

Differential Evolution und dichteste Kugelpackung (3D)

Inhaltsverzeichnis

Differential Evolution und dichteste Kugelpackung (3D)	1
1. Einleitung und Problemstellung	1
1.1. Problem der dichtesten Kugelpackung.....	1
1.2. Differential Evolution	1
1.3. Problemstellung	2
2. Grundlagen und Konzept	2
3. Programmstruktur.....	4
3.1. Benutzeroberfläche.....	4
3.2. Initialisierung	4
3.3. Optimierung.....	4
4. Test und Ergebnisse	7
4.1. Beste Kombination von F und CR	7
4.2. Abhängigkeit von der Populationsgröße	8
4.3. Berechnung von Beispielwerten	8
5. Literaturverzeichnis	8

Symbolverzeichnis

Größe	Beschreibung
x^t	Population
x, y, z	kartesische Koordinaten
n	Anzahl an Mittelpunkten bzw. Kugeln
i	Index: Individuum
j	Index: Mittelpunkt bzw. Kugel
u	Nachkomme
d	Durchmesser
t	Index: Generation
F	Skalierungsfaktor
CR	Rekombinationskonstante
μ	Populationsanzahl/Anzahl an Individuen
max_iter	maximale Anzahl an Iterationen/Generationen
V	Volumen

Differential Evolution und dichteste Kugelpackung (3D)

1. Einleitung und Problemstellung

Im Rahmen dieser Arbeit wird ein Ansatz vorgeschlagen, um das Problem der dichtesten Kugelpackung im euklidischen Raum mit Hilfe evolutionärer Algorithmen hinreichend genau zu lösen. Dabei dient der evolutionäre oder evolutionsorientierte Algorithmus *Differential Evolution* als Grundlage und wird anhand strategischer Parameter für den vorliegenden Anwendungsfall optimiert.

Im Verlauf dieser Niederschrift wird, im Anschluss an die weitere Einleitung, die Problemstellung samt Annahmen beschrieben und genauer spezifiziert. Des Weiteren werden die Herangehensweise (s.a. 2), die Umsetzung (s.a. 3) sowie Beispielergebnisse aufgezeigt und diskutiert.

1.1. Problem der dichtesten Kugelpackung

Die Frage nach der dichtesten Kugelpackung findet ihren Ursprung im 16. Jahrhundert. Der englische Mathematiker Thomas Hariot beschäftigte sich seinerzeit mit der Frage nach der Maximierung von Kanonenkugeln im Laderaum eines Schiffes. Das so entstandene Problem erreichte zu Beginn des 17. Jahrhunderts den deutschen Mathematiker und Astronom Johannes Kepler, der in den nachfolgenden Jahren die *Keplersche Vermutung* aufstellen sollte. Diese besagt, dass die dichteste Packung einer endlichen Zahl gleicher Kugeln einer kubisch-flächenzentrierten oder hexagonalen Packung mit einer maximalen Packungsdichte von 74,05 % ($\pi\sqrt{18}$) entspricht. Entsprechende Vermutung konnte erst zum Ende des 20. Jahrhunderts, durch den US-amerikanischen Mathematiker Thomas Hales und dessen Doktoranden Samuel Ferguson mit Hilfe eines Computerbeweises, bestätigt werden. (GUAN & MIN, 2011)

Somit zeigt die historische Entwicklung, dass die hinreichend genaue Lösung des Problems einer computergestützten Optimierung bedarf.

1.2. Differential Evolution

Die in den 90er Jahren entwickelte Optimierungsmethode *Differential Evolution* geht auf Rainer Storn und Kenneth Price zurück. Das Namenspräfix *Differential* ist dabei dem Grundprinzip des Algorithmus geschuldet, welches auf der Mutationsmethode mittels Differenzvektor (s.a. Gl. 2) zwischen zwei zufällig gewählten Individuen einer Population beruht. Neben der Mutation weist *Differential Evolution* weitere typische Methoden eines evolutionären Algorithmus auf, die Rekombination und die Selektion (s.a. 3). Es sei erwähnt, dass die natürliche Evolution keine Differenzbildung aufweist und die Optimierungsmethode daher als evolutionsnah oder evolutionsorientiert bezeichnet wird. (KOMMER, 2008)

1.3. Problemstellung

Die dichteste Kugelpackung beschreibt die optimale Anordnung von endlich vielen, gleich großen Kugeln im euklidischen Raum. Die optimale Anordnung weist die höchst mögliche Packungsdichte¹ auf. Es gilt eine Verteilung der Kugeln zu finden, die dieser Packungsdichte möglichst nahekommt.

2. Grundlagen und Konzept

Im Folgenden wird der Ansatz der *Differential Evolution* benutzt, um die dichteste Kugelpackung einer definierten Anzahl an Kugeln innerhalb eines Würfels zu finden. Allgemein wird für die Optimierungsmethode *Differential Evolution* zunächst eine Startpopulation initialisiert. Darauf folgt die eigentliche Optimierung über die einzelnen Iterationsschritte. Zunächst werden die Individuen entsprechend Gl. 2 mutiert.

Die mutierten Individuen werden anschließend mit dem entsprechenden Individuum der Parentalgeneration rekombiniert. Zuletzt wird mittels Selektion ermittelt, ob sich das neuentstandene Individuum gegen seinen Vorgänger durchsetzen kann oder nicht. Das Individuum mit der besseren Fitness geht in die nächste Generation ein. Dieser Vorgang (Mutation, Rekombination und Selektion) wird wiederholt, bis ein Abbruchkriterium erfüllt ist.

Dieser Algorithmus, nach der Strategie *DE/rand/1/bin*, wird an das Problem der dichtesten Kugelpackung angepasst. *DE* steht hierbei für *Differential Evolution*, die angewandte Optimierungsmethode. *rand* bedeutet, dass für die Mutation eines Individuums zufällige Basisvektoren ausgewählt werden. Ein Basisvektor ist dabei als ein Individuum der aktuellen Population zu verstehen. Bei der *1* handelt es sich um die Anzahl der zu bildenden Differenzvektoren (s.a. Gl. 2) während der Mutation. Zuletzt ist *bin* eine Abkürzung für Binomial-Schema (s.a. Gl. 3), zur Beschreibung der Rekombination. Demnach wird für die Rekombination zufällig entschieden, ob eine Variable des mutierten Individuums oder eine Variable des entsprechenden Individuums der aktuellen Generation gewählt wird. Der Parameter *CR* bestimmt dabei, wie wahrscheinlich das Überleben des mutierten Individuums ist.

Um eine Annäherung an die dichtest mögliche Kugelpackung zu erhalten, ist es wesentlich, den zu optimierenden Parameter festzulegen. Vorstellbare Ansätze wären zum Beispiel die Anzahl der Kugeln zu optimieren, die in ein definiertes Gefäß passen oder aber das Gefäß selbst bei konstanter Kugelzahl zu minimieren. In dieser Projektarbeit wird der Durchmesser der Kugeln maximiert. Die Anzahl der Kugeln sowie die Größe des Gefäßes² bleiben während der Optimierung gleich. Diese Variante hat den Vorteil, dass man Ergebnisse für jede gewünschte Kugelanzahl erhält. Zudem ist es möglich, einfache Gefäß-Geometrien

¹ Quotient aus Raum- und (Gesamt-)Kugelvolumen

² gemeint sind alle zulässigen Aufenthaltsorte der (Kugel-)Mittelpunkte

sowie -Abmaße zu verwenden. So wird für den Algorithmus ein Einheitswürfel genutzt, der zwischen $[0 \ 0 \ 0]$ und $[1 \ 1 \ 1]$ (euklidischer Raum) aufgespannt ist. Die Mittelpunkte der Kugeln müssen sich allesamt innerhalb dieses Würfels, inklusive seiner Kanten befinden. Das Tatsächliche Volumen ergibt sich somit aus Gl. 1, da für die tatsächliche dichteste Kugelpackung mindestens zwei Kugeln gegenüberliegende Wände berühren müssen³. Somit ist zweimal der Radius zu der Kantenlänge hinzuzufügen.

$$V = (1 + d)^3 \quad \text{Gl. 1}$$

Die Individuen werden definiert als eine legale, endlich abzählbare Menge von Kugeln. Legal bedeutet in diesem Kontext, dass keine Nebenbedingungen durch Elemente des Individuums verletzt werden (s.a. Gl. 4). Die Nebenbedingung die es in diesem Fall einzuhalten gilt, bezieht sich auf die Position des Mittelpunktes einer Kugel (s.o.). Die Population beschreibt die Menge aller Kugelsammlungen.

Elemente eines Individuums zu addieren oder mit einem Skalar zu multiplizieren sowie ihre Distanz untereinander zu bestimmen, entspricht dabei den äquivalenten Operationen für den euklidischen Raum, angewandt auf die Mittelpunkte der Kugeln.

Die Fitness wird wiederum repräsentiert durch den größtmöglichen Durchmesser eines Individuums. Je größer dieser Durchmesser ist, desto besser die Fitness des Individuums.

³ Seien die Kugeln 1 bis n dichtest gepackt. Berührt in einem Würfel keine Kugel die Wand, gibt es stets einen kleineren Würfel, in den weiterhin die gleiche Anzahl an Kugeln passt. Berührt eine Kugel die Wand, kann weiterhin die Kantenlänge des Würfels reduziert werden, ebenso bei zwei benachbarten Wänden. Erst, wenn die Kugeln zwei gegenüberliegende Wänden berühren, kann der Würfel nicht verkleinert werden.

3. Programmstruktur

Im Folgenden soll der strukturelle Ablauf des Programms schrittweise anhand des Strukturdiagramms in Abbildung 2 erläutert werden.

3.1. Benutzeroberfläche

Das Programm stellt in erster Instanz eine intuitive Benutzeroberfläche Abbildung 2 zur Verfügung. Darin kann die Variation der strategischen Parameter (F, CR, μ , max. Iterationen) oder der Kugelanzahl (n) erfolgen. Des Weiteren erfolgt neben einer graphischen Ausgabe des Optimierungsergebnisses eine Ausgabe der erreichten Packungsdichte sowie der Laufzeit des Optimierungsalgorithmus.

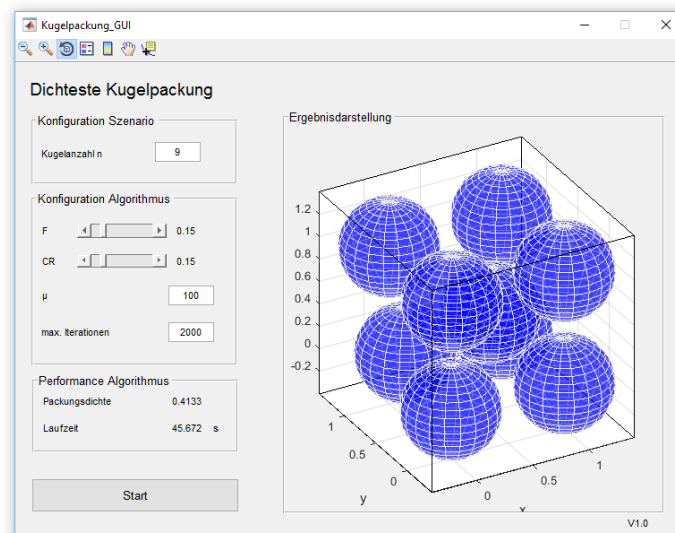


Abbildung 1 Benutzeroberfläche des Programms

3.2. Initialisierung

Im Rahmen der Initialisierung erfolgt die zufällige Generierung der Parentalgeneration $x_{ij}^{t=1}$. Dafür werden die euklidischen-Koordinaten jeder einzelnen Kugel n in einem Intervall zwischen 0 und 1 zufällig ausgewählt. Dieser Vorgang wird dabei, gemäß der Populationsgröße, μ -mal wiederholt, um eine Population aus Individuen (enthält n zufällig verteilte Kugeln bzw. Mittelpunkte) zu erzeugen.

3.3. Optimierung

Die Optimierungsschleife des Algorithmus schließt direkt an die Bildung der Parentalgeneration $x_{ij}^{t=1}$ an und wird solange wiederholt, bis die maximale Anzahl an Iterationen (*max_iter*) erreicht ist.

Während der Optimierung durchlaufen die einzelnen Populationen verschiedene Funktionsblöcke, die als funktionale Grundlage der Optimierungsmethode *Differential Evolution* dienen. Diese sollen nachfolgend erläutert werden.

Mutation und Rekombination von x_{ij}^t :

Hierbei ist zu beachten, dass die Optimierungsmethode *Differential Evolution* einzelne Individuen zunächst mutiert und anschließend rekombiniert.

$$v_{ij}^t = M_1(x_{ij}^t) = x_{r1,j}^t + F \cdot (x_{r2,j}^t - x_{r3,j}^t) \quad \text{Gl. 2}$$

$$r_1 = rand[0,1] \cdot \mu$$

$$r_2 = rand[0,1] \cdot \mu$$

$$r_3 = rand[0,1] \cdot \mu$$

$$r_1 \neq r_2 \wedge r_1 \neq r_3 \wedge r_2 \neq r_3$$

$$u_{ij}^t = R_{bin}(v_{ij}^t) = \begin{cases} v_{ij}^t & \text{falls } rand_j[0,1] < CR \\ x_{ij}^t & \text{sonst} \end{cases} \quad \text{Gl. 3}$$

Überprüfung der Legalität des Kindes u_{ij}^t :

$$u_{ij,legal}^t = L(u_{ij}^t) = \begin{cases} \frac{x_{r1,j}^t}{2} & \text{falls } u_{ij}^t < 0 \\ \frac{(x_{r1,j}^t + 1)}{2} & \text{falls } u_{ij}^t > 1 \\ u_{ij}^t & \text{sonst} \end{cases} \quad \text{Gl. 4}$$

Selektion von $u_{ij,legal}^t$ (Kind) und x_{ij}^t (Elter):

$$x_{ij}^{t+1} = S(u_{ij,legal}^t, x_{ij}^{t+1}) = \begin{cases} u_{ij,legal}^t & \text{falls } f(u_{ij,legal}^t) \geq f(x_{ij}^t) \\ x_{ij}^t & \text{sonst} \end{cases} \quad \text{Gl. 5}$$

$$f(x_{ij}^t) = \min\{\Delta_{lm} \mid l, m = 1, 2, \dots, n; l \neq m\} \text{ mit } l = j \quad \text{Gl. 6}$$

$$\Delta_{lm} = \sqrt{(x_l - x_k)^2 + (y_l - y_k)^2 + (z_l - z_k)^2} \quad \text{Gl. 7}$$

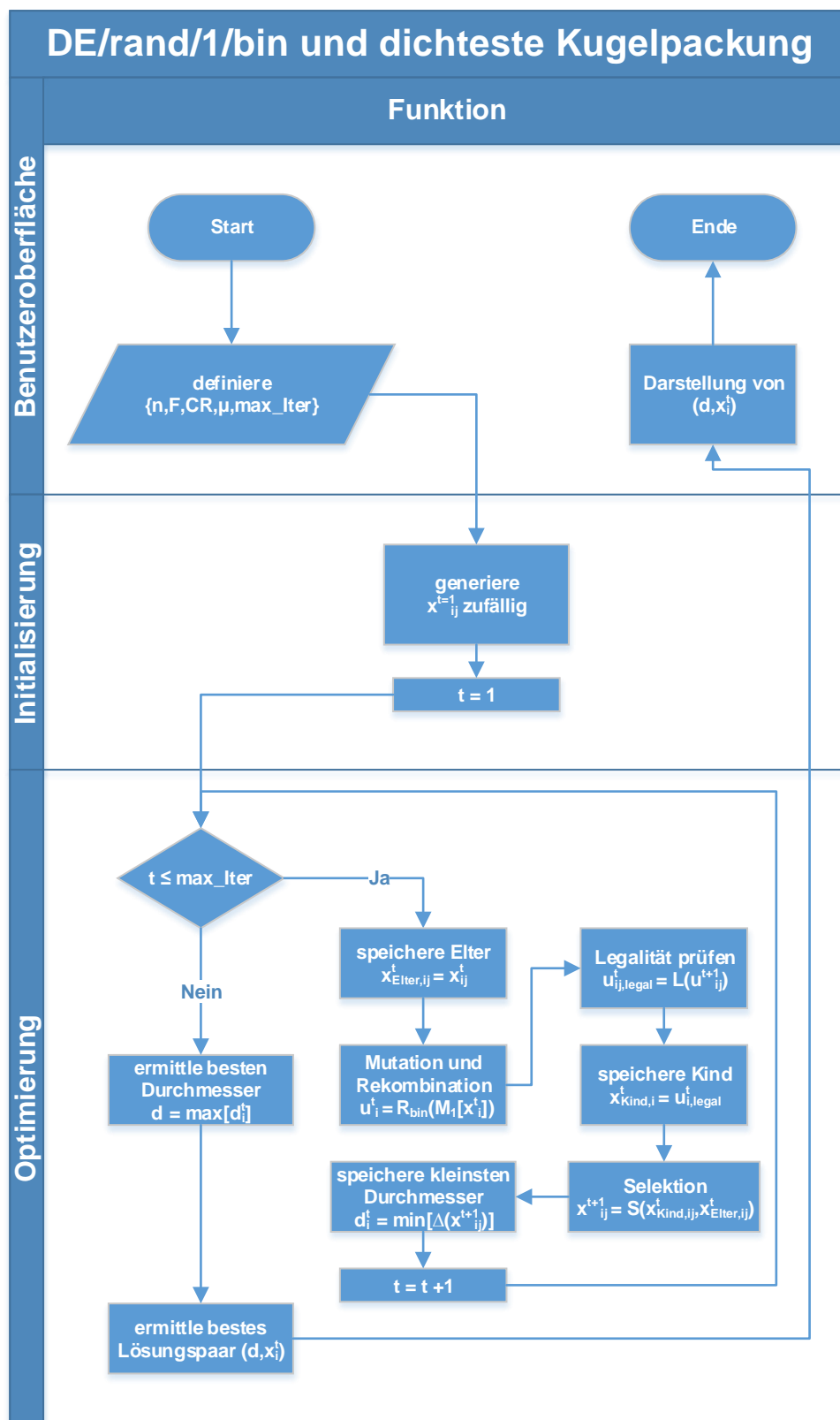


Abbildung 2 (schematisches) Strukturdiagramm des Programmablaufs. Benutzeroberfläche, Initialisierung und Optimierung

4. Test und Ergebnisse

Die strategischen Parameter in dem Algorithmus *Differential Evolution* sind F , CR sowie μ . Nun gilt es passende Werte für diese Parameter zu finden, mit denen die Optimierung zufriedenstellende Ergebnisse bezüglich der Packungsdichte liefert. Als Referenz schlägt (KOMMER, 2008) folgende Werte vor:

$$F = 0,5$$

$$CR = 0,9$$

$$\mu = 10n$$

Im Folgenden werden diese Werte als Ausgang genutzt.

4.1. Beste Kombination von F und CR

Es soll damit begonnen werden F und CR zu bestimmen. Hierzu werden die Werte von F und CR gemäß des Schemas 0,15/0,5/0,9 variiert. Hieraus resultieren neun mögliche Kombinationen, die getestet werden sollen. Dies wird für drei und neun Kugeln durchgeführt.

Tabelle 1 Bestimmung des Einflusses von F und CR

Kugelmanzahl (n)	F	CR	$\mu=10n$	Iterationen	Packungsdichte	maximaler Durchmesser
3	0,15	0,15	30	200	0,3156	1,4138
	0,15	0,5	30	200	0,3077	1,3825
	0,15	0,9	30	200	0,3077	1,3854
	0,5	0,15	30	200	0,3157	1,4142
	0,5	0,5	30	200	0,3157	1,4142
	0,5	0,9	30	200	0,3157	1,4142
	0,9	0,15	30	200	0,3154	1,4128
	0,9	0,5	30	200	0,3157	1,4142
	0,9	0,9	30	200	0,3157	1,4142
9	0,15	0,15	90	2000	0,4703	0,8651
	0,15	0,5	90	2000	0,3288	0,6997
	0,15	0,9	90	2000	0,3223	0,6919
	0,5	0,15	90	2000	0,4707	0,8656
	0,5	0,5	90	3000	0,4707	0,8657
	0,5	0,9	90	3000	0,4648	0,8589
	0,9	0,15	90	4000	0,4047	0,7895
	0,9	0,5	90	4000	0,3220	0,6915
	0,9	0,9	90	4000	0,2207	0,5636

In Tabelle 1 ist der zuvor beschriebene Versuch dokumentiert. Wie aus der Tabelle zu entnehmen, liefert folgende Kombination die besten Ergebnisse für die aufgezeigten Kugelmanzahlen:

1. $F = 0,15$; $CR = 0,15$
2. $F = 0,5$; $CR = 0,15$

Die weiteren Kombinationen von F und CR haben entweder zu schlechte Ergebnisse bei der Packungsdichte erreicht oder eine zu hohe Iterationsanzahl bis zur Konvergenz benötigt. Der

Orientierungswert für F konnte somit bestätigt werden. Der Algorithmus lieferte jedoch bessere Werte für einen CR-Wert von 0,15.

4.2. Abhängigkeit von der Populationsgröße

Nachfolgend soll die Abhängigkeit der erreichbaren Packungsdichte von der Startpopulation μ bestimmt werden. Hierzu werden F und CR auf die im vorherigen Versuch bestimmten Parameter fixiert und μ zwischen $2n$ und $20n$ variiert.

Tabelle 2 Bestimmung der Abhängigkeit von der Populationsgröße

Kugelanzahl (n)	μ	Iterationen	Packungsdichte	maximaler Durchmesser
9	18	2000	0,2345	0,5818
	36	2000	0,4497	0,8415
	54	2000	0,3631	0,7407
	72	2000	0,4585	0,8516
	90	2000	0,4705	0,8653
	108	2000	0,4708	0,8657
	126	2000	0,4709	0,8658
	144	2000	0,4654	0,8595
	162	2000	0,4513	0,8434
	180	2000	0,4711	0,866

Tabelle 2 zeigt den Einfluss der Startpopulation auf die Berechnung der Packungsdichte bei einer Anzahl von 2000 Iterationen. Man bemerkt, dass sich das Ergebnis der Packungsdichte ab einer Startpopulation von $10n$ nur noch unwesentlich verändert. Somit kann der Ausgangswert bestätigt werden.

4.3. Berechnung von Beispielwerten

Mit den zuvor Bestimmten Werten für F, CR und μ wurden nun einige Beispiel berechnet, die in Tabelle 3 zu sehen sind:

Tabelle 3 Berechnung für 2 bis 16 Kugeln

Kugelanzahl	Packungsdichte	maximaler Durchmesser	Kugelanzahl	Packungsdichte	maximaler Durchmesser
			9	0,4682	0,8627
2	0,2659	1,7263	10	0,3721	0,7071
3	0,3006	1,3599	11	0,3929	0,6909
4	0,3801	1,3051	12	0,3587	0,6262
5	0,3629	1,0727	13	0,4793	0,7034
6	0,4080	1,0259	14	0,3983	0,6096
7	0,4581	1,0000	15	0,3967	0,5864
8	0,5236	1,0000	16	0,4099	0,5767

5. Literaturverzeichnis

(GUAN & MIN, 2011) Kepler's sphere packing problem (2011)

(KOMMER, 2008) Differential Evolution: Eine neue evolutionäre Optimierungsmethode
Vdm Verlag, 2008 — ISBN 3-8364-7098-5