SOS Modelle und ihre Behandlung mittels minimum-cost flow Algorithmen

Björn Naundorf

November 1999

1 Physikalische Problemstellung

Das Solid-on Solid Modell, welches im folgenden diskutiert wird, ist ein mikroskopisches Modell, welches die Anlagerung von Atomen auf einer kristallinen Oberfläche beschreibt. Dabei soll die Höhe n_i der aufeinander gelagerten Atome durch eine ganze Zahl gegeben sein.

Zusätzlich soll als Randbedingung gelten, daß die Oberfläche eine Rauhigkeit aufweist, d.h. jeder Gitterplatz besitzt einen Offset d_i im Intervall $[-0.5, 0.5]$.

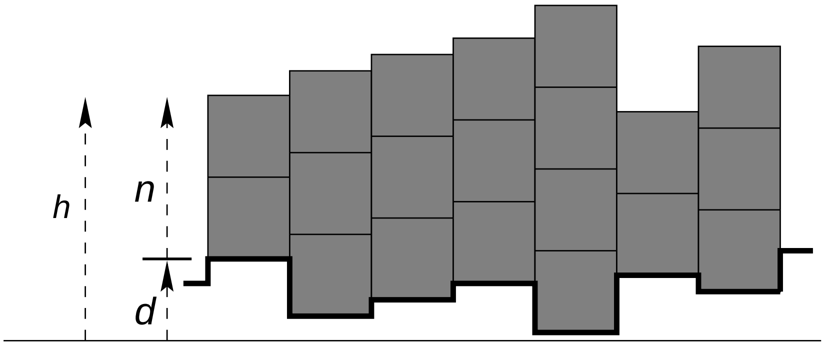


Abbildung 1: Eindimensionale Darstellung des Wachstums

Die Art des Wachstum wird dann bestimmt durch eine Energiefunktion (Hamiltonian):

$$\begin{aligned} H &= \sum_{(ij)} \theta((d_i + n_i) - (d_j + n_j)) \\ &= \sum_{(ij)} \theta(d_{ij} + x_{ij}) \end{aligned}$$

mit $d_{ij} := d_i - d_j$ und $x_{ij} := n_i - n_j$.

Also der Summe über nächste Nachbarn (ij) einer konvexen symmetrischen Funktion θ mit der Differenz der Höhen $d_i + n_i$ als Argument. Gesucht ist die Konfiguration $\{n_i\}$ mit minimaler Energie, also der Grundzustand.

2 Verwandte Systeme

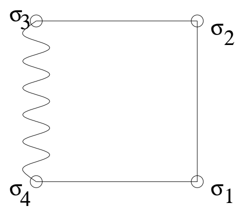
Motiviert wird die Betrachtung dieses Systems auch durch ein Modell, welches die Verteilung von Flußlinien in Hochtemperatursupraleitern beschreibt. dem zweidimensionalen Sine-Gordon-Modell mit zufälliger Phasenverschiebung, da dieses ein ähnliches Symmetrieverhalten aufweist.

Nah beim Phasenübergang sollten beide das gleiche kritische Verhalten zeigen.

3 Frustrierte Systeme

Das SOS-Modell gehört zur Klasse der sogenannten frustrierten Systeme.

Ein frustriertes System kann aufgrund von Randbedingungen seine minmale Energiekonfiguration lokal nicht annehmen. Ein Beispiel ist ein Ising-Spinsystem:



$$H = -\sigma_1\sigma_2 - \sigma_2\sigma_3 - \sigma_3\sigma_4 + \sigma_4\sigma_1$$

Es ist nicht möglich an jedem Platz eine optimale Spinkonfiguration zu verwirklichen.

Wie hier am Beispiel eines diskreten Systems führt Frustration häufig zu vielfacher Entartung eines Energiezustandes. Aus diesem Grund ist es praktisch nicht möglich einen Tieftemperatur-Gleichgewichtszustand mittels Monte-Carlo-Simulation zu berechnen.

4 Warum Grundzustandsberechnung ?

Es stellt sich natürlich die Frage, welche Relevanz eine Berechnung des Grundzustandes hat, den das System bei $T = 0$ annimmt. Das Vorgehen läßt sich durch folgende Argumente motivieren:

- Der **exakte** Grundzustand ist besser als schlechte Tieftemperatur Simulation.
- Einige Eigenschaften des Grundzustandes bleiben bei endlichen Temperaturen erhalten, z.B Korrelationen
- Wenn ein Phasenübergang in einen geordneten Zustand bei $T = 0$ stattfindet, lassen sich kritische Exponenten bestimmen.
- Überprüfung von Rechnungen bei endlichem T mit anderen Methoden. (z.B. Renormierungsgruppen). Wenn es einen Fixpunkt bei $T = 0$ gibt, dann lassen sich sogar kritische Exponenten für endliche Temperaturen bestimmen.

Im folgenden Abschnitt wird auf die mathematische Umformulierung des Problems eingegangen.

5 Das SOS-Modell als Graph

Um das Optimierungsproblem besser beschreiben zu können und damit auch einen Algorithmus zu formulieren, wird das Modell zunächst in folgender Weise auf einen Graphen abgebildet.

Zunächst wird jeder Gitterplatz, auf dem sich Atome anlagern, zu ein Vertex. Die benachbarten Gitterplätze, mit denen eine Wechselwirkung stattfinden soll, werden mit diesem durch eine Kante verbunden.

Im nächsten Schritt wird jede Kante mit einem gerichteten Fluß versehen mit dem Betrag der Höhendifferenzen der zwei verbundenen Gitterplätzen und mit der Richtung von der größeren zur kleineren Höhe.

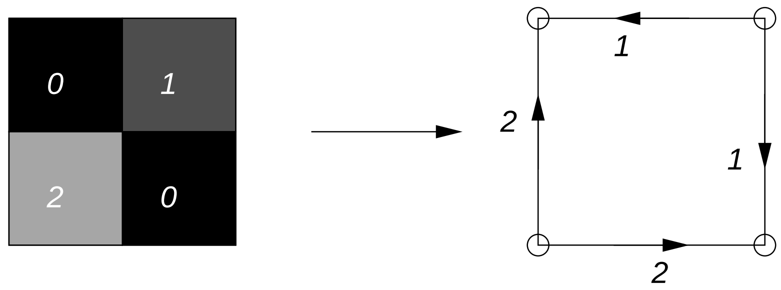


Abbildung 2: Übersetzung des SOS-Modells in einen Graphen. Die Zahlen in den Plaketten geben die Höhe an, die Zahlen an den Pfeilen die Flußstärke.

Zu bemerken ist, daß in jedem geschlossenen Umlauf die Summe der gerichteten Flüsse der durchlaufenen Kanten gleich Null ist (Flüsse, die in Gegenrichtung durchlaufen werden, zählen negativ). Dies ist sofort klar, wenn man daran denkt, daß die Flüsse eine Höhendifferenz beschreiben.

Bezeichnet man die Höhendifferenz mit x_{ij} und die Differenz der Oberflächenoffsets mit $d_{ij} = d_i - d_j$, so läßt sich die Hamiltonfunktion schreiben als:

$$H(X) = \sum_{(ij)\in E} \theta(d_{ij} + x_{ij})$$

Hierbei bezeichnet E die Kantenmenge des Graphen.

6 Der Dualgraph

Um das Optimierungsproblem auf die große Klasse der Netzwerkflußsysteme abzubilden, ist es zweckmäßig, den oben eingeführten Graphen in seinen zugehörigen Dualgraphen zu übersetzen. Dieser wird folgendermaßen konstruiert:

In jedes Gebiet des ursprünglichen Graphen wird ein neuer Vertex gelegt. Dann wird für jede Kante des ursprünglichen Graphen eine neue, die alte schneidende Kante mit demselben Fluß gelegt. Bei der Flußrichtung besitzt man eine globale

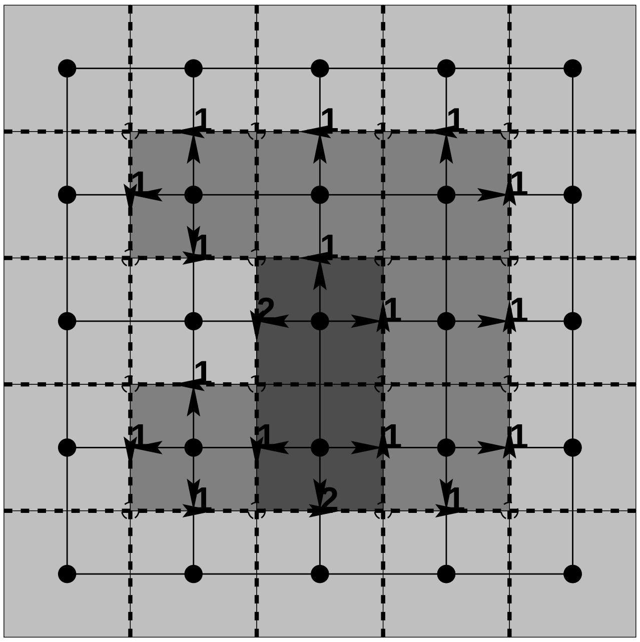


Abbildung 3: Das SOS-Modell als Graph (durchgezogene Linien) und dem Dualgraphen (gestrichelt).

Freiheit, die man jedoch für jede Kante konsistent wählen muß. (z.B. alle Flußrichtungen gehen aus einer Linksdrehung der ursprünglichen Richtung hervor.)

Durch diese Konstruktion wird die Potentialzwangsbedingung ($\sum x_{ij} = 0$ für eine geschlossenen Umlauf) zu einer Stromerhaltung an jedem Vertex.

Zusammengefaßt sucht man also eine Lösung, die folgende Bedingungen erfüllt:

1. Die Lösung soll optimal sein, d.h, daß $H(X) = \sum_{(ij)\in E}(d_{ij} + x_{ij})^2$ minimal.
2. Die einzelnen Höhen sollen ganzzahlig sein, also gilt für die Flüsse $x_{ij} \in Z$ für alle Kanten $(ij) \in E$.
3. Es soll der Strom an jedem Vertex erhalten sein: $\sum_{(ij)} x_{ij} - \sum_{(ji)} x_{ji} = 0$

Ein Problem dieser Art bezeichnet man auch als **minimum-cost-flow** Problem mit Massen(Fluß-)erhaltung. Der Umweg über den Dualgraphen gestattet es nun auf einen eleganten Algorithmus aus der Netzwerktheorie zurückzugreifen.

7 Bemerkungen zur Implementation

Zur Lösung des Minimierungsproblems bieten sich zwei Algorithmen an, die nach zwei unterschiedlichen Prinzipien arbeiten.

Beim ersten, dem sogenannten **negative cycling canceling** Algorithmus, sind in jedem Schritt die Zwangsbedingungen erfüllt, während iterativ die Energie verkleinert wird. Zu diesem Zweck werden zu Anfang alle x_{ij} auf 0 gesetzt, und danach Bereiche im Dualgraphen in der Höhe kollektiv erhöht, bzw. erniedrigt.

Dieser Algorithmus hat den Nachteil, daß der Rechenaufwand nicht polynomial ist.

Beim zweitem, dem **successive shortest path** Algorithmus, startet man mit einer optimalen Lösung für $H(\mathbf{x})$, die aber i.A. die Stromerhaltung an mehreren Vertizes verletzt. Diese Verletzungen werden iterativ durch Vergrößern bzw. Verkleinern des Flusses an diesen Vertizes aufgelöst. Der Algorithmus besitzt den Vorteil, daß der Rechenaufwand polynomial ist.

Im folgenden soll dieser weiter diskutiert werden.

8 Successive Shortest Path Algorithmus

Zunächst führt man den Begriff der **dual feasability** ("Gegenseitigen Verträglichkeit") ein:

Definition: Ein Vektor $X = (x_{ij})$ heißt dual feasible, wenn es eine ganze Zahl π_i für jeden Knoten gibt, sodaß die reduzierten Kostenungleichungen erfüllt sind:

$$\begin{aligned} c_{ij}^{\pi} &:= \theta_{ij}(x_{ij}+1) - \theta_{ij}(x_{ij}) - \pi_i + \pi_j \geq 0 \\ c_{ji}^{\pi} &:= \theta_{ij}(x_{ij}-1) - \theta_{ij}(x_{ij}) + \pi_i - \pi_j \geq 0 \end{aligned}$$

mit $\theta_{ij}(x_{ij}) = (x_{ij} + d_{ij})^2$.

Bemerkung: Für $\pi = 0$ gibt c_{ij}^{π} an, was es kostet, den Fluß an einer Kante um den Wert 1 zu erhöhen, bzw. zu erniedrigen.

Theorem: Sei X ein dual feasible Vektor und X_{opt} die optimale Lösung des zu lösenden Flußproblems, dann ist $H(X) \leq H(X_{\mathrm{opt}})$.

Korollar: Ein dual feasible Vektor X, der die Zwangsbedingungen erfüllt, ist eine optimale Lösung.

Der **successive shortest path** Algorithmus erfüllt in jedem Schritt die reduzierten Kostenungleichungen und versucht dabei die Randbedingungen zu erfüllen:

procedure successive shortest path
  begin
    Let X be a solution for which $H(X)$ minimal
    $\pi = 0$
    while there is a node $g_s(X) > 0$
      compute the reduced costs $c^{\pi}(X)$
      determine the shortest path distances $d(i)$ from

s to all other nodes with respect to the
reduced costs
choose a node t with $g_t(X) < 0$
let $w(s,t)$ denote the shortest path vector from
s to t in the dual lattice
increase x_{ij} by 1 along $w(s,t)$
$\pi = \pi - d$
end
end

Der Algorithmus startet mit dem Potential $\pi = 0$ und der optimalen Lösung für H.

Diese Anfangskonfiguration gewährleistet, daß die Kostenungleichungen erfüllt sind, da für mindestens einen Summanden in H eine Änderung des Flusses zu einer Erhöhung von H führt,

Um die Stromerhaltung zu erfüllen wird, wird in jedem Durchlauf ein Knoten mit $g_s(X) = \sum_{j=\text{Nachbar von s}} x_{sj} > 0$ (supply node) und $g_t(X) < 0$ (demand node) gesucht.

Von der supply node wird nun mit Hilfe eines kürzesten Weg-Algorithmus (siehe nächster Abschnitt) die Entfernung $d(j)$ von s zu allen anderen Knoten im Graphen bestimmt. Entlang des kürzesten Weges von s nach t wird nun der Fluß um den Wert 1 erhöht.

Das folgende wesentliche Theorem gewährleistet die Optimalitätsbedingung:

Theorem:

1. Mit dem Potential $\pi' = \pi - d$ sind ebenfalls die reduzierten Kostenungleichungen erfüllt.
2. $c_{ij}^{\pi'}$ erfüllt nach der Flußänderung ebenfalls die reduzierten Kostenungleichungen.

Bei der zweiten Bedingung geht wesentlich ein, daß θ konvex ist.

Der Algorithmus wird nun iteriert und beendet, wenn alle supply- und demand Knoten ausgeglichen sind. Am Ende liegt dann automatisch eine optimale Lösung vor.

Bemerkung: Um wieder den ursprünglichen Graphen zu rekonstruieren und damit die Lösung für das SOS-Modell zu erhalten nutzt man, daß der Dualgraph des Dualgraphen wieder den ursprünglichen Graphen liefert.

9 Kürzester Weg Algorithmus

Der **successive shortest path** Algorithmus hängt wesentlich davon ab, den kürzesten Weg zwischen einem vorgegebenen Knoten s zu allen anderen zu finden. Dies liefert der folgende Algorithmus in polynomialer Zeit.

```
procedure shortest path
    begin
        S := ∅, S̄ = N
        d(i) := ∞ for each node i ∈ N
        d(s) := 0 and pred(s) := 0
        while |S| < n do
            let i ∈ S̄ be a node for which
                d(i) := min{d(j)|j ∈ S̄}
            S := S ∪ {i}, S̄ := S̄\{i}
            for each (i,j) ∈ A(i) do
                if d(j) > d(i) + c_ij then
                    d(j) := d(i) + c_ij and pred(j) := i
        end
    end
```

Die Knotenmenge N wird in jedem Schritt des Algorithmus in zwei Mengen geteilt. Den permanent mit einer Entfernung versehenen Knoten in einer Menge S und den vorübergehend versehenen in einer Menge $\overline{S} = N \backslash S$. Zunächst werden alle Entfernungen $d(i)$ auf ∞ gesetzt, bzw. $d(s)$ auf 0.

Nun wird ein Knoten i aus der Menge $\overline{S}$ ausgewählt, dessen Entfernungswert $d(i)$ minimal ist und danach in die Menge der permanent mit einer Entfernung versehenen Knoten S übernonommen. Für alle Nachbarknoten wird nun untersucht, ob sich diese weiter als $d(i) + c_{ij}$ von i entfernt sind und gegebenenfalls deren Entfernung korrigiert.

Solange nun die Menge $\overline{S}$ Knoten enthält wird dies iteriert. Am Ende besitzen alle Knoten einen Wert $d(j)$, die kürzeste Entfernung zu s.

Über das Feld $pred(s)$ wird darüberhinaus jedem Knoten ein Vorgängerknoten zugeordnet, über welchen der kürzeste Weg zu s bestimmt werden kann.

Bemerkung: Eine notwendige Bedingung für den Algorithmus ist es, daß alle Kosten positiv sind. (Dies ist wegen der Optimalitätsbedingung stets gewährleistet)

10 Stabilität gegenüber thermischen Fluktuationen

Hat man nun den Grundzustand des SOS-Modells berechnet, so kann man sich für physikalische Eigenschaften interessieren.

Interessant ist z.B. die Stabilität gegenüber thermischen Fluktuationen. Zu deren Untersuchung wählt man das Konzept der **domain wall renormalization**.

Dies ist bekannt aus Untersuchungen von **random spin** Systemen. Dabei wird die Energie berechnet, die eine Treppenstufe der Höhe 1 in einem System mit linearer Größe L kostet. Nimmt nun die Energie mit der Systemgröße L zu, so ist der Grundzustand stabil.

Zunächst wählt man folgende Randbedingungen für das rechteckige Gitter des SOS-Modells:

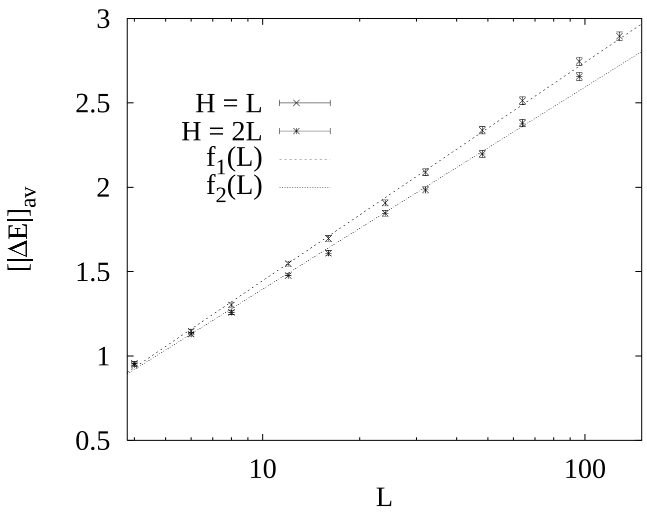
- Die Höhe am unteren Rand des Gitters soll identisch Null sein: $h_{(x,0)} = 0$.
- Am unteren Rand soll es gerade die Höhe 1 besitzen: $h_{(x,L)} = 1$.
- Für die linke, bzw. rechte Seite soll gelten:

$$h_{(0,y)} = h_{(L+1,y)} = \theta(y - L/2)$$

Besitzt das Gitter keine Oberflächenrauhigkeit, also ist $d_i \equiv 0$, so wird $\Delta E = L$, die Konfiguration ist also stabil, da im thermodynamischen Limes $L \to \infty$ die Energie zum spontanen Aufbau einer Stufe sehr groß wird im Vergleich zur thermischen Energie.

11 Computerexperiment

Für eine rauhe Oberfläche wurde die Stufenenergie $[|\Delta E|]_{\text{av}}$ gemittelt über 10^4 Ergebnisse. Als Geometrie wurde ein Rechteck mit der Größe $L \times H$ gewählt.

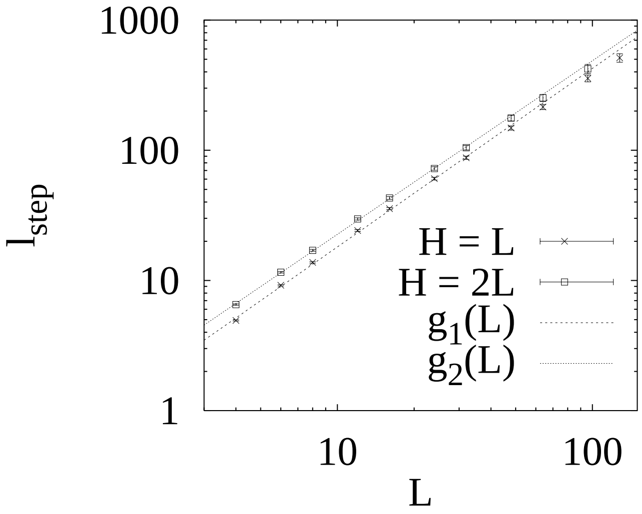


Die Ergebnisse lassen sich gut fitten an:

$$[|\Delta E|]_{\text{av}} \sim a + b \ln L$$

Es stellt sich also heraus, daß auch für eine rauhe Oberfläche die Konfiguration stabil ist.

Ein weiteres bemerkenswertes Ergebnis ist, daß die $1-0$ Stufe, die das Gitter durchquert fraktal ist. Trägt man die Länge der Stufe gegen die Systemlänge L auf, so ergibt sich folgender Zusammenhang:



welcher sich gut fitten läßt an:

$$\mathcal{L}_{\text{step}} = L^{d_{\text{step}}} \qquad \text{mit} \qquad d_{\text{step}} = 1.35 \pm 0.02$$

Bemerkung: Diese Eigenschaft ist universal, auch für verschiedene konvexe Funktionen $\theta(x)$.

Literatur

[1] U. Blasum, W. Hochstättler, C. Moll, H. Rieger, *Using network-flow techniques to solve an optimization problem from surface-physics*, J. Phys. A **29**, Juni 1999

[2] H. Rieger, *Frustrated Systems: Ground State Properties via Combinatorial Optimization*, Lecture given on the Eötvös summer school in Physics, 1996

[3] H. Rieger, U. Blasum, *Ground state properties of solid-on-solid models with disordered substrates*, cond-mat-9608136, August 1996

[4] M. Alava, H. Rieger, *Chaos in random field Ising model*, cond-mat/9804136, August 1998

[5] R. Ahuja, T. Magnanti, J. Orlin, *Network Flows*, Prentice Hall, Ner Jersey, 1993

[6] C. Berge, *Graphs*, North-Holland, Amsterdam, 1985