A Graph-based Ant System and its Convergence

Siyeong Lee

Dept. of Electronic Engineering

Sogang University

April 5, 2017

Introduction

- Lots of interesting Problem: NP-Problem
- Therefore, we need to solve this problem using heuristics algorithm
    - To find high quality solution for these problems in reasonable time
    - One of properties about Heuristics algorithm
        - This kinds of algorithm may be applied to a broad class of optimization problem.
            * Not restricted to specific problem types with suitable medications.
        - For such a reason, these "general-purpose" algorithm called metaheuristics

Intuition

- How do ants find the shortest path?

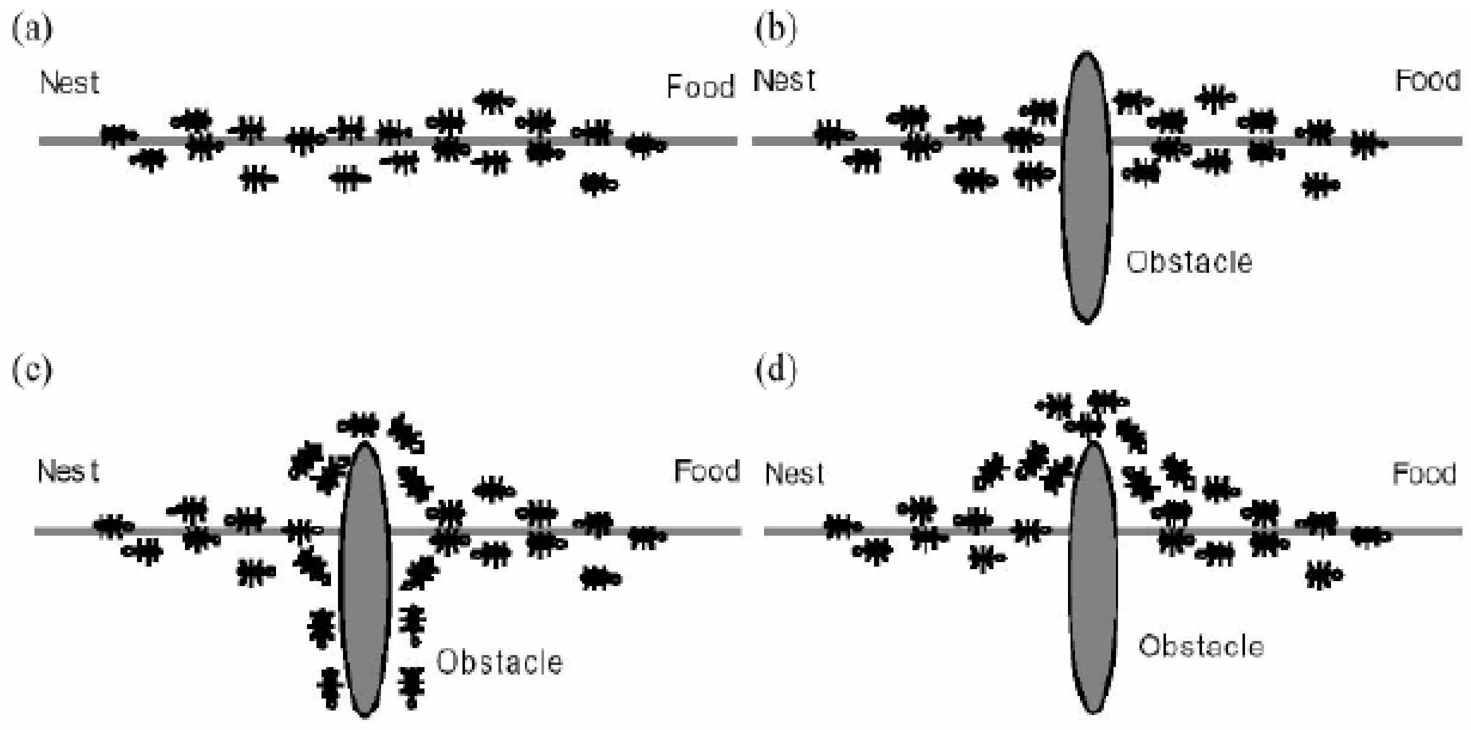


Figure 1: Ants find the shortest path around an obstacle [8]

Intuition

- How do ants find the shortest path?

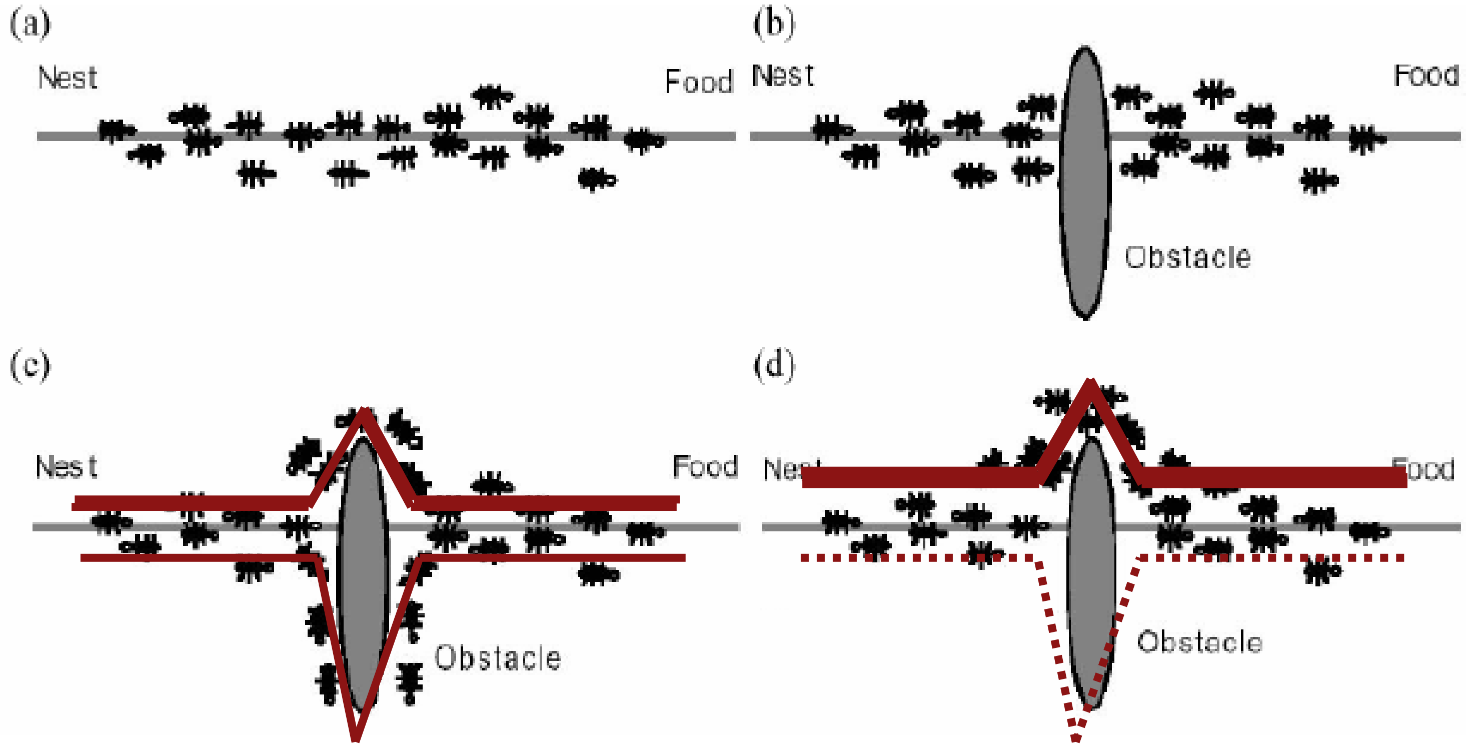


Figure 1: Ants find the shortest path around an obstacle [8]

Intuition

- Each ant leaves information on which path it has traversed by depositing a chemical substance, called pheromone, on the ground. Ants have a tendency to follow these pheromone trails. Within a fixed period, **shorter paths between nest and food** can be traversed more often than longer paths, and so they obtain **a higher amount of pheromone**
- The probability of a transition along a specific arc is computed from
  - (a) the pheromone value assigned to this arc
  - (b) the length of the arc

✓ *The higher the pheromone value and the smaller the length*
The higher is the probability that the agent follows this arc in his next move

Problem setting – Optimization problem

Definition 1. Optimization problem

$$Minimize\ f_0(x)$$

$$Subject\ to\ f_i(x) \leq b_i, i = 1, \dots, m$$

$f_0 : \boldsymbol{R}^n \to \boldsymbol{R}$; Optimization variables
$f_i : \boldsymbol{R}^n \to \boldsymbol{R}, i = 1, \dots m$: Constraint functions

$x = (x_1, \dots, x_n)$ *: Optimization variables*

Specially, solutions which satisfy all constraints is called *feasible solutions*

Combinatorial optimization

- **Combinatorial optimization** is a topic that consists of finding an optimal object from a finite of objects.[3]
- It operates on the domain of those optimization problems, in which
  - the ***set of feasible solutions is discrete*** or
  - ***can be reduced to discrete***

Goal: to *find the best solution* in feasible solutions

How to represent?

Definition 3.1.

Let $\boldsymbol{I}$ be an instance of a combinatorial optimization problem be given. By a construction graph for this instance,

We can understand a directed graph $\boldsymbol{C} = (\boldsymbol{V}, \boldsymbol{A})$ together with a function $\boldsymbol{\Phi}$ with the following properties:

(1) In $\boldsymbol{C}$, a unique node $\boldsymbol{u}$ is marked as the so-called start node.

(2) Let $\boldsymbol{W}$ be the set of directed walks $\boldsymbol{w}$ in $\boldsymbol{C}$ satisfying the following conditions:

a. $\boldsymbol{w}$ starts at the start node of $\boldsymbol{C}$.

b. $\boldsymbol{w}$ contains each node of $\boldsymbol{C}$ at most once.

c. The last node on $\boldsymbol{w}$ has no successor node in $\boldsymbol{C}$ that is not already contained in $\boldsymbol{w}$

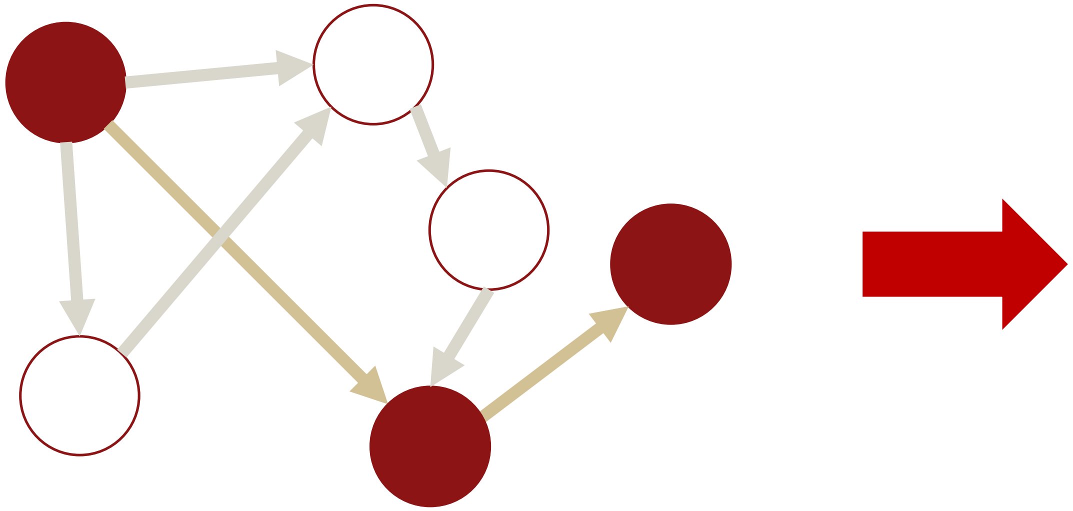
Then $\boldsymbol{\Phi}$ maps the set $\boldsymbol{W}$ onto the set of feasible solutions of $\boldsymbol{I}$

How to represent?

✓ *This graph is called "Construction graph"*

- From "***the representation of a feasible solution***" to "***a walk in a directed graph***"

Walk w



*Feasible solution **S***

Figure 2: Representation of feasible solution

How to represent feasible solutions?

- in **Construction graph**

(1) Let $\boldsymbol{W}$ be the set of directed walks w in $\boldsymbol{C}$ satisfying the following conditions:

a. w starts at the start node of $\boldsymbol{C}$.

b. w contains each node of $\boldsymbol{C}$ at most once.

c. The last node on w has no successor node in $\boldsymbol{C}$ that is not already contained in w

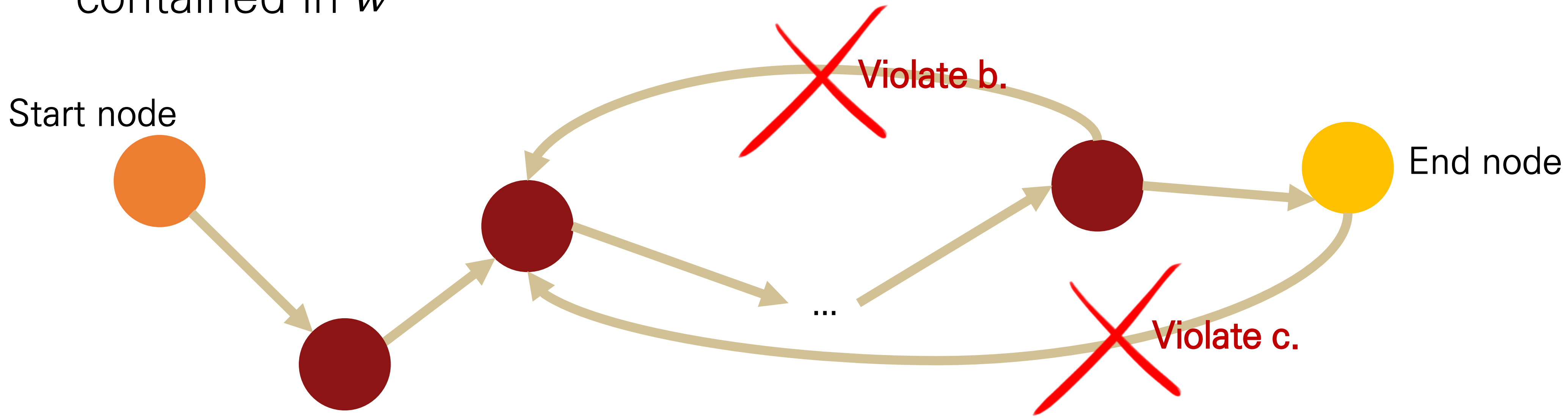
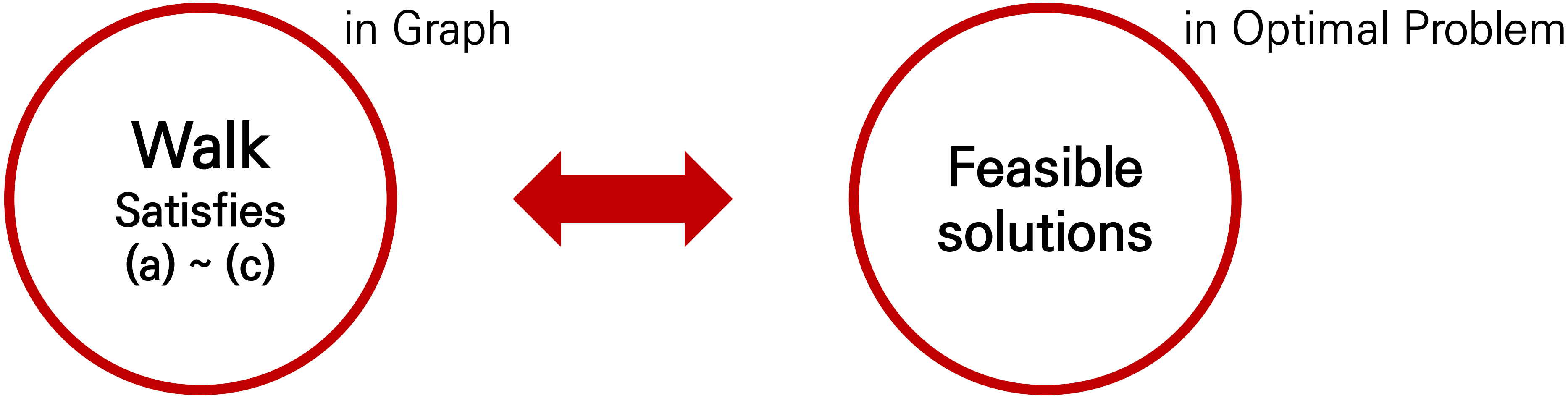


Figure 3: Relations between walks and feasible solutions

How to represent feasible solutions?

- In other words,
  - Let $\Phi : \boldsymbol{W} \rightarrow \boldsymbol{F}$,
    where W is the set of directed walk w, and F is the set of solution in I

A walk $w \in \boldsymbol{W}$ satisfies condition (a) ~ (c),
iff $\boldsymbol{\Phi}(\boldsymbol{w})$ is a feasible solution in $\boldsymbol{I}$



Representation of Objective function.

- Solution Encoding:
  - the feasible solutions is encoded as "walks" in a construction graph $(\boldsymbol{C}, \boldsymbol{\Phi})$

- **The objective function value of the walk**
  - the objective function value(or cost function in minimization problem) of the corresponding feasible solution of $\boldsymbol{I}$.

How to represent?

- Graph–based Ant System consist in **five** components.

1. A **construction graph** (C, Φ) according to Definition 3.1.

2. A **set $A_1, \ldots, A_S$ of agents** (it is called "ants").
   - Each agent performs a random walk with carefully chosen transition probabilities (see component 3 below) on the construction graph
   - A time period in which each agent performs a walk through the construction graph will be called a cycle.
   - An application of Ant System consists of several cycles $1, \ldots, M$; the number M of cycles may be fixed in advance or be determined at a later time during the execution of the algorithm.

How to represent?

- Graph−based Ant System consist in five components.
  3 Transition probabilities for the random moves of the agents during each cycle.
  - Let $u = (u_0, \dots, u_{t-1})$ denote the partial walk an agent has already traversed before its tth transition step in a fixed cycle m, where $u_0, \dots, u_{t-1}$are node indices in the construction graph (u_0 referring to the start node).
    * $l \in u$ if l is contained in the partial walk u, and $l \notin u$ otherwise.
  - The general form of the transition probabilities
    * let A be the set of arcs in the construction graph.

$$p_{kl}(m,u) = \begin{cases} \dfrac{[\tau_{kl}(m)]^{\alpha} \cdot [\eta_{kl}(u)\ ]^{\beta}}{\sum_{\{r \notin u,(k,r) \in A\}} [\tau_{kr}(m)]^{\alpha} \cdot [\eta_{kr}(u)\ ]^{\beta}} & if\ \ l \notin u \\ 0 & if\ \ l\ \in u \end{cases}$$

Where the numbers $\tau_{kl}(m)$ are called “pheromone values” (see component 4 below),
the numbers $\eta_{kl}(u)$ are called “desirability values” (see component 5 below), and α and β are parameters.

How to represent?

$$p_{kl}(m,u) = \begin{cases} \dfrac{[\tau_{kl}(m)]^{\alpha} \cdot [\eta_{kl}(u)\ ]^{\beta}}{\sum_{\{r \notin u,(k,r) \in A\}}[\tau_{kr}(m)]^{\alpha} \cdot [\eta_{kr}(u)\ ]^{\beta}} & if\ \ l \notin u \\ 0 & if\ \ l\ \in u \end{cases}$$

Note that,
this probability is only defined if $k\ =\ u_{t-1}$.
(it means this has *markov property*)

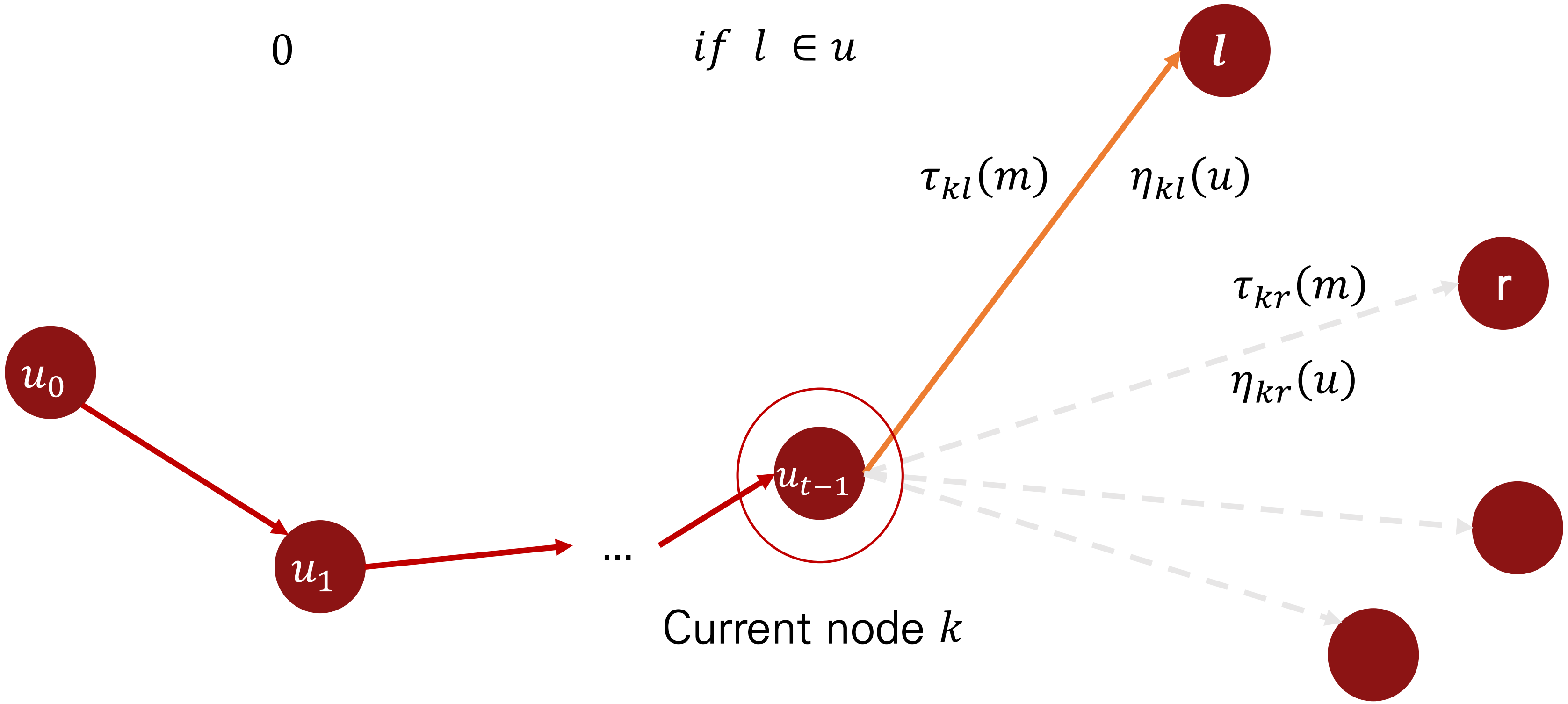


Figure 4: The way that transition probability is calculated.

How to represent?

- Graph–based Ant System consist in five components.
  4 $\tau_{kl}(m)$: **Pheromone values**
  - It depends on cycle m
    - * Initialize: at cycle 1, $\tau_{kl} = \frac{1}{number\ of\ arcs}\ \ for\ each\ arc\ (k, l).$
    - * Update: **At the end of each cycle $\boldsymbol{m}$,**
      - ✓ for each agent A_s and each arc (k, l), a value $\Delta\tau_{kl}^{(s)}$ is determined as a function of the solution assigned to the walk of A_s in the current cycle m.
      - ✓ Suppose this solution has a cost value f_s.
        - For each agents A_s and for each arc (k, l),

$$\Delta\tau_{kl}^{(s)} = \begin{cases} \varphi(f_s) & if\ \ agent\ A_s\ has\ traversed\ arc\ (k, l) \\ 0 & otherwise \end{cases}$$

        - Therein, φ is a non–increasing function which may depend on the walks of the agents in the cycles $1, \ldots, m - 1$.

How to represent?

- Let $C = \sum_{(k,l)\in\mathcal{A}} \sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}$. (4)
- $\tau_{kl}(m+1) = \begin{cases} \tau_{kl}(m) & if\ C = 0 \\ (1-\rho)\tau_{kl}(m) + \rho\Delta\tau_{kl} & if\ C > 0 \end{cases}$ (5)

  $where\ \Delta\tau_{kl} = \frac{1}{C}\sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}$ (6)
- The number ρ is usually called the evaporation factor

It is easily verified from (4)–(6) that the sum of pheromone values, P(k,l)∈Aτkl(m), always remains equal to 1.

(simply as a re-normalization which favors the numerical stability of the algorithm.)

How to represent?

- Graph–based Ant System consist in five components.
  5 $\eta_{kl}(u)$: **Desirability values**
  - It depends on the partial walk u
  - It is interpreted as the value of a so–called $greedy\ function$

    for all feasible arcs (k, l) leaving node k, and determines the next node l of the walk by the "greedy principle" that the weight of (k, l) is maximum.

  - *How to allocate?*
    * $\eta_{kl}(u) = weight(k, l)$
    * $\eta_{kl}(u) = \begin{cases} 1 & if\ weight(k,l)\ is\ maximum\ among\ all\ succesor\ node\ k \\ 0 & otherwise \end{cases}$
  - The values $\eta_{kl}(u)$ can also be used for preventing walks corresponding to infeasible solutions.

Psuedocode

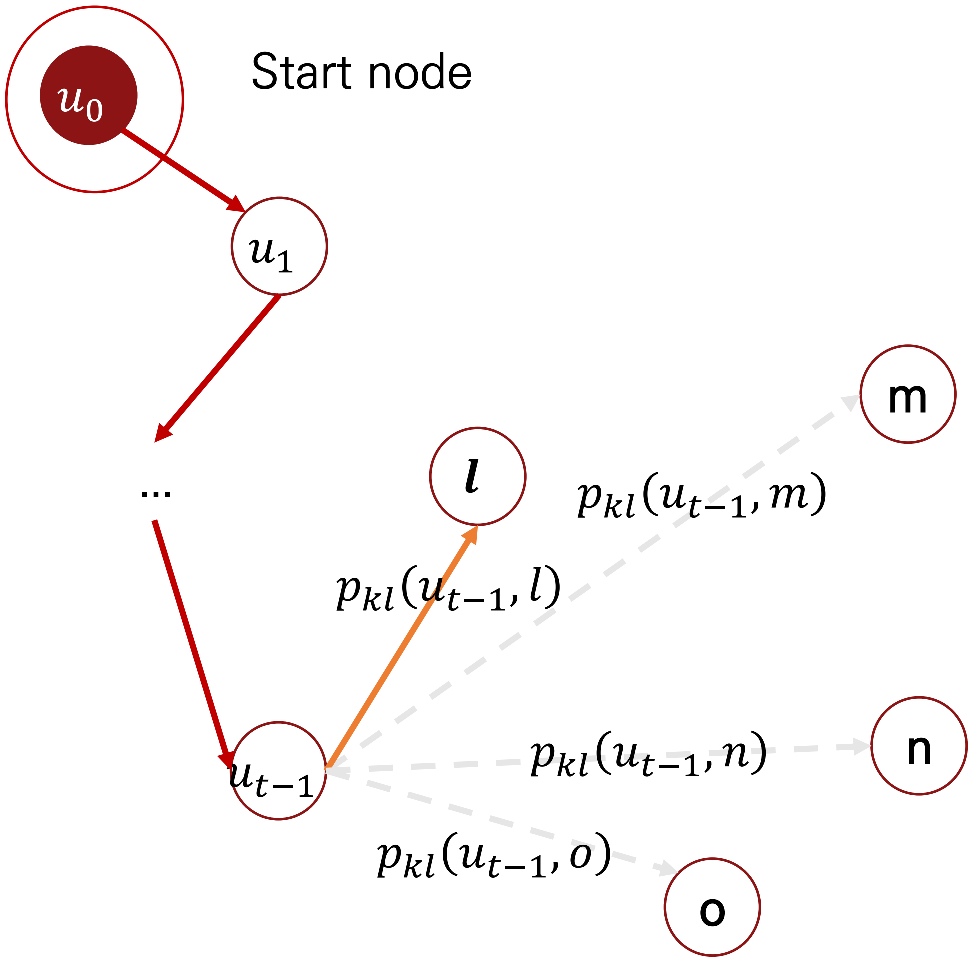
Algorithm 1. Ant colony optimization (ACO)

while termination conditions not met **do**
  ScheduleActivities
    AntBasedSolutionConstruction()
    PheromoneUpdate()
  end ScheduleActivities
end while

Algorithm 2. Procedure AntBasedSolutionConstruction() of Algorithm 1

$s = \langle\rangle$
Determine $N(s)$
while $N(s) \notin \emptyset$ **do**
  $c \leftarrow$ ChooseFrom($N(s)$)
  $s \leftarrow$ extend s by appending solution component c
  Determine $N(s)$
end while

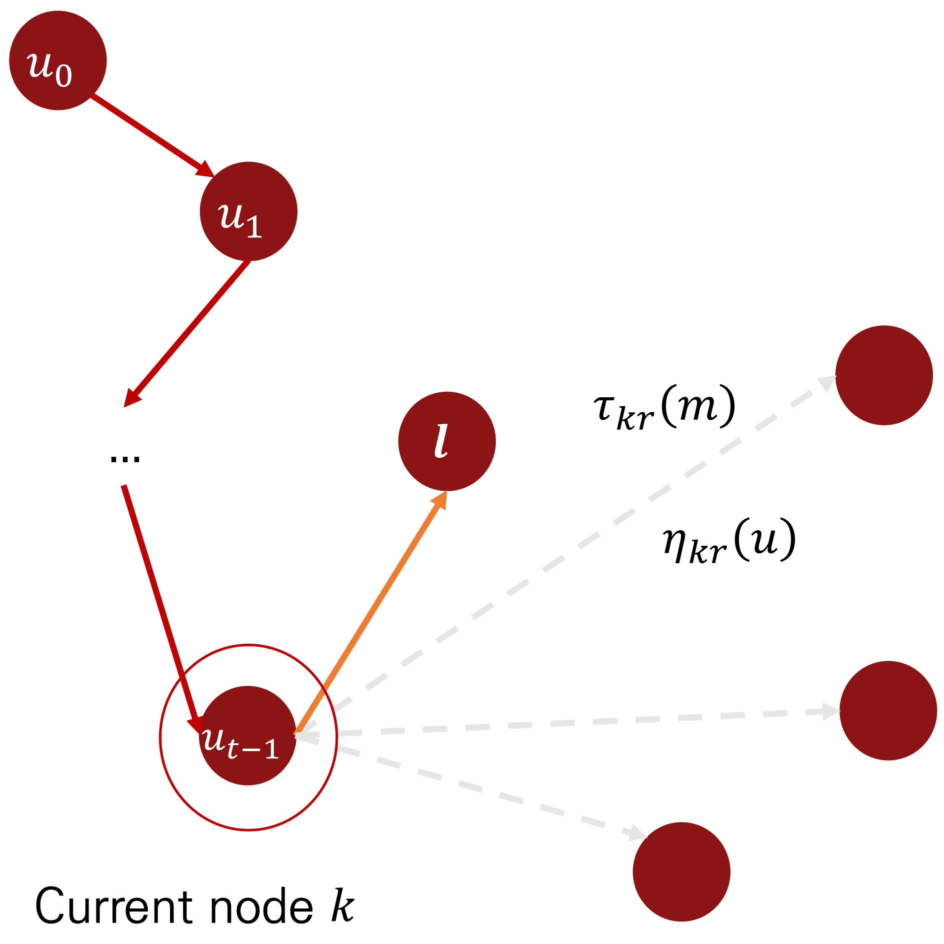
Psuedocode



Algorithm 2. Procedure AntBasedSolutionConstruction() of Algorithm 1

$s = \langle\rangle$
$\boxed{\text{Determine } N(s)}$
while $N(s) \notin \emptyset$ **do**
    $c \leftarrow$ ChooseFrom($N(s)$)
    $s \leftarrow$ extend s by appending solution component c
    Determine $N(s)$
end while

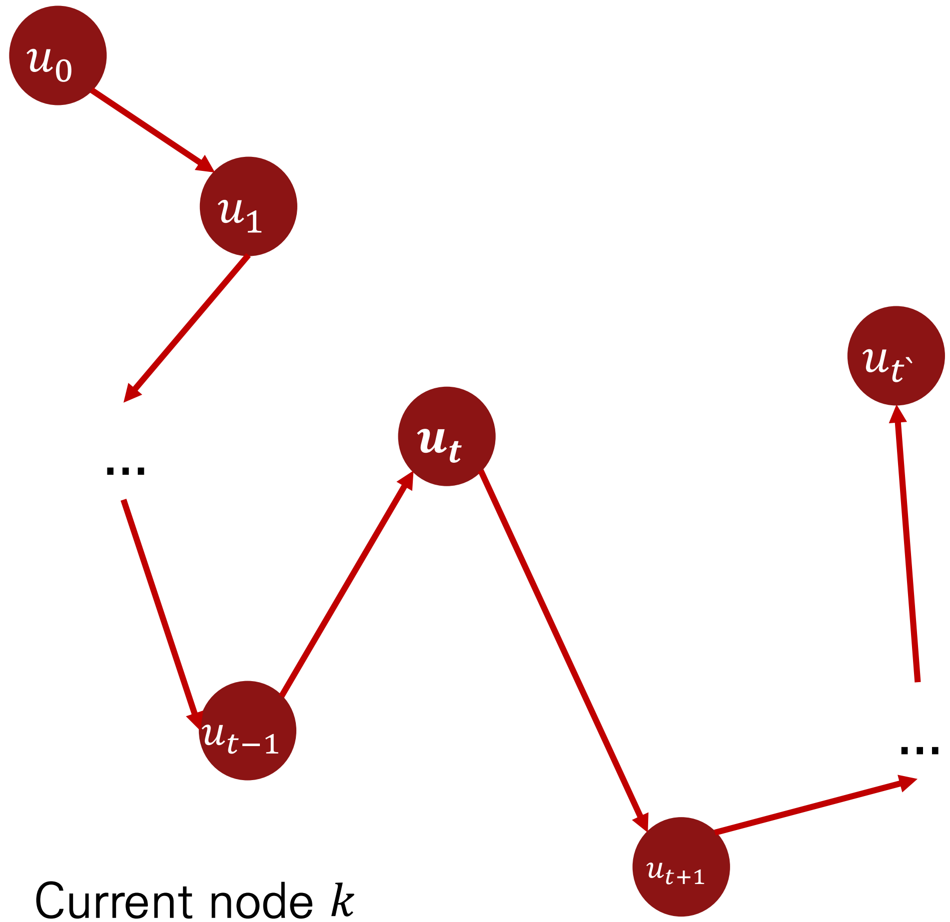
Psuedocode



Algorithm 2. Procedure AntBasedSolutionConstruction() of Algorithm 1

$s = \langle\rangle$
Determine $N(s)$
while $N(s) \notin \emptyset$ **do**
  $c \leftarrow$ ChooseFrom($N(s)$)
  $s \leftarrow$ extend s by appending solution component c
  Determine $N(s)$
end while

Psuedocode



Current node k

Algorithm 2. Procedure AntBasedSolutionConstruction() of Algorithm 1

$s = \langle\rangle$
Determine $N(s)$
while $N(s) \notin \emptyset$ **do**
 $c \leftarrow$ ChooseFrom($N(s)$)
 $s \leftarrow$ extend s by appending solution component c
 Determine $N(s)$
end while

Output: walk (it is feasible solution)

Psuedocode

Algorithm 1. Ant colony optimization (ACO)

```
while termination conditions not met do
    ScheduleActivities
        AntBasedSolutionConstruction()
        PheromoneUpdate()
    end ScheduleActivities
end while
```

for **each** ***Cycle***, this part is performed

1. **Solution Construction**
   – In this step, generate a feasible solution
2. **Pheromone Update**

Psuedocode

Algorithm 1. Ant colony optimization (ACO)

while termination conditions not met **do**
    ScheduleActivities
        AntBasedSolutionConstruction()
        PheromoneUpdate()
    end ScheduleActivities
end while

Algorithm 2. Procedure AntBasedSolutionConstruction() of Algorithm 1

$s = \langle\rangle$
Determine $N(s)$
while $N(s) \notin \emptyset$ **do**
    $c \leftarrow$ ChooseFrom($N(s)$)
    $s \leftarrow$ extend s by appending solution component c
    Determine $N(s)$
end while

Psuedocode

Algorithm 1. Ant colony optimization (ACO)

while termination conditions not met **do**
  ScheduleActivities
    AntBasedSolutionConstruction()
    PheromoneUpdate()
  end ScheduleActivities
end while

Updating Rule

Let $C = \sum_{(k,l)\in\mathcal{A}} \sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}$.

$$\tau_{kl}(m+1) = \begin{cases} \tau_{kl}(m) & if\ C = 0 \\ (1-\rho)\tau_{kl}(m) + \rho\Delta\tau_{kl} & if\ C > 0 \end{cases}$$

$$where\ \Delta\tau_{kl} = \frac{1}{C}\sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}$$

Suppose this solution has a cost value f_s.
for each arc (k, l),

$$\Delta\tau_{kl}^{(s)} = \begin{cases} \varphi(f_s) \\ 0 \end{cases}$$

where φ is a non-increase function

in Summary

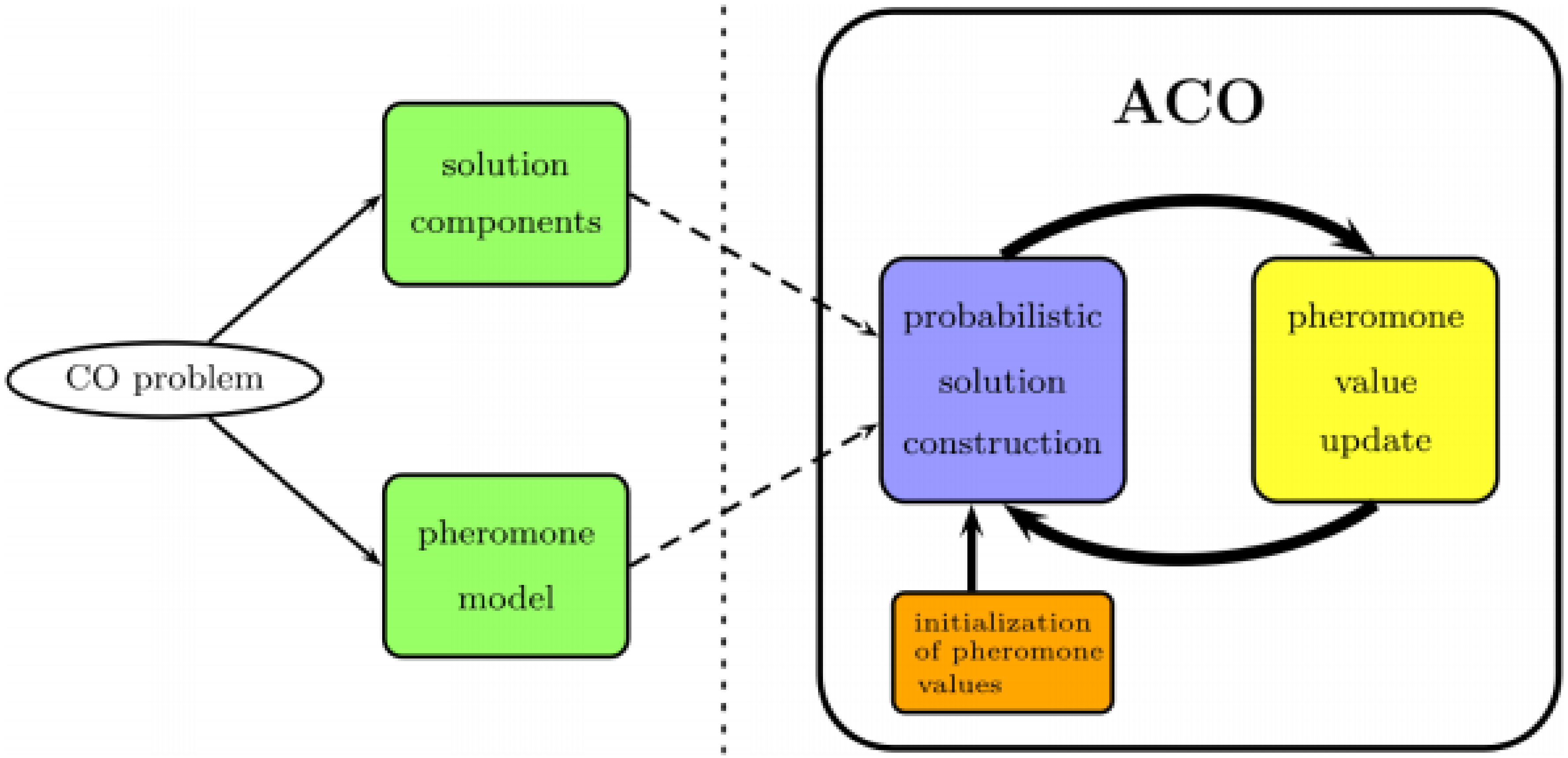


Figure 5: The working of the ACO metaheuristic. [5]

in Summary

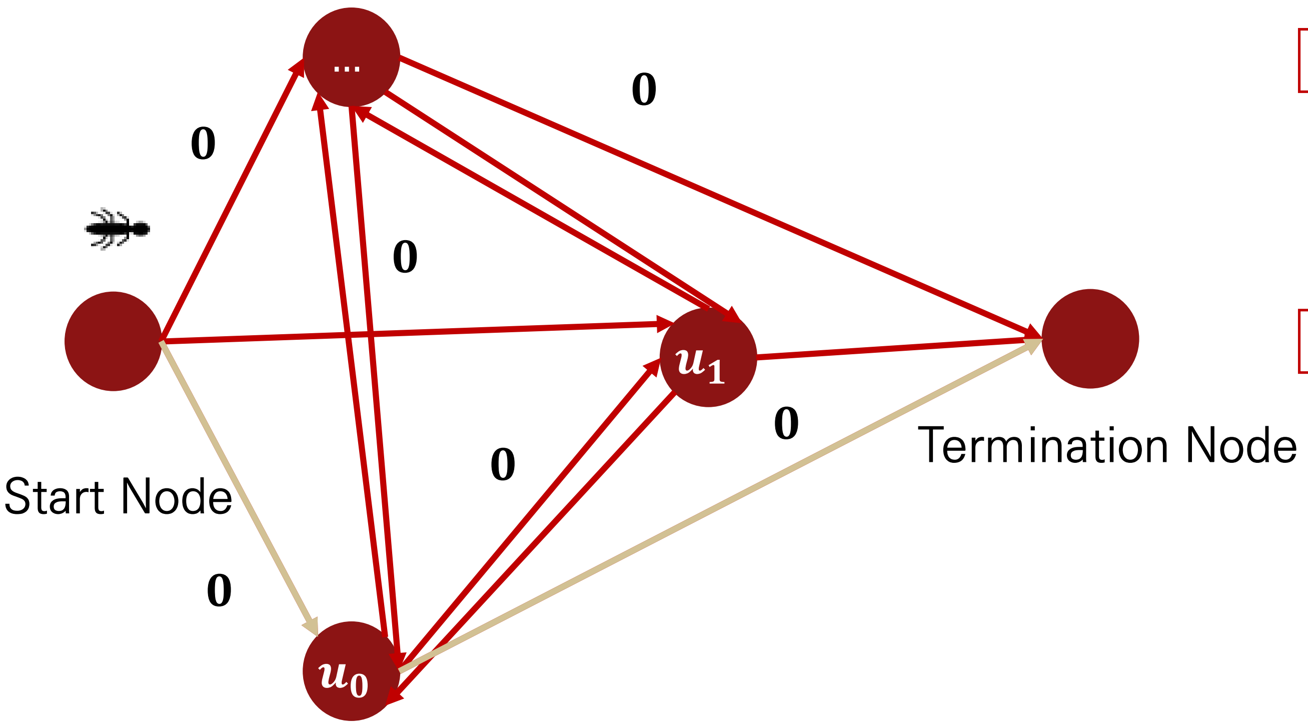
Remark 3.3

- Encode each feasible solution by a **binary string of fixed length**
- Design a construction graph with
  - (i) a **start node**,
  - (ii) a **termination node**,
  - (iii) a **completely interconnected subgraph** containing a node for each possible bit position (the visited nodes are then the 1-bits)
  - (iv) arcs leading from the start node to each other node,
  - (v) arcs leading from each other node to the termination node;
- exclude infeasible binary strings by **locking the corresponding walks** via the process described in component 5 above.

Toy Problem

1	0	0	⋯	1	0	1	1	0

u_0 u_1



- Algorithm 1. Ant colony optimization (ACO)

while termination conditions not met **do**
 ScheduleActivities
  AntBasedSolutionConstruction()
  PheromoneUpdate()
 end ScheduleActivities
end while

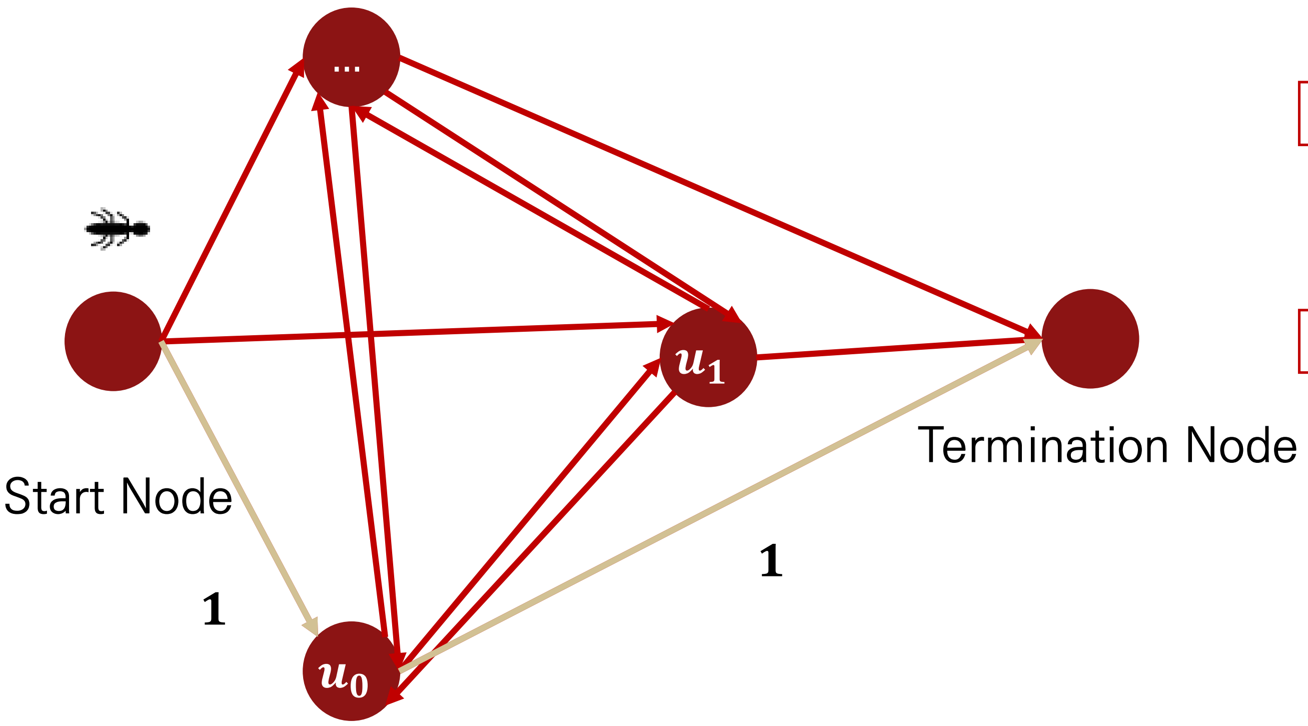
Walk w → (1, 0, 0, 0, ⋯,0)

Transition probabilities

$$p_{kl}(m,u) = \begin{cases} \dfrac{[\tau_{kl}(m)]^{\alpha} \cdot [\eta_{kl}(u)]^{\beta}}{\sum_{\{r \notin u,(k,r)\in A\}}[\tau_{kr}(m)]^{\alpha} \cdot [\eta_{kr}(u)]^{\beta}} & if \;\; l \notin u \\ 0 & if \;\; l \in u \end{cases}$$

Toy Problem

1	0	0	⋯	1	0	1	1	0
u_0	u_1							



- Algorithm 1. Ant colony optimization (ACO)

while termination conditions not met **do**
    ScheduleActivities
        AntBasedSolutionConstruction()
        PheromoneUpdate()
    end ScheduleActivities
end while

Walk w → (1, 0, 0, 0, ⋯,0)

Calculating Objective function value of *w*

Update Pheromone

Updating Rule

Let $C = \sum_{(k,l)\in\mathcal{A}} \sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}$.

$$\tau_{kl}(m+1) = \begin{cases} \tau_{kl}(m) & if\ C = 0 \\ (1-\rho)\tau_{kl}(m) + \rho\Delta\tau_{kl} & if\ C > 0 \end{cases}$$

$$where\ \Delta\tau_{kl} = \frac{1}{C}\sum_{s=1}^{S} \Delta\tau_{kl}^{(s)}$$

Suppose this solution has a cost value f_s.
for each arc (k,l),

$$\Delta\tau_{kl}^{(s)} = \begin{cases} \varphi(f_s) \\ 0 \end{cases}$$

where φ is a non-increase function

Cautions!!

- Consider the solution space
  - No continuous Opt.
  - No Dynamic Opt.

without greedy element!

$$p_{kl}(m,u) = \begin{cases} \dfrac{[\tau_{kl}(m)]^{\alpha}}{\sum_{\{r \notin u,(k,r) \in A\}} [\tau_{kr}(m)]^{\alpha}} & if\ \ l \notin u \\ 0 & if\ \ l \in u \end{cases}$$

- in the special case $\beta = 0$, we can guarantee the fact that this algorithm find the optimal solution.

- But, in case ρ=0, we don't Trivial!

$$\tau_{kl}(m+1) = \begin{cases} \tau_{kl}(m) & if\ C = 0 \\ (1-\rho)\tau_{kl}(m) + \rho\Delta\tau_{kl} & if\ C > 0 \end{cases} \Rightarrow \tau_{kl}(m+1) = \begin{cases} \tau_{kl}(m) & if\ C = 0 \\ \tau_{kl}(m) & if\ C > 0 \end{cases}$$

Convergence behavior to an optimal solution

- Condition

(a) $\alpha = 1$ in Equation (1).

(b) There is only one optimal walk in W, that is, the optimal solution is unique, and it is encoded by only one walk in W.

(c) Along the optimal walk w^* , $\eta_{kl}(u) > 0$ for all arcs (k, l) of w^* and the corresponding partial walks u of w^* .

(d) Let $f^* = f^*(m)$ be the lowest cost value observed in the cycles $1, \ldots, m-1$,

- If $m = 1$, let $f^* = \infty$
- Let the function ϕ chosen for the definition of the values $\Delta\tau_{kl}^{s}$ at the beginning of cycle $m+1$ have the following properties:
  - (i) $\phi(f_s) > 0 \; for \; f_s \leq f^*$
  - (ii) $\phi(f_s) = 0 \; for \; f_s > f^*$.

Convergence behavior to an optimal solution

Goal – Theorem 4.1

- Let conditions (a) – (d) be satisfied, and let P_m denote the probability that a fixed agent, say agent A_1, traverses the optimal walk in cycle m. Then the following two assertions are valid:

1. For every $\epsilon > 0$ and for fixed parameters ρ and β, it can be achieved by the choice of a sufficiently large number S of agents that
$P_m \geq 1 - \epsilon$ holds for all $m \geq m_0$ (with an integer m_0 depending on ϵ).

2. For every $\epsilon > 0$ and for fixed parameters S and β, it can be achieved by the choice of an evaporation factor ρ sufficiently close to zero that
$P_m \geq 1 - \epsilon$ holds for all $m \geq m0$ (with an integer m_0 depending on ϵ).

Convergence behavior to an optimal solution

1 *(Proposition 4.1)*

we show that the search procedure of Graph–based Ant System can be understood as a ***Markov process***,

2 *(Lemma 4.1)*

Next, a ***lower bound for the probability*** that at least one agent traverses the optimal walk in a fixed cycle m is derived

3 *(Lemma 4.2)*

the optimal walk is traversed **at least once** by some agent at some time

the pheromone values of arcs in the ***optimal walk*** $\rightarrow \frac{1}{length\ of\ the\ optimal\ walk}$
And the ***other arcs*** → *0.*

4 *(Lemma 4.3)*

under the mentioned condition 3,
the last– mentioned phenomenon has the consequence that the computed transition probability values responsible for the traversal of the arcs of the optimal walk get closer to unity

5 *(Lemma 4.4)*

under the mentioned condition,
probability for the event that a fixed agent traverses the optimal walk gets ***closer and closer to unity*** during the execution of the algorithm

6

by combining the last observation with an estimation of the probability that no agent will ever traverse the optimal walk:

Convergence behavior to an optimal solution

1 (Proposition 4.1)

we show that the search procedure of Graph-based Ant System can be understood as a *Markov process*,

2 (Lemma 4.1)

Next, a *lower bound for the probability* that at least one agent traverses the optimal walk in a fixed cycle m is derived

3 (Lemma 4.2)

the optimal walk is traversed **at least once** by some agent at some time

the pheromone values of arcs in the *optimal walk* → $\frac{1}{\textit{length of the optimal walk}}$

And the *other arcs* → *0.*

4 (Lemma 4.3)

About Simplity

1) Condition (a) is applied merely as a simplification of the pheromone update rule

2) Condition (c) enables the lower bound estimation of Lemma 4.1.

5

probability for the event that a fixed agent traverses the optimal walk gets *closer and closer to unity* during the execution of the algorithm

6

by combining the last observation with an estimation of the probability that no agent will ever traverse the optimal walk:

Convergence behavior to an optimal solution

(Proposition 4.1)

1 we show that the search procedure of Graph-based Ant System can be understood as a ***Markov process***,

(Lemma 4.1)

2 Next, a ***lower bound for the probability*** that at least one agent traverses the optimal walk in a fixed cycle m is derived

(Lemma 4.2)

3 the optimal walk is traversed **at least once** by some agent at some time

the pheromone values of arcs

in the ***optimal walk*** $\rightarrow \frac{1}{length\ of\ the\ optimal\ walk}$

And the ***other arcs*** → ***0.***

(Lemma 4.3)

4 under the mentioned condition 3, the last- mentioned phenomenon has the consequence that the computed transition probability values responsible for the traversal of the arcs of the optimal walk get closer to unity

(Lemma 4.4)

5

About Solution Optimality

1) By condition (b), the proof can be reduced to an investigation of what happens on the unique optimal walk.

2) Condition (d), finally, is used to avoid premature convergence to a suboptimal solution.

traverse the optimal walk.

Convergence behavior to an optimal solution

Proposition 4.1.

The state variables $(\tau(m), w(m), f^*(m))$, $(m = 1,2,\dots)$ form a Markov process.

- $\tau(m)$ is the vector of the pheromone values $\tau_{kl}(m)$ for all arcs (k, l) during cycle m
- $w(m)$ is the vector of the walks $w^s(m)$ $(s = 1,\dots,S)$ of the agents $A_1, \dots, A_S$ in cycle m
- $f^*(m)$ is the best found cost value corresponding to the walk of any agent in cycle $1,\dots,m - 1$. For cycle m = 1, we set $f^*(1) = \infty$.

proof

- $\tau(m)$ results deterministically from $\tau(m - 1)$, $w(m - 1)$ and $f^*(m - 1)$ according to the update rule for the pheromone values.
- The probability distribution of $w(m)$ only depends on $\tau(m)$ and is hence determined by $\big(\tau(m - 1),\ w(m - 1),\ f^*(m - 1)\big)$. (the values $\eta_{kl}(u)$ are deterministic!)
- $f^*(m - 1)$ results deterministically from $w(m - 1)$ and $f^*(m - 1)$.

Convergence behavior to an optimal solution

- In the sequel, the following abbreviations shall be used:
  - w^* denotes the (unique) optimal walk.
  - L denotes the length (number of arcs) of w^*.
  - Pr is written for the probability measure on the Markov process defined above.
  - $E_m^{(s)}$ denotes the event that $w_{(m)}^{(s)} = w^*$, that is, the event that agent A_s traverses in cycle m the optimal walk.
  - B_m is an abbreviation for $\neg E_m^{(1)} \wedge \cdots \wedge \neg E_m^{(S)}$, that is, for the event that $w^s(m) \neq w^*$ for all $s = 1, \ldots, S$ *(the event that no agent traverses the optimal walk in cycle m)*.
  - F_m is an abbreviation for $B_1 \wedge \cdots \wedge B_{m-1} \wedge \neg B_m$, that is, for the event that the optimal walk is traversed by some agent in cycle m, but by no agent in the cycles $1, \ldots, m-1$. Obviously, the events $F_1, F_2, \ldots$ are mutually exclusive.
  - A is an abbreviation for $F_1 \vee F_2 \vee \cdots$, that is, for the event that there is an m and an s such that $w^{(s)}(m) = w^*$ (the event that the optimal walk is traversed by some agent in some cycle).

Convergence behavior to an optimal solution

- In the sequel, the following abbreviations shall be used:
  - Because of condition (c) at the beginning of this section and the fact that there are only finitely many arcs (k, l) and only finitely many feasible partial walks u,

$$\gamma \ = \ \min\{[\eta_{kl}(u)]^{\beta} : (k,l) \in w^*,\ u \text{ partial walk of } w^*\} \ > \ 0 \tag{7}$$

and $\Gamma \ = \max\left[\eta_{kl(u)}\right]^{\beta} < \infty$.

Convergence behavior to an optimal solution

Lemma 4.1.

The probability $\Pr(\boxed{\neg \boldsymbol{B_m}})$ that at least one agent traverses the optimal walk in cycle m is larger or equal to $\boxed{\mathbf{1-\left(1-c^{m-1}p\right)^{S}}}$, where $c = (1-\rho)^L$ and $p = \gamma^L \prod_{(k,l)\in w^*} \tau kl(1)$ with γ defined by (7).

Next, a ***lower bound for the probability*** that **at least one agent traverses the optimal walk** in a fixed cycle m is derived

proof

Since $\Delta\tau_{kl} \geq 0$ *and* $\rho > 0$, $\tau_{kl}(m+1) \geq (1-$

By induction, $\tau_{kl}(m) \geq (1-\rho)^{m-1}\tau_{kl}(1)$.

And since $\sum_{(k,l)} \tau_{kl}(m) = 1$, $\sum_{r\notin u,(k,r)\in A} \tau kr(m) \cdot [\eta_{kr}(u)]^{\beta} \leq \sum_{r\notin u,(k,r)\in A} \tau_{kr}(m) \leq 1$

Therefore,

$$p_{kl}(m,u) = \frac{\tau_{kl}(m)\,[\eta_{kl}(u)]^{\beta}}{\sum_{r\notin u,(k,r)\in\mathcal{A}} \tau_{kr}(m)\,[\eta_{kr}(u)]^{\beta}} \geq \tau_{kl}(m)\,[\eta_{kl}(u)]^{\beta}$$

Convergence behavior to an optimal solution

Lemma 4.1.

The probability $\Pr(\neg B_m)$ that at least one agent traverses the optimal walk in cycle m is larger or equal to $1-(1-c^{m-1}p)^S$, where $c = (1-\rho)^L$ and $p = \gamma^L \prod_{(k,l)\in w^*} \tau kl(1)$ with γ defined by (7).

proof
Since $\Delta\tau_{kl} \geq 0$ *and* $\rho > 0$, $\tau_{kl}(m+1) \geq (1-\rho)\tau_{kl}(m)$.

By induction, $\tau_{kl}(m) \geq (1-\rho)^{m-1}\tau_{kl}(1)$.
And since $\sum_{(k,l)} \tau_{kl}(m) = 1$, $\sum_{r\notin u,(k,r)\in A} \tau kr(m) \cdot [\eta_{kr}(u)]^\beta \leq \sum_{r\notin u,(k,r)\in A} \tau_{kr}(m) \leq 1$

Therefore,

$$p_{kl}(m,u) = \frac{\tau_{kl}(m)\,[\eta_{kl}(u)]^\beta}{\sum_{r\notin u,(k,r)\in\mathcal{A}} \tau_{kr}(m)\,[\eta_{kr}(u)]^\beta} \geq \tau_{kl}(m)\,[\eta_{kl}(u)]^\beta$$

Convergence behavior to an optimal solution

Lemma 4.1.

The probability $\Pr(\neg B_m)$ that at least one agent traverses the optimal walk in cycle m is larger or equal to $1-(1-c^{m-1}p)^S$, where $c = (1-\rho)^L$ and $p = \gamma^L \prod_{(k,l)\in w^*} \tau kl(1)$ with γ defined by (7).

proof Let $w^* = (v_0, \ldots, v_L)$.

$$Pr(E_m^{(s)}) = \prod_{i=0}^{L-1} p_{v_i v_{i+1}}(m,(v_0,\ldots,v_i)) \geq \prod_{i=0}^{L-1} \tau_{v_i v_{i+1}}(m)\,[\eta_{v_i v_{i+1}}]^\beta \geq \gamma^L \prod_{i=0}^{L-1} \tau_{v_i v_{i+1}}(m)$$

$$\geq \gamma^L \prod_{i=0}^{L-1} (1-\rho)^{m-1}\, \tau_{v_i v_{i+1}}(1) = \gamma^L\,(1-\rho)^{L(m-1)} \prod_{(k,l)\in w^*} \tau_{kl}(1) = c^{m-1}p.$$

Since the walks of the S agents are independent, this implies

$$Pr(B_m) \leq (1-c^{m-1}p)^S,$$

whence the assertion follows. □

Convergence behavior to an optimal solution

Lemma 4.2.

the optimal walk is traversed **at least once** by some agent at some time

For each $\epsilon > 0$ and each $m \in \boldsymbol{N}$ there is an integer $d(\epsilon, m) \in \boldsymbol{N}$ such that
$Pr\{|\tau_{kl}(m') - \frac{1}{L}| < \epsilon \; for \; all \; (k,l) \in w^* | F_m\} \geq 1 - \epsilon$ for all $m' \geq m + d(\epsilon, m)$,
and $Pr\{\tau_{kl}(m') < L\epsilon \; for \; all \; (k,l) \notin w^* | F_m\} \geq 1 - \epsilon$ for all $m' \geq m + d(\epsilon, m)$.

the pheromone values of arcs in the *optimal walk* $\rightarrow \frac{\boldsymbol{1}}{\boldsymbol{length\ of\ the\ optimal\ walk}}$

And the *other arcs* $\rightarrow$ *0.*

Convergence behavior to an optimal solution

Lemma 4.3.

Let $u^*(k)$ denote the partial walk on w^* leading to node k ($k \in w^*$). Then for each $\epsilon > 0$ and each $m \in \boldsymbol{N}$ there is an integer $d'(\epsilon, m) \in \boldsymbol{N}$ such that
$\Pr\{\boxed{\boldsymbol{p_{kl}(m', u^*(k)) \geq 1 - \epsilon}}\ for\ all\ (k,l) \in w^* | \boxed{\boldsymbol{F_m}}\} \geq \boxed{\boldsymbol{1-\epsilon}}$ for all $m' \geq m + d'(\epsilon, m)$.

under the mentioned condition 3, the last‑ mentioned phenomenon has the consequence that the **computed transition probability values** responsible for the traversal of the arcs of the optimal walk get closer to **unity**

Proof

By Lemma 4.2, for each $m' \geq m + d(\tilde{\epsilon}, m)$, with a
$|\tau_{kl}(m') - \frac{1}{L}| < \epsilon\ for\ all\ (k,l) \in w^*$ and $\tau_{kl}(m') < L$
Let $(k,l) \in w^*$, and let $u = u^*(k)$.

$$p_{kl}(m', u) = \frac{\tau_{kl}(m')[\eta_{kl}(u)]^\beta}{\sum_{r \notin u, r \neq l, r \in A} \tau_{kr}(m')[\eta_{kr}(u)]^\beta + \tau_{kl}(m')[\eta_{kl}(u)]^\beta}$$

Set $\eta = [\eta_{kl}(u)]^\beta > \gamma$ for abbreviation. With ν denoting the maximal outdegree of a node in C

$$p_{kl}(m', u) \geq \frac{\left(\frac{1}{L} - \tilde{\epsilon}\right)\eta}{\nu L \tilde{\epsilon} + \left(\frac{1}{L} + \tilde{\epsilon}\right)\eta} = \frac{1 - L\tilde{\epsilon}}{1 + \tilde{\epsilon}\left(\frac{\nu L^2}{\eta} + L\right)}.$$

Convergence behavior to an optimal solution

Lemma 4.3.

Let $u^*(k)$ denote the partial walk on w^* leading to node k ($k \in w^*$). Then for each $\epsilon > 0$ and each $m \in \boldsymbol{N}$ there is an integer $d'(\epsilon, m) \in \boldsymbol{N}$ such that $\Pr\{p_{kl}(m', u^*(k)) \geq 1 - \epsilon \; for \; all \; (k,l) \in w^* | F_m\} \geq 1 - \epsilon$ for all $m' \geq m + d'(\epsilon, m)$.

Proof

By Lemma 4.2, for each $m' \geq m + d(\tilde{\epsilon}, m)$, with a probability (conditional on F_m) of at least $1 - \tilde{\epsilon}$, $|\tau_{kl}(m') - \frac{1}{L}| < \epsilon \; for \; all \; (k,l) \in w^*$ and $\tau_{kl}(m') < L\epsilon \; for \; all \; (k,l) \notin w^*$.

Let $(k,l) \in w^*$, and let $u = u^*(k)$.

$$p_{kl}(m', u) = \frac{\tau_{kl}(m')[\eta_{kl}(u)]^{\beta}}{\sum_{r \notin \mathrm{u}, r \neq l, r \in A} \tau_{kr}(m')[\eta_{kr}(u)]^{\beta} + \tau_{kl}(m')[\eta_{kl}(u)]^{\beta}}$$

Set $\eta = [\eta_{kl}(u)]^{\beta} > \gamma$ for abbreviation. With ν denoting the maximal outdegree of a node in C

$$p_{kl}(m', u) \geq \frac{\left(\frac{1}{L} - \tilde{\epsilon}\right)\eta}{\nu L\tilde{\epsilon} + \left(\frac{1}{L} + \tilde{\epsilon}\right)\eta} = \frac{1 - L\tilde{\epsilon}}{1 + \tilde{\epsilon}\left(\frac{\nu L^2}{\eta} + L\right)}.$$

Convergence behavior to an optimal solution

Lemma 4.3.

Let $u^*(k)$ denote the partial walk on w^* leading to node k ($k \in w^*$). Then for each $\epsilon > 0$ and each $m \in \boldsymbol{N}$ there is an integer $d'(\epsilon, m) \in \boldsymbol{N}$ such that $\Pr\{p_{kl}(m', u^*(k)) \geq 1 - \epsilon \, for \, all \, (k, l) \in w^* | F_m\} \geq 1 - \epsilon$ for all $m' \geq m + d'(\epsilon, m)$.

Proof

Since $(1 + x)^{-1} \geq 1 - x$ for $x \geq 0$,

$$p_{kl}(m', u) \geq (1 - L\tilde{\epsilon})(1 - \tilde{\epsilon}(\nu L^2/\eta + L)) \geq 1 - (2L + \nu L^2/\eta)\tilde{\epsilon} \geq 1 - (2L + \nu L^2/\gamma)\tilde{\epsilon}.$$

✓

$$\boxed{p_{kl}(m', u^*(k)) \to 1}$$

Convergence behavior to an optimal solution

Lemma 4.4.

For each $\epsilon > 0$, there is an integer $d'''(\epsilon, m) \in \boldsymbol{N}$, such that for fixed s, $\boxed{\Pr(E^s_{m'}\,|\boxed{F_m}) \geq 1 - \epsilon}$ for all $m' \geq m + d'''(\epsilon, m)$.

under the mentioned condition, probability for the event that a fixed agent traverses the optimal walk gets ***closer and closer to unity*** during the execution of the algorithm

Proof. $E^{(s)}_{m'}$ is the event $w^s(m') = w^*$. For a fixed giv the probability of this event is the value $Y_{m'} = \prod_{(k,}$ variable with a certain distribution. So, in order to ge previous state, we have still to take the expected value with respect to the distribution of $Y_{m'}$.

In particular, $Pr\{Y_{m'} \geq 1 - \tilde{\epsilon}|F_m\} \geq 1 - \tilde{\epsilon}, for\ m' \geq m + d''(\tilde{\epsilon}, m)$.
Hence for such an m', $Pr\left(E^{(s)}_{m'}\middle|F_m\right) \geq Pr\left\{E^{(s)}_{m'} \wedge (Y_{m'} \geq 1 - \tilde{\epsilon})\middle|F_m\right\}$
$= pr\{E^{(s)}_{m'}|(Y_{m'} \geq 1 - \tilde{\epsilon}) \wedge F_m\} \cdot pr\{(Y_{m'} \geq 1 - \tilde{\epsilon}) \wedge F_m\} \geq (1-\tilde{\epsilon}) \cdot (1-\tilde{\epsilon}) \geq 1-2\tilde{\epsilon}$.

Convergence behavior to an optimal solution

Lemma 4.4.

For each $\epsilon > 0$, there is an integer $d'''(\epsilon, m) \in \boldsymbol{N}$, such that for fixed s, $\Pr(E^s_{m'} | F_m) \geq 1 - \epsilon$ for all $m' \geq m + d'''(\epsilon, m)$.

Proof. $E^{(s)}_{m'}$ is the event $w^s(m') = w^*$. For a fixed given state in cycle $m' - 1$ of the Markov process, the probability of this event is the value $Y_{m'} = \prod_{(k,l)\in w^*} p_{kl}(m', u^*(k))$. $Y_{m'}$ itself is a random variable with a certain distribution. So, in order to get the probability of $E^{(s)}_{m'}$ without fixing the previous state, we have still to take the expected value with respect to the distribution of $Y_{m'}$.

In particular, $Pr\{Y_{m'} \geq 1 - \tilde{\epsilon} | F_m\} \geq 1 - \tilde{\epsilon}, for\ m' \geq m + d''(\tilde{\epsilon}, m)$.
Hence for such an m', $Pr\left(E^{(s)}_{m'} \middle| F_m\right) \geq Pr\left\{E^{(s)}_{m'} \wedge (Y_{m'} \geq 1 - \tilde{\epsilon}) \middle| F_m\right\}$
$= pr\{E^{(s)}_{m'} | (Y_{m'} \geq 1 - \tilde{\epsilon}) \wedge F_m\} \cdot pr\{(Y_{m'} \geq 1 - \tilde{\epsilon}) \wedge F_m\} \geq (1 - \tilde{\epsilon}) \cdot (1 - \tilde{\epsilon}) \geq 1 - 2\tilde{\epsilon}$.

Convergence behavior to an optimal solution

Proof. $E_{m'}^{(s)}$ is the event $w^s(m') = w^*$. For a fixed given state in cycle $m' - 1$ of the Markov process, the probability of this event is the value $Y_{m'} = \prod_{(k,l)\in w^*} p_{kl}(m', u^*(k))$. $Y_{m'}$ itself is a random variable with a certain distribution. So, in order to get the probability of $E_{m'}^{(s)}$ without fixing the previous state, we have still to take the expected value with respect to the distribution of $Y_{m'}$.

In particular, $Pr\{Y_{m'} \geq 1 - \tilde{\epsilon} | F_m\} \geq 1 - \tilde{\epsilon}, for\ m' \geq m + d''(\tilde{\epsilon}, m)$.
Hence for such an m', $Pr\left(E_{m'}^{(s)} \middle| F_m\right) \geq Pr\left\{E_{m'}^{(s)} \wedge (Y_{m'} \geq 1 - \tilde{\epsilon}) \middle| F_m\right\}$
$= pr\{E_{m'}^{(s)} | (Y_{m'} \geq 1 - \tilde{\epsilon}) \wedge F_m\} \cdot pr\{(Y_{m'} \geq 1 - \tilde{\epsilon}) \wedge F_m\} \geq (1 - \tilde{\epsilon}) \cdot (1 - \tilde{\epsilon}) \geq 1 - 2\tilde{\epsilon}$.

Convergence behavior to an optimal solution

- *Proof of Theorem 4.1*
  - $P_m = Pr\left(E_m^{(1)}\right) = \ \dots \ = Pr\left(E_m^{(s)}\right).$
  - $Pr(B_1 \ \wedge \cdots \wedge B_m) \ = \ Pr(B_1) \cdot Pr(B_2|B_1) \cdot \cdots \cdot Pr(B_m|B_1 \ \wedge \cdots B_{m-1}).$
  - *by the Corollary to Lemma 4.1,*

$$Pr(B_1 \wedge \ldots \wedge B_m) \leq (1-p)^S (1-cp)^S \ldots (1-c^{m-1}p)^S = \left[\prod_{i=1}^{m}(1-c^{i-1}p)\right]^S := w(p,c,S)$$

$$\boxed{Pr(A) = 1 - \lim_{m\to\infty} Pr(B_1 \wedge \ldots \wedge B_m) \geq 1 - \lim_{m\to\infty}\left[\prod_{i=1}^{m}(1-c^{i-1}p)\right]^S = 1 - w(p,c,S).}$$

  - On the other hand, $w(p,c,S)$ can be made arbitrarily small
    either by choosing S sufficiently large, or by choosing ρ sufficiently small:

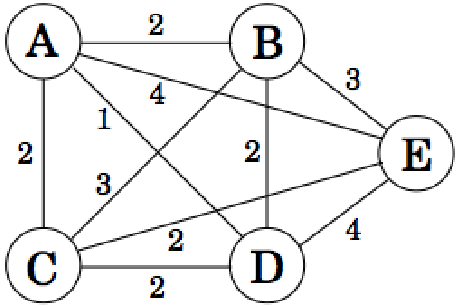
Convergence behavior to an optimal solution

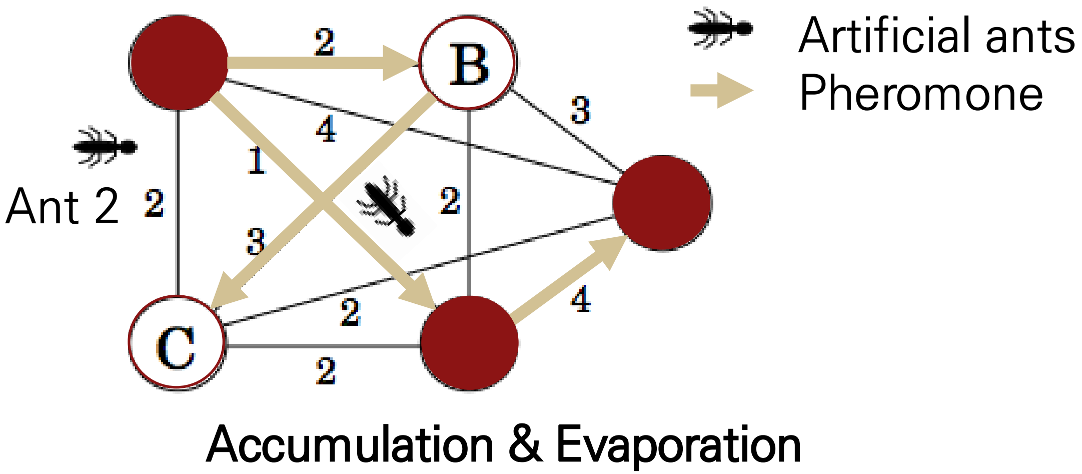
- *Proof of Theorem 4.1*
  - For all $m' \geq m_0$,
    - $$\begin{aligned} P_{m'} &= Pr\left(E_{m'}^{(1)}\right) = Pr\left(E_{m'}^{(1)}|F_1\right) \cdot Pr(F_1) + \cdots + Pr\left(E_{m'}^{(1)}|F_k\right) \cdot Pr(F_k) \\ &\quad + Pr\left(E_{m'}^{(1)}|\neg(F_1 \vee F_2 \vee \cdots \vee F_K)\right) \cdot Pr(\neg(F_1 \vee F_2 \vee \cdots \vee F_K)) \\ &\geq Pr\left(E_{m'}^{(1)}\right) = Pr\left(E_{m'}^{(1)}|F_1\right) \cdot Pr(F_1) + \cdots + Pr\left(E_{m'}^{(1)}|F_k\right) \cdot Pr(F_k) \\ &\geq \left(1 - \frac{\epsilon}{2}\right)(Pr(F_1) + \cdots + Pr(F_k)) \geq \left(1 - \frac{\epsilon}{2}\right)\left(1 - \frac{\epsilon}{2}\right) \geq 1 - \epsilon. \end{aligned}$$

Application of the algorithm for an NP–C/NP–hard example problem

- Traveling Salesman Problem
  - Finding a shortest closed tour
    - Visiting each node of a given graph with given edge length exactly once.

Figure 6: Traveling Salesman Problem and Ants build solutions, that is paths, from a source to a destination.





- Let $G = (X, E, W)$ be a complete weight graph,
  - $X = (x_1, x_2, \dots, x_n)\ (n \geq 3)$
  - $E = \{e_{ij} | x_i, x_j \in X\}$
  - $W = \{w_{ij} | w_{ij} \geq 0 \ and \ w_{ii} = 0, \ for \ all \ i, j \in \{1, 2, \dots, n\}\}$

Application of the algorithm for an NP-C/NP-hard example problem

- How to calculate Transient probability

the probability that city j is selected by ant k to be visited after city i

$$p_{ij}^k = \begin{cases} \dfrac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{s \in allowed_k} [\tau_{is}]^\alpha \cdot [\eta_{is}]^\beta} & j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

- τ_{ij} is the intensity of pheromone trail between cities i and j
- η_{ij} is the visibility of city j from city i, which is always set as $\frac{1}{d_{ij}}$ (d_{ij} is the distance between city i and j)
- $allowed\_k$ is the set of cities that have not been visited yet

Application of the algorithm for an NP–C/NP–hard example problem

- Update Pheromone

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k} \tag{3}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ travels on edge } (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

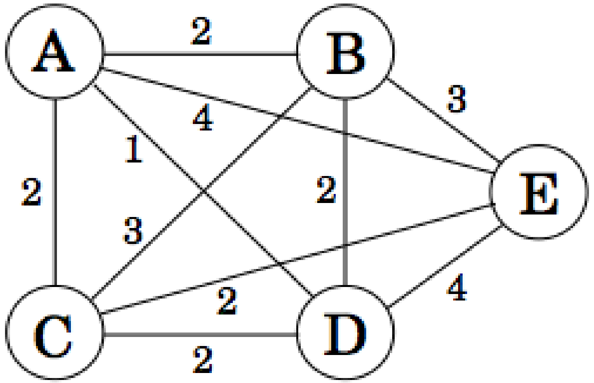
where Q is a constant and L_k the length of its tour

Application of the algorithm for an NP-C/NP-hard example problem

- Pseudocode of ACO for TSP

```
Initialize
For t=1 to iteration number do
   For k=1 to l do
      Repeat until ant k has completed a tour
        Select the city j to be visited next
        With probability p_ij given by Eq. (1)
      Calculate L_k
   Update the trail levels according to Eqs. (2-4).
End
```

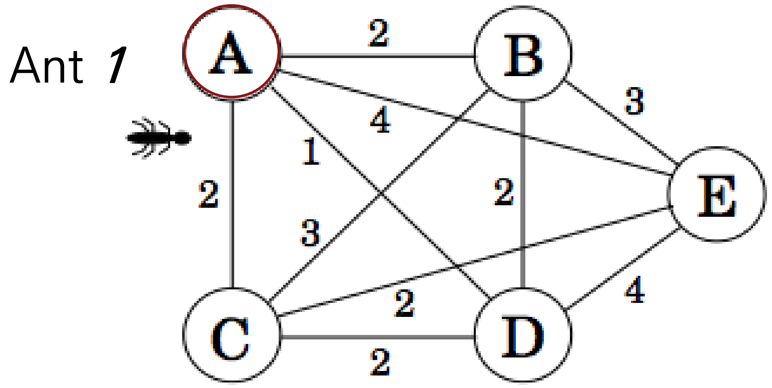
For every edge (i,j) set an initial $\tau_{ij} = c$ for trail density and $\Delta\tau_{ij} = 0$.



Application of the algorithm for an NP-C/NP-hard example problem

- Pseudocode of ACO for TSP

```
Initialize
For t=1 to iteration number do
  For k=1 to l do
    Repeat until ant k has completed a tour
      Select the city j to be visited next
      With probability p_ij given by Eq. (1)
    Calculate L_k
  Update the trail levels according to Eqs. (2-4).
End
```

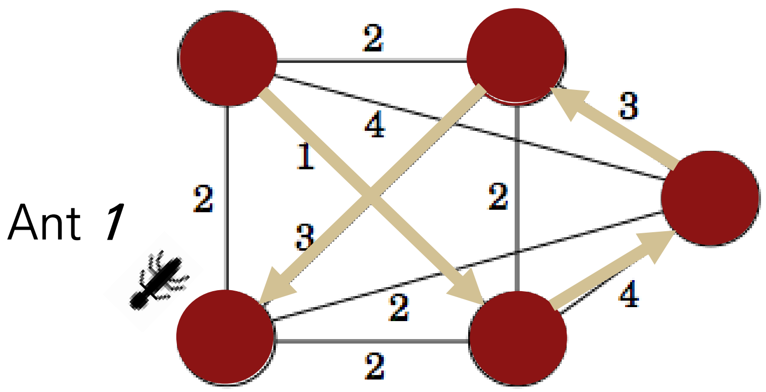


Application of the algorithm for an NP-C/NP-hard example problem

- Pseudocode of ACO for TSP

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{s \in allowed_k} [\tau_{is}]^\alpha \cdot [\eta_{is}]^\beta} & j \in allowed_k \\ 0 & \text{otherwise} \end{cases}$$

Initialize
For t=1 to iteration number do
  For k=1 to l do
    Repeat until ant k has completed a tour
      Select the city j to be visited next
      With probability p_{ij} given by Eq. (1)
    Calculate L_k
  Update the trail levels according to Eqs. (2-4).
End

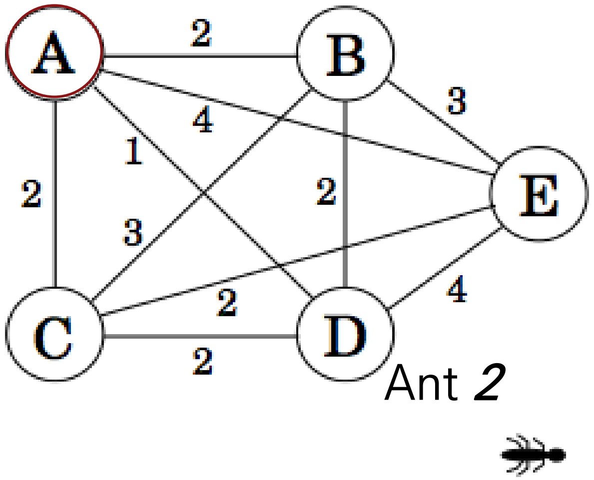


L_k : the length of its tour

Application of the algorithm for an NP-C/NP-hard example problem

- Pseudocode of ACO for TSP

```
Initialize
For t=1 to iteration number do
  For k=1 to l do
    Repeat until ant k has completed a tour
      Select the city j to be visited next
      With probability p_ij given by Eq. (1)
    Calculate L_k
  Update the trail levels according to Eqs. (2-4).
End
```



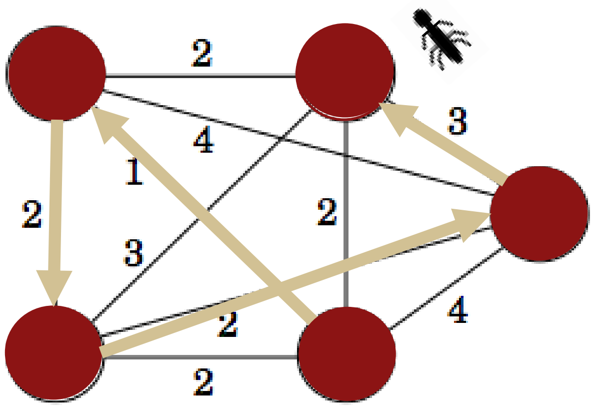
Application of the algorithm for an NP-C/NP-hard example problem

- Pseudocode of ACO for TSP

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{s\in allowed_k}[\tau_{is}]^{\alpha}\cdot[\eta_{is}]^{\beta}} & j \in allowed_k \\ 0 & \text{otherwise} \end{cases}$$

Initialize
For t=1 to iteration number do
  For k=1 to l do
    Repeat until ant k has completed a tour
      Select the city j to be visited next
      With probability p_{ij} given by Eq. (1)
    Calculate L_k
  Update the trail levels according to Eqs. (2-4).
End

Ant *2*



L_k : the length of its tour

Application of the algorithm for an NP–C/NP–hard example problem

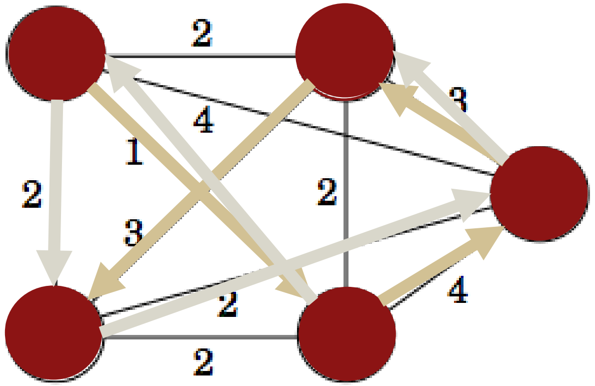
- Pseudocode of ACO for TSP

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k} \tag{3}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ travels on edge } (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Initialize
For t=1 to iteration number do
  For k=1 to l do
    Repeat until ant k has completed a tour
      Select the city j to be visited next
      With probability p_{ij} given by Eq. (1)
    Calculate L_k
  Update the trail levels according to Eqs. (2-4).
End



References

[1] Papadimitriou, Christos H., and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.

[2] Aarts, E. H. L., and J. Korst. "Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural networks." (1989).

[4] Cruz-Bernal, Alejandra. *Meta-Heuristic Optimization Techniques and Its Applications in Robotics*. INTECH Open Access Publisher, 2013.

[5] Blum, Christian. "Ant colony optimization: Introduction and recent trends." *Physics of Life reviews* 2.4 (2005): 353-373.

[6] Combinatorial optimization, Wikipedia, 2017.03.22

[7] Yang, Jinhui, et al. "An ant colony optimization method for generalized TSP problem." *Progress in Natural Science* 18.11 (2008): 1417-1422.

References

[8] Kaveh, A., and S. Talatahari. "Hybrid algorithm of harmony search, particle swarm and ant colony for structural design optimization." Harmony search algorithms for structural design optimization. Springer Berlin Heidelberg, 2009. 159-198.