Faculty of Engineering & Materials Science (EMS)
Mechatronics Engineering Department

Optimization Techniques of Multi-Cooperative Systems (MCTR 1021)

Tutorial (09)

Ant Colony Optimization (ACO)

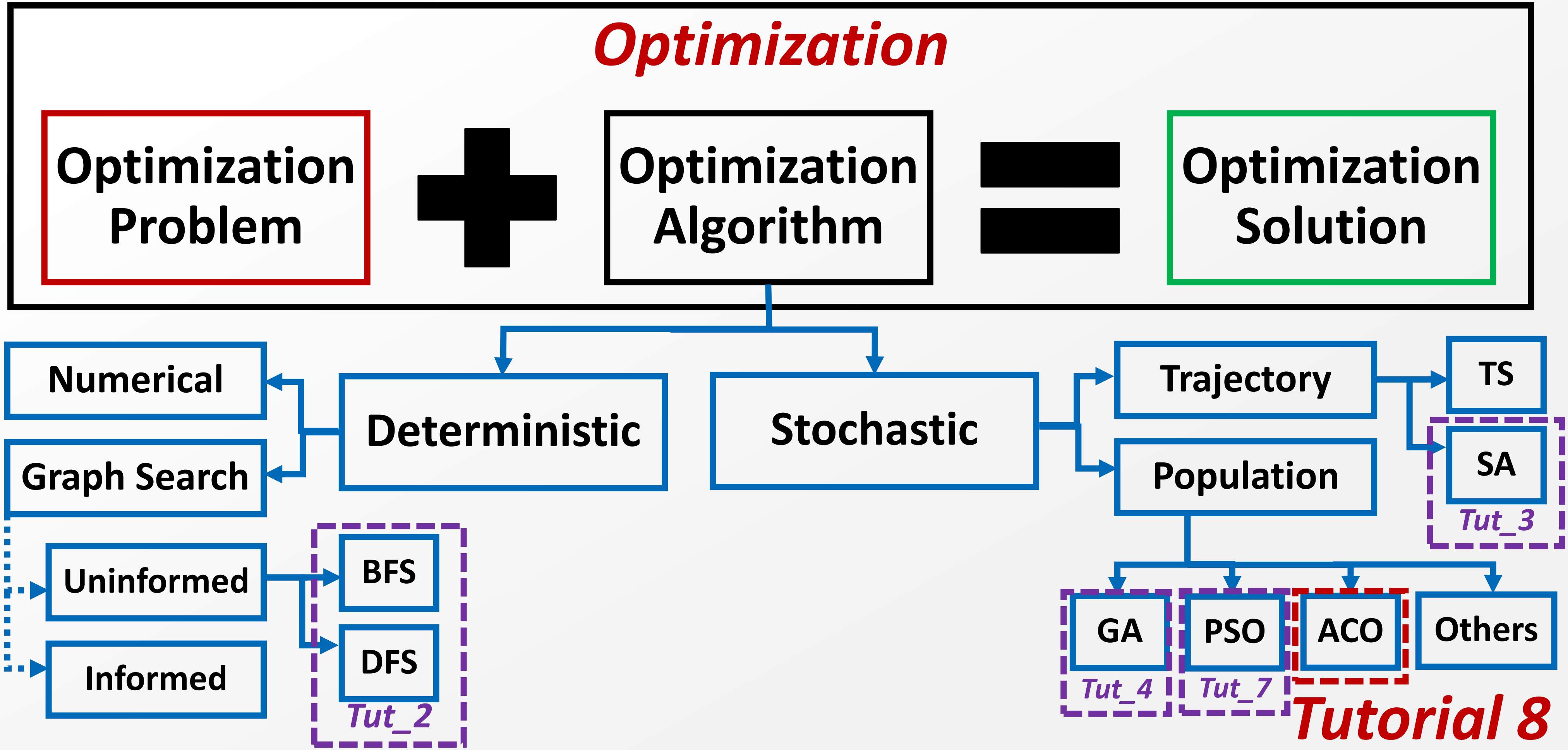
Presented by:
Mohamed A. Ibrahim , Jessica Magdy, Dalia Mamdouh, Mohammed Ashraf, Abdulrahman Yasser, Dr.Omar M. Shehata

Outline

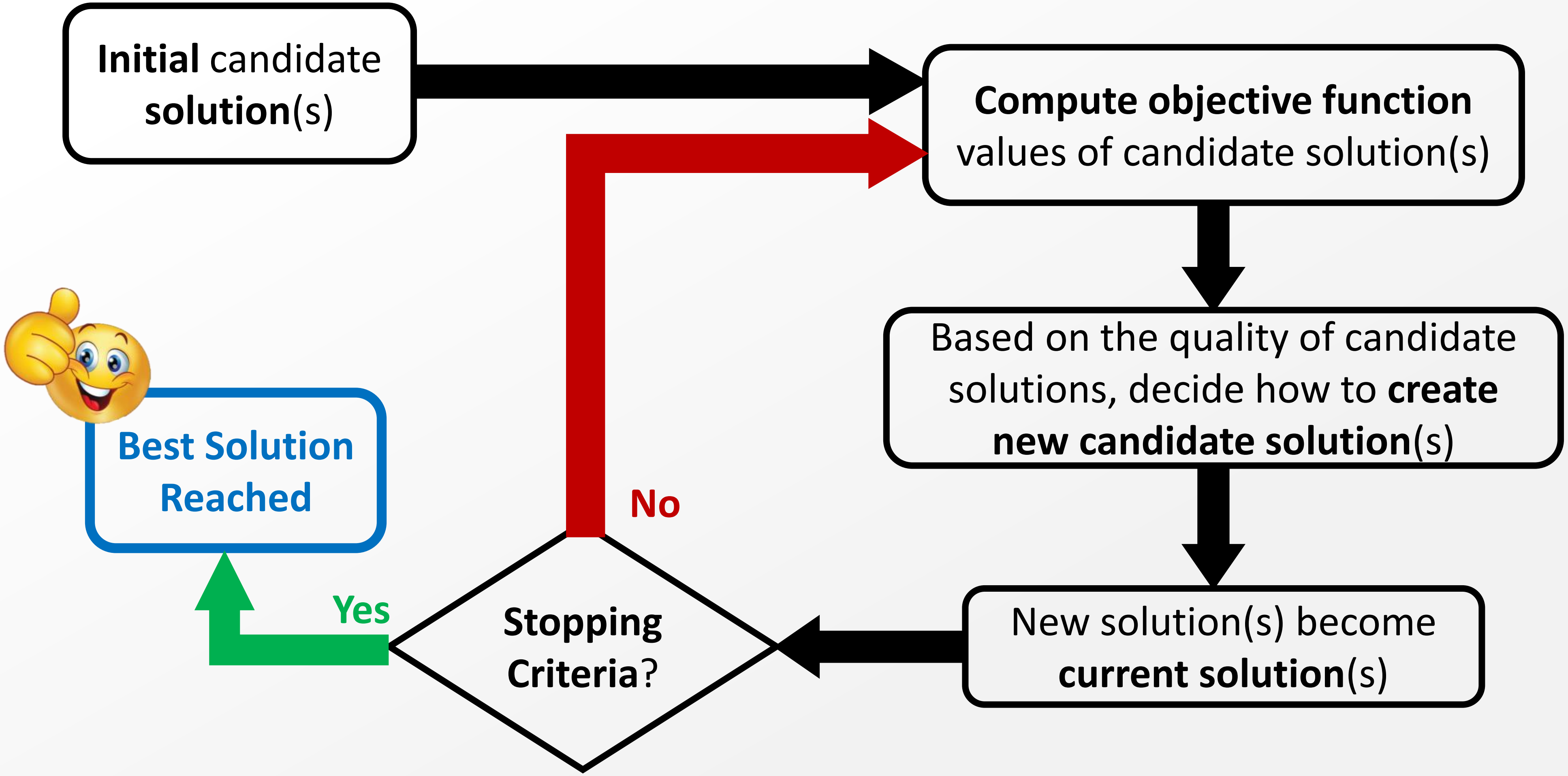
- Tutorial 7 Recap
- Ant Colony Optimization
- Application to Knapsack Problem
- Techniques Comparison

- **Tutorial 7 Recap**
- Ant Colony Optimization
- Knapsack Problem
- Techniques Comparison

Tutorial 7 Recap: Course Overview



Tutorial 7 Recap: Metaheuristics Flow Chart



- **Tutorial 7 Recap**
- Ant Colony Optimization
- Knapsack Problem
- Techniques Comparison

Tutorial 7 Recap: Opt. Techniques Comparison

	Genetic Algorithm (GA)	Particle Swarm Optimization (PSO)
Technique Category	Population-based	Population-based
Inspiration	Genetic Evolution of Living Creatures	Bird flocks (or schools of fish) moving and searching for food
Solution(s) Representation	• Each iteration is composed of **(1) Population** • Each population has **(N) Chromosomes** • Each Chromosome has **(M) Genes**	• Each iteration is composed of **(1) Population** • Each population has **(N) Particles** • Each particle has **(M) dimensions**
Generation of New Solution Criteria	GA Operators • Elitism • Cross-Over • Mutation	Motion Update • Inertia • Cognitive Component • Social Component

- **Tutorial 7 Recap**
- Ant Colony Optimization
- Knapsack Problem
- Techniques Comparison

Tutorial 7 Recap: Opt. Techniques Comparison

	Genetic Algorithm (GA)	Particle Swarm Optimization (PSO)
Convergence Criteria	Number of Generations	Number of Iterations
Acceptance Criteria	All Solutions in the populations are **accepted** as long as: • Solutions generated are **feasible.** • **No duplicate** solutions in the population specially in the early iterations to avoid exploitation.	All Solutions in the populations are **accepted** as long as: • Solutions generated are **feasible.** • **No duplicate** solutions in the population specially in the early iterations to avoid exploitation.
Stopping Criteria	• Final number of generations • Convergence to one solution • Elite is the same across many generations	• Final number of iterations • Convergence to one solution • Neighborhood best is the same across many iterations.

Ant Colony Optimization (ACO)

- Ant Colony Optimization is a biologically inspired optimization technique.
- It simulates how ants discover food sources and communicate their discoveries with other ants.
- ACO predates PSO and differential evolution and is used generally for combinatorial optimization problems.

How does the real ant think??

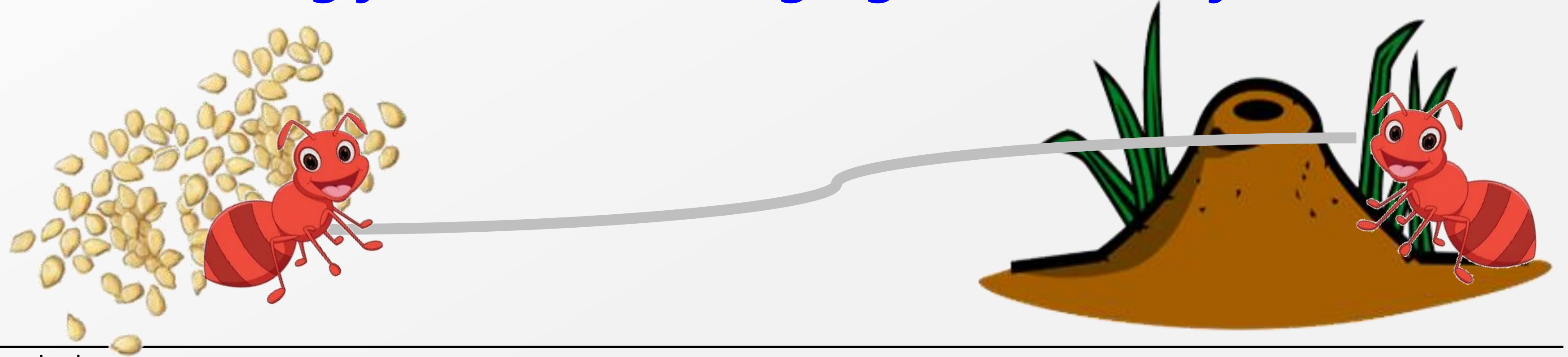
- Here's the thinking underlying the approach.

Suppose you're an ant

Food Source ← **Colony's Nest**

- The problem is that you don't know **where the food source is**. So, you have to search for it.
- Once you find it, you have to **get back home**, so you have to search for that too.

Searching for Food = Foraging Behavior of Ants



How does the real ant think??

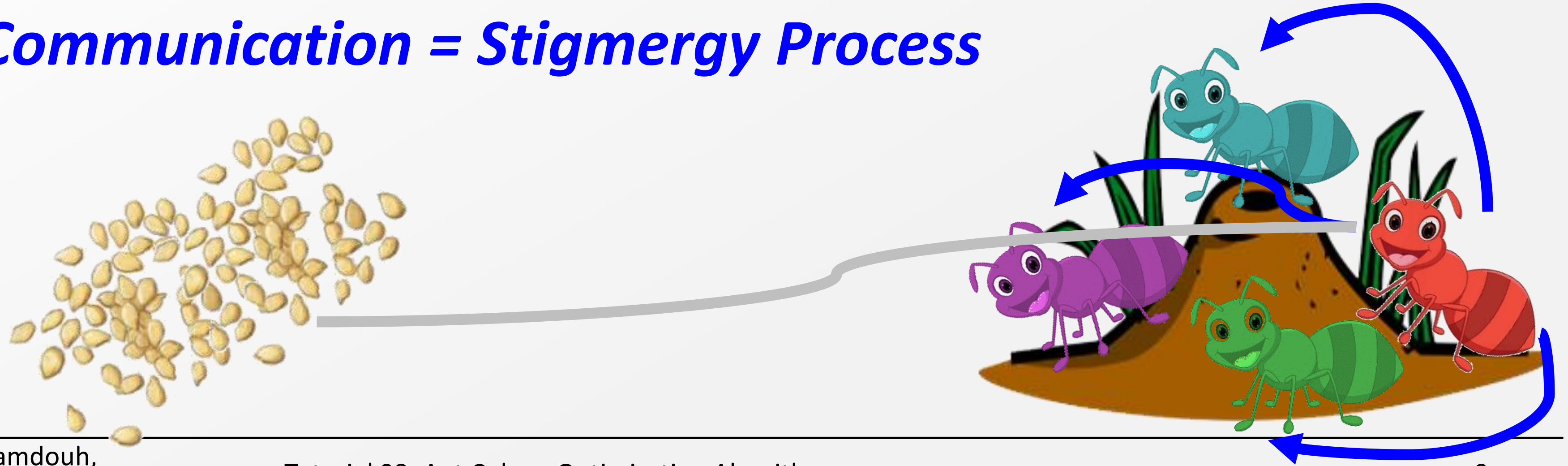
- Here's the thinking underlying the approach.

Suppose you're an ant in a colony

Food Source ← **Colony's Nest**

- It would be great if you could **share what you learn** as you search with other ants, so that they could search **more efficiently**.

Communication = Stigmergy Process



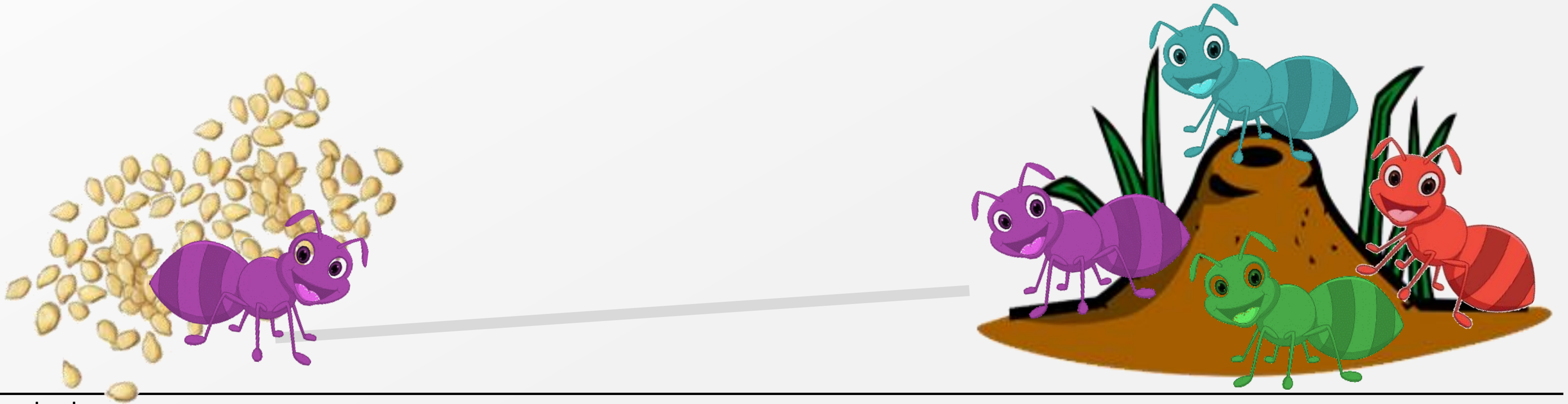
How to do that??

- Nature provides a way to do that. It's called a **pheromone trail**.

Ants leave pheromone trails as they move.

- The **strength** of the trail is dependent on **the number of ants that traverse the trail** and how recently the ants have traversed the trail.

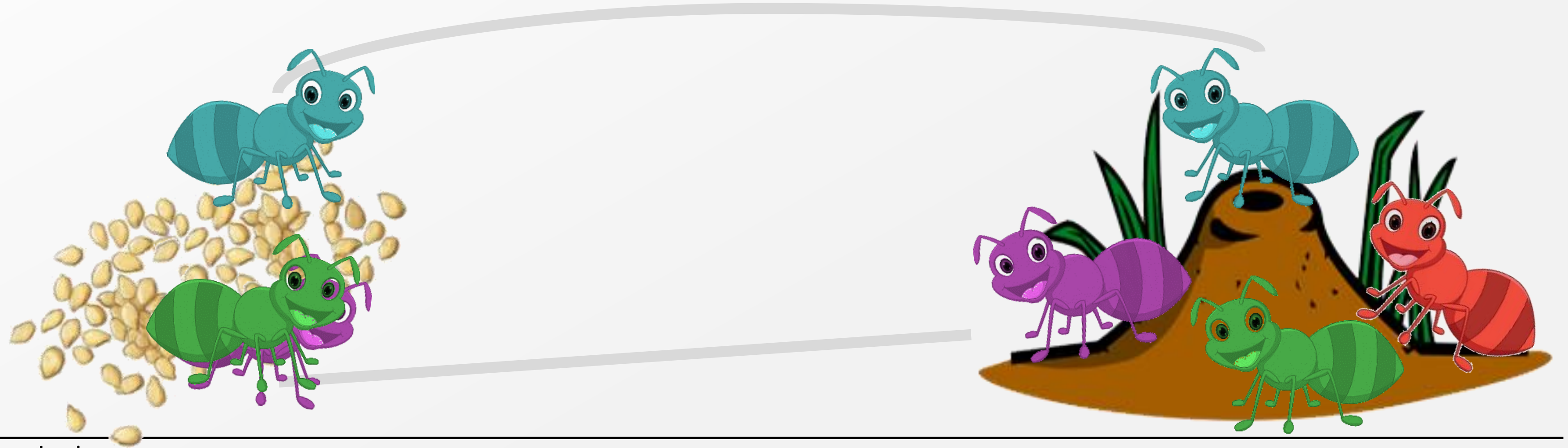
The more an ant traverses the trail, the strongest it gets



How to do that??

- **More ants** on a trail means **more pheromone**, which entices other ants to follow that trail rather than search some where else.
- The pheromone **evaporates** overtime, so once food sources are exhausted, the ants don't continue to travel to an "empty refrigerator" forever.

What is the shortest path to take???



- Tutorial 7 Recap
- **Ant Colony Optimization**
- Knapsack Problem
- Techniques Comparison

ACO Algorithm: Analogy

Optimization Algorithm	Ant Colony Optimization (ACO)
Solution	Ant Path from nest to food
(N) Solutions/Iteration	Colony Size
Max. Number of Iterations	Number of Generated Colonies
Objective Function	Path Length for Each Ant
Old Solution	Old Ant Path
New Solution	New Ant Path
Best Solution	Path with Strongest Pheromone

- Tutorial 7 Recap
- **Ant Colony Optimization**
- Knapsack Problem
- Techniques Comparison

ACO Algorithm: Solution Representation

- Ant Representation: ($\boldsymbol{n\ Variables}$)

Var_n	…	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0

Permutation Problem

Combination Problem

Binary Problem

Arithmetic Problem

- Colony Representation: ($\boldsymbol{m\ Ants}$)

$\boldsymbol{x}_{\boxed{\boldsymbol{i}}\,\boxed{\boldsymbol{j}}}$

Colony ($\boldsymbol{i}$)

Ant ($\boldsymbol{j}$)

$\boldsymbol{x_{i1}}$	Var_n	…	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0
$\boldsymbol{x_{i2}}$	Var_n	…	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0
$\boldsymbol{x_{i3}}$	Var_n	…	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\boldsymbol{x_{im}}$	Var_n	…	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0

ACO Algorithm: Fitness Function

- ***For a multi-objective function***

$$F(x) = \min\left(f_1 + f_2 + \frac{1}{f3}\right)$$

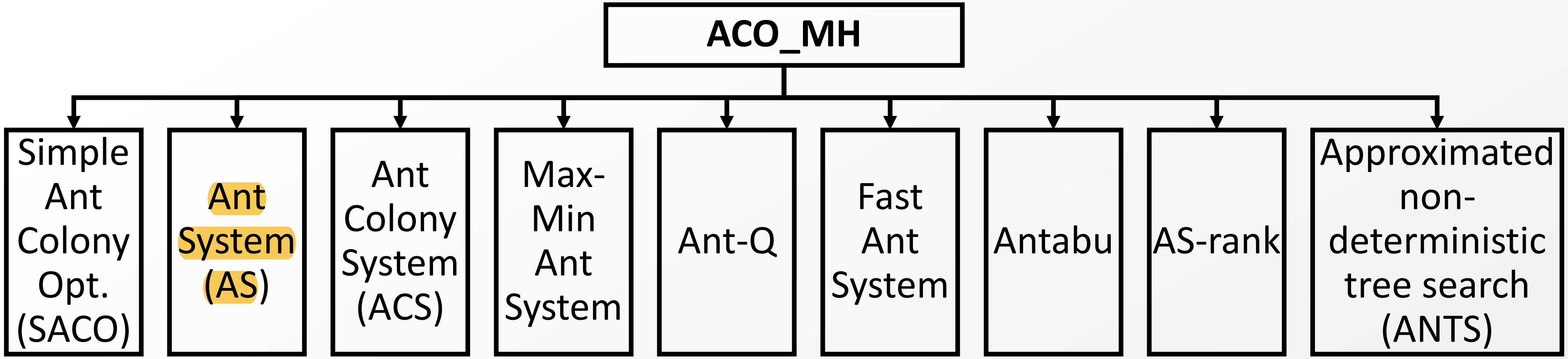
Minimize f_1 and f_2 & maximize f_3

- Evaluate the fitness value for each ant

Generated Ants Paths → ***Fitness Values***

									Fitness
$\boldsymbol{x_{i1}}$	Var_n	...	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0	$\boldsymbol{F_{i1}}$
$\boldsymbol{x_{i2}}$	Var_n	...	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0	$\boldsymbol{F_{i2}}$
$\boldsymbol{x_{i3}}$	Var_n	...	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0	$\boldsymbol{F_{i3}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{x_{im}}$	Var_n	...	Var_5	Var_4	Var_3	Var_2	Var_1	Var_0	$\boldsymbol{F_{im}}$

ACO Algorithm: Types

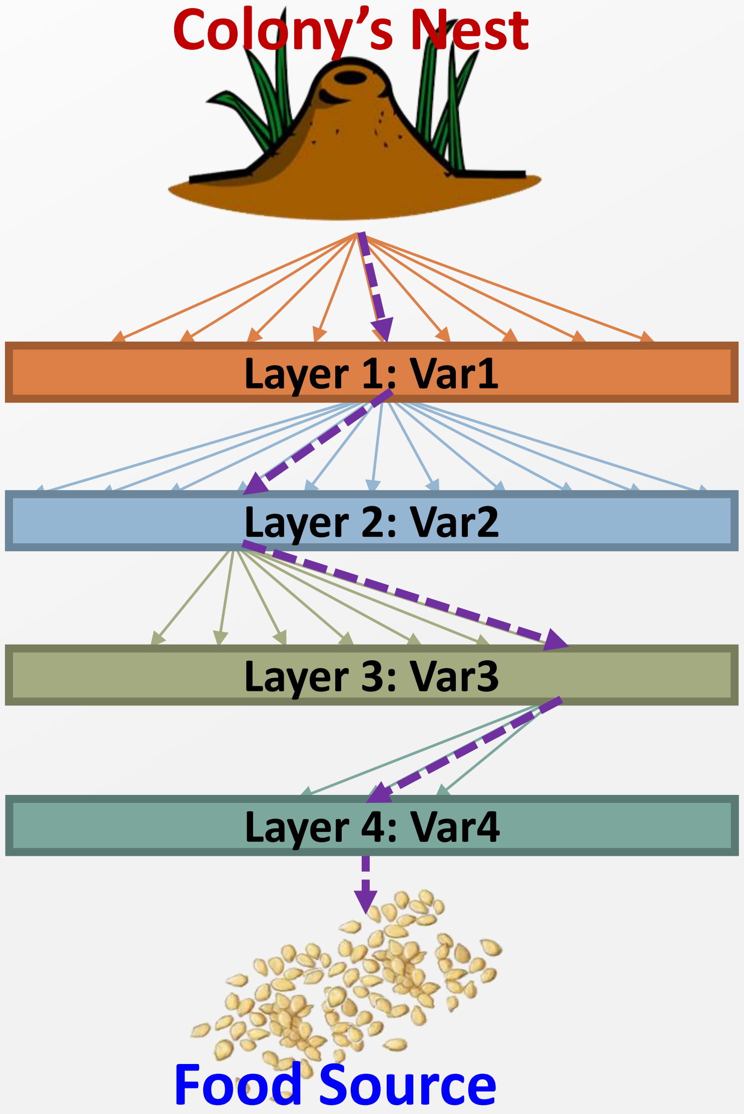


Read Only Material : (Engelbrecht, Andries P. Computational intelligence: an introduction. John Wiley & Sons, 2007.)

- Tutorial 7 Recap
- **Ant Colony Optimization**
- Knapsack Problem
- Techniques Comparison

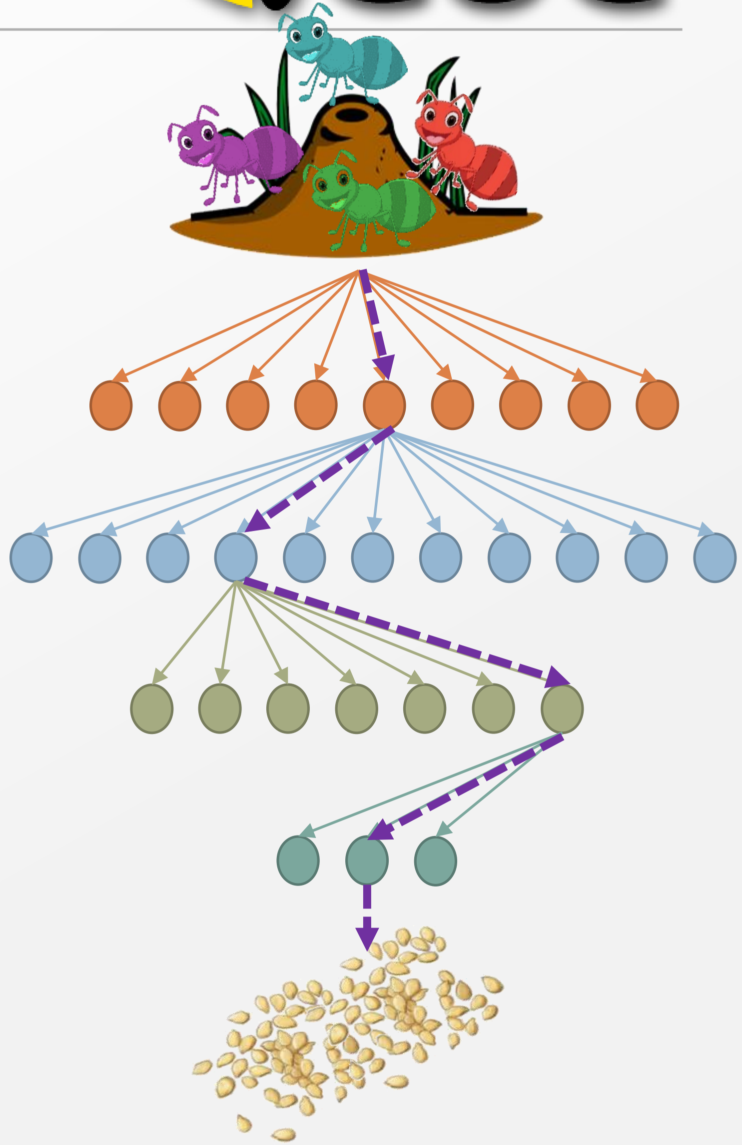
ACO Algorithm: Process

- The ACO Process can be represented in the form of **Multi-Layer Graph** where:
  - The **start** parent node is considered the **colony's nest** and the **end** node is the **food**.
  - Each **layer** represents a specific **decision variable** in the solution.
  - Each layer consists of multiple nodes, where each **node** is a **discrete value** permitted to the corresponding decision variable.
  - **By moving from one layer to another** going through the whole variables selection, there will be a **path from the start to end**.



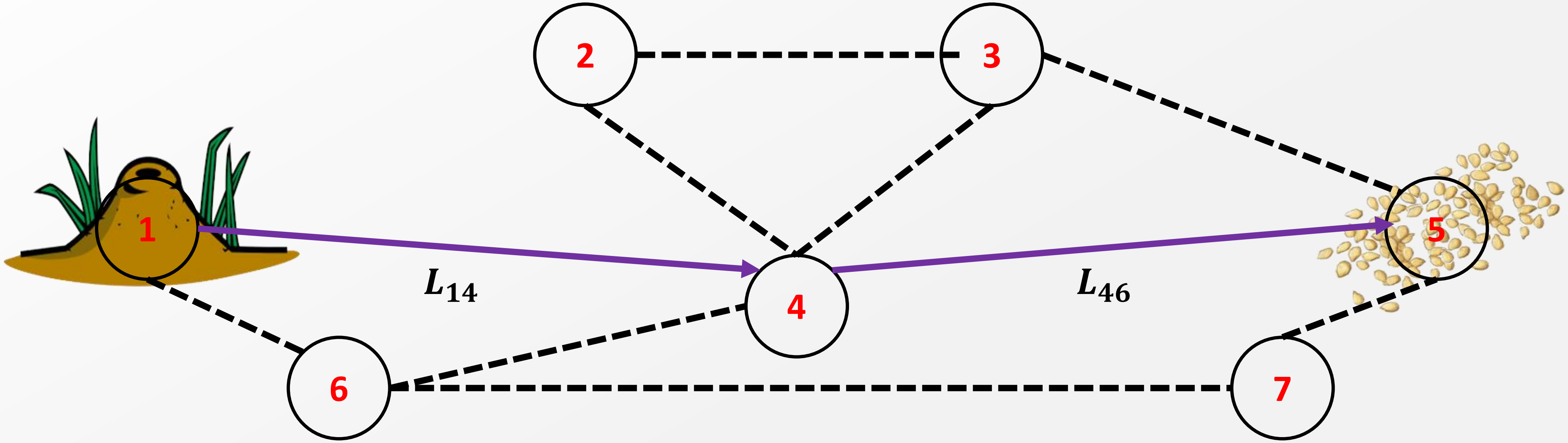
ACO Algorithm: Process

1. The ants start from the colony's nest. They travel throughout all layers.
2. Each ant only selects one node in each layer with the use of ***(Transition Rule)*** to form a candidate solution.
3. Once a path (solution) is completed, the ant starts to spread some pheromone based on ***(Local Updating Rule = Deposition + Evaporation)***
4. When all ants complete their paths, the global best path pheromone is updated using ***(Global Updating Rule for ACS ONLY)***



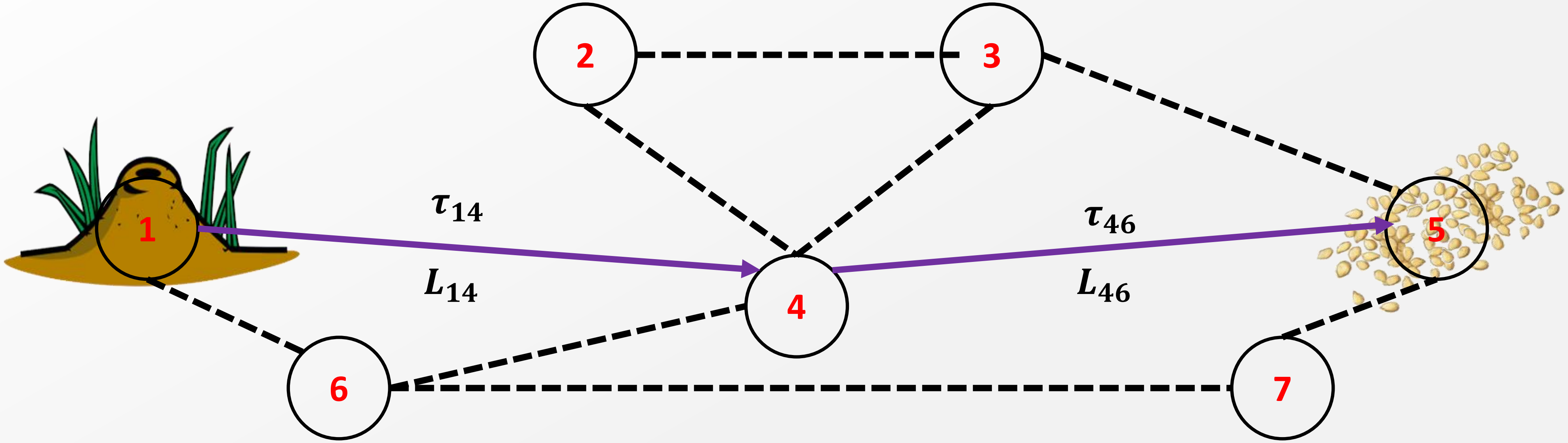
SACO Algorithm: Pheromone Concentration

- Consider the general problem of finding the shortest path between two nodes on a graph.
- The length L_i of the path constructed by ant_i is calculated as the number of hops in the path from the origin to the destination node.
- For the below **indicated route**, The length is 2.



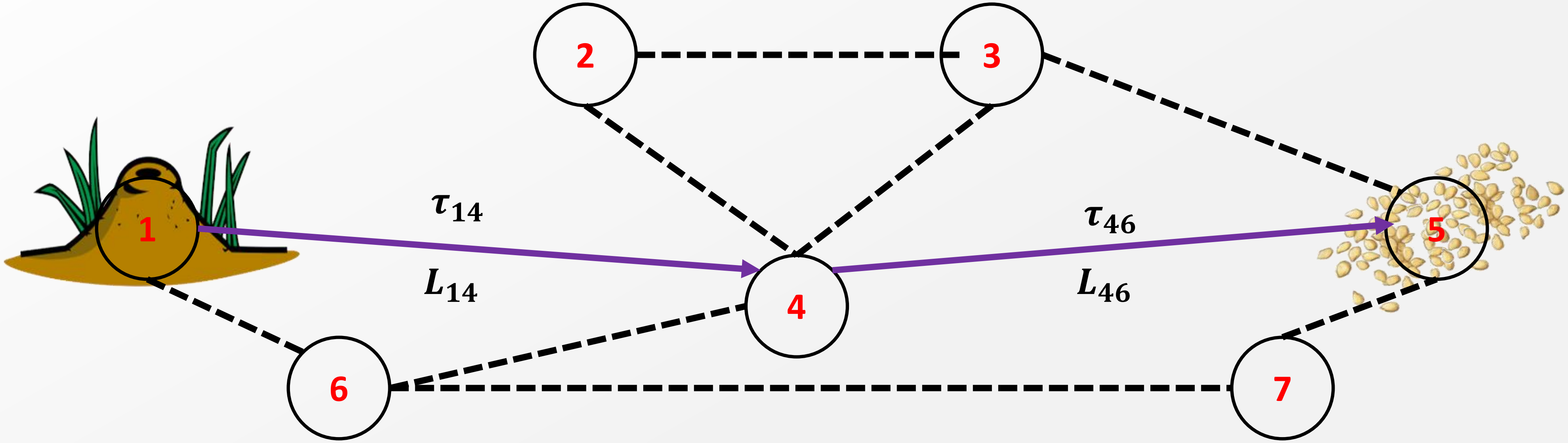
SACO Algorithm: Pheromone Concentration

- Assuming that we talk about two nodes i and j, an edge (i, j) is defined with associated **pheromone concentration**, $\boldsymbol{\tau_{ij}}$.
- Ex. τ_{64} is the pheromone concentration between nodes 4 and 6.



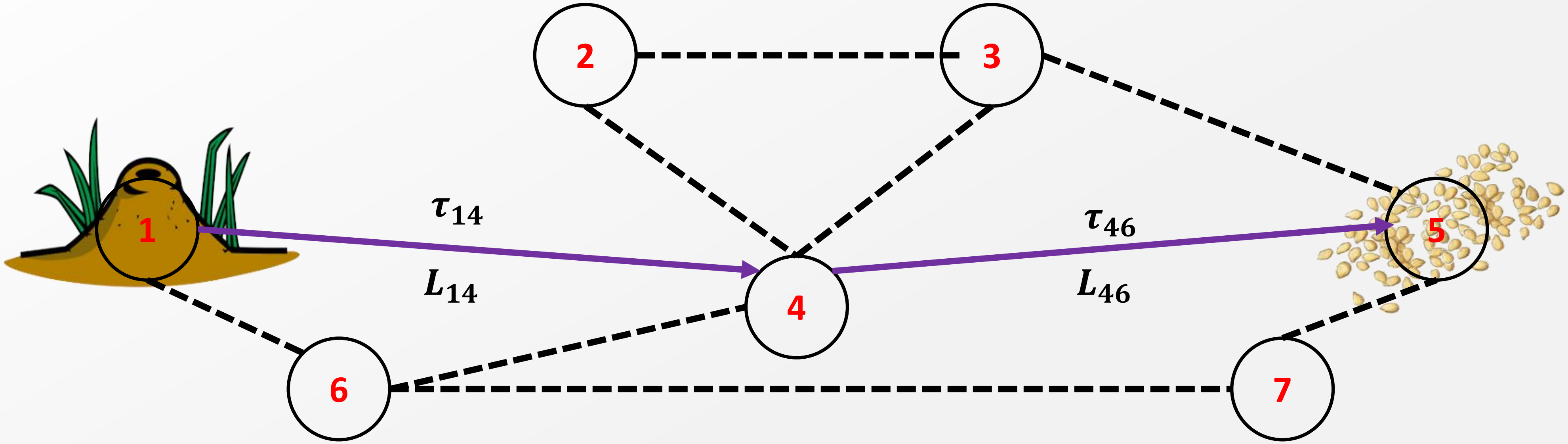
SACO Algorithm: Pheromone Concentration

- For the SACO, each edge is assigned a small random value to indicate the **initial pheromone** $\boldsymbol{\tau_{ij}(0)}$.
- A number of ants ($k = [1, \ldots, n]$), are placed on the source node.
- For each iteration of SACO, each ant incrementally constructs a path (solution) to the destination node.



SACO Algorithm: Transition Rule

- At each node, each ant executes a **decision policy** to determine the next link of the path.
- If $\boldsymbol{ant_k}$ is currently located at $\boldsymbol{node_i}$, it selects the next $\boldsymbol{node_j} \in N_i^k$, based on **the transition probability**.



- Tutorial 7 Recap
- **Ant Colony Optimization**
- Knapsack Problem
- Techniques Comparison

SACO Algorithm: Transition Rule

- **The transition probability:**

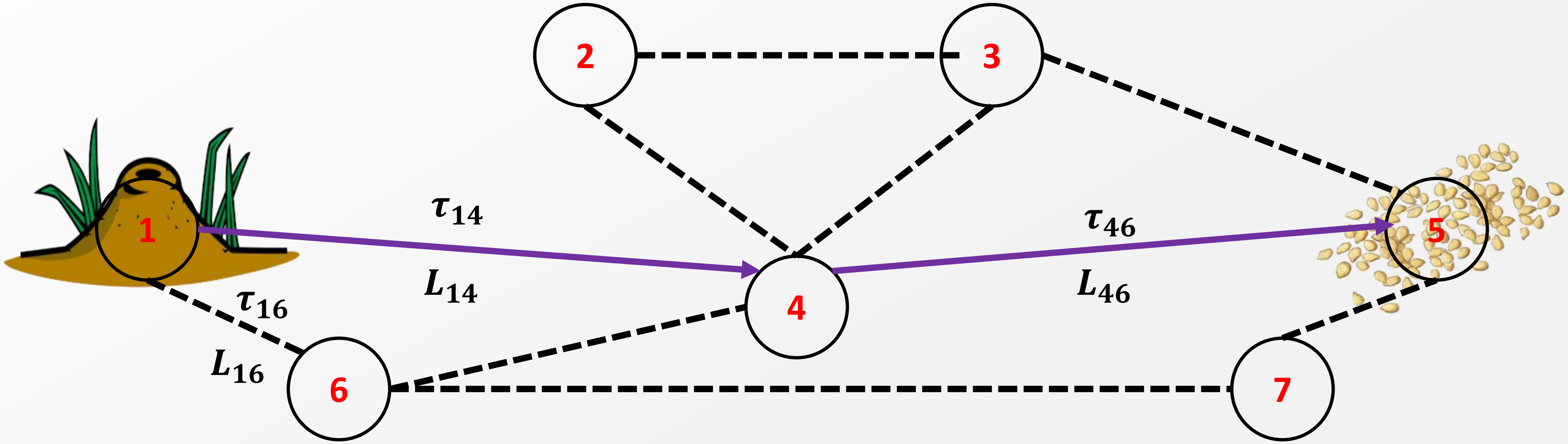
$$\boxed{p_{ij}^{k}} = \begin{cases} \dfrac{\boxed{\tau_{ij}^{\alpha}}}{\sum_{j \in N_i^k} \tau_{ij}^{\alpha}} & ,\text{if } j \in \boxed{N_i^k} \\ 0 & ,\text{if } j \notin N_i^k \end{cases}$$

- i is the current node, j is the next node, k is the ant number.
- For ant_k, N_i^k is the feasible set of nodes connected to node i.
- τ_{ij} is the pheromone concentration between nodes i and j.
- $\alpha \geq 0$ is parameter to control the influence of τ_{ij}.

SACO Algorithm: Transition Rule

- Ex. if ant_1 is currently moved at start

$$P_{14} = \frac{\tau_{14}^{\alpha}}{\tau_{14}^{\alpha} + \tau_{16}^{\alpha}}$$



- Tutorial 7 Recap
- **Ant Colony Optimization**
- Knapsack Problem
- Techniques Comparison

AS Algorithm: Transition Rule

- **The transition probability:**

$$\boxed{p_{ij}^{k}} = \begin{cases} \dfrac{\boxed{\tau_{ij}^{\alpha}} \times \boxed{\eta_{ij}^{\beta}}}{\sum_{j \in N_i^k} \tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}} & ,\text{if } j \in \boxed{N_i^k} \\ 0 & ,\text{if } j \notin N_i^k \end{cases}$$

- i is the current node, j is the next node, k is the ant number.
- For ant_k, N_i^k is the feasible set of nodes connected to node i.
- η_{ij} is the desirability of edge (i, j).
- $\beta \leq 1$ is parameter to control the influence of η_{ij}.

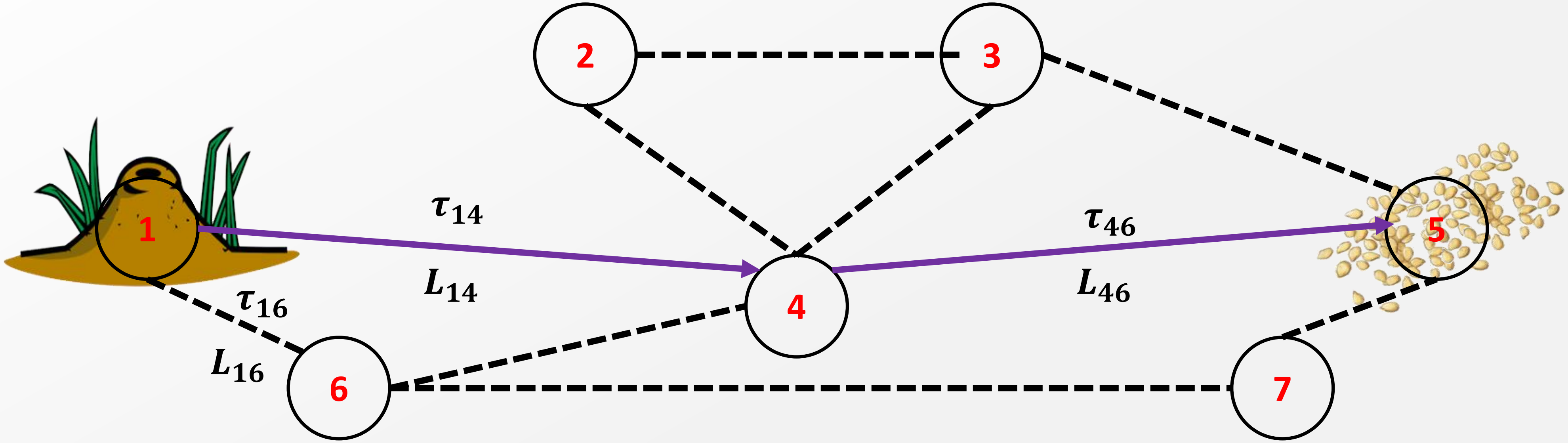
AS improves on SACO by changing the transition probability, to include heuristic information, and by adding a memory capability by the inclusion of a tabu list.

AS Algorithm: Transition Rule

- Ex. if ant_1 is currently moved at start

$$P_{14} = \frac{\tau_{14}^{\alpha} \times \boxed{\eta_{14}^{\beta}}}{\tau_{14}^{\alpha} \times \eta_{14}^{\beta} + \tau_{16}^{\alpha} \times \eta_{16}^{\beta}}$$

$$\boxed{\eta_{14}^{\beta} = \frac{1}{L_{14}}}$$



AS Algorithm: Transition Rule

- $\boldsymbol{\tau_{ij}}$ represents the *a* **posteriori effectiveness** of the move from node i to node j, as expressed in the pheromone intensity of the corresponding link, (i, j).
- $\boldsymbol{\eta_{ij}}$ represents the *a* **priori effectiveness** of the move from i to j (i.e. the attractiveness, or desirability, of the move), computed using some heuristic.
- The pheromone concentrations, $\boldsymbol{\tau_{ij}}$, indicate how **profitable** it has been in the past to make a move from i to j, serving as a memory of previous best moves.

The best balance between exploration and exploitation is achieved through proper selection of the parameters α and β.

SACO/AS Algorithm: Evaporation Rule

- To force ants to explore more, and to prevent premature convergence, pheromone intensities on links are allowed to "evaporate" at each iteration of the algorithm before being reinforced on the basis of the newly constructed paths.

- For each edge (i, j) let

$$\boldsymbol{\tau_{ij}(t) = (1-\rho) \times \tau_{ij}(t)}$$

  – $\boldsymbol{\rho}$ specifies the rate at which pheromones evaporate, causing ants to "forget" previous decisions.
  – In other words, $\boldsymbol{\rho}$ controls the influence of search history.

SACO/ AS Algorithm: Deposition Rule

- Once all ants have constructed a complete path from the origin node to the destination node, and all loops have been removed, each ant retraces its path to the source node deterministically, and deposits a pheromone amount to each edge (i, j) of the corresponding path;

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n} \Delta\tau_{ij}^{k}(t)$$

Where,

$$\Delta\tau_{ij}^{k}(t) = \frac{Q}{L^{k}(t)}$$

$L^{k}(t)$ is the length of the path constructed by ant k at time step t and Q is a large number.

SACO/AS Algorithm: Parameters

Parameter	Simple Ant Colony Optimization Algorithm (SACO)
$\boldsymbol{\alpha} \geq \mathbf{0}$	• Large values of $\boldsymbol{\alpha}$ give excessive importance to pheromone, especially the initial random pheromones, which may lead to rapid convergence to suboptimal paths. • If $\boldsymbol{\alpha}$ increases, then you increase the exploration capacity of the ants. • for smaller $\boldsymbol{\alpha}$, the algorithm generally converges to the shortest path. For complex problems, large values of $\boldsymbol{\alpha}$ result in worse convergence behavior.
$\boldsymbol{\beta} \leq \mathbf{1}$	• if $\boldsymbol{\beta}$ increases, then you promote exploitation more as the weights on the edge desirability (inversely proportional to the path length) increase.

SACO/AS Algorithm: Parameters

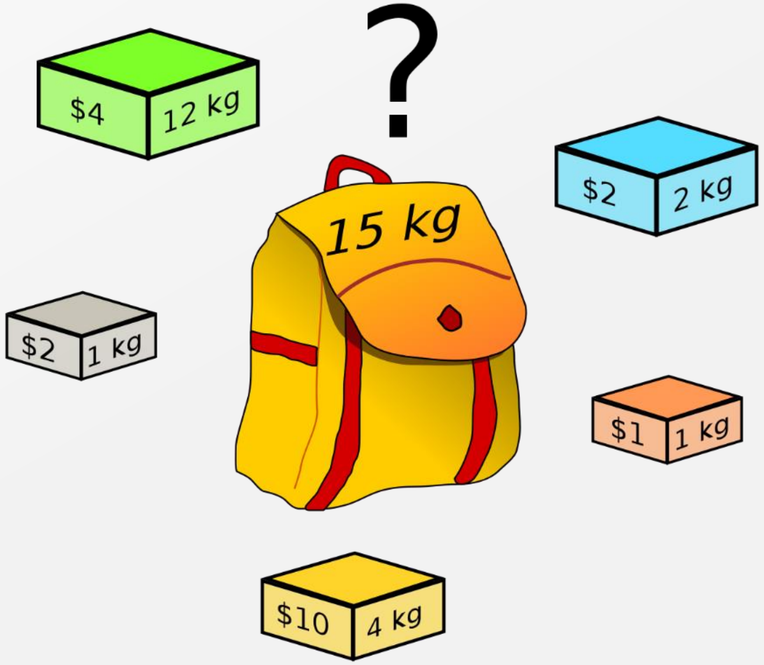
Parameter	Simple Ant Colony Optimization Algorithm (SACO)
Pheromone Decay Parameter $\boldsymbol{\rho} \in [\mathbf{0}, \mathbf{1}]$	• For large values of $\boldsymbol{\rho}$, pheromone evaporates rapidly, while small values of $\boldsymbol{\rho}$ result in slower evaporation rates. • The more pheromones evaporate, the more random the search becomes, facilitating better exploration. • If pheromone evaporates too much (a large $\boldsymbol{\rho}$ is used), the algorithm often converged to sub-optimal solutions for complex problems. • For $\boldsymbol{\rho} = \mathbf{1}$, the search is completely random. • If $\boldsymbol{\rho} = \mathbf{0}$, i.e. no evaporation, the algorithm does not converge.

SACO/AS Algorithm: Stopping Criteria

- When we reach a maximum number of iterations.
- When the objective function value has reached a certain pre-defined value.
- When there has been no improvement in the best over a number of iterations.

Case#2: Knapsack Problem

1. The Knapsack problem objective is, given a set of items (each with a weight and a value), to include as many items as possible in the given bag whose weight is constrained. This version of the problem includes profits for each item representing its priority or its value to be included in the bag.

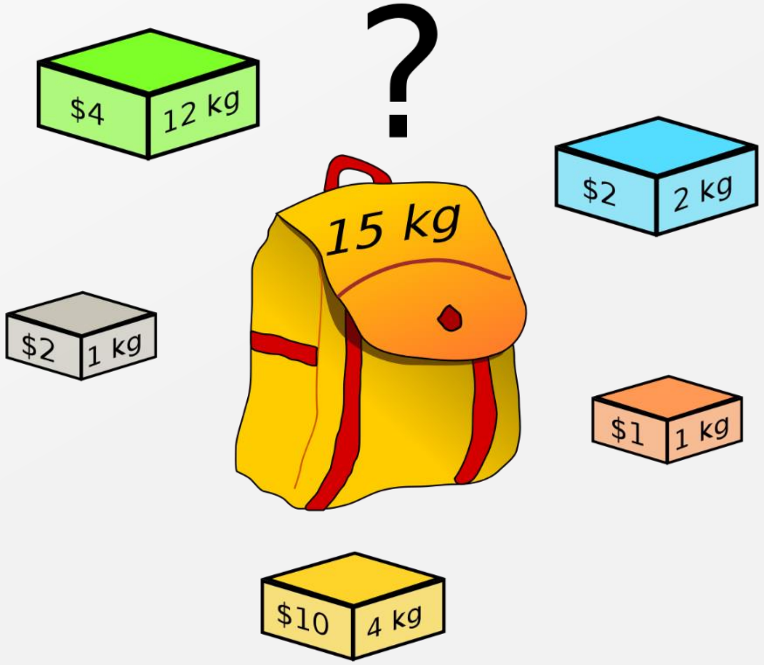


- **Decision Variables:**
  - Which item i is to be included.
- **Objective Function:**
  - Maximize number of items in bag.
  - Maximize the profit of the knapsack by adding the most valuable items

Case#2: Knapsack Problem

1. The Knapsack problem objective is, given a set of items (each with a weight and a value), to include as many items as possible in the given bag whose weight is constrained. This version of the problem includes profits for each item representing its priority or its value to be included in the bag.

- **Constraint Equation and Inequalities:**
  - The maximum allowed weight of the bag.



Knapsack Problem: Formulation

- ***Objective Function:***
  - Maximize number of items in bag

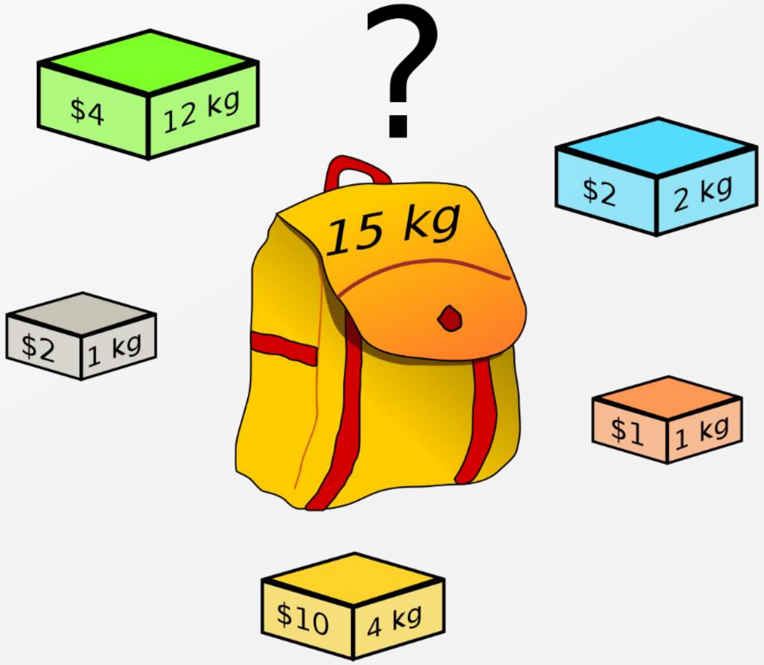
$$\max f_1 = \max \sum_{i=1}^{n} x_i$$

OR

  - Maximize the profit of the knapsack by adding the most valuable items

$$\max f_2 = \max \sum_{i=1}^{n} z_i x_i$$

$$\boxed{\begin{aligned} &z_i = profit\ of\ each\ item\ i \\ &x_i = 0 \rightarrow item\ i\ is\ not\ included\ in\ bag \\ &x_i = 1 \rightarrow item\ i\ is\ included\ in\ bag \\ &n = number\ of\ available\ items \end{aligned}}$$

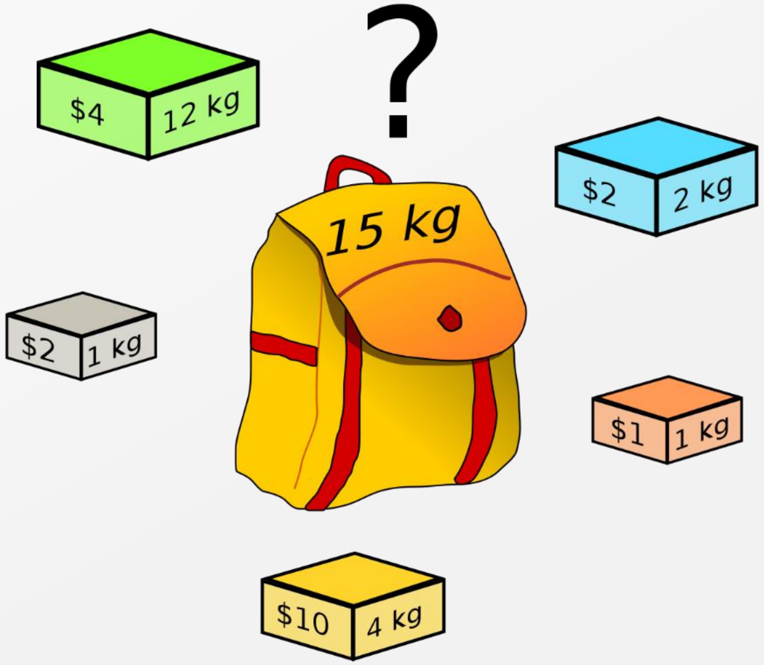


Knapsack Problem: Formulation

- ***Constraint Equation and Inequalities:***
  - The maximum allowed weight of the bag.

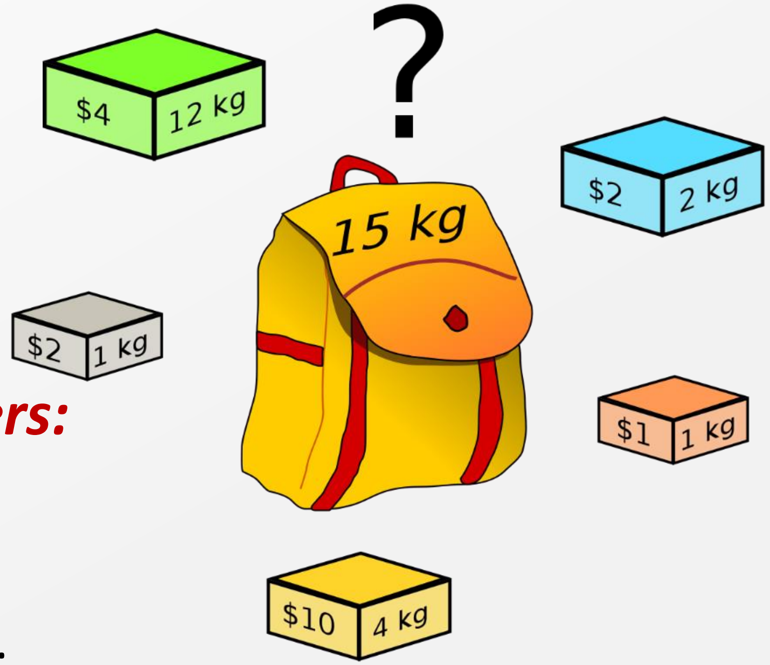
$$\sum_{i=1}^{n} w_i x_i \leq C$$

$$\boxed{\begin{aligned} &w_i = weight\ of\ each\ item\ i \\ &C = total\ weight\ allowed\ of\ the\ bag \end{aligned}}$$



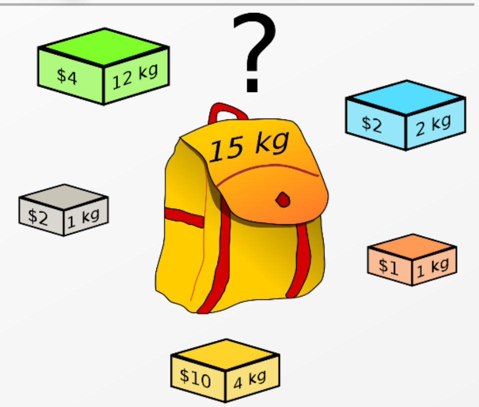
Knapsack Problem: ACO

- ***Given Data:***
  - Here, the given items properties (weights and profits) are:
    - $Number\ of\ items$ is 5
    - Weights W=[w_i]=[1, 12, 2, 1, 4]
    - Profits Z=[z_i]=[2, 4, 2, 1, 10]
  - The total allowed weight of the bag is: C=15



- ***Use ACO with the following parameters:***
  - Use 2 ants and perform only 1 iteration.
  - The initial pheromone level is 1.
  - The pheromone decay parameter $\rho = 0.5$.
  - The parameters that control the influences of the amount of pheromone on the edge and the desirability of the edge *are given as* $\alpha = \beta = 1$.

Knapsack Problem: ACO



- ***Solution Representation:***
  - Ant Representation: (Binary)

$Item_5$	$Item_4$	$Item_3$	$Item_2$	$Item_1$
x_{i-5}	x_{i-4}	x_{i-3}	x_{i-2}	$\boxed{x_{i-1}}$

$$\begin{cases} 1 \text{ if included} \\ 0 \text{ if not included} \end{cases}$$

Ex. $X=[1, 1, 0, 1, 0]$, taking into consideration only f_2

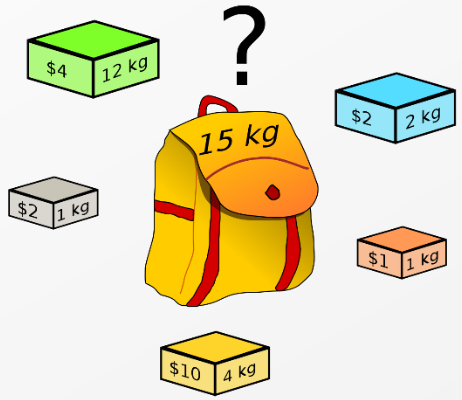
Then $f(X) = 2 + 4 + 1 = 7$, Checking the constraints $1 + 12 + 1 = 14 \leq C$

This means that the solution is feasible as the items didn't exceed the bag weight.

  - Colony Representation: (2 ants)

$\boldsymbol{x_{i1}}$	$Item_5$	$Item_4$	$Item_3$	$Item_2$	$Item_1$
$\boldsymbol{x_{i2}}$	$Item_5$	$Item_4$	$Item_3$	$Item_2$	$Item_1$

Knapsack Problem: ACO



- ***Iteration_0:***
  - Initial Solution: initially unformed yet

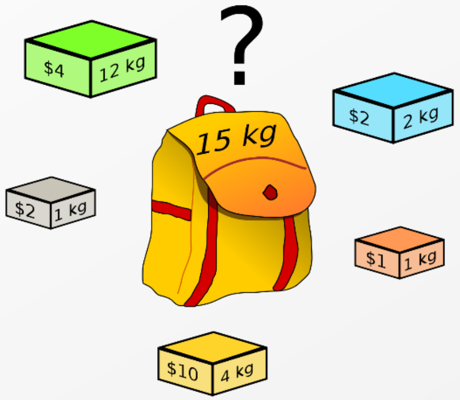
x_{01}	0	0	0	0	0
x_{02}	0	0	0	0	0

  - Initial Pheromone

τ_i	1	1	1	1	1

  - Feasibility Check $0 + 0 + 0 + 0 + 0 \leq C$

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

1. Ant 1 solution is unformed. Thus, the available moves are to add any of the 5 items.
2. Applying the state transition rule to find the probability of adding item j with pheromone level τ_j and desirability η_j:

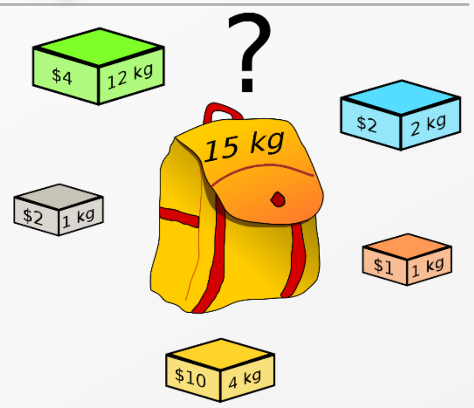
$$p_j = \frac{\tau_j^{\alpha}\, \eta_j^{\beta}}{\sum_{i=1}^{n_r} \tau_i^{\alpha}\, \eta_i^{\beta}} \; ; \; \eta_j = \frac{z_j}{w_j}$$

3. The desirability here is formulated as shown in order to promote higher profit and less weights.

Item Num (j)	Probability (p_j)
1	0.2927
2	0.0488
3	0.1463
4	0.1463
5	0.3659

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

4. To select an item to add to the bag, we will use the ***roulette wheel selection***. First, we will compute a cumulative probability table for the available items where:

$$q_1 = p_1, q_j = q_{j-1} + p_j$$

5. Then, generate a random number r between 0 and 1 and select the item with cumulative probability $q_j \geq r$.

6. This method always guarantees a selection.

$$r = 0.4 \rightarrow q_3 = 0.4878 \geq r$$

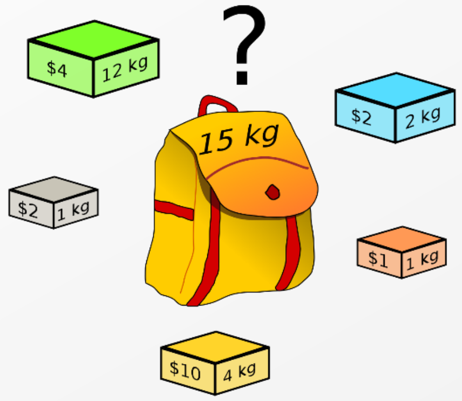
Therefore, Ant 1 selects item 3 to add.

$$X = [0, 0, 1, 0, 0]$$

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.2927	0.2927
2	0.0488	0.3415
3	0.1463	0.4878
4	0.1463	0.6341
5	0.3659	1

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

Now, Ant 1 selects item 3 to add.

$$X = [0, 0, 1, 0, 0]$$

We have to check the constraint satisfaction:

$$\sum_{i=1}^{n} w_i x_i \leq C \rightarrow 2 \leq 15$$

Therefore, Ant 1 solution is accepted so far and the remaining weight:

$$C_r = 13$$

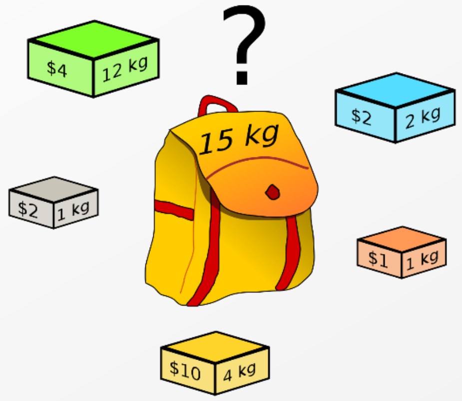
and the remaining number of items:

$$n_r = 4$$

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.2927	0.2927
2	0.0488	0.3415
3	0.1463	0.4878
4	0.1463	0.6341
5	0.3659	1

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

8. Now, the available moves are to add any of items 1,2,4 and 5.
9. Applying the state transition rule to find the probability of adding item j with pheromone level τ_j and desirability η_j:

$$p_j = \frac{\tau_j^{\alpha}\,\eta_j^{\beta}}{\sum_{i=1}^{n_r} \tau_i^{\alpha}\,\eta_i^{\beta}} \;;\; \eta_j = \frac{z_j}{w_j}$$

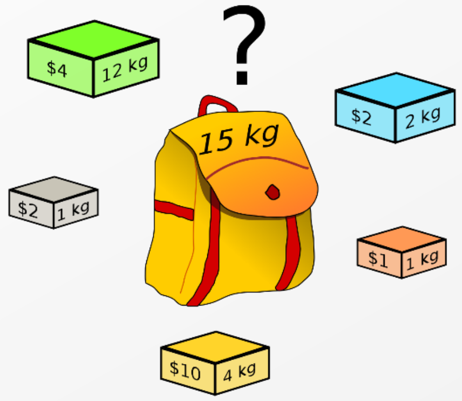
The probabilities and cumulative probabilities are shown in the table.

10. Generate a random number to apply roulette wheel selection.

$$r = 0.8 \rightarrow q_5 = 1 \geq r$$

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.3429	0.3429
2	0.0571	0.4
4	0.1714	0.5714
5	0.4286	1

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

Therefore, Ant 1 adds item 5 to the bag.

Now, Ant 1 becomes:

$$X = [0, 0, 1, 0, 1]$$

We have to check the constraint satisfaction:

$$\sum_{i=1}^{n} w_i x_i \leq C \rightarrow 2 + 4 = 6 \leq 15$$

Therefore, Ant 1 solution is accepted so far and the remaining weight:

$$C_r = 9$$

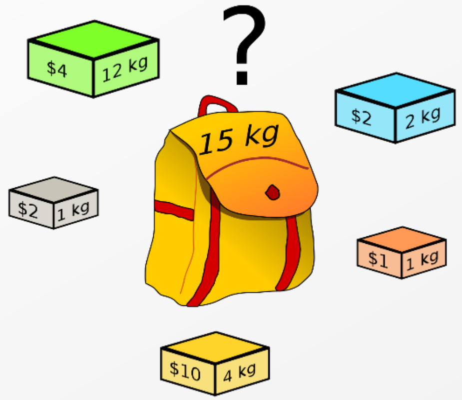
and the remaining number of items:

$$n_r = 3$$

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.3429	0.3429
2	0.0571	0.4
4	0.1714	0.5714
5	0.4286	1

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

11. Now, the available moves are to add any of items 1,2 and 4.
12. Applying the state transition rule to find the probability of adding item j with pheromone level τ_j and desirability η_j:

$$p_j = \frac{\tau_j^{\alpha}\ \eta_j^{\beta}}{\sum_{i=1}^{n_r} \tau_i^{\alpha}\ \eta_i^{\beta}}\ ;\ \eta_j = \frac{z_j}{w_j}$$

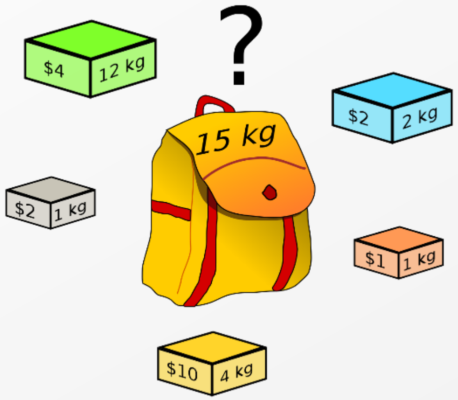
The probabilities and cumulative probabilities are shown in the table

13. Generate a random number to apply roulette wheel selection.

$$r = 0.65 \rightarrow q_2 = 0.7 \geq r$$

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.6	0.6
2	0.1	0.7
4	0.3	1

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

Therefore, Ant 1 adds item 2 to the bag.

Now, Ant 1 becomes:

$$X = [0, 1, 1, 0, 1]$$

We have to check the constraint satisfaction:

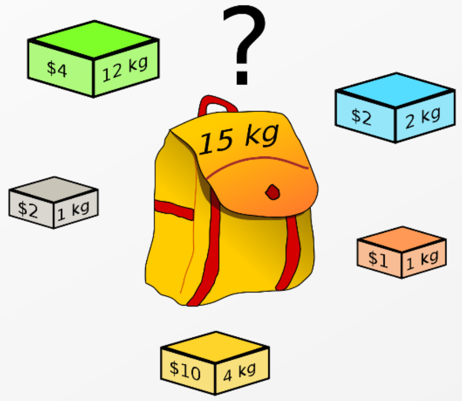
$$\sum_{i=1}^{n} w_i x_i \leq C \rightarrow 2 + 4 + 12 = 18 > 15$$

Therefore, this solution is rejected for violating the constraint.

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.6	0.6
2	0.1	0.7
4	0.3	1

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

Another random number is generated:

$$r = 0.2 \rightarrow q_1 = 0.6 \geq r$$

Therefore, Ant 1 adds item 1 to the bag.

$$X = [1, 0, 1, 0, 1]$$

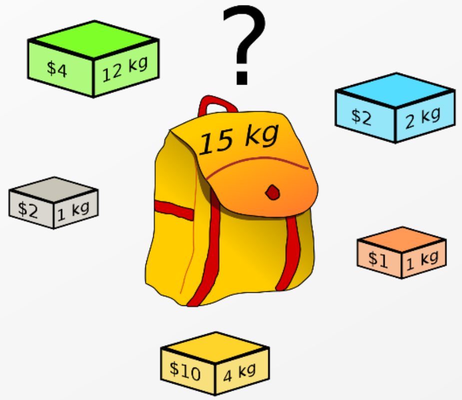
We have to check the constraint satisfaction:

$$\sum_{i=1}^{n} w_i x_i \leq C \rightarrow 2 + 4 + 1 = 7 \leq 15$$

Therefore, Ant 1 solution is accepted so far

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
1	0.6	0.6
2	0.1	0.7
4	0.3	1

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

14. Now, the available moves are to add any of items 2 and 4.
15. Applying the state transition rule to find the probability of adding item j with pheromone level τ_j and desirability η_j:

$$p_j = \frac{\tau_j^{\alpha}\ \eta_j^{\beta}}{\sum_{i=1}^{n_r} \tau_i^{\alpha}\ \eta_i^{\beta}}\ ;\ \eta_j = \frac{z_j}{w_j}$$

The probabilities and cumulative probabilities are shown in the table.

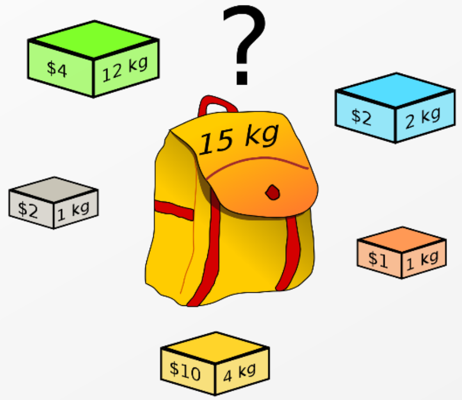
Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
2	0.25	0.25
4	0.75	1

16. Generate a random number to apply roulette wheel selection.

$$r = 0.9 \rightarrow q_4 = 1 \geq r$$

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_1:**

Therefore, Ant 1 adds item 4 to the bag.

Now, Ant 1 becomes:

$$X = [1, 0, 1, 1, 1]$$

We have to check the constraint satisfaction:

$$\sum_{i=1}^{n} w_i x_i \leq C \rightarrow 2 + 4 + 1 + 2 = 9 \leq 15$$

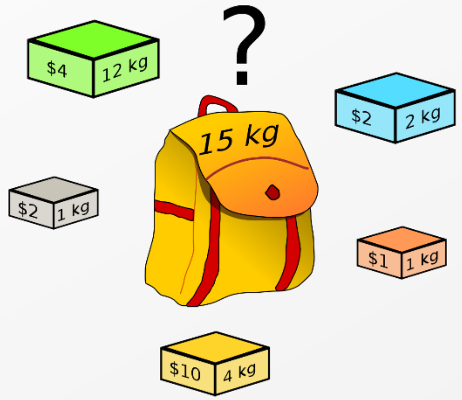
Therefore, Ant 1 solution is accepted with objective function value:

$$f(X) = \sum_{i=1}^{n} z_i x_i = 15$$

Item Num (j)	Probability (p_j)	Cumulative Probability (q_j)
2	0.25	0.25
4	0.75	1

- Tutorial 7 Recap
- Ant Colony Optimization
- **Application to Knapsack Problem**
- Techniques Comparison

Knapsack Problem: ACO



- ***Iteration_1:***
  - **For Ant_2:**

1. Following the same procedure, Ant 2 is:

$$X = [1, 1, 0, 1, 0]$$

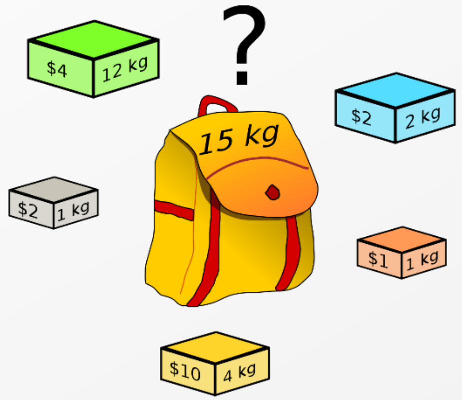
We have to check the constraint satisfaction:

$$\sum_{i=1}^{n} w_i x_i \leq C \rightarrow 1 + 12 + 1 = 14 \leq 15$$

Therefore, Ant 2 solution is accepted with objective function value:

$$f(X) = \sum_{i=1}^{n} z_i x_i = 7$$

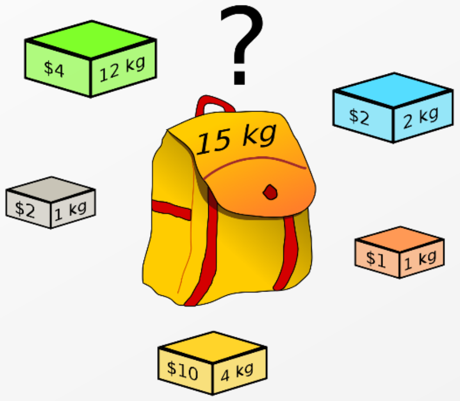
Knapsack Problem: ACO



- ***Iteration_1:***
  - **Update Pheromone:**

1. **Pheromone Evaporation:**
   First, all pheromone trails get evaporated.
   $$\tau_i = (1-\rho)\tau_i$$
   $$\tau = (1-0.5) \times 1 = 0.5$$
2. **Pheromone Deposit:**
   Now, each ant deposits pheromone on its path
   - using the **Ant Quantity Model**.
     $$\tau_i(t+1) = \tau_i(t) + \frac{z_i}{w_i}$$
   - using the **Ant Cycle model**.
     $$\tau_i(t+1) = \tau_i(t) + Q \times \sum_{i=1}^{n} z_i x_i$$

Knapsack Problem: ACO



- ***Iteration_1:***
  - **Update Pheromone:**
    a. Pheromone Deposit for Ant 1 → $X_1 = [1, 0, 1, 1, 1]$:

    $$\tau_1 = 0.5 + \frac{2}{1} = 2.5, \qquad \tau_3 = 0.5 + \frac{2}{2} = 1.5,$$
    $$\tau_4 = 0.5 + \frac{1}{1} = 1.5, \qquad \tau_5 = 0.5 + \frac{10}{4} = 3$$

    b. Pheromone Deposit for Ant 2 → $X_2 = [1, 1, 0, 1, 0]$:

    $$\tau_1 = 2.5 + \frac{2}{1} = 4.5, \qquad \tau_2 = 0.5 + \frac{4}{12} = 0.8333,$$
    $$\tau_4 = 1.5 + \frac{1}{1} = 2.5$$

  - Thus, the pheromone matrix becomes:

    $$\tau_i = [4.5, 0.8333, 1.5, 2.5, 3]$$

    And the iterations go on …

Using Ant Quantity Model

References

- Parsons, S. (2005). Ant Colony Optimization by Marco Dorigo and Thomas Stützle, MIT Press, 305 pp., $40.00, ISBN 0-262-04219-3. The Knowledge Engineering Review, 20(1), 92.
- Keller, J. M., Liu, D., & Fogel, D. B. (2016). Fundamentals of computational intelligence: neural networks, fuzzy systems, and evolutionary computation. John Wiley & Sons.
- Engelbrecht, A. P. (2007). Computational intelligence: an introduction. John Wiley & Sons.

See you Next time … ☺