\phi_p(\mathcal{D}_{\text{test}}) < \phi_p(\mathcal{D}^*)$ **or** $P \leq Z \sim U(0, 1)$ **then** \triangleright Tausch akzeptiert
 - 13: Neues optimiertes SLHD akzeptieren: $\mathcal{D}^* \leftarrow \mathcal{D}_{\text{test}}$
 - 14: Temperatur senken: $\theta \leftarrow 0.9 \cdot \theta$
-

Für jede Iteration wird zunächst in Zeile 3 ein zufälliger Faktor bzw. eine zufällige Dimension m für den Koordinatenaustausch ausgewählt. Abhängig von der Zufallszahl $X \sim U(0, 1)$ werden dann zwei mögliche Tauschpartner entweder innerhalb eines Slices für $P_s \leq X$ (Zeile 5) oder zwischen zwei Slices für $P_s > X$ (Zeile 7-8) ausgewählt. Der Austausch dieser Koordinaten in Zeile 10 führt zu einem neuen Versuchsplan $\mathcal{D}_{\text{test}}$, der ähnlich wie [9, 11, 13] in Abhängigkeit von einem erweiterten ϕ_p -Kriterium

$$\phi_p(\mathcal{D}) = (1 - \beta) \phi_p^{\text{gl}}(\mathcal{D}) + \frac{\beta}{T} \sum_{t=1}^T \phi_p^{\text{sl}}(\mathcal{D}_t). \quad (1)$$

und dem daraus resultierenden Optimierungsziel

$$\mathcal{D}^* = \arg \min_{\mathcal{D}} \phi_p(\mathcal{D}) \quad (2)$$

angenommen oder verworfen wird. Das ϕ_p -Kriterium ist eine kontinuierliche Approximation der minimalen Distanz [15] und wird zur Optimierung von

SLHDs aus einer Aggregation des ϕ_p -Kriterium des globalen Designs

$$\phi_p^{\text{gl}}(\mathcal{D}) = \left(\frac{2}{N(N-1)} \sum_{\mathbf{x}(n) \in \mathcal{D}} \sum_{\mathbf{x}(j) \in \mathcal{D} | j > n} \frac{1}{d(\mathbf{x}(n), \mathbf{x}(j))^p} \right)^{p^{-1}} \quad (3)$$

und einem Mittelwert der Slices

$$\phi_p^{\text{sl}}(\mathcal{D}_t) = \left(\frac{2}{K(K-1)} \sum_{\mathbf{x}(k) \in \mathcal{D}_t} \sum_{\mathbf{x}(j) \in \mathcal{D}_t | j > k} \frac{1}{d(\mathbf{x}(k), \mathbf{x}(j))^p} \right)^{p^{-1}} \quad (4)$$

berechnet, wobei $\beta \in [0, 1]$ die beiden Terme in (1) gewichtet und $d(\mathbf{x}(n), \mathbf{x}(j))$ oder $d(\mathbf{x}(k), \mathbf{x}(j))$ der Abstand (z.B. der euklidische Abstand) zwischen zwei Versuchspunkten ist. Gemäß (2) wird der Versuchsplan als neuer optimaler Plan $\mathcal{D}^* = \mathcal{D}_{\text{test}}$ in Zeile 13 akzeptiert, wenn $\phi_p(\mathcal{D}_{\text{test}}) < \phi_p(\mathcal{D}^*)$ erfüllt ist. Damit der Algorithmus den Suchraum erkunden und lokale Minima vermeiden kann, werden mit einer Wahrscheinlichkeit von $P = \exp\left(\frac{\phi_p(\mathcal{D}^*) - \phi_p(\mathcal{D}_{\text{test}})}{\theta}\right)$ auch Versuchspläne akzeptiert, die (2) nicht weiter minimieren. Nach jeder Iteration wird die Temperatur, die die Wahrscheinlichkeit für die Annahme eines schlechteren Versuchsplans steuert, in Zeile 14 um $\theta = 0,9 \cdot \theta$ verringert, so dass der Algorithmus den Suchraum weniger exploriert.

Die Optimierung wird beendet, wenn der Algorithmus ein bestimmtes Konvergenzkriterium erreicht, z. B. wenn nach einer vorab definierten Anzahl von Iterationen keine Optimierung mehr erreicht werden kann.

3 Dichte-skaliertes Optimierungskriterium

Damit für des globalen Design und die Slices zwei vergleichbare Optima erzielt werden, müssen ähnliche Änderungen im globalen Design und in den Slices auch vergleichbare Wertänderungen der zwei Summanden $\phi_p^{\text{gl}}(\mathcal{D})$ und $\sum_{t=1}^T \phi_p^{\text{sl}}(\mathcal{D}_t)$ bewirken. Unter Idealbedingungen sollten das globale Design und die Slices bei vergleichbaren raumfüllenden Eigenschaften auch die gleiche Größenordnung aufweisen. Diese Voraussetzungen werden aufgrund der folgenden zwei Effekte nicht erfüllt:

1. **Unterschiedliche Wertebereiche:** Eine Erhöhung der Anzahl der Versuchspunkte N in einem LHD bewirkt eine dichtere Verteilung der Punkte in $[0, 1]^M$, was zu geringeren Abständen zwischen den Versuchspunkten führt. Dies hat zur Folge, dass abstands-basierte Kriterien wie ϕ_p bei unterschiedlich großen Versuchsplänen mit identischen raumfüllenden Eigenschaften unterschiedliche Werte annehmen. Aus diesem Grund hat ϕ_p^{gl} in (3) einen größeren Wertebereich als ϕ_p^{sl} in (4).
2. **Unterschiedliche Sensitivitäten:** Ein Koordinatentausch zwischen zwei Punkten kann drastische Veränderungen des ϕ_p -Werts des globalen Designs und der ϕ_p -Werte von bis zu zwei Slices bewirken. Je mehr Slices jedoch vorhanden sind, desto weniger wirken sich diese Veränderungen auf den Mittelwert von $\phi_p^{\text{sl}}(\mathcal{D}_i)$ in (1) aus, da sich die ϕ_p -Werte der verbleibenden Slices nicht ändern.

Folglich optimiert $\phi_p(\mathcal{D})$ in (1) das globale Design stärker als die Slices. Um dieser Unausgewogenheit entgegenzuwirken muss (4) einerseits auf eine vergleichbare Größenordnung wie (3) skaliert werden und andererseits muss der Einfluss der einzelnen Slices erhöht werden, so dass das Optimierungskriterium empfindlicher auf eine Änderung in einem Slice reagiert.

Dichte-basierte Skalierung: Damit ϕ_p^{gl} und ϕ_p^{sl} einen ähnlichen Wertebereich aufweisen, müssen die Versuchspunkte im globalen Design und in den Slices eine vergleichbare Dichteverteilung haben. Um eine vergleichbare Dichteverteilung zwischen den Slices und dem globalen Design zu gewährleisten, werden die Distanzen in den Slices auf einen Wert herunterskaliert, der eine mit dem globalen Design vergleichbare Punktdichte ergibt. Die Punktdichte des globalen Designs errechnet sich im Einheitswürfel aus

$$\rho_g = \frac{N}{1} \quad (5)$$

und in den Slices aus

$$\rho_s = \frac{\frac{N}{T}}{\alpha^M}, \quad (6)$$

wobei α^M das Volumen des Eingangsraums der Slices beschreibt und für eine vergleichbare Punktdichte entsprechend parametrisiert werden muss. Die

Parametrisierung für α ergibt sich aus einer Gleichsetzung von (5) und (6)

$$\alpha = \left(\frac{1}{T}\right)^{\frac{1}{M}} \quad (7)$$

und erweitert (4) durch eine Skalierung der Distanzen zu

$$\phi_p^{\text{sl}}(\mathcal{D}_t) = \left(\frac{2}{K(K-1)} \sum_{\mathbf{x}(k) \in \mathcal{D}_t} \sum_{\mathbf{x}(j) \in \mathcal{D}_t | j > k} \frac{1}{\alpha d(\mathbf{x}(k), \mathbf{x}(j))^p} \right)^{p^{-1}}. \quad (8)$$

Sensitivitätserhöhung der Slices: Um die gleiche Empfindlichkeit zwischen dem globalen Design und den mit α skalierten Slices zu gewährleisten, wird zudem der ϕ_p -Wert der Slices über die Abstände aller Slices mit

$$\phi_p^{\text{sl}}(\mathcal{D}_1, \dots, \mathcal{D}_T) = \left(\frac{2}{N(N-1)} \sum_{t=1}^T \sum_{\mathbf{x}(k) \in \mathcal{D}_t} \sum_{\mathbf{x}(j) \in \mathcal{D}_t | j > k} \frac{1}{\alpha d(\mathbf{x}(k), \mathbf{x}(j))^p} \right)^{p^{-1}} \quad (9)$$

berechnet, statt den ϕ_p -Wert für jeden einzelnen Slice zu berechnen und den Mittelwert zu bilden.

Finales Optimierungskriterium: Basierend auf der skalierten, sensitiveren Berechnung der Slices in (9) kann das Kriterium in (1) zu einem neuen dichte-skalierten Optimierungskriterium

$$\phi_p^{\text{ds}}(\beta, \mathcal{D}) = (1 - \beta) \phi_p^{\text{gl}}(\mathcal{D}) + \beta \phi_p^{\text{sl}}(\mathcal{D}_1, \dots, \mathcal{D}_T) \quad (10)$$

erweitert werden, welches im folgenden hinsichtlich eines ausgewogenen Optimums evaluiert wird.

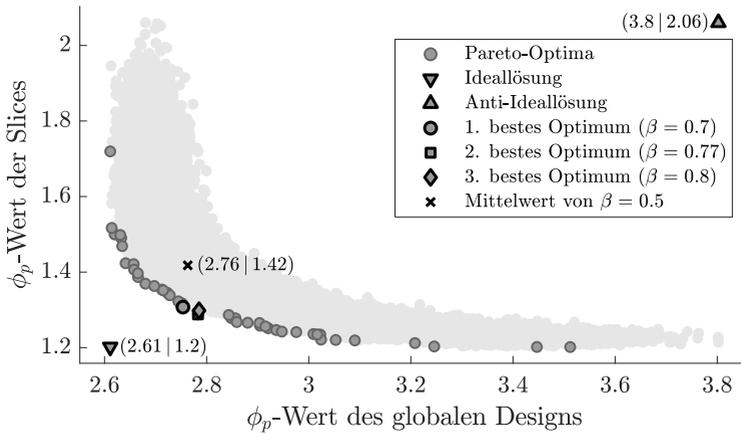
4 Evaluation

Um mit Algorithmus 1 und dem unskalierten Kriterium (1) ein ausgewogenes Optimum zu erzielen, muss β zur Gewichtung der zwei Summanden im Rahmen einer Pareto-Optimierung variiert werden. Zur Erzeugung der Pareto-Menge werden mehrere initiale SLHDs mit verschiedenen β -Werten durch

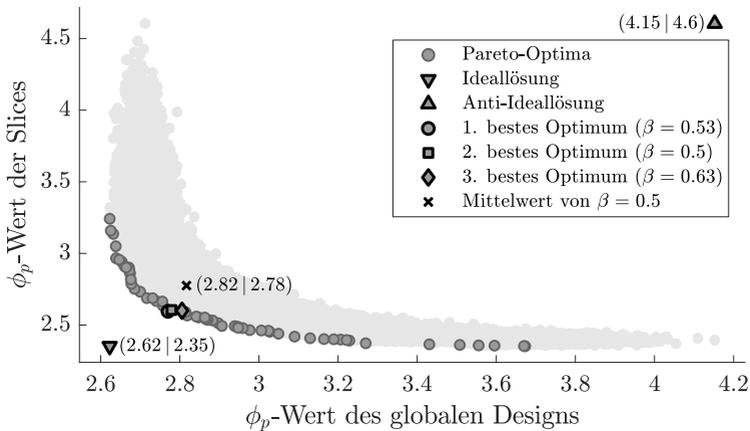
Algorithmus 1 optimiert. Die Auswahl eines Pareto-Optimums, das den besten Kompromiss zwischen den beiden konkurrierenden Optimierungszielen in Form eines ausgewogen optimalen SLHD darstellt, wird mit der Mehrziel-Entscheidungsmethode *Technique for Order Preference by Similarity to an Ideal Solution* (TOPSIS) durchgeführt. TOPSIS wählt aus der Pareto-Front die Lösung aus, die den geringsten Abstand zur Ideallösung und den größten Abstand zur Anti-Ideallösung hat. Die Ideallösung ergibt sich aus den Minima und die Anti-Ideallösung aus den Maxima der erzielbaren ϕ_p -Werte der Slices und des globalen Designs. Um repräsentative Ergebnisse bei der Auswahl einer ausgewogenen Lösung zu erzielen, werden die β -Werte der drei ausgewogensten Pareto-Optima betrachtet.

Bild 3a zeigt die Pareto-Optimierung für einen SLHD mit $N = 90$ Versuchspunkten, $T = 15$ Slices und $M = 4$ Faktoren. Zur Erzeugung der Pareto-Menge wurden 4000 initiale SLHDs durch Algorithmus 1 und (1) mit je 29 verschiedenen β -Werten im Intervall $(0, 1)$ optimiert. Anhand der Pareto-Optimierung wird ersichtlich, dass die ϕ_p -Werte des globalen Designs in einem Intervall von $[2.61, 3.8]$ deutlich größere Werte annehmen können, als die ϕ_p -Werte der Slices, welche auf einen Wertebereich von $[1.2, 2.06]$ begrenzt sind. Optimierungen, die mit $\beta = 0.5$ durchgeführt werden, führen im Mittel zu einem ϕ_p -Wert von 2.76 beim globalen Design und zu einem ϕ_p -Wert von 1.42 bei den Slices. Als die drei besten Pareto-Optima wählt TOPSIS Versuchspläne aus, die mit einem β -Wert von 0.7, 0.77 und 0.8 optimiert wurden. Daraus lässt sich schließen, dass für eine ausgeglichene Optimierung des SLHD ein β -Wert im Intervall von $[0.7, 0.8]$ verwendet werden muss. Zudem kann belegt werden, dass bei $\beta = 0.5$ das globale Design deutlich stärker optimiert wird als die Slices.

In Bild 3b ist die Pareto-Optimierung mit dem neuen dichte-skalierten Optimierungskriterium (10) dargestellt. Im Gegensatz zur Pareto-Optimierung mit dem unskalierten Kriterium weisen die ϕ_p -Werte des globalen Designs mit $[2.62, 4.15]$ und die ϕ_p -Werte der Slices mit $[2.35, 4.6]$ einen vergleichbareren Wertebereich auf. Eine Optimierung mit $\beta = 0.5$ erzielt im Mittel annähernd die selben ϕ_p -Werte für das globale Design (2.82) und die Slices (2.78). Ähnliche Effekte sind bei der Auswahl der drei besten Pareto-Optima zu erkennen, die mit einem β -Wert von 0.53, 0.5 und 0.63 erzielt wurden. Folglich ist mit dem



(a) Pareto-Optimierung mit dem unskalierten Optimierungskriterium in (1)



(b) Pareto-Optimierung mit dem neuen dichte-skalierten Optimierungskriterium in (10)

Bild 3: Pareto-Optimierung eines SLHD mit $N = 90$ Versuchspunkten, $T = 15$ Slices und $M = 4$ Faktoren. Auf der Abszisse ist der erzielte ϕ_p -Wert für das globale Design und auf der Ordinate der (gemittelte) ϕ_p -Wert der Slices dargestellt. Zur Erzeugung der Pareto-Menge wurden eine Vielzahl unterschiedlicher initialer SLHDs mit Algorithmus 1 und variierendem β optimiert. Mit der TOPSIS-Methode werden die drei besten und ausgewogensten Lösungen ausgewählt, um die Verzerrung des unskalierten (a) und dichte-skalierten (b) Kriteriums zu überprüfen und zu vergleichen.

dichte-skalierten Kriterium bei einem β -Wert im Intervall von $[0.5, 0.63]$ ein ausgewogen optimaler SLHD zu erwarten.

5 Zusammenfassung und Ausblick

Um einen optimalen SLHD zu erhalten, müssen sowohl das globale Design als auch die Slices optimiert werden, was eine aufwendig zu berechnende Pareto-Optimierung mit zwei konkurrierenden Optimierungszielen darstellt. Existierende Optimierungskriterien fassen diese beiden Optimierungsziele additiv zu einem einzelnen Kriterium zusammen und überführen somit die Mehrzieloptimierung in eine Einzieloptimierung. Aufgrund einer ungeeigneten Gewichtung der beiden Optimierungsziele wird jedoch das globale Design bei den existierenden Kriterien stärker optimiert als die Slices. Ein ausgewogenes Optimum kann folglich nur durch eine Pareto-Optimierung erreicht werden.

Ausgehend von einem existierenden ϕ_p -Kriterium (1) wurde in dieser Arbeit ein dichte-skaliertes Optimierungskriterium (10) entworfen und evaluiert, das ein annähernd ausgewogenes Optimum ohne die Verwendung einer aufwendigen Pareto-Optimierung ermöglicht. Dafür wurde eine dichte-basierte Skalierung der Distanzen in den Slices entwickelt, wodurch die ϕ_p -Werte des globalen Designs und der Slices vergleichbare Werte annehmen. Außerdem wurde die Empfindlichkeit des Kriteriums bezüglich Distanzänderungen in den Slices erhöht.

Für die Evaluation des neuen Kriteriums wurden dessen Summanden (3) und (9) als konkurrierende Ziele in einer Pareto-Optimierung untersucht. Dabei wurde überprüft, welche Gewichtung (β) für das neue Kriterium zur Erzielung einer ausgewogenen Optimierung notwendig ist. Die Ergebnisse der Evaluation zeigen, dass eine gleichmäßige Gewichtung des globalen Designs und der Slices ($\beta = 0.5$) zu einem annähernd ausgewogenen Optimum führt und beide Summanden ähnliche Wertebereiche bei der Optimierung aufweisen. Bei einer Pareto-Optimierung mit dem existierenden unskalierten Kriterium wurde hingegen bestätigt, dass für ein ausgewogenes Optimum die Slices mindestens im Verhältnis 7:3 gegenüber dem globalen Design gewichtet werden müssen.

In zukünftigen Arbeiten soll die Evaluation des dichte-skalierten Optimierungskriteriums weiter intensiviert werden, z. B. durch die Verwendung von Versuchsplänen mit einer anderen Anzahl an Slices, kontinuierlichen Faktoren und Versuchspunkten. Um neue Optimierungskriterien zu generieren, soll

die dichte-basierte Skalierung zudem auf andere distanz-basierte Metriken angewendet werden. Darüber hinaus sollen im Rahmen eines industriellen Forschungsprojektes mit dem entwickelten dichte-skalierten Kriterium und dem hier vorgestellten SA Algorithmus reale Versuchspläne generiert werden.

Acknowledgment

Dieser Beitrag ist Teil des it's OWL Projekts *AI4ScaDa*, das vom *Ministerium für Wirtschaft, Industrie, Klimaschutz und Energie des Landes Nordrhein-Westfalen* unter der Förderkennzahl 005-2111-0015 gefördert wird.

Literatur

- [1] Pfrommer, Julius and Zimmerling, Clemens and Liu, Jinzhao and Kärger, Luise and Henning, Frank and Beyerer, Jürgen „Optimisation of Manufacturing Process Parameters Using Deep Neural Networks as Surrogate Models“. *Procedia CIRP* 2018.
- [2] Kant, Girish and Sangwan, Kuldip Singh „Predictive Modelling and Optimization of Machining Parameters to Minimize Surface Roughness Using Artificial Neural Network Coupled with Genetic Algorithm“. *Procedia CIRP* 2015.
- [3] Weissman, Steven A. and Anderson, Neal G. „Design of Experiments (DoE) and Process Optimization. A Review of Recent Publications“. *Organic Process Research & Development* 2015.
- [4] McKay, M. D. and Beckman, R. J. and Conover, W. J. „Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code“. *Technometrics* 1979.
- [5] Viana, Felipe A C „Things You Wanted to Know about the Latin Hypercube Design and Were Afraid to Ask“. 10th World Congress on Structural and Multidisciplinary Optimization 2013.

- [6] Ba, Shan and Myers, William R. and Brenneman, William A. „Optimal Sliced Latin Hypercube Designs“. *Technometrics* 2015.
- [7] Qian, Peter Z. G. „Sliced Latin Hypercube Designs“. *Journal of the American Statistical Association* 2012.
- [8] Yang, Xue and Chen, Hao and Liu, Min-Qian „Resolvable Orthogonal Array-Based Uniform Sliced Latin Hypercube Designs“. *Statistics & Probability Letters* 2014.
- [9] Zhang, Jing and Xu, Jin and Jia, Kai and Yin, Yimin and Wang, Zhengming „Optimal Sliced Latin Hypercube Designs with Slices of Arbitrary Run Sizes“. *Statistics & Probability Letters* 2019.
- [10] Yang, Jinyu and Chen, Hao and Lin, Dennis K. J. and Liu, Min-Qian „Construction of Sliced Maximin-Orthogonal Latin Hypercube Designs“. *Statistica Sinica* 2016.
- [11] Yuan, Ru and Guo, Bing and Liu, Min-Qian „Flexible Sliced Latin Hypercube Designs with Slices of Different Sizes“. *Statistical Papers* 2021.
- [12] Chen, Hao and Yang, Jinyu and Lin, Dennis K.J. and Liu, Min-Qian „Sliced Latin Hypercube Designs with Both Branching and Nested Factors“. *Statistics & Probability Letters* 2019.
- [13] Chen, Hao and Huang, Hengzhen and Lin, Dennis K. J. and Liu, Min-Qian „Uniform Sliced Latin Hypercube Designs“. *Applied Stochastic Models in Business and Industry* 2016.
- [14] Guo, Bing and Li, Xiao-Rong and Liu, Min-Qian and Yang, Xue „Construction of Orthogonal General Sliced Latin Hypercube Designs“. *Statistical Papers* 2023.
- [15] Johnson, M.E. and Moore, L.M. and Ylvisaker, D. „Minimax and Maximin Distance Designs“. *Journal of Statistical Planning and Inference* 1990.

- [16] Chen, Ray-Bing and Hsieh, Dai-Ni and Hung, Ying and Wang, Weichung „Optimizing Latin Hypercube Designs by Particle Swarm“. Statistics and Computing 2013.
- [17] Jin, Ruichen and Chen, Wei and Sudjianto, Agus „An Efficient Algorithm for Constructing Optimal Design of Computer Experiments“. Journal of Statistical Planning and Inference 2005.
- [18] Ebert, Tobias and Fischer, Torsten and Belz, Julian and Heinz, Tim Oliver and Kampmann, Geritt and Nelles, Oliver „Extended Deterministic Local Search Algorithm for Maximin Latin Hypercube Designs“. IEEE Symposium Series on Computational Intelligence 2015.
- [19] Hwang, Ching-Lai and Yoon, Kwangsun „Methods for Multiple Attribute Decision Making“. Multiple Attribute Decision Making 1981.
- [20] Méndez, Máximo and Frutos, Mariano and Miguel, Fabio and Aguascalomo, Ricardo „TOPSIS Decision on Approximate Pareto Fronts by Using Evolutionary Algorithms: Application to an Engineering Design Problem“. Mathematics 2020.
- [21] Nelles, Oliver „Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes“. 2020.