Das 3-Peso-Problem bzw. eine Theorie der optimalen Währungsstückelung

Einführung: Bei einem Besuch der Insel Kuba musste ich feststellen, dass es dort kein 2-Peso-Stück gibt (ich rede im folgenden immer von Stück, auch wenn es sich dabei vielleicht um einen Schein handelt) - dafür aber ein 3-Peso-Stück. Wie soll man dies nun bewerten? Ist vielleicht diese Art der Währungsstückelung optimaler als unsere (mit dem 2-Euro-Stück)? Aber in welcher Hinsicht - vielleicht weil man dann im Mittel weniger Geldstücke braucht im täglichen Zahlungsverkehr? Und vielleicht gibt es sogar noch “bessere” Währungsstückelungen. Diese Gedanken gingen mir durch den Kopf. Und so kam es zu dem folgenden Papier.

Definition 1: Mit $\mathbb{A}(n)$ für $n \in \mathbb{N}$ wird der Anfangsteil der natürlichen Zahlen $\{1,2,3,...,n\}$ bezeichnet.

Bemerkung: Es schien mir sinnvoll den zu betrachtenden Geldraum nach oben zu beschränken. Zum einen macht es die folgende Theorie überhaupt erst möglich und schafft auch eine Grundlage für die damit verbundenen Berechnungen, die weiter unten folgen. Im Übrigen entspricht dies aber auch der Praxis, in der es für den Bargeldverkehr sicher eine ganz pragmatische obere Grenze gibt - mag die nun bei 100 oder bei 1000 Euro liegen; darüber hinaus läuft doch alles bargeldlos ab. Man kann dann natürlich auch noch entscheiden, ob man auf Basis ganzer Euros rechnet oder auf Cent-Basis (Faktor 100 ! = mangelnde Rechnerkapazität). Eine Berechnung auf Cent-Basis würde aber sehr wahrscheinlich zu drastischen Abweichungen von der konventionellen Stückelung führen, wie wir aus den folgenden Überlegungen noch sehen werden (es gäbe dann vermutlich nicht mal mehr das 1-Euro-Stück), und damit zu großen Akzeptanzproblemen.

Definition 2: Sei $T=\{t_1,t_2,...,t_m\} \subset \mathbb{A}(n)$ und sei $x \in \mathbb{A}(n)$. Dann heißt $S=(s_1,s_2,...,s_m) \in \mathbb{N}_0^m$ ($\mathbb{N}_0 = \mathbb{N} \cup \{0\}$) eine <u>Stückelung</u> von x bezüglich T, wenn gilt: $x = \sum_{i=1}^{m} s_i t_i$.
Mit $\mathbb{S}_\mathbf{T}(x)$ bezeichnen wir die <u>Menge aller Stückelungen</u> von x bezüglich T.

Definition 3: Sei $x \in \mathbb{A}(n)$, sei $T = \{t_1,t_2,...,t_m\} \subset \mathbb{A}(n)$ und $S=(s_1,s_2,...,s_m)$ eine Stückelung von x bezüglich T.
Dann heißt $\mathrm{Anz}(S) := \sum_{i=1}^{m} s_i$ die <u>Stückanzahl</u> von S.

Beispiel: Mit $T = \{1,5\} \subset \mathbb{A}(9)$ gilt z.B. (1,1) ist eine Stückelung von 6 bezüglich T (weil $1 * 1 + 1 * 5 = 6$) mit Stückanzahl $\mathrm{Anz}((1,1)) = 1 + 1 = 2$.

Definition 3: Sei $B \subset \mathbb{A}(n)$, dann heißt B eine <u>Basis</u> von $\mathbb{A}(n)$, wenn gilt: $\forall x \in \mathbb{A}(n)\ \exists$ (mindestens) eine Stückelung von x bezüglich B.

Beispiel: $\{1,5\}$ ist eine Basis von $\mathbb{A}(9)$ - jede natürliche Zahl zwischen 1 und 9 lässt sich als Summe von Elementen aus $\{1,5\}$ darstellen, oft sogar auf mehrere verschiedene Weisen - so ist etwa $6 = 1 + 1 + 1 + 1 + 1 + 1$ und $6 = 5 + 1$.

Lemma 1: $\forall n \in \mathbb{N}$ und für alle Teilmengen B von $\mathbb{A}(n)$ gilt: $1 \in B \Longleftrightarrow B$ Basis von $\mathbb{A}(n)$
Beweis: Jede natürliche Zahl x lässt sich als Summe $\sum_{i=1}^{x} 1$ darstellen.1 lässt sich nur auf genau eine Weise als Summe natürlicher Zahlen darstellen ($1 = \sum_{i=1}^{1} 1$).

Lemma 2: $\forall\, n \in \mathbb{N}$ gilt: {1} ist eine Basis von $\mathbb{A}(n)$ (wir nennen sie auch die minimale Basis)
Beweis: folgt aus Lemma 1

Lemma 3: $\forall\, n \in \mathbb{N}$ gilt: $\mathbb{A}(n)$ ist eine Basis von $\mathbb{A}(n)$ (wir nennen sie auch die maximale Basis)
Beweis: folgt aus Lemma 1.

Definition 4: Sei B eine Basis von $\mathbb{A}(n)$ und $x \in \mathbb{A}(n)$. Dann definieren wir $\text{Min}_B(x) := \underset{S \in \mathbb{S}_B(x)}{\text{Minimum}}(\text{Anz}(S))$.

Bemerkung: Dies ist die minimale Anzahl Stücke zur Darstellung von x mit der Basis B. Wegen B Basis existiert für alle $x \in \mathbb{A}(n)$ mindestens eine Stückelung. Es kann nur endlich viele Stückelungen geben, also existiert immer auch das Minimum.

Beispiel: Mit B:= {1, 5} Basis von $\mathbb{A}(9)$ gilt $\text{Min}_B(1) = 1$, $\text{Min}_B(2) = 2$, $\text{Min}_B(3) = 3$, $\text{Min}_B(4) = 4$, $\text{Min}_B(5) = 1$, $\text{Min}_B(6) = 2$, $\text{Min}_B(7) = 3$, $\text{Min}_B(8) = 4$, $\text{Min}_B(9) = 5$.

Definition 5: Sei B eine Basis von $\mathbb{A}(n)$. Dann definieren wir $\text{AvgMin}(B) := (\sum_{x=1}^{n} \text{Min}_B(x)) / n$.

Bemerkung: Dies ist die durchschnittliche Anzahl Stücke zur minimalen Darstellung aller Elemente von $\mathbb{A}(n)$.

Beispiel: Wie man leicht für die Basis B:= {1, 5} von $\mathbb{A}(9)$ errechnen kann, ist AvgMin(B) = 2,77.

Satz 1: Für die minimale Basis von $\mathbb{A}(n)$ gilt AvgMin({1}) = (n + 1) / 2 und für die maximale Basis gilt $\text{AvgMin}(\mathbb{A}(n)) = 1\ \forall\, n \in \mathbb{N}$.
Beweis: Wie man leicht sieht gilt für die minimale Basis $\text{Min}_{\{1\}}(x) = x$ und damit AvgMin({1}) = (n + 1) / 2. Für die maximale Basis gilt $\text{Min}_{\mathbb{A}(n)}(x) = 1\ \forall\, x \in \mathbb{A}(n)$ und damit $\text{AvgMin}(\mathbb{A}(n)) = 1$.

Bemerkung: Mit Hilfe von Definition 5 kann man nun Basen hinsichtlich ihrer Optimalität (im Mittel möglichst wenig Stücke) vergleichen. Allerdings sieht man sofort, dass man sich dabei auf Basen der gleichen Mächtigkeit beschränken muss, denn es gibt natürlich immer eine größere Basis mit kleinerem Wert von AvgMin - z.B. die maximale Basis. Es gilt sogar der folgende Satz.

Satz 2: Sei B eine Basis von $\mathbb{A}(n)$ der Mächtigkeit m < n. Dann gibt es eine Basis B' der Mächtigkeit m + 1 mit AvgMin(B') < AvgMin(B)
Beweis: Sei $x \in \mathbb{A}(n) \setminus B$. Dann ist $B' := B \cup \{x\}$ eine Basis von $\mathbb{A}(n)$ in der sich x durch sich selbst darstellen lässt und damit minimaler als in B (wo man mindestens 2 Stücke braucht), während sich alle anderen Elemente von $\mathbb{A}(n)$ mindestens ebenso minimal darstellen lassen.

Bemerkung: Jetzt lässt sich auch das 3-Peso-Problem entscheiden. Da mir nur begrenzte Rechnerkapazitäten zur Verfügung standen, habe ich beispielhaft nur den Geldraum $\mathbb{A}(99)$ betrachtet - als Einheit kann man sich dann den Euro bzw. Peso vorstellen. Mit einem Computerprogramm wurden auf dem Geldraum $\mathbb{A}(99)$ die (deutsche) Basis D := {1, 2, 5, 10, 20, 50} mit der (kubanischen) Basis K := {1, 3, 5, 10, 20, 50} verglichen. Für beide kam interessanterweise derselbe Wert AvgMin(D) = AvgMin(K) = 3,43 heraus (was man vielleicht auch erwarten konnte, wenn man sich nur die Darstellung der ersten 10 Zahlen in den beiden Basen anguckt).

Auch für die modifizierte kubanische Basis K' := {1, 3, 5, 10, 30, 50} ergibt sich derselbe Wert. Doch gibt es vielleicht noch bessere Basen derselben Mächtigkeit?

Definition 6: Sei $\mathbb{B}(m)$ die Menge aller Basen von $\mathbb{A}(n)$ der Mächtigkeit m <= n. Sei $B' \in \mathbb{B}(m)$. Dann heißt B' m-optimal in $\mathbb{A}(n)$, wenn $AvgMin(B') \leqslant AvgMin(B)\ \forall\ B \in \mathbb{B}(m)$.

Bemerkung: Für $\mathbb{B}(6)$ ergeben sich (per Programm) die beiden Basen {1,4,6,21,30,37} und {1,5,8,20,31,33} als 6-optimal in $\mathbb{A}(99)$ mit AvgMin = 2,94 und sind damit deutlich besser als die deutsche oder kubanische Basis. Auch die Basis {1,3,4,10,30,40} ist mit einem AvgMin = 3,23 etwas besser (und ist trotzdem ziemlich 'dezimal'). Interessanterweise gibt es sogar eine Basis der Mächtigkeit 5 (also kleiner 6!), die optimaler ist als die deutsche bzw. kubanische Basis: {1,5,16,23,33} mit AvgMin = 3,32.

Bemerkung: Unbefriedigend an den bisherigen Betrachtungen ist die Tatsache, das wir kein Optimalitätskriterium für die zu wählende Mächtigkeit der Basis haben. Natürlich ist die maximale Basis immer besser (mit AvgMin = 1) als jede Basis mit kleinerer Mächtigkeit - aber gleichzeitig dürfte diese Basis (ein Geldstück für jeden Betrag) für den täglichen Umgang mit Geld sehr unpraktisch sein. Wir brauchen daher noch ein weiteres Kriterium.

Definition 7: Ein Zahlentupel $E = (e_1,e_2,...,e_m) \in \mathbb{N}^m$ mit $e_1 \leqslant e_2 \leqslant ... \leqslant e_m \leqslant n$ heißt ein Erzeuger von $\mathbb{A}(n)$ ($m,n \in \mathbb{N}$), wenn gilt: $\forall\ x \in \mathbb{A}(n)\ \exists\ \delta_i \in \{0,1\}\ (1 \leqslant i \leqslant m)$ mit $x = \sum_{i=1}^{m} (\delta_i * e_i)$.
Die Zahl $Dim(E) := m \in \mathbb{N}$ heißt dann die Dimension des Erzeugers.

Bemerkung: Es darf in dem Zahlentupel eines Erzeugers eine Zahl auch mehrfach auftreten. So sind sowohl (1,2,4,8) als auch (1,1,3,6) Erzeuger von $\mathbb{A}(9)$ (neben etlichen anderen Erzeugern). Anschaulich: Wenn man nur Eurostücke mit den Werten (1,2,4,8) oder (1,1,3,6) in der Geldbörse hat, so läßt sich damit jeder Geldbetrag zwischen 1 und 9 Euro bezahlen.

Bemerkung: Die Reihenfolge der Zahlen im Tupel spielt eigentlich keine Rolle (jede mögliche Kombination wäre wieder ein Erzeuger). Wir nehmen aber im folgenden o.B.d.A. immer $e_1 \leqslant e_2 \leqslant ... \leqslant e_m$ an. Triviale Erzeuger in $\mathbb{A}(n)$ sind die n-dimensionalen Tupel (1,1,1,...,1) und (1,2,...,n).

Lemma 4: Sei $E = (e_1,e_2,...,e_m)$ ein Erzeuger von $\mathbb{A}(n)$. Dann ist $(e_1,e_2,...,e_k)$ ein Erzeuger von $\mathbb{A}(e_{k+1}-1)$ für $1 \leqslant k < m$.
Beweis folgt aus der Definition.

Definition 8: Es sei $\mathbb{E}(n)$ die Menge aller Erzeuger von $\mathbb{A}(n)$. Sei $E' \in \mathbb{E}(n)$. Dann heißt E' minimaler Erzeuger von $\mathbb{A}(n)$, wenn $Dim(E') \leqslant Dim(E)\ \forall\ E \in \mathbb{E}(n)$. Dim(E') heißt dann Dimension von $\mathbb{A}(n)$, geschrieben $Dim(\mathbb{A}(n))$.

Bemerkung: Wie man leicht nachrechnen kann sind (1,2,4,8), (1,2,4,7), (1,2,4,6), (1,2,4,5), (1,2,4,4), (1,2,3,7), (1,2,3,6), (1,2,3,5), (1,2,3,4), (1,2,3,3), (1,2,2,6), (1,2,2,5), (1,2,2,4), (1,1,3,6), (1,1,3,5), (1,1,3,4), (1,1,2,5) minimale Erzeuger von $\mathbb{A}(9)$. $\mathbb{A}(9)$ hat also die Dimension 4. Anschaulich gesprochen braucht man mindestens 4 Geldstücke um damit alle möglichen Geldwerte im Geldraum $\mathbb{A}(9)$ darzustellen.

Lemma 5: Sei $m < n$ ($m,n \in \mathbb{N}$). Dann ist $\mathrm{Dim}(\mathbb{A}(m)) \leqslant \mathrm{Dim}(\mathbb{A}(n))$.
Beweis: Sei $E = (e_1,e_2,...,e_k)$ mit $k \in \mathbb{N}$ ein minimaler Erzeuger von $\mathbb{A}(n)$. Sei j der größte Index für den $e_j \leqslant m$. Da nach Lemma 4 $E' := (e_1,e_2,...,e_j)$ ein Erzeuger von $\mathbb{A}(m)$ ist, gilt $\mathrm{Dim}(\mathbb{A}(m)) \leqslant \mathrm{Dim}(E') \leqslant \mathrm{Dim}(E) = \mathrm{Dim}(\mathbb{A}(n))$.

Bemerkung: Trivialerweise ist (1) minimaler Erzeuger von $\mathbb{A}(2^1\text{-}1)=\{1\}$. Wie man leicht sieht ist $(2^0,2^1) = (1,2)$ ein Erzeuger in $\mathbb{A}(2^2\text{-}1) = \mathbb{A}(3)$ (es ist nämlich 1+2=3). (1,2) ist ein minimaler Erzeuger, denn alle möglichen 1-er Tupel ((1),(2),(3)) sind keine Erzeuger von $\mathbb{A}(3)$. (1,2) ist sogar der einzige minimale Erzeuger (mit Dimension 2), denn alle anderen möglichen 2-er Tupel - (1,1), (1,3), (2,2), (2,3), (3,3) - sind ebenfalls keine Erzeuger. Dies führt zum folgenden Satz.

Satz 3: Für alle $n \in \mathbb{N}$ ist $E^n := (2^0,2^1,2^2,...,2^{n-1})$ einziger minimaler Erzeuger von $\mathbb{A}(2^n\text{-}1)$, das damit die Dimension n hat.
Beweis: Eine n-elementige Menge besitzt genau 2^n Teilmengen (darunter auch die leere Menge). Man kann also aus den n Komponenten eines n-dimensionalen Erzeugers durch Summenbildung (in der jede Komponente höchstens einmal auftreten darf) maximal 2^n-1 verschiedene Zahlen erzeugen. Es kann also keinen Erzeuger von $\mathbb{A}(2^n\text{-}1)$ geben, der weniger als n Komponenten hat. Wie man leicht sieht (Binärdarstellung der natürlichen Zahlen!) ist E^n ein Erzeuger von $\mathbb{A}(2^n\text{-}1)$ und damit sogar ein minimaler Erzeuger.
E^n ist sogar der einzige minimale Erzeuger von $\mathbb{A}(2^n\text{-}1)$. Angenommen es gäbe einen weiteren Erzeuger $E' := (e_1,e_2,...,e_n) \neq E^n$ der Dimension n. Sei $i > 1$ der kleinste index, für den gilt $e_i \neq 2^{i-1}$. Falls $e_i > 2^{i-1}$, dann läßt sich 2^{i-1} nicht durch $(e_1,e_2,...,e_{i-1})$ erzeugen. Falls $e_i < 2^{i-1}$, dann läßt sich die Zahl e_i sowohl durch e_i als auch durch $(e_1,e_2,...,e_{i-1})$ erzeugen und es gibt nicht mehr genug Möglichkeiten um mit E' alle Zahlen von $\mathbb{A}(2^n\text{-}1)$ zu erzeugen. Also ist E^n der einzige minimale Erzeuger von $\mathbb{A}(2^n\text{-}1)$.

Bemerkung: Wir sehen also, dass sich die Dimensionalität eines wachsenden Anfangsteils der natürlichen Zahlen immer bei der nächst höheren Zweierpotenz um eins erhöht. Es folgt direkt der nächste Satz.

Satz 4: Für alle $m, n \in \mathbb{N}$ mit $2^{n-1} \leqslant m < 2^n$ gilt $\mathrm{Dim}(\mathbb{A}(m))=n$ und $E^n := (2^0,2^1,2^2,...,2^{n-1})$ ist minimaler Erzeuger von $\mathbb{A}(m)$. Für $m=2^n\text{-}1$ ist E^n sogar der einzige minimale Erzeuger.

Bemerkung: Für den bereits oben betrachteten Geldraum $\mathbb{A}(99)$ ergibt sich wegen $2^6=64 \leqslant 99 < 2^7=128$ die Dimension 7. Dies würde für eine 7-teilige Stückelung sprechen, zumal sich in der Menge der minimalen Erzeuger des $\mathbb{A}(99)$ auch kein Tupel mit mehrfach wiederholten Werten findet. Man könnte mit 7 Stücken sogar den $\mathbb{A}(127)$ abdecken. Hierfür gibt es dann allerdings nur einen einzigen minimalen Erzeuger (1,2,4,8,16,32,64), der im $\mathbb{A}(99)$ den guten Wert von AvgMin = 3,19 hat. Für den Mathematiker wäre dies sicher die eleganteste Lösung, doch würde eine solche Stückelung vermutlich wegen der großen Abweichungen vom Dezimalsystem auf Widerstände stoßen.

Bemerkung: Per Programm wurden für den $\mathbb{A}(99)$ insgesamt 1154 minimale Erzeuger ermittelt und für jeden Erzeuger auch der AvgMin. Eine kleine Auswahl davon befindet sich im Anhang 1. Als minimale Erzeuger mit dem kleinsten Wert für AvgMin (= 2,83) erhalten wir die zwei Basen {1, 2, 4, 8, 13, 29, 48} und {1, 2, 4, 7, 15, 27, 45}, die leider sehr weit vom dezimalen System der konventionellen Stückelung {1, 2, 5, 10, 20, 50} abweichen. Es wurde deshalb unter den minimalen Erzeugern nach den Basen mit maximaler Ähnlichkeit zur konventionellen Basis gesucht: insgesamt 12 minimale Erzeuger stimmen mit 4 Stücken der konventionellen Stückelung überein. Unter diesen wurden schließlich die beiden Basen mit dem geringsten AvgMin ausgewählt:

B1 := {1, 2, 4, 7, 15, 20, 50} mit AvgMin = 3 und B2 := {1, 2, 4, 8, 15, 20, 50} mit AvgMin = 2,96.
Als zusätzliches Kriterium könnte man jetzt noch die Gesamtsumme der Basis minimieren - man möchte auch wertmäßig möglichst wenig Geld mit sich herumtragen. Hier schneidet B1 mit der kleinstmöglichen Summe von 99 geringfügig besser als B2 mit 100 ab. Dafür hat allerdings B2 einen geringfügig besseren AvgMin als B1.

Konklusion: B1 und B2 wäre beide sicher sehr geeignete Kandidaten für eine deutlich bessere Währungsstückelung. Beide Basen haben einen um mehr als 10% geringeren AvgMin als die konventionelle Stückelung (= 3,43). Mit beiden Basen braucht man nur 7 Stücke, um jedes Geldgeschäft im $\mathbb{A}(99)$ tätigen zu können, während mit der konventionellen Basis mindestens 8 Stücke notwendig sind - wie man sich leicht selbst überlegen kann, muss man konventionell mindestens zwei 1- oder 2-Euro-Stücke und auch zwei 10- oder 20-Euro-Stücke dabei haben. Und schließlich hält sich die Abweichung von der konventionellen Stückelung in Grenzen. Für die vielleicht vermissten 5- und 10-Euro-Scheine gibt es jetzt den 15-Euro-Schein, und mit den neuen 4- und 7-Euro- (bzw. 8-Euro-) Scheinen viel Flexibilität. Natürlich würde man den Cent-Bereich dann ganz analog stückeln.

Thomas Blunck / 28.11.2019

Anhang 1 - Der AvgMin für ausgewählte Basen im $\mathbb{A}$(99) und $\mathbb{A}$(127)

AvgMin für Basen der Mächtigkeit 6 im $\mathbb{A}$(99):

1 2 5 10 20 50 ---> 3,43
1 3 5 10 20 50 ---> 3,43
1 3 5 10 30 50 ---> 3,43
1 3 4 10 30 40 ---> 3.23
1 2 7 19 29 33 ---> 3
1 3 9 16 35 40 ---> 2,95
1 4 6 21 30 37 ---> 2,94 !!! Optimum
1 5 8 20 31 33 ---> 2,94 !!! Optimum
1 6 9 16 29 40 ---> 2,95
1 7 10 21 26 44 ---> 2,97
1 8 10 13 36 40 ---> 3,03
1 2 4 8 16 32 ---> 3.55

AvgMin für Basen der Mächtigkeit 5 im $\mathbb{A}$(99):

1 2 7 24 33 ---> 3,44
1 3 11 27 34 ---> 3,34
1 5 16 23 33 ---> 3,32 !!! Optimum

AvgMin für ausgewählte Basen, die zugleich minimale Erzeuger sind im $\mathbb{A}$(99):

1 2 4 8 13 29 48 ---> 2,83 !!! Optimum
1 2 4 7 15 27 45 ---> 2,83 !!! Optimum
1 2 4 8 15 20 50 ---> 2,96 (maximale Übereinstimmung mit konventioneller Stückelung)
1 2 4 7 15 20 50 ---> 3 (maximale Übereinstimmung mit konventioneller Stückelung)
1 2 4 8 16 32 64 ---> 3.19 (die vom Autor bevorzugte Stückelung)

AvgMin für Basen der Mächtigkeit 7 im $\mathbb{A}$(127):

1 2 5 10 20 50 100 ---> 3.39
1 2 4 8 16 32 64 ---> 3.52

Anhang 2 - Quellcode ‘FixBase’ zur Berechnung von AvgMin

```
Option Compare Database
Option Base 1
Option Explicit
Const sMax = 127
Const bMax = 7
Const MAXINT As Integer = (2 ^ 15) - 1
Const strPath = “C:\Users\tb\Desktop\Test\TESTFILE.TXT”
Dim base(bMax) As Integer
Dim stringArray() As String
Dim min As Integer
Dim timer As Date

Private Sub findFixbase()  ‘ Suche nach optimaler Basis
    timer = Time
    newFile
    base(1) = 1
    min = 340 ‘ Anfangsschätzung
    indexLoop 2, 2
    Print #1, Time
    Close #1
End Sub

Private Sub calcFixBase() ‘ Ermittlung Durchschnitt Stückanzahl für gegebene Basis
    base(1) = 1
    base(2) = 2
    base(3) = 4
    base(4) = 8
    base(5) = 16
    base(6) = 32
    base(7) = 64
    min = MAXINT
    calc (True)
End Sub

Private Sub indexLoop(index As Integer, start As Integer)
    Dim i As Integer
    Dim j As Integer
    Dim line As String
    Dim sum As Integer
    For i = start To sMax - bMax + index Step 1
        base(index) = i
        If index = bMax Then
            sum = calc(False)
            If sum <= min Then
                If sum < min Then
                    min = sum
                    newFile
                    Print #1, “Summe: “ & Str(min) & “ Average: “ & Str(min / sMax)
                End If
                line = “”
                For j = 1 To bMax Step 1
                   line = line & Str(base(j)) & “ “
                Next j
                Print #1, line
            End If
        Else
            indexLoop index + 1, i + 1
        End If
    Next i
End Sub

Private Function calc(individual As Boolean) As Integer
    Dim space(sMax) As Integer
```

```
Dim stepCounter As Integer
Dim spaceCounter As Integer
Dim s As Integer
Dim b As Integer
Dim zw1, zw2, zw3 As Integer
Dim sum As Integer
Dim line As String
For s = 1 To sMax Step 1
   space(s) = 0
Next s
For b = 1 To bMax Step 1
   space(base(b)) = 1
Next b
sum = bMax
spaceCounter = bMax
stepCounter = 1
Do While spaceCounter < sMax
   If Not individual Then
   ‘ **********************************************
   ‘ grobe Abschätzung über die zu erwartende Summe
   ‘ kann zum vorzeitigen Abbruch führen
      zw1 = spaceCounter
      zw2 = sum
      zw3 = stepCounter + 1
      Do While zw1 * bMax < sMax - zw1
         zw2 = zw2 + zw1 * bMax * zw3
         zw1 = zw1 + zw1 * bMax
         zw3 = zw3 + 1
      Loop
      zw2 = zw2 + (sMax - zw1) * zw3
      If (zw2 > min) Then
         sum = MAXINT
         Exit Do
      End If
   ‘ **********************************************
   End If
   For s = 1 To sMax Step 1
      If space(s) = stepCounter Then
         For b = 1 To bMax Step 1
            zw1 = s + base(b)
            If zw1 > sMax Then
               Exit For
            End If
            If space(zw1) = 0 Then
               space(zw1) = stepCounter + 1
               sum = sum + stepCounter + 1
               spaceCounter = spaceCounter + 1
            End If
         Next b
      End If
   Next s
   stepCounter = stepCounter + 1
Loop
calc = sum
If individual Then
   newFile
   line = “”
   For b = 1 To bMax Step 1
      line = line & Str(base(b)) & “ “
   Next b
   line = line & “ - Summe: “ & Str(sum) & “ - Average: “ & Str(sum / sMax)
   Print #1, line
   For s = 1 To sMax Step 1
      line = Str(s) & “: “ & Str(space(s))
      Print #1, line
   Next s
   Close #1
End If
```

```
End Function

Private Sub newFile()
   Close #1
   Open strPath For Output As #1
   Print #1, timer
   Print #1, “ sMax = “ & Str(sMax) & “; bMax = “ & Str(bMax)
   Close #1
   Open strPath For Append As #1
End Sub

Private Sub BigLoop()
Dim i As Integer
Dim testString As String
Open “C:\Users\tb\Desktop\Test\TESTFILE0.TXT” For Input As #2
newFile
Do While Not EOF(2)
   testString = “”
   Line Input #2, testString
   stringArray() = Split(testString, “;”, 7)
   For i = 0 To bMax - 1 Step 1
      min = MAXINT
      base(i + 1) = val(stringArray(i))
   Next i
   Print #1, testString & “;” & (calc(False) / sMax)
Loop
Close #1
Close #2
End Sub
```

Anhang 3 - Quellcode ‘FlexBase’ zur Berechnung minimaler Erzeuger

```
Option Compare Database
Option Explicit
Option Base 1
Const strStart = "C:\Users\tb\Desktop\Test\TEST"
Dim sMax As Integer
Dim strPath As String
Dim space(255) As Integer
Dim min As Integer
Dim timer As Date

Private Sub findFlexBase()
    Dim i As Integer
    timer = Time
    For sMax = 2 To 15 Step 1
        min = sMax  ' min muss <= sMax sein !
        strPath = strStart & Str(sMax) & ".TXT"
        newFile
        indexLoop 1, 0, 0
    Next sMax
    Write #1, Time
    Close #1
End Sub

Private Sub checkFlexBase()
    Dim i As Integer
    sMax = 14
    For i = 1 To sMax Step 1
        space(i) = 0
    Next i
    space(1) = 1
    space(2) = 1
    space(4) = 1
    space(6) = 1
    Debug.Print check(space)
End Sub

Private Sub indexLoop(index As Integer, sum As Integer, total As Integer)
    Dim i As Integer
    Dim j As Integer
    Dim s As Integer
    Dim zwsum As Integer
    Dim line As String
    If index = 1 Then
        i = 1
    Else
        i = 0
    End If
    For i = i To min Step 1
        zwsum = sum + i
        If index * i > sMax Or zwsum > min Then
            Exit For
        End If
        space(index) = i
        If index = sMax Then
            If check(space()) Then
                If zwsum < min Then
                    min = zwsum
                    newFile
                End If
                line = ""
                For j = 1 To sMax Step 1
                    If (space(j) > 0) Then
                        line = line & Str(j) & ":" & Str(space(j)) & " "
                    End If
```

```
            Next j
            Print #1, line
         End If
      Else
         If i >= 1 Then
            indexLoop index + 1, zwsum, total + i * index
         Else
            If total >= index Then
               indexLoop index + 1, zwsum, total
            End If
         End If
      End If
   Next i
End Sub

Private Function check(ByRef space() As Integer) As Boolean
   Dim i As Integer
   Dim j As Integer
   Dim s As Integer
   Dim val As Integer
   Dim test(255) As Boolean
   For s = 1 To sMax Step 1
      test(s) = False
   Next s
   For s = sMax To 1 Step -1
      val = space(s)
      If val > 0 Then
         For j = sMax To 1 Step -1
            If test(j) Then
               For i = 1 To val Step 1
                  If (j + i * s) <= sMax Then
                     test(j + i * s) = True
                  End If
               Next i
            End If
         Next j
         For i = 1 To val Step 1
            If (i * s) <= sMax Then
               test(i * s) = True
            End If
         Next i
      End If
   Next s
   check = True
   For s = 1 To sMax Step 1
      If Not test(s) Then
         check = False
         Exit For
      End If
   Next s
End Function

Private Function printArray(ByRef arr() As Boolean)
   Dim i As Integer
   For i = LBound(arr) To UBound(arr) Step 1
      Debug.Print i, arr(i)
   Next i
End Function

Private Sub newFile()
   Close #1
   Open strPath For Output As #1
   Print #1, timer
   Print #1, "Minimum: " & Str(min)
   Close #1
   Open strPath For Append As #1
End Sub
```