3 Numerische Methoden

3.1 Der branch-and-bound Algorithmus

3.1.1 Einleitung

Das Problem der Grundzustandssuche der Energiefunktion (2.2) läßt sich mit dem Vektor $\bar{\mathbf{s}}$ der Spinkoordinaten und der Matrix **A** der Wechselwirkungen umschreiben zu

$$H(\mathbf{s}) = -\mathbf{s}^T \mathbf{A} \mathbf{s} \rightarrow \text{Minimum}\,, \tag{3.1}$$

wobei

$$\mathbf{s} = \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} \text{ mit } s_i = \pm 1 \text{ und } A_{ij} = \begin{cases} I_{ij} & \text{für } i \text{ und } j \text{ benachbart} \\ 0 & \text{sonst.} \end{cases} \tag{3.2}$$

Das in [19] auf Grundlage des rekursiven branch-and-bound vorgestellte Programm ermöglicht die Lösung dieser Aufgabe als Spezialfall der allgemeineren Minimierungsaufgabe

$$\mathbf{xAx} + \mathbf{bx} \rightarrow \text{Minimum} \tag{3.3}$$

für diskrete $x_i = \pm 1$ und beliebige, reelle, symmetrische Matrizen A.

Der branch-and-bound Algorithmus selbst zählt prinzipiell zu den Standardmethoden der numerischen Optimierung (vgl. [75]). Er wurde jedoch erstmalig in [19] in einer rekursiven Form zur Minimierung von Hamiltonian für Spingläser etc. verwendet. Im folgenden sollen hier in Kürze noch einmal die Grundprinzipien der Methode unter Berücksichtigung der speziellen Aufgabenstellung erläutert werden. Weiterhin werden einige Verfeinerungen des Verfahrens und Verbesserungsansätze diskutiert sowie der numerische Aufwand abgeschätzt.

3.1.2 Grundlegende Prinzipien der Methode

Im allgemeinen besteht die Verfahrensweise beim branch-and-bound darin, den Baum der möglichen Zustände eines betrachteten Systems nach dem globalen Minimum der jeweiligen Gütefunktion zu durchsuchen. Dabei wird in jedem Knoten des Baums geprüft, ob der zugehörige Teil des Zustandsbaums noch weiter durchsucht werden muß oder ‘abgeschnitten’ werden kann. Dies geschieht durch die Berechnung geeigneter Schranken für die zu minimierende Funktion (Energie- oder Gütefunktion).

Im Falle eines Ising-Spinglases mit N Spins entspricht der Zustandsbaum einem binären Baum mit N Verzweigungsebenen (siehe Abb. 3.1). Das Durchsuchen dieses Baumes von oben nach unten führt auf ein sukzessives Hinzufügen der einzelnen Spins. Jeder Endpunkt des Zustandsbaums repräsentiert jeweils genau eine Spinkonfiguration des Systems.

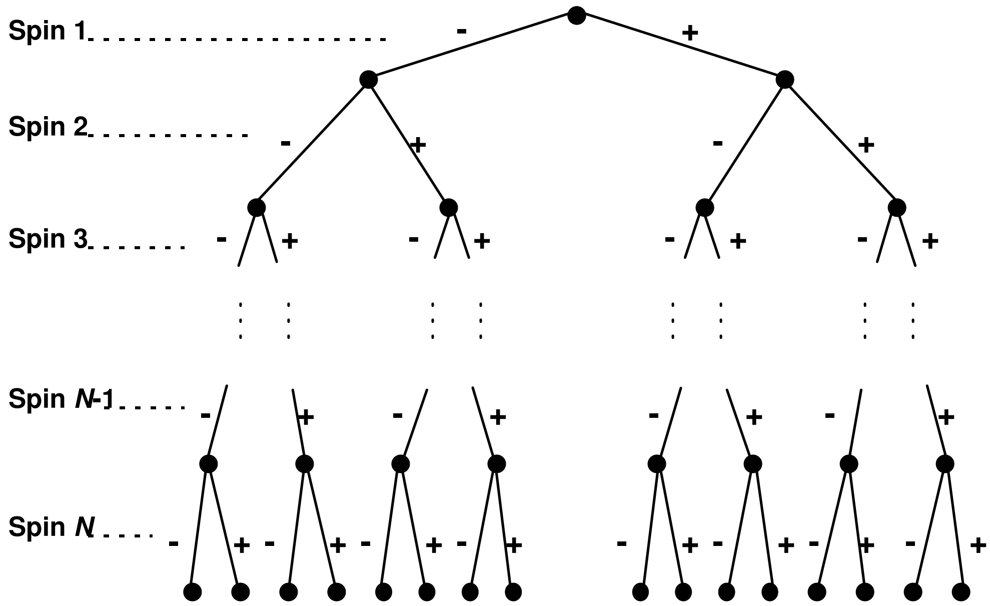

Abbildung 3.1 Schema des binären Zustandsbaums eines Ising-Spin-Systems

Die Aufgabe besteht nun darin, alle Endpunkte zu finden, die ein globales Minimum der Energiefunktion darstellen. Dies geschieht, indem man in jedem noch zu verzweigenden Knoten des Baums eine untere Schranke für den darunterliegenden Teilbaum errechnet und mit der besten schon gefundenen Lösung des Problems vergleicht. Liefert die errechnete untere Schranke eine höhere Energie, so braucht dieser Teilbaum nicht weiter untersucht zu werden. Der Algorithmus funktioniert um so besser, je besser die schon gefundene Lösung und je schärfer die unteren Schranken in den einzelnen Knoten des Baumes sind.

In der Realisierung des Algorithmus nach [19] ordnet man als Abschneide-Schranke jedem Knoten in Abbildung 3.1 einen theoretisch noch erreichbaren minimalen Energiewert zu, indem man annimmt, daß die Wechselwirkungen aller tiefer liegenden Spins optimal befriedigt werden könnten und nur die Wechselwirkungen der höher liegenden Spins untereinander exakt betrachtet. Wenn $E_{k,l}$ den Energiewert des $(l+1)$-ten Knotens von links in der k-ten Ebene (nach Hinzunehmen des k-ten Spins) bezeichnet, so gehört zum ersten Knoten die theoretisch mögliche Minimalenergie (bei Befriedigung aller Wechselwirkungen)

$$E_{0,0} = -\sum_{i,j} |I_{ij}| . \tag{3.4}$$

Da der erste Spin beliebig gewählt werden kann, gilt $E_{1,0} = E_{1,1} = E_{0,0}$. Bezieht man nach und nach die Wechselwirkungen der einzelnen Spins exakt in die Betrachtung ein, so erhöht sich diese Energie. Man kann zeigen, daß stets

$$E_{k,l} \leq \min(E_{k+1,2l}, E_{k+1,2l+1}) \qquad (3.5)$$

gilt, da durch Hinzunehmen eines weiteren Spins die Anzahl der vorhandenen unbefriedigten Bindungen höchstens wachsen kann. Somit stellt dieser theoretische Energiewert zu einem Knoten eine untere Schranke für die Energiewerte des gesamten darunter liegenden Teilbaums dar.

Zur Gewinnung einer guten Anfangslösung des Problems als Vergleichswert für die unteren Schranken in den einzelnen noch zu verzweigenden Knoten ist natürlich jedes der üblichen Näherungsverfahren denkbar. Die in [19] dazu verwendete rekursive Variante des branch-and-bound Algorithmus, benutzt die optimale Lösung des Teilproblems der ersten k Spins um durch Hinzunehmen des nächsten Spins eine gute Anfangslösung des Problems der ersten $k+1$ Spins zu finden. Damit ist es möglich, eine exakte Lösung für das Problem mit $k+1$ Spins zu finden. Durch Fortsetzung der Rekursion bis zur Berücksichtigung aller Spins erhält man eine exakte Lösung des Gesamtproblems.

Nach nochmaligem Durchsuchen und Abschneiden aller anderen Teilbäume entsprechend der gefundenen Minimalenergie, gewinnt man auch alle anderen Lösungen. Erhöht man beim letzten Schritt die Vergleichsenergie, so werden auch höhere Anregungen des Systems gefunden.

3.1.3 Verbesserungsansätze

Eine Verbesserung bzw. Beschleunigung des in [19] benutzten Programms war zwar nicht Inhalt dieser Arbeit, trotzdem ergaben sich im Laufe der Untersuchungen verschiedene Ansätze zur Optimierung dieser Methode, welche bei der Weiterentwicklung und der praktischen Realisierung des Verfahrens berücksichtigt werden sollten.

Eine grundsätzliche Möglichkeit besteht natürlich in der Anpassung des Programms an die heutige Rechentechnik und entsprechende neue Konzepte wie Vektorisierung und Parallelisierung von Programmen. Der Versuch, den Algorithmus in eine vektorisierbare Form zu bringen, brachte jedoch keine wesentliche Geschwindigkeitssteigerung. Bei der einfachen Parallelisierung erhält man für jeden zusätzlich zu berechnenden Spin des Systems einen Faktor 2 für die Prozessorzahl, was bei z.B. 128 Prozessoren lediglich eine Vergrößerung des Systems um 7 Spins ermöglicht.

Ein Problem der praktischen Realisierung des branch-and-bound stellt die Speicherung der zu den einzelnen Knoten gehörenden Schranken dar. Dies wird im wesentlichen durch den Hauptspeicher des genutzten Rechners beschränkt. Mit den Hauptspeicherkapazitäten der heutigen Computertechnik eröffnet sich somit die Möglichkeit der Implementierung einer speicherintensiven Variante des Algorithmus, bei der alle noch zu verzweigenden Knoten des Zustandsbaums abgespeichert werden und nicht wie in der ursprünglichen Variante jeweils neu errechnet werden müssen. Dieser Verbesserungsansatz wurde bisher jedoch noch nicht umgesetzt.

Die Geschwindigkeit beim Durchsuchen des Zustandsbaums hängt sehr drastisch von der Reihenfolge der Spins beim Hinzufügen zum System ab. Numeriert man z.B die Spins eines kubische Systems entsprechend der beiden Untergitter, so erhält man im Vergleich zur 'üblichen' Numerierung eine Geschwindigkeitserhöhung um etwa den Faktor 10. Weitere Untersuchungen dazu sollten auch die Kenntnis der schon gefundenen Grundzustände in die Umsortierung der Spins einfließen lassen.

Der effektivste Ansatzpunkt für eine Verbesserung des Algorithmus liegt jedoch in einer Verschärfung der unteren Schranken in den einzelnen Knoten des Zustandsbaums. So berücksichtigt die zur Zeit benutzte Vorschrift zur Bestimmung dieser Schranken in keiner Weise die Kenntnis der Frustrationsstruktur des Systems. Da in jeder frustrierten Plaquette eines Systems unabhängig von der jeweiligen Spinkonfiguration stets mindestens eine Wechselwirkung unbefriedigt sein muß, läßt sich schon die (3.4) gegebene ideale Grundzustandsenergie entsprechend verschärfen.

3.1.4 Numerischer Aufwand

Die mathematisch exakte Ermittlung der Komplexität des Algorithmus ist recht kompliziert. Daher sollen hier nur grobe Abschätzungen anhand der durchgeführten Rechnungen für den Spezialfall eines quadratischen bzw. kubischen Grundgitters behandelt werden.

Der Aufwand (oder die Rechenzeit) t_{run} für einen Enumerationsalgorithmus steigt unabhängig von der Verteilungsfunktion der Wechselwirkungen exponentiell mit der Größe des Konfigurationsraumes. Für Ising-Spins ergibt dies $t_{\text{run}} \sim 2^N$. Da der Algorithmus prinzipiell nicht aus der Klasse der NP-Probleme herausführt, ist also ein Aufwand, der mit $t_{\text{run}} \sim 2^{\alpha N}$ skaliert, zu erwarten. Abbildung 3.2 zeigt die benötigte CPU-Zeit zur Berechnung aller Grundzustände von Systemen mit $p_{\text{afm}} = 0.5$ in Abhängigkeit von der Systemgröße.

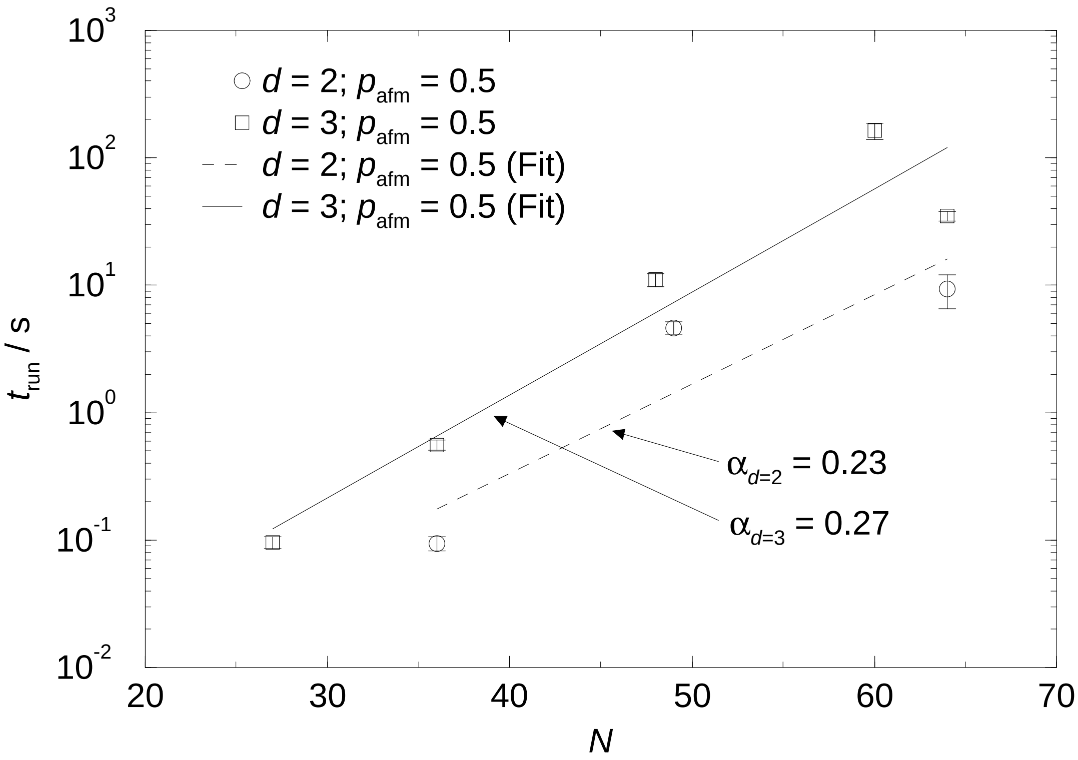

Abbildung 3.2 Abhängigkeit der mittleren CPU-Zeit t_{run} von der Systemgröße N und der Dimension d des Problems

Der ermittelte Exponent α liegt sowohl für $d = 3$ als auch für $d = 2$ bei etwa 0.25. Dies entspricht bei vergleichbarem Aufwand der Berechnung von Systemen der vierfachen Größe im Vergleich zur einfachen Enumeration. Natürlich müssen bei einer genaueren Gegenüberstellung der beiden Verfahren noch die entsprechenden skalaren Vorfaktoren berücksichtigt werden.

Da der Algorithmus des branch-and-bound alle Grundzustände eines Systems findet, sollte der numerische Aufwand mit der Zahl dieser Zustände korrelieren. Die Grundzustandsentartung ist jedoch stark von der Konzentration p_{afm} der antiferromagnetischen Bindungen abhängig (vgl. Kapitel 4.3). Die Abbildungen 3.3 und 3.4 zeigen die Abhängigkeit der CPU-Zeit von der Grundzustandsentropie für verschiedene Konzentrationen p_{afm}. Die Laufzeit hängt bei ein und derselben Systemgröße für dreidimensionale Systeme in erster Näherung exponentiell von der Grundzustandsentropie des Gesamtsystems S_0 ab (Abb. 3.3). Bei den zweidimensionalen Systemen (Abb. 3.4) treten bei geringer Entartung Abweichungen von diesem exponentiellen Verhalten auf.

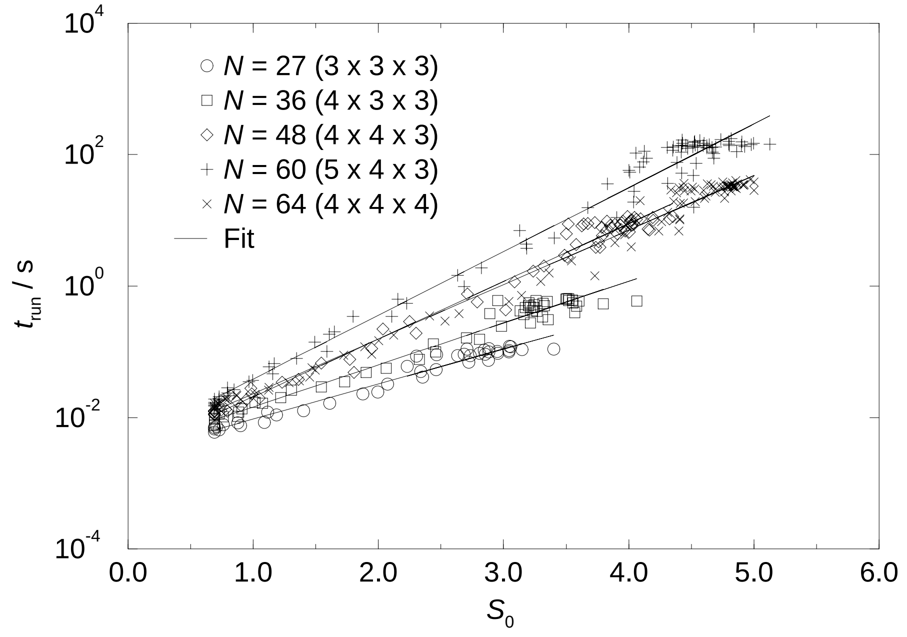

Abbildung 3.3 Abhängigkeit der CPU-Zeit von der Grundzustandsentropie für dreidimensionale Systeme verschiedener Größe.

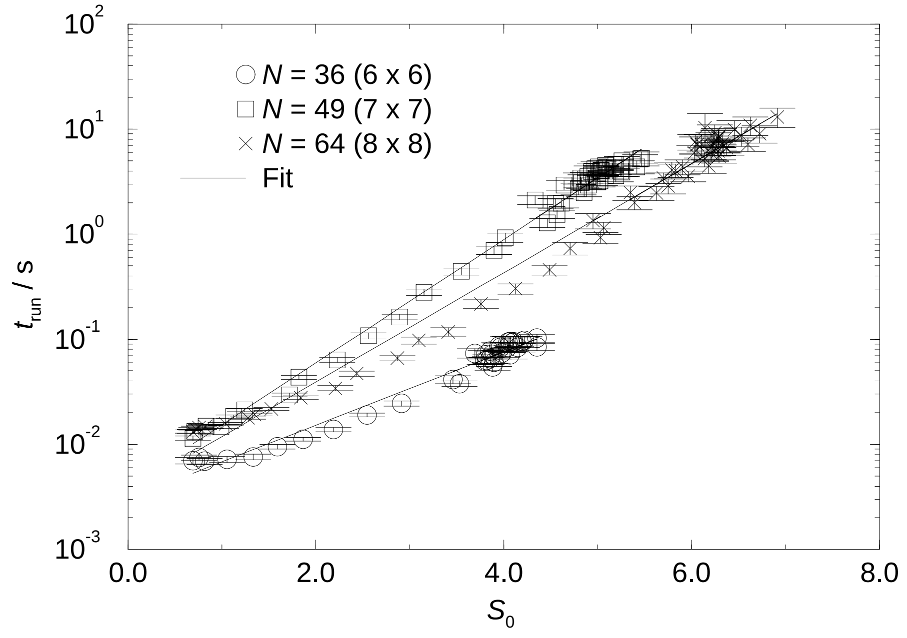

Abbildung 3.4 Abhängigkeit der CPU-Zeit von der Grundzustandsentropie für zweidimensionale Systeme verschiedener Größe.

3.2 Erzeugung und Manipulation der Systeme

Zur Generierung der Systeme wurden Wechselwirkungsmatrizen mit zufälligen Wechselwirkungen ±1 für nächste Nachbarn erzeugt. Dabei können die Ausdehnung des Gitters in alle drei Raumrichtungen, die Randbedingungen und der Prozentsatz p_{afm} der antiferromagnetischen Wechselwirkungen variiert werden. Es wurde der Zufallszahlengenerator aus der Engeneering and Scientific Subroutine Library (ESSL) [76] benutzt. Um den Einfluß von Korrelationen zu verringern wurde nur jede zweite erzeugte Zufallszahl verwendet.

Durch nachträgliches, zufälliges Klappen von Bindungen wurde der Anteil der antiferromagnetischen Bindungen exakt auf die Wahrscheinlichkeit p_{afm} korrigiert. Dabei wurden auch anisotrope Verteilungen zugelassen. Die Speicherung des Anfangswertes für den Zufallszahlengenerator ermöglicht eine spätere Rekonstruktion der Wechselwirkungsmatrix, ohne die gesamte Matrix speichern zu müssen.

3.3 Numerik der Analyse der Konfigurationsraumstruktur

Zur Analyse der Struktur des Konfigurationsraums werden Spinkonfigurationen der niederenergetischen Anregungen in mikrokanonische Cluster des Konfigurationsraums sortiert. Diese Sortierung ist momentan für hohe Anregungen zeitaufwendiger als die Berechnung der tiefliegenden Anregungen selbst und ihrer Spinkonfigurationen.

Zunächst werden die vom branch-and-bound Programm gefundenen Zustände binär kodiert und zusammen mit den zugehörigen Energien zur weiteren Verarbeitung abgespeichert. Entsprechend der interessierenden Größe werden die gefundenen Konfigurationen dann sortiert. Die Sortierung kann z.B. nach der Energie oder nach der Anzahl der nächsten Nachbarn mit gleicher Energie erfolgen. Im weiteren wird von einer Sortierung nach Energien ausgegangen.

Zur Analyse der sortierten Daten nach Clustern im Konfigurationsraum werden diese zuerst linear durchlaufen. Dabei wird die jeweils aktuelle Spinkonfiguration auf Nachbarschaft zu den nächsten n Konfigurationen untersucht. Die Reihenfolge der Konfigurationen wird anschließend entsprechend der schon gefundenen Cluster umsortiert. Die Information über diese Cluster, wie erste Spinkonfiguration, Größe des Clusters und vor allem die im gesamten Cluster unveränderlichen Spins, werden in eine zusätzliche Beschreibungsdatei eingetragen. Diese Vorsortierung hat den Vorteil, daß sie linear von der Anzahl der Konfigurationen abhängt. Sie berücksichtigt außerdem, daß die vom branch-and-bound Algorithmus nacheinander gefundenen Konfigurationen oft auch benachbart sind.

Nach der Vorsortierung erfolgt die exakte Sortierung unter Einbeziehung von notwendigen Bedingungen an die festen Spins von benachbarten Clustern. Damit erhält man mikrokanonische Cluster. Kombiniert man die Cluster von Grundzustand und erster Anregung, so kann durch nochmalige exakte Sortierung die Talstruktur bezüglich der ersten Anregungsenergie ermittelt werden. Durch Einbeziehen der höheren Anregungen kann auch für diese Anregungsenergien die Talstruktur ermittelt werden. Da für die exakte Sortierung im Extremfall alle vorsortierten Cluster mit allen Spinkonfigurationen verglichen werden müssen, wächst der Aufwand quadratisch mit der Anzahl der Konfigurationen, wodurch die maximal untersuchbare Anregungsenergie stark begrenzt wird.