Formulation and solution technique for agricultural waste collection and transport network design

Trung Hieu Tran, Thu Ba T. Nguyen, Hoa Sen T. Le, Duc Chinh Phung

DSSC master's student
Pietro Morichetti

Chapters

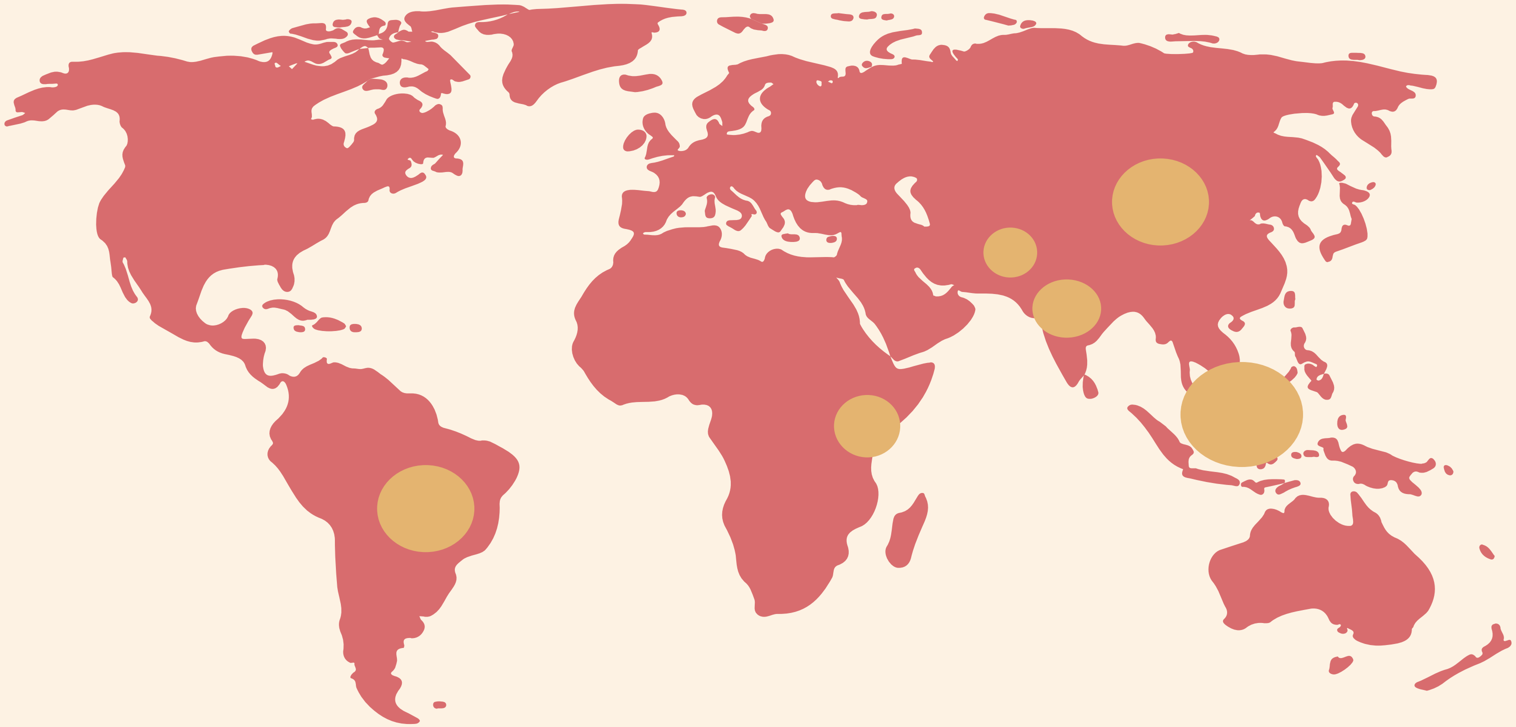
01

Problem Introduction

Agricultural waste management

- In developing countries, agricultural sectors have rapidly developednto become major contributions in the economy.
- Agricultural waste management has posed a challange for rural planner due to lack of efficient planning tool.
- Burning the agricultural waste at fields after each harvesting season is the present solution, this caused air and water pollution in rural areas.

Agricultural waste around the world for developing countries



source: International Journal of Environmental Research and Public Health

Vietnam 2019

45,22 million tons

Rice straw

10 millions tons

Vegetable by-products

6,33 millions tons

Maize by-products

...

source: Ministry of Agriculture and Rural Development of Vietnam

An efficient way to collect agricultural waste

02

Modelling

Location-Assignment-Routing-Problem

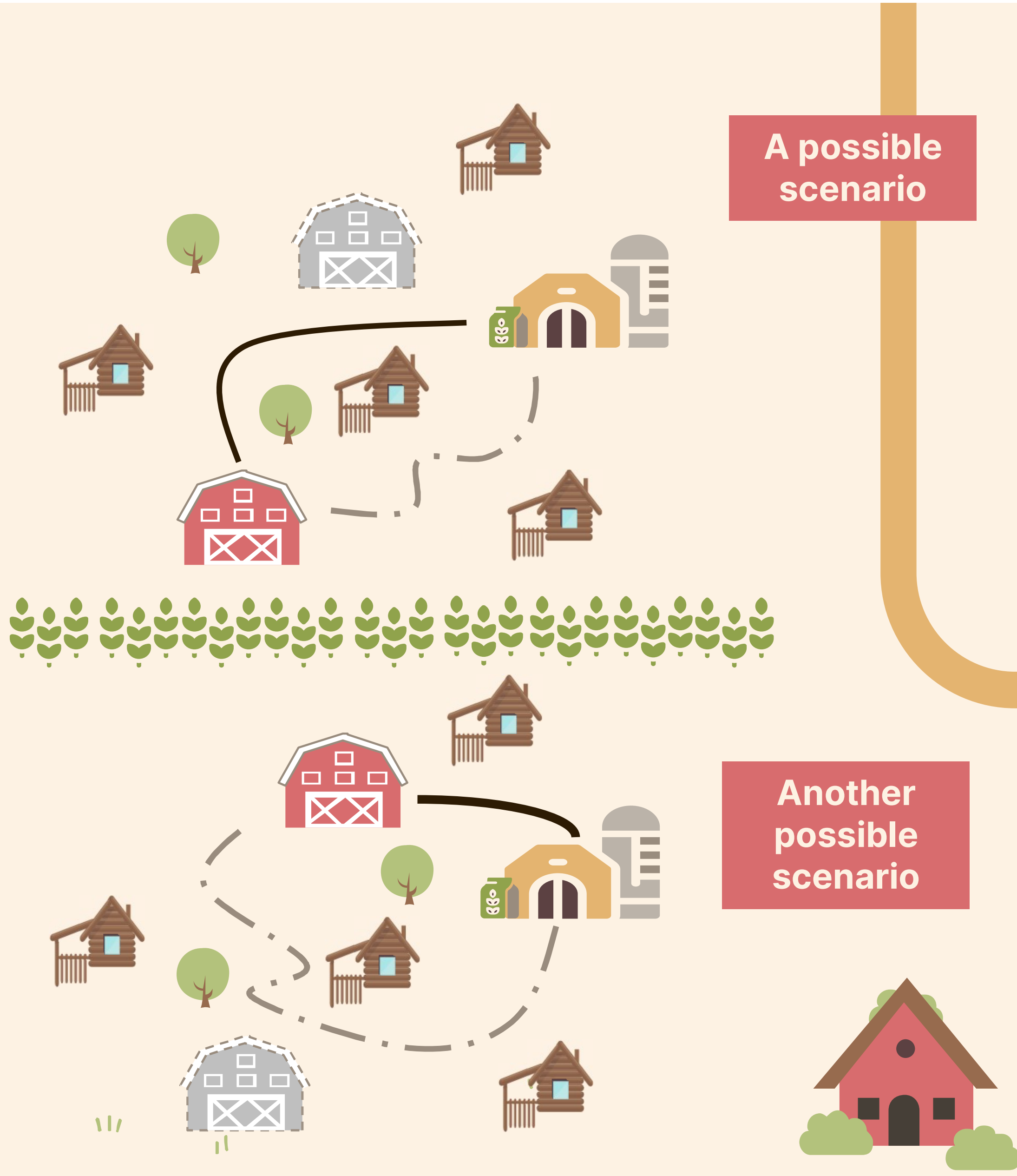
Logistic problem

Which and how many places should be chosen where storages should be built to ensure coverage over all cultivation fields?

Location Assignment Problem

Which roads should trucks take to visit all the storages and bring agricultural waste to the fertiliser factory?

Routing Problem



Model Formulation (1/3)

Sets and indexes

I Set of agricultural waste fields (indexed by) i
J Set of storages (indexed by) j
J_0 Set of storages and the facility, i.e., $J_0 = J \cup 0$ indexed by u, v

Paramenters

n Number of agricultural waste fields
m Number of storages
f_jFixed cost to open a storage j
$c_{i,j} c_{u,v}$ Distance between nodes i, j or u, v
d_i Ammount of agricultural waste to be collected at field i
q_j Capacity of storage j
k Number of vehicles
Q Capacity of vehicle

Decision variables

$$X_j = \begin{cases} 1 & \text{if storage is located at site j} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{if field i is served by storage j} \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{uv} = \begin{cases} 1 & \text{if a vehicle travels from node u to node v} \\ 0 & \text{otherwise} \end{cases}$$

T_u, T_v Auxiliary variables

Model Formulation (2/3)

storage cost — shipping cost — transportation cost

$$\min \sum_{j\in J} f_j \cdot X_j + \sum_{i\in I}\sum_{j\in J} c_{ij}\cdot d_i \cdot Y_{ij} + \sum_{u\in J_0}\sum_{v\in J_0:u!=v} c_{uv}\cdot Z_{uv}$$

$$(C1)\ \sum_{j\in J} Y_{ij} = 1 \quad \text{for each} \quad i \in I$$

$$(C2)\ \sum_{i\in I} d_i \cdot Y_{ij} \le q_j \cdot X_j \quad \text{for each} \quad j \in J$$

$$(C3)\ \sum_{u\in J} Z_{u0} = k$$

$$(C4)\ \sum_{v\in J} Z_{0v} = k$$

$$(C5)\ \sum_{u\in J_0:u!=v} X_u \cdot Z_{uv} = X_v \quad \text{for each} \quad v \in J$$

$$(C6)\ \sum_{v\in J_0:v!=u} X_v \cdot Z_{uv} = X_u \quad \text{for each} \quad u \in J$$

$$(C7)\ T_u - T_v + Q\cdot Z_{uv} \le Q - \frac{1}{k}\cdot\sum_{i\in I} d_i \cdot Y_{iv} \quad \text{for each} \quad u,v \in J : u! = v$$

$$(C8)\ \frac{1}{k}\cdot\sum_{i\in I} d_i\cdot Y_{iv} \le T_u \le Q \quad \text{for each} \quad u \in J$$

$$(C9)\ X_j \in \{0,1\} \quad \text{for each } j \in J$$

$$(C10)\ Y_{ij} \ge 0 \quad \text{for each} \quad i \in I, \text{for each} \quad j \in J$$

$$(C11)\ Z_{uv} \in \{0,1\} \quad \text{for each } u,v \text{ in } J_0 : u \neq v$$

$$(C12)\ T_u \ge 0 \quad \text{for each } u \in J$$

Not required to empty the storages

Legenda

- Capacitated facility location
- TSP adjusted for CVRP
- Assignment

Model Formulation (3/3)

Product of decision vari

$$(C5) \quad \sum_{u \in J_0 : u != v} X_u \cdot Z_{uv} = X_v \quad \text{for each} \quad v \in J$$

$$(C6) \quad \sum_{v \in J_0 : v != u} X_v \cdot Z_{uv} = X_u \quad \text{for each} \quad u \in J$$

$$u \in J_0 : u != v$$
$$v \in J_0 : v != u$$

$$(C5) \quad \sum_{u \in J : u != v} X_u \cdot Z_{uv} = X_v \quad \text{for each} \quad v \in J$$

$$(C6) \quad \sum_{v \in J : u != v} X_v \cdot Z_{uv} = X_u \quad \text{for each} \quad u \in J$$

$$D_0 = 1 \quad \text{location of facility}$$
$$D_0 \cdot Z_{0v} = 1 \quad \text{for each} \quad v \in J$$
$$D_0 \cdot Z_{u0} = 1 \quad \text{for each} \quad u \in J$$

Linearization of non-linear constraints

$W_{uv} = X_u \cdot Z_{uv}$ for ea $, v \in J_0 : u \neq v$

$W'_{uv} = X_v \cdot Z_{uv}$ for each $u, v \in J_0 : u \neq v$

$$(C13) \quad \sum_{v \in J_0 : u \neq v} W_{uv} \leq X_u \quad \text{for each} \quad v \in J$$
$$(C14) \quad W_{uv} \leq X_u \quad \text{for each} \quad u, v \in J_0 : u \neq v$$
$$(C15) \quad W_{uv} \leq Z_{uv} \quad \text{for each} \quad u, v \in J_0 : u \neq v$$
$$(C16) \quad W_{uv} \geq X_u + Z_{uv} - 1 \quad \text{for each} \quad u, v \in J_0 : u \neq v$$
$$(C17) \quad W_{uv} \in 0, 1 \quad \text{for each} \quad u, v \in J_0 : u \neq v$$

$$(C18) \quad \sum_{v \in J_0 : u \neq v} W'_{uv} \leq X_u \quad \text{for each} \quad u \in J$$
$$(C19) \quad W'_{uv} \leq X_v \quad \text{for each} \quad u, v \in J_0 : u \neq v$$
$$(C20) \quad W'_{uv} \leq Z_{uv} \quad \text{for each} \quad u, v \in J_0 : u \neq v$$
$$(C21) \quad W'_{uv} \geq X_v + Z_{uv} - 1 \quad \text{for each} \quad u, v \in J_0 : u \neq v$$
$$(C22) \quad W'_{uv} \in 0, 1 \quad \text{for each} \quad u, v \in J_0 : u \neq v$$

03

Case studies & Scalability analysis

LARP optimal solution

Three scenarios for a proof-of-concept

X 3 Main facilities where bio-fertilizer is produced from the agricultural wastes (only one per scenario)

X 6 Storages with a certain capacity and fix opening cost

X 7 Cluster of agricultural fields with a certain amount of agricultural waste

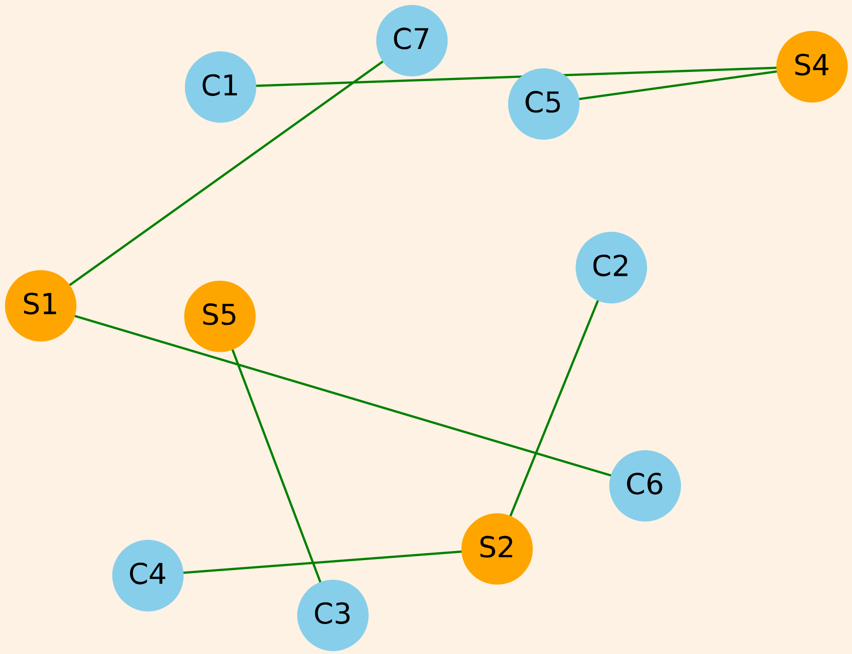
X 3 Truks with a certain capacity and capable to move agricultural wastes from storages to the main facility

Graphical representations (1/2)

- All the scenarios share the same assignments among storages and agricultural fields
- Storages «S3» and «S6» are not considered in the result, it means that opening a storage to these locations is not suggested

Location cost = 400
Assignment cost = 80,2

Same result as the **paper**!

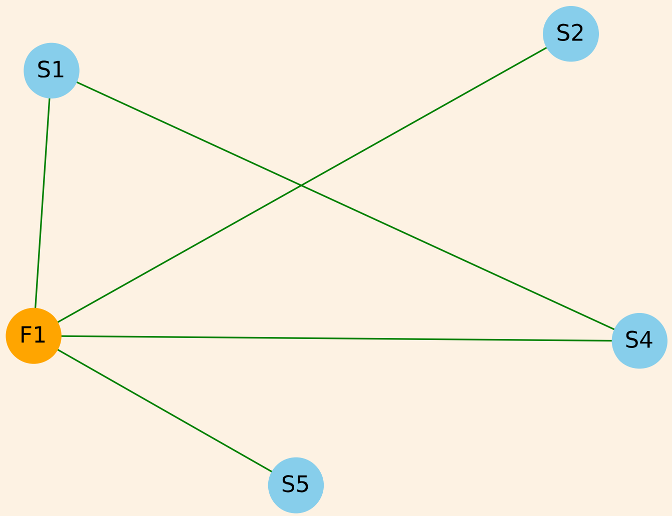


Graphical representations (2/2)

Same results as the paper!

scenario 1

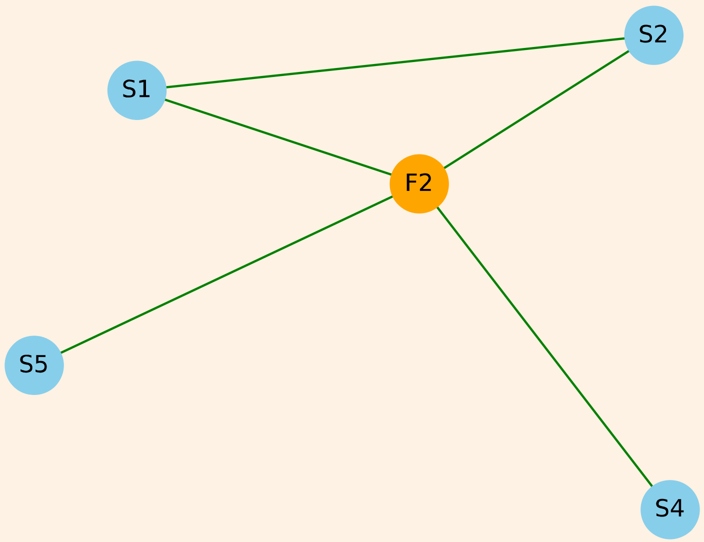
Transportation cost = 57,0
Total cost = 537,2



R1 composed by: F1, S1, S4, F1
R2 composed by: F1, S2, F1
R3 composed by: F1, S5, F1

scenario 2

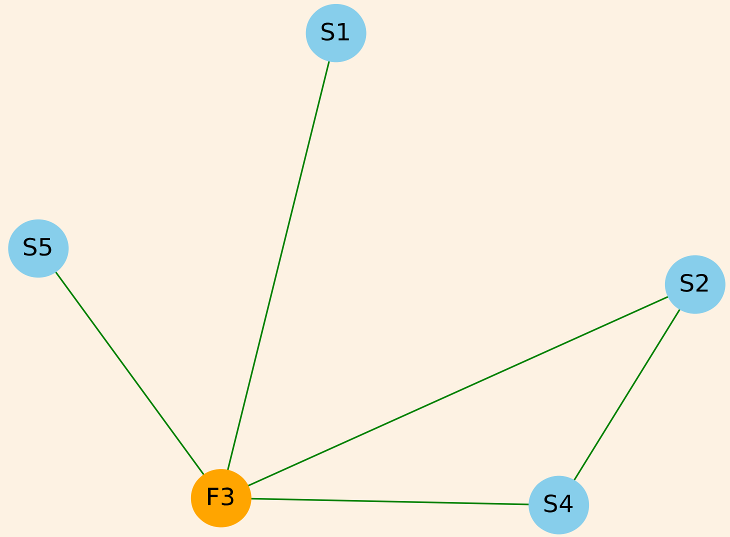
Transportation cost = 119,2
Total cost = 599,4



R1 composed by: F2, S1, S2, F2
R2 composed by: F2, S4, F2
R3 composed by: F2, S5, F2

scenario 3

Transportation cost = 55,0
Total cost = 535,2



R1 composed by: F3, S1, F3
R2 composed by: F3, S4, S2, F3
R3 composed by: F3, S5, F3

Scalability analysis

Parameters used for the experiment:

Facility = ‘F’

Num iterations = 10

Num fields = 100

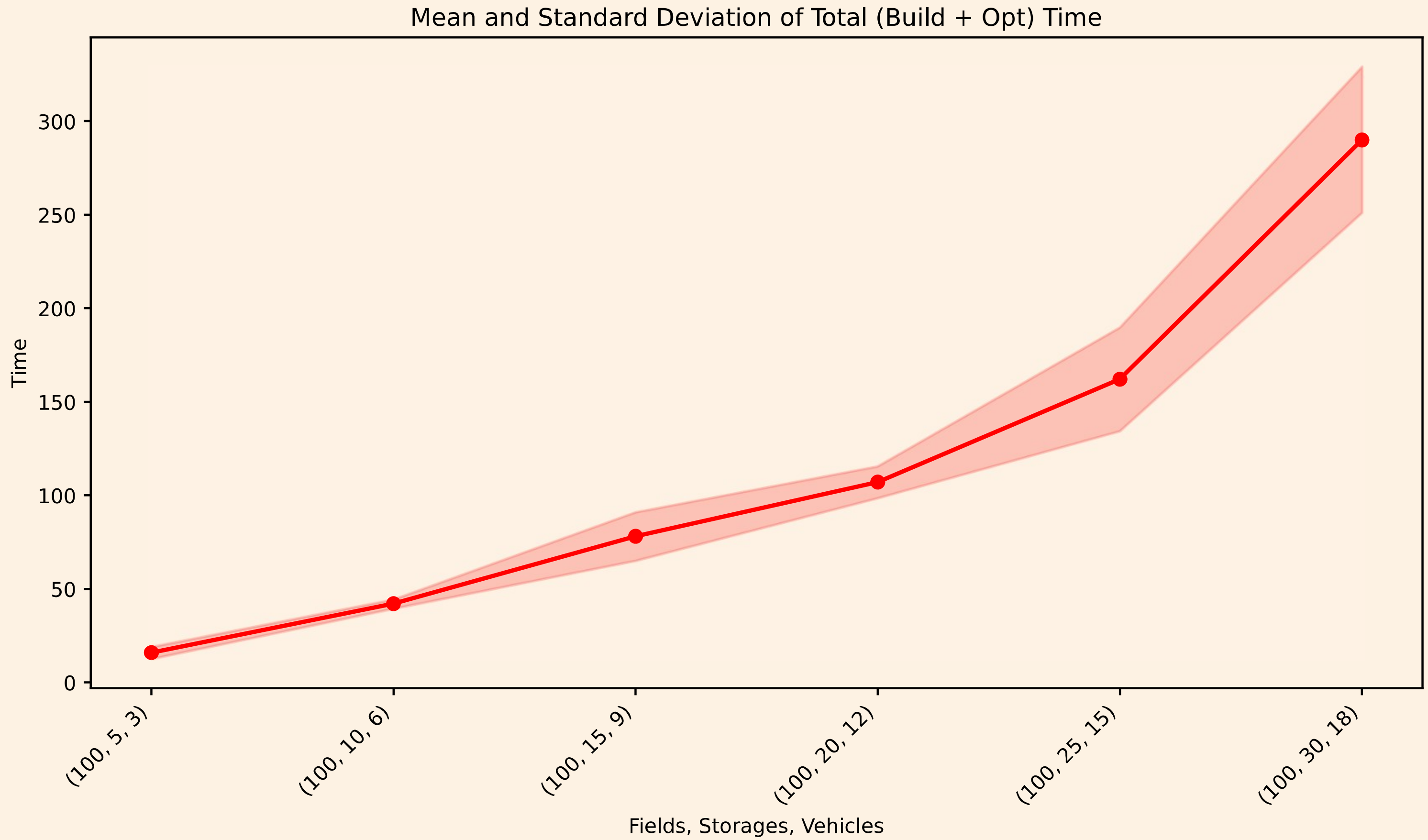
Num storages = [5, 10, 15, 20, 25, 30]

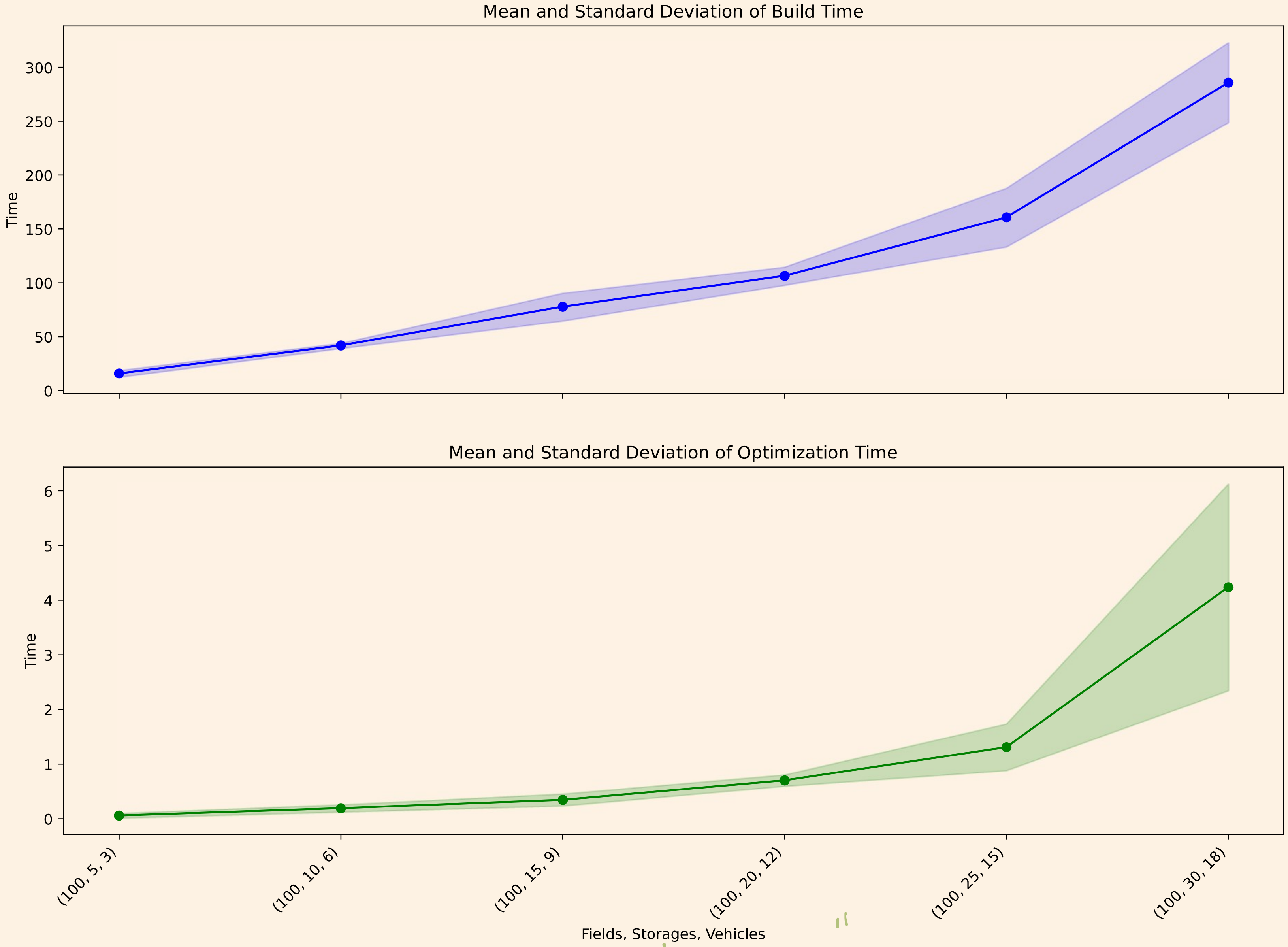
Num vehicles = [3, 6, 9, 12, 15, 18]

Vehicle capacity = 2000 (tons)

Proposed model formulation is clearly not able to provide an optimal solution for different sizes of the problem, in «acceptable» timeframe.

Scalability plot match the optimization model **complexity** of O(m2 + nm).





Major contribution to the Scalability Analysis comes from the **Build** phase

Optimization phase starts to relatively impact later

04

Water Flow Algorithm

A meta-heuristic improvement

Water Flow Algorithm

A couple of disclaimers

Proposed description of the WFA is short, poor in detail and, at times, confusing

↓

A «personal» interpretation is implemented for this project

The authors of the paper proposed a parallelised version of the WFA

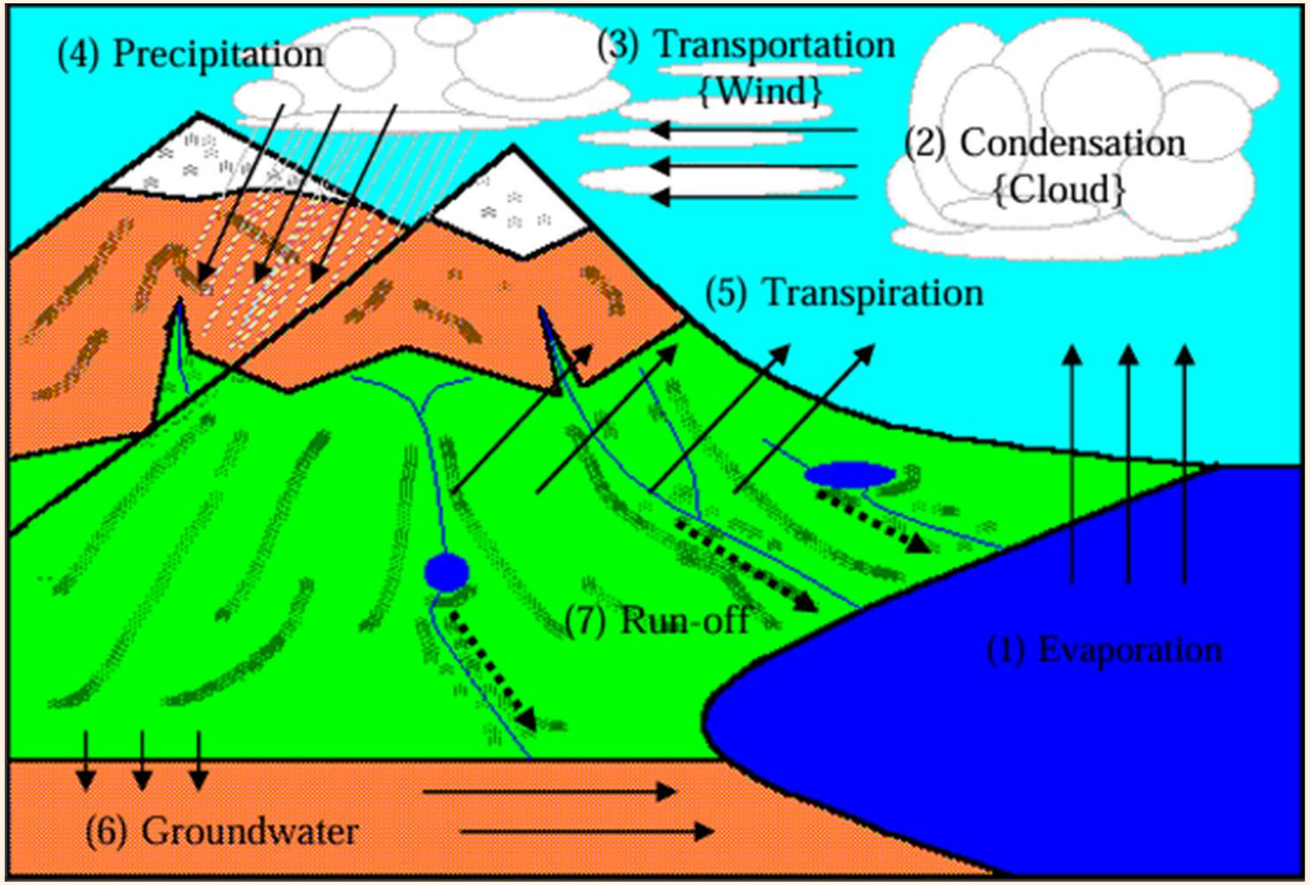
↓

Exclusively the WFA is shown

↓

A comparison with the LARP Scalability Analysis is not possible

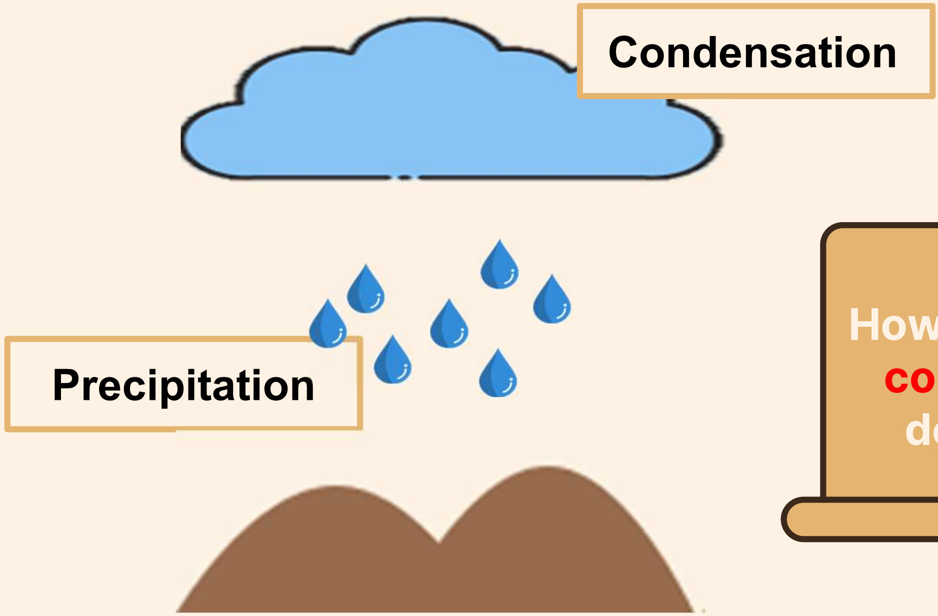
Basic Components of the Hydrological Cycle



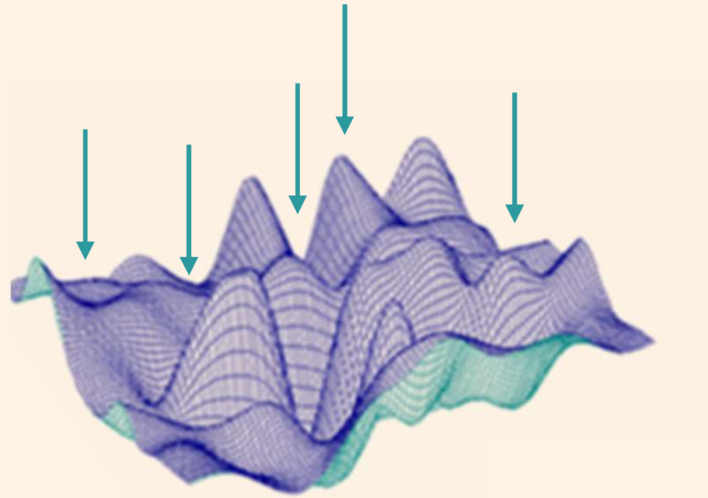
We focus just on the following stages:

- **(2) Condensation**
- **(4) Precipitation**
- **(6) Groundwater**

From Water Cycle to Water Flow Formulation (1/3)



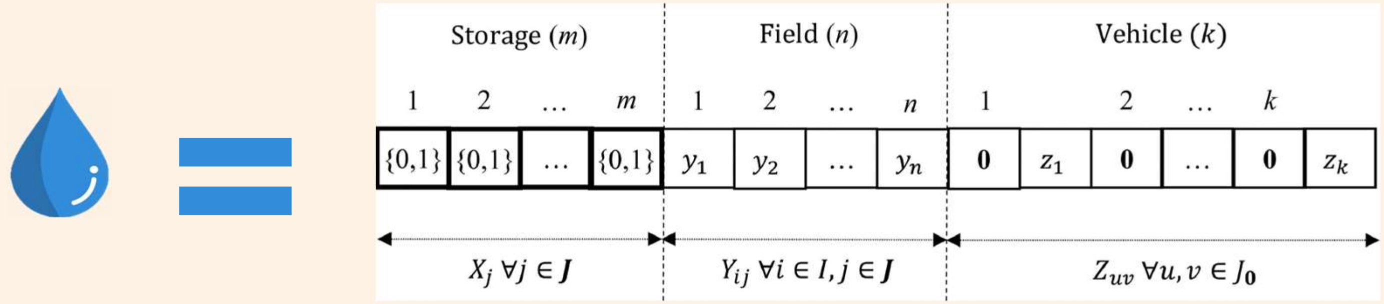
How is a DOW **concretely** defined?



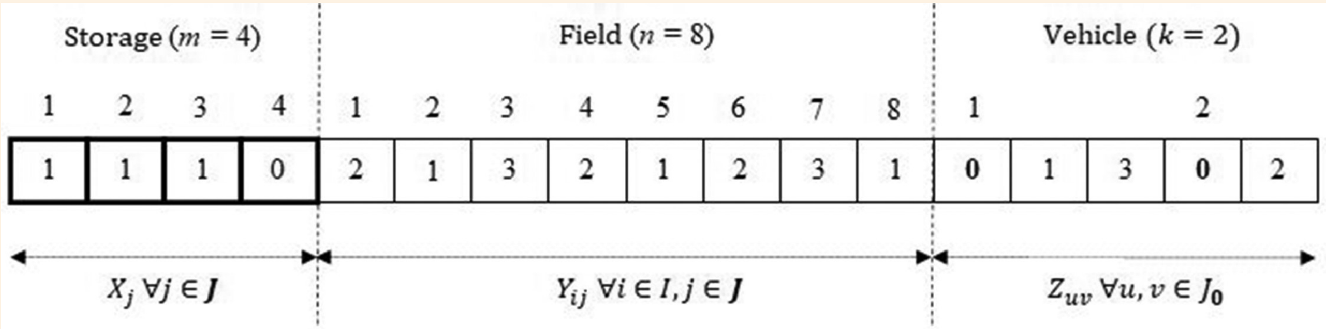
Drop-of-Waters (DOWs) are generated from a cloud and they falls on the earth

A «DOW» identify a location (point) in the solution space for the LARP model, and a set of DOWs is randomly generated

From Water Cycle to Water Flow Formulation (2/3)



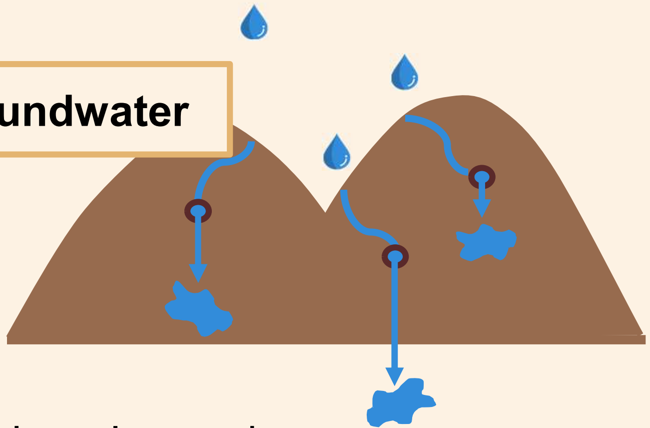
A «DOW» represents a solution for LARP model



An example of DOW/Solution formulation

From Water Cycle to Water Flow Formulation (3/3)

Groundwater



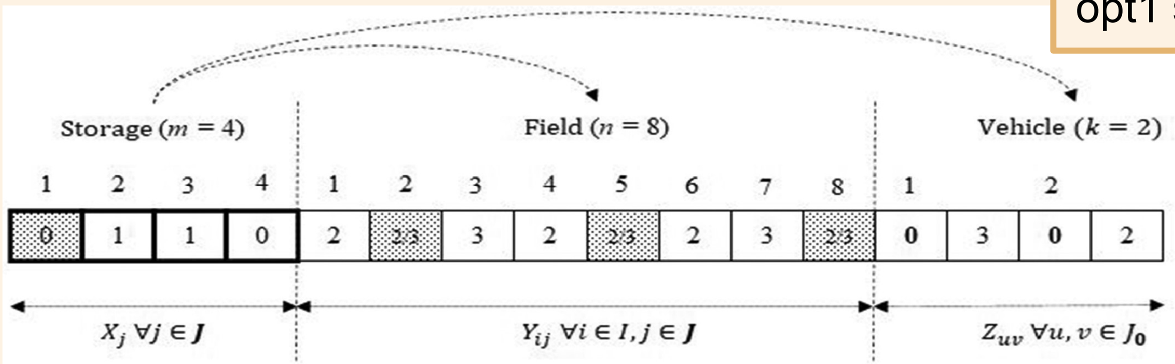
<u>Local search</u>

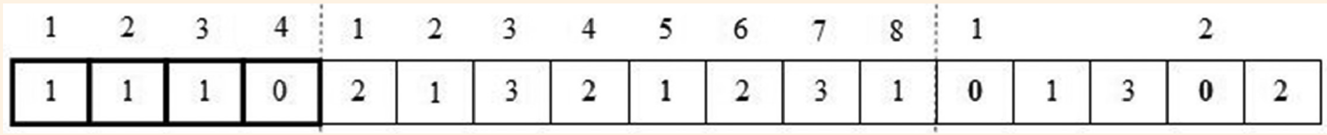
«DOWs» on the ground start to move down due to the Gravity force until they stop somewhere

<u>Erosion process</u>

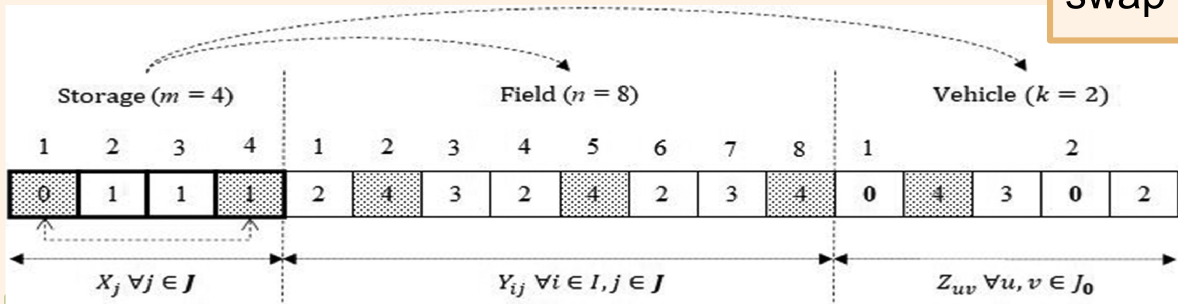
«DOWs», gathered in puddle, begin to seepage the ground

opt1 strategy





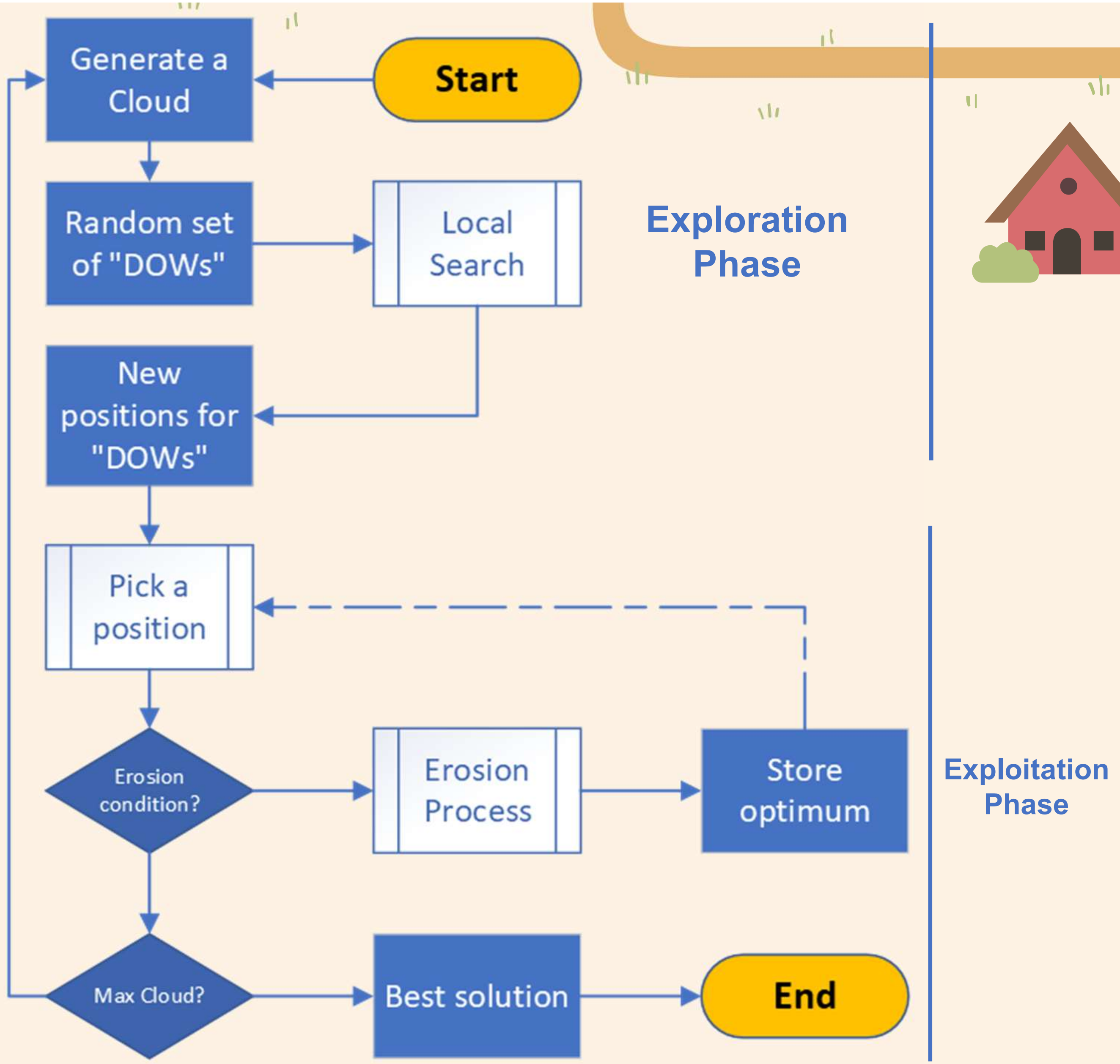
swap strategy

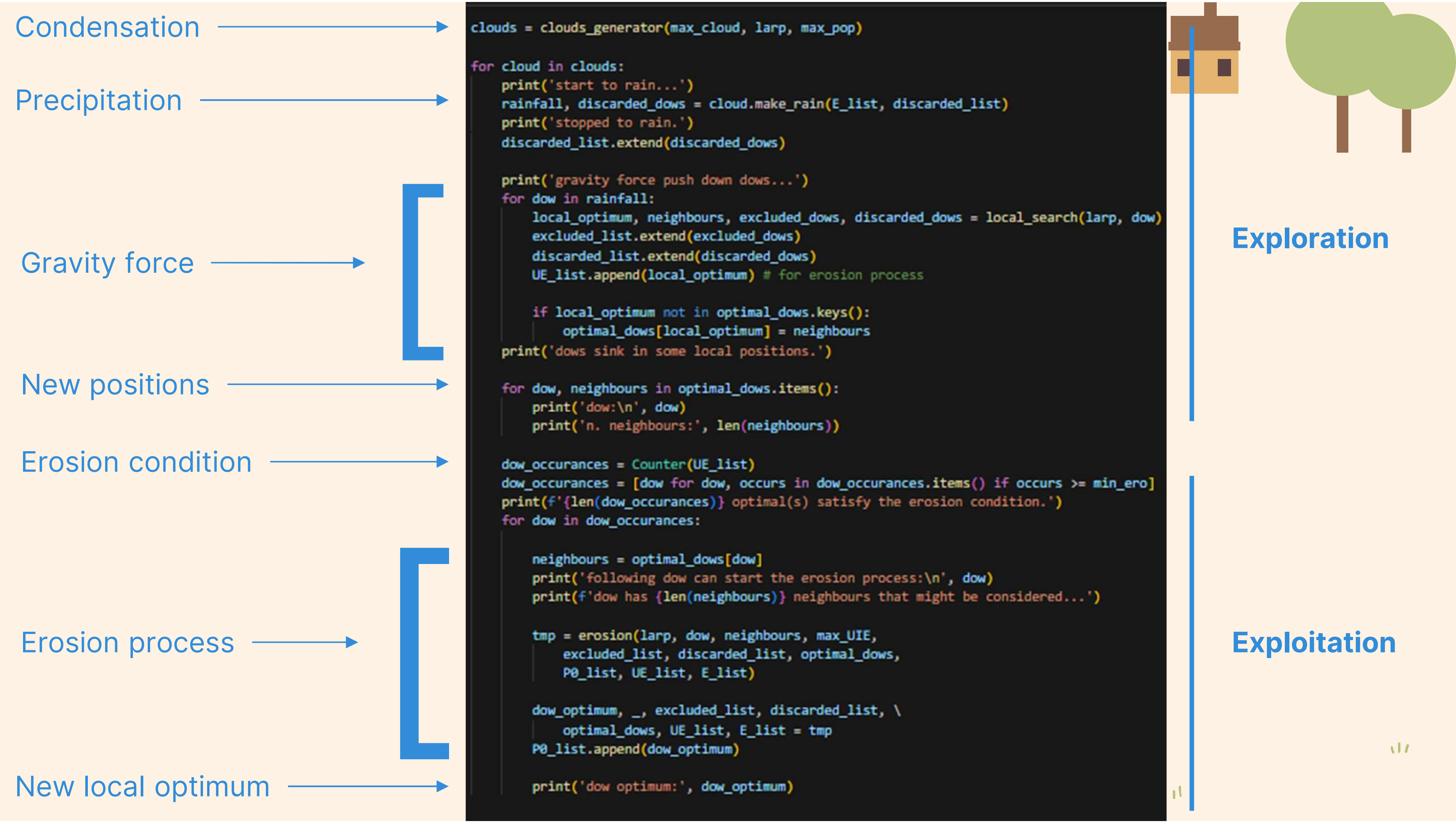


Simplified flow-chart for the Water Flow Algorithm

During the **Exploration phase** a Cloud is generated, the cloud rains a fixed number of water drops randomly which fall on the earth at some position and start to move somewhere down

During the **Exploitation phase** those water drops which are gathered in a single point start to erode the ground and infiltrate through the ground until they stop somewhere in a groundwater






```python
clouds = clouds_generator(max_cloud, larp, max_pop)

for cloud in clouds:
    print('start to rain...')
    rainfall, discarded_dows = cloud.make_rain(E_list, discarded_list)
    print('stopped to rain.')
    discarded_list.extend(discarded_dows)

    print('gravity force push down dows...')
    for dow in rainfall:
        local_optimum, neighbours, excluded_dows, discarded_dows = local_search(larp, dow)
        excluded_list.extend(excluded_dows)
        discarded_list.extend(discarded_dows)
        UE_list.append(local_optimum) # for erosion process

        if local_optimum not in optimal_dows.keys():
            optimal_dows[local_optimum] = neighbours
    print('dows sink in some local positions.')

    for dow, neighbours in optimal_dows.items():
        print('dow:\n', dow)
        print('n. neighbours:', len(neighbours))

    dow_occurances = Counter(UE_list)
    dow_occurances = [dow for dow, occurs in dow_occurances.items() if occurs >= min_ero]
    print(f'{len(dow_occurances)} optimal(s) satisfy the erosion condition.')
    for dow in dow_occurances:

        neighbours = optimal_dows[dow]
        print('following dow can start the erosion process:\n', dow)
        print(f'dow has {len(neighbours)} neighbours that might be considered...')

        tmp = erosion(larp, dow, neighbours, max_UIE,
            excluded_list, discarded_list, optimal_dows,
            P0_list, UE_list, E_list)

        dow_optimum, _, excluded_list, discarded_list, \
            optimal_dows, UE_list, E_list = tmp
        P0_list.append(dow_optimum)

        print('dow optimum:', dow_optimum)
```

Thank you for the attention!

References

- Formulation and solution technique for agricultural waste collection and transport network design, European Journal of Operational Research

- Waste Mismanagement in Developing Countries: A Review of Global Issues, International Journal of Environmental Research and Public Health

- Statistics Report (2019), Ministry of Agriculture and Rural Development of Vietnam

- A Waterflow Algorithm for Optimization Problems (2011), Tran Trung Hieu