Simulated Annealing of the Travelling Salesman Problem

Advanced Session: Algorithmic approximate solution to a combinatorial problem

Giancola Simone[1]

[1]Bocconi University, Milan, Italy

Mini-Course on Computation, Harvard University, January 2022

An ironic quote, hope this is not the case!

Before I came here I was confused about this subject. Having listened to your lecture I am still confused, but on a higher level.

Enrico Fermi, 1938 Physics
Nobel Prize

About & Main sources

About me
Currently an MS in Data Science candidate at Bocconi University
General interest in science (still exploring)
Enjoys coding

Contacts:
simonegiancola09@gmail.com
personal webpage

Acknowledgements
The opportunity could **not** have been possible **without**:
PhD student Chou Chi-Ning, Harvard University, School of Engineering and Applied Sciences

The whole presentation could **not** have been possible **without** the contents from:
Computer Programming, Bocconi University, 30509 (awesome course!)
Prof. Baldassi Carlo
Prof. Lucibello Carlo

Thanks!

Lecture Contents

Lecture Path

Notation

This is a definition

Here I define something

This is a theorem

Something is gnihtemoS backwards

Proof

This is a proof

A remark an observation or an example

for example, I observe or remark that this is an observation

Brief Introduction

In short:

- Simulated Annealing (SA) is a technique used to solve complex non linear problems

Brief Introduction

In short:

- first application to the Travelling Salesman problem is attributed to Kirkpatric et Al. [6]

Brief Introduction

In short:

- It is a *metaheuristic* method using Statistical Mechanics concepts

Brief Introduction

In short:

- Clever cross implementation of many subjects altogether

Brief Introduction

In short:

- perfect example of inspiration from *natural phenomena*

Lecture Contents

- Recap the framework of Statistical Mechanics

Lecture Contents

- Recap the framework of Statistical Mechanics
- Present and analyze the Traveling Salesman Problem

Lecture Contents

- Recap the framework of Statistical Mechanics
- Present and analyze the Traveling Salesman Problem
- Propose a setting that relaxes its complexity

Lecture Contents

- Recap the framework of Statistical Mechanics
- Present and analyze the Traveling Salesman Problem
- Propose a setting that relaxes its complexity
- Derive a Simulated Annealing algorithm that attempts to respect those requirements

Link to course Lectures

- Lecture II.a: Statistical Mechanics:

Microstates ω

many, probabilistically distributed on Ω

Macrostates $X(\omega)$

properties of microstates common to many ω

$$X : \Omega \mapsto \mathbb{R}$$

Link to course Lectures

- Lecture II.a: Statistical Mechanics:

Microstates ω

many, probabilistically distributed on Ω

Macrostates $X(\omega)$

properties of microstates common to many ω

$$X : \Omega \mapsto \mathbb{R}$$

- The problems
  - We can measure efficiently a macrostate, but do not identify the microstate
  - We can observe one realization of ω across time, not many realizations $\omega_1, \ldots, \omega_n$

Thermodynamics Example

Microstate

Configurations (positions in $\mathbb{R}^3$) of particles with a non measurable energy

Macrostate

Temperature as a result of the geometrical configuration

Thermodynamics Example

Microstate

Configurations (positions in $\mathbb{R}^3$) of particles with a non measurable energy

Macrostate

Temperature as a result of the geometrical configuration

Space of possible configurations, easy

If they are k, all distinct, to be placed in an $(n \times n) \in \mathbb{R}^2$ grid, and we do not account for symmetry, we have:

$$\binom{n^2}{k}$$

arrangements.
Not easy at this Stack Question. Anyway **Big!**

Not available vs available

Ensemble Average

$$\mathbb{E}[X(\omega)] := \int_{\mathcal{X}} x(\omega) f(x(\omega)) dx(\omega)$$

integral over Ω

Not available vs available

Ensemble Average

$$\mathbb{E}[X(\omega)] := \int_{\mathcal{X}} x(\omega) f(x(\omega)) dx(\omega)$$

integral over Ω

Time Average

$$\mathbb{E}[X_t] := \frac{1}{t_{max}} \sum_{k=1}^{t_{max}} x_k(\omega)$$

sum for a single realization ω

Ergodicity

- Under appropriate assumptions:

$$\implies \mathbb{E}[X_t] \overset{a.s.}{\to} \mathbb{E}[X(\omega)]$$

Ergodicity

- Under appropriate assumptions:

$$\implies \mathbb{E}[X_t] \overset{a.s.}{\to} \mathbb{E}[X(\omega)]$$

- We could then sample iteratively and almost surely get to the mean of the distribution (actually any bounded function, more details later).

Z encodes all possible microstates! It is big indeed.

Ergodicity

- Under appropriate assumptions:

$$\implies \mathbb{E}[X_t] \overset{a.s.}{\to} \mathbb{E}[X(\omega)]$$

- We could then sample iteratively and almost surely get to the mean of the distribution (actually any bounded function, more details later).
- We will see an energy fashioned application of this including Boltzmann distribution:

$$\mathbb{P}(\frac{Energy_i}{T} = u_i) = \frac{e^{u_i}}{Z} \ : \ Z = \sum_i e^{u_i}$$

Z encodes all possible microstates! It is big indeed.

Z Notable elements

- Z depends on T

Z Notable elements

- Z depends on T
- Z normalizes the energy configuration to a probability

Z Notable elements

- Z depends on T
- Z normalizes the energy configuration to a probability
- Boltzmann distribution allows for a link between configurations and properties.
  - It denotes a phase space as we saw in class

Lecture Path

A difficult problem

Travelling Salesman Problem (TSP)

We are given a set of N cities, and a matrix $\mathcal{D} = \{d_{ij}\}_{i=1,\dots,N}^{j=1,\dots,N} \in \mathbb{R}^N \times \mathbb{R}^N$ storing **symmetric** distances between each of the cities. The well known **Travelling Salesman Problem**[a] resorts to finding a minimum length cycle of the cities.

[a] In terms of optimization

A difficult problem

Travelling Salesman Problem (TSP)

We are given a set of N cities, and a matrix $\mathcal{D} = \{d_{ij}\}_{i=1,\dots,N}^{j=1,\dots,N} \in \mathbb{R}^N \times \mathbb{R}^N$ storing **symmetric** distances between each of the cities. The well known **Travelling Salesman Problem**[a] resorts to finding a minimum length cycle of the cities.

[a] In terms of optimization

Why is it difficult? We will formalize it and give a degree of complexity.

A difficult problem

Travelling Salesman Problem (TSP)

We are given a set of N cities, and a matrix $\mathcal{D} = \{d_{ij}\}_{i=1,\dots,N}^{j=1,\dots,N} \in \mathbb{R}^N \times \mathbb{R}^N$ storing **symmetric** distances between each of the cities. The well known **Travelling Salesman Problem**[a] resorts to finding a minimum length cycle of the cities.

[a] In terms of optimization

Why is it difficult? We will formalize it and give a degree of complexity.

Minimization Problem Statement

If the total distance is $E(r)$ for a route r then we wish to find:

$$r_{min} = \underset{\mathcal{R}}{argmin}\{E(r)\}$$

Lecture Path

Solver A

Algorithm 1 Enumeration (not really) Algorithm

$r_{min} \leftarrow None$
$E_{min} \leftarrow \infty$
for $r \in \mathcal{R}$ **do**
    if $E(r) < E_{min}$ **then**
        $r_{min} \leftarrow r$
        $E_{min} \leftarrow E(r)$
    end if
end for
return r_{min}

Enumeration Attempt

An Enumeration attempt from FourmiLab.ch (Autodesk creator)[6]

Assume we have at disposal a computer that does $2.59 \cdot 10^9$ operations per second (just to simplify things). Let $N = 31$ cities, then:

$$(N-1)! = \prod_{i=1}^{N-1} (N-i) = 30! \approx 2.65 \cdot 10^{32}$$

Assuming that the distance is calculated in negligible time we would need a total time of

$$\frac{30!}{2.65 \cdot 10^9} sec = 10^{23} sec \approx 3 \cdot 10^{16} \ years \approx 2 \times 10^6 \text{ stories of the universe}^a$$

[a] Assuming the universe is about 13.8 Billion years old, first google suggestion

Formalisms

NP-hard class of problems

$$NP\text{-}hard := \{H \ : \ \forall L \in NP \exists \textit{efficient reduction}\, \{L_i\} \rightarrow H\} \tag{3.1}$$

Difficult to solve, difficult to check for a candidate solution with a deterministic Turing Machine

TSP Hardness

TSP is NP-hard

Proof Sketch

TSP is combinatorially exploding, searching the space is inefficient with a deterministic Turing Machine. Also, given a claim that an instance is a solution, it is not efficient to check it in polynomial time.
Precisely: reduction of a Hamiltonial Cycle Problem $\in$ *NP-Complete*.

Solution B

Algorithm 2 Greedy Algorithm $O(N^2 log(N))$

```
arr ← sort(cities)
edges ← [ ]
while len(edges) != N do
    Select minimum distance tuple (i, j) ∈ arr
    if [check no subcycles if add (i, j) to edges] then
        if [check degrees <= 2 if add (i, j) to edges] then
            edges.append((i, j))
        end if
    end if
end while
return edges
```

Heuristics result

Big-Theta bound

Given a function $g(\cdot)$

$$\Theta[g(N)] := \{f(N) \ : \exists c_1,\, c_2 \in \mathbb{R}^+ \ N_0 \in \mathbb{N}^+ \ : \\ 0 \leq c_1 g(N) \leq f(N) \leq c_2 g(N) \ \forall N > N_0\}$$

Broadly speaking, $g(\cdot)$ bounds a set of functions $f(\cdot)$ after some point.

Approximation ratio of an Algorithm

Ratio cost of Algorithm solution & exact solution

Heuristics are not reliable

- approximation ratio of Solution B is $\Theta[log(N)]$ [1]

Heuristics are not reliable

- approximation ratio of Solution B is $\Theta[log(N)]$ [1]
- on average in the 15-20% more than best known method for exact solution[3]
  - Held-Karp Algorithm

$P = NP$?

Not at all, heuristics are not general exact solutions. Solution B is just **satisficing**.
Nothing is ever guaranteed

Lecture Path

Framework

Random sequential samples

Following what we observed in class about ergodicity, we could envision a system that:

- explores options efficiently
- does not get stuck at satisficing options (so called local minimas)
- resembles the actual distribution wrt $E(\cdot)$

Notation

Routes will be called states in some cases. We will refer to r with the pedix i or j to follow a canonical notation when we deal with multiple states.

Setting

Routes Space $\mathcal{R}$

$$\mathcal{R} := \{r \text{ valid}\}$$

feature u_i

$$u_i = f(r_i) \forall i \in \mathcal{R} \text{ for some } f(\cdot)$$

Here f can be anything (it is the macrostate measurement!).

probability distribution ρ

The feature, and thus r_i have a distribution $r_i \sim \rho(\cdot)$

Transition Matrix $Q^{(t)}$

$$Q^{(t)} := \{p_{ji}(t) := \mathbb{P}[X_t = j | X_{t-1} = i,\ t]\ \forall i, j \in \mathcal{R}\}$$

Boltzmann Fashion

Expressing a probability distribution as a Boltzmann Distribution

With this setting $\forall i\ \rho(r_i) > 0$ and up to an additive constant we can find $\left\{ Z, \{u_i\} \right\}$ such that

$$\forall r\ \rho(r_i) = \rho_i = \frac{e^{u_i}}{Z} \ : \ Z = \sum_i e^{u_i} \tag{4.1}$$

Which is just a rewording of the distribution. It is **not** easy to sample directly, Z is a huge sum.

Boltzmann precisely

$u_i = -\frac{E(r_i)}{T}$ for a temperature T. We will use this later.

Theoretical MC requirements I

Strong Stationarity Necessary conditions

$$\{X_t\} : \exists Q : Q\rho = \rho \iff \forall i \in \mathcal{X} \textit{ non null recurrent} \tag{4.2}$$

Theoretical MC requirements I

Strong Stationarity Necessary conditions

$$\{X_t\} : \exists Q : Q\rho = \rho \iff \forall i \in \mathcal{X} \textit{ non null recurrent} \tag{4.2}$$

This is **not** enough, imagine if we sampled from a distribution stuck at one point forever. It would be stationary, but it would always depend on its starting point and never explore the space. We need something else.

Theoretical MC requirements I

Strong Stationarity Necessary conditions

$$\{X_t\} : \exists Q : Q\rho = \rho \iff \forall i \in \mathcal{X} \textit{ non null recurrent} \tag{4.2}$$

This is **not** enough, imagine if we sampled from a distribution stuck at one point forever. It would be stationary, but it would always depend on its starting point and never explore the space. We need something else.

What is missing

After a property of the distribution we need a property of the process itself

Theoretical MC requirements II

Ergodic theorem

$$\{X_t\} : \forall i \in \mathcal{X} \; i \; ergodic \implies \lim_{t_{max}\to\infty} \prod_{t}^{t_{max}} Q^{(t)} X_0 = \rho \tag{4.3}$$

$$\forall j \quad \lim_{t\to\infty} \mathbb{P}[X_n = j] = \lim_{t\to\infty} \sum_{i\in\mathcal{X}} \mathbb{P}[X_t = j | X_0 = i]\mathbb{P}[X_0 = i] \tag{4.4}$$

$$\sum_{i\in\mathcal{X}} \mathbb{P}[X_0 = i] p^*[X = j] \tag{4.5}$$

$$= p^*[X = j] \perp\!\!\!\perp t \implies p^*[X = j] = \rho_j \tag{4.6}$$

Moreover this implies that if g is a bounded function:

$$\implies \mathbb{E}[\hat{g}(X)] \overset{a.s.}{\to} \mathbb{E}[g(X)] \tag{4.7}$$

Strong Stationarity effect

Strong Stationarity implies DBC

$$\{X_t\} : \exists Q : Q\rho = \rho \implies \forall j \in \mathcal{X} \sum_{i \neq j} Q_{ji}\rho_i = \sum_{k \neq j} Q_{kj}\rho_j$$

We call this condition Global Balance Condition (GBC). Intuitively, *inflow* = *outflow* for every state.

Detailed Balance (DBC) Assumption

GBC is difficult to check or impose. We will assume detailed balance holds:

$$\forall i, j \in \mathcal{X} \quad Q_{ji}\rho_i = Q_{ij}\rho_j$$

Intuitively, each tuple has *inflow* = *outflow*. No joint dynamics considered.

Idea

Create an ergodic process such that:

- It is easy to propose

Idea

Create an ergodic process such that:

- It is easy to propose
- given a configuration we propose another one accordingly

Idea

Create an ergodic process such that:

- It is easy to propose
- given a configuration we propose another one accordingly
- Ideally, this is done by comparing the Distance/Energy

Idea

Create an ergodic process such that:

- It is easy to propose
- given a configuration we propose another one accordingly
- Ideally, this is done by comparing the Distance/Energy
- For any tuning of any parameter, we always accept when the Energy/Distance is lower.

Idea

Create an ergodic process such that:

- It is easy to propose
- given a configuration we propose another one accordingly
- Ideally, this is done by comparing the Distance/Energy
- For any tuning of any parameter, we always accept when the Energy/Distance is lower.
- We will use this notion:

PA split

In our setting, we wish to propose candidates that are valid. For this reason, for each i, j tuple we will *split* the matrix into a proposal part P and an acceptance part A

$$Q_{ji} = P_{ji} A_{ji} \tag{4.8}$$

Intuitively, Q is the distribution of shifts where each entry can be seen as: $\mathbb{P}(\textit{sample } j|i)\mathbb{P}(\textit{accept } j|i)$.

Simplifying work

Symmetric Proposals Assumption

$$P = P^T \iff P_{ji} = P_{ij} \; \forall i, j \in \mathcal{X}$$

Delta Notation

$$\Delta_{ji} :== u_j - u_i = -\left(\frac{E(r_j) - E(r_i)}{T} \right)$$

Again, the $E(\cdot)$ part will be used later!

Building A (I): The rule

Metropolis Rule

In terms of practice, the most widely used proposal auxiliary function is called Metropolis Rule. It merges both previous rules.

$$h(\Delta_{ji}) = |\Delta_{ji}| \tag{4.9}$$

Metropolis Rule Properties

If the Metropolis Rule is used for a matrix A then $\forall i, j \in \mathcal{X}$:

$$A_{ji} = min\left\{1, \frac{\rho_j}{\rho_i}\right\} = min\left\{1, \frac{\mathbb{P}(r_{candidate})}{\mathbb{P}(r_{current})}\right\} \tag{4.10}$$

Further details in the lecture notes!

Building A (II): The rule

Proof part one

$$A_{ji} = exp\left\{\frac{1}{2}(\Delta_{ji} - |\Delta_{ji}|)\right\} \quad \text{applying M rule} \quad (4.11)$$

$$\iff \begin{cases} e^0 = 1 \ if \ \Delta_{ji} \geq 0 \\ e^{\Delta_{ji}} = exp\{-\frac{\Delta E_{ji}}{T}\} \ if \ \Delta_{ji} < 0 \end{cases} \quad \text{Expanding the modulus} \quad (4.12)$$

$$\iff A_{ji} = min\left\{1, e^{\Delta_{ji}}\right\} \quad \text{considering both cases} \quad (4.13)$$

$$\iff A_{ji} = min\left\{1, \frac{\rho_j}{\rho_i}\right\} \quad \text{Explained below} \quad (4.14)$$

Building A (III): The rule

Proof part two

Where the last passage comes from the fact that:

$$e^{\Delta_{ji}} = e^{u_j - u_i} = exp\left\{ - \frac{E(r_j) - E(r_i)}{T} \right\} = \frac{\frac{exp\left(\frac{-E(r_j)}{T} \right)}{Z}}{\frac{exp\left(\frac{-E(r_i)}{T} \right)}{Z}} = \frac{\rho_j}{\rho_i}$$

Building A (III): The rule

Proof part two

Where the last passage comes from the fact that:

$$e^{\Delta_{ji}} = e^{u_j - u_i} = exp\left\{ - \frac{E(r_j) - E(r_i)}{T} \right\} = \frac{\frac{exp\left(\frac{-E(r_j)}{T} \right)}{Z}}{\frac{exp\left(\frac{-E(r_i)}{T} \right)}{Z}} = \frac{\rho_j}{\rho_i}$$

- whenever a move is beneficial in terms of reduced distance we accept it
- in the opposite case acceptance depends on the relative change and decays quickly (being inside an exponent)

Building A (III): The rule

Proof part two

Where the last passage comes from the fact that:

$$e^{\Delta_{ji}} = e^{u_j - u_i} = exp\left\{ - \frac{E(r_j) - E(r_i)}{T} \right\} = \frac{\frac{exp\left(\frac{-E(r_j)}{T} \right)}{Z}}{\frac{exp\left(\frac{-E(r_i)}{T} \right)}{Z}} = \frac{\rho_j}{\rho_i}$$

- whenever a move is beneficial in terms of reduced distance we accept it
- in the opposite case acceptance depends on the relative change and decays quickly (being inside an exponent)
- In any non-decreasing-distance proposal, the probability of acceptance depends on T.

The role of T

T extreme cases

- for $T \to \infty$ we have $\rho \to \mathcal{U}(\mathcal{R}) \implies$ Random Walk, always accept candidates
- for $T \to 0$ we have $\rho \to \mathbb{1}(r_{min}) \implies$ accept iff $\Delta_{ji} \geq 0$

These results are proved in the Lecture Notes!

Example

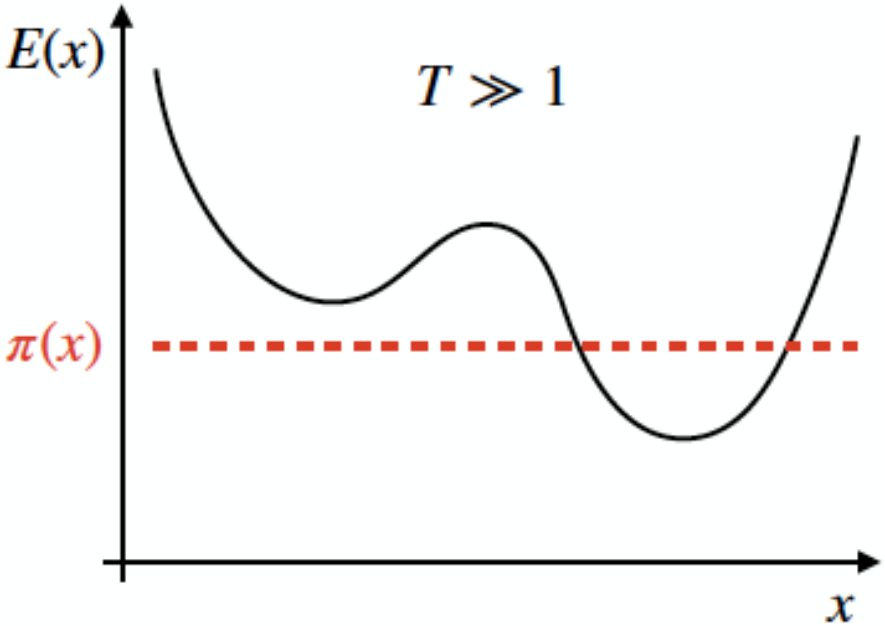


Figure: Uniform for $T \to \infty$

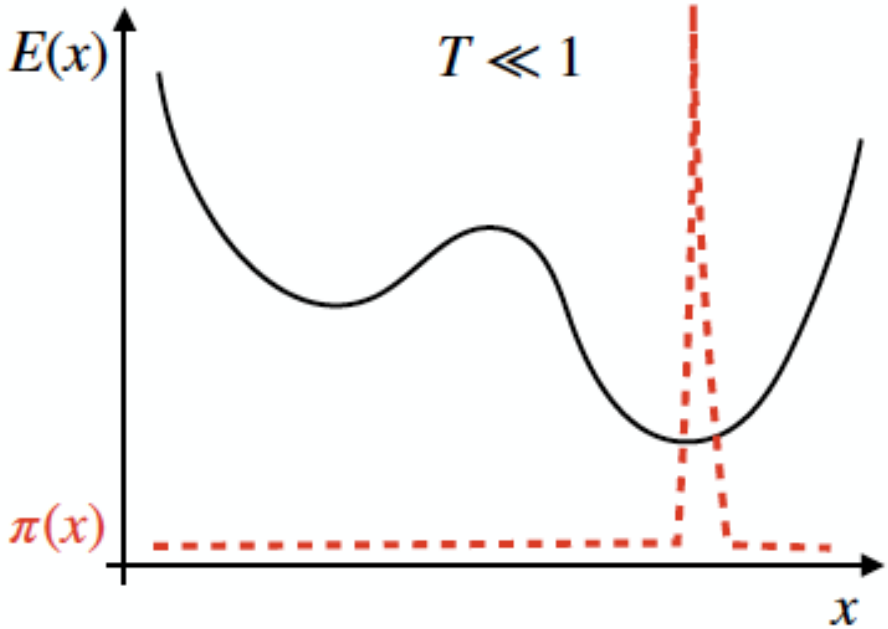


Figure: Concentrated for $T \to 0$

Credits: Bocconi University, Computer Programming, 30509 (awesome class!)

Building P (I)

P represents the distribution of feasible proposal routes. It must hold that an instance:

- starts and ends at the same city

Building P (I)

P represents the distribution of feasible proposal routes. It must hold that an instance:

- starts and ends at the same city
- touches all cities only once $\implies |r_{cand}| = N$

An efficient P for TSP

propose a switch of cities

$$r_{curr} : \{i \leftrightarrows j,\ v \leftrightarrows r\} \quad r_{cand} : \{i \leftrightarrows v,\ j \leftrightarrows r\}$$

That satisfies the requirements. Under random sampling and appropriate checking of the candidate, sample randomly from $\mathcal{R}_{valid}(x) \forall r \in \mathcal{R}$ configurations

Building P (I)

P represents the distribution of feasible proposal routes. It must hold that an instance:

- starts and ends at the same city
- touches all cities only once $\implies |r_{cand}| = N$
- does not dis-join the tour $\implies$ keeps the path valid.

An efficient P for TSP

propose a switch of cities

$$r_{curr} : \{i \leftrightarrows j,\ v \leftrightarrows r\} \quad r_{cand} : \{i \leftrightarrows v,\ j \leftrightarrows r\}$$

That satisfies the requirements. Under random sampling and appropriate checking of the candidate, sample randomly from $\mathcal{R}_{valid}(x) \forall r \in \mathcal{R}$ configurations

Building P (I)

P represents the distribution of feasible proposal routes. It must hold that an instance:

- starts and ends at the same city
- touches all cities only once $\implies |r_{cand}| = N$
- does not dis-join the tour $\implies$ keeps the path valid.
- Is possibly easy to evaluate in terms of comparison with different rs

An efficient P for TSP

propose a switch of cities

$$r_{curr} : \{i \leftrightarrows j,\ v \leftrightarrows r\} \quad r_{cand} : \{i \leftrightarrows v,\ j \leftrightarrows r\}$$

That satisfies the requirements. Under random sampling and appropriate checking of the candidate, sample randomly from $\mathcal{R}_{valid}(x) \forall r \in \mathcal{R}$ configurations

Building P (II)

ΔE is easy

$$\Delta E = E(r_{cand}) - E(r_{curr}) \quad (4.15)$$
$$= d_{iv} + d_{jr} - d_{ij} - d_{jr} \quad (4.16)$$

As all the other distances are the same and cancel out.

We will refer to P as a kernel $k(\cdot|r_{current})$. It is easy to sample from this kernel.

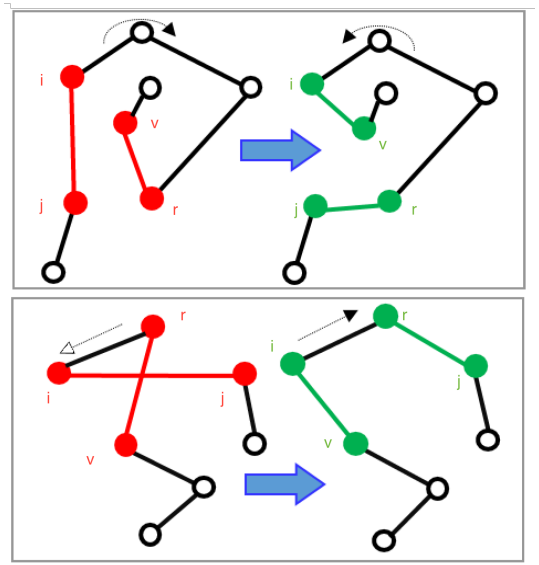


Figure: City swap graphically

Lecture Path

A Stochastic Solution

One T is **not** enough!

A Stochastic Solution

One T is **not** enough!

- When $T = \infty$ we would need $O(N!)$ operations to reach the solution in the worst case
- When $T = 0$ we would get stuck at local minimas if the energy function E is non-convex (highly likely this is the case)

A Stochastic Solution

One T is **not** enough!

- When $T = \infty$ we would need $O(N!)$ operations to reach the solution in the worst case
- When $T = 0$ we would get stuck at local minimas if the energy function E is non-convex (highly likely this is the case)
- $\forall T \in (0, \infty)$ the distribution concentrates around the global minima but does not avoid escaping **all** local minimas, as the selectiveness blocks the procedure at depression areas.

What if we could use all of them?

Again, inspiration from Nature

Informal Simulated Annealing (SA)

Simulated Annealing is an approach that finds a balance between the extremes, gradually decreasing the temperature to explore at the beginning and sequentially become more selective as $T \to 0$.
Its name comes from the Physical process of annealing, which Wikipedia defines as follows:

> *[...](annealing) involves heating a material above its recrystallization temperature, maintaining a suitable temperature for an appropriate amount of time and then* cooling

Dealing with T

Temperature Schedule T

Given a sequence of natural numbers $\{1, \dots, t_{max}\} \subset \mathbb{N}$:

$$T \ : \ \{1, \dots, t_{max}\} \to [0, \infty) \quad : \quad \forall c' > c \ \ T(c') \leq T(c) \tag{5.1}$$

We could also impose:

$$T(0) = \infty \vee T(t_{max}) = 0$$

But this is not necessary. T is thus a decreasing function in the region.

Using this schedule:

- choose a random starting configuration r_0,

Dealing with T

Temperature Schedule T

Given a sequence of natural numbers $\{1, \dots, t_{max}\} \subset \mathbb{N}$:

$$T \ : \ \{1, \dots, t_{max}\} \to [0, \infty) \quad : \quad \forall c' > c \ \ T(c') \leq T(c) \tag{5.1}$$

We could also impose:

$$T(0) = \infty \vee T(t_{max}) = 0$$

But this is not necessary. T is thus a decreasing function in the region.

Using this schedule:

- choose a random starting configuration r_0,
- for a given number of iterations $t_{max} \in \mathbb{N}$ explore the space $\mathcal{R}$

Dealing with T

Temperature Schedule T

Given a sequence of natural numbers $\{1, \dots, t_{max}\} \subset \mathbb{N}$:

$$T \ : \ \{1, \dots, t_{max}\} \to [0, \infty) \quad : \quad \forall c' > c \ \ T(c') \leq T(c) \tag{5.1}$$

We could also impose:

$$T(0) = \infty \vee T(t_{max}) = 0$$

But this is not necessary. T is thus a decreasing function in the region.

Using this schedule:

- choose a random starting configuration r_0,
- for a given number of iterations $t_{max} \in \mathbb{N}$ explore the space $\mathcal{R}$
- with different selectiveness granularities

Procedure

Algorithm 3 Simulated Annealing

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0,1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

Procedure, line by line

Require: r_0 and $E(\cdot)$, t_{max} and $T(\cdot)$, $k(\cdot|\cdot)$

$r \leftarrow r_0$ ▷ we assign r as the starting configuration, r is current

for $i = 1, \ldots, t_{max}$ **do** ▷ for a given number of iterations

$r_{cand} \sim k(\cdot|r)$ ▷ sample a valid candidate from P

$t_i = T(i)$ ▷ t_i is the current temperature

$\Delta E = E(r_{cand}) - E(r)$ ▷ new-old energy change

draw $u_i \sim \mathcal{U}(0, 1)$ ▷ u_i used to simulate a probability

if $u_i \leq min\left\{1, exp\left[-\frac{\Delta E}{t_i}\right]\right\}$ **then** ▷ Metropolis rule

$r \leftarrow r_{cand}$ ▷ x_{cand} is the update of x, move accepted

end if ▷ otherwise x is unchanged

end for

return x

An observation

Output

The returned value r will be a configuration, the result of an iterative process of exploration of routes which gradually accepts less and less worse proposals until it reaches a minimum solution.

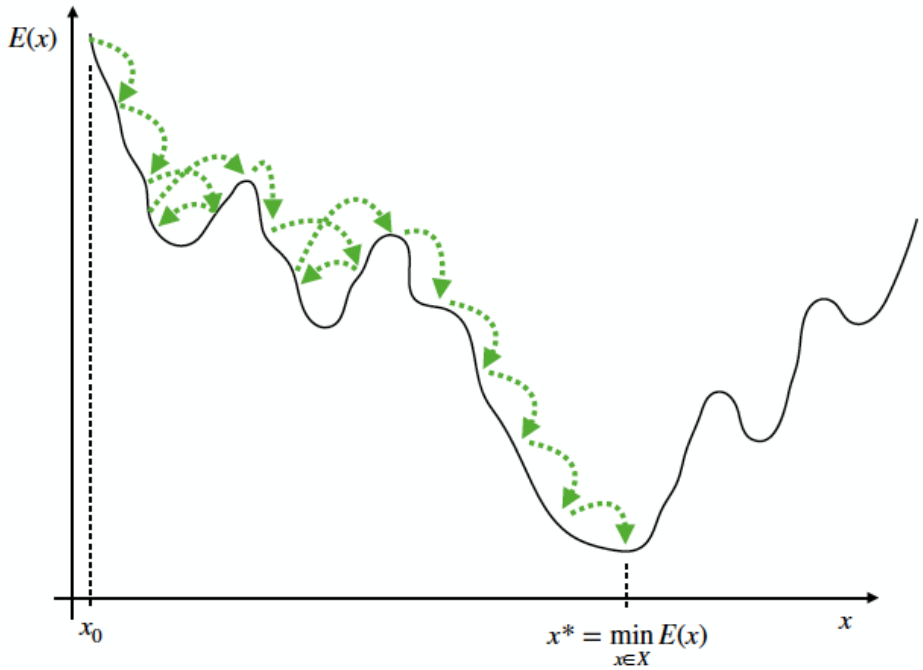


Figure: Algorithm Desired behavior

Credits: Bocconi University, Computer Programming, 30509 (awesome class!)

Lecture Path

Relaxing assumptions

Throughout the process, we could have made things more difficult.

Relaxing assumptions

Throughout the process, we could have made things more difficult.

- Asymmetric *TSP*
- Code the problem (many sources on the internet)

Relaxing assumptions

Throughout the process, we could have made things more difficult.

- Asymmetric *TSP*
- Code the problem (many sources on the internet)
- Asymmetric proposals
  - Acceptance rule slightly more complicated

Relaxing assumptions

Throughout the process, we could have made things more difficult.

- Asymmetric *TSP*
- Code the problem (many sources on the internet)
- Asymmetric proposals
  - Acceptance rule slightly more complicated
- Proving all the statements

Main points

- We are given a complex problem in combinatorics

Main points

- We are given a complex problem in combinatorics
- Find an iterative solution with a metaheuristic method

Main points

- We are given a complex problem in combinatorics
- Find an iterative solution with a metaheuristic method
- All thanks to the detailed balance condition!

Limitations

- Clearly, not exact
  - solving TSP efficiently would imply $P = NP$

Limitations

- Clearly, not exact
  - solving TSP efficiently would imply $P = NP$
- Needs tuning, case by case analysis

Limitations

- Clearly, not exact
  - solving TSP efficiently would imply $P = NP$
- Needs tuning, case by case analysis
- Requires efficient sampling, otherwise no time saved

Limitations

- Clearly, not exact
  - solving TSP efficiently would imply $P = NP$
- Needs tuning, case by case analysis
- Requires efficient sampling, otherwise no time saved
- smooth energy function makes SA redundant
  - slower than more straightforward optimization

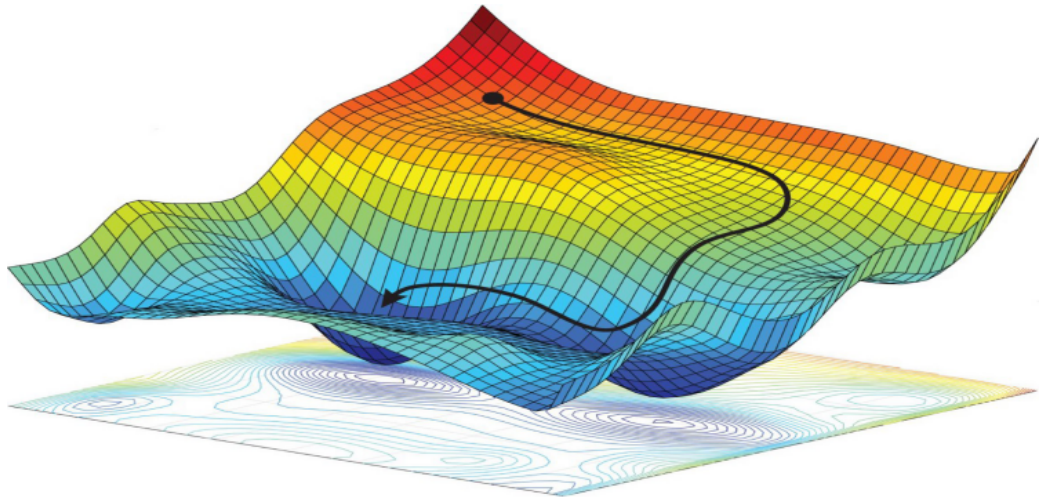
Concluding

Any question/discussion, let me know!

Thank you!

simonegiancola09@gmail.com

personal webpage



References I


Judith Brecklinghaus and Stefan Hougardy. "The Approximation Ratio of the Greedy Algorithm for the Metric Traveling Salesman Problem". In: *arXiv:1412.7366 [cs, math]* (Dec. 23, 2014). arXiv: 1412.7366. URL: http://arxiv.org/abs/1412.7366 (visited on 01/14/2022).


Introduction to Statistical Mechanics — Introduction to Statistical Mechanics. URL: https://web.stanford.edu/~peastman/statmech/ (visited on 01/14/2022).


David S Johnson and Lyle A McGeoch. "The Traveling Salesman Problem: A Case Study in Local Optimization". In: (), p. 103.


MarkovChains. URL: https://www.cs.yale.edu/homes/aspnes/pinewiki/MarkovChains.html (visited on 01/14/2022).

References II

On the nearest neighbor rule for the metric traveling salesman problem — Elsevier Enhanced Reader. DOI: 10.1016/j.dam.2014.03.012. URL: https://reader.elsevier.com/reader/sd/pii/S0166218X14001486?token=5088E0AF37C6C017686E01FD171CA1756D9951F78D9B65F40E29AEC268originRegion=eu-west-1&originCreation=20220114002155 (visited on 01/14/2022).

Simulated Annealing: The Travelling Salesman Problem. URL: https://www.fourmilab.ch/documents/travelling/anneal/ (visited on 01/14/2022).