7ZG-01

アニーリングにおける新たな次元削減手法の性能評価

小津 泰生 †
名古屋大学 †

1 概要

アニーリングは社会における様々な問題に対して適用することができるが、制約充足問題 (CSP) 及び制約最適化問題 (COP) をアニーリングで解く場合、現状のアニーリングマシンではまず問題を二値組み合わせ最適化問題に変換した上で、次元削減をを行うことにより二次制約なし最適化問題 (QUBO) に変換する、という手順を踏む必要がある。この次元削減の方法として、これまで 2 つの変数の組を 1 つの補助変数及び 1 つの制約に置き換える方法がこれまで使われてきた。しかしながら、論的上制約を導入せずに、より一般的な次元削減を行えることが石川によって示されている [1]。本研究では、石川の方法を用いることにより、アニーリング解の精度がどの程度向上するか論し、シミュレーション及び実際のアニーリングマシンを用いて評価する。

2 はじめに

我々の社会における問題の多くは制約最適化問題 (COP) で記述することができ、これは古典的コンピュータでは求解が困難な NP 完全もしくは NP 困難とよばれるクラスに属している。これを解くための新たな手法としてアニーリングという技術が案され、実際に広く用いられている．アニーリングによる求解は，次のような手順により行われる．

1. 解きたい問題を制約および目的関数の組として定式化する．
2. 1. における制約を目的関数の形式に変換する．
3. 1. 及び 2. による目的関数に対してパラメータを掛け足し合わせることで最終的な目的関数を生成する．
4. 3. による目的関数の次数を高々 2 次に削減する．
5. 使用するアニーリングマシンが粗結合の場合、組み込みを行う．
6. 制約条件のパラメータを決定する．
7. シミュレーテッドアニーリング又は量子アニーリングを用いて目的関数の最適解を求める．
8. 7. で得られた解が 1. で定式化した問題の制約を満たすことを確認する．
9. 8. で満たされない場合，パラメータを調整し 6. からやり直す．

この処理を式で表すと次のようになる。まず、1. で定式化される制約最適化問題が

$$(f_{\mathrm{optim}(\mathbf{x})}, \{C_n(\mathbf{x})\}_n)$$

と表されるとする。ただし、$\mathbf{x} \in \{True, False\}^N$ は N 個の二値変数の組合せを、f_{optim} は問題の的関数を表す実数値関数を、C_n は問題の制約を表す真理値関数を表す。

2. の処理において、真理値関数 C_n を実数値関数 f_{C_n} に変換する。このとき、$f_{C_n}(\mathbf{x})$ の値が最小値を取るような $\mathbf{x}$ を与えたときに $C_n(\mathbf{x}) = True$ となるようにする。すなわち、

$$\mathbf{x} = \underset{\mathbf{x}}{\mathrm{argmin}} f_{C_n}(\mathbf{x}) \Rightarrow C_n(\mathbf{x}) = True$$

となる。

3. では 1. 及び 2. おける目的関数にパラメータ λ_0 及び λ_n を掛け足し合わせることで $(\lambda_0, \lambda_n > 0)$、最終的な目的関数

$$f(\mathbf{x}) = \lambda_0 f_{\mathrm{optim}}(\mathbf{x}) + \sum_n \lambda_n f_{C_n}(\mathbf{x})$$

を生成する。

ここで、$f(\mathbf{x})$ は二値変数の組をとり実数値を与える多項式関数であるが、ここで二値変数は通常バイナリ変数 $(\{1, 0\})$ たはスピン変数 $(\{1, -1\})$ である。これを実際にアニーリングを用いて解くために、次元削減を行うことで高々 2 次の多項式に変換する必要がある。以降の議論は、すべてバイナリ変数 $\mathbf{q} \in \{1, 0\}^N$ を用いることを前提とする。

Performance evaluation of new order-reducing algorithm in annealing
† Taisei Ozu, Nagoya University

3　次元削減の手法

前章での要請により、二値変数を取る多項式関数の次元削減を行うことが必要であるが、例えばこの処理を自動で行うことのできる PyQUBO 等のソフトウェアは以下のような処理を行う (これを下、従来の方法と呼ぶ)。

1. 与えられた式の中で、$n(>2)$ 次の項を見つける。
2. n 次の変数の積を、後述の方法を用いて $n-1$ 次の積に変換する。
3. 以上の操作を、高々 2 次になるまで繰り返す。

2. の方法は以下の通りである。注目する項が $aq_{r_1}q_{r_2}\cdots q_{r_d}$(ただし a は実数値係数、q_n はバイナリ変数、$1 \leq r_n \leq N$ を任意の添え字とする) と表されるき

- 新しいバイナリ変数 q_{N+1} を用意し、当該項を $aq_{N+1}q_{r_3}\cdots q_{r_d}$ に置き換える。
- $q_{N+1} = q_{r_1}q_{r_2}$ という制約 C_M を問題の制約に付け加える
- $\lambda_M f_M(\mathbf{q}) = \lambda_M[1+q_{N+1}(3-2q_{r_1}-2q_{r_2})]$ を問題の目的関数に足し合わせる

とする。最後の操作は

$$q_{N+1} = \underset{q_{N+1}}{\operatorname{argmin}}[1+q_{N+1}(3-2q_{r_1}-2q_{r_2})]$$

となることを保証している。表 1 より、このとき制約 C_M が満たされていることがわかる。

一方、石川の方法 [1] では以下の方法を用いる。

$$n_d = \lfloor \frac{d-1}{2} \rfloor$$
$$S_1 = \sum_{n=1}^{d} q_{r_n},\quad S_2 = \sum_{m=1}^{d}\sum_{n=m+1}^{d} q_{r_m}q_{r_n}$$

として

$a<0$ とき

新しいバイナリ変数 q_{N+1} を用意し、注目する項 $aq_{r_1}q_{r_2}\cdots q_{r_d}$ を

$$aq_{N+1}[q_{r_1}+\cdots+q_{r_d}-(d-1)]$$

に置き換える。

$a>0$ かつ d が偶数のとき

新しいバイナリ変数 $q_{N+1},\ldots,q_{N+n_d}$ を用意し、

$$a\sum_{i=1}^{n_d} q_{N+i}(-2S_1+4i-1)+aS_2$$

表 1　$f_M(\mathbf{q})$ の値

q_{r_1}	q_{r_2}	$f_M(\mathbf{q})$ $(q_{N+1}=1)$	$f_M(\mathbf{q})$ $(q_{N+1}=0)$
0	0	4	1
0	1	2	1
1	0	2	1
1	1	0	1

に置き換える。

$a>0$ かつ d が奇数のとき

新しいバイナリ変数 $q_{N+1},\ldots,q_{N+n_d}$ を用意し、

$$a\sum_{i=1}^{n_d-1} q_{N+i}(-2S_1+4i-1)$$
$$+aq_{N+n_d}(-S_1+2n_d-1)+aS_2$$

に置き換える。

4　むすび

石川の方法が従来の方法より優れていると考えられる理由として以下の点がある。

1. 同じ次数 d の項に対して、追加する新しいバイナリ変数の数が少ない。
2. 制約 C_M を追加する必要がない。

1 つ目の点については、現実的なアニーリングマシンでは使用できる変数の数が限られているため、より少ない変数で再現された方が効率が良いと考えられる。

2 つ目の点については、制約最適化問題をアニーリングによよって求解する場合、すべての制約が満たされない場合、にパラメータを調整しアニーリングを再度行う必要がある (「アニーリングによる求解」9. を参照)。よって、制約が少ないほどアニーリング解が制約を満たしやすく、効率的であると考えられる。

これらの点を踏まえ、本発表では石川の方法が従来の手法に比べてどの程度効率的であるかを、実際のアニーリングマシンを用いて検証する。

参考文献

[1] 石川博, "高階グラフカット", 画像の認識・理解シンポジウム (MIRU2009), July 2019.