Stochastic Simulation
Simulated annealing

Bo Friis Nielsen

Institute of Mathematical Modelling

Technical University of Denmark

2800 Kgs. Lyngby – Denmark

Email: bfni@dtu.dk

A general optimisation problem

A general optimisation problem

- $f^{\star}$

A general optimisation problem

- $f^{\star} = \min_x$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_x$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$

A general optimisation problem

- $f^\star = \min_{x\in\mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x\in\mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} =$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S}$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S} | f(x) = f^\star\}$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star =$ argmin$_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S} | f(x) = f^\star\}$ Assume $|\mathcal{M}|$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S} | f(x) = f^\star\}$ Assume $|\mathcal{M}| < \infty$, that is the cardinality of $\mathcal{M}$ (number of elements in $\mathcal{M}$) is finite

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S} | f(x) = f^\star\}$ Assume $|\mathcal{M}| < \infty$, that is the cardinality of $\mathcal{M}$ (number of elements in $\mathcal{M}$) is finite
- This will typically be the case for discrete optimisation,

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S} | f(x) = f^\star\}$ Assume $|\mathcal{M}| < \infty$, that is the cardinality of $\mathcal{M}$ (number of elements in $\mathcal{M}$) is finite
- This will typically be the case for discrete optimisation, where also $|\mathcal{S}|$

A general optimisation problem

- $f^\star = \min_{x \in \mathcal{S}} f(x)$
- The set $\mathcal{S}$ can be quite general
- $x^\star = \text{argmin}_{x \in \mathcal{S}} f(x)$
- Note $x^\star$ might not be unique so we can define the set $\mathcal{M}$ of minimising points
- $\mathcal{M} = \{x \in \mathcal{S} | f(x) = f^\star\}$ Assume $|\mathcal{M}| < \infty$, that is the cardinality of $\mathcal{M}$ (number of elements in $\mathcal{M}$) is finite
- This will typically be the case for discrete optimisation, where also $|\mathcal{S}| < \infty$.

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x)$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) \quad = \qquad e^{-f(x)/T}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) = \frac{e^{-f(x)/T}}{\sum_y}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) \quad = \quad \frac{e^{-f(x)/T}}{\sum_{y\in}}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) \quad = \quad \frac{e^{-f(x)/T}}{\sum_{y \in \mathcal{S}}}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) = \frac{e^{-f(x)/T}}{\sum_{y \in \mathcal{S}} e^{-f(y)/T}}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) = \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \qquad e^{-f(x)/T}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) \quad = \quad \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) \quad = \quad \frac{e^{-f(x)/T}}{\sum_{y \in \mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M| e^{-f^\star/T}}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) \quad = \quad \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}}}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) = \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$\begin{aligned} P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\ &= \frac{e^{(f^\star-f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star-f(y))/T}} \end{aligned}$$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$
\begin{aligned}
P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\
&= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}}
\end{aligned}
$$

- we have a probability function with an

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) = \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}}$$

$$= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}}$$

- we have a probability function with an “easy” to calculate expression

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$P_T(x) = \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}}$$

$$= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}}$$

- we have a probability function with an "easy" to calculate expression multiplied with a difficult to calculate constant

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$\begin{aligned} P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\ &= \frac{e^{(f^\star-f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star-f(y))/T}} \end{aligned}$$

- we have a probability function with an "easy" to calculate expression multiplied with a difficult to calculate constant
- For fixed T we can sample, states x

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$
\begin{aligned}
P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\
&= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}}
\end{aligned}
$$

- we have a probability function with an “easy” to calculate expression multiplied with a difficult to calculate constant
- For fixed T we can sample, states x with low “energy” (low valuels of $f(x)$) will be more frequent/likely

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$
\begin{aligned}
P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\
&= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}}
\end{aligned}
$$

- we have a probability function with an “easy” to calculate expression multiplied with a difficult to calculate constant
- For fixed T we can sample, states x with low “energy” (low valuels of $f(x)$) will be more frequent/likely
- As T

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$\begin{aligned} P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\ &= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}} \end{aligned}$$

- we have a probability function with an "easy" to calculate expression multiplied with a difficult to calculate constant
- For fixed T we can sample, states x with low "energy" (low valuels of $f(x)$) will be more frequent/likely
- As $T \to 0$

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$\begin{aligned} P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\ &= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}} \end{aligned}$$

- we have a probability function with an “easy” to calculate expression multiplied with a difficult to calculate constant
- For fixed T we can sample, states x with low “energy” (low valuels of $f(x)$) will be more frequent/likely
- As $T \rightarrow 0$ the distribution will degenerate

Optimisation problem - probability distribution

We introduce a probability distribution over $\mathcal{S}$ to be

$$
\begin{aligned}
P_T(x) &= \frac{e^{-f(x)/T}}{\sum_{y\in\mathcal{S}} e^{-f(y)/T}} = \frac{e^{-f(x)/T}}{|M|e^{-f^\star/T} + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{-f(y)/T}} \\
&= \frac{e^{(f^\star - f(x))/T}}{|M| + \sum_{y\in\mathcal{S}\setminus\mathcal{M}} e^{(f^\star - f(y))/T}}
\end{aligned}
$$

- we have a probability function with an "easy" to calculate expression multiplied with a difficult to calculate constant
- For fixed T we can sample, states x with low "energy" (low valuels of $f(x)$) will be more frequent/likely
- As $T \to 0$ the distribution will degenerate to states with minimum energy

Simulated annealing

Simulated annealing

- Stochastic algorithm for optimisation

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods
  - a metaheuristic

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods
  - a metaheuristic
- Simulated annealing one of the first,

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods
  - a metaheuristic
- Simulated annealing one of the first, inspired from Metropolis-Hastings

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods
  - a metaheuristic
- Simulated annealing one of the first, inspired from Metropolis-Hastings - Kirkpatrick paper Science 1983

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods
  - a metaheuristic
- Simulated annealing one of the first, inspired from Metropolis-Hastings - Kirkpatrick paper Science 1983
- Alternatives: Stochastic gradient

Simulated annealing

- Stochastic algorithm for optimisation
- Large scale (typically discrete) problems
- Attempts to find the global optimum in presence of multiple local optima

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- One among many stochastic optimisation methods
  - a metaheuristic
- Simulated annealing one of the first, inspired from Metropolis-Hastings - Kirkpatrick paper Science 1983
- Alternatives: Stochastic gradient and several other

Physical inspiration (with apologies)

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be “caught” in other states

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be “caught” in other states which are only locally stable.

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be “caught” in other states which are only locally stable.

This is likely to happen when welding, machining, etc.

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be "caught" in other states which are only locally stable.

This is likely to happen when welding, machining, etc.

By heating the material

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be "caught" in other states which are only locally stable.

This is likely to happen when welding, machining, etc.

By heating the material and **slowly** cooling,

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be "caught" in other states which are only locally stable.

This is likely to happen when welding, machining, etc.

By heating the material and **slowly** cooling, we ensure that the material ends in the ground state.

Physical inspiration (with apologies)

Steel and other materials can exist in several crystalline structures.

One - the ground state - has lowest energy.

The material may be “caught” in other states which are only locally stable.

This is likely to happen when welding, machining, etc.

By heating the material and **slowly** cooling, we ensure that the material ends in the ground state.

This process is called **annealing**.

P.d.f. of the state at fixed temperature

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics,

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x, T) =$$

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x,T) = c_T \cdot \exp\left(-\frac{U(x)}{T}\right)$$

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x, T) = c_T \cdot \exp\left(-\frac{U(x)}{T}\right)$$

So states with low U are more probable;

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x, T) = c_T \cdot \exp\left(-\frac{U(x)}{T}\right)$$

So states with low U are more probable; in particular at low T.

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x, T) = c_T \cdot \exp\left(-\frac{U(x)}{T}\right)$$

So states with low U are more probable; in particular at low T.

Note the normalization constant c_T is unknown;

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x,T) = c_T \cdot \exp\left(-\frac{U(x)}{T}\right)$$

So states with low U are more probable; in particular at low T.

Note the normalization constant c_T is unknown; can be found by integration,

P.d.f. of the state at fixed temperature

Use $X \in \mathcal{S}$ to denote the state of the system (e.g., positions of atoms).

Let $U(x)$ denote the energy of state $x \in \mathcal{S}$.

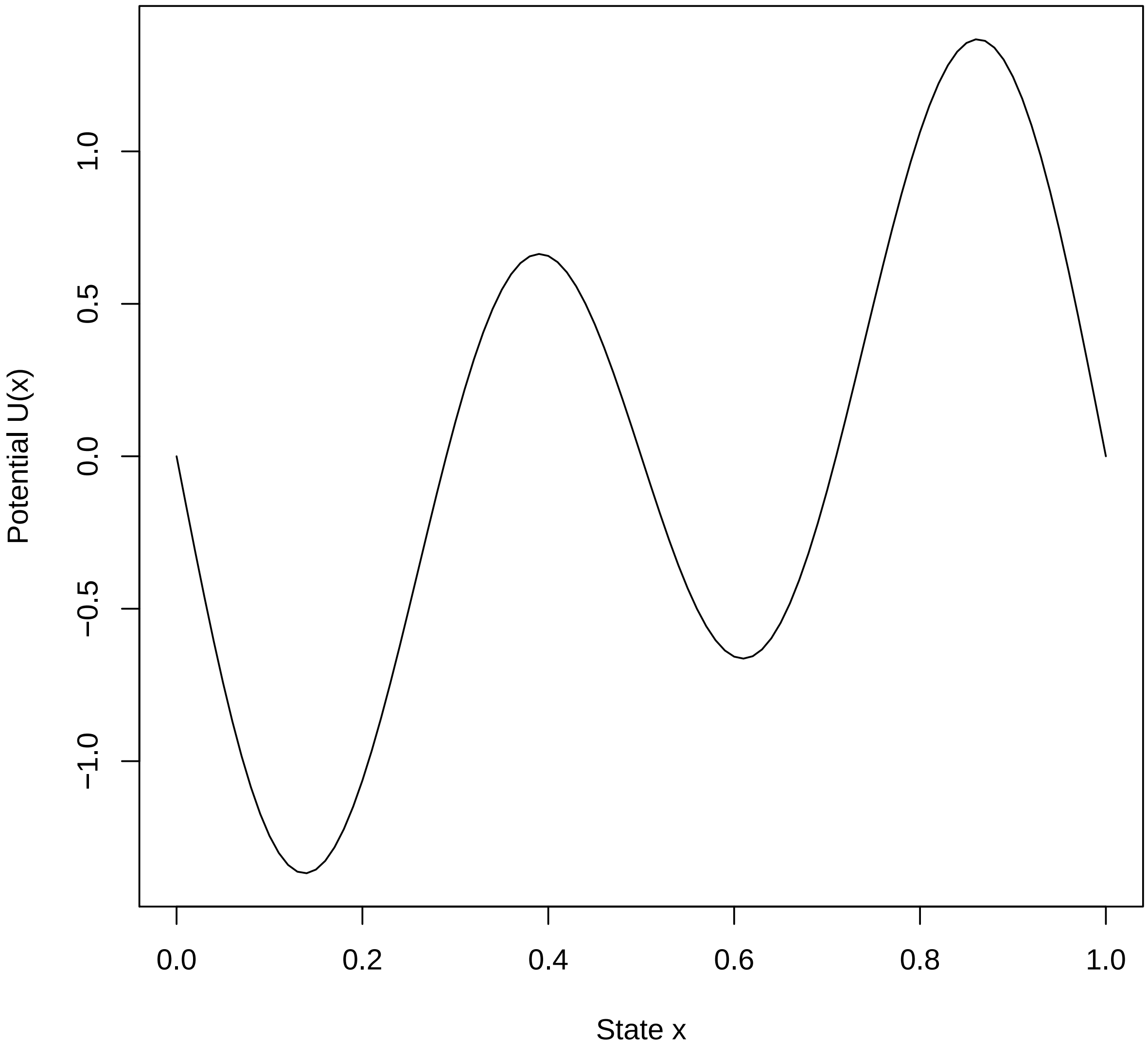
According to statistical physics, if the temperature is T, the p.d.f. of X is *the Canonical Distribution*

$$f(x,T) = c_T \cdot \exp\left(-\frac{U(x)}{T}\right)$$

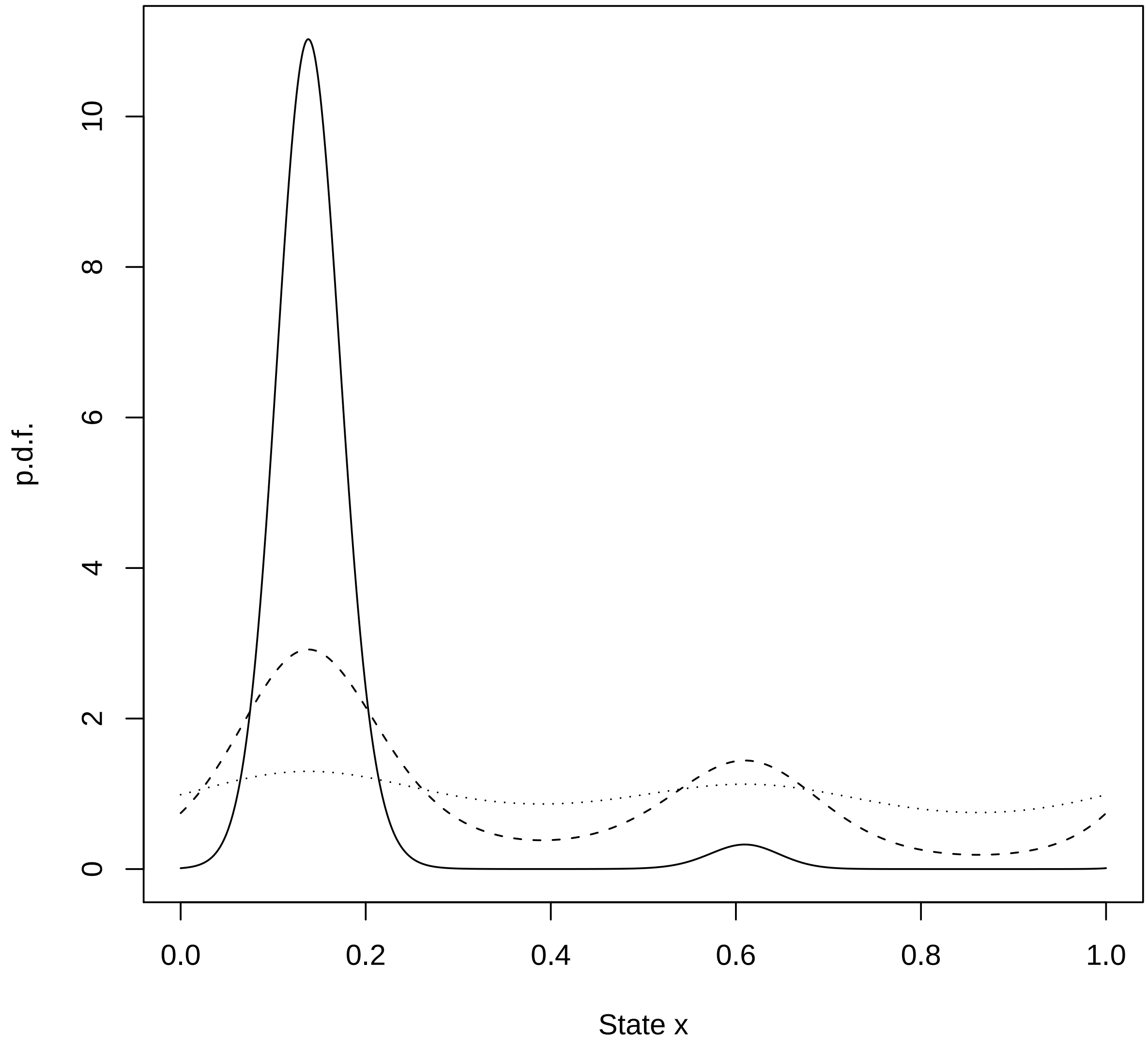
So states with low U are more probable; in particular at low T.

Note the normalization constant c_T is unknown; can be found by integration, but our algorithms will not require it.

Example energy potential



Corresponding p.d.f., for $T = 0.2, 1, 5$



An algorithm for Simulated Annealing

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC,

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f.

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly to generate a candidate Y_i.

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly to generate a candidate Y_i. If the candidate has lower energy than the old state, accept.

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly to generate a candidate Y_i. If the candidate has lower energy than the old state, accept. Otherwise, accept only with probability

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly to generate a candidate Y_i. If the candidate has lower energy than the old state, accept. Otherwise, accept only with probability

$$\exp(-(U(Y_i) - U(X_i))/T_i)$$

An algorithm for Simulated Annealing

Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly to generate a candidate Y_i. If the candidate has lower energy than the old state, accept. Otherwise, accept only with probability

$$\exp(-(U(Y_i) - U(X_i))/T_i)$$

for a symmetric proposal distribution

An algorithm for Simulated Annealing

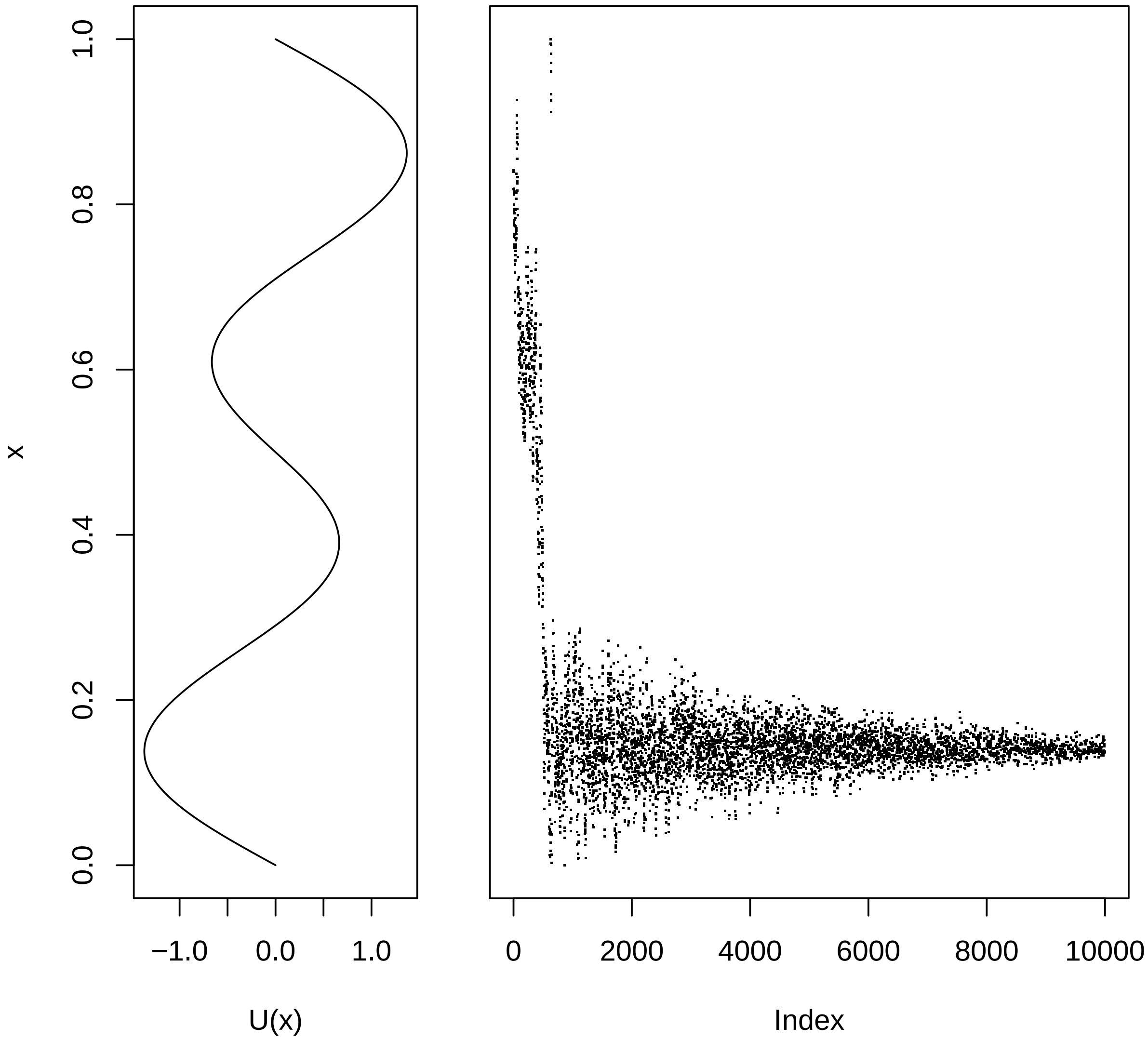
Let the temperature be a decreasing function of time or iteration number - k.

At each time step, update the state according to the random walk Metropolis-Hastings algorithm for MCMC, where the target p.d.f. is $f(x, T_i)$.

I.e., permute the state X_i randomly to generate a candidate Y_i. If the candidate has lower energy than the old state, accept. Otherwise, accept only with probability

$$\exp(-(U(Y_i) - U(X_i))/T_i)$$

for a symmetric proposal distribution (to keep the probabilistic interpreation)



1.0
0.8
0.6
0.4
0.2
0.0
x
−1.0
0.0
1.0
U(x)
0
2000
4000
6000
8000
10000
Index

Different issues

Different issues

- Try with different schemes for lowering the temperature

Different issues

- Try with different schemes for lowering the temperature
- Alternative initial solutions

Different issues

- Try with different schemes for lowering the temperature
- Alternative initial solutions
- Different candidate generation algorithms

Different issues

- Try with different schemes for lowering the temperature
- Alternative initial solutions
- Different candidate generation algorithms
- Refine with local search

Travelling salesman problem (TSP)

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations,

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations, and an n-by-n matrix A giving the cost of going from station i to j.

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations, and an n-by-n matrix A giving the cost of going from station i to j.

Find a route S (a permutation of $1, \ldots, n$) which

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations, and an n-by-n matrix A giving the cost of going from station i to j.

Find a route S (a permutation of $1, \ldots, n$) which

- starts and ends at station 1, $S_1 = 1$

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations, and an n-by-n matrix A giving the cost of going from station i to j.

Find a route S (a permutation of $1, \ldots, n$) which

- starts and ends at station 1, $S_1 = 1$
- has minimal total cost

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations, and an n-by-n matrix A giving the cost of going from station i to j.

Find a route S (a permutation of $1, \ldots, n$) which

- starts and ends at station 1, $S_1 = 1$
- has minimal total cost

$$\sum_{i=1}^{n-1}$$

Travelling salesman problem (TSP)

A basic problem in combinatorial optimisation

Given n stations, and an n-by-n matrix A giving the cost of going from station i to j.

Find a route S (a permutation of $1, \ldots, n$) which

- starts and ends at station 1, $S_1 = 1$
- has minimal total cost

$$\sum_{i=1}^{n-1} A(S_i, S_{i+1})$$

Cost matrix - an example

Town from	Town to 1	2	3	4	5	6
1	-	5	3	1	4	12
2	2	-	22	11	13	30
3	6	8	-	13	12	5
4	33	9	5	-	60	17
5	1	15	6	10	-	14
6	24	6	8	9	40	-

Cost matrix - an example

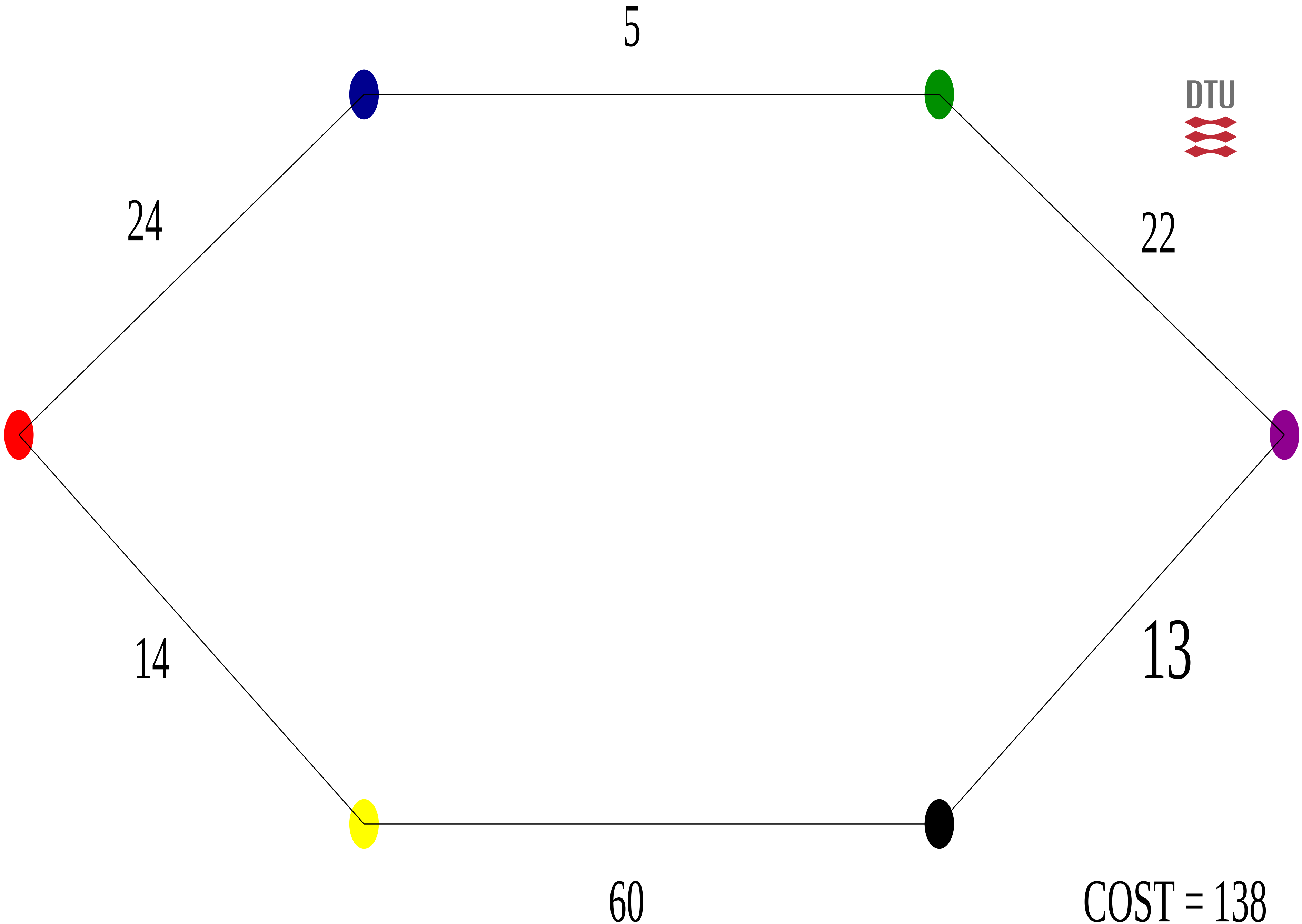
Town from	Town to 1	2	3	4	5	6
1	-	5	3	1	4	12
2	2	-	22	11	13	30
3	6	8	-	13	12	5
4	33	9	5	-	60	17
5	1	15	6	10	-	14
6	24	6	8	9	40	-

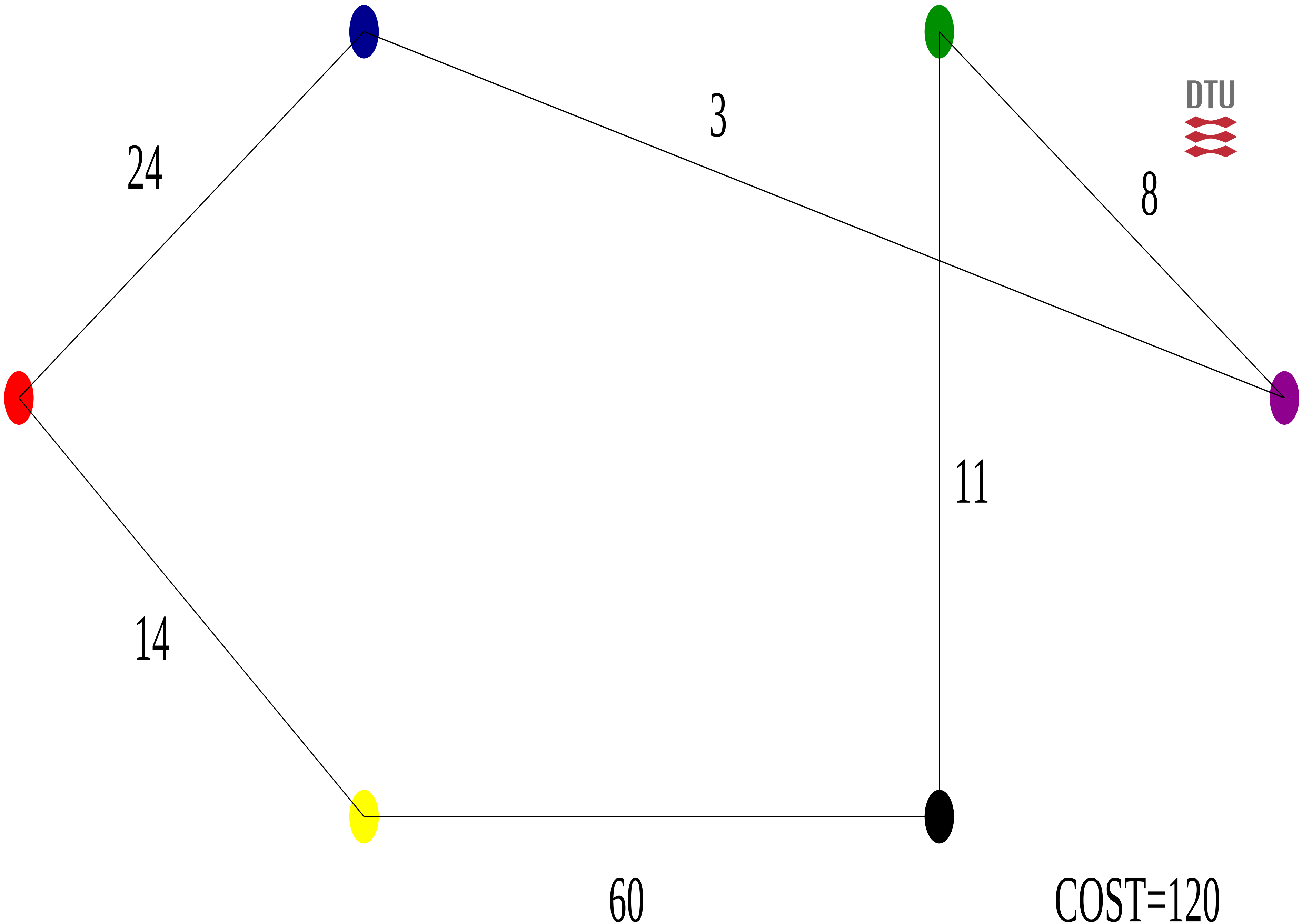
- Initial solution: $\{1, 2, 3, 4, 5, 6, 1\}$

Cost matrix - an example

Town from	Town to 1	2	3	4	5	6
1	-	5	3	1	4	12
2	2	-	22	11	13	30
3	6	8	-	13	12	5
4	33	9	5	-	60	17
5	1	15	6	10	-	14
6	24	6	8	9	40	-

- Initial solution: $\{1, 2, 3, 4, 5, 6, 1\}$ initial cost: 5+22+13+60+14+24 $=$ 138


5
24
22
14
13
60
COST = 138


24
3
8
11
14
60
COST=120

Exercise 7

1. Implement simulated annealing for the travelling salesman. As proposal, permute two random stations on the route. As cooling scheme, you can use e.g. $T_k = 1/\sqrt{1+k}$. or $T_k = -\log(k+1)$, feel free to experiment with different choices. The route must end where it started. Initialise with a random permutation of stations.
   (a) Have input be positions in the plane of the n stations.
       Let the cost of going $i \mapsto j$ be the Euclidian distance between station i and j.
       Plot the resulting route in the plane.
       Debug with stations on a circle.
   (b) Then modify your progamme to work with costs directly and apply it to the cost matrix from the course homepage.