



中国科学院大学

University of Chinese Academy of Sciences

硕士学位论文

基于扩散蒙特卡洛方法的模拟量子退火研究

作者姓名: 李泽

指导教师: 乔从丰 教授 中国科学院大学物理科学学院

学位类别: 理学硕士

学科专业: 理论物理

培养单位: 中国科学院大学物理科学学院

2023 年 6 月

Research of Simulated Quantum Annealing
Based On
Diffusion Monte Carlo Method

A thesis submitted to
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Master of Natural Science
in Theoretical Physics

By

Li Ze

Supervisor: Professor Qiao Congfeng

School of Physical Sciences,
University of Chinese Academy of Sciences

June, 2023

中国科学院大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日 期：

中国科学院大学 学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：

日 期：

导师签名：

日 期：

摘 要

基于扩散蒙特卡洛方法的模拟量子退火研究,受限玻尔兹曼机,车间调度问题

关键词: 模拟量子退火, 扩散蒙特卡罗方法, 受限玻尔兹曼机, 车间调度问题

Abstract

Research of simulated quantum annealing based on diffusion Monte Carlo method,
Restricted Boltzmann Machine, Job Shop Scheduling Problem

Keywords: Simulated Quantum Annealing, Diffusion Monte Carlo Method, Restricted
Boltzmann Machine, Job Shop Scheduling Problem

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 研究意义	1
第 2 章 组合优化问题	3
第 3 章 变分蒙特卡罗方法	5
3.1 变分蒙特卡罗方法的原理	5
3.2 蒙特卡罗采样方法	6
3.3 试探波函数参数的更新	6
3.3.1 最速梯度下降法	6
3.4 试探波函数	8
第 4 章 神经网络量子态	9
4.1 循环神经网络波函数	9
4.1.1 循环神经网络 (RNN)	9
第 5 章 扩散蒙特卡罗方法	13
5.1 扩散蒙特卡罗方法的原理	13
第 6 章 车间调度问题	15
6.1 车间调度问题的定义	15
6.1.1 车间调度问题的描述	15
6.1.2 车间调度问题的析取图表示	15
6.2 车间调度问题模型的构建	15
6.2.1 变量定义	15
6.2.2 添加约束	16
6.2.3 目标函数	18
第 7 章 总结与展望	21
参考文献	23
致谢	25
作者简历及攻读学位期间发表的学术论文与研究成果	27

图形列表

4.1 循环神经网络	10
------------------	----

表格列表

第 1 章 引言

1.1 研究背景

1.2 研究意义

第 2 章 组合优化问题

第3章 变分蒙特卡罗方法

3.1 变分蒙特卡罗方法的原理

考虑试探波函数 $|\psi_w\rangle = \sum_i a_i |\psi_i\rangle$ ，其中 $|\psi_i\rangle$ 是该物理体系下包括 \hat{H} 在内的一组力学量完全集的共同本征态，相应的能量本征值为 E_i 。 a_i 即为该试探波函数展开的系数。

该试探波函数的能量期望值 E_w 可以写为：

$$E_w = \frac{\langle \psi_w | \hat{H} | \psi_w \rangle}{\langle \psi_w | \psi_w \rangle} = \frac{\sum_i |a_i|^2 E_i}{\sum_i |a_i|^2} \geq E_0 \frac{\sum_i |a_i|^2}{\sum_i |a_i|^2} = E_0 \quad \dots (3.1)$$

式 (3.1) 表明， E_w 给出了体系基态能量的一个上限。因此我们可以通过调整试探波函数的参数来不断减小 E_w ，使得其不断逼近基态能量 E_0 ，相应的试探波函数也会更加逼近于基态波函数 ψ_0 。

同直接求解定态薛定谔方程一样， E_w 准确计算的复杂度随着物理系统中粒子的数量指数上升。因此通常采取蒙特卡罗采样的方法估算 E_w ：

$$\begin{aligned} E_w &= \frac{\langle \psi_w | \hat{H} | \psi_w \rangle}{\langle \psi_w | \psi_w \rangle} \\ &= \sum_{s, s'} \frac{\langle \psi_w | s \rangle \langle s | \hat{H} | s' \rangle \langle s' | \psi_w \rangle \langle s | \psi_w \rangle}{\langle \psi_w | \psi_w \rangle \langle s | \psi_w \rangle} \\ &= \sum_s \frac{\langle \psi_w | s \rangle \langle s | \psi_w \rangle}{\langle \psi_w | \psi_w \rangle} \sum_{s'} \frac{\langle s | \hat{H} | s' \rangle \langle s' | \psi_w \rangle}{\langle s | \psi_w \rangle} \\ &= \sum_s P(s) E_{\text{loc}}(s) \end{aligned} \quad \dots (3.2)$$

其中， $|s\rangle$ 是该物理系统下的一套完备基底，式中假设该物理系统是离散的，连续物理系统下的推导只是将式中求和改成积分，因此不再赘述。

$P(s) = |\psi_w(s)|^2 / (\sum_s |\psi_w(s)|^2)$ 是采样的目标概率分布。

$E_{\text{loc}}(s) = \langle s | \hat{H} | \psi_w \rangle / \psi_w(s)$ 称之为局域能量。

由式 (3.2) 可知， E_w 的值等于概率分布 $P(s)$ 下局域能量 $E_{\text{loc}}(s)$ 的期望。因此我们可以对概率分布 $P(s)$ 进行蒙特卡罗采样，采样得到的态 (s_1, s_2, \dots) 相应的局域能量平均值即为 E_w 的估计值。

3.2 蒙特卡罗采样方法

3.3 试探波函数参数的更新

通过蒙特卡罗采样的方式估算 E_w 后，仍需要对试探波函数的参数进行不断的迭代更新，使得其不断逼近基态能量 ψ_0 。最简单直接的方法是计算 E_w 对参数 w_i 的偏导 $\partial E_w / \partial w_i$ ，继而沿着梯度方向不断减小参数 w_i ，当 E_w 收敛时，得到最终的基态能量估计值以及相应的基态波函数，这种方法称之为最速梯度下降法。除此之外还有许多更加高效的方法，例如随机重配法^[1]，共轭梯度下降法，随机梯度下降法等等。

3.3.1 最速梯度下降法

最速梯度下降法通过直接计算 E_w 对参数 w_i 的偏导 $\partial E_w / \partial w_i$ 来更新参数，因此其具有方便简单，计算量小，容易实现的优点。

但是，仅根据梯度来更新参数的方式很容易陷入局部最优。在更新迭代过程伊始，最速梯度下降法往往能取得较好的效果，但对于复杂的优化情形，例如试探波函数参数众多， E_w 与各个参数关系复杂时，最速梯度下降法往往会导致 E_w 止步于某一较小值，难以继续优化。

考虑 E_w 对参数 w_i 偏导 $\partial E_w / \partial w_i$ ，有：

$$\frac{\partial E_w}{\partial w_i} = \sum_s (E_{\text{loc}}(s) \frac{\partial P(s)}{\partial w_i} + P(s) \frac{\partial E_{\text{loc}}(s)}{\partial w_i}) \quad \dots (3.3)$$

其中概率分布 $P(s)$ 对参数 w_i 偏导 $\frac{\partial P(s)}{\partial w_i}$ ：

$$\begin{aligned} \frac{\partial P(s)}{\partial w_i} &= \frac{\psi_w^*(s)}{\sum_s |\psi_w(s)|^2} \frac{\partial \psi_w(s)}{\partial w_i} - \frac{|\psi_w(s)|^2}{(\sum_s |\psi_w(s)|^2)^2} \sum_{s'} \psi_w^*(s') \frac{\partial \psi_w(s')}{\partial w_i} \\ &= P(s) \frac{\partial \psi_w(s)}{\psi_w(s) \partial w_i} - P(s) \sum_{s'} \frac{\psi_w^*(s') \psi_w(s')}{\sum_s |\psi_w(s)|^2} \frac{\partial \psi_w(s')}{\psi_w(s') \partial w_i} \\ &= P(s) \frac{\partial \psi_w(s)}{\psi_w(s) \partial w_i} - P(s) \sum_{s'} P(s') \frac{\partial \psi_w(s')}{\psi_w(s') \partial w_i} \\ &= P(s) \Delta_{w_i}(s) - P(s) \sum_{s'} P(s') \Delta_{w_i}(s') \end{aligned} \quad \dots (3.4)$$

其中第一步利用了 $|\psi_w(s)|^2 = \psi_w^*(s)\psi_w(s)$ ，由于采用的试探波函数的参数是复数，有 $\partial\psi_w^*(s)/\partial w_i = 0$ 。第二、三步利用了 $P(s) = |\psi_w(s)|^2/(\sum_s |\psi_w(s)|^2)$ 。第四步定义试探波函数取对数后对参数 w_i 偏导 $\Delta_{w_i}(s) = \partial\psi_w(s)/[\psi_w(s)\partial w_i]$

接下来考虑局域能量 $E_{\text{loc}}(s)$ 对参数 w_i 偏导 $\partial E_{\text{loc}}(s)/(\partial w_i)$:

$$\begin{aligned} \frac{\partial E_{\text{loc}}(s)}{w_i} &= \sum_{s'} \langle s | \hat{H} | s' \rangle \frac{\partial(\psi_w(s')/\psi_w(s))}{\partial w_i} \\ &= \sum_{s'} \langle s | \hat{H} | s' \rangle \left[\frac{\partial\psi_w(s')}{\psi_w(s)\partial w_i} - \frac{\psi_w(s')}{\psi_w(s)} \frac{\partial\psi_w(s)}{\psi_w(s)\partial w_i} \right] \quad \dots (3.5) \\ &= \sum_{s'} \langle s | \hat{H} | s' \rangle \Delta_{w_i}(s') \frac{\psi_w(s')}{\psi_w(s)} - \Delta_{w_i}(s) E_{\text{loc}}(s) \end{aligned}$$

将式 (3.4) 和式 (3.5) 的结果代入式 (3.3)，有：

$$\begin{aligned} \frac{\partial E_w}{\partial w_i} &= \sum_s [E_{\text{loc}}(s)P(s)\Delta_{w_i}(s) - E_{\text{loc}}(s)P(s) \sum_{s'} P(s')\Delta_{w_i}(s')] \\ &\quad + P(s) \sum_{s'} \langle s | \hat{H} | s' \rangle \Delta_{w_i}(s') \frac{\psi_w(s')}{\psi_w(s)} - P(s)\Delta_{w_i}(s)E_{\text{loc}}(s)] \\ &= - \sum_s E_{\text{loc}}(s)P(s) \sum_{s'} P(s')\Delta_{w_i}(s') + \sum_s P(s) \sum_{s'} \langle s | \hat{H} | s' \rangle \Delta_{w_i}(s') \frac{\psi_w(s')}{\psi_w(s)} \\ &= - E_w \sum_s P(s)\Delta_{w_i}(s) + \sum_{s'} \Delta_{w_i}(s') |\psi_w(s')|^2 \sum_s \frac{P(s)\langle s | \hat{H} | s' \rangle}{\psi_w(s)\psi_w^*(s')} \\ &= - E_w \sum_s P(s)\Delta_{w_i}(s) + \sum_{s'} \Delta_{w_i}(s') P(s') \left[\sum_s \frac{\langle s' | \hat{H} | s \rangle \psi_w(s)}{\psi_w(s')} \right]^* \\ &= - E_w \sum_s P(s)\Delta_{w_i}(s) + \sum_s \Delta_{w_i}(s) P(s) E_{\text{loc}}^*(s) \\ &= \sum_s P(s)\Delta_{w_i}(s) (E_{\text{loc}}^*(s) - E_w) \quad \dots (3.6) \end{aligned}$$

其中第三步利用了 $E_w = \sum_s P(s)E_{\text{loc}}(s)$ 。

第四步利用了 $P(s) = |\psi_w(s)|^2/(\sum_s |\psi_w(s)|^2)$ 以及哈密顿算符的厄米性。第五步利用了 $E_{\text{loc}}(s) = \langle s | \hat{H} | \psi_w \rangle / \psi_w(s)$ 。

由式 (3.6) 可知， E_w 对参数 w_i 偏导 $\partial E_w / \partial w_i$ 同样可以通过对概率分布 $P(s)$ 进行蒙特卡罗采样进行计算，采样得到的态 (s_1, s_2, \dots) 相应的 $\Delta_{w_i}(s)(E_{\text{loc}}^*(s) - E_w)$ 平均值即为 $\partial E_{\text{loc}}(s)/(\partial w_i)$ 的估计值。

因此在更新试探波函数参数的每一步，我们都对概率分布 $P(s)$ 进行蒙特卡罗采样，根据采样结果计算 E_w 以及其对参数的偏导 $\partial E_w / \partial w_i$ ，进而沿着梯度方向不断减小参数 w_i 。当 E_w 收敛时，取 E_w 的实部作为最终的基态能量估计值，相应的试探波函数即为近似的基态波函数。

3.4 试探波函数

近年来，许多种试探波函数被尝试用来描述量子多体系统，例如神经网络量子态，张量网络态以及传统的经验波函数等等。本文采用基于神经网络的受限玻尔兹曼机作为试探波函数。

受限玻尔兹曼机可以基于对目标概率分布的采样学习描述相应的概率分布。。。。。。

第4章 神经网络量子态

4.1 循环神经网络波函数

循环神经网络（recurrent neural network, RNN）是一类用于处理序列数据的神经网络，其具有自回归性（autoregressive property）：条件概率 $P(\sigma_n|\sigma_{<n})$ 仅取决于 $\sigma_1, \dots, \sigma_{n-1}$ 。因此常用来预测前后相关的变量模型，在语言建模、机器翻译、语音识别等场景应用广泛。

相较于玻尔兹曼机，用自回归神经网络表示量子态的优势主要在于其高效、准确的采样能力。用玻尔兹曼机表示量子态时，为了优化参数以及计算物理量期望值，往往要采用马尔科夫链蒙特卡罗（Markov chain Monte Carlo）采样的方法，从引导波函数中采样。然而这种方式在问题规模大、目标函数势能图景复杂的情况下，一方面往往带来很高的计算成本，另一方面也很难避免得到真正独立、没有关联的样本。这在一定程度上阻碍了神经网络的表示能力，使得优化过程变得低效。而自回归神经网络在采样时，依照条件概率对各个变量进行依次采样，因此准确、高效，便于并行化。得到的样本彼此之间独立，没有关联，可以更好地发挥神经网络的表示能力。

4.1.1 循环神经网络（RNN）

我们采用循环神经网络模拟一个未知的概率分布。假定该概率分布中每一个变量组态 $\sigma \equiv (\sigma_1, \sigma_2, \dots, \sigma_N)$ 包含 N 个二进制变量 $\sigma_n \in \{0, 1\}$ 。

我们采用 $P(\sigma) \equiv P(\sigma_1, \sigma_2, \dots, \sigma_N)$ 表示一个变量组态对应的概率。由于循环神经网络的自回归性，有：

$$P(\sigma) = P(\sigma_1)P(\sigma_2|\sigma_1) \cdots P(\sigma_N|\sigma_{N-1}, \dots, \sigma_2, \sigma_1) \quad \dots (4.1)$$

其中 $P(\sigma_i|\sigma_{i-1}, \dots, \sigma_2, \sigma_1) \equiv P(\sigma_i|\sigma_{<i})$ 是给定所有 $j < i$ 变量 σ_j 下 σ_i 的条件概率。当所有条件概率 $P(\sigma_i|\sigma_{<i})$ 都确定时，变量组态 σ 对应的概率 $P(\sigma)$ 也就确定了。

RNN 的基本构建模块是一个循环单元，如图 4.1 所示，最简单的循环单元是一个非线性函数，该函数将 d_h 维隐藏矢量 \mathbf{h}_{n-1} 和二维输入矢量 σ_{n-1} 映射为 d_h

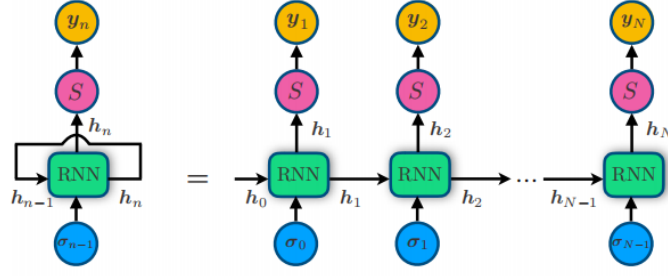


图 4.1 循环神经网络

Figure 4.1 Recurrent Neural Network

维输出隐藏矢量 h_n 。即：

$$h_n = f(W[h_{n-1}; \sigma_{n-1}] + b) \quad \dots (4.2)$$

其中 f 是非线性激活函数，例如 sigmoid 函数、双曲正切函数、ELU 函数等等。 $W \in \mathbb{R}^{d_h \times (d_h + 2)}$ 是权重矩阵。 $b \in \mathbb{R}^{d_h}$ 是偏差矢量。 $[a; b]$ 表示向量 a 和 b 的直和。为起始循环，需要选定 h_0 和 σ_0 的值。 σ_n 是输入 σ_n 的独热编码，即每个 σ_n 的值对应 σ 的一位。由于我们所关心的问题中，变量都是二进制的，因此 $\sigma_n = 0$ 对应 $\sigma = (1, 0)$ ， $\sigma_n = 1$ 对应 $\sigma = (0, 1)$ 。

变量组态 σ 对应的概率 $P(\sigma)$ 可以通过循序地计算条件概率得到：

$$P(\sigma_n | \sigma_{n-1}, \dots, \sigma_1) = y_n \cdot \sigma_n \quad \dots (4.3)$$

其中 $y_n = (y_n^1, y_n^2)$ 是一个二维矢量，其分量是正实数，所有分量之和等于 1。 y_n 由下式定义：

$$y_n \equiv S(Uh_n + c) \quad \dots (4.4)$$

其中 $U \in \mathbb{R}^{2 \times d_h}$ 是 Softmax 层的权重项， $c \in \mathbb{R}^2$ 是偏差项。Softmax 激活函数 S 为：

$$S(v_n) = \frac{\exp(v_n)}{\sum_i \exp(v_i)} \quad \dots (4.5)$$

由式 (4.1) 和式 (4.3) 可知，变量组态 σ 对应的概率 $P(\sigma)$ 为：

$$P(\sigma) = \prod_{n=1}^N y_n \cdot \sigma_n \quad \dots (4.6)$$

由 y_n 分量之和等于 1 的性质可知， $\sum_{\sigma} P(\sigma) = 1$ ，因此用 RNN 表示的概率分布是归一的。

对 RNN 进行采样时, 同样根据条件概率分布, 循序地确定 $\mathbf{h}_1, \mathbf{y}_1$, 然后根据 \mathbf{y}_1 确定 σ_1 的值, 以此类推, 直到 $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_N)$ 完全确定。

实际应用时, 由于 RNN 在捕捉变量间的长程关联时可能会导致梯度爆炸或消失, 因此常采用更加复杂的改进结构。

第 5 章 扩散蒙特卡罗方法

5.1 扩散蒙特卡罗方法的原理

第6章 车间调度问题

6.1 车间调度问题的定义

6.1.1 车间调度问题的描述

6.1.2 车间调度问题的析取图表示

6.2 车间调度问题模型的构建

为了求解车间调度问题，我们需要将其映射到一个量子系统上，求解车间调度问题也就等价于求解该量子系统的基态。因此我们需要构造模型，将问题进行等价表述，再采用前面章节介绍的变分、扩散蒙特卡罗方法进行求解。

6.2.1 变量定义

设待求解车间调度问题总的工序数量为 N ，对各个工件上的各个工序依次标号 $1 \sim N$ 。析取图中，虚拟的起始工序标号 0 ，虚拟的结束工序标号 $N + 1$

对于任意两个不属于同一工件但需要在同一机器上加工的工序 $\{i, j | i, j \in \{1, \dots, N\} \wedge J_i \neq J_j \wedge M_i = M_j\}$ ，定义二进制变量 $\{x_{i,j} | x_{i,j} \in \{0, 1\} \wedge (i < j)\}$ 。

$x_{i,j} = 1$ 意味着在这个变量对应的方案下，工序 i 比工序 j 更早加工； $x_{i,j} = 0$ 意味着在这个变量对应的方案下，工序 i 比工序 j 更晚加工。

对应析取图中，每个二进制变量 $x_{i,j}$ 一对一对应于未确定方向的析取弧，因此当所有定义的二进制变量的值确定时，相应的调度方案也就确定了。我们将一个确定的调度方案对应的所有二进制变量取值的集合记作 x ，对应于一个解。解 x 对应的有向图记作 D_x ，有向图中已经确定了方向的析取弧的集合记作 E_x ，受工件上工序加工顺序限制的方向固定的边记作 A 。

车间调度问题等价于寻找任意一个最优解 $x_0 \in \mathcal{R}_{\text{opt}}$ ，其中 \mathcal{R}_{opt} 表示全局最优解的集合，该集合中任意一个解所对应的调度方案都能使得完工时间最短。

然而，这样一个确定的调度方案并不一定是合理的，因为确定析取弧方向后的析取图可能是一个循环的有向图，不存在最长路径。我们也不能从一个循环图中确定循环路径上工序的先后顺序。因此，相应的 x 是一个不合理解。

6.2.2 添加约束

解决不合理解的方法有两种：一种是在目标函数上附加对不合理解的惩罚项，这种方式会趋使生成的解朝着可行解的方向更新，但并不能杜绝不合理解，并且会使得目标函数更加复杂，增大优化的难度。我们称这种约束方式为软约束；另一种是在解的更新方式上作限制，使其总是在可行解的范围内迭代，这种方式构建难度大，需要保证解的收敛性，往往需要进行额外的限制。我们称这种约束方式为硬约束。

我们采用 Van Laarhoven 等^[2]更新解的方式来添加硬约束，以下就约束方式以及收敛性予以说明：

在每次更新解的时候，不同于常规随机翻转所有变量的更新方式，我们仅仅随机翻转对应于关键弧 (v, w) 的变量 $x_{v,w}$ ($v < w$)，方便起见，我们将这些变量称为关键变量。从图论的观点看，我们仅仅变更有向图中，处在最长路径中弧的方向 $(v, w) \rightarrow (w, v)$ ，相应的变量 $x_{v,w} = 1 - x_{v,w}$ ($v < w$)。

选取这种更新方式有以下几点原因：

1. 翻转有向图中非关键路径所产生的新的非循环有向图，其最长路径的长度一定大于等于原有有向图最长路径的长度（因为新的非循环有向图仍包含原最长路径）。
2. 翻转有向图中的关键路径 $(v, w) \rightarrow (w, v)$ 所产生的新的有向图一定仍是一个非循环图。
3. 对于任意一个合理解 x ，总存在有限的翻转关键变量的步骤，使其到达某个最优解 $x_0 \in \mathcal{R}_{\text{opt}}$ ，相应的调度方案即是使得完工时间最短的最优方案。

值得注意的是，第一条原因有一定贪婪的元素在其中，因此并不充分。实际上我们在更新解的时候，通常也会接受相对更差的解，这样往往能够在一定程度上避免陷入局部最优，这种思想在经典拟退火算法中也有所体现。以下对第二、三条原因作进一步分析证明：

断言 6.1. 假设弧 $e = (v, w) \in E_i$ 是一个非循环有向图 D_i 的关键弧，令 D_j 表示翻转图 D_i 中弧 e 所得到的有向图。那么图 D_j 一定也是非循环的。

证明. 假设图 D_j 是一个循环图，由于原图 D_i 是非循环的，图 D_j 相较于 D_i 只是删除了弧 (v, w) ，增加了弧 (w, v) 。删除一个弧不会使图变得循环，增加弧才有

可能。因此弧 (w, v) 一定处在 D_j 中的循环路径上, 即图 D_j 中同时存在弧 (w, v) 和路径 (v, x, y, \dots, w) 。由于图 D_j 是由图 D_i 翻转弧 (v, w) 得到的, 图 D_i 中也一定存在路径 (v, x, y, \dots, w) 。显然, 这条路径要比直接经过路径 (v, w) 要长 (因为车间调度问题对应的图的权重在顶点上, 对应于各个工序的加工时间), 与前提弧 $e = (v, w)$ 是一个关键弧 (位于最长路径中) 矛盾, 因此假设不成立, 图 D_j 一定是非循环的。 \square

为了证明上述第三条原因, 我们需要先证明一个引理:

引理 6.2. 对于任意一个非最优的合理解 $x \notin \mathcal{R}_{opt}$ 和任意一个最优解 $x_0 \in \mathcal{R}_{opt}$, 集合 $K_x(x_0) = \{e = (v, w) \in E_x | e \text{ 是关键弧} \wedge (w, v) \in E_{x_0}\}$ 一定非空。其中, E_x 表示 x 所对应的有向图中确定方向的析取弧。

证明. 我们将分三部分进行证明: 首先证明 E_x 总包含关键弧; 然后证明总是存在关键弧 $e \in E_x$ 且 $e \notin E_{x_0}$; 最后利用这两个结论说明集合 $K_x(x_0)$ 一定非空。

1. 假设 E_x 不包含关键弧, 那么所有的关键弧都属于 A , 即最长路径是某一工件对应的所有工序的加工顺序。因此其完工时间便是该工件上所有工序的加工时间之和。显然, 这是所有可能的生产调度方案完工时间的下限。即任意一个有向图的最长路径长度都要大于 x 所对应的有向图的最长路径长度, 因此 x 一定是最优解, 这与前提矛盾, 因此假设不成立, E_x 一定包含关键弧。

2. 假设 E_x 中所有关键弧 e 都属于 E_{x_0} , 那么 x_0 所对应的图一定包含了 x 所对应的有向图的最长路径, 其最长路径的长度是 x_0 所对应的图的最长路径长度下限。而 x_0 是最优解, 其对应的图的最长路径长度应当是所有可能的有向图中最小的, 因此 x 一定也是一个最优解, 这与前提矛盾, 因此假设不成立, 一定关键弧 $e \in E_x$ 且 $e \notin E_{x_0}$ 。

3. 因为 E_x 总是包含不属于 E_{x_0} 的关键弧 $e = (v, w)$, 且有向图中要确定析取弧方向, 必然包含 (v, w) 和 (w, v) 二者之一, 所以必有 $(w, v) \in E_{x_0}$, 即集合 $K_x(x_0)$ 一定非空。 \square

定理 6.3. 对于任意一个非最优的合理解 $x \notin \mathcal{R}_{opt}$, 总可以通过翻转关键变量的方式构造一个有限长度的更新序列, 使得解 x 最终到达某个最优解 $x_0 \in \mathcal{R}_{opt}$

证明. 我们可以按照如下规则构造一个有限长度的更新序列 $\{\lambda_0, \lambda_1, \dots\}$:

1. 首先, 初始解为选定的任意一个非最优的合理解 $\lambda_0 = x$
2. 接下来每一步更新解 $\lambda_k \rightarrow \lambda_{k+1}$ 时, 都只翻转 E_{λ_k} 中属于集合 $K_{\lambda_k}(x_0)$ 的边 e 所对应的变量。因为 e 是关键弧, 由断言 6.1 可知, 新解 λ_{k+1} 对应的图 $D_{\lambda_{k+1}}$ 一定是非循环的。并且由引理 6.2 可知, 集合 $K_{\lambda_k}(x_0)$ 非空, 意味着这种更新操作总是存在的。最后, 翻转的变量总是关键变量, 属于之前我们定义的更新操作。

分析: 对于任意一个非最优的合理解 x , 定义集合 $M_x(x_0) = \{e = (v, w) \in E_x | (w, v) \in E_{x_0}\}$ 。容易看出 $|M_{\lambda_{k+1}}(x_0)| = |M_{\lambda_k}(x_0)| - 1$ 。因此, 令 $k = |M_x(x_0)|$, 有 $|M_{\lambda_k}(x_0)| = 0$ 。由于 $K_x(x_0) \subseteq M_x(x_0)$, 可知 $K_{\lambda_k}(x_0) = \emptyset$ 。由引理 6.2 可知, 解 λ_k 一定是最优解 $\lambda_k \in \mathcal{R}_{\text{opt}}$ 。这意味着我们构建了 $k + 1$ 长度的更新序列, 使得任意一个非最优合理解 x 通过翻转关键变量的方式到达了某个最优解 x_0 , 定理得证。 \square

综上所述, 随机翻转对应于关键弧 (v, w) 的关键变量 $x_{v,w}$ 的更新方式能够确保解 x 在合理理解的范围内更新, 并且任意一个合理解都能通过这种方式更新到最优解, 因此可以用来构建马尔科夫链, 在变分和扩散蒙特卡罗方法中加以应用。

然而需要指出的是, 这样的一个马尔科夫链是可约的, 即链中任意两个态之间并不一定能够通过这种更新关键变量的方式到达彼此, 这是因为虽然任意一个合理解都可以到达最优解, 但是最优解并不一定能够到达任意一个合理解。

6.2.3 目标函数

因为我们采用了硬约束的方式, 没有在目标函数上附加对不合理解额外的惩罚项, 所以目标函数仅仅是合理解 x 对应的调度方案的完工时间。

为方便表示, 我们定义 $\text{makespan}(x|v)$ 为合理解 x 对应的图 $D_x = (V, E \cup A)$ 中, 虚拟起始顶点到顶点 $v \in V$ 的最长路径长度。即:

$$\text{makespan}(x|v) = P_v + \max_{u \in \mathcal{N}_v} (\text{makespan}(x|u)) \quad (6.1)$$

其中 $\mathcal{N}_v = \{u | (uv \in A) \vee (uv \in E_{M_v} \wedge (x_{uv} = 1 \vee x_{vu} = 0))\}$ 表示与 v 邻接的, 存在边指向 v 的顶点的集合。 uv 是有向边 (u, v) 的简写。 P_v 代表顶点 v 对应的工序的加工时间。 E_{M_v} 代表顶点 v 对应的工序所需加工的机器 M_v 在析取图中对应的析取弧的集合。

按上述定义，目标函数可以写为：

$$H = makespan(x|end) \quad (6.2)$$

其中 end 代表有向图中虚拟的结束顶点。求任意一个解 x 对应的目标函数值可以采用一个简单的递归算法实现，其复杂度为 $\mathcal{O}(|V| + |E| + |A|)$ 。

用 x_k 表示解 x 中第 k 个二进制变量，需要指出，由于我们采用的二进制变量 $x_k \in \{1, 0\}$ ，与量子伊辛模型中 $\hat{\sigma}_z^{(k)}$ 的本征值 $\sigma_z^{(k)} \in \{1, -1\}$ ，存在简单的映射关系 $x_k = (1 + \sigma_z^{(k)})/2$ ，因此目标函数可以映射到由各个 $\hat{\sigma}_z^{(k)}$ 组成的量子系统。该量子系统的哈密顿量在该组二进制变量构成的基底中的矩阵为：

$$\langle x | \hat{H} | x' \rangle = \delta_{x,x'} * makespan(x|end) \quad (6.3)$$

第 7 章 总结与展望

参考文献

- [1] Sorella S. Generalized lanczos algorithm for variational quantum monte carlo [J]. Physical Review B, 2001, 64(2): 024512.
- [2] Van Laarhoven P J, Aarts E H, Lenstra J K. Job shop scheduling by simulated annealing [J]. Operations research, 1992, 40(1): 113-125.

致 谢

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历：

李泽，河南省商水县人，中国科学院大学物理科学学院硕士研究生。

已发表（或正式接受）的学术论文：

- 1.

