universität wien

Diplomarbeit

Titel der Diplomarbeit

„Diskrete Portfolioselektion mit Metaheuristik“

Verfasser

Bakk. phil. Chang Yuan LI

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissentschaften

(Mag. rer. soc. oec.)

Wien, im 16. Mai 2007

Studienkennzahl (lt. Studienblatt):	A 157 000
Studienrichtung lt. Studienblatt:	Internationale Betriebswirtschaft
Betreuerin / Betreuer:	Univ.-Prof. Dr. Christian KEBER

Danksagung

Ich möchte mich zunächst bei meinen Eltern und meinem Bruder bedanken, ohne ihre Unterstützung wäre mein Studium nicht möglich gewesen wäre. Professor Keber möchte ich mich für die ausgezeichnete Betreuung der Diplomarbeit bedanken, und dass er in seiner Lehrveranstaltung Computational Finance durch seine detaillierten und doch leicht verständlichen Ausführungen und Beispielen über genetischen Algorithmen mich persönlich zur Implementierung von genetischen Algorithmen sehr begeistert hat. Hu Bin möchte ich auch Dank aussprechen, da er mich auf die EALib aufmerksam gemacht hat und mich bei dem Vorhaben, die EALib zu verwenden, unterstützt hat. Zum Schluss möchte ich noch Professor Raidl vom Institut für Computergraphik und Algorithmen auf der TU-Wien für das Bereitstellen der EALib danken, und für eine schnelle unbürokratische Einrichtung eines Benutzerkontos für Zugriffe auf Rechner des Institutes.

Abstract

This master thesis deals with close-to-reality portfolio selection problems like budget constraint, cardinality of securities and limited number of securities.
Because generally integer programming has to revert to exact algorithms to obtain optimality, for large instances, it is therefore computationally too intensive. The present thesis tries to tackle the discrete portfolio optimization problem by using three independent meta-heuristics: genetic algorithm, local search and variable neighborhood search. The results show, that by choosing a big number of securities, the risk-diversification-effect does only increase marginally compared with choosing the right securities of a small number of companies.

Kurzfassung

Diese Diplomarbeit möchte grundsätzlich realitätsnahe Portefeuilleselektionsprobleme wie Budgetbeschränkung, Ganzzahligkeit von Wertpapieren und beschränkte Anzahl von Wertpapieren behandeln, und Lösungsansätze aufzeigen. Da bei ganzzahliger Optimierung im Allgemeinen auf exakte Lösungsverfahren zurückgegriffen werden muss, um das globale Optimum zu bestimmen, ist es bei großen Instanzen zu rechenintensiv.
Die vorliegende Arbeit versucht unabhängig mit drei verschiedenen Metaheuristiken (Genetischer Algorithmus, Lokale Suche und Variable Neighborhood Search) das diskrete Portefeuille-Optimierungsproblem zu lösen. Die Ergebnisse dieser Arbeit zeigen, dass man bei der richtigen Wahl einer kleinen Anzahl von riskanten Finanzierungstiteln der Risikodiversifizierungseffekt nur marginal schlechter als die Wahl von einer großen Anzahl von riskanten Finanzierungstiteln ist.

Inhaltsverzeichnis

Liste der Variablen und Parameter

α	Anteil eines Portefeuilles, das risikolos ist
$\delta(S)$	Definierende Länge eines Schemas S
$\eta(S)$	Anzahl der *don't care* Symbole eines Schemas S
θ	Steigung der Tobineffizienzlinie
$\xi(S,t)$	Anzahl der Chromosomen eines Schemas S zum Zeitpunkt t
$\sigma(i)$	Standardabweichung von i
$\sigma^2(i)$	Varianz von i
ϕ	Risikomaß
b_i	i-te Hilfsvariable zur Formulierung von k aus n riskanten Wertpapieren
$Cov(i,j)$	Kovarianz zwischen i und j
$E(.)$	Erwartungswert einer Zufallsvariable
$eval(v)$	Bewertungsfunktion zur Fitnessbestimmung
$eval_a(v)$	Adjustierte Bewertungsfunktion zur Fitnessbestimmung
$F(t)$	Totale Fitness zum Zeitpunkt t
$\overline{F(t)}$	Durchschnittliche Fitness zum Zeitpunkt t
k	Anzahl gewählter Wertpapiere aus einem Portefeuille
m	Länge eines binären Vektors
MVP	Minimum Varianz Portfolio
$\mathbb{N}_0^+$	Positive natürliche Zahlen einschließlich Null
$N(x)$	Nachbarschaft von x
n	Anzahl riskanter Wertpapiere
$o(S)$	Ordnung des Schemas S
P	Optimierungsproblem
$P(t)$	Population zum Zeitpunkt t
P_{0i}	Preis des i-ten riskanten Wertpapiers zum Zeitpunkt $t=0$

p_c	Crossoverwahrscheinlichkeit
p_i	Proportionale Fitness des i-ten Chromosoms
p_m	Mutationswahrscheinlichkeit
$PopSize$	Konstante Populationsgröße
q_i	Ganze Stückzahl des i-ten riskanten Wertpapiers eines Portefeuilles
q_i^{Sel}	Selektionswahrscheinlichkeit des i-ten Chromosoms
r_f	Risikoloser Zinssatz
r_j	Rendite des j-ten Finanzierungstitels in % p.a.
S	Suchraum (Lösungsraum) eines Problems P
S	Schema S
S_G	Suchraum (Lösungsraum) des Genetischen Algorithmus
$U(.)$	Nutzenfunktion
V_0	Zur Verfügung stehendes Budget Zeitpunkt $t=0$
v	Chromosom (Individuum) einer Population
x_j	Wertmäßiger Anteil des j-ten Finanzierungstitels an einem Portefeuille

Abkürzungsverzeichnis

bzw.	beziehungsweise
E-V	Erwartungswert und Varianz
EALib	Evolutionary Algorithms Library
GA	Genetischer Algorithmus
i. d. R.	in der Regel
LS	Lokale Suche
MA	Memetischer Algorithmus
Nasdaq100	National Association of Securities Dealers Automated Quotation 100 (Aktienindex)
S.	Seite
SA	Simulated Annealing
s.t.	Subject to (unter den Nebenbedingungen)
TS	Tabu Search
vgl.	Vergleiche
VND	Variable Neighborhood Descent
VNS	Variable Neighborhood Search
z. B.	zum Beispiel

Kapitel 1

Motivation und Einführung

Eine wesentliche Erkenntnis der Modernen Portefeuilletheorie ist die optimale Risikodiversifikation[1] von Finanzierungstiteln in einem Portefeuille mittels geeigneter Gewichtung der Anteile und einer großen Anzahl von unterschiedlichen Finanzierungstiteln, das in Grunde genommen auf der bahnbrechende Arbeit „Portfolio Selection“ [45] von Markowitz zurückzuführen ist. Empirische Untersuchungen aus der Praxis haben aber gezeigt, dass Portefeuilles häufig nur eine geringe Anzahl von Finanzierungstiteln enthalten, wie etwa von Blume und Friend [5], Guiso et al. [24], Bertraut und Starr-McCluer [3] oder Börsch-Supan und Eymann [6] – was zwar auf dem ersten Blick nicht ganz in das Bild von Risikodiversifizierung mit größtmöglicher Anzahl von Finanzierungstiteln passt, aber mit dem Argument von Transaktionskosten bei jedem Kauf von Finanzierungstitel plausibel erklärt werden kann. Ein weiteres Argument liefert Solnik in seiner Arbeit [59], welche zeigt, dass bei internationaler Diversifikation bereits mit vergleichweise wenigen Titeln eine hinreichend gute Risikoreduktion erzielt werden kann.[2] Somit kommt zur zentralen Fragestellung bei Investitionsendscheidungen für einen Investor neben der Gewichtung der Titeln auch noch hinzu, ob ein Finanzierungstitel überhaupt in das Portefeuille aufgenommen werden soll.

Burtscher hat in seiner Diplomarbeit [7] dargelegt, dass man selbst ohne jegliche Stückzahlentscheidungen, also nur reine Ja-Nein-Kaufentscheidungen bei k aus n

[1] in Bezug auf ein Optimierungsmodell

[2] Vgl. Maringer [43, S. 1222]

Finanzierungstiteln zu folgenden alternativen Möglichkeiten für das Portfolio kommt[3]:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Das bedeutet, dass bei einer Börse mit 100 gehandelten Aktien, wobei der Investor sein Portefeuille nur mit 10 verschiedenen Aktien zusammenstellen möchte, sich bereits $1{,}73 \cdot 10^{13}$ mögliche Kombinationen ergeben.

Ein häufiger Kritikpunkt der Portfolioselektion nach Markowitz ist, dass es sich um eine nicht lineare Optimierung handelt. Die Berechnung ist bei einer großen Anzahl von Wertpapieren sehr schwierig beizukommen. Eine Möglichkeit wäre, die Problemstellung in ein lineares Programm umzuformulieren.[4] Markowitz führte den sogenannten *Critical Line* Algorithmus [44] ein, von dem lange Zeit angenommen wurde, dass er nur unter bestimmten Voraussetzungen funktionieren kann. Lewis [40] zeigte jedoch, dass derselbe Algorithmus auf viel allgemeinere quadratische Optimierungsprobleme anwendbar ist.[5] Eine andere Möglichkeit wäre, diese Menge an Wertpapieren mit Hilfe von sogennannte Wertpapier-Klassen[6] zu unterteilen. Mit dieser Strategie ließe sich zwar sehr schnell und einfach die Anzahl der Finanzierungstiteln reduzieren, doch kann es passieren, dass die Ergebnisse schon allein in Bezug auf die Risikodiversifikation suboptimal wären.[7]

Hebt man die Annahme der beliebigen Teilbarkeit der Finanzierungtiteln auf, um einen stärkeren Praxisbezug zu erlangen, so erhält man ein ganzzahliges quadratisches Optimierungsmodell. Da bei ganzzahliger Optimierung im Allgemeinen auf exakte Lösungsverfahren zurückgegriffen werden muss, um das globale Optimum zu bestimmen, ist es bei großen Instanzen meist „sehr“ rechenintensiv. Im schlimmsten Fall muß man bei ganzzahliger Optimierung mit vollständiger Enumeration bei Vollständigkeit des Problems das globale Optimum bestimmen. In unserem Fall bedeutet das, dass mit zunehmender Anzahl von Finanzierungtiteln die Suchzeit exponentiell steigen würde.[8]

Ein weiterer Punkt, was auch einen Praxisbezug besitzt, ist die Berücksichtigung des zur

---

[3] Vgl. auch Maringer [43, S. 1222]

[4] Siehe Konno und Yamazaki [37]

[5] Vgl. Keber [32, S. 1027]

[6] Siehe Farrell [14] Part 5: Asset Class Management

[7] Vgl. Maringer [43, S. 1222]

[8] Vgl. Keber [32, S.1027]

Verfügung stehenden Budget eines Investors. Das Budget sollte zwar vollständig ausgenutzt werden, dennoch sollte es nicht überschritten werden.

Diese folgende Arbeit möchte die vorangegangenen Probleme als ein diskretes Portefeuilleoptimierungsproblem mit Budgetrestriktion formulieren und mit Hilfe von metaheuristischen Lösungsverfahren das formulierte Optimierungsproblem lösen. Die in dieser Arbeit eingesetzten metaheuristischen Lösungsverfahren sind Genetische Algorithmen, Lokale Suche und Variable Neighborhood Search.

Bevor wir uns der Formulierung des diskreten Portefeuilleoptimierungsproblems widmen, möchten wir in zwei Kapiteln die Grundlagen der Modernen Portefeuilletheorie (Kapitel 2) und die Arbeitsweise der verwendeten metaheuristischen Algorithmen (Kapitel 3) kurz vorstellen. Kapitel 4 vereinigt das Wissen der vorhergehenden Kapitel, und das diskrete Portfolioselektionsproblem wird als Maximierungsproblem ausgestaltet. In weiterer Folge werden noch die Implementierungen der metaheuristischen Algorithmen für unser Optimierungproblem vorgestellt. In Kapitel 5 werden die experimentellen Resultate der verwendeten Metaheuristiken dargestellt und besprochen. Diese Arbeit schließt mit einer Zusammenfassung der Ergebnisse und einigen Anmerkungen (Kapitel 6) ab.

Kapitel 2

Moderne Portefeuille Theorie

2.1 Rendite und Risiko von Portefeuilles

Markowitz verwendete für Rendite und Risko die Konzepte „erwartete Renditen“ und „Varianz von Renditen“ aus der Statistik.[9] Nachfolgend möchten wir uns kurz einige ausgewählte Konzepte über Erwartungswert und Varianz widmen.[10]

Wenn a und b zwei Zufallsvariablen sind, so läßt sich der Erwartungswert der Summe von a und b als die Summe der einzelnen Erwartungswerte berechnen:

$$E(a+b) = E(a) + E(b)$$

Der Erwartungswert der Zufallsvariable a multipliziert mit einer Konstante c enpricht der Multiplikation von c mit dem Erwartungswert von a:

$$E(c \cdot a) = c \cdot E(a)$$

Wenn $P_{i,t}$ der Preis eines i-ten Wertpapieres zum Zeitpunkt t ist, so läßt sich die diskrete Rendite des i-ten Wertpapiers folgendermaßen ausrechnen:

$$r_i = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}$$

bzw.

$$r_i = \frac{P_{i,t}}{P_{i,t-1}} - 1$$

---

[9] Eigene Übersetzung aus dem Englischen »*expected returns*« und »*variance of returns*« siehe Markowitz [45, S. 77]

[10] Vgl. Burtscher [7, S. 3] und Markowitz [45, S. 79]

Grundsätzlich behilft man sich bei der Prognose von Renditen der Wahrscheinlichkeitsrechnung. Die Renditen in einer Folgeperiode werden hierbei als Zufallswerte interpretiert, für die man den Erwartungswert $E(r_i)$ und die Varianz $Var(r_i)$ bzw. Standardabweichung $\sigma(r_i)$ berechnet.[11]

Weiters kann man mit Hilfe der Szenariotechnik für zukünftige „Umweltzustände" z_j subjektive Eintrittswahrscheinlichkeiten $p(z_j)$ zuordnen, wobei

$$\sum_{j=1}^{m} p(z_j) = 1$$

gilt und mit m die Anzahl der Umweltzustände bezeichnet wird. Der Erwartungswert der zustandsabhängigen Rendite $r_i(z_j)$ läßt sich folgendermaßen ausrechnen:[12]

$$E(r_i) = \sum_j p(z_j) \cdot r_i(z_j)$$

Die Standardabweichung $\sigma(r_i)$ (*Risko*[13], *Volatilität*) der zustandsabhängigen Rendite $r_i(z_j)$ erhält man durch[14]

$$\begin{aligned} \sigma(r_i) &= \sqrt{Var(r_i)} \\ &= \sqrt{E(r_i^2) - E(r_i)^2} \\ &= \sqrt{\left[\sum_j r_i^2(z_j) \cdot p(z_j)\right] - \left[\sum_j r_i(z_j) \cdot p(z_j)\right]^2} \end{aligned} .$$

Unterstellen wir einem Investor, der zum Zeitpunkt $t = 0$ W_0 Geldeinheiten in einem Portefeuille mit n riskanten Finanzierungstiteln investiert, wobei wir den Anteil am Anfangsvermögen W_0, der in den Finanzierungstitel i investiert wird, mit x_i bezeichnen, so erhält man die erwartete Portefeuillerendite $E(r_P)$ als gewichteten Mittelwert aller im Portefeuille vertretenen Einzelrenditen:[15]

---

[11] Vgl. Fischer [19, S. 36]

[12] Vgl. Markowitz [45, S. 80]

[13] Im Gegensatz zu Fischer [19, S. 36] setzt Markowitz [45, S. 89] Risiko mit Varianz gleich

[14] Vgl. Fischer [19, S. 36]

[15] Vgl. Fischer [19, S. 37] , Fischer [20, S. 259f] und Markowitz [45, S.81]

$$E(r_P) = \sum_{i=1}^{n} E(r_i) \cdot x_i$$

mit

$$\sum_{i=1}^{n} x_i = 1$$

Das Risiko des Portefeuilles hängt aber nicht nur vom Risiko der einzelnen Wertpapiere ab, auch von den Kovarianzen der Wertpapierrenditen:[16]

$$Cov(r_i, r_j) = E(r_i \cdot r_j) - E(r_i) \cdot E(r_j)$$

Die Varianz der Rendite eines Portfeuilles erhält man durch

$$Var(r_P) = \sum_{i=1}^{n} x_i \cdot Var(r_i) + 2 \cdot \sum_{i=1}^{n} \sum_{j>1}^{n} x_i x_j Cov(r_i, r_j)$$

bzw.

$$Var(r_P) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j Cov(r_i, r_j)$$

wobei gilt

$$Cov(r_i, r_j) = Cov(r_j, r_i)$$

$$Cov(r_i, r_i) = Var(r_i)$$

2.2 Risikoeinstellung von Investoren

Ein Investor versucht grundsätzlich in der Entscheidungstheorie den erwarteten Nutzen aus dem Endvermögen W_1 zu maximieren.

$$E[U(W_1)] = \sum_{i=1}^{n} U(W_1(z_i)) \cdot p(z_i)$$

Zur Beurteilung der Einstellung des Investors vergleicht man den Nutzen aus dem erwarteten Endvermögen mit dem erwarteten Nutzen aus dem Endvermögen. Die Differenz dieser beiden Größen wird als Risikomaß, das wir mit Φ bezeichnen, verwendet:

$$\Phi = U[E(W_1(z_i))] - E[U(W_1(z_i))]$$

Bei einer quadratischen Nutzenfunktion oder unter der Annahme, dass dem Endvermögen des Investors eine symmetrische Verteilung mit zwei Parametern zugrunde

[16] Vgl. Markowitz [45, S. 80-81]

liegt, hängt das Risikomaß Φ nur von der Standardabweichung des Endvermögens $\sigma(W_1)$ ab.

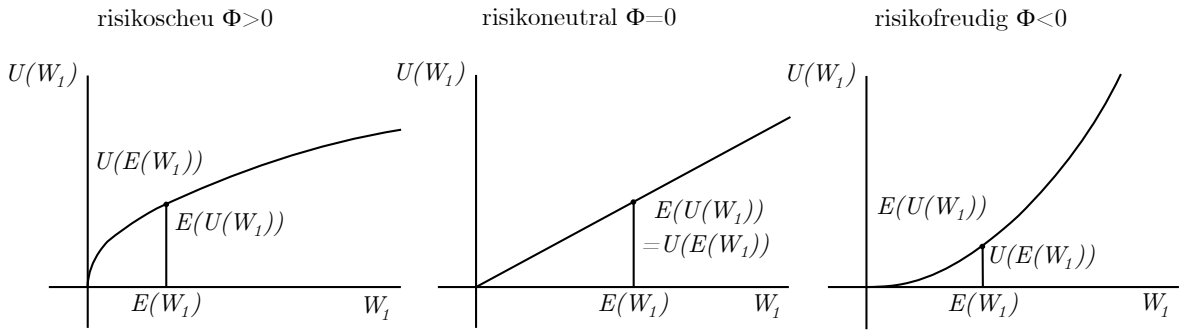

Abbildung 1: Risikoeinstellung; Quelle: Fischer [20, S. 261]

Bei den drei unterschiedlichen Arten von Investoren (vgl. Abbildung 1) wie risikoscheu, risikoneutral und risikoavers beschränken sich die interessanten Parameter, die seinen Nutzen generieren, auf das erwartete Endvermögen $E(W_1)$ und auf die Standardabweichung $\sigma(W_1)$.

Daraus resultiert die dreidimensionale Nutzenfunktion

$$U[E(r_P), \sigma(r_P)].$$

Um die dreidimensionale Nutzengebirgsfunktion in der zweidimensionalen $E\sigma$-Ebene abbilden zu können, berechnet man sich mathematisch gesehen Niveaulinien und erhält so genannte Isonutzenkurven (vgl. Abbildung 2).[17]

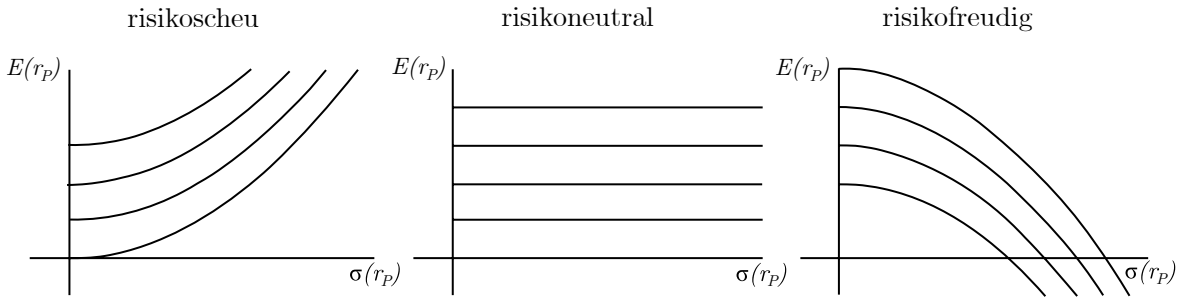

Abbildung 2: Isonutzenkurven eines Investors; Quelle: Fischer [20, S.264]

[17] Vgl. Fischer [19, S. 40ff]

2.3 Markowitz Modell

Ein wichtiger Meilenstein in Richtung moderner Portfolioanalyse ist der Artikel „Portfolio Selection“ [45] von Harry Markowitz aus dem Jahre 1952. Darin beschrieb er ein einperiodiges Modell zur Kapitalanlage[18], wo er das Konzept der Portfolioselektion statt „algebraisch-analytisch“ eher anhand „graphisch-geometrischen Illustrationen“ mit drei und vier riskanten Finanzierungstiteln erklärt.[19]

Markowitz teilte den Prozess zur Portfolioselektion in zwei Phasen[20] ein. In der ersten Phase beobachtet der Investor die Situation. Der Investor kommt dann zu einer persönlichen Einschätzung über die zukünftige Entwicklung der verfügbaren Wertpapiere. Die zweite Phase beginnt mit der persönlichen Einschätzung über die zukünftige Entwicklung und endet schließlich in der getätigten Wahl der Wertpapiere.

Markowitz macht folgende Annahmen zum Modell[21]:

- Auf dem Kapitalmarkt gibt es keine Transaktionskosten.
- Wertpapiere sind beliebig teilbar.
- Wertpapiere dürfen nicht beliebig leerverkauft[22] werden.
- Normalverteilte Renditen der riskanten Wertpapiere.[23]
- Auf dem Kapitalmarkt herrscht vollständige Konkurrenz und die Investoren sind Preisnehmer und verhalten sich kompetitiv.
- Die Preise auf dem Markt sind zufällig, somit ist das Endvermögen W_1 ebenso zufällig.
- Die rationalen Investoren sind risikoavers und möchten ihren erwarteten Nutzen aus ihren Endvermögen maximieren: $\max E[U(W_1)]$.[24]

Levy und Markowitz zeigten in einer späteren Arbeit, dass die Optimierung von

[18] Vgl. Fischer [19, S. 42]

[19] Vgl. Markowitz [45, S. 79]

[20] Vgl. Burtscher [7, S. 2] und Markowitz [45, S. 77]

[21] Vgl. Fischer [19, S. 42-43]

[22] Markowitz [45] definiert eine nicht Negativitätsbedingung für $x_i \geq 0$ im Gegensatz zu Black [4]

[23] Obwohl empirische Untersuchungen nicht exakt normalverteilt sind, kann diese theoretische Verteilung gute Approximationen der empirischen Verteilung liefern – vgl. Fama [13]

[24] Siehe Kapitel 2.2

Erwartungswert und Varianz (E-V) gegenüber der Optimierung von Nutzenfunktionen zu fast identischen Ergebnissen führen. [25] Der Nachteil der Optimierung von Nutzenfunktionen liegt darin, dass man oft sehr schwierig die exakte Nutzenfunktion eines Investors feststellen kann. Dieses Argument spricht grundsätzlich für die Verwendung der E-V Optimierung.

Besitzt nun ein Investor zum Zeitpunkt t=0 W_0 Geldeinheiten und möchte er in einem Portefeuille mit n riskanten Finanzierungstiteln investieren, wobei wir den Anteil am Anfangsvermögen W_0, der in den Finanzierungstitel i investiert wird, mit x_i bezeichnen, so wird sich der Investor die Frage stellen, wie er die x_i Anteile wählen soll, damit das Portefeuillerisiko effizient minimiert wird. Aus Kapitel 2.1 wissen wir, dass das Risiko eines Portefeuilles mit riskanten Finanzierungstiteln folgendermaßen lautet:

$$Var(r_P) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j Cov(r_i, r_j)$$

Nun könnten wir unser Problem als Optimierungsproblem bzw. Minimierungsproblem des Gesamtportefeuillerisiko formulieren, wobei wir Verbot von Leerverkäufen und Definition von x_i als prozentuelle Anteile von W_0 als Nebenbedingungen formulieren:

$$\min \sigma^2(r_P) = \sum_{j=1}^{N}\sum_{k=1}^{N} Cov(r_{j,} r_k) \cdot x_j \cdot x_k$$

Unter den Nebenbedingungen:

$$\sum_{j=1}^{N} x_j = 1 \text{ (Prozentuelle Anteile von } W_0)$$

$$x_j \geq 0 \text{, für } j = 1,...,N \text{ (keine Leerverkäufe)}$$

Diese Optimierung würde noch keine Präferenzen über Erwartungswerte des Investors berücksichtigen und würde jene x_i Anteile liefern, wo das Portefeuillerisiko minimal wäre. Das Portefeuille mit dem minimalen Risiko wird auch als Minimum Varianz Portfolio (MVP) bezeichnet.

[25] Vgl. Levy und Markowitz H. [39] und Kroll et al. [36]

Berücksichtigen wir in unserem vorangegangen Optimierungsproblem eine Erwartungswertpräferenz des Investors, erhalten wir dann das sogenannte Markowitz-Modell, das wie folgt beschrieben wird:

$$\min \sigma^2(r_P) = \sum_{j=1}^{N} \sum_{k=1}^{N} Cov(r_j, r_k) \cdot x_j \cdot x_k$$

Unter den Nebenbedingungen:

$$\sum_{j=1}^{N} E(r_j) \cdot x_j = E(r_P)$$

$$\sum_{j=1}^{N} x_j = 1$$

$$x_j \geq 0 \text{ , für } j = 1,...,N$$

Abbildung 3 zeigt eine skizzenhafte Darstellung einer Portfoliomöglichkeitskurve (dünne Kurve) und einer dazugehörigen Markowitz-Effizienzkurve[26] (dicke Kurve), auf der alle effizienten Portfeuilles liegen.

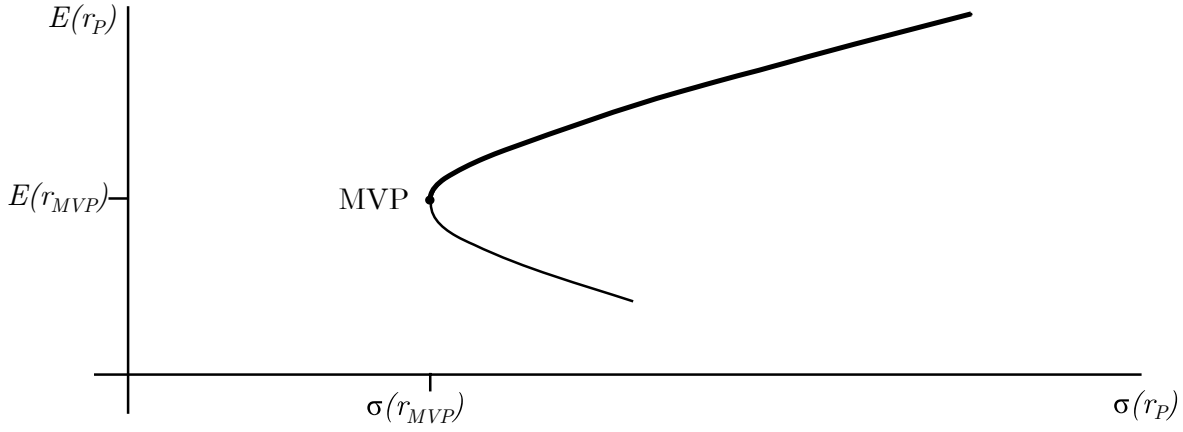

Abbildung 3: Markowitz-Effizienzkurve; Quelle: Fischer [19, S. 45]

2.4 Tobin Modell

Das Markowitz Modell sieht in der Zusammenstellung des Portefeuilles nur riskante Finanzierungstitel vor. Tobin [62] [63] erweiterte das Markowitz Modell um die Möglichkeit, zusätzlich in einen Finanzierungstitel zu veranlagen, der ein Risiko von Null aufweist und eine konstante Rendite r_f besitzt.

[26] In der englischen Literatur meist als »*Efficient Frontier*« bezeichnet.

Ein Investor hat grundsätzlich die Möglichkeit α Anteile seines Kapitals in den risikolosen Finanzierungstitel mit der risikolosen Rendite r_f und den Rest des Kapitals, $(1-\alpha)$ Anteile in ein riskantes Portefeuille M mit der erwarteten Rendite $E(r_M)$ und dem Risiko $\sigma(r_M)$ zu veranlagen. Die erwartete Rendite des Gesamtportefeuilles P ist[27]:

$$E(r_P) = \alpha \cdot r_f + (1-\alpha) \cdot E(r_M)$$

Weil r_f ein risikoloses Wertpapier ist, gilt:

$$\sigma^2(r_f) = 0$$

$$Cov(r_f, r_M) = 0$$

Somit ist die Gesamtvarianz des Portefeuilles P:

$$\sigma^2(r_P) = (1-\alpha)^2 \cdot \sigma(r_M)^2$$

oder:

$$\sigma(r_P) = (1-\alpha) \cdot \sigma(r_M)$$

Drückt man α mit Hilfe des Portefeuillerisikos aus, und setzt man dies in die Formel der Portefeuillerendite ein, so erhält man:

$$E(r_P) = \left(1 - \frac{\sigma(r_P)}{\sigma(r_M)}\right) \cdot r_f + \left[1 - \left(1 - \frac{\sigma(r_P)}{\sigma(r_M)}\right)\right] \cdot E(r_M)$$

Nach einer kurzen algebraischen Umformung erhält man dann:

$$E(r_P) = r_f + \frac{E(r_M) - r_f}{\sigma(r_M)} \cdot \sigma(r_P)$$

Wenn r_f, $E(r_M)$ und $\sigma(r_M)$ konstante Werte sind, ist der Zusammenhang zwischen der Portefeuillerendite $E(r_P)$ und dem Portefeuillerisiko $\sigma(r_P)$ linear. Die Steigung dieser Linie wird mit

$$\frac{E(r_M) - r_f}{\sigma(r_M)}$$

festgelegt.

[27] Vgl. Fischer [19] und Burtscher [7, S. 13]

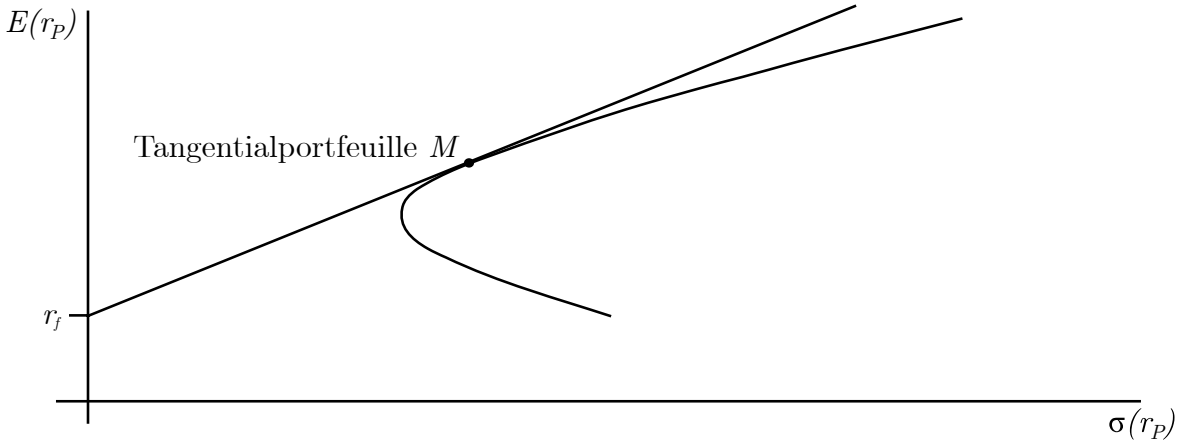

Abbildung 4: Tobin-Effizienzlinie; Quelle: Fischer [19, S. 57]

Letztendlich muss man die Steigung der Linie so wählen, dass sie die Markowitz Effizienzkurve (vgl. Abbildung 4) berührt, damit die Linie effizient wird. Man spricht dann auch von der Tobin-Effizienzlinie. Bei dieser Tobin-Effizienzlinie sind nur Kombinationen von risikolosen und riskanten Finanzierungstiteln effizient, wenn diese auf dieser Tobin-Effizienzlinie liegen.

Fasst man die obige Überlegung zusammen, erhält man folgendes Optimierungsmodell zur Bestimmung des Tobin-Tangentialportfeuilles M:

$$\max \frac{E(r_M)-r_f}{\sigma(r_M)}$$

unter den Nebenbedingungen:

$$\sum_{j=1}^{N}\sum_{k=1}^{N} Cov(r_j, r_k)\cdot x_j \cdot x_k = \sigma^2(r_M)$$

$$\sum_{j=1}^{N} E(r_j)\cdot x_j = E(r_M)$$

$$\sum_{j=1}^{N} x_j = 1$$

$$x_j \geq 0\,.$$

Die optimalen Anteile der riskanten Finanzierungstitel im Portefeuille des Investors sind

$$x_i = (1-\alpha)\cdot x_j^M$$

Vergleicht man die Effizienzkurven von Markowitz und Tobin, so können Investoren nach Tobin bei gleichen erwarteten Renditen ein Portefeuille bei viel niedrigerem Risiko

zusammenstellen als es bei Markowitz möglich wäre. Wenn man stattdessen das Portefeuillerisiko gleich lässt, so erreicht man bei Tobin eine höhere Portefeuillerendite als bei Markowitz.[28]

[28] Vgl. Fischer [19, S. 57ff]

Kapitel 3

Metaheuristik

Das Wort „Metaheuristik“ wurde erstmals von Glover[29] eingeführt. Das Wort ist eine Komposition aus zwei griechischen Wörtern, nämlich dem Verb „*heuriskein*“ (ευρισκειν), was soviel wie „finden“ bedeutet und dem Präfix „meta“, das man mit der deutschen Präposition „über“ übersetzen kann.

Das Besondere an den Metaheuristiken im Vergleich zu Heuristiken ist, dass Metaheuristiken grundsätzlich nicht an ein bestimmtes Problem gebunden sind, womit sie sich auf eine Vielzahl von Optimierungsprobleme anwenden lassen. Zu dieser Kategorie gehören unter anderem Genetische Algorithmen (GA), Lokale Suche (LS), Memetische Algorithmen (MA), Simulated Annealing (SA), Tabu Search (TS), Variable Neighborhood Search (VNS) und viele mehr. Metaheuristiken bieten aber wie Heuristiken keine Garantie dafür, dass die gefundene Lösung ein globales Optimum ist. Wolpert und Macready [65] [66] haben festgestellt, dass alle Optimierungsverfahren über alle Probleme hinweg gesehen gleich gut sind. Bekannt auch als das „No free-lunch-theorem“ für Suchverfahren.

Metaheuristiken haben auch Gemeinsamkeiten und genau hier setzt auch die in dieser Diplomarbeit verwendete metaheuristische Softwarebibliothek EALib[30] an.

Viele Metaheuristiken besitzen Strategien, lokalen Optima zu entkommen, ohne dass der Algorithmus von Neuen gestartet werden muss. Häufig wird dies über Änderungen der

---

[29] Vgl. Glover [23, S. 541]

[30] Siehe Kapitel 4.3.1

Nachbarschaften[31] erreicht. Auch Genetische Algorithmen besitzen eine solche Strategie. Diese Strategie wird in der Literatur in Bezug auf den Genetischen Algorithmus als Mutationsfunktion bezeichnet, die in einem späteren Kapitel genauer erläutert wird.

3.1 Definition

3.1.1 Lösungsraum

Bezeichnen wir mit P ein zu lösendes Problem, so heißt die Menge aller möglichen Lösungen des Problems Lösungsraum und Suchraum, die wir mit

$$S$$

bezeichnen.[32] Wir können zunächst vereinfachend annehmen, dass jedes Element des Lösungsraumes auch eine gültige Lösung darstellt:

$$x \in S$$

3.1.2 Nachbarschaftstruktur

Eine Nachbarschaftsstruktur ist eine Funktion

$$N : S \rightarrow 2^S,$$

die jeder gültigen Lösung eine Menge von Nachbarn

$$N(x) \subseteq S$$

zuweist. Eine Nachbarschaft ist meist implizit durch mögliche Veränderungen (Züge) definiert.[33]

3.2 Genetischer Algorithmus

Es ist nun über 30 Jahre her, dass Rechenberg [53], der als der Begründer von Evolutionsstrategien gesehen wird, evolutionäre Lösungsansätze für technische Probleme im Ingenieurbereich präsentierte. Unabhängig davon veröffentlichte Holland [29], der als Begründer vom Genetischen Algorithmus gilt, kurz darauf seine Studie über das Phänomen der natürlichen Adaption für Anwendungen in der Informatik.

---

[31] Siehe Kapitel 2.1.2

[32] Vgl. Schuster [57, S. 2]

[33] vgl. Gendreau [22, S. 41-42], Raidl [52, S. 34] und Wagner [64, S. 18]

3.2.1 Definition

Bevor wir uns mit den Funktionsweisen von Genetischen Algorithmen befassen, werden zunächst Definitionen und Begriffserklärungen behandelt.

3.2.1.1 Chromosom

Ein Chromosom oder Individuum ist ein Element von einem codierten Lösungsraum S_G. Chromosomen werden häufig als binäre Vektoren[34] codiert, die wir beim GA mit

$$v$$

bezeichnen. Das bedeutet, dass

$$v \in \mathrm{S}_G$$

Die Länge des binären Vektors wird mit

$$m$$

bezeichnet.

3.2.1.2 Population

Der Genetische Algorithmus operiert auf einer Menge von Individuen oder Chromosomen. Diese Menge von Individuen wird Population genannt. Die Population zum Zeitpunkt t bezeichnen wir mit

$$P(t).$$

Die Populationsgröße bzw. die Anzahl der Individuen bezeichnen wir mit

$$PopSize.$$

Somit ist die Population auch:

$$P(t) = \{v_1, v_2, ..., v_{PopSize}\},$$

wobei

$$P(t) \subset \mathrm{S}_G$$

gilt.

3.2.1.3 Genetische Operatoren

Mit Hilfe der genetischen Operatoren werden die Chromosomen verändert. Genetische

[34] Holland [29] verwendete in seiner ursprünglichen Arbeit auch die binäre Codierung.

Operatoren sind Mutations-Operatoren und Crossover-Operatoren, auf die wir später noch näher eingehen werden.

3.2.1.4 Initialisierung der Startpopulation

Die Startpopulation wird mit gültigen Chromosomen des Lösungsraumes S_G gefüllt, wobei die Elemente des Lösungsraumes meist zufällig ausgewählt werden. Denkbar wäre auch, dass man die Startpopulation neben den zufällig gewählten Elementen auch mit intuitiv guten Lösungen initialisiert.[35] Eine andere mögliche Überlegung könnte sein, dass man die Startpopulation mit den zuletzt besten Wert initialisiert, um den Algorithmus „weiterarbeiten“ zu lassen, den man zuvor abgebrochen hat.

3.2.1.5 Abbruchbedingung

Verglichen mit einem einfachen Nachbarschaftssuchverfahren, wo beim Erreichen eines lokalen Optimums abgebrochen wird, können Genetische Algorithmen grundsätzlich ewig laufen.[36] Abbruchbedingungen bei genetischen Algorithmen können wie folgt sein:[37]

- Die vordefinierte Laufzeit wurde erreicht.
- Vordefinierte Rundenzahl (Schleifendurchläufe) wurde erreicht.
- Keine weitere Verbesserung der Fitness konnte nach einer vordefinierten Rundenzahl oder vordefinierten Laufzeit erzielt werden.
- Chromosomen erreichen einen vorgegebenen Fitnesswert.

Die oben angeführten Möglichkeiten sind nicht als erschöpfende Aufzählungen zu betrachten, sondern nur als mögliche Beispiele anzusehen.

3.2.1.6 Bewertung und Fitness

Um das darwinsche Prinzip des „*survival of the fittest*“ anwenden zu können, müssen die Chromosomen quantitativ bewertet werden.[38] Grundsätzlich sind Bewertung[39] und

[35] Kann aber vielleicht auch den Nachteil haben, dass die Chromosomen meistens zunächst in Richtung der intuitiv guten Lösungen konvergieren, falls dies nicht sowieso erwünscht ist.

[36] Vgl. Reeves [54, S. 64]

[37] Vgl. Michalewicz [48, S. 67] und Schuster [57, S. 6-7]

[38] Vgl. Michalewicz [48, S. 38ff] und Schuster [57, S. 3-4]

[39] In der englischen Literatur wird Bewertung meist mit *Evaluation* bezeichnet.

Fitness verschiedene Begriffe, die jedoch in manchen Fällen vereinfachterweise gleichgesetzt werden. Die Bewertung misst die Güte der Lösung, also wie gut jedes einzelne Chromosom das Optimum approximiert. Mit der Fitness wird errechnet, mit welcher Chance sich ein Chromosom in der nächsten Generation reproduzieren kann.[40] Wir bezeichnen die Bewertungsfunktion für das i-te Chromosom v_i mit

$$eval(v_i).$$

Total Fitness

Die Total Fitness einer Population zum Zeitpunkt t ergibt aus:

$$F(t) = \sum_{i=1}^{PopSize} eval(v_i).$$

Durchschnittliche Fitness

Die durchschnittliche Fitness einer Population ergibt sich durch:

$$\overline{F(t)} = \frac{F(t)}{PopSize}$$

3.2.1.7 Fitness proportionale Selektion

Eine häufig eingesetzte Fitnessfunktion für Genetische Algorithmen ist die proportionale Fitness. Um die Werte der Bewertungsfunktion für die fitness proportionale Selektion einsetzen zu können, sollte sichergestellt werden, dass einerseits sich die Werte im positiven Wertebereich befinden, und andererseits höhere Fitnesswerte eine bessere Approximation des Optimums bedeuten. Dies kann durch eine Transformation der Bewertungsfunktion in die adjustierte Fitness

$$eval_a(v_i)$$

mit ein einem positiven Wertebereich erreicht werden, wobei dann

$$F_a(t) = \sum_{j=1}^{PopSize} eval_a(v_j)$$

gilt.[41]

Proportionale Fitness

Die normalisierte proportionale Fitness eines Chromosoms bezeichnen wir mit p_i und

---

[40] Vgl. Baumgartner [2, S. 143]

[41] Vgl. Schuster [57, S. 3-4]

lässt sich wie folgt berechnen:

$$p_i = \frac{eval_a(v_i)}{F_a(t)}$$

mit

$$\sum_{i=1}^{PopSize} p_i = 1$$

$$0 \le p_i \le 1.$$

In der Literatur wird häufig zur Veranschaulichung für die proportionale Selektion das sogenannte „Glücksrad- oder Rouletteprinzip“ verwendet. Dabei wird jedem Chromosom soviel Platz am Glückradumfang gewährt, wie die bereits berechnete Fitness an Wert angibt. Hilfreich wäre es, die Fitnesswerte p_i zu Selektionswahrscheinlichkeiten, die wir mit q_i^{Sel} bezeichnen, zu kumulieren. q_i^{Sel} wird wie folgt berechnet:

$$q_i^{Sel} = \sum_{j=1}^{i} p_j .$$

Zur Auswahl oder Selektion eines Chromosoms muss man nur noch das Glücksrad gedanklich drehen. Dies geschieht mit einer Zufallszahl[42], die wir mit

$$r$$

bezeichnen und einen Wertebereich von

$$0 \le r \le 1$$

aufweist.[43]

Haben wir die Zufallszahl r, wird sie mit den kumulierten Selektionswahrscheinlichkeiten q_i^{Sel} verglichen. Bei zutreffen der Bedingung

$$q_{i-1}^{Sel} < r \le q_i^{Sel}$$

wird das i-te Chromosom ausgewählt, wobei

$$q_0^{Sel} = 0$$

ist. Es ist offensichtlich, dass Chromosomen bei der Zufallsziehung öfter als ein Mal ausgewählt werden können. Dies kann dazu führen, dass gute Chromosomen öfters gewählt werden und schlechte Chromosomen aussterben. Die genaue Begründung wird

[42] Gewöhnlich nimmt man gleichverteilte Zufallsvariable (Gleitkommazahl).

[43] Michalewicz [48, S. 34-35]

in einen späteren Kapitel „Schema Theorem“ behandelt.[44]

3.2.1.8 Crossover

Der Crossover Operator[45] stellt vielleicht den wichtigsten Operator bei den Genetischen Algorithmen dar. Der Crossover Operator vermischt oder rekombiniert den Chromosomeninhalt zweier Chromosomen, die durch vorangegangener Selektion ausgewählt wurden.

In vielen Anwendungen wird häufig der Ein-Punkt-Crossover[46] herangezogen. Wir möchten in dieser Arbeit den Ein-Punkt-Crossover anhand von binären Chromosomen mit fester Länge genauer betrachten.

Angenommen, dass durch Selektionen mit Glücksrad folgende zwei Chromosomen selektiert wurden:

$$\begin{array}{lcccccccccc} v_1 & = & [1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1] \\ v_2 & = & [0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0] \end{array}$$

Nun bestimmen wir mit einer gleichverteilten ganzzahligen Zufallsvariable, die wir mit

$$r_{cpos}$$

bezeichnen und einen Wertebereich von

$$1 \le r_{cpos} \le m-1$$

aufweist, jene Stelle, wo das Chromosom gedanklich „durchgeschnitten“ wird. Der Term m bezeichnet die Anzahl der Bits eines Chromosoms. Nehmen wir einfach an, dass wir zufällig $r_{cpos} = 6$ wählen. Dies bedeutet, dass wir die beiden selektierten Chromosomen v_1 und v_2 ab der sechsten Bitposition miteinander tauschen:[47]

$$\begin{array}{lcccccc|cccc} v_1' & = & [1 & 0 & 0 & 1 & 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}] \\ v_2' & = & [0 & 1 & 1 & 1 & 1 & 1 & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1}] \end{array}$$

[44] Michalewicz [48, S. 35]

[45] In der englischen Literatur häufig mit »*Recombination Operator*« bezeichnet.

[46] Auch in der deutschen Literatur wird manchmal der Anglizismus »One-Point-Crossover« verwendet.

[47] Vgl. Michalewicz [48, S. 39-40]

Nun haben wir bereits zwei neue Individuen bzw. Chromosomen aus der alten Population $P(t)$ für die neue Population $P(t+1)$ erzeugt. Der Crossover-Vorgang wird solange wiederholt bis die neue Population vollständig aufgefüllt wird.
Weitere Varianten von Crossover sind z.B. Zwei-Punkt-Crossover oder Multi-Punkt-Crossover. Zum gegenwärtigen Zeitpunkt existieren für die wenigsten Problemstellungen eine allgemeingültige Empfehlung für ein optimales Crossover-Verfahren. Nur für ganz wenige Problemstellungen hat man gute Erfahrungen mit einer bestimmten Crossovervariante gemacht. So z. B. weiß man vom *Travelling-Sales-Man* Problem, dass man die alte Reihenfolge beim Crossover eher im groben Zügen belassen sollte, da diese Reihenfolge der Orte oft schon gut das Optimum approximieren können.

3.2.1.9 Mutation

Wenn man den Genetischen Algorithmus allein mit dem Crossover-Operator betreibt, wird man feststellen, dass der Genetische Algorithmus früher oder später gegen ein lokales Optimum bzw. globales Optimum konvergiert.[48] Um lokale Optima entfliehen zu können, kommt der Mutation-Operator ins Spiel. Bei einer binären Implementierung von Chromosomen mit fester Länge kann die Mutation durch Invertierung von einem Bit[49] an einer beliebigen Bitposition realisiert werden. Nachfolgend wird die Mutation anhand der Invertierung eines einzelnen Bits eines Chromosoms mit konstanter Mutationswahrscheinlichkeit vorgestellt.
Zunächst einmal definieren wir eine konstante Mutationswahrscheinlichkeit, die wir hier mit

$$p_m$$

bezeichnen, die einen Wertebereich von

$$0 \leq p_m \leq 1$$

aufweist. Diese konstante Mutationswahrscheinlichkeit gibt an, mit welcher Wahrscheinlichkeit ein einzelnes Chromosom einer Mutation unterzogen wird. Sie hängt stark von der Implementierung einer Problemstellung ab, sodass man im Allgemeinen keinen konkreten Wert empfehlen kann. Grundsätzlich soll es ein Wert sein, der dem

[48] Schuster [57, S. 8]

[49] Möglich wären auch Implementierung mit Veränderungen an einer Vielzahl von Bits eines Chromosoms.

Genetischen Algorithmus erlaubt, gegen gute Lösungen zu konvergieren, aber auch den Algorithmus zwingt, nicht allzu lange in lokalen Optima zu verweilen und weitersucht. Denkbar wäre auch eine dynamische Regelung des Mutationsoperators im laufenden Programm.[50] Eine Möglichkeit der Implementierung wäre in einer Schleife beginnend mit dem ersten Chromosom bis zu Anzahl der Chromosomen für jedes einzelnes Chromosom folgende zwei Schritte durchzuführen:

Schritt 1:

Mit einer gleichverteilten Zufallsvariable

$$r_m$$

bei einem Wertebereich von

$$0 \leq r_m \leq 1$$

Erzeugen wir einen Zufallswert.

Schritt 2:

Diesen Zufallswert r_m vergleichen wir mit der vordefinierte Mutationswahrscheinlichkeit p_m. Wenn

$$r_m \leq p_m$$

wird der Mutation-Operator in Gang gesetzt.[51]

3.2.1.10 Elitismus

Was wir hier auch nicht unterschlagen möchten, aber aus Gründen des Themenumfanges nur anschneiden möchten, ist die Rolle der Eliten bzw. der Elitismus. Grundsätzlich wird bei allen Metaheuristiken die beste Lösung mit dem besten Ergebnis gespeichert. Trifft man bei der Population von Genetischen Algorithmen keine Vorkehrungen zum Erhalt der besten Lösungen, können sehr gute Individuen nach einiger Zeit wieder verloren gehen, sodass der Algorithmus sehr langsam konvergiert. Um diese Tatsache entgegen zu wirken, führt man so genannte Eliten ein, die vor Veränderungen wie Crossover und Mutationen geschützt sind. Diese Eliten bestehen einfach aus Chromosomen mit sehr guten Lösungen. Oft werden diese Eliten bereits vor dem Crossover von der alten Population in die neue Population kopiert. Danach werden erst

[50] Vgl. Davis [11]

[51] Vgl. Baumgartner [S. 145-147] und Michalewicz [48, S. 40ff]

die restlichen unbesetzten Chromosomenstellen der neuen Population mit Hilfe des Crossover-Operators aufgefüllt. Diese Eliten dürfen auch nicht vom Mutations-Operator verändert werden. Die Eliten verbleiben solange in der Population bis neue bessere Individuen sie ablösen.[52]

3.2.2 Ablauf

Die grundlegende Idee von GA basiert auf der Imitation des Evolutionsprozesses mit seiner Strategie: Die besten Individuen einer Population überleben und geben ihr Erbmaterial an die nächste Generation ab. Dies kann man sich auch abstrakt als eine „iterative Schleife“ vorstellen, wobei eine Generation ein einzelner Rundendurchlauf der iterativen Schleife ist. Innerhalb dieser Schleife wird zunächst die genetische Struktur der einzelen Individuen bewertet, um eine Selektion der Individuen vornehmen zu können. Die Bewertung erfolgt mit Hilfe des Fitneßkonzeptes, wobei die Fitness als das Maß für die Angepaßheit eines Individuums an seine Umwelt bzw. an die Problemstellung dient. Eine hohe Fitness bedeutet nun, dass mit größerer Wahrscheinlichkeit das genetische Material an Nachkommen weitergegeben wird. Die Weitergabe des Genmaterials erfolgt durch in der Crossover-Phase (Rekombination bzw. Reproduktion). Die Mutation bei den GA imitiert die Strategie von der Natur, dass damit auch neue „Arten“ entstehen können, die mit herkömmlicher Crossover-Strategie nicht in dieser Geschwindigkeit oder nicht in diesem Umfang in Bezug auf die Angepasstheit der Umwelt erreicht werden können. [53]
Der grundsätzliche Ablauf eines GA in Pseudo-Code ist wie im Algorithmus 1 dargestellt.[54]

[52] Vgl. Baumgartner [S. 148-149] und Michalewicz [48, S. 58ff]

[53] Vgl. Keber [32 ,S. 1030]

[54] Vgl. Michalewicz [47]

```
procedure GA
begin
  t:=0;
  Initialisierung der Startpopulation(P(t));
  while (not Abbruchbedingung) do
    Fitness(P(t));
    t:=t+1;
    Selektion(P(t-1));
    P(t):=Crossover(P(t-1));
    P(t):=Mutation(P(t));
  end;
end;
```

Algorithmus 1: Genetischer Algorithmus

3.2.3 Schema Theorem

Das Konzept des Schemas führte Holland[55] in seinem zentralen Werk ein, um zu erklären, wie genetische Algorithmen funktionieren können.[56] Spätere Arbeiten wie von Macready und Wolpert [67] zeigen jedoch, dass das Schema Theorem im Allgemeinen über mehrere Generationen hinweg nicht haltbar ist. Dies bedeutet aber nicht, dass das Schema Theorem grundsätzlich nutzlos ist, sondern es bedeutet, dass das Schema Theorem nur kurzfristige Geltung besitzt.[57]

Schemata repräsentieren eine Menge von Chromosomen, die aus binären Genen von $\{0,1\}$ bestehen. Ein Schema S besteht also aus den binären Genen erweitert um ein *don't care* Symbol „*" auf $\{0,1,*\}$, wobei das *don't care* Symbol in unseren Fall lediglich aus einem Symbol der binären Symbolmenge $\{0,1\}$ beinhalten kann. Ein Schema S repräsentiert somit all jene Chromosomen, die an allen Bitpositionen, außer an jenen Bitpositionen an denen im Schema S ein *don't care* Symbol steht, mit dem Schema S übereinstimmen.[58] Beispielsweise repräsentiert das Schema [1,0,*,1,*] folgende vier

[55] Vgl. Holland [29] und Michalewicz [47]

[56] Vgl. Baumgartner [2, S. 140ff]

[57] Vgl. Reeves [54, S.60]

[58] Vgl. Schuster [57, S. 9]

Instanzen:

[1,0,0,1,0]

[1,0,0,1,1]

[1,0,1,1,0]

[1,0,1,1,1]

Jedes Schema S mit der Anzahl von $\eta(S)$ *don't care* Symbolen hätte $2^{\eta(S)}$ Instanzen bzw. 2^{η} Chromosomen.

Ordnung eines Schemas[59]:

Bezeichnen wir zunächst die Bit-Länge des Chromosoms mit m, das auch gleichzeitig die Länge des Schemas S ist. Unter Ordnung eines Schemas $o(S)$ versteht man die Anzahl der fixen Bit-Positionen, die kein *don't care* Symbol enthalten. Die Ordnung des Schemas erhält man mit:[60]

$$o(S) = m - \eta(S).$$

Definierende Länge eines Schemas[61]:

Die definierende Länge eines Schemas S bezeichnen wir mit $\delta(S)$. Man versteht darunter die Differenz zwischen der ersten und der letzten Bitposition eines Schemas, die kein *don't care* Symbol mehr enthält.[62]

	Bitposition										
Schema	1	2	3	4	5	6	7	8	$\eta(S)$	$o(S)$	$\delta(S)$
S_1	*	*	1	1	1	0	*	1	3	5	8 – 3 = 5
S_2	*	1	*	*	0	1	1	*	4	4	7 – 2 = 5
S_3	1	0	*	*	*	*	*	1	5	3	8 – 1 = 7

Abbildung 5: Beispiele zur Ordnung und definierender Länge eines Schemas

Bezeichnen wir die Anzahl der Chromosomen zum Zeitpunkt t, die durch das Schema S repräsentiert werden, mit $\xi(S,t)$. Die Fitness eines Schemas S zum Zeitpunkt t ergibt sich aus:

---

[59] In der englischen Literatur wird es als » Order of the Schema « bezeichnet; siehe Michaelewicz 1994

[60] Vgl. Michaelewicz [48, S. 46] und Schuster [57, S. 9]

[61] In der englischen Literatur wird es als »*Defining Length of the Schema*« bezeichnet; siehe Michaelewicz [48, S. 48ff]

[62] Vgl. Michaelewicz [48, S. 46] und Schuster [57, S. 9]

$$eval(S,t) = \frac{\sum_{j=1}^{\xi(S,t)} eval\left(v_{i_j}\right)}{\xi(S,t)}.$$

Zum Zeitpunkt $t+1$ ergibt sich mit der Populationsanzahl $PopSize$ folgende erwartete Anzahl an Chromosomen, die durch das Schema S repräsentiert wird:

$$\xi(S,t+1) = \underbrace{\xi(S,t)}_{a} \cdot \underbrace{PopSize}_{b} \cdot \underbrace{\frac{eval(S,t)}{F(t)}}_{c}.$$

a ist die Anzahl der Chromosomen, die zum Zeitpunkt t das Schema S repräsentieren.
b ist die Anzahl der Chromosomen oder auch Populationsgröße.
c ist die Wahrscheinlichkeit, dass ein Chromosom des Schemas S in einem Selektionsschritt ausgewählt wird.

Da wir aus dem vorhergehenden Kapitel 3.2.1.6 wissen, dass $\overline{F(t)} = F(t)/PopSize$ ist, können wir auch schreiben:

$$\xi(S,t+1) = \xi(S,t) \cdot \frac{eval(S,t)}{\overline{F(t)}}.$$

Diese Gleichung wird auch *Reproductive Schema Growth Equation* genannt. In der obigen Gleichung lässt sich leicht ablesen, ob in der nachfolgenden Periode $t+1$ die Anzahl der Chromosomeninstanzen des Schemas S sich verringert oder vergrößert. Liegt die Fitness des Schemas S über der durchschnittlichen Fitness $\overline{F(t)}$, so gilt:

$$eval(S,t) > \overline{F(t)} \Rightarrow \frac{eval(S,t)}{\overline{F(t)}} > 1 \Rightarrow \xi(S,t+1) > \xi(S,t).$$

Man sieht, dass bei einer Fitness des Schemas S, die über der durchschnittlichen Fitness liegt, die Anzahl der Chromosomeninstanzen in der folgenden Periode $t+1$ wächst.

Liegt die Fitness des Schemas S konstant um ε über der durchschnittlichen Fitness $\overline{F(t)}$:

$$eval(S,t) = \overline{F(t)} + \varepsilon \cdot \overline{F(t)},$$

dann gilt:

$$\xi(S,t) = \xi(S,0) \cdot (1+\varepsilon)^t.$$

Wenn ein Schema S konstant um ε über der durchschnittlichen Fitness $\overline{F(t)}$ liegt,

steigt die Anzahl der Chromosomen zum Zeitpunkt t exponentiell an.[63]
Nun haben wir die Rekombinationsstrategien wie Crossover und Mutation noch nicht mitberücksichtigt.

Crossover Operator

Betrachten wir zunächst den Einfluss des Crossover-Operators. Nehmen wir an, dass es zwei Schemata S_1 und S_2 und zwei Chromosomen v_1 und v_2 gibt. Weiters nehmen wir an, dass an der Bitposition 6 das Crossover vollzogen wird.

v_1	=	[1	0	0	1	0	1	1	0	1	1]	
v_2	=	[0	1	1	1	1	1	0	0	0	0]	
S_1	=	[*	*	*	1	0	1	*	*	*	*]	$,\delta(S_1)=2$
S_2	=	[1	0	*	*	*	*	*	0	1	1]	$,\delta(S_1)=9$
v_1'	=	[1	0	0	1	0	1	0	0	0	0]	
v_2'	=	[0	1	1	1	1	1	1	0	1	1]	

Das Schema S_2 wurde nach dem Crossover zerstört. Lediglich das Schema S_1 wird vom Chromosom v_1' repräsentiert. Wie die obige Tabelle darstellt, ist es intuitiv selbsterklärend, dass die Überlebenswahrscheinlichkeit eines Schemas sehr stark von der definierende Länge $\delta(S)$ abhängt. Ist die definierende Länge $\delta(S)$ eines Schemas S kurz, so ist die Wahrscheinlichkeit hoch, dass das Schema S eines Chromosoms den Crossover überlebt.
Bezeichnen wir die Wahrscheinlichkeit eines Crossovers mit p_C, die Länge eines Chromosoms mit m und die definierende Länge eines Chromosoms im Bezug auf ein Schema S mit $\delta(S)$. So lautet die Wahrscheinlichkeit $p_{sc}(S)$, dass ein Chromosom mit dem Schema S überlebt, wie folgt:

$$p_{sc}(S)=1-p_C\cdot\frac{\delta(S)}{m-1}$$

Die obige Gleichung ist noch nicht ganz vollständig. Würden die beiden Chromosomen v_1 und v_2 mit [0,1,...] beginnen und beide auf [...,0,1,1] enden, so hätte auch das Schema S_2 den Crossover überlebt. Die Wahrscheinlichkeit, dass ein Schema S ein

[63] Vgl. Michaelewicz [48, S. 48]

Crossover überlebt, sollte daher lauten:

$$p_{sc}(S) \geq 1 - p_C \cdot \frac{\delta(S)}{m-1}.$$

Berücksichtigen wir auch den Crossover Operator in der Formel über die Anzahl der Chromosomen zum Zeitpunkt $t+1$, die ein Schema S repräsentieren, ergibt sich folgende Formel

$$\xi(S,t+1) \geq \xi(S,t) \cdot \frac{eval(S,t)}{\overline{F(t)}} \cdot \underbrace{\left[1 - p_C \cdot \frac{\delta(S)}{m-1}\right]}_{\text{Korrekturterm für Crossover}}.$$

Mutations-Operator

Essenziell für die Bestimmung der Wahrscheinlichkeit p_{sm}, dass Schema S die Mutation überlebt, sind jene Bitpositionen, die keine *don't care* Symbole enthalten. Die Anzahl jener Bitpositionen ist gleich der Ordnung eines Schemas S oder auch $o(S)$. Bezeichnen wir die Mutationswahrscheinlichkeit mit p_m. Die Überlebenswahrscheinlichkeit eines Schemas S wäre dann

$$p_{sm} = (1 - p_m)^{o(S)}.$$

Wenn $p_m << 1$, dann können wir p_m durch

$$p_{sm} \approx 1 - p_m \cdot o(S).$$

approximieren.

Berücksichtigen wir den Crossover-Operator und den Mutation-Operator, erhalten wir eine neue *Reproductive Schema Growth Equation*:

$$\xi(S,t+1) \geq \xi(S,t) \cdot \frac{eval(S,t)}{\overline{F(t)}} \cdot \underbrace{\left[1 - p_C \cdot \frac{\delta(S)}{m-1} - p_m \cdot o(S)\right]}_{\text{Korrekturterm für Crossover und Mutation}}$$

3.2.4 Codierung

Grundsätzlich stellte Holland über das Schema Theorem das Konzept der binären Codierung mit fixer Länge vor. Nun gibt es Situationen in der Praxis, wo es besser wäre, einerseits die Länge der Codierung dynamisch zu halten, und andererseits die Codierung über eine Graycode Codierung vorzunehmen. Mit der Graycode-Codierung (vgl. Abbildung 6) bleibt nämlich die Hammingdistanz [25] zweier aufeinanderfolgende Werte

immer gleich eins.[64]

Dezimal	Binär	Graycode
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Abbildung 6: Graycode

3.2.5 Nebenbedingungen

Bei Optimierung von mehr als einer Variable, wo zusätzlich noch Nebenbedingungen eingehalten werden müssen, produzieren die Crossover-Operatoren und Mutation-Operatoren meist auch ungültige Lösungen. Häufig werden folgende Strategien verwendet, um die Einhaltung der Nebenbedingungen zu gewährleisten:

- Decodierer (*decoders*)
- Straffunktionen (*penalty functions*)
- Reparatur Algorithmen (*repair algorithms*)

3.2.5.1 Decodierer (decoders)

Decodierer sind spezielle Repräsentationsmechanismen, die Lösungen in gültige Lösungen überführen. Weitere Anforderungen an Decodierer sind:[65]

- Der Transformationsprozess soll möglichst effizient implementiert werden, damit der GA nicht durch die Implementierung des Decodierers zu stark gebremst wird.
- Eine kleine Änderung am Individuum soll auch nach der Transformation noch kleine Änderung der gültigen Lösung erhalten bleiben.

[64] vgl. Michaelewicz [48, S. 97-99]

[65] vgl. Michalewicz [48, S. 323-324] und Schuster [57, S. 16]

3.2.5.2 Straffunktionen (penalty functions)

Eine Möglichkeit das Problem mit der Einhaltung der Nebenbedingungen zu umgehen ist, dass man versucht, die Nebenbedingungen als Strafterme in die Zielfunktion einzubauen, und somit die Zielfunktion um die Anzahl der Nebenbedingungen mit Strafterme erweitert. [66] Bei Maximierungsproblemen wird die Zielfunktion um die Strafterme verringert, bei Minimierungsprobleme erhöht. Eine andere Möglichkeit von Straffunktion ist, dass ungültige Lösungen komplett ausgeschlossen werden (*death penalty*).

3.2.5.3 Reparaturalgorithmen (repair algorithms)

Damit die Nebenbedingungen eingehalten werden, werden die ungültigen Chromosomenvektoren mit Reparaturalgorithmen behandelt, um gültige Lösungen zu erhalten. Die Probleme liegen darin, dass es einerseits schwierig ist, geeignete Reparaturalgorithmen dafür zu finden und andererseits die verwendeten Reparaturalgorithmen zu rechenintensiv sein könnten, sodass die Performance darunter allzu sehr leidet. Ein weiterer wichtiger Punkt ist, dass die reparierten Chromosomenvektoren den vollständigen gültigen Lösungsraum mit gleicher Wahrscheinlichkeit erreichen können, falls eine Gleichverteilung der Wahrscheinlichkeit erwünscht ist.
Welche der oben angeführten Strategien optimal ist, hängt sehr stark von der Problemstellung und von der Implementierung der Problemstellung ab, sodass keine allgemeine Empfehlung dafür existiert.

3.3 Lokale Suche

3.3.1 Definition

Analog zu Kapitel 3.2.1 muß man bei LS grundsätzlich zunächst einmal den Suchraum als Lösungsrepräsentation der Zielfunktion definieren. Dann folgt eine Definition bzw. Auswahl der Nachbarschaften, auf das wir später näher eingehen werden.

---

[66] Vgl. Michalewicz [48, S. 141]

3.3.1.1 Initialisierung der Startlösung

Die Startlösung muss eine gültige Lösung aus dem Suchraum sein. Es ist in der Realität häufig nicht ganz trivial gleich eine gültige Lösung für die Initialisierung zu finden, wenn komplizierte Nebenbedingungen berücksichtigt werden.[67] Bei anspruchsvollen Zielfunktionen mit vielen Nebenbedingungen können auch Reparaturalgorithmen[68] zum Einsatz kommen.

3.3.1.2 Auswahl von Nachbarschaften

$N(x)$ ist die Nachbarschaft von x, wie sie in Kapitel 3.1.2 beschrieben wurde. Die einfache Lokale Suche besitzt in ihrer ursprünglichen Form häufig geringe Anzahl von Nachbarschaften. Bedingt durch die geringe Anzahl von Nachbarschaften kann LS, wenn überhaupt, nur schwer lokalen Optima entfliehen.

3.3.1.3 Abbruchbedingung

Als mögliche Abbruchbedingungen könnten definiert sein:[69]

- Wenn keine weiteren Verbesserungen erreicht werden
- Wenn eine voreinstellte Rechenzeit erreicht wird
- Wenn eine voreingestellte Anzahl von Iterationen durchgeführt wurde

3.3.2 Ablauf

Wurde eine adäquate Nachbarschaftsstruktur in Bezug auf den Lösungsraum definiert, beginnt die einfache LS grundsätzlich mit der Initialisierung der Startlösung. Die einfache LS wird auch als *Iterative Improvement* oder *Hill-Climbing* bezeichnet, weil in jeder Iteration nur jene neue Lösung übernommen wird, wenn diese besser als die alte Lösung in Bezug auf die Zielfunktion ist.[70] Die Iteration wird solange durchgeführt, bis die Abbruchbedingung erfüllt ist. Der Ablauf von der einfachen LS für ein Minimierungsproblem in Pseudo-Code (vgl. Algorithmus 2) ist folgendermaßen:[71]

---

[67] Vgl. Foacci et al. [17, S. 370]

[68] Siehe Kapitel 2.2.5.3

[69] Es handelt sich nicht um eine vollständige Aufzählung.

[70] Vgl. Wagner [64, S. 19]

[71] vgl. Raidl [52, S. 22]

```
Procedure Lokale Suche
begin
  x :=Initialisierung einer Startlösung;
  repeat
    wähle ein x' ∈ N(x);
    if f(x') < f(x) then
      x := x';
  until Abbruchbedingung erfüllt;
end;
```

Algorithmus 2: Lokale Suche

3.3.3 Diskussion

Die Lokale Suche hat in ihrer ursprünglichen Form einerseits den Vorteil, dass sie meist rasch ein lokales Optimum findet. Eine Verbesserung der Ergebnisse lässt sich dadurch erzielen, dass man die LS mehrmals hintereinander startet und daraus die beste Lösung auswählt.[72] Diese Vorgehensweise ist zwar einfach zu realisieren, kann dafür aber recht zeitaufwendig sein.[73] Mit zunehmenden Trend zu *Parallel Computing* relativiert sich aber auch diese Hürde.

Es ist aber nicht von der Hand zu weisen, dass die Implementierung der Lokalen Suche recht einfach vonstatten geht. Häufig wird LS auch mit anderen Heuristiken oder exakten Lösungsverfahren kombiniert.

3.4 Variable Neighborhood Search

Variable Neighborhood Search ist eigentlich eine recht junge Metaheuristik-Disziplin. Hansen und Mladenovic publizierten ihre ersten Arbeiten über Variable Neighborhood Search [49] [50] Mitte der 90-er Jahre. Die Idee dafür ist recht einfach beschrieben: systematischer Wechsel von Nachbarschaften innerhalb einer Suche. [74] Variable

[72] In der Literatur auch als Multi-Start Lokale Suche bezeichnet, siehe auch Raidl [52, S. 40] und Martí [46]

[73] Vgl. Foacci et al. [17, S. 370]

[74] Vgl. Mladenovic und Hansen [26, S. 146]

Neighborhood Search macht sich folgende drei Eigenschaften zu nutze:[75]

- Ein lokales Optimum in Bezug auf eine Nachbarschaftsstruktur ist nicht unbedingt ein lokales Optimum in Bezug auf eine andere.
- Ein globales Optimum ist ein lokales Optimum in Bezug auf alle möglichen Nachbarschaftsstrukturen.
- Für viele Probleme liegen lokale Optima relativ nahe beieinander.

Nun kann man diese Eigenschaften auf drei unterschiedliche Möglichkeiten zur Optimierung verwenden:[76]

- Deterministisch mit der Variable Neighborhood Descent Methode
- Stochastisch mit Reduced Variable Neighborhood Search Methode
- Als Vereinigung von deterministisch und stochastisch mit Basic Variable Neighborhood Search

Diese angeführten Lösungsmöglichkeiten sind auch die Grundzutaten für das General Variable Neighborhood Search, das in weiterer Folge vorgestellt wird.

3.4.1 Variable Neighborhood Descent

Wir bezeichnen die Anzahl der Nachbarschaften mit $l_{\max}$, die Startlösung mit x und die aktuelle gefundene Lösung mit x'. x'' stellt alle Lösungen der l-ten Nachbarschaft dar, wobei x' die beste gefundene Lösung von allen x'' ist. Variable Neighborhood Descent (VND) beginnt grundsätzlich mit der Nachbarschaft $l = 1$ und durchläuft iterativ $l_{\max}$ Nachbarschaften. Innerhalb einer Nachbarschaft wird die aktuelle gefundene Lösung x' mit der Lösung x verglichen. Falls $f(x')$ besser als $f(x)$ in Hinblick auf die Zielfunktion ist, so wird weiter in dieser Nachbarschaft gesucht, sonst wird systematisch in die nächste Nachbarschaft gewechselt. Der Ablauf von VND für ein Minimierungsproblem in Pseudo Code (vgl. Algorithmus 3) ist folgendermaßen:[77]

[75] Vgl. Mladenovic und Hansen [26, S. 146] und Raidl [52, S. 55]

[76] Vgl. Mladenovic und Hansen [26, S. 147]

[77] Vgl. Hansen und Mladenovic [27, S. 5] und Raidl [52, S. 56]

```
procedure Variable Neighborhood Descent (x)
begin
  l:=1;
  repeat
    finde ein x' mit f(x')≤f(x''),∀x''∈N_l(x);
    if f(x')<f(x) then
      x:=x';
      l:=1;
    else
      l:=l+1;
  until l>l_max;
  return x;
end;
```

Algorithmus 3: Variable Neighborhood Descent

Die grundsätzliche Strategie von Variable Neighborhood Descent ist die Erweiterung der Anzahl der Nachbarschaften. Mit zunehmender Anzahl der Nachbarschaften steigt auch die Wahrscheinlichkeit, ein globales Optimum zu erreichen.[78]

3.4.2 Reduced Variable Neighborhood Search

Bevor Reduced Variable Neighborhood Search (RVNS) zum Einsatz kommt, wird wiederum x mit der Startlösung initialisiert. Wir bezeichnen die Anzahl der Nachbarschaften mit $k_{\max}$, somit gibt es hier wieder einen Pool an Nachbarschaften $\left(N_1,...,N_{k_{\max}}\right)$, wobei diese oft nach ihrer Größe sortiert sind.[79] Grundsätzlich enthält RVNS zwei verschachtelte Schleifen, eine äußere und innere eine Schleife (vgl. Algorithmus 4).

Beginnen wir zunächst mit der inneren Schleife[80]: Man generiert eine zufällige Lösung x' dieser k-ten Nachbarschaft (*Shaking*). Nun wird die aktuelle gefundene Lösung x' mit der Lösung x verglichen. Falls $f(x')$ besser als $f(x)$ in Hinblick auf die Zielfunktion ist,

[78] Vgl. Hansen und Mladenovic [26, S. 147]

[79] Vgl. Hansen und Mladenovic [27, S. 8-9] und Raidl [52, S. 58]

[80] Siehe Algorithmus 4

so wird die Nachbarschaft wieder auf $k=1$ gesetzt, sonst wird systematisch in die nächste Nachbarschaft gewechselt. Die innere Schleife wird solange ausgeführt, bis die Bedingung $k > k_{max}$ erfüllt ist.

Die äußere Schleife[81] wird zunächst wie bei VND auf die $k=1$ Nachbarschaft auf gesetzt. Dann folgt die bereits beschriebene innere Schleife. Die äußere Schelife wird solange durchlaufen bis eine vordefinierte Abbruchbedingung, wie z. B. vorgegebene maximale Rundenanzahl, erreicht ist. Der Ablauf von Reduced VNS für ein Minimierungsproblem in Pseudo Code (vgl. Algorithmus 4) ist folgendermaßen:[82]

```
procedure Reduced VNS (x)
begin
  repeat //Beginn der äußeren Schleife
    k:=1;
    repeat //Beginn der inneren Schleife
      Shaking: generiere zufälliges x'∈N_k(x);
      if f(x')< f(x) then
        x:=x';
        k:=1;
      else
        k:=k+1;
    until k > k_max;
  until Abbruchbedingung erfüllt;
  return x;
end;
```

Algorithmus 4: Reduced Variable Neighborhood Seach

Reduced Variable Neighborhood Search ist sehr hilfreich bei großen Instanzen, wo die Lokale Suche sehr zeitintensiv ist.[83] Durch das Zufallselement kann man leichter lokalen Optima entkommen.

---

[81] Siehe Algorithmus 4

[82] Vgl. Hansen und Mladenovic [27, S. 12] und Raidl [52, S. 59]

[83] vgl. Hansen und Mladenovic [26, S. 147]

3.4.3 Basic Variable Neighborhood Search

Wie bei VND und RVNS wird auch bei Basic Variable Neighborhood Search (Basic VNS) zunächst einmal x mit der Startlösung initialisiert. Wir bezeichnen die Anzahl der Nachbarschaften mit k_{max} und die aktuelle gefundene Lösung mit x'.

Basic VNS unterscheidet sich vom RVNS nur in einem Punkt: Die innere Schleife[84] von Basic VNS enthält zusätzlich unmittelbar nach dem Shaking eine Lokale Suche, die versucht die zufällig gefundene Lösung x' weiter zu verbessern.

Der Ablauf von Basic VNS für ein Minimierungsproblem in Pseudo Code (vgl. Algorithmus 5) ist folgendermaßen:[85]

```
procedure Basic VNS (x)
begin
  repeat //Beginn der äußeren Schleife
    k:=1;
    repeat //Beginn der inneren Schleife
      Shaking: generiere zufälliges x' ∈ N_k(x);
      x':=lokale Suche mit (x');
      if f(x') < f(x) then
        x := x';
        k:=1;
      else
        k := k+1;
    until k > k_max;
  until Abruchbedingung erfüllt;
  return x;
end;
```

Algorithmus 5: Basic Variable Neighborhood Seach

Wie unschwer aus dem Pseudo-Code abzulesen, ist Basic Variable Neighborhood Search eine Kombination von Reduced Variable Neighborhood Search, das stochastische Elemente enthält, und Lokaler Suche, die deterministisch arbeitet.

---

[84] Siehe Algorithmus 5

[85] Vgl. Hansen und Mladenovic [27, S. 10] und Raidl [52, S. 61]

Die Abbruchbedingungen können z. B. wie folgt definiert sein:[86]

- Maximale definierte Prozessorzeit erreicht.
- Maximale Rundenzahl erreicht.
- Maximale Iterationen erreicht, wo keine Verbesserungen stattfand.

3.4.4 General Variable Neighborhood Search

Wir bezeichnen bei General Variable Neighborhood Search (General VNS) die Anzahl der Nachbarschaften mit $k_{\max}$ (VNS) bzw. mit $l_{\max}$ (VND), wobei die Nachbarschaftsstrukturen von VNS $(N_1,...,N_{k_{\max}})$ und VND $(\mathrm{N}_1,...,\mathrm{N}_{l_{\max}})$ nicht ident sind.[87]

Der Ablauf von General VNS unterscheidet gegenüber Basic VNS nur in einem Punkt: In der inneren Schleife[88] von General VNS wird zur Verbesserung der zufälligen Lösung x' unmittelbar nach dem Shaking nicht die Lokale Suche wie bei Basic VNS angewendet, sondern der Variable Neighborhood Descent Algorithmus angewendet. Der Ablauf einer General Variable Neighborhood Search für ein Minimierungsproblem in Pseudo Code (vgl. Algorithmus 6) ist folgendermaßen:[89]

[86] Vgl. Hansen und Mladenovic [26, S. 148]

[87] Vgl. Raidl [52, S. 62]

[88] Siehe Algorithmus 6

[89] Vgl. Hansen und Mladenovic [27, S. 11] und Raidl [52, S. 63]

```
procedure General VNS (x)
begin
  repeat //Beginn der äußeren Schleife
    k:=1;
    repeat //Beginn der inneren Schleife
      Shaking: generiere zufälliges x' ∈ N_k(x);
      x':=lokale Suche mit VND (x'):
        l:=1;
        repeat
          finde ein x'' mit
            f(x'') ≤ f(x'''), ∀x''' ∈ N_l(x');
          if f(x'') < f(x') then
            x' := x'';
            l:=1;
          Else
            l:=l+1;
        until l > l_max;
      if f(x') < f(x) then
        x := x'';
        k:=1;
      else
        k:=k+1;
    until k > k_max;
  until Abbruchkriterium erfüllt;
  return x;
end;
```

Algorithmus 6: General Variable Neighborhood Seach

Die Verallgemeinerung von Basic Variable Neighborhood Search zu General Variable Neighborhood Search wurde mit dem Austausch von der allgemeinen Lokalen Suche gegen den Variable Neighborhood Descent vollzogen.

3.4.5 Nachbarschaftsstruktur

Nach der Vorstellung der Verfahren von Variable Neighborhood Search stellt sich die Frage nach der allgemeinen Anforderung an die Nachbarschaftsstruktur, da sie einen sehr wichtigen Stellenwert bei diesem Algorithmus hat. Um lokale Optima zumindest theoretisch entkommen zu können, sollte die Vereinigung aller Nachbarschaften $N_1(x)...N_{\max}(x)$ den gesamten Suchraum S aller gültigen Lösungen beinhalten:

$$S \subseteq N_1(x) \cup N_2(x) \cup ... \cup N_{\max}(x), \forall x \in S$$

Die einzelnen Nachbarschaften können, müssen aber nicht disjunkt sein. Häufig sind geschachtelte Nachbarschaften die einfachere Wahl für die Implementierung:

$$N_1(x) \subset N_2(x) \subset ... \subset N_{\max}(x), \forall x \in S$$

Um die Nachbarschaft N_k zu erreichen, könnte man z. B. durch k-maliges Shaking der N_1-Nachbarschaftsstruktur realisieren.[90]

3.4.6 Diskussion

Variable Neighborhood Search ist eine recht junge Metaheuristik und besitzt vielversprechende Qualitäten. Obwohl das einfache Prinzip vom systematischen Wechsel der Nachbarschaften auf den ersten Blick recht bescheiden ausfällt, sind die Ergebnisse mit dieser Metaheuristik umso beeindruckender. Durch den Einsatz des allgemein gehaltenen Nachbarschaftskonzeptes eröffnet sich eine Vielzahl von Implementierungsmöglichkeiten. Auch lässt sich der Algorithmus im Gegensatz zu manchen Metaheuristiken laut Hansen [91] mathematisch leichter beschreiben und begründen.

[90] Vgl. Hansen und Mladenovic [27, S. 12] und Raidl [52, S. 60]

[91] Vgl. Hansen und Mladenovic [26, S. 174-176]

Kapitel 4

Diskrete Portfolio Selektion

4.1 Modellierung der diskreten Portfolio Selektion

Zunächst einmal möchte ich festhalten, dass wir in dieser Arbeit grundsätzlich vom Tobin-Modell ausgehen und es Schritt für Schritt für unser Problem „umgestalten“. Zum Zeitpunkt $t = 0$ wird das Anfangsvermögen V_0 in n verschiedene riskante Finanzierungstiteln investiert, und das ergebende Portefeuille P eine Periode lang gehalten. Der Finanzierungstitel mit Risiko Null weist eine konstanten Rendite von r_f auf. Wie bereits in der Einleitung erwähnt, geht die klassische Markowitz-Optimierung, wie auch die darauf aufbauende Tobin-Optimierung, von der Annahme der beliebigen Teilbarkeit von Finanzierungstiteln aus. Da wir in unserer Arbeit aber diese Annahme fallen lassen, um einen stärkeren Praxisbezug zu erreichen, führen wir zunächst diskrete Entscheidungsvariablen $q_i \in \mathbb{N}_0^+$ für $i = 1,...,n$ ein. Wir bezeichnen den Preis (je Mengeneinheit) des i-ten riskanten Finanzierungstitels mit P_{0i} zum Zeitpunkt $t = 0$. Um die Beziehung zum klassischen Tobin-Model aufrecht zu erhalten gilt folgendes:[92]

$$x_i = \frac{P_{0i} \cdot q_i}{V_0},$$

wobei x_i den Variablen der gewichteten Anteile, die wir vom Tobin-Modell kennen, entsprechen. Das Gesamtportefeuillerisiko $\sigma^2(r_P)$ mit der diskreten Entscheidungsvariable q_i ausgedrückt würde also lauten:

---

[92] Vgl. Keber [32, S. 1028]

$$\sigma^2(r_P) = \frac{1}{V_0^{\,2}} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} q_i \cdot q_j \cdot P_{0i} \cdot P_{0j} \cdot Cov(r_i, r_j)$$

Die erwartete Gesamtportefeuillerendite $E(r_P)$ ergibt:

$$E(r_P) = \frac{1}{V_0^{\,2}} \cdot \sum_{i=1}^{n} x_i \cdot E(r_i) \cdot q_i$$

Weiters möchten wir dem Investor selbst überlassen, wie viele verschiedene riskante Finanzierungstitel er in seinem Portefeuille halten möchte. Diesen Parameter bezeichnen wir mit k, wobei k einen Wertebereich von 1 bis maximal n besitzt. Die Entscheidungsvariablen, die die jeweiligen riskanten Titeln für das Portefeuille auswählen, bezeichnen wir mit b_i , wobei intuitiv die Summe von b_i gleich dem Parameter k ist:

$$\sum_{i=1}^{n} b_i = k \; mit \; b_i \begin{cases} 1 \, wenn \, x_i \neq 0 \\ 0 \, sonst \end{cases}$$

Was noch fehlt ist die direkte Verknüpfung der Entscheidungsvariablen von x_i und b_i . Dies wird wie folgt definiert:

$$x_j \leq b_j \forall j$$

$$x_j > b_j - 1 \forall j$$

Als letzte „Erweiterung“ des Tobin Modells ist die Berücksichtigung einer Budgetrestriktion, wobei das Anfangsvermögen dem verfügbaren Budget entspricht. Sollte das Budget vollständig ohne Rest ausgenutzt werden, wobei die dazugehörige Nebenbedingung z. B. folgendermaßen

$$\sum_{i=1}^{n} P_{0i} \cdot q_i = V_0$$

aussehen könnte, ist es intuitiv einleuchtend, dass damit der Lösungsraum sehr stark eingeschränkt wird, was nicht im Sinne des Modells ist. Im Extremfall könnte es sogar dazu kommen, dass dadurch nur unzulässige Lösungen existieren. Während Keber [32] dem Investor selbst überlässt, die Schranken für Überziehung und nicht vollständige Ausnutzung des Budgets zu definieren, nimmt das Modell diese Entscheidung dem Investor ab. Budgetüberschreitungen werden nicht erlaubt, dennoch kann ein Restbetrag des Budget übrigbleiben, solange sich keine ausgewählte ganze Aktie sich mehr im Restbetrag ausgeht. Diese Budgetrestriktion wird wie folgt definiert:

$$b_j\left(V_0 - \sum_{i=1}^{n} q_i P_{0i}\right) < P_{0j} \forall j$$

Nun werden die oben genannten Punkte zusammengefasst und nachfolgend zu einem Optimierungsproblem formuliert:

Die Zielfunktion:

$$\max \theta = \frac{E(r_P) - r_f}{\sigma(r_P)}$$

s.t.

$$E(r_P) = \sum_{i=1}^{n} x_i \cdot E(r_i)$$

$$\sigma^2(r_P) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i \cdot x_j \cdot Cov(r_i, r_j)$$

$$x_i = \frac{P_{0i} \cdot q_i}{V_0}$$

$$x_j \leq b_j \forall j$$

$$x_j > b_j - 1 \forall j$$

$$b_j\left(V_0 - \sum_{i=1}^{n} q_i P_{0i}\right) < P_{0j} \forall j$$

$$\sum_{i=1}^{n} b_i = k \; mit \; b_i \begin{cases} 1 \; wenn \; x_i \neq 0 \\ 0 \; sonst \end{cases}$$

$$q_i \in \mathbb{N}_0^+$$

$$x_j \geq 0 .$$

Somit haben wir zwar ein ganzzahliges quadratisches Maximierungsproblem mit den zugehörigen Nebenbedingungen aufgestellt, das sich mathematisch aber nicht trivial lösen lässt. Grundsätzlich handelt es sich um eine endliche Menge an Lösungsmöglichkeiten – vollständiges Problem. Eine logische Möglichkeit wäre, alle Kombinationen von q_i durchzuprobieren, was auch als vollständige Enumerierung bezeichnet wird. Weil die möglichen Kombinationen mit jeder zunehmenden Variable x_i polynomial ansteigen, würde die benötigte Rechenzeit für die Enumerierung auch polynomial ansteigen. Es handelt sich hierbei um ein sogenanntes NP-schweres

Problem[93]. Was einen Rechner bei dem derzeitigen Stand der Prozessoren bei sehr großen Instanzen überfordern würde.

4.2 Implementierung des Genetischen Algorithmus

Der Genetische Algorithmus wurde in der Programmiersprache Delphi unter Microsoft Windows XP implementiert.

4.2.1 Definition

Chromosom

In Analogie zu Kapitel 3.2.1.1 bezeichnen wir ein Chromosom oder Individuum als ein Element des Lösungsraumes mit

$$v.$$

Ein Chromosom v setzt sich wiederum zusammen aus ganzzahligen Genen, die wir mit

$$q$$

bezeichnen, wobei

$$q \in \mathbb{N}_0^+$$

analog zu der Problemstellung in Kapitel 4.1 wie folgt gewählt wurde. Die Länge eines Chromosomenvektors ist zugleich die Anzahl der Aktien n, somit ist

$$v = \{q_1, q_2, ..., q_n\}.$$

Bei der Wahl von k aus n Aktien werden die nicht ausgewählten Aktien mit dem Wert Null belegt.

4.2.2 Ablauf

Der Ablauf des implementierten GA unterscheidet sich im Grunde nicht vom Kapitel 3.2 vorgestellten GA, da die Strategien bei den Metaheuristiken sehr allgemein gehalten wurden. Ein wesentlicher Unterschied liegt darin, dass folgende Abschnitte (vgl. Algorithmus 1) zusätzlich implizit einen Reparaturalgorithmus enthalten:

- Initialisierung der Startpopulation
- Crossover

[93] abgekürzt für *Non-deterministic Polynomial-time hard*, ein Begriff aus der Komplexitätstheorie der Informatik

- Mutation

Der verwendete Reparaturalgorithmus wird im nächsten Kapitel noch genauer vorgestellt. Ein anderer wichtiger Unterschied gegenüber der beschriebenen Implementierung von Holland [29] liegt darin, wir mit einer ganzzahligen Repräsentation des Lösungsvektors statt der Binärrepräsentationen arbeiten. Wir nutzen den Vorteil von modernen Programmiersprachen, die eine ganzzahlige Repräsentationen über den ganzzahligen Datentyp *Integer* ermöglichen und Operationen auf *Integer* unterstützen. Weiters enthält Integer implizit die Binärrepräsentation, sodaß eine i. d. R. rechenzeitaufwendige Transformation zwischen binärer und ganzzahliger Repräsentation entfallen kann.[94]

4.2.3 Reparaturalgorithmus

Ein häufiger Kritikpunkt von Reparaturalgorithmen ist, dass die Performance des Algorithmus darunter leiden könnte. Darum wurde in dieser Diplomarbeit versucht, eine Reparaturfunktion zu entwickeln, die schnell ist und doch den Suchraum nicht allzu sehr einschränkt.

Der Reparaturalgorithmus muss dafür sorgen, dass einerseits die maximale Budgetbedingung eingehalten wird, und andererseits dass das Budget wirklich ausgereizt wird. Letzteres wird dadurch erreicht, dass im Reparaturalgorithmus überprüft wird, ob sich noch die Aktie mit dem kleinsten Preis von den ausgewählten Aktien im Budget ausgeht.

Implementierung in Pseudo Code (vgl. Algorithmus 7) ist folgendermaßen:

[94] Vgl. Keber [32, S. 1031]

```
procedure Reparaturalgorithmus (v)
begin
  //Prüfen ob MaxBudget überschritten wurde:
  while currentBudget(v)>MaxBudget do
  begin
    x:=Wähle eine zufällige Aktie von v, das ≥1 ist;
    //Berechne wieviel Stück von Aktie x weggenommen wird
    takeaway:=(currentBudget(v)-MaxBudget)/Aktienpreis[x];
    if (v[x]-takeaway) >= 1 then
      v[x]:=v[x]-takeaway
    else
      v[x]:=1;
  end;
  MinPreis:=Kleinsten ausgewählten Aktienpreis ermitteln;
  BudgetLowBound:=MaxBudget-MinPreis;
  //Prüfen ob das MaxBudget vollständig ausgereizt wurde:
  while currentBudget(v)<BudgetLowBound do
  begin
    x:=Wähle eine zufällige Aktie von v, welche ≥1 ist;
    AddOn:=(MaxBudget-currentBudget(v))/Aktienpreis[x];
    if AddOn>0 then v[x]:=v[x]+Addon;
  end;
  return (v);
end;
```

Algorithmus 7: Implementierter Reparaturalgorithmus

Ein weiterer Grund, der für die Verwendung des Reparaturalgorithmus in dieser Diplomarbeit spricht, ist die Wiederverwendbarkeit des Algorithmus bei der Implementierung der anderen Metaheuristiken mit EALib für die Evaluierung.

4.2.4 Fitness

Zur Bewertung möchten wir dies an einem 3-Wertpapierfall erläutern. Die detaillierten Angaben können Anhang A.1 entnommen werden.

Nehmen wir an, wir hätten insgesamt 3 riskante Finanzierungstiteln (n=3) aus der wir zwei Finanzierungstiteln (k=2) auswählen müssen. Die empfohlene gültige riskante

Wertpapierstückzahl bezeichnen wir hier mit q_i. Der Lösungsvektor, der die einzelnen q_i zusammensetzt bezeichnen wir mit v_1.

$$v_1 = [0 \quad 134 \quad 246]$$

Das obige Chromosom v_1 bedeutet, das man Null Stück vom ersten, 134 Stück vom zweiten und 246 Stück vom dritten riskanten Wertpapier kaufen soll. Wenn wir die Formel

$$x_i = \frac{P_{0i} \cdot q_i}{V_0}$$

aus Kapitel 4.1 nehmen, so erhalten wir folgende x_i:

i	1	2	3
x_i	0	0,49312	0,50676

Jetzt können wir die erwartete Portefeuillerendite $E(r_M)$ der riskanten Wertpapiere und die Gesamtvarianz $\sigma^2(r_M)$ mit folgenden Formeln berechnen:

$$E(r_M) = \sum_{i=1}^{n} x_i E(r_i)$$

$$\sigma^2(r_M) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j Cov(r_i, r_j)$$

Wir erhalten für unser Beispiel:

$$E(r_M) = 0{,}134779$$

$$\sigma(r_M) = 0{,}156528$$

Nun rechnen wir uns die Steigung der Tobin-Effizienzlinie mit

$$\max \theta = \frac{E(r_M) - r_f}{\sigma(r_M)}$$

aus. Wir erhalten die endgültige Bewertung des Chromosoms v_1, das durch die Steigung ausgedrückt wird:

$$eval(v_1) = \max \theta = 0{,}541624$$

4.2.5 Selektion

Hier wird eine Roulette-basierte proportionale Selektion, wie im Kapitel 3.2.1.7 beschrieben, eingesetzt. Bevor die Selektion durchgeführt wird, wird jedoch aufgrund des Elitismusprinzipes das beste Chromosom in die neue Population kopiert. Diese

Vorgehensweise garantiert, dass das beste Chromosom nicht wieder verloren geht. Wir müssen somit nur noch *PopSize*-1 Selektionen durchführen.[95]

4.2.6 Crossover

Grundsätzlich sollte der Crossover-Operator dafür sorgen, dass einerseits das Produkt nach der Kreuzung die Eigenschaften der beiden Elternchromosomen enthält und andererseits die strikte Nebenbedingung der Entscheidungsvariable k aus n-Aktien berücksichtigen. Um dies zu erreichen muß man mehrstufig vorgehen. Da Burtscher[96] von einer etwas anderen, aber sehr ähnlichen Situation, stand[97], haben wir seine Vorgehensweise für unseren Fall erweitert und angepasst. Wir möchten die einzelnen Schritte anhand zweier Beispielchromosomen, die wir mit Mutter- und Vaterchromosom bezeichnen, demonstrieren. Das Produkt des Crossover bezeichnen wir als Kindchromosom. In folgenden Fall haben wir zehn riskante Wertpapiere ($n=10$), aus denen wir drei ($k=3$) auswählen.

Chromosomposition	1	2	3	4	5	6	7	8	9	10
Mutterchromosom	[0	1	1	0	0	5	0	0	0	0]
Vaterchromosom	[0	2	0	0	0	0	3	0	2	0]

blau … Gene mit Wert Null

rot … Gene, die an gleicher Chromosomposition einen Wert ungleich Null sind

grün … Gene, die nicht an gleicher Chromosomenposition einen Wert ungleich Null sind

Schritt 1:

Wir suchen jene Chromosomstellen der beiden Elternchromosomen, die an gleicher Chromosomenposition einen Wert ungleich Null haben. In diesem Fall finden wir dies an der Chromosomposition 2:

Mutterchromosom	[*	**2**	*	*	*	*	*	*	*	*]
Vaterchromosom	[*	**1**	*	*	*	*	*	*	*	*]

[95] Vgl. Burtscher 2004 [7, S. 42]

[96] Vgl. Burtscher 2004 [7, S. 42]

[97] Die Arbeit von Burtscher [7] besitzt weniger restriktive Nebenbedingungen

Schritt 2:

Die grundlegende Strategie lautet die Gene der beiden Eltern auf das Kindchromosom zu übertragen, wobei das Kindchromosom die Eigenschaften von Mutter- oder Vaterchromosom enthalten soll. Wir nehmen in unserem Fall an, dass zufällig zunächst das Vaterchromosom ausgewählt wird. Jetzt erzeugen wir eine gleichverteilte Zufallszahl, die als eine Art Bitmaske für dieses Gen an der Chromosomposition 2 dienen soll. Die obere Schranke für den Wertebereich dieser Zufallszahl wird in dieser Arbeit aus Performancegründen angenommen, dass es die Anzahl der Aktien sei, in welche man mit dem ganzen Budget V_0 investiert. Somit besitzt die Zufallszahl für ein i-tes Gen einen Wertebereich von *Null* bis $\frac{V_0}{P_{oi}}$. In unserem Fall nehmen wir an, dass wir eine Zufallszahl von 3 zufällig gezogen haben, die wir als Bitmaske verwenden. Die Binärrepräsentation von 3 entspricht 011.

Der Wert 2 des Mutterchromosoms an dieser Position zwei entspricht einer Binärrepräsentation von 010. Der Wert des Mutterchromosoms wird mit der Bitmaske mittels dem logischen UND verknüpft:

$$010 \wedge 011 = 010$$

Der Wert 1 des Vaterchromosoms an der Position 2 entspricht der Binärrepräsentation 001. Dieser Wert wird diesesmal mit der *invertierten* Bitmaske logisch UND verknüpft:

$$001 \wedge (\neg 011) = 000$$

Zum Schluss werden die Ergebnisse von der Verknüpfung der Elternchromosomen noch einmal mit dem logischen ODER verknüpft:

$$010 \vee 000 = \mathbf{010}$$

Die Binärrepräsentation **010** lautet in dezimaler Schreibweise **2**. Diesen Wert **2** schreiben wir an der Position zwei in das Kindchromosom.

Kindchromosom | [- **2** - - - - - - - -]

Schritt 3:

Wir suchen jene restlichen Chromosomstellen der beiden Elternchromosomen, die einen Wert ungleich Null haben und merken uns jeweils die Chromosomenpositionen in zwei Hilfsvektoren, die wir mit h_1 und h_2 bezeichnen.

h_1	[3 2]
h_2	[7 9]

Schritt 4:

In unserem Fall bleiben noch zwei Stellen vom Kindchromosom übrig, die wir mit Elternchromosomwerten besetzen müssen. So müssen wir in unserem Fall zwei Mal zufällig entweder einen Wert von h_1 oder einen Wert h_2 auswählen, das gleichzeitig ein Positionswert des jeweiligen Elternchromosoms ist, sodass entweder der Wert vom Mutterchromosom oder der Wert vom Vaterchromosom übernommen wird. Wir nehmen in unserem Fall an, dass zuerst die Mutter und dann der Vater ausgewählt wurde.

Kindchromosom	[- 2 **1** - - - - - **2** -]

Schritt 5:

Die restlichen unbesetzten Stellen werden dann mit Nullen ausgefüllt. Wir erhalten zunächst ein Lösungselement, das aber nicht unbedingt gültig in Hinblick auf die Bugdetnebenbedingungen sein muss:

Kindchromosom	[**0** 2 1 **0** **0** **0** **0** **0** 2 **0**]

Schritt 6:

Nun kommt noch der Reparaturalgorithmus siehe Kapitel 4.2.3 ins Spiel, der diese Lösung auf eine gültige Lösung hin repariert.

4.2.7 Mutation

Wie im Kapitel 3.2.1.9 beschrieben, muss der Mutation-Operator im Allgemeinen dafür Sorge tragen, dass einerseits der Genetische Algorithmus aus lokalen Optima entfliehen kann, um potentielle bessere Lösungen finden zu können, und dass andererseits der gesamte gültige Suchraum erreicht werden kann. Da in unserem Fall die Implementierung des Chromosomenvektors nicht skalar binär[98] sondern mehrdimensional ganzzahlig erfolgt, greifen hier die Konzepte der Invertierung eines einzelnen Bits an einer Zufallsposition des Chromosoms oder Inversion nicht. Um dennoch das allgemeine Prinzip der Mutation zu gewährleisten, haben wir zwei Mutationsarten[99] implementiert.

[98] Burtscher [7] verwendet in seiner Arbeit einen binären Chromosomenvektor als Element des Suchraumes.

[99] Der Mutations-Operator muß nicht zwangsläufig genau zwei Nachbarschaften enthalten, sondern könnte auch mit viel mehr Nachbarschaften implementiert werden.

Die Mutationswahrscheinlichkeit beträgt konstant 0,01, dabei wird gleich wahrscheinlich eine der beiden Mutationsarten verwendet. Diese Werte wurden aus Versuchen ermittelt, der als Kompromiss zu sehen ist, wo dem Algorithmus Zeit gegeben wird, um gegen gute Optima konvergieren zu können und andererseits der Algorithmus sich nicht allzu lange bei einem lokalen Optima aufhält.
Mit folgenden zwei Mutationsarten sollte sichergestellt werden, dass der gesamte Suchraum zumindest theoretisch erreicht werden kann. Am Ende der Mutation folgt wieder der Reparaturalgorithmus, der sicherstellt, dass es sich um eine gültige Lösung aus dem Lösungsraum handelt.

4.2.7.1 GA - Mutationsart 1

Man verringert den Wert einer Chromosomposition, deren Wert größer Eins ist, um den Wert Eins. Dann wird der Wert einer anderen Chromosomposition, deren Wert auch ungleich Null ist, um den Wert Eins erhöht.

4.2.7.2 GA - Mutationsart 2

Man vertauscht den Wert einer Chromosomposition, die ungleich Null ist, mit dem Wert einer beliebigen anderen Chromosomenposition.

4.3 Implementierung weiterer Metaheuristiken

4.3.1 EALib

Historisch entstand die „*Evolutionary Algorithm Library*“ (EALib) aus Forschungsprojekten von Raidl [51], die ursprünglich „nur“ evolutionäre Algorithmen enthielt. Wagner [64] erweiterte die EALib im Zuge seiner Diplomarbeit um weitere Metaheuristiken. Die in dieser Arbeit verwendete EALib trägt die Version 2.0 und enthält im Gegensatz zu Wagners Arbeit [56] unter anderen die Metaheuristik Variable Neighborhood Search. Da die EALib als C++ Bibliothek dem Autor bei Abfassung dieser Diplomarbeit nur für das Linux Betriebssystem vorlag, mussten die Algorithmen weiterer Metaheuristiken nicht wie beim Genetischen Algorithmus mit Delphi unter Windows XP, sondern auch mit C++ unter Linux implementiert werden.
Die Metaheuristiken bei EALib arbeiten grundsätzlich mit den gleichen Eingangsdaten bzw. Instanzen, somit muss bei einem Vergleich der verschiedenen Algorithmen lediglich

die Instanz der Eingangsdaten einmal implementiert werden.

4.3.2 Definition

Wir bezeichnen ein Element des Lösungsraumes mit

$$x,$$

wobei x ein Vektor aus ganzen Zahlen ist.

$$x \in \mathrm{S}$$

x hat die Dimension n, das die Anzahl der Wertpapiere darstellt. Die ganzen Zahlen von Vektor x entsprechen den Stückzahlen der einzelnen Wertpapiere, die wir mit

$$q$$

bezeichnen.

$$x = \{q_1, q_2, ..., q_n\}$$

$$q \in \mathbb{N}_0^+$$

Bei der Wahl von k aus n Aktien werden die nicht ausgewählten Aktien mit dem Wert Null belegt.

Die Evaluationsfunktion von x bezeichnen wir mit

$$eval(x).$$

4.3.3 Reparaturalgorithmus

Der verwendete Reparaturalgorithmus von EALib entspricht genau den beschriebenen Reparaturalgorithmus von Kapitel 4.2.3, damit eine „grobe“ Vergleichbarkeit[100] erzielt werden kann. Der Reparaturalgorithmus von EALib wird bei der Initialisierung der Instanz erstmalig aufgerufen und ist implizit auch in jeder Nachbarschaft[101] enthalten, um die Gültigkeit der Lösung vom Suchraum auch im Bezug auf die Nebenbedingungen zu garantieren.

4.3.4 Nachbarschaftstruktur

Als Nachbarschaften wurde die beiden Mutationsarten des Genetischen Algorithmus von

[100] Bei den zwei unterschiedlichen Betriebssystemen ist es schwierig exakte Vergleiche anzustellen.

[101] Siehe Kapitel 4.3.4

Kapitel 4.2.7 verwendet, die nachfolgend einzeln vorgestellt werden.

4.3.4.1 Increment-Decrement Nachbarschaft

Diese Nachbarschaft entspricht in Prinzip der Idee von *Mutationsart 1* [102] beim Genetischen Algorithmus.

Um zu einer Nachbarschaftslösung zu kommen, werden aus dem Lösungsvektor x ein q_i und ein q_j ausgewählt, die Werte größer Eins aufweisen. Der Wert von q_i wird um ein Eins inkrementiert und der Wert von q_j wird um Eins dekrementiert.

Die Implementierung der Suchstrategie in Pseudo-Code (vgl. Algorithmus 8) ist folgendermaßen:

```
procedure Increment-Decrement Nachbarschaft
begin
  original:=x;        //x sichern
  bestobj:=eval(x);
  for i:=0 to i<n do
    for j:=0 to j<n do
    begin
      if (i<>j and x[i]>1 and x[j]>1) then
      begin
        x[i]:=x[i]+1;
        x[j]:=x[j]-1;
        Reparaturalgorithmus;
        if (eval(x)>bestobj) then break;
        x:=original; //original x wiederherstellen
      end;
    end;
end;
```

Algorithmus 8: Increment-Decrement Nachbarschaft

4.3.4.2 Swap Nachbarschaft

Diese Nachbarschaft entspricht der *Mutationsart 2 vom* Kapitel 4.2.7.2.

---

[102] Siehe Kapitel 4.2.7.1

Um zu einer Nachbarschaftslösung zu kommen, werden aus dem Lösungsvektor x ein q_i und ein q_j ausgewählt, deren Werte untereinander vertauscht.

Die Implementierung in Pseudo-Code (vgl. Algorithmus 9) ist folgendermaßen:

```
procedure Swap Nachbarschaft
begin
  bestobj:=eval(x);
  original:=x;                    //x sichern
  for i:=0 to i<n-1 do
    for j:=i+1 to j<n do
    begin
      temp:=x[i];
      x[i]:=x[j];                 // x[i] und x[j] vertauschen
      x[j]:=temp;
      Reparaturalgorithmus;
      if (eval(x)>bestobj) then break;
      x:=original;      // altes x zurückschreiben
    end;
end;
```

Algorithmus 9: Swap Nachbarschaft

4.3.5 Implementierung der Lokalen Suche

Da das Simple Randomized Local Search Modul nur noch die implementierten Nachbarschaften [103] als Übergabeparameter benötigt, sieht man hier einen offensichtlichen Vorteil von EALib, dass implementierte Algorithmen von anderen Metaheuristiken wiederverwendet warden können. Die Auswahl zwischen der *Swap Nachbarschaft* und der *Increment-Decrement Nachbarschaft* erfolgt zufällig. Diese zufällige Auswahl der Nachbarschaft ist bereits Bestandteil des Simple Randomized Local Search Moduls von EALib. Die zufällige Auswahl der zwei Nachbarschaften in Pseudo-Code (vgl. Algorithmus 10) ist folgendermaßen:

[103] Siehe Kapitel 4.3.4

```
procedure selectImprovement(bool find_best)
begin
  if random_int(2)=0 then Swap Nachbarschaft
  else Increment-Decrement Nachbarschaft;
end;
```

Algorithmus 10: Swap Nachbarschaft

4.3.6 Implementierung der Variable Neighborhood Search

Wie auch bei LS verwendet VNS die zwei implementierten Nachbarschaften[104]. Der Unterschied gegenüber LS liegt darin, dass bei VNS die Auswahl der Nachbarschaft systematisch erfolgt.

Das Shaking bei VNS wurde, wie der folgende Pseudo-Code (vgl. Algorithmus 11) zeigt, implementiert:

```
procedure shake(int k)
begin
  for i:=0 to i<k do
  begin
    z:=random(2);
    case z of
      0: Increment Decrement Nachbarschaft;
      1: Swap Nachbarschaft;
    end;
  end;
end;
```

Algorithmus 11: Implementiertes Shaking

[104] Siehe Kapitel 4.3.4

Kapitel 5

Anwendungsergebnisse

Wie auch in anderen Arbeiten wurde feststellt, dass mit zunehmender Anzahl von Wertpapieren die Suchzeit aufgrund des vergrößerten Suchraumes zunimmt. Dieses Kapitel beschäftigt sich mit den Ergebnissen des Genetischen Algorithmus und der mit EALib implementierten Metaheuristiken.

5.1 Daten und Parameter

Als Eingangsdaten für die durchgeführten Experimente wurden Daten aus dem Nasdaq100 mit den enthaltenen 100 Wertpapieren entnommen. Konkret wurde die Covarianzmatrix aus dem zweiten Halbjahr 2005 geschätzt. Für die Schätzung der Erwartungswerte wurde ein längerer Zeitraum von 2000 bis 2005 verwendet. Als letzten Parameter der klassischen Portefeuilleoptimierung nach Tobin für unsere Problemstellung wird der risikolose Zinssatz mit 5 % p. a. festgelegt. Weiters wurde die maximale Budgetobergrenze mit 100.000,- festgesetzt.
Um die Metaheuristiken grob vergleichen zu können, wurden statt Anzahl der Runden, eine Zeitdauer für alle Metaheuristiken veranschlagt. Es wurde für jedes k ein Durchlauf von mindestens 60 Sekunden verwendet und jeder Durchlauf 30 mal wiederholt. [105] Da bei EALib unter Linux die Threadzeit gemessen wurde, benötigt die ein kompletter Durchlauf von einem bestimmten mindestens 30 Minuten für eine verwendete

[105] Leider ist es nicht ohne weiteres möglich ein Programm unter dem Betriebsystem Linux mit einem Programm in Windows XP zu vergleichen. Die Zeitdauer sollten daher nur grobe Anhaltspunkte betrachtet warden.

Metaheuristik. Auch der Genetische Algorithmus unter Delphi wurde in einem eigenen Thread verlagert und die beste Fitness bei einer Threadzeit von mindestens 60 Sekunden ermittelt.[106]

[106] Implementierungsbedingt enthält GA eine größere Zeittoleranz von maximal ca. 2 Sekunden als EALib.

5.2 Fitness und Suchzeit von GA, LS und VNS im Vergleich

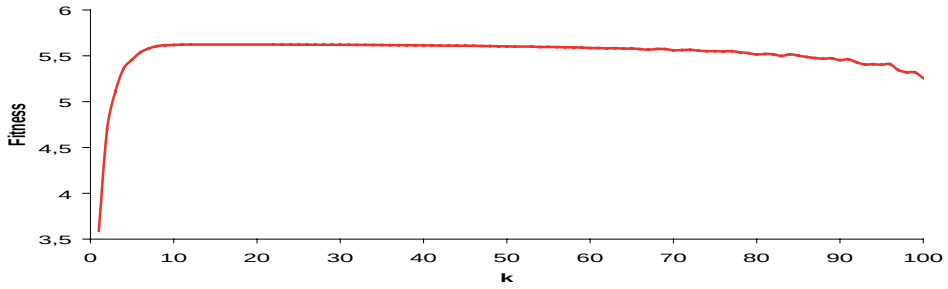

Abbildung 7: GA – Durchschnittliche Fitness

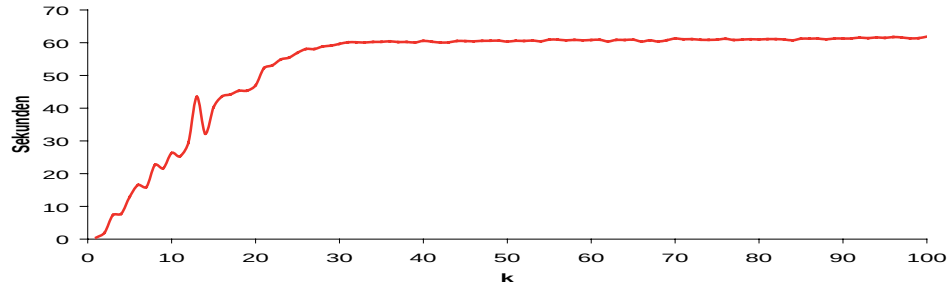

Abbildung 8: GA – Durchschnittliche Suchzeit

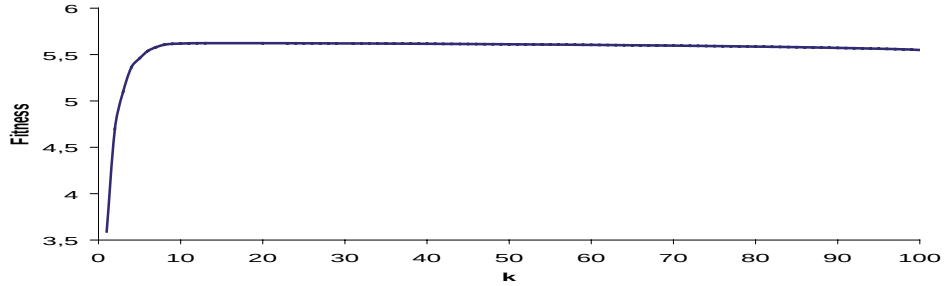

Abbildung 9: LS – Durchschnittliche Fitness

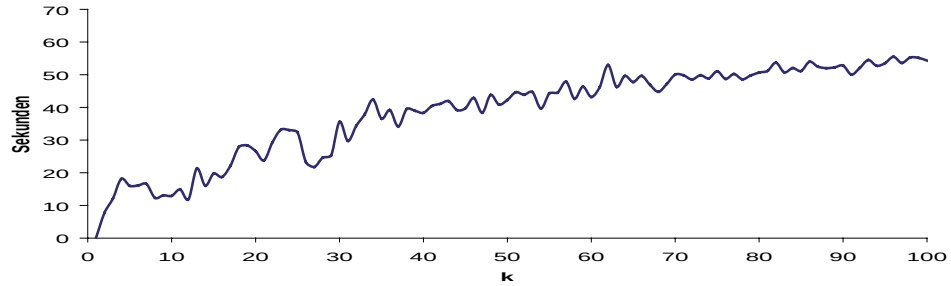

Abbildung 10: LS – Durchschnittliche Suchzeit

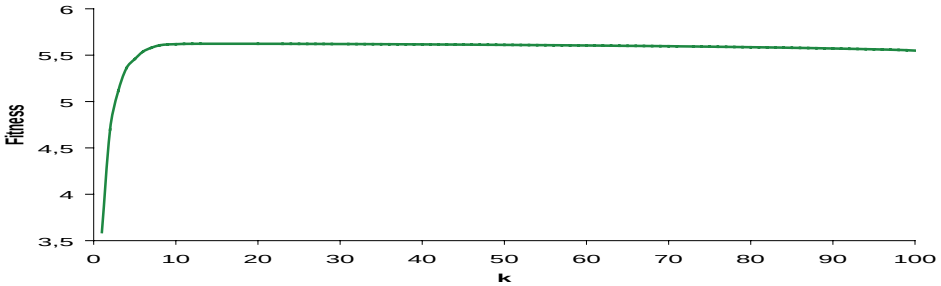

Abbildung 11: VNS – Durchschnittliche Fitness

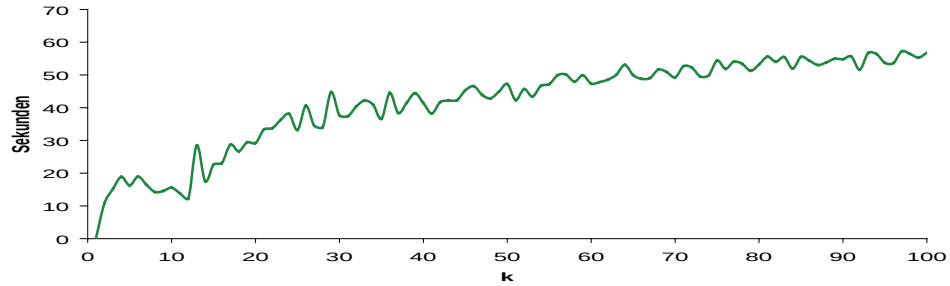

Abbildung 12: VNS – Durchschnittliche Suchzeit

5.3 Diskussion

Wenn man sich die Auswertung der Daten betrachtet, sieht man, dass einmal gewählte Wertpapiere meist wieder gewählt werden. Diese Tatsache hatte Burtscher auch schon feststellen müssen.[107] Bemerkenswert bei dieser verwendeten Instanz ist die Tatsache, dass ein optimales k aus n Wertpapieren bei einem fix vorgegebenen Budget existiert. Aus Anhang A.3 sieht man, dass die optimale Anzahl an riskanten Finanzierungtiteln von dieser verwendeten Instanz bei $k = 14$ liegt. Alle der drei verwendeten Metaheuristiken haben dieses Optimum gefunden.[108] Bei weiterer Erhöhung von k nimmt einerseits die Fitness wieder ab, und andererseits sieht man auch, dass die weiteren Aktien quasi annähernd mit der Anzahl von einem Stück in dem Portfolio aufgenommen werden, um die Nebenbedingung von k zu erfüllen. Diese Gesetzmäßigkeit ließe sich ausnutzen, um das nächste Portfolio unter der Nebenbedingung $k+1$ zu finden. Eine Möglichkeit bei der Initialisierung der Population wäre für ein bestimmtes k die sehr gute Lösungen von $k-1$ mitzuberücksichtigen.[109]

5.3.1 Genetischer Algorithmus

Bei kleinen k (ca. $k \leq 25$) liegt die Lösungsgüte des implementierten Genetischen Algorithmus auf dem selben Niveau wie bei den anderen. Bei großen k findet der genetische Algorithmus gegenüber den beiden anderen Metaheuristiken gute Optima langsamer, wie aus Abbildung 8 (im Vergleich zu Abbildung 10 und Abbildung 12) unschwer zu erkennen ist. Anhand der Daten im Anhang A.4 sollte es intuitiv ersichtlich sein, dass die lokalen Optima sehr eng beieinander liegen, sodass man nach einer Konvergenz des Genetischen Algorithmus zu einem lokalen Optimum auf Mutationen warten muss, bis wieder ein besseres Optimum gefunden werden kann. Das erklärt auch, warum die durchschnittlichen Fitnesswerte (vgl. Abbildung 7) schlechter sind als bei den anderen beiden Metaheuristiken (vgl. Abbildung 9 und Abbildung 11). Außerdem muß man fairer Weise auch die Effekte von zwei unterschiedlichen verschiedenen

[107] Vgl. Burtscher [7, S. 45]

[108] Siehe Anhang A.4

[109] Vgl. Kapitel 3.2.1.4

Betriebssystemen und mit zwei unterschiedlichen Programmiersprachen berücksichtigen.

5.3.2 Lokale Suche

Es scheint, dass die zufällige Auswahl der Nachbarschaften die besten Resultate der verwendeten Metaheuristiken von dieser Arbeit liefert.[110] Die Ergebnisse der Lokalen Suche ist zwar gegenüber dem Genetischen Algorithmus schon bei einer kleinen Anzahl von Finanzierungstiteln im Portefeuille deutlich besser. Der Vorsprung der Lokalen Suche bei den Ergebnissen gegenüber Variable Neighborhood Search ist aber erst bei sehr großer Anzahl von Finanzierungstiteln marginal besser.

5.3.3 Variable Neighbourhood Search

Durch den systematischen Wechsel der Nachbarschaften sind die Ergebnisse von Variable Neighborhood Search vergleichbar mit denen der Lokalen Suche. Was nicht ganz verwunderlich ist, da wir bei VNS auch genau dieselben zwei Nachbarschaften verwenden. Der Unterschied besteht nur in der Abfolge der Nachbarschaften, dass bei Variable Neighborhood systematisch erfolgt.

[110] Siehe Anhang A.2

Kapitel 6

Zusammenfassung

Ein wichtiges Kapitel der moderen Portefeuille Theorie ist die Selektion von Finanzierungstiteln eines Portfeuilles sowie deren optimale Gewichtung in Bezug auf Risikodiversifikation. Investoren mit einem beschränkten Budget können schon aufgrund von Transaktionskosten nicht in alle am Markt erhältlichen Finanzierungstitel investieren, um optimale Risikodiversifikation zu erreichen. Solnik konnte in seiner Arbeit [59] zeigen, dass bei internationaler Diversifikation bereits mit vergleichweise wenigen Titeln eine hinreichend gute Risikoreduktion erzielt werden kann. Empirische Untersuchungen zeigen auch, dass Portefeuilles in der Praxis häufig nur eine kleine Anzahl von Finanzierungstiteln enthalten.[111] Ein weiterer Kritikpunkt an der klassischen Portfolioselektion nach Markowitz ist, dass in der Realität die beliebige Teilbarkeit von Finanzierungstiteln nicht existiert. Somit tut sich für den Investor die Frage auf, welche Finanzierungstitel er in sein Portefeuille aufnehmen soll aufgrund der hohen Anzahl von Kombinationsmöglichkeiten[112].

Die vorliegende Arbeit formuliert die vorangegangen Punkte zu einem ganzzahligen Optimierungsproblem bzw. diskretes Portfeuilleoptimierungsproblem, welches das klassische Tobin-Modell zugrunde liegt. Eine Budgetrestriktion wird als Nebenbedingung im Optimierungsmodell berücksichtigt. Im Gegensatz zur Literatur[113], wird in dieser Arbeit dem Investor zusätzlich die Wahl überlassen, wie viele unterschiedliche

[111] Siehe z. B. Blume und Friend [5], Guiso et al. [24], Bertraut und Starr-McCluer [3] oder Börsch-Supan und Eymann [6]

[112] Siehe Kapitel 1

[113] Siehe Keber [32]

Finanzierungstitel er in seinem Portefeuille aufnehmen möchte. Ausgehend von der Problemstellung wurden die verwendeten Metaheuristiken, wie Genetischer Algoritmus, Lokale Suche und Variable Neighborhood Search, zur Lösung des Optimierungsproblems entwickelt und im Experiment auf eine reelle Datenbasis [114] angewendet. Die experimentellen Resultate zeigen, dass alle verwendeten Metaheuristiken in geringer Rechenzeit gute Lösungen gefunden haben. Weiters zeigen die Resultate, dass es bei dieser verwendeten Instanz eine optimale Anzahl von verschiedenen Finanzierungtiteln existiert. Danach nimmt die Fitness der Portefeuilles mit steigender Titelanzahl langsam kontinuierlich ab.[115] Dies ist auch als ein Erfolg dieser Diplomarbeit zu werten, da bei einer richtigen Wahl einer kleinen Anzahl von Finanzierungiteln aus der großen Gesamtmenge der Risikodiversifizierungseffekt bei dieser verwendeten Instanz bereits nicht mehr verbesserbar ist. Interessant wären weitere Untersuchungen mit weiteren reellen Instanzen.

Weitere metaheuristische Ansätze werden sicherlich in naher Zukunft in der Finanzwirtschaft folgen, schon alleine aus dem Grund, da Metaheuristiken zunächst relativ einfach zu implementieren sind und sich untereinander beliebig kombinieren lassen. Auch die Kombination mit einem exakten Verfahren wie *Branch and Bound* ist möglich. Da viele metaheuristische Algorithmen sich gut parallelisieren lassen, wächst die Performance der metaheuristischen Algorithmen mit zunehmender Parallelisierung der Prozessoren weiter an. [116] Als Beispiel lassen sich bei GA Fitness-, Crossover-, Mutations-[117] und Reparaturalgorithmus sehr gut parallelisieren. Somit stehen generell die Zeichen zum Lösen von diskreten Portefeuille-Optimierungsproblemen recht günstig.

[114] Nasdaq100; siehe Kapitel 5.1

[115] Siehe z. B. Abbildung 8

[116] Vgl. Keber [32, S. 1047]

[117] Vgl. Keber und Schuster [31, S. 707]

Anhang

A.1 Testdaten Small

Folgende Testdaten sind ident mit Fischer[118] und Keber[119].

Maximales Budget:

$$V_0 = 100000,-$$

Risikoloser Zinssatz

$$r_f = 0{,}05\%$$

i	A	B	C
P_{0i} (Preis)	139,-	368,-	206,-
$E(r_i)$ in % p.a.	8,00	15,00	12,00

$$Cov(r_i, r_j) = \begin{Bmatrix} 0{,}1^2 & 0{,}008 & 0{,}003 \\ & 0{,}2^2 & 0{,}018 \\ & & 0{,}15^2 \end{Bmatrix}, \text{ i,j} \in \{\text{A,B,C}\}$$

[118] Siehe Fischer 1996 [19, S. 237]

[119] Siehe Keber [32, S. 1037]

A.2 Durchschnittliche Fitness

k	GA:	LS	VNS:
1	3,5924916720	3,5924916720	3,5924916720
2	4,6976929500	4,6976881365	4,6976929460
3	5,1117103160	5,0960289989	5,1137244830
4	5,3635491660	5,3619084183	5,3624071810
5	5,4582863170	5,4590497943	5,4578222680
6	5,5381975210	5,5381937344	5,5381968960
7	5,5791351170	5,5790778752	5,5791348780
8	5,6059666750	5,6059678937	5,6059661580
9	5,6153170310	5,6153028101	5,6153031580
10	5,6185183930	5,6185175635	5,6185174110
11	5,6216544420	5,6216540625	5,6216542870
12	5,6227045460	5,6227029535	5,6227026170
13	5,6228382000	5,6228532352	5,6228514620
14	5,6229562520	5,6229778977	5,6229781350
15	5,6229638630	5,6229662129	5,6229667040
16	5,6229428350	5,6229438798	5,6229464890
17	5,6229172300	5,6229224384	5,6229223230
18	5,6228759510	5,6228812529	5,6228804300
19	5,6228132350	5,6228152832	5,6228153050
20	5,6226969940	5,6227035985	5,6227025940
21	5,6225697880	5,6225758079	5,6225772670
22	5,6223598110	5,6223710908	5,6223742440
23	5,6221444540	5,6221620112	5,6221669360
24	5,6219133530	5,6219419563	5,6219429380
25	5,6216247330	5,6216686333	5,6216737210
26	5,6212852250	5,6210991235	5,6213978670
27	5,6210072340	5,6210991235	5,6210971070
28	5,6206779080	5,6207996762	5,6208077550
29	5,6203771690	5,6204838599	5,6204927820
30	5,6199679750	5,6201721094	5,6201719200
31	5,6195608990	5,6198430345	5,6198404230
32	5,6190082550	5,6194893329	5,6194875840
33	5,6186747450	5,6191146276	5,6191186700
34	5,6182357710	5,6187372470	5,6187338950
35	5,6176917610	5,6183348632	5,6183344000
36	5,6165762760	5,6179384945	5,6179402260
37	5,6160678860	5,6175224140	5,6175210300
38	5,6156185230	5,6170987297	5,6170984780
39	5,6150463230	5,6166575134	5,6166596350
40	5,6135371850	5,6161732837	5,6161790950
41	5,6133729810	5,6156971139	5,6156966720
42	5,6113403270	5,6151958538	5,6151909640
43	5,6110337210	5,6146922856	5,6146896990
44	5,6106201660	5,6141813183	5,6141812490
45	5,6101294780	5,6136800875	5,6136781760
46	5,6090981850	5,6131605102	5,6131583500
47	5,6068348530	5,6126124935	5,6126157550
48	5,6062148090	5,6120677649	5,6120695390
49	5,6042576890	5,6114684025	5,6114708880
50	5,6023949300	5,6108722610	5,6108757510
51	5,6008146290	5,6102651451	5,6102596120
52	5,6006595780	5,6096460965	5,6096397820
53	5,6003560010	5,6090284283	5,6090199760
54	5,5960593550	5,6084057885	5,6084061120
55	5,5969271030	5,6077750501	5,6077806560
56	5,5952563690	5,6071420283	5,6071438960
57	5,5938852740	5,6064846917	5,6064862000
58	5,5921828900	5,6058133956	5,6058168260
59	5,5897544040	5,6051426234	5,6051406820
60	5,5843648060	5,6044000664	5,6043933440
61	5,5843441050	5,6036517185	5,6036499560
62	5,5818752910	5,6028870288	5,6028796180
63	5,5829770030	5,6021068826	5,6021080960
64	5,5785534480	5,6013279772	5,6013237740
65	5,5793252520	5,6005431496	5,6005446220
66	5,5735369430	5,5997618545	5,5997579600
67	5,5660297890	5,5989719385	5,5989637450
68	5,5743966450	5,5981482914	5,5981538380
69	5,5736065240	5,5973209541	5,5973092860
70	5,5608705190	5,5964122362	5,5964037410
71	5,5629438440	5,5954603013	5,5954607900
72	5,5653693270	5,5945007184	5,5944905160
73	5,5564454890	5,5935192789	5,5935267420
74	5,5509120210	5,5925442578	5,5925415540
75	5,5508846480	5,5914934796	5,5914957380
76	5,5469763850	5,5904226906	5,5904201600
77	5,5508725070	5,5893385543	5,5893249520
78	5,5377621830	5,5881922769	5,5881909440
79	5,5294241840	5,5870508350	5,5870335390
80	5,5144281210	5,5858796136	5,5858649610
81	5,5199326000	5,5847020362	5,5847081910
82	5,5162229230	5,5835101686	5,5835129050
83	5,4987615320	5,5822696714	5,5822658220
84	5,5173127590	5,5809678740	5,5809641290
85	5,5011803900	5,5796383991	5,5796242690
86	5,4874727200	5,5782144644	5,5782280050
87	5,4760199220	5,5767800179	5,5767719480
88	5,4678020020	5,5753377389	5,5753292530
89	5,4737162830	5,5737630282	5,5737665130
90	5,4529980300	5,5720690219	5,5720456190
91	5,4621827970	5,5703725090	5,5703375900
92	5,4294183640	5,5686463027	5,5686357360
93	5,4053220670	5,5668765265	5,5668481750
94	5,4085745870	5,5649742775	5,5649611560
95	5,4019852150	5,5629382048	5,5629172580
96	5,4088868870	5,5609111207	5,5608927150
97	5,3454933890	5,5587998115	5,5587786230
98	5,3206864540	5,5565652007	5,5565798940
99	5,3194051890	5,5534521331	5,5534402350
100	5,2571641190	5,5490987642	5,5490914460

A.3 Durchschnittliche Suchzeit

k	GA in Sekunden	LS in Sekunden	VNS in Sekunden
1	0.45	0.19	0.29
2	1.97	7.78	10.87
3	7.37	12.12	15.15
4	7.73	18.13	18.87
5	12.93	16.02	16.28
6	16.57	16.11	18.95
7	15.90	16.58	16.51
8	22.67	12.34	14.27
9	21.63	13.10	14.58
10	26.30	12.94	15.56
11	25.30	14.89	13.83
12	29.47	11.88	12.28
13	43.47	21.30	28.49
14	32.23	16.05	17.53
15	40.27	19.75	22.65
16	43.57	18.71	23.12
17	44.20	22.15	28.71
18	45.30	27.87	26.68
19	45.37	28.38	29.44
20	46.97	26.60	29.22
21	52.23	23.77	33.33
22	53.07	29.26	33.74
23	54.87	33.18	36.31
24	55.47	33.06	38.10
25	56.93	32.38	33.19
26	58.07	23.21	40.61
27	58.07	21.77	34.51
28	58.83	24.63	34.03
29	59.10	25.30	44.75
30	59.63	35.59	37.66
31	60.13	29.75	37.39
32	60.13	34.41	40.35
33	60.07	37.88	42.18
34	60.23	42.39	40.94
35	60.27	36.55	36.59
36	60.43	39.18	44.51
37	60.20	34.15	38.39
38	60.23	39.50	41.43
39	60.07	38.96	44.44
40	60.57	38.34	41.39
41	60.30	40.42	38.23
42	60.07	41.08	41.68
43	60.10	41.91	42.21
44	60.53	39.19	42.29
45	60.50	39.73	45.28
46	60.43	42.90	46.57
47	60.57	38.36	43.92
48	60.60	43.81	42.86
49	60.63	40.86	44.92
50	60.33	42.30	47.23
51	60.60	44.64	42.32
52	60.53	43.86	45.70
53	60.67	44.67	43.39
54	60.43	39.65	46.64
55	60.93	44.26	47.22
56	60.93	44.56	49.90
57	60.73	47.87	50.12
58	60.87	42.65	47.95
59	60.73	46.34	49.90
60	60.83	43.19	47.34
61	60.90	46.20	47.84
62	60.40	52.97	48.59
63	60.87	46.28	50.08
64	60.83	49.63	53.08
65	60.93	47.76	49.94
66	60.37	49.64	48.83
67	60.73	46.90	49.06
68	60.37	44.77	51.65
69	60.77	47.19	50.89
70	61.27	50.05	49.27
71	61.07	49.81	52.65
72	61.10	48.53	52.22
73	60.90	49.85	49.50
74	60.87	48.89	49.85
75	60.93	51.05	54.43
76	61.20	48.72	51.85
77	60.83	50.22	54.05
78	60.97	48.60	53.37
79	61.07	49.77	51.27
80	61.00	50.66	53.21
81	61.10	51.18	55.58
82	61.10	53.68	54.02
83	61.00	50.73	55.46
84	60.73	52.04	51.94
85	61.23	51.12	55.52
86	61.23	53.97	54.37
87	61.27	52.48	53.05
88	61.03	52.01	53.81
89	61.30	52.25	54.94
90	61.27	52.80	54.82
91	61.30	50.03	55.64
92	61.57	52.10	51.62
93	61.40	54.50	56.72
94	61.63	52.72	56.39
95	61.47	53.52	53.71
96	61.73	55.48	53.62
97	61.60	53.61	57.20
98	61.33	55.27	56.43
99	61.40	55.15	55.30
100	61.87	54.33	56.80

A.4 Beste Fitness

k	LS	VNS	GA
1	3,5924916720	3,5924916720	3,5924916720
2	4,6976929455	4,6976929455	4,6976929455
3	5,1138471944	5,1138471944	5,1138471944
4	5,3635491662	5,3635491662	5,3635491662
5	5,4628588910	5,4628588910	5,4628588910
6	5,5382040010	5,5382040010	5,5382040010
7	5,5791447133	5,5791447133	5,5791447133
8	5,6059694869	5,6059694869	5,6059694869
9	5,6153171150	5,6153171150	5,6153171157
10	5,6185205554	5,6185205554	5,6185205554
11	5,6216581955	5,6216581955	5,6216581955
12	5,6227069852	5,6227069852	5,6227069852
13	5,6228755617	5,6228755617	5,6228755617
14	5,6229810540	5,6229810540	5,6229810540
15	5,6229677439	5,6229677439	5,6229677463
16	5,6229528340	5,6229529933	5,6229529957
17	5,6229241647	5,6229241647	5,6229241671
18	5,6228850558	5,6228850558	5,6228850583
19	5,6228168648	5,6228168648	5,6228168673
20	5,6227063560	5,6227060748	5,6227063585
21	5,6225799785	5,6225799785	5,6225799810
22	5,6223822832	5,6223825333	5,6223810921
23	5,6221751686	5,6221751686	5,6221710150
24	5,6219492656	5,6219492656	5,6219492680
25	5,6216854382	5,6216854382	5,6216814929
26	5,6214018410	5,6214018410	5,6213946012
27	5,6211079639	5,6211087323	5,6211048511
28	5,6208148733	5,6208152882	5,6208066710
29	5,6205041966	5,6205048991	5,6204792175
30	5,6201809853	5,6201809853	5,6201563618
31	5,6198515811	5,6198511981	5,6197739776
32	5,6194955438	5,6194954246	5,6193812521
33	5,6191260988	5,6191260988	5,6190040023
34	5,6187482740	5,6187480262	5,6186522152
35	5,6183538617	5,6183557537	5,6182717140
36	5,6179507781	5,6179507781	5,6178534896
37	5,6175360870	5,6175317582	5,6173266177
38	5,6171180000	5,6171180000	5,6168547882
39	5,6166746345	5,6166746345	5,6162425120
40	5,6161891261	5,6161994120	5,6158577034
41	5,6157134909	5,6157111175	5,6151628302
42	5,6152133083	5,6152145305	5,6144747001
43	5,6147079479	5,6147076323	5,6138262699
44	5,6142019739	5,6142018234	5,6130930273
45	5,6136943936	5,6136943118	5,6128041714
46	5,6131777884	5,6131777066	5,6123674574
47	5,6126340613	5,6126339796	5,6104266991
48	5,6120862612	5,6120885692	5,6105355507
49	5,6114895887	5,6114935460	5,6095037409
50	5,6108944296	5,6108943478	5,6094203757
51	5,6102826708	5,6102818188	5,6070320036
52	5,6096712347	5,6096689335	5,6080546386
53	5,6090504024	5,6090483987	5,6078849333
54	5,6084304585	5,6084292979	5,6063509868
55	5,6078079996	5,6078079179	5,6045196624
56	5,6071642747	5,6071677797	5,6037620397
57	5,6065056520	5,6065086516	5,6028098355
58	5,6058373798	5,6058333235	5,6013037295
59	5,6051595473	5,6051595008	5,6001407622
60	5,6044277257	5,6044244981	5,5994035903
61	5,6036791456	5,6036806621	5,5982818381
62	5,6029147212	5,6029111703	5,5963749281
63	5,6021410296	5,6021409479	5,5957663871
64	5,6013573755	5,6013583515	5,5920207013
65	5,6005735400	5,6005701957	5,5956936968
66	5,5997872567	5,5997883266	5,5918735807
67	5,5990002646	5,5990001829	5,5873907226
68	5,5981792844	5,5981804315	5,5880451723
69	5,5973474171	5,5973473354	5,5905027191
70	5,5964403740	5,5964399106	5,5914278910
71	5,5954983634	5,5954977138	5,5840028482
72	5,5945381036	5,5945348934	5,5858380308
73	5,5935736536	5,5935710144	5,5855336452
74	5,5925836738	5,5925785368	5,5809965772
75	5,5915376255	5,5915380921	5,5791797200
76	5,5904690337	5,5904681975	5,5800870923
77	5,5893760517	5,5893794591	5,5792217542
78	5,5882363016	5,5882389116	5,5774448782
79	5,5870853469	5,5870892502	5,5752057080
80	5,5859247320	5,5859121249	5,5736234233
81	5,5847540174	5,5847437671	5,5724751437
82	5,5835570779	5,5835565766	5,5735538190
83	5,5823170632	5,5823135920	5,5660534982
84	5,5810149331	5,5810133196	5,5687629951
85	5,5796933039	5,5796893814	5,5578060301
86	5,5782719837	5,5782813949	5,5677299035
87	5,5768467648	5,5768466835	5,5648849636
88	5,5753907119	5,5753902612	5,5496267516
89	5,5738123135	5,5738196505	5,5523071759
90	5,5721370623	5,5721131074	5,5561557278
91	5,5704411799	5,5704456925	5,5507535521
92	5,5687051197	5,5687028305	5,5215723788
93	5,5669503316	5,5669491716	5,5506521145
94	5,5650297024	5,5650254061	5,5270497856
95	5,5630065572	5,5630048503	5,5342999243
96	5,5609801466	5,5609685792	5,5312671606
97	5,5588613405	5,5588666528	5,5142983554
98	5,5566528626	5,5566539440	5,5281762852
99	5,5535475277	5,5535392889	5,4938847047
100	5,5491912488	5,5491908347	5,4632640549

A.5 Genetischer Algorithmus selektierte Wertpapiere

k:	Fitness	8	21	24	28	33	40	45	47	57	69	70	80	91	94
1	3,5924916720								241						
2	4,6976929455								85		2721				
3	5,1138471944							312	67		2106				
4	5,3635491662						130	257	61		1917				
5	5,4628588910				180		135	224	55		1722				
6	5,5382040010				170		112	222	52		1677		74		
7	5,5791447133				167		114	200	48		1605	276	72		
8	5,6059694869				163		116	185	46		1551	299	70		
9	5,6153171157	56			147		110	170	46		1537	276	66	167	
10	5,6185205554	56		32	145		111	164	45		1521	275	64	167	
11	5,6216581955	47		37	145		111	162	44		1503	279	64	169	46
12	5,6227069852	42		33	140		110	161	44	27	1501	277	64	167	41
13	5,6228755617	41		33	141	18	110	159	44	25	1494	275	64	166	40
14	5,6229810540	41	6	32	140	17	110	158	44	24	1496	275	64	167	39
15	5,6229677439	40	7	32	140	17	110	158	44	24	1495	275	64	166	40
16	5,6229529933	40	6	33	140	17	110	158	44	24	1495	274	64	166	40
17	5,6229241647	40	7	32	140	17	110	158	44	25	1495	273	64	166	39
18	5,6228850558	40	6	32	140	17	110	158	44	24	1496	274	64	167	38
19	5,6228168648	40	6	32	140	17	110	158	44	24	1497	273	64	166	38
20	5,6227063560	40	6	32	140	16	110	158	44	25	1493	273	64	167	39
21	5,6225799785	40	6	32	140	16	110	158	44	24	1494	273	64	167	38
22	5,6223810897	40	7	32	140	18	110	157	44	24	1494	273	64	166	38
23	5,6221710126	40	6	32	140	17	110	157	44	24	1496	272	64	167	38
24	5,6219492656	40	6	32	139	16	110	158	44	24	1493	274	64	166	39
25	5,6216814905	40	6	32	139	16	110	157	44	24	1494	274	64	167	40
26	5,6213945988	40	7	32	139	17	110	157	44	24	1497	272	64	165	37
27	5,6211048187	40	5	32	139	16	110	158	44	24	1493	274	64	166	38
28	5,6208066685	40	5	32	139	13	110	159	44	23	1495	271	64	166	39
29	5,6204792151	40	7	32	139	14	110	157	44	22	1496	272	64	166	38
30	5,6201563593	40	5	32	139	11	110	158	44	25	1495	269	64	168	37
31	5,6197739753	41	5	30	139	13	110	157	44	24	1491	276	64	168	38
32	5,6193812496	39	5	32	139	12	110	157	44	24	1494	273	64	167	38
33	5,6190039998	38	5	29	138	16	110	158	44	25	1496	273	64	166	38
34	5,6186522127	38	7	31	139	19	110	155	44	24	1499	270	64	164	36
35	5,6182717117	41	6	32	137	14	109	157	44	23	1495	271	64	166	37
36	5,6178534869	36	6	30	140	14	110	157	44	21	1498	277	65	165	37
37	5,6173266151	36	7	28	136	24	109	159	44	24	1503	267	65	161	34
38	5,6168547855	35	3	31	137	17	109	160	44	30	1484	270	64	164	42
39	5,6162425096	41	6	31	137	11	110	160	44	17	1502	267	64	163	37
40	5,6158577008	40	2	33	137	2	109	159	44	20	1507	272	63	165	41
41	5,6151628283	47	5	34	142	9	111	156	44	14	1486	261	64	164	37
42	5,6144746977	42	7	32	137	12	108	158	44	16	1517	256	64	163	33
43	5,6138262669	37	8	28	137	4	108	163	44	15	1523	255	66	161	37
44	5,6130930255	47	5	37	127	9	109	152	44	28	1506	269	63	164	30
45	5,6128041678	28	2	32	141	8	109	162	44	25	1508	260	65	165	36
46	5,6123674544	34	3	33	144	26	112	159	43	18	1479	278	64	168	32
47	5,6104266964	32	8	22	144	34	112	146	44	18	1522	263	65	164	28
48	5,6105352212	35	3	27	141	8	111	160	44	16	1519	255	63	167	31
49	5,6095037384	39	14	36	140	29	111	146	43	12	1488	268	63	167	47
50	5,6094203722	25	1	34	133	12	110	164	44	22	1484	277	67	167	47
51	5,6070319201	50	3	29	130	5	109	156	44	19	1554	240	63	144	23
52	5,6080545541	36	1	31	140	12	105	162	44	18	1494	262	66	160	40
53	5,6078849308	37	2	23	130	19	109	164	44	26	1500	264	63	166	35
54	5,6063507422	43	6	30	127	6	109	165	45	12	1498	272	62	151	32
55	5,6045195785	42	5	33	123	10	108	160	44	40	1527	283	64	177	28
56	5,6037620397	49	3	32	144	10	110	144	44	17	1517	238	65	158	29
57	5,6028098355	42	13	30	132	30	110	155	44	14	1502	236	65	170	18
58	5,6013037295	33	5	35	123	26	112	164	45	8	1495	234	64	171	26
59	5,6001407622	48	5	24	139	14	115	168	44	10	1466	258	61	156	14
60	5,5994035903	66	3	29	130	27	107	148	45	10	1513	230	63	153	13
61	5,5982818381	18	8	21	130	4	111	178	44	25	1487	267	68	166	25
62	5,5963749281	55	4	26	133	2	101	150	44	13	1517	249	72	157	24
63	5,5957663871	43	15	20	114	1	103	159	45	48	1506	257	70	149	15
64	5,5920207013	44	17	32	109	1	108	168	44	25	1565	203	59	159	14
65	5,5956936968	58	1	34	134	5	100	148	44	30	1527	261	59	144	21
66	5,5918735807	39	19	31	122	9	106	138	45	20	1482	280	69	180	50
67	5,5873907226	39	14	45	119	9	105	133	42	1	1505	295	64	178	51
68	5,5880451723	16	6	23	138	9	109	159	43	22	1594	225	59	174	32
69	5,5905027191	49	13	37	102	11	105	155	44	18	1540	273	64	163	15
70	5,5914278910	26	1	26	149	18	113	146	43	29	1492	249	65	179	42
71	5,5840028482	51	8	22	121	60	113	161	43	7	1476	220	59	137	25
72	5,5858380308	30	1	29	142	9	98	166	43	29	1511	257	71	130	14
73	5,5855336452	50	1	32	113	21	112	142	43	26	1502	270	58	176	13
74	5,5809965772	35	3	35	147	56	120	135	43	25	1504	212	56	156	17
75	5,5791797200	63	13	54	111	2	104	147	44	45	1532	198	60	125	10
76	5,5808070923	15	8	19	148	60	105	168	43	7	1586	241	62	116	20
77	5,5792217542	19	1	26	111	13	106	180	45	10	1538	249	71	132	20
78	5,5774448782	46	30	20	114	28	104	143	44	18	1519	294	59	151	9
79	5,5752057080	38	5	34	117	13	114	173	45	29	1418	257	70	140	7
80	5,5736234233	24	4	66	125	44	108	149	44	7	1603	256	72	147	75
81	5,5724751437	16	2	32	143	47	114	155	43	46	1378	281	77	140	18
82	5,5735538190	28	6	9	150	2	115	160	44	51	1518	219	57	120	7
83	5,5660534982	15	18	39	106	75	103	144	46	26	1540	226	60	125	25
84	5,5687629951	20	2	37	121	27	110	167	45	10	1577	215	60	101	18
85	5,5578060301	54	4	25	87	24	106	143	45	79	1459	262	74	148	1
86	5,5677299035	46	14	32	128	1	99	136	44	6	1555	284	70	124	4
87	5,5648849636	69	3	51	124	24	116	143	43	1	1515	201	52	128	37
88	5,5496267516	31	9	1	112	1	100	188	46	8	1664	147	67	97	2
89	5,5523071759	36	1	17	125	8	100	157	45	41	1394	363	67	131	9
90	5,5561557278	45	4	17	128	4	96	184	46	36	1530	159	64	94	9
91	5,5507535521	37	1	28	131	25	119	150	46	23	1404	157	70	174	17
92	5,4632640549	56	20	13	63	1	78	182	45	31	1666	233	57	84	26
93	5,5281762852	20	10	25	136	3	112	161	40	33	1536	307	71	95	12
94	5,5270497856	55	5	20	93	7	95	167	45	26	1699	87	73	72	2
95	5,5342999243	6	1	17	155	13	88	183	43	54	1417	295	66	151	5
96	5,5312671606	25	6	32	122	8	97	150	45	4	1658	136	58	115	37
97	5,5142983554	22	5	6	151	31	115	200	47	3	1243	251	69	151	20
98	5,5281762852	26	12	41	82	3	95	158	46	19	1613	134	71	110	31
99	5,4938847047	31	9	25	104	23	103	212	51	9	1327	182	70	43	20
100	5,4632640549	27	10	12	116	2	80	187	43	13	1781	31	72	22	1

A.6 Lokale Suche selektierte Wertpapiere

	|Wertpapier Nummer:|																																																																																																					
k:	Fitness	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1|3,5924916720|																																												|241|																																																											
2	4,6976929455																																														85																							2721																																	
3	0,0215457752																																												312		67																							2106																																	
4	5,3635491662																																							130					257		61																							1917																																	
5	5,4628588910																											180													135					224		55																							1722																																	
6	0,0059014067																											170													112					222		52																							1677											74																					
7	5,5791447133																											167													114					200		48																							1605	276										72																					
8	5,6059694869																											167													114					200		48																							1605	276										72																					
9	5,6153171150								56																			147													110					170		46																							1537	276										66											167									
10	5,6185205554								56															32				145													111					164		45																							1521	275										64											165									
11	5,6216581955								47															37				145													111					162		44																							1503	279										64											169			46						
12	5,6227069852								42															33				140													110					161		44									27														1501	277										64											167			41						
13	5,6228755617								41															33				141					18							110					159		44									25														1494	275										64											166			40						
14	5,6229810540								41													6			32				140					17							110					158		44									24														1496	275										64											167			39						
15	5,6229677439								40													7			32				140					17							110					158		44									24												1	1495	275										64											166			40						
16	0,0059014067								40													7			32				140					17							110					158		44									24												1	1494	275										64							1				167			40						
17	5,6229241647	1							40													7			32				140					17							110					158		44									25												1	1495	273										64							1				166			39						
18	5,6228850558	1						1	40													6			32				140					17							110					158		44									24												1	1496	274										64							1				167			38						
19	5,6228168648	1						1	40													6			32				140					16							110	1				158		44									24												1	1497	273										64							1				166			38						
20	5,6227063560	1						1	40													6			32				140					16							110	1				158		44									25												1	1493	273										64					1		1				167			39						
21	5,6225799785	1						1	40													6			32				140					16							110	1	1			158		44									24												1	1494	273										64					1		1				167			38						
22	5,6223822832	1						1	40													6			32				140					15							110	1	1			158		44				1						25												1	1494	273										64					1		1				166			38						
23	5,6221751686	1						1	40													6			32				140					17							110	1	1			158		44				1						24												1	1493	273	1									64					1		1				166			38						
24	5,6219492656	1						1	40													6			32				139					16							110	1	1			158		44				1						24												1	1493	274	1		1							64					1		1				166			39						
25	5,6216854382	1						1	40													6			32				139					15							110	1	1			158		44				1						24												1	1495	274	1		1							64					1		1				166			38						
26	5,6214018410	1						1	40			1										6			32				139					15							110	1	1			158		44				1						24												1	1494	273	1		1							64					1		1				166			39						
27	5,6211079639	1						1	40			1					1					6			32				139					15							110	1	1			158		44				1						23												1	1495	272	1		1							64					1		1				166			39						
28	5,6208148733	1						1	40			1					1					6			31				139					15						1	110	1	1			158		44				1						24												1	1494	273	1		1							64					1		1				166			38						
29	5,6205041966	1						1	39		1	1					1					6			32				139					16						1	110	1	1			158		44				1						23												1	1495	272	1		1							64					1		1				165			39						
30	5,6201809853	1						1	39		1	1					1					6			32				139					16						1	110	1	1			158		44				1						23												1	1493	272	1		1							64					1		1		1		166			39						
31	5,6198515811	1						1	39		1	1					1					5			32				139					14						1	110	1	1			158		44				1						24												1	1492	273	1		1							64					1		1		1		166			38		1				
32	5,6194955438	1						1	39		1	1					1		1			5			32				139					15						1	110	1	1			158		44				1						24												1	1492	272	1		1							64					1		1		1		166			37		1				
33	5,6191260988	1						1	40		1	1					1		1			6			32				139					14						1	110	1	1			157		44				1						22												1	1492	272	1	1	1							64					1		1		1		167			38		1				
34	5,6187482740	1						1	40		1	1					1		1			5			32				139					15					1	1	110	1	1			157		44				1						22												1	1492	271	1	1	1							64					1		1	1	1		167			38		1				
35	5,6183538617	1						1	40		1	1					1		1			5			32				138					16					1	1	110	1	1			157		44				1						23												1	1491	271	1	1	1							64					1		1	1	1		166			38		1				
36	5,6179507781	1						1	39		1	1					1		1			5			32				138					17					1	1	110	1	1			157		44				1						23	1											1	1492	271	1	1	1							64					1		1	1	1	1	166			37		1				
37	5,6175360870	1						1	40		1	1					1		1			5			32				138					15					1	1	110	1	1			157		44				1						23	1											1	1492	270	1	1	1							64	1				1		1	1	1	1	165			37		1				
38	5,6171180000	1						1	39		1	1		1			1		1			5			32				138					14					1	1	110	1	1			157		44				1						23	1											1	1493	272	1	1	1							64	1				1		1	1	1	1	166			37		1				
39	5,6166746345	1				1		1	39		1	1		1			1		1			5			32				138					15					1	1	110	1	1			157		44				1						23	1											1	1491	271	1	1	1							64	1				1		1	1	1	1	167			37		1				
40	5,6161891261	1				1		1	39		1	1		1			1		1			4			32				139					16	1				1	1	110	1	1			158		44				1						22	1											1	1488	271	1	1	1							64	1				1		1	1	1	1	166			37		1				
41	5,6157134909	1				1		1	39		1	1		1			1		1			5			32				138					16	1				1	1	110	1	1			157		44				1						23	1			1							1	1490	269	1	1	1							64	1				1		1	1	1	1	166			37		1				
42	5,6152133083	1				1		1	39		1	1		1			1		1			5			32				138					14	1				1	1	110	1	1			157		44				1						23	1		1	1							1	1491	271	1	1	1							64	1				1		1	1	1	1	164			37		1				
43	5,6147079479	1				1		1	39		1	1		1			1		1			5			32				138					14	1				1	1	110	1	1			157		44				1	1					23	1		1	1							1	1493	269	1	1	1							64	1				1		1	1	1	1	165			35		1				
44	5,6142019739	1				1		1	39		1	1		1			1		1			6			31				138					14	1				1	1	110	1	1	1		157		44				1	1					23	1		1	1							1	1492	269	1	1	1							64	1				1		1	1	1	1	165			36		1				
45	0,0059014067	1				1		1	39		1	1		1			1	1	1			5			32				138					13	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1							1	1492	269	1	1	1							64	1				1		1	1	1	1	165			37		1				
46	0,0059014067	1				1		1	39		1	1		1			1	1	1			5			32				138					14	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1							1	1490	270	1	1	1							64	1				1		1	1	1	1	165			36		1			1	
47	5,6126340613	1				1		1	39		1	1	1	1			1	1	1			5			32				137					13	1				1	1	110	1	1	1		157		44				1	1					23	1		1	1							1	1492	267	1	1	1							64	1				1		1	1	1	1	165			36		1			1	
48	5,6120862612	1				1		1	39		1	1	1	1			1	1	1			5			32		1		137					13	1				1	1	110	1	1	1		157		44				1	1					23	1		1	1							1	1492	267	1	1	1							64	1				1		1	1	1	1	164			36		1			1	
49	5,6114895887	1				1		1	39		1	1	1	1			1	1	1			5			32		1		137					13	1				1	1	110	1	1	1		157		44				1	1					23	1		1	1							1	1492	267	1	1	1							64	1				1		1	1	1	1	163		1	35		1			1	
50	5,6108944296	1				1		1	39		1	1	1	1			1	1	1			5			32		1		137			1		12	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1							1	1488	270	1	1	1							64	1				1		1	1	1	1	165		1	36		1			1	
51	5,6102826708	1				1		1	39		1	1	1	1			1	1	1			5			31		1		137			1		12	1				1	1	110	1	1	1		157		44				1	1				1	23	1		1	1							1	1492	267	1	1	1							64	1				1		1	1	1	1	163		1	35		1			1	
52	5,6096712347	1				1		1	39		1	1	1	1			1	1	1			5			32		1		137			1		12	1		1		1	1	110	1	1	1		157		44				1	1				1	22	1		1	1							1	1490	267	1	1	1							64	1				1		1	1	1	1	164		1	35		1			1	
53	5,6090504024	1				1		1	39		1	1	1	1			1	1	1			4			32		1		137			1		11	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1							1	1491	267	1	1	1							64	1				1		1	1	1	1	164		1	35		1			1	
54	5,6084304585	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	137			1		13	1		1		1	1	110	1	1	1		157		44				1	1				1	22	1		1	1							1	1489	267	1	1	1							64	1				1		1	1	1	1	163		1	35		1			1	
55	5,6078079996	1		1		1		1	38		1	1	1	1			1	1	1			5			32		1	1	137			1		12	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1							1	1488	267	1	1	1							64	1				1		1	1	1	1	164		1	36		1			1	
56	5,6071642747	1		1		1		1	39		1	1	1	1			1	1	1			5			32		1	1	136			1		13	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1							1	1487	267	1	1	1		1					64	1				1		1	1	1	1	164		1	34		1			1	
57	5,6065056520	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	137			1	1	12	1		1		1	1	110	1	1	1		156		44				1	1		1		1	23	1		1	1							1	1489	268	1	1	1		1					63	1				1		1	1	1	1	163		1	35		1			1	
58	5,6058373798	1		1		1		1	38		1	1	1	1			1	1	1			5			32		1	1	137			1	1	13	1		1		1	1	110	1	1	1		156		44				1	1		1		1	22	1		1	1							1	1487	267	1	1	1		1	1				64	1				1		1	1	1	1	164		1	35		1			1	
59	5,6051595473	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1				1			1	1487	268	1	1	1		1	1				63	1				1		1	1	1	1	164		1	35		1			1	
60	5,6044277257	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	137			1	1	12	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1				1			1	1487	267	1	1	1		1	1				63	1				1		1	1	1	1	163	1	1	34		1			1	
61	5,6036791456	1		1		1		1	38		1	1	1	1			1	1	1			4			32		1	1	137			1	1	11	1		1		1	1	110	1	1	1		157		44		1		1	1		1		1	22	1		1	1				1			1	1486	268	1	1	1		1	1				63	1				1		1	1	1	1	164	1	1	36		1			1	
62	5,6029147212	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	136			1	1	11	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1			1	1490	266	1	1	1		1	1				63	1				1		1	1	1	1	164	1	1	34		1			1	
63	5,6021410296	1	1	1		1		1	38		1	1	1	1			1	1	1			4			32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1			1	1486	267	1	1	1		1	1				64	1				1		1	1	1	1	163	1	1	35		1			1	
64	5,6013573755	1	1	1		1		1	37		1	1	1	1			1	1	1			5		1	32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1			1	1486	267	1	1	1		1	1				64	1				1		1	1	1	1	162	1	1	36		1			1	
65	5,6005735400	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1		1	1	1484	267	1	1	1		1	1				64	1				1		1	1	1	1	163	1	1	35		1			1	
66	5,5997872567	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136			1	1	11	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1				1		1	1	1484	267	1	1	1		1	1				64	1				1		1	1	1	1	163	1	1	35		1			1	
67	5,5990002646	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136	1		1	1	11	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1				1		1	1	1484	267	1	1	1		1	1				64	1				1		1	1	1	1	163	1	1	34		1			1	
68	5,5981792844	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136	1		1	1	10	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1			1	1		1	1	1484	268	1	1	1		1	1				63	1				1		1	1	1	1	164	1	1	35		1			1	
69	5,5973474171	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	135	1		1	1	9	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	23	1		1	1			1	1		1	1	1488	265	1	1	1		1	1				63	1				1	1	1	1	1	1	163	1	1	33		1			1	
70	5,5964403740	1	1	1		1		1	38		1	1	1	1			1	1	1			5		1	32		1	1	135	1		1	1	9	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1			1	1		1	1	1485	265	1	1	1	1	1	1				64	1				1	1	1	1	1	1	162	1	1	33		1			1	
71	5,5954983634	1	1	1		1		1	37		1	1	1	1			1	1	1			4		1	32		1	1	135	1		1	1	10	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	23	1		1	1			1	1		1	1	1482	265	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	163	1	1	34		1			1	
72	5,5945381036	1	1	1		1		1	37		1	1	1	1	1		1	1	1			4		1	32		1	1	135	1		1	1	10	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	21	1		1	1			1	1		1	1	1484	265	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	163	1	1	34		1			1	
73	5,5935736536	1	1	1		1		1	37		1	1	1	1	1		1	1	1			5		1	32		1	1	135	1		1	1	8	1		1		1	1	109	1	1	1		156	1	44	1	1	1	1	1		1		1	22	1		1	1			1	1		1	1	1487	263	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	163	1	1	34		1			1	
74	5,5925836738	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		5		1	32		1	1	135	1		1	1	6	1		1		1	1	109	1	1	1		156	1	44	1	1	1	1	1		1		1	22	1		1	1			1	1		1	1	1488	264	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	163	1	1	33		1			1	
75	5,5915376255	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	134	1		1	1	8	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1		1		1	22	1		1	1			1	1		1	1	1489	261	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	162	1	1	34		1			1	
76	5,5904690337	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	31		1	1	134	1		1	1	8	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1		1		1	22	1		1	1			1	1		1	1	1489	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	162	1	1	33		1			1	
77	5,5893760517	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	134	1		1	1	6	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1		1	22	1		1	1			1	1		1	1	1484	263	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	164	1	1	34		1			1	
78	5,5882363016	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	134	1		1	1	9	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1		1	21	1		1	1		1	1	1		1	1	1485	260	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	162	1	1	34		1			1	
79	5,5870853469	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		3		1	32		1	1	134	1		1	1	9	1		1	1	1	1	110	1	1	1		155	1	44	1	1	1	1	1	1	1		1	21	1		1	1		1	1	1		1	1	1483	263	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33		1	1		1	
80	5,5859247320	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	31		1	1	133	1		1	1	7	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1		1	1		1	1	1		1	1	1487	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	33		1	1		1	
81	5,5847540174	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		3		1	32		1	1	133	1		1	1	7	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1		1	1		1	1	1		1	1	1483	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	34	1	1	1		1	
82	5,5835570779	1	1	1		1		1	36		1	1	1	1	1		1	1	1	1		3		1	31	1	1	1	133	1		1	1	7	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1		1	1		1	1	1		1	1	1486	263	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1		1	
83	5,5823170632	1	1	1		1		1	36		1	1	1	1	1		1	1	1	1		4		1	31	1	1	1	134	1	1	1	1	6	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	21	1		1	1		1	1	1		1	1	1484	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	34	1	1	1		1	
84	5,5810149331	1	1	1		1		1	36		1	1	1	1	1		1	1	1	1		3		1	32	1	1	1	133	1	1	1	1	5	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1		1	1	1	1	1		1	1	1485	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	34	1	1	1		1	
85	5,5796933039	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		3		1	31	1	1	1	133	1	1	1	1	5	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1	1	1	1	1	1	1		1	1	1484	260	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1		1	
86	5,5782719837	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1		4		1	31	1	1	1	133	1	1	1	1	1	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1	1	1	1	1	1	1		1	1	1487	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	159	1	1	32	1	1	1		1	
87	5,5768467648	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1		3	1	1	32	1	1	1	133	1	1	1	1	5	1		1	1	1	1	109	1	1	1		155	1	44	1	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1		1	1	1483	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	33	1	1	1		1	
88	5,5753907119	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1		3	1	1	31	1	1	1	133	1	1	1	1	4	1		1	1	1	1	109	1	1	1		155	1	44	1	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1		1	1	1486	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	32	1	1	1	1	1	
89	5,5738123135	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1	1	2	1	1	31	1	1	1	132	1	1	1	1	2	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	1	22	1	1	1	1	1	1	1		1	1	1482	263	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	33	1	1	1	1	1	
90	5,5721370623	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	133	1	1	1	1	2	1		1	1	1	1	109	1	1	1	1	156	1	44	1	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1		1	1	1486	258	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	159	1	1	33	1	1	1	1	1	
91	5,5704411799	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1	1	3	1	1	32	1	1	1	132	1	1	1	1	3	1		1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1		1	1	1481	262	1	1	1	1	1	1	1	1		64	1			1	1	1	1	1	1	1	160	1	1	32	1	1	1	1	1	
92	5,5687051197	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1	1	2	1	1	31	1	1	1	132	1	1	1	1	2	1		1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1		1	1	1485	261	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	160	1	1	32	1	1	1	1	1	
93	5,5669503316	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1	1	3	1	1	32	1	1	1	132	1	1	1	1	4	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1		1	1	1481	259	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	160	1	1	32	1	1	1	1	1	
94	5,5650297024	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	133	1	1	1	1	3	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1483	260	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	160	1	1	31	1	1	1	1	1	
95	5,5630065572	1	1	1		1		1	36	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	32	1	1	1	132	1	1	1	1	3	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1484	257	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	158	1	1	31	1	1	1	1	1	
96	5,5609801466	1	1	1		1		1	36	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	132	1	1	1	1	2	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1483	258	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	158	1	1	32	1	1	1	1	1	1
97	5,5588613405	1	1	1	1	1		1	36	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	132	1	1	1	1	1	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1487	257	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	156	1	1	30	1	1	1	1	1	1
98	5,5566528626	1	1	1	1	1		1	37	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	132	1	1	1	1	2	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	1	19	1	1	1	1	1	1	1	1	1	1	1	1479	258	1	1	1	1	1	1	1	1	1	64	1	1		1	1	1	1	1	1	1	159	1	1	30	1	1	1	1	1	1
99	5,5535475277	1	1	1	1	1	1	1	37	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	131	1	1	1	1	2	1	1	1	1	1	1	109	1	1	1	1	154	1	44	1	1	1	1	1	1	1	1	1	19	1	1	1	1	1	1	1	1	1	1	1	1480	259	1	1	1	1	1	1	1	1	1	64	1	1		1	1	1	1	1	1	1	159	1	1	30	1	1	1	1	1	1
100	5,5491912488	1	1	1	1	1	1	1	36	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	131	1	1	1	1	2	1	1	1	1	1	1	109	1	1	1	1	154	1	44	1	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1482	256	1	1	1	1	1	1	1	1	1	64	1	1	1	1	1	1	1	1	1	1	156	1	1	29	1	1	1	1	1	1

A.7 Variable Neighborhood Search selektierte Wertpapiere

		Wertpapiernummer:																																																																																																		
k:	Fitness	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	3,5924916720																																															241																																																						
2	4,6976929455																																															85																								2721																														
3	5,1138471944																																													312		67																								2106																														
4	5,3635491662																																							130					257		61																								1917																														
5	5,4628588910																												180												135					224		55																								1722																														
6	5,5382040010																												170												112					222		52																								1677											74																				
7	5,5791447133																												167												114					200		48																								1605	276										72																				
8	5,6059694869																												163												116					185		46																								1551	299										70											167									
9	5,6153171150								56																				147												110					170		46																								1537	276										66											167									
10	5,6185205554								56														32				145												111					164		45																								1521	275										64											165									
11	5,6216581955								47														37				145												111					162		44																								1503	279										64											169			46						
12	5,6227069852								42														33				140												110					161		44											27													1501	277										64											167			41						
13	5,6228755617								41														33				141				18							110					159		44											25													1494	275										64											166			40						
14	5,6229810540								41													6			32				140				17							110					158		44											24													1496	275										64											167			39						
15	5,6229677439								40													7			32				140				17							110					158		44											24												1	1495	275										64											166			40						
16	5,6229529933								40													6			33				140				17							110					158		44											24												1	1495	274										64							1				166			40						
17	5,6229241647	1							40													7			32				140				17							110					158		44											25												1	1495	273										64							1				166			39						
18	0,0183134439	1						1	40													6			32				140				17							110					158		44											24												1	1496	274										64							1				167			38						
19	5,6228168648	1						1	40													6			32				140				16							110	1				158		44											24												1	1497	273										64							1				166			38						
20	5,6227060748	1						1	40													6			32				140				18							110	1				158		44											23												1	1495	273										64					1		1				166			39						
21	5,6225799785	1						1	40													6			32				140				16							110	1	1			158		44											24												1	1494	273										64					1		1				167			38						
22	5,6223825333	1						1	40													6			32				139				16							110	1	1			158		44				1						24												1	1496	273										64					1		1				167			38						
23	5,6221751686	1						1	40													6			32				140				17							110	1	1			158		44				1						24												1	1493	273	1									64					1		1				166			38						
24	5,6219492656	1						1	40													6			32				139				16							110	1	1			158		44				1						24												1	1493	274	1		1							64					1		1				166			39						
25	5,6216854382	1						1	40													6			32				139				15							110	1	1			158		44				1						24												1	1495	274	1		1							64					1		1			1	166			38						
26	5,6214018410	1						1	40			1										6			32				139				15							110	1	1			158		44				1						24												1	1494	273	1		1							64					1		1			1	166			39						
27	5,6211087323	1						1	39			1					1					6			32				139				16							110	1	1			158		44				1						24												1	1495	272	1		1							64					1		1			1	167			38						
28	5,6208152882	1						1	40			1					1					5			32				139				15						1	110	1	1			158		44				1						24												1	1495	272	1		1							64					1		1			1	165			38						
29	5,6205048991	1						1	40		1	1					1					6			32				139				16						1	110	1	1			157		44				1						24												1	1494	272	1		1							64					1		1			1	166			38						
30	5,6201809853	1						1	39		1	1					1					6			32				139				16						1	110	1	1			158		44				1						23												1	1493	272	1		1							64					1		1		1	1	166			39						
31	5,6198511981	1						1	40		1	1					1					5			32				139				14						1	110	1	1			157		44				1						23												1	1495	271	1		1							64					1		1		1	1	167			38		1				
32	5,6194954246	1						1	40		1	1					1		1			5			32				139				16						1	110	1	1			157		44				1						23												1	1492	272	1		1							64					1		1		1	1	166			38		1				
33	5,6191260988	1						1	40		1	1					1		1			6			32				139				14						1	110	1	1			157		44				1						22												1	1492	272	1	1	1							64					1		1		1	1	167			38		1				
34	5,6187480262	1						1	39		1	1					1		1			5			32				138				16					1	1	110	1	1			157		44				1						23												1	1493	272	1	1	1							64					1		1		1	1	167			38		1				
35	5,6183557537	1						1	39		1	1					1		1			5			32				138				15					1	1	110	1	1			158		44				1						23												1	1492	272	1	1	1							64					1		1	1	1	1	166			37		1				
36	5,6179507781	1						1	39		1	1					1		1			5			32				138				17					1	1	110	1	1			157		44				1						23	1										1	1492	271	1	1	1							64					1		1	1	1	1	166			37		1				
37	5,6175317582	1						1	39		1	1					1		1			5			32				138				15					1	1	110	1	1			158		44				1						23	1										1	1489	271	1	1	1							64	1				1		1	1	1	1	166			38		1				
38	5,6171180000	1						1	39		1	1		1			1		1			5			32				138				14					1	1	110	1	1			157		44				1						23	1										1	1493	272	1	1	1							64	1				1		1	1	1	1	166			37		1				
39	5,6166746345	1				1		1	39		1	1		1			1		1			5			32				138				15					1	1	110	1	1			157		44				1						23	1										1	1491	271	1	1	1							64	1				1		1	1	1	1	167			37		1				
40	5,6161994120	1				1		1	39		1	1		1			1		1			6			32				138				15	1				1	1	110	1	1			157		44				1						22	1										1	1493	270	1	1	1							64	1				1		1	1	1	1	165			37		1				
41	5,6157111175	1				1		1	39		1	1		1			1		1			5			32				138				16	1				1	1	110	1	1			157		44				1						23	1			1						1	1491	271	1	1	1							64	1				1		1	1	1	1	164			36		1				
42	5,6152145305	1				1		1	39		1	1		1			1		1			5			32				138				15	1				1	1	110	1	1			157		44				1						23	1		1	1						1	1491	269	1	1	1							64	1				1		1	1	1	1	166			36		1				
43	5,6147076323	1				1		1	39		1	1		1			1		1			6			32				138				14	1				1	1	110	1	1			157		44				1	1					22	1		1	1						1	1492	269	1	1	1							64	1				1		1	1	1	1	164			37		1				
44	5,6142018234	1				1		1	39		1	1		1			1	1	1			5			32				138				13	1				1	1	110	1	1			157		44				1	1					23	1		1	1						1	1491	269	1	1	1							64	1				1		1	1	1	1	166			36		1				
45	5,6136943118	1				1		1	39		1	1		1			1	1	1			5			32				138				13	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1						1	1492	269	1	1	1							64	1				1		1	1	1	1	165			37		1				
46	5,6131777066	1				1		1	39		1	1		1			1	1	1			5			32				138				14	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1						1	1490	270	1	1	1							64	1				1		1	1	1	1	165			36		1			1	
47	5,6126339796	1				1		1	39		1	1	1	1			1	1	1			5			32				137				13	1				1	1	110	1	1	1		157		44				1	1					23	1		1	1						1	1492	267	1	1	1							64	1				1		1	1	1	1	165			36		1			1	
48	5,6120885692	1				1		1	39		1	1	1	1			1	1	1			5			32		1		138				14	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1						1	1489	269	1	1	1							64	1				1		1	1	1	1	164			36		1			1	
49	5,6114935460	1				1		1	39		1	1	1	1			1	1	1			5			32		1		138				14	1				1	1	110	1	1	1		157		44				1	1					21	1		1	1						1	1489	268	1	1	1							64	1				1		1	1	1	1	165		1	36		1			1	
50	5,6108943478	1				1		1	39		1	1	1	1			1	1	1			5			32		1		137			1		12	1				1	1	110	1	1	1		157		44				1	1					22	1		1	1						1	1488	270	1	1	1							64	1				1		1	1	1	1	165		1	36		1			1	
51	5,6102818188	1				1		1	39		1	1	1	1			1	1	1			5			32		1		137			1		13	1		1		1	1	110	1	1	1		157		44				1	1					22	1		1	1						1	1488	268	1	1	1							64	1				1		1	1	1	1	164		1	36		1			1	
52	5,6096689335	1				1		1	39		1	1	1	1			1	1	1			4			32		1		137			1		12	1		1		1	1	110	1	1	1		157		44				1	1				1	22	1		1	1						1	1488	270	1	1	1							64	1				1		1	1	1	1	164		1	36		1			1	
53	5,6090483987	1				1		1	39		1	1	1	1			1	1	1			4			32		1	1	137			1		13	1		1		1	1	110	1	1	1		157		44				1	1				1	22	1		1	1						1	1489	268	1	1	1							64	1				1		1	1	1	1	163		1	35		1			1	
54	5,6084292979	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1		137			1		13	1		1		1	1	110	1	1	1		157		44				1	1		1		1	21	1		1	1						1	1489	267	1	1	1							64	1				1		1	1	1	1	165		1	35		1			1	
55	5,6078079179	1		1		1		1	38		1	1	1	1			1	1	1			5			32		1	1	137			1		12	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1						1	1488	267	1	1	1							64	1				1		1	1	1	1	164		1	36		1			1	
56	5,6071677797	1		1		1		1	38		1	1	1	1			1	1	1			5			32		1	1	137			1		14	1		1		1	1	110	1	1	1		157		44				1	1		1		1	21	1		1	1						1	1487	267	1	1	1		1					64	1				1		1	1	1	1	164		1	35		1			1	
57	5,6065086516	1		1		1		1	38		1	1	1	1			1	1	1			5			31		1	1	137			1	1	13	1		1		1	1	110	1	1	1		157		44				1	1		1		1	22	1		1	1						1	1488	268	1	1	1		1					64	1				1		1	1	1	1	163		1	34		1			1	
58	5,6058333235	1		1		1		1	39		1	1	1	1			1	1	1			4			33		1	1	137			1	1	11	1		1		1	1	110	1	1	1		157		44				1	1		1		1	21	1		1	1						1	1487	268	1	1	1		1	1				63	1				1		1	1	1	1	164		1	35		1			1	
59	5,6051595008	1		1		1		1	38		1	1	1	1			1	1	1			5			32		1	1	137			1	1	12	1		1		1	1	110	1	1	1		157		44				1	1		1		1	21	1		1	1				1			1	1487	266	1	1	1		1	1				64	1				1		1	1	1	1	162		1	36		1			1	
60	5,6044244981	1		1		1		1	39		1	1	1	1			1	1	1			3			32		1	1	136			1	1	11	1		1		1	1	110	1	1	1		157		44				1	1		1		1	23	1		1	1				1			1	1488	268	1	1	1		1	1				63	1				1		1	1	1	1	163	1	1	35		1			1	
61	5,6036806621	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		156		44		1		1	1		1		1	23	1		1	1				1			1	1489	266	1	1	1		1	1				63	1				1		1	1	1	1	163	1	1	35		1			1	
62	5,6029111703	1		1		1		1	39		1	1	1	1			1	1	1			4			32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1			1	1487	268	1	1	1		1	1				63	1				1		1	1	1	1	163	1	1	35		1			1	
63	5,6021409479	1	1	1		1		1	38		1	1	1	1			1	1	1			4			32		1	1	136			1	1	12	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1			1	1486	267	1	1	1		1	1				64	1				1		1	1	1	1	163	1	1	35		1			1	
64	5,6013583515	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136			1	1	13	1		1		1	1	110	1	1	1		156	1	44		1		1	1		1		1	22	1		1	1				1			1	1484	268	1	1	1		1	1				64	1				1		1	1	1	1	162	1	1	35		1			1	
65	5,5915380921	1	1	1		1		1	38		1	1	1	1			1	1	1			5		1	32		1	1	136			1	1	10	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	23	1		1	1				1			1	1488	264	1	1	1		1	1				64	1				1		1	1	1	1	161	1	1	34		1			1	
66	5,5997883266	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136			1	1	10	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1				1		1	1	1487	265	1	1	1		1	1				64	1				1		1	1	1	1	164	1	1	34		1			1	
67	5,5990001829	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	136	1		1	1	11	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1				1		1	1	1484	267	1	1	1		1	1				64	1				1		1	1	1	1	163	1	1	34		1			1	
68	5,5981804315	1	1	1		1		1	38		1	1	1	1			1	1	1			3		1	33		1	1	136	1		1	1	10	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1			1	1		1	1	1485	267	1	1	1		1	1				63	1				1		1	1	1	1	163	1	1	35		1			1	
69	5,5973473354	1	1	1		1		1	38		1	1	1	1			1	1	1			4		1	32		1	1	135	1		1	1	9	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	23	1		1	1			1	1		1	1	1488	265	1	1	1		1	1				63	1				1	1	1	1	1	1	163	1	1	33		1			1	
70	5,5964399106	1	1	1		1		1	38		1	1	1	1			1	1	1			5		1	32		1	1	135	1		1	1	11	1		1		1	1	109	1	1	1		156	1	44		1	1	1	1		1		1	21	1		1	1			1	1		1	1	1487	264	1	1	1	1	1	1				64	1				1	1	1	1	1	1	163	1	1	34		1			1	
71	5,5954977138	1	1	1		1		1	37		1	1	1	1			1	1	1			4		1	32		1	1	136	1		1	1	9	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	22	1		1	1			1	1		1	1	1482	266	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	163	1	1	34		1			1	
72	5,5945348934	1	1	1		1		1	37		1	1	1	1	1		1	1	1			5		1	32		1	1	135	1		1	1	9	1		1		1	1	110	1	1	1		156	1	44		1	1	1	1		1		1	21	1		1	1			1	1		1	1	1486	264	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	161	1	1	34		1			1	
73	5,5935710144	1	1	1		1		1	38		1	1	1	1	1		1	1	1			5		1	31		1	1	135	1		1	1	9	1		1		1	1	109	1	1	1		156	1	44	1	1	1	1		1		1	22	1		1	1			1	1		1	1	1487	263	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	162	1	1	33		1			1	
74	5,5925785368	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		5		1	32		1	1	134	1		1	1	5	1		1		1	1	109	1	1	1		157	1	44	1	1	1	1		1		1	23	1		1	1			1	1		1	1	1488	263	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	162	1	1	33		1			1	
75	5,5904681975	1	1	1		1		1	36		1	1	1	1	1		1	1	1	1		4		1	32		1	1	135	1		1	1	9	1		1	1	1	1	110	1	1	1		156	1	44	1	1	1	1		1		1	21	1		1	1			1	1		1	1	1485	263	1	1	1	1	1	1	1			64	1				1	1	1	1	1	1	162	1	1	34		1			1	
76	5,5893794591	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	134	1		1	1	8	1		1	1	1	1	110	1	1	1		155	1	44	1	1	1	1		1		1	22	1		1	1			1	1		1	1	1484	263	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	163	1	1	34		1			1	
77	5,5882389116	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	134	1		1	1	8	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1		1	22	1		1	1			1	1		1	1	1484	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	162	1	1	34		1			1	
78	5,5870892502	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	134	1		1	1	7	1		1	1	1	1	110	1	1	1		156	1	44	1	1	1	1	1	1		1	21	1		1	1		1	1	1		1	1	1484	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33		1			1	
79	5,5859121249	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	32		1	1	133	1		1	1	6	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1		1	23	1		1	1		1	1	1		1	1	1486	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	32		1	1		1	
80	5,5859121249	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	31		1	1	133	1		1	1	6	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	23	1		1	1		1	1	1		1	1	1490	260	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	159	1	1	31		1	1		1	
81	5,5847437671	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		3		1	31		1	1	133	1		1	1	3	1		1	1	1	1	109	1	1	1		157	1	44	1	1	1	1	1	1	1	1	22	1		1	1		1	1	1		1	1	1489	260	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1		1	
82	5,5835565766	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		4		1	31	1	1	1	133	1		1	1	4	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	22	1		1	1		1	1	1		1	1	1487	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1		1	
83	5,5823135920	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		3		1	31	1	1	1	133	1	1	1	1	4	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	22	1		1	1		1	1	1		1	1	1488	260	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1		1	
84	5,5810133196	1	1	1		1		1	37		1	1	1	1	1		1	1	1	1		3		1	31	1	1	1	133	1	1	1	1	4	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	22	1		1	1	1	1	1	1	1		1	1	1485	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1		1	
85	5,5796893814	1	1	1		1		1	36		1	1	1	1	1		1	1	1	1		4		1	32	1	1	1	133	1	1	1	1	7	1		1	1	1	1	109	1	1	1		155	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1		1	1	1485	262	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	33	1	1	1		1	
86	5,5782813949	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1		3		1	32	1	1	1	133	1	1	1	1	5	1		1	1	1	1	109	1	1	1		155	1	44	1	1	1	1	1	1	1	1	22	1	1	1	1	1	1	1	1		1	1	1485	260	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	32	1	1	1		1	
87	5,5768466835	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1		3	1	1	32	1	1	1	133	1	1	1	1	5	1		1	1	1	1	109	1	1	1		155	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1		1	1	1483	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	160	1	1	33	1	1	1		1	
88	5,5753902612	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1		3	1	1	32	1	1	1	133	1	1	1	1	5	1		1	1	1	1	109	1	1	1		155	1	44	1	1	1	1	1	1	1	1	22	1	1	1	1	1	1	1	1		1	1	1482	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1	1	1	
89	5,5738196505	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	133	1	1	1	1	3	1		1	1	1	1	109	1	1	1		156	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1		1	1	1482	261	1	1	1	1	1	1	1			64	1			1	1	1	1	1	1	1	161	1	1	33	1	1	1	1	1	
90	5,5721131074	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1	1	2	1	1	33	1	1	1	133	1	1	1	1	6	1		1	1	1	1	110	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1		1	1	1476	263	1	1	1	1	1	1	1			63	1			1	1	1	1	1	1	1	160	1	1	33	1	1	1	1	1	
91	5,5704456925	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	133	1	1	1	1	2	1		1	1	1	1	109	1	1	1	1	156	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1		1	1	1483	260	1	1	1	1	1	1	1	1		64	1			1	1	1	1	1	1	1	160	1	1	32	1	1	1	1	1	
92	5,5687028305	1	1	1		1		1	37		1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	132	1	1	1	1	1	1		1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1		1	1	1487	260	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	158	1	1	32	1	1	1	1	1	
93	5,5669491716	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	132	1	1	1	1	3	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1		1	1	1485	259	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	159	1	1	32	1	1	1	1	1	
94	5,5650254061	1	1	1		1		1	36		1	1	1	1	1	1	1	1	1	1	1	4	1	1	31	1	1	1	132	1	1	1	1	1	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	21	1	1	1	1	1	1	1	1	1	1	1	1485	259	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	160	1	1	31	1	1	1	1	1	
95	5,5630048503	1	1	1		1		1	37	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	132	1	1	1	1	2	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1484	258	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	159	1	1	30	1	1	1	1	1	
96	5,5609685792	1	1	1		1		1	37	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	30	1	1	1	132	1	1	1	1	1	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	22	1	1	1	1	1	1	1	1	1	1	1	1485	258	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	158	1	1	29	1	1	1	1	1	1
97	5,5588666528	1	1	1	1	1		1	36	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	132	1	1	1	1	3	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1484	258	1	1	1	1	1	1	1	1	1	64	1			1	1	1	1	1	1	1	156	1	1	30	1	1	1	1	1	1
98	5,5566539440	1	1	1	1	1		1	36	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	31	1	1	1	132	1	1	1	1	2	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1481	258	1	1	1	1	1	1	1	1	1	64	1	1		1	1	1	1	1	1	1	158	1	1	30	1	1	1	1	1	1
99	5,5535392889	1	1	1	1	1	1	1	36	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	131	1	1	1	1	4	1	1	1	1	1	1	109	1	1	1	1	155	1	44	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1476	258	1	1	1	1	1	1	1	1	1	64	1	1		1	1	1	1	1	1	1	158	1	1	31	1	1	1	1	1	1
100	5,5491908347	1	1	1	1	1	1	1	37	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	32	1	1	1	130	1	1	1	1	1	1	1	1	1	1	1	109	1	1	1	1	154	1	44	1	1	1	1	1	1	1	1	20	1	1	1	1	1	1	1	1	1	1	1	1481	257	1	1	1	1	1	1	1	1	1	64	1	1	1	1	1	1	1	1	1	1	158	1	1	28	1	1	1	1	1	1

Abbildungsverzeichnis

Algorithmusverzeichnis

Literaturverzeichnis

[1] Bauer R. J., *Genetic Algorithms and Investment Strategies*, John Wiley & Sons, Inc. 1994

[2] Baumgartner P., *Vergleich der Anwendung Neuronaler Netze und Genetischer Algorithmen zur Lösung von Problemen der Finanzprognose*, Dissertation, Universität St. Gallen, 1998

[3] Bertraut C.C., Starr-McCluer M., *Household Portfolios in the United States*, The European University Institute's Conference on Household Portfolios, 1999

[4] Black F., *Capital Market Equilibrium with Restricted Borrowing*, Journal of Business, 1972, S. 444-455

[5] Blume M.E., Friend I., *The Asset Structure of Individual Portfolios and Some Implications for Utility Functions*, Journal of Finance 30, 1975, S. 16-40

[6] Börsch-Supan A., Eymann A., *Household Portfolios in Germany*, Arbeitspapier, Universität Mannheim, 2000

[7] Burtscher M., *Portfolio selection with genetic algorithm*, Diplomarbeit, Universität Wien, 2004

[8] Chang T.J., Meade N., J.E. Bealey, Y.M. Sharaiha, *Heuristics for cardinality constrained portfolio optimization*, Computers and Operations Research 27, 2000, S. 1271-1302

[9] Chen S.-H., *Evolutionary Computation in Economics and Finance*, Physica-Verlag Heidelberg, 2002

[10] Chen S.-H., *Evolutionary Computation in Financial Engineering: A Roadmap of GAs and GP*, Financial Engineering News 2, 1998, S. 1-10

[11] Davis L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991

[12] Derigs U., Nickel N.-H., *Meta-heuristic based decision support for portfolio optimization with a case study on tracking error minimization in passive portfolio management*, OR Spectrum 25, 2003, S. 345–378

[13] Fama E.F., *The Behavior of Stock Markets*, Journal of Business, 1965, S. 34-105

[14] Farrel Jr. J.L., *Portfolio Management, Theory and Application*, 2. Auflage, McGraw-Hill, New York et al., 1997

[15] Fernández A., Sergio G., *Portfolio selection using neural networks*, Computers and Operations Research 34, 2007, S. 1177-1191

[16] Fieldsend J., Matatko J., Peng M., *Cardinality constrained portfolio optimisation*, Proceedings of the fifth international conference on intelligent data engineering and automated learning (IDEAL'04), Exeter, 25.–27. August 2004

[17] Foacci F., Laburthe F., Lodi A., *Local Search and Constraint Programming*, in: Handbook of Metaheuristics, ed. Glover F., Kochenberger A.G., Kluwer, Boston et al., 2003, S. 369-404

[18] Fischer E.O., *Finanzwirtschaft für Anfänger*, 2. Auflage, Oldenbourg, München, 1996

[19] Fischer E.O., *Finanzwirtschaft für Fortgeschrittene*, 2. Auflage, Oldenbourg, München, 1996

[20] Fischer E.O., *Kapitalmarktforschung und Investmentanalyse*, 4. Auflage, Universität Wien, Oktober 2001

[21] Gardes I., Klawonn F., Kruse R., *Evolutionäre Algorithmen, Genetische Algorithmen – Strategien – und Optimierungsverfahren – Beispielanwendungen*, 1. Auflage, Friedr. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden, 2004

[22] Gendreau M., *An Introduction to Tabu Search*, in: Handbook of Metaheuristics, ed. Glover F., Kochenberger A.G., Kluwer, Bosten et al., 2003, S. 37-54

[23] Glover F.W., *Future paths for integer programming and links to artificial intelligence*, Computers and Operations Research 13, 5 (May 1986), S. 533–549

[24] Guiso L., Japelli T., Terlizzese D., *Income Risk, Borrowing Constraints and Portfolio Choice*, American Economic Review 86, 1996, S. 158-172

[25] Hamming W.R., *Error-detecting and error-correcting codes*, Bell System Technical Journal 29(2), 1950, S. 147-160

[26] Hansen P., Mladenovic N., *Variable Neighborhood Search*, in: Handbook of Metaheuristics, ed. Glover F., Kochenberger A.G., Kluwer, Boston et al., 2003, S. 145-184

[27] Hansen P., Mladenovic N., *A Tutorial on Variable Neighborhood Search*, 2003

[28] Hillier F.S., Lieberman G.J., *Operations Research: Einführung*, 5. Auflage, Oldenbourg, München/Wien, 1997

[29] Holland J.H., *Adaption in Natural and Artificial Systems*, Ann Arbor, 1975

[30] Holland J.H., *Adaption in Natural and Artificial Systems*, First MIT Press, 1992

[31] Keber C., Schuster M., *Evolutionary Computation and the Vega Risk of American Put Options*, IEEE Transactions on Neural Networks 12, Juli 2001, S. 704-715

[32] Keber C., *Diskrete Portfeuilleoptimierung mit Hilfe Genetischer Algorithmen*, Zeitschrift für Betriebswirtschaft 69, 1999, S. 1025-1051

[33] Keber C., Maringer D.G., *On Genes, Insects, and Crystals: Determining Marginal Diversification Effects With Nature Based Algorithms*, Conference on Evolutionary Computation in Economics and Finance 2001, Yale University, 28.-29. Juni 2001

[34] Kellerer H., D.G. Maringer, *Optimization of Cardinality constrained portfolios with a hybrid local search algorithm*, OR Spectrum 25, 2003, S. 481-495

[35] Kistner K.P., *Optimierungsmethoden: Einführung in die Unternehmensforschung für Wirtschaftswissentschaftler*, 2. Auflage, Physica, Heidelberg, 1993

[36] Kroll Y., Levy H., Markowitz H., *Mean Variance Versus Direct Utility Maximization*, Journal of Finance 39, Mar. 1984, S. 47-61

[37] Konno H., Yamazaki H., *Mean-Absolute Deviation Portfolio Optimization Model and its Applications to Tokyo Stock Market*, Management Science, 1991, S. 519-531

[38] Koza J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992

[39] Levy H., Markowitz H., *Approximating Expected Utility by a Function of Mean and Variance*, American Economic Review 69, Juni 1979, S. 308-317

[40] Lewis A.L., *A simple Algorithm for the Portfolio Selection Problem*, Journal of Finance, 1988, S. 71-82

[41] Loraschi A., Tomassini M., Tettamanzi A., Paolo V., *Distributed Genetic Algorithms with an Application to Portfolio Selection Problems*, in: Artificial Neural Nets and Genetic Algorithms, Proceedings of the International Conference in Alès, ed. Pearson D. W., Steele N. C., Albrecht R. F., France, 1995, S. 384-387

[42] Maringer D.G., *Portfolioselektion mit Tranksaktionskosten und Ganzzahligkeitsbeschränkungen*, Zeitschrift für Betriebswirtschaft 72, 2002, S. 1155-1175

[43] Maringer D.G., *Wertpapierselektion mittels Ants Systems*, Zeitschrift für Betriebswirtschaft 72, 2002, S. 1221-1240

[44] Markowitz H., *The Optimization of a Quadratic Function Subject to Linear Constraints*, Naval Research Logistics Quarterly, Vol. 3, 1956, S. 111-133

[45] Markowitz H., *Portfolio Selection*, Journal of Finance 7, 1952, S. 77–91

[46] Martí R., *Multi-Start Methods*, in: Handbook of Metaheuristics, ed. Glover F., Kochenberger A.G., Kluwer, Boston et al., 2003, S. 355–368

[47] Michaelewicz Z., *Genetic Algorithms + Data Structures = Evolutions Programs*, 2. Auflage, Springer, Berlin et al., 1994

[48] Michaelewicz Z., *Genetic Algorithms + Data Structures = Evolutions Programs*, 3. Auflage, Springer, Berlin et al., 1999

[49] Mladenovic N., *A Variable Neighborhood algorithm-a new metaheuristic for combinatorial optimization*, Abstracts of papers presented at Optimization Days, Montreal 1995, S. 112

[50] Mladenovic N., Hansen P., *Variable neighborhood search*, Computers Operations Research, 24, 1997, S. 1097-1100

[51] Raidl G., *EAlib 1.1 – A Generic Library for Metaheuristics*, Institut für Computer Graphik und Algorithmen, TU Wien, 2004

[52] Raidl G., *Heuristische Optimierungsverfahre*n, http://www.ads.tuwien.ac.at/teaching/ws05/HeuOpt/folien.pdf, Institut für Computer Graphik und Algorithmen, TU Wien, Stand 20. Juni 2006

[53] Rechenberg I., *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart, 1973

[54] Reeves C., *Genetic Algorithms*, in: Handbook of Metaheuristics, ed. Glover F., Kochenberger A.G., Kluwer, Boston et al., 2003, S. 55-82

[55] Steuer R.E., Qi Y., Hirschberger M., *Portfolio Optimization: New Capabilities and Future Methods*, Zeitschrift für Betriebswirtschaft, 2006, S. 199–219

[56] Schaerf A., *Local search techniques for constrained portfolio selection problems*, Computational Economics 20, 2002, S. 90-177

[57] Schuster M.G., *Einsatzmöglichkeiten von genetischen Techniken in der Finanzwirtschaft*, Diplomarbeit, Universität Wien, 1998

[58] Sharpe W.F., *Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risks*, Journal of Finance 19, September 1964, S. 425-442

[59] Solnik B., *Why Not Diversify Internationally Rather Than Domestically*, Financial Analyst Journal 30, 1974, S. 48-54

[60] Speranza M.G., *A Heuristic Algorithm for a Portfolio Optimization Model Applied to the Milan Stock Market*, Computers & Operations Research, 1996, S. 433-441

[61] Streichert F., Ulmer H., Zell A., *Evolutionary algorithms and the cardinality constrained portfolio optimization problem*, in: Selected papers of the international conference on operations research (OR 2003), Heidelberg, 3.–5. September 2003

[62] Tobin J., *Liquidity Preference as a Behaviour toward Risk*, Review of Economic Studies, 1958, S. 251-278

[63] Tobin J., *The Theory of Portfolio Selection*, in: The Theory of Interest Rates, ed. Hahn, F.H., Brechling, F. P. R., London, 1965

[64] Wagner D., *Eine generische Bibliothek für Metaheuristiken und ihre Anwendung auf das Quadratic Assignment Problem*, Diplomarbeit, TU Wien, 2005

[65] Wolpert D.H., Macready W.G., *No Free Lunch Theorems for Optimization*, 1996

[66] Wolpert D.H., Macready W.G., *No Free Lunch Theorems for Search*, 1995

[67] Wolpert D.H., Macready W.G., *On 2-armed Gaussian Bandits and Optimization*, Technical Report SFI-TR-96-03-009, Sana Fe Institute, Santa Fe, New Mexiko, 1996

[68] Xia Y., Liu B., Wang S., Lai K., *A model for portfolio selection with order of expected returns*, Computers & Operations Research 27, 2000, S. 409–422