Mini-Course on Computation, Harvard University **Organizer: Chi-Ning Chou**

Notes: Simulated Annealing, derivation and Travelling Salesman Problem

Jan. 14, 2022 **Written by: Simone Giancola**

> 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, Physics Nobel Prize, 1938

Description The following document is the expanded content of an advanced section on Simulated Annealing, part of the Mini-Course on Computation offered by Harvard university on January 2022. The ideal reader is a person interested in the topic of Markov Chain Monte Carlo methods. Being almost self-contained, it can accommodate beginners and mid-experienced individuals willing to dive deeper into the topic. Experts can also enjoy a refresher or a different perspective on the problem.
The main source is a course I had: Computer Programming, Bocconi University, held by Prof. Lucibello C. and Prof. Baldassi C.
Other references [*] can be found at the end.

Contents

1 Link to course lectures **2**

2 Problem and Complexity **3**
2.1 Why is it difficult? . . . 3
2.2 Formalisms . . . 4

3 Why is it linked to what we saw in class? **6**

4 Sampling Scheme **9**
4.1 Building A . . . 10
4.2 Building P . . . 14

5 Algorithm Design **16**

6 Conclusions **18**

A Markov Chain Redux **19**

1 Link to course lectures

In lecture II.a we introduced the topic of statistical mechanics. There the concepts of *microstate* and *macrostate* are studied, deriving properties and notions.
What we usually can *measure efficiently* is the macrostate, but how can we draw conclusions for the microstate features? Formulated differently, how can we efficiently extract relevant information from easy to measure signals of a system?

Evidently, it is not possible to look at the space of outcomes Ω all at once. Indeed, we only see one realization of it $X_t(\omega) : \omega \in \Omega$. With time we could collect a sample of consequent realizations of the same ω, and with additional assumption claim that a measurement on the sequence is a reliable estimation of the measurement on the actual distribution. This is equivalent to assuming that the sequence is ergodic and that the values of interest are stationary[1]. With this setting, we can resort to sampling techniques that simulate the actual target *almost surely*.

We need these properties to be able to generate a consecutive sample that resembles what the distribution would look like for a given time t. In a microstate/macrostate fashion (remember what we saw in class?) the time average is equal to the ensemble average, which is the target distribution. Thanks to this property, we can think of drawing sequentially while being sure that this sample resembles the whole.

To check that these hold, we should look for an aperiodic and irreducible chain. Formally, ergodicity is impossible to verify for *long to enumerate* cases. We just wish that the sequence explores all possible configurations of the variable of interest without systematic periods of occurrence inside. This would ensure that the Markov Chain is stationary and that the process is ergodic.
In this brief Bonus lecture, we will have a glimpse on how to solve a difficult problem using these concepts for a technique called *Simulated Annealing*.
Talk Outline

1. Introduction to the application
2. Complexity assessment
    - Linking class topics
    - Modifying the distribution of routes
3. Stochastic Algorithmic Solution
    - Requirements
    - An interesting Matrix split to interpret the procedure
    - Easy case Simulated Annealing
4. Takeaways & food for thought

[1] Where ergodicity is a property of a generated sequence and stationarity is a property of the generated distribution

2 Problem and Complexity

Problem 1 (Travelling Salesman Problem).
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**[2] resorts to finding a *minimum length cycle of the cities*. In other words we are asked to find a permutation of the indices $\{1, \dots, n\}$ such that it is the least distance consuming trip across all cities.

Definition 2 (Permutation of cities r). Given a list of cities $\{1, \dots, N\}$ and a permutation function α then a permutation of cities/route/tour[3] is:

$$r = \alpha(\{1, \dots, N\}) \in \mathcal{R} \coloneqq \{space\ of\ possible\ routes\} \tag{2.1}$$

Definition 3 (Cost function $E(\cdot)$ for TSP). Assume we are using zero based array-indexing, and that a permutation r is used for a distance matrix $\mathcal{D}$. Then, the cost function $E(\cdot)$ is such that[4]:

$$E(r) = \sum_{k=0}^{N-1} d_{r[k],r[(k+1)\%N]} \tag{2.2}$$

Definition 4 (TSP as an integer programming formulation). Using the concept of Adjacency matrix $\mathcal{A}$ for a given tour r we can define $\mathcal{A}^{(r)} = \{a_{ij}^{(r)} = a_{ji}^{(r)} = 1 \iff (i,j)\ adjacent\ in\ r\}$ and claim that a solution to Problem 1 satisfies:

$$r^* = \underset{\mathcal{R}}{argmin}\left\{E(r)\right\} = \underset{\mathcal{R}}{argmin}\left\{\sum_{i<j} d_{ij}a_{ij}\right\} \tag{2.3}$$

Where the sum for $i < j$ is done to avoid double counting connections.
We also have that a valid r induces an adjacency matrix $\mathcal{A}$:

$$\begin{cases} \sum_j a_{ij} = 2 \forall i \ \text{ degree of a node is two} \\ \nexists subcycles \in \mathcal{A} \end{cases} \tag{2.4}$$

Other problem formulations, are not needed for the sake of the lecture and will be skipped.

2.1 Why is it difficult?

If we wish to permute N cities, there are $N!$ ways to do so. Assuming that we start from any city, there will be $(N-1)!$ possible ways to visit all the others and come back to the starting point.

Example 2.1 (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). Usually, we are around

[2] In terms of optimization
[3] We will consider them to be equivalent in this document
[4] Slightly adapted, as we are representing the cost function of N possible permutations, can you see why?

Algorithm 1 Enumeration (not really) Algorithm

```
r_min ← None
E_min ← ∞
for r ∈ R do
    if E(r) < E_min then
        r_min ← r
        E_min ← E(r)
    end if
end for
return r_min
```

hundreds of millions per seconds, so consider it to be **very** powerful.
Let $N = 31$ cities, then:

$$(N-1)! = (N-1)(N-2)\dots(N-(N-2))(N-(N-1)) = \prod_{i=1}^{N-1}(N-i) = 30! \approx 2.65 \cdot 10^{32}$$

This is clearly a large number of possible options to evaluate. Still, someone might say that a good procedure would be enumerating all of them and evaluating the distance for each one. Assuming that the calculation is done efficiently in negligible time, we would still have 30! enumerations to do. Then we would need a total time of

$$\frac{30!}{2.65 \cdot 10^9} sec = 10^{23} sec \equiv 31709791983764584\, years \approx 3 \cdot 10^{16}\ years \approx 2 \times 10^6 \text{ stories of the universe}^{5}$$

Clearly this is not the most efficient way to tackle the problem!

2.2 Formalisms

Definition 5 (P class of Problems).

$$P \coloneqq \{L\, problems\ :\ \exists \text{ polynomial time solution with a deterministic turing machine } M\} \tag{2.5}$$

Definition 6 (NP class of problems).

$$NP \coloneqq \{L\, problems\ :\ \exists \text{ polynomial time solution for a deterministic turing machine } M\} \tag{2.6}$$

Definition 7 (NP-$hard$ class of problems).

$$NP\text{-}hard \coloneqq \{H\ :\ \forall L \in NP \exists efficient\, reduction\, \{L_i\} \to H\} \tag{2.7}$$

Were by efficient we mean in Polynomial time.

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

Observation 8 ($P = NP$ with NP-*hard* solutions?).

$$if\ \exists\ deterministic\ polynomial\ solution\ for\ H \in NP\text{-}hard \implies P = NP \tag{2.8}$$

However, this is one of the most debated unsolved Computer Science problems known as the P **versus** NP **problem**.

The traveling salesman problem optimization problem is part of the class of problems defined as NP-hard. There, solutions and results cannot be found or checked in Polynomial time by a deterministic Turing Machine.

In theoretical terms, there is no known efficient algorithm for an exact solution, nor a candidate route r_{cand} can be validated as the lowest distance permutation.
In practical terms, there are instances that have easy solutions, heuristics, satisficing solutions. Let us introduce the second and third procedure we can think of apart from enumeration.

Algorithm 2 Nearest Neighbor Algorithm $O(N^2)$

$r \leftarrow [\,]$
Select a random city c
$r.append(c)$
while $len(r)\ != N$ **do**
    $c_{curr} \leftarrow r[-1]$
    $c_{new} \leftarrow \underset{c_i \notin r}{argmin}\{d_{c_{curr},c_i}\}$
    $r.append(c_{new})$
end while
return r

Algorithm 3 Greedy Algorithm $O(N^2 log(N))$

$arr \leftarrow sort(cities)$
$edges \leftarrow [\,]$
while $len(edges)\ != \ N$ **do**
    Select minimum distance tuple $(i, j) \in arr$
    if [check no subcycles if add (i, j) to edges] **then**
        if [check degrees <= 2 if add (i, j) to edges] **then** ▷ Namely, check conditions at Eq 2.4
            $edges.append((i, j))$
        end if
    end if
end while
return edges

Definition 9 (Big-Theta bound). Given a function $g(\cdot)$

$$\Theta[g(N)] \coloneqq \{f(N)\ :\ \exists c_1,\, c_2 \in \mathbb{R}^+\ N_0 \in \mathbb{N}^+\ :\ 0 \leq c_1 g(N) \leq f(N) \leq c_2 g(N)\ \forall N > N_0\} \tag{2.9}$$

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

Definition 10 (Approximation ratio of an Algorithm). Ratio cost of Algorithm solution & exact solution

In [1] the authors claim is that a greedy solution from Algorithm 3 is $\Theta[log(N)]$ longer than the optimum solution. It is also empirically found that such a process leads to results that are on average in the 15-20% bound of the best known method to find an exact solution[3], the Held-Karp Algorithm, which runs in exponential time[6]. Surprisingly, they also claim that in some applications we can have a $< 25\%$ difference of Algorithm 2 and the lower bound.

In most cases, this result is satisfactory. This is also due to the fact that NP hardness generalizes to any instance of the problem, and thus accounts for the **worst case** scenario. In real life, the average case prevails, and many times this makes computations easier.

Observation 11. Does this mean that $P = NP$? Not at all, we would need to find an instance-overall efficient procedure to claim that $H \in NP\text{-}hard$ has a deterministic solution in polynomial time as in Observation 8.

We can thus link the following notions together and claim that:

- TSP is a combinatorially exploding problem in computational time
- greedy non-optimal solutions can be found in efficient time
- real life cases are not edge/corner cases

Nevertheless:

Why would we want to find a better solution?

We would like to, since another efficiently retrievable and reasoned solution is (hopefully) going to generalize better. At the moment, we are not exploiting any peculiarity of the structure. Algorithm 1 is impossible to implement, Algorithms 2 & 3 are highly dependent on the starting point.

3 Why is it linked to what we saw in class?

Coming back to the concepts of microstate and macrostate, in this setting we have that it is easy to measure the total distance of a route, but highly inefficient to do so for each and every case. Due to this finding exact bounds of values [7] is hard, factorially hard, which is worse than exponentially. In practice, there are no trivial solutions to the problem.

Now, Imagine transferring the same question to an isolated system of particles and being asked to find the possible configuration to grant minimum energy E. Instead of finding a path, we have to find an arrangement that does the same. In the simplest case, without considering many restrictions on the types of particles and possible matching configurations in terms of temperature, we have that for n^2 spots in $\mathbb{R}^2$ and k particles there are $\binom{n^2}{k}$ options to arrange them. For a more difficult case not yet completely generalizable, I reroute you to this question asked on the Math StackExchange.

[6] This is the standard best working exact solution algorithm for TSP
[7] maxima and minima, where in this case we are interested in the latter

We have to envision a system that is able to explore options efficiently and does not get stuck at satisficing options (so called local minimas).
To do so, we will link the problem to the concept of energy and build up on this.

Assumption 12 (States as routes & indexing). 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.

Theorem 13 (Expressing a probability distribution as a Boltzmann Distribution). Assume the possible configurations $\mathcal{R} \coloneqq \{r_i \; routes\}$ have a feature $u_i \in \mathcal{X}$ probability distribution $\rho(\cdot)$ with its usual properties. Then $\forall i \; \rho(r_i) > 0$ and up to an additive constant we can find $\left\{ Z, \{u_i\} \right\}$ such that

$$\forall r \; \rho(r_i) = \rho_i = \frac{e^{u_i}}{Z} \; : \; Z = \sum_i e^{u_i} \tag{3.1}$$

Which is just a rewording of the distribution.

Assume that we could somehow exploit a draw of consequent routes indexed by time[8] $t = \{1, \dots, t_{max}\}$ expressed by a random variable $X_t \in \mathcal{X} = \mathcal{R}$. For simplicity, we assume each draw depends on its previous state only. This is equivalent to drawing Markov Chains (MCs). We need to define some objects before making meaningful claims. For readers not knowledgeable of Markov Chains, I prepared a small redux on the topic in Appendix A.

Theorem 14 (Strong Stationarity Necessary conditions).

$$\{X_t\} : \exists Q : Q\rho = \rho \iff \forall i \in \mathcal{X} non \; null \; recurrent \tag{3.2}$$

Theorem 15 (Ergodic theorem).

$$\{X_t\} : \forall i \in \mathcal{X} \; i \; ergodic \implies \lim_{t_{max} \to \infty} \prod_t^{t_{max}} Q^{(t)} X_0 = \rho \tag{3.3}$$

In other words, as the the size of the sampled MC increases, we reach independence from the starting point X_0 and obtain the real distribution of the variable $X \sim \rho(\cdot)$.

$$\forall j \; \lim_{t \to \infty} \mathbb{P}[X_n = j] = \lim_{t \to \infty} \sum_{i \in \mathcal{X}} \mathbb{P}[X_t = j | X_0 = i] \mathbb{P}[X_0 = i] = \sum_{i \in \mathcal{X}} \mathbb{P}[X_0 = i] p^*[X = j] \tag{3.4}$$

$$= p^*[X = j] \perp\!\!\!\perp t \implies p^*[X = j] = \rho_j \tag{3.5}$$

Moreover this implies that if g is a bounded function:

$$\implies \mathbb{E}[\hat{g}(X)] \overset{a.s.}{\to} \mathbb{E}[g(X)] \tag{3.6}$$

Where if $g(X) = X$ we have the easy case where the time average is equal to the ensemble average.

[8] Note: $t_{max} \neq N$. N is the number of cities, t_{max} is the number of draws from the random variable. X can take any configuration $r_i \in \mathcal{R}$ at any time t where $|\mathcal{R}| = N!$

If we had a stationary ergodic and aperiodic distribution, the probability distribution ρ would be invariant to the transition matrix. Namely, this denotes an equilibrium condition in a closed system.
If we also had that this equilibrium condition concentrated around the global minimum distance configuration we could find a way to simulate such a system and evaluate its limiting behavior.
A way to concentrate a set of configurations depending on some feature (in this case, the distance) is exactly the Boltzmann distribution we saw in class for a given parameter $\left\{T,\ u_i = -\frac{E(r_i)}{T}\right\}$ where $E(r) = dist(route)$ in our case:

$$\rho_i = \rho(r_i) = \frac{e^{\frac{-E(r_i)}{T}}}{Z} \tag{3.7}$$

Here:

- At lower distance configurations the probability is higher
- The temperature T is a common scaling parameter

If we found an efficient way of sampling from this distribution we would be able to find one of those configurations.

Observation 16 (Transition Representation). Given a transition matrix $Q \coloneqq \{p_{ji} = \mathbb{P}(i \to j)\}$ we can see that:

$$\mathbb{P}(j|i \in \mathcal{X}) = \sum_{i \in \mathcal{X}} Q_{ji}\rho_i = (Q\rho)_j \tag{3.8}$$

Namely, for a target configuration j we extract the contribution from all configurations and cross multiply with the matrix that allocates the probability of an outcome.

A desirable stability property follows:

Definition 17 (Global Balance Condition (GBC)).

$$\forall j \in \mathcal{X} \quad \sum_{i \neq j} Q_{ji}\rho_i = \sum_{k \neq j} Q_{kj}\rho_j \tag{3.9}$$

In plain words the contribution to other configurations is balanced by the same other configurations for each j. $Influx = Outflux$ for all configurations.

Theorem 18 (Strong stationarity Implications). MC strongly stationary as in Definition 43 satisfies GBC.

Proof. By the strong stationarity requirement of the chain, using Observation 16 we impose $\forall j \in \mathcal{X}$:

$$\mathbb{P}(j|i \in \mathcal{X}) = \rho_j \qquad \text{The probability of reaching } j \text{ is the actual probability of } j \tag{3.10}$$

$$\sum_{i \in \mathcal{X}} Q_{ji}\rho_i = \rho_j \qquad \text{We just translated it into the matrix} \tag{3.11}$$

$$\sum_{i \neq j} Q_{ji}\rho_i + Q_{jj}\rho_j = \left[\sum_{k \in \mathcal{X}} Q_{kj}\right]\rho_j \qquad \text{Split on the left and same on the right} \tag{3.12}$$

Where in the last passage we added a term on the RHS which is 1, as shown in Equation A.3. Going on:

$$\sum_{i\neq j} Q_{ji}\rho_i + Q_{jj}\rho_j = \sum_{k\neq j} Q_{kj}\rho_j + Q_{jj}\rho_j \qquad \text{Repeating the passage of Equation 3.12 on the RHS} \tag{3.13}$$

$$\sum_{i\neq j} Q_{ji}\rho_i = \sum_{k\neq j} Q_{kj}\rho_j \qquad \text{by direct implication } \forall j \in \mathcal{X} \tag{3.14}$$

□

Definition 19 (Detailed Balance Condition (DBC)).

$$\forall i, j \in \mathcal{X} \;\; Q_{ji}\rho_i = Q_{ij}\rho_j \tag{3.15}$$

Namely, this stricter condition imposes that each couple of configurations (i, j) satisfies the $Influx = Outflux$ requirement.

Theorem 20 (DBC vs DBC). Detailed balance is stricter than global balance as a restriction, it allows for less variability of flows.

$$DBC \implies GBC \tag{3.16}$$

Nevertheless, it is easier to check and will help us further. Somehow we could say:

$$DBC \succ GBC \tag{3.17}$$

For a relation $\succ$.

Assumption 21 (DBC holds). We impose that DBC of Definition 19 holds and derive the properties of a chain of this type.
Given that we are in a setting in which $size(Q) = (N-1)!$ we will avoid evaluating all possible cases and simplify the requirement by making it stricter . This would ensure that each contribution is balanced with respect to its opposite direction and not their joint dynamics.

4 Sampling Scheme

Observation 22 (Idea). If we had an efficient way of exploring the space $\mathcal{X}$ with easy to check properties such as the DBC of Assumption 21, we might find a reasoned solution in efficient time. Given a starting configuration the target would be proposing possible options efficiently and in a clever way. Some basic requirements would be creating a process such that:

- It is easy to propose
- given a configuration we propose another one accordingly[9]
- Ideally, this is done by comparing the Distance/Energy
- For any tuning of any parameter, we always accept when the Energy/Distance is lower.

[9]This is formally equivalent to building a non homogeneous Markov Chain as moves will depend on time t *somehow*

One can argue that we wish to find a balance between greedy algorithms such as Algorithms 2 and 3 and a Random Walk that just wanders the space $\mathcal{X}$.

Definition 23 (Proposal/Acceptance (PA) matrix 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

$$Q_{ji} = P_{ji} A_{ji} \tag{4.1}$$

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

Theorem 24 (PA split properties). $\forall i, j$

$$P_{ji} \in (0, 1) \qquad \text{is the distribution of the } j \text{ proposals} \tag{4.2}$$

$$\sum_j P_{ji} = 1 \qquad \text{As before, we always propose something} \tag{4.3}$$

$$A_{ji} \in (0, 1] \qquad \text{is the probability of accepting} \tag{4.4}$$

$$P_{ii} = 0 \implies A_{ii}\ ignored \qquad \text{No null moves} \tag{4.5}$$

Where the last point is worth focusing on since:

$$Q_{ii} = 1 - \sum_{j \neq i} Q_{ji} = \mathbb{P}[rejection] \tag{4.6}$$

Is implied in this setting.

Assumption 25 (Symmetric proposals). For simplicity, we assume symmetric proposals.

$$P = P^T \iff P_{ji} = P_{ij}\ \forall i, j \in \mathcal{X} \tag{4.7}$$

This can be relaxed quickly, but is out of the scope of the lecture.

Definition 26 (Distance change Δ_{ji}). $\Delta_{ji} \forall i, j \in \mathcal{X}$ is the change in the Energy of interest scaled by the Temperature T. The reader can think both in terms of u or $E(\cdot), T$.

$$\Delta_{ji} :== u_j - u_i = -\left(\frac{E(r_j) - E(r_i)}{T}\right) \tag{4.8}$$

4.1 Building A

Going back to how we defined the distribution in Equation 3.7 and using the detailed balance condition of Equation 3.15 we have that:

$$Q_{ji}\rho_i = P_{ji}A_{ji}\rho_i = P_{ij}A_{ij}\rho_j = Q_{ij}\rho_j \qquad \text{using PA split of Def 23 and DBC of Def 19} \tag{4.9}$$

$$\iff A_{ji}\rho_i = A_{ij}\rho_j \qquad \text{by symmetric proposals, Assumption 25} \tag{4.10}$$

$$\iff A_{ji}\rho_i = A_{ji}\frac{e^{\frac{-E(r_i)}{T}}}{Z} = A_{ij}\frac{e^{\frac{-E(r_j)}{T}}}{Z} = A_{ij}\rho_j \qquad \text{by Boltzmann Distribution, Thm 13} \tag{4.11}$$

$$\iff ln(A_{ji}) - \frac{E(r_i)}{T} = ln(A_{ij}) - \frac{E(r_j)}{T} \qquad \text{Moving A to the exponent and equating powers} \tag{4.12}$$

Definition 27 (A Design). Now, the equation does not tell us anything yet. Indeed, here the design portion of the procedure comes into play. We can choose to build an acceptance matrix A with the easiest settings, where it is defined by an auxiliary function $h(\cdot)$ dependent on the energy change and the Temperature only, $h(\cdot) = f(\Delta_{ji})$ for some $i, j \in \mathcal{X}$. The choice for A is clever as it simplifies calculations:

$$\forall i, j \in \mathcal{X} \;\; A_{ji} = exp\left\{\frac{1}{2}(\Delta_{ji} - h(\Delta_{ji}))\right\} \tag{4.13}$$

Now this can serve to go on in this analysis with:

$$A_{ji} = exp\left\{\frac{1}{2}(\Delta_{ji} - h(\Delta_{ji}))\right\} \qquad \text{Where } A_{ji} \in [0,1] \text{ is a probability} \tag{4.14}$$

$$\iff ln(A_{ji}) = \frac{1}{2}((\Delta_{ji} - h(\Delta_{ji})) \leq 0 \qquad \text{And thus it has a negative logarithm} \tag{4.15}$$

$$\iff h(\Delta_{ji}) \geq \Delta_{ji} \qquad \text{setting the value of the logarithm to negative} \tag{4.16}$$

Then, noting that $\Delta_{ij} = -\Delta_{ji}$ our condition at equation 4.12 becomes:

$$\frac{1}{2}\Big[\Delta_{ji} - h(\Delta_{ji})\Big] - \frac{E(r_i)}{T} = \frac{1}{2}\Big[\Delta_{ij} - h(\Delta_{ij})\Big] - \frac{E(r_j)}{T} \tag{4.17}$$

$$\iff \frac{1}{2}\Big[\Delta_{ji} - h(\Delta_{ji})\Big] - \frac{E(r_i)}{T} = \frac{1}{2}\Big[-\Delta_{ji} - h(-\Delta_{ji})\Big] - \frac{E(r_j)}{T} \tag{4.18}$$

$$\iff \frac{1}{2}\Big(\Delta_{ji} - h(\Delta_{ji}) + \Delta_{ji} + h(-\Delta_{ji})\Big) - \Delta_{ji} = 0 \tag{4.19}$$

$$\iff h(\Delta_{ji}) = h(-\Delta_{ji}) \tag{4.20}$$

Where Equations 4.16 and 4.20 together make the function even and always less than its argument. This double restriction can be expressed as:

$$h(\Delta_{ji}) \geq |\Delta_{ji}| \tag{4.21}$$

Definition 28 (Metropolis Rule). In terms of practice, the most widely used proposal auxiliary function is this same restriction and is called Metropolis Rule.

$$h(\Delta_{ji}) = |\Delta_{ji}| \tag{4.22}$$

Theorem 29 (Metropolis Rule Properties). If the rule introduced in Definition 28 is used for a matrix A then $\forall i, j \in \mathcal{X}$:

$$A_{ji} = min\left\{1, \frac{\rho_j}{\rho_i}\right\} = min\left\{1, \frac{\mathbb{P}(r_{candidate})}{\mathbb{P}(r_{current})}\right\} \tag{4.23}$$

Proof.

$$A_{ji} = exp\Big\{\frac{1}{2}(\Delta_{ji} - |\Delta_{ji}|)\Big\} \qquad \text{applying Definition 28} \tag{4.24}$$

$$\iff \begin{cases} e^0 = 1 \; if \; \Delta_{ji} \geq 0 \\ e^{\Delta_{ji}} = exp\{-\frac{\Delta E_{ji}}{T}\} \; if \; \Delta_{ji} < 0 \end{cases} \qquad \text{Expanding the modulus} \tag{4.25}$$

$$\iff A_{ji} = min\Big\{1, e^{\Delta_{ji}}\Big\} \qquad \text{considering both cases of the modulus} \tag{4.26}$$

$$\iff A_{ji} = min\Big\{1, \frac{\rho_j}{\rho_i}\Big\} \qquad \text{Explained below} \tag{4.27}$$

Where the last passage comes from the fact that:

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

□

As hoped, whenever a move is beneficial in terms of reduced distance we accept it, while in the opposite case acceptance depends on the relative change and decays quickly [10]. In any non-decreasing-distance proposal, the probability of acceptance depends on T.

Theorem 30 (T extreme cases). We have that for $T \to \infty$ the distribution is perfectly uniform, while for $T \to 0$ the curve concentrates around the minimum.

Proof. If $T \to \infty$ then:

$$\lim_{T\to\infty} \frac{exp\Big(\frac{-E(r_j)}{T}\Big)}{Z} = \lim_{T\to\infty} \frac{exp\Big(\frac{-E(r_j)}{T}\Big)}{\sum_{r_k\in\mathcal{R}} exp\Big(\frac{-E(r_k)}{T}\Big)} \tag{4.28}$$

$$= \frac{1}{\sum_{r_k\in\mathcal{R}} 1} \tag{4.29}$$

$$= \frac{1}{|\mathcal{R}|} \tag{4.30}$$

$$\implies \forall i, j \in \mathcal{X} \; A_{ji} = min\Big\{1, e^{\Delta_{ji}}\Big\} \to min\Big\{1, e^0\Big\} = 1 \tag{4.31}$$

This is equivalent to always accepting with a uniform distribution on $\mathcal{X}$.

[10] being inside an exponent

If $T \to 0$ instead:

$$\lim_{T\to 0} \frac{exp\left(\frac{-E(r_j)}{T}\right)}{Z} = \lim_{T\to 0} \frac{exp\left(\frac{-E(r_j)}{T}\right)}{\sum_{r_k\in\mathcal{R}} exp\left(\frac{-E(r_k)}{T}\right)} \tag{4.32}$$

$$= \lim_{T\to 0} \frac{exp\left(\frac{-E(r_j)}{T}\right)}{exp\left(\frac{-E(r_{min})}{T}\right) + \sum_{r_k\neq r_{min}} exp\left(\frac{-E(r_k)}{T}\right)} \tag{4.33}$$

$$= \lim_{T\to 0} \frac{\frac{exp\left(\frac{-E(r_j)}{T}\right)}{exp\left(\frac{-E(r_{min})}{T}\right)}}{\frac{exp\left(\frac{-E(r_{min})}{T}\right)}{exp\left(\frac{-E(r_{min})}{T}\right)} + \frac{\sum_{r_k\neq r_{min}} exp\left(\frac{-E(r_k)}{T}\right)}{exp\left(\frac{-E(r_{min})}{T}\right)}} \tag{4.34}$$

Were we just collected the minimum distance configuration for all elements.

$$\iff \lim_{T\to 0} \frac{exp\left(\frac{-(E(r_j)-E(r_{min}))}{T}\right)}{1 + \sum_{r_k\neq r_{min}} exp\left(\frac{-(E(r_k)-E(r_{min}))}{T}\right)} \tag{4.35}$$

$$= \begin{cases} 1 \; if \; r_j = r_{min} \\ 0 \; otherwise \end{cases} \tag{4.36}$$

□

Observation 31 (On Theorem 30). In the case in which $T \to \infty$ we would randomly walk around the space of configurations, in the case in which $T \to 0$ we would only accept moves that decrease the energy. None of the two options are feasible in a complex optimization problem. As we will later see.

Below, two graphical examples of the extreme cases[11]:

[11] Credits: Bocconi University, Computer Programming, 30509. Lucibello C., Baldassi C.

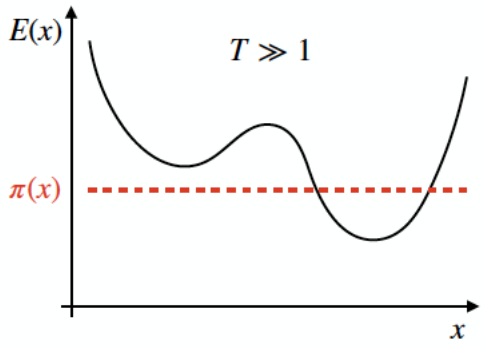

Figure 1: Energy and distribution plot at $T \to \infty$

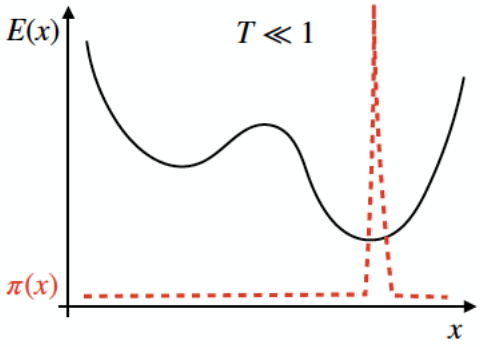

Figure 2: Energy and distribution plot at $T \to 0$

4.2 Building P

What is missing is just an efficient way to evaluate a candidate move. As observed at the beginning, we would never propose moves that are null, but we can go further than this.
P needs to produce a distribution of valid candidates. Restricting ourselves to the TSP problem, for each candidate, it must hold that the tour:

- 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

All together these requirements, other than being dependent on the current configuration[12] r_{curr}, restrict the possible options to a smaller set $\mathcal{R}_{valid}(r_{curr}) \subseteq \mathcal{R}$.

[12] again we notice that the Chain is not homogeneous

Definition 32 (An efficient P for TSP). For the TSP problem an easy and intuitive approach is to propose a swap of cities

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

That satisfies the requirements. Under random sampling and appropriate checking of the candidate, we basically sample randomly from $\mathcal{R}_{valid}(x) \forall r \in \mathcal{R}$ configurations [13]. Provided that the move is valid, the change in energy will simplify to:

$$\Delta E = E(r_{cand}) - E(r_{curr}) = d_{iv} + d_{jr} - d_{ij} - d_{jr} \tag{4.37}$$

As all the other distances are the same and cancel out. Evidently, this is easy to evaluate.

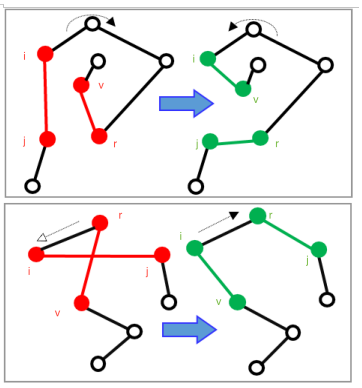

Figure 3: An example of a city swap proposal

Definition 33 (Kernel $k(\cdot|r)$). From now onwards, we will refer to P as the kernel of r $k(\cdot|r)$

[13] It is efficient to check for the validity of these

5 Algorithm Design

Observation 34 (One T is not enough). Before it was claimed that there is no optimal T choice for a complex optimization problem. These three claims will clear any doubt:

- 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.

Definition 35 (Informal Simulated Annealing (SA)). Simulated Annealing is an approach that finds a balance between the extremes of Theorem 30, 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***

We exploit the concept in this particular environment to attempt to explore and then sequentially reach a minimum.

Definition 36 (SA problem). We are given a general minimization problem such as:

$$r^* = \underset{\mathcal{R}}{argmin}\{E(r)\} \tag{5.1}$$

Definition 37 (Temperature Schedule T). Given a sequence of natural numbers $\{1, \dots, t_{max}\} \subset \mathbb{N}$:

$$T \ : \ \{1, \dots, t_{max}\} \to [0, \infty) \quad : \quad \forall c' > c \ \ T(c') \leq T(c) \ \ T(0) = \infty, \ T(t_{max}) = 0 \tag{5.2}$$

T is thus a decreasing function in the region.
Thanks to it, we can choose a random starting configuration r_0, and for a given number of iterations $t_{max} \in \mathbb{N}$ explore the space $\mathcal{R}$ with different selectiveness granularities.

Observation 38 (On Definition 37). The requirements at the extremes are not necessarily enforced.
Fine tuning of the function could present better performances.
Notwithstanding, running the algorithm for some time with $T = 0$ at the end would ensure that the local minima is reached for the last valley encountered.

Given this setting we can construct a procedure of this kind:

Algorithm 4 Simulated Annealing

Require: r_0 and $E(\cdot)$ specification as in 5.1

Require: t_{max} and $T(\cdot)$ in accordance with t_{max} as in 5.2

Require: $k(\cdot|\cdot)$

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

for $i = 1, \dots, t_{max}$ **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, use Eq 4.37 for TSP

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

if $u_i \leq min\left\{1, \frac{\Delta E}{t_i}\right\}$ **then** ▷ Metropolis rule Def 28

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

end if ▷ otherwise r is unchanged

end for

return x

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

Using Boltzmann Distribution and an energy-based fashion we derived an efficient procedure to sample from a Markov Chain. Provided that the parameters are well tuned, it will reach a satisfactory configuration. This ensures competitive solutions to combinatorially exploding problems such as TSP.

Ideally, Simulated Annealing is capable of escaping local minimas thanks to its adapting temperature, as shown in the picture below[14]:

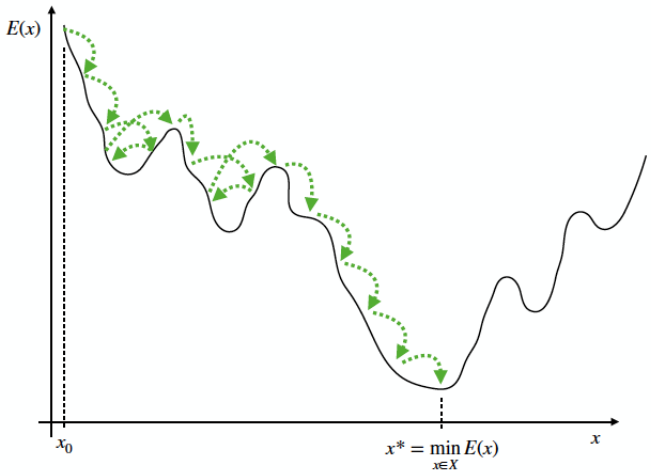

Figure 4: Simulated Annealing desired behavior

[14]Credits: Bocconi University, Computer Programming, 30509. Lucibello C., Baldassi C.

6 Conclusions

On the importance of the Detailed Balance Condition

The Detailed Balance Condition (DBC) of Definition 19 is a simplification of a stationary system with interacting elements. It guarantees specific properties, better explained by the PA split introduced in Definition 23. This perspective, joined with DBC, allows for the derivation of easy to apply acceptance rules, of which we only mention the most widely used, the Metropolis rule of Definition 28. This is a well-designed system, able to explore non convex spaces thanks to its flexibility.

Limitations

On the other hand, it requires fine tuning of the temperature schedule T which is highly dependent on the application, or in other words, the shape of the function.

Being a metaheuristic method, it does not ensure finding an optimal solution, and is thus not generalizable to different problem specifications.

In addition to this, it is believed to be less efficient in flat energy problems where less complicated methods such as Gradient Descent present a more favourable trade off.

In conclusion, Simulated Annealing is an adaptive technique that can be implemented in a wide range of difficult problems such as the Travelling Saleman Problem, for which we gave basic intuitions for a solving method.

A Markov Chain Redux

Definition 39 ($\{X_t\}$ Markov Chain (MC)).

$$\{X_t\} \; is \; MC \iff X_t | X_{t-1} \perp\!\!\!\perp \{X_{t-2}, \ldots\} \; \forall t \tag{A.1}$$

Basically, X_t is influenced by its past state only.

Definition 40 (Transition matrix $Q^{(t)}$).

$$Q^{(t)} \coloneqq \{p_{ji}(t) \coloneqq \mathbb{P}[X_t = j | X_{t-1} = i, \, t] \; \forall i, j \in \mathcal{R}\} \tag{A.2}$$

Where, noticing the order of the indices, we can claim that:

$$\sum_{k \in \mathcal{X}} Q_{ki} = 1 \tag{A.3}$$

As starting from a route i the transition *must* reach another one.

Theorem 41 (Charachterizing an MC). MC is perfectly characterized by the couple:

$$\{X_0, \, Q^{(t)}\} \tag{A.4}$$

Definition 42 (Weakly Stationary Distribution). $\{X_t\} = \{X_1, \ldots, X_n\} \; : \; X_i \in \mathcal{X}$ is said to be weakly stationary if the following conditions hold:

$$\mathbb{E}[X_t] = \mu \perp\!\!\!\perp t \tag{A.5}$$

$$V[X_t] = \sigma^2 \perp\!\!\!\perp t \tag{A.6}$$

$$CoV[X_t, X_{t'}] = \gamma_h \; : \; h = |t - t'| \tag{A.7}$$

Definition 43 (Strongly Stationary Distribution). $\{X_t\} = \{X_1, \ldots, X_{t_{max}}\} \; : \; X_i \in \mathcal{X}, \, X \sim \rho(\cdot)$ is said to be strongly stationary if the following condition holds:

$$\exists Q : Q\rho = \rho \tag{A.8}$$

Ergo, the distribution is invariant to a transition matrix.

Definition 44 (Accessibility $\rightarrow$ & Communication Relation $\leftrightarrow$).

- j is accessible from i and we write $i \rightarrow j$ if $\exists t \in \mathbb{N} \; : \; p_{ji}(t) > 0$
- j and i communicate if $j \rightarrow i \wedge i \rightarrow j$ and we write $i \leftrightarrow j$

Theorem 45 ($\leftrightarrow$ Communication Property).

$$\leftrightarrow \text{ is an equivalence relation} \tag{A.9}$$

Definition 46 (Closet states set $\mathcal{C}$).

$$\mathcal{C} \coloneqq \{i \in \mathcal{X} \ : \ i \not\rightarrow j \ \forall j \notin \mathcal{C}\} \tag{A.10}$$

Definition 47 (Types of states).

- i recurrent $\iff \exists t \geq 1 \ : \ p_{ii}(t) = 1$
- i transient $\iff \exists t \geq 1 \ : \ p_{ii}(t) < 1$
- i periodic $\iff \exists t \geq 1, \mathcal{T} \ : \ t \, mod \, \mathcal{T} = 0 \ \wedge \ p_{ii}(t) = 1$

Theorem 48 (Facts about states types).

$$i \ recurrent \wedge i \leftrightarrow j \implies j \ recurrent \tag{A.11}$$

$$i \ transient \wedge i \leftrightarrow j \implies j \ transient \tag{A.12}$$

$$\mathcal{X} = \bigcup \mathcal{X}_i \ : \ \mathcal{X}_i = \{transient \ states\} \vee \mathcal{X}_i = \{recurrent \ states\} \tag{A.13}$$

$$\forall \mathcal{X}_i = \{transient \ states\} \ \mathcal{X}_i \ is \ closed \tag{A.14}$$

Definition 49 (Irreducible MC).

$$\{X_t\} \ irreducible \iff \forall i, j \in \mathcal{X} \ i \leftrightarrow j \tag{A.15}$$

Theorem 50. MC properties with chain types

$$MC : |\mathcal{X}| < \infty \implies \exists i \in \mathcal{X} \ recurrent \tag{A.16}$$

$$MC : |\mathcal{X}| < \infty \wedge irreducible \implies \forall i \in \mathcal{X} \ i \ recurrent \tag{A.17}$$

$$MC : |\mathcal{X}| < \infty \wedge \nexists \mathcal{C} \subset \mathcal{X} \implies MC \ irreducible \tag{A.18}$$

Definition 51 (mean recurrence time m_i).

$$m_i \coloneqq \sum n f_{ii}(n) \ : \ f_{ii}(n) \coloneqq \mathbb{P}[X_n = i, X_{n-1} \neq i, \dots X_1 \neq i | X_0 = i] \tag{A.19}$$

Namely the expectation of the minimum time to reach again a state starting from it.

Definition 52 (Null and Non-null states).

- i null $\iff m_i = \infty$
- i non-null $\iff m_i < \infty$

p_{ii}	m_i	type
< 1	∞	transient
1	∞	null recurrent
1	$< \infty$	non null recurrent

Table 1: States classification

Definition 53 (Ergodic state).

$$i \in \mathcal{X} \; ergodic \iff i \; recurrent \; aperiodic \; non \; null \tag{A.20}$$

References

[1] 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).

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

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

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

[5] *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=5088E0AF37C6C017686E01FD171CA1756D9951F78D9B65F40E29AEC268535ED86760256B789
originRegion=eu-west-1&originCreation=20220114002155 (visited on 01/14/2022).

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