Key Strategies of

Team_SB

Samsung SDS & Bielefield Univ.

< Static Problems >

Modification of

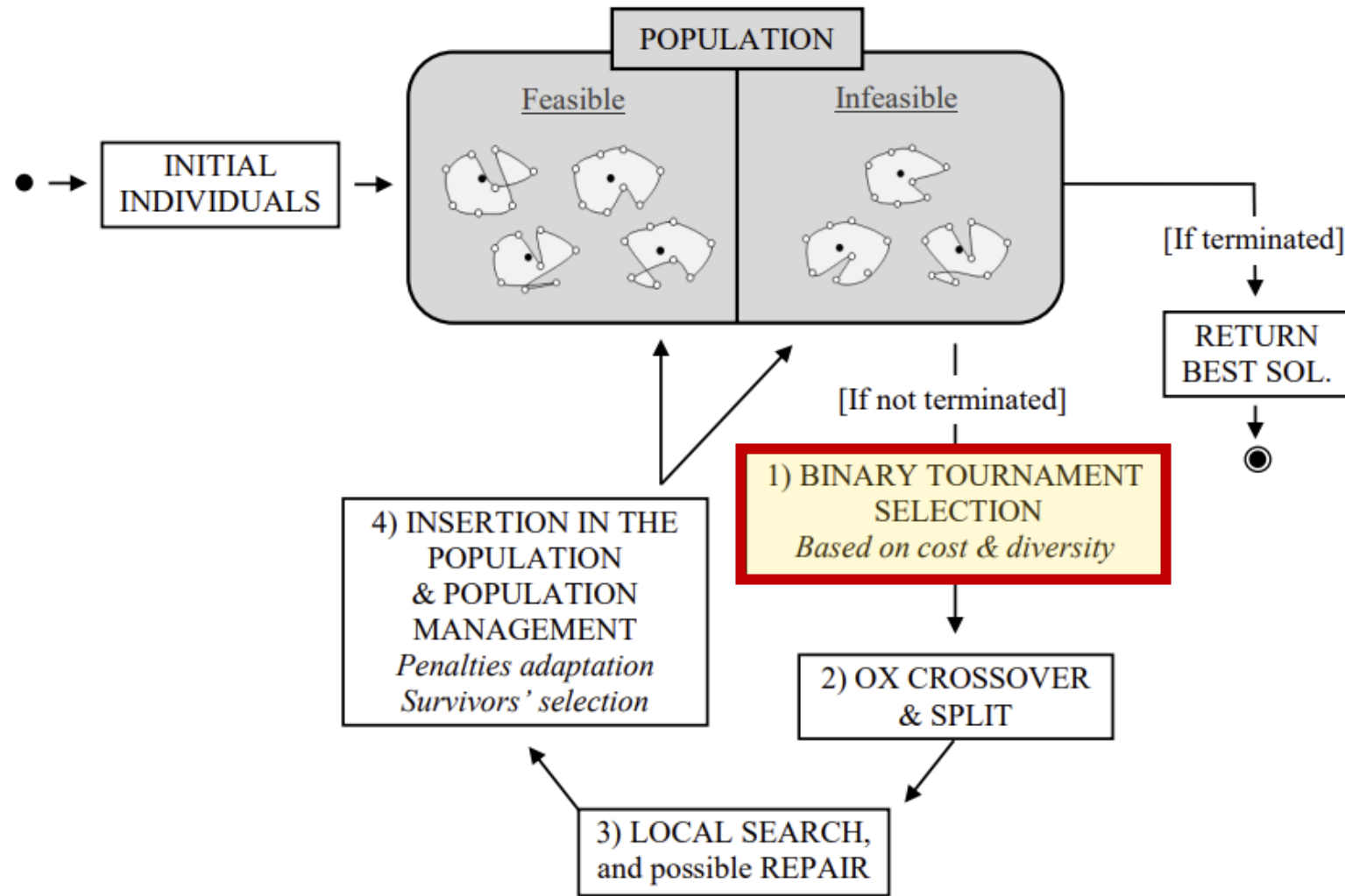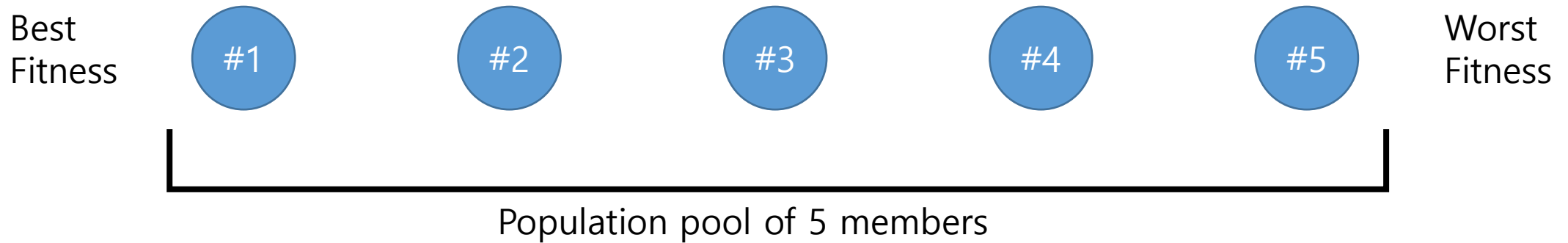**"Binary Tournament Selection" of HGS**

Recap: HGS

Figure 1: General structure of the hybrid genetic search

Hybrid Genetic Search for the CVRP: Open-Source Implementation and SWAP* Neighborhood
Thibaut Vidal

Original "Binary Tournament Selection"

- How to select a parent for Genetic Algorithm (GA)?

Best
Fitness

#1   #2   #3   #4   #5

Worst
Fitness

Population pool of 5 members

Method:
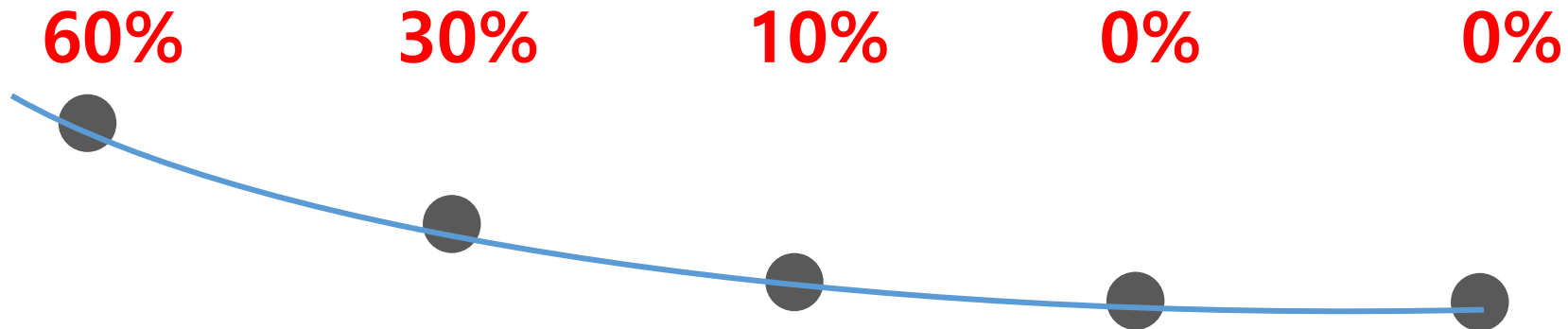
1. Randomly choose 2

2. Of the two, choose the one with better fitness
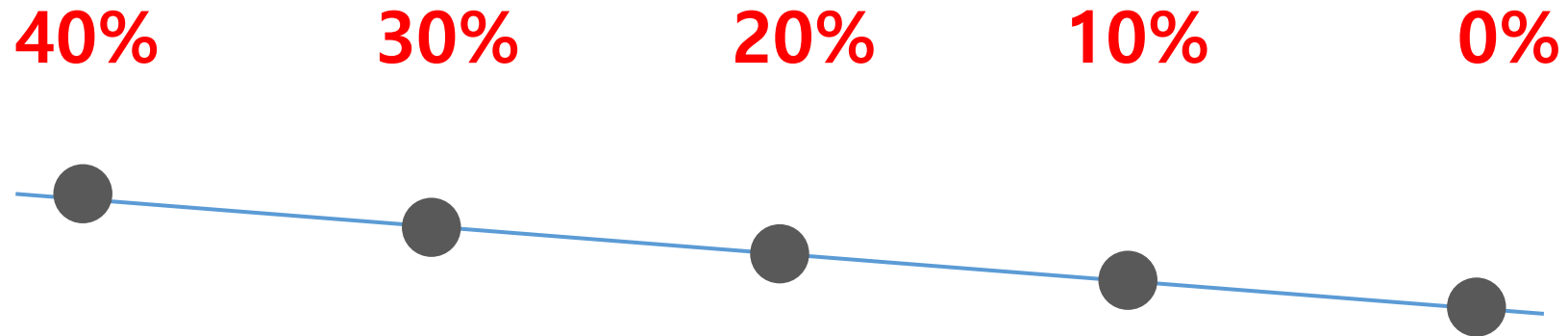
Modification: 3-Way Tournament Selection

1. Randomly choose **3**
2. Of the **three**, choose the one with best fitness

Best Fitness #1 #2 #3 #4 #5 Worst Fitness

Selection Probabilities:

60% **30%** **10%** **0%** **0%**

Modification: 1-Way Tournament Selection

Best Fitness   #1   #2   #3   #4   #5   Worst Fitness

Selection Probabilities:

20%   **20%**   **20%**   **20%**   **20%**

Our Scheme

- How to select a parent for Genetic Algorithm (GA)?

1. Initially, use 4-Way Tournament.

[More Exploitation]

← **"High Rank" candidates are more likely to be chosen at the beginning**

2. When HGS does not produce a new incumbent solution for many iterations,

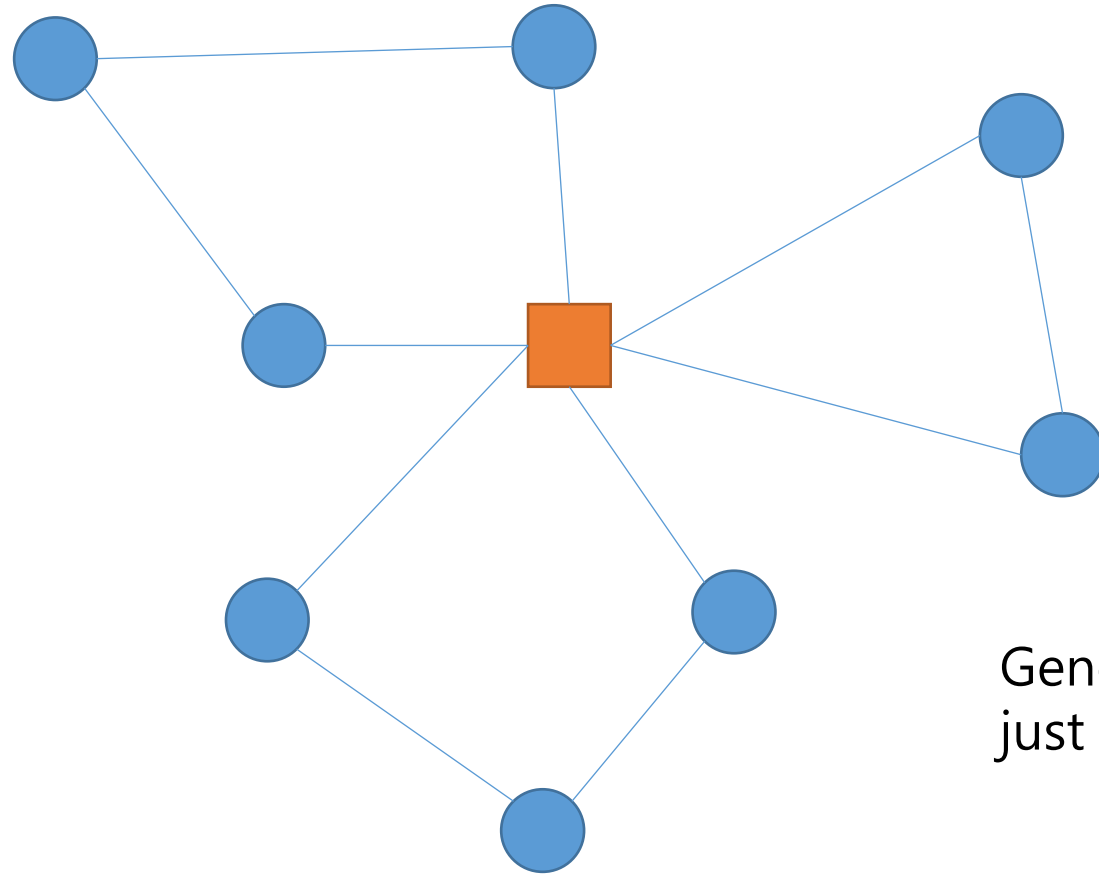reduce it to 3-Way, 2-Way, and then to 1-Way (before RESET).

[More Exploration]

← **"Low Rank" candidates are chosen increasingly more frequently over time**
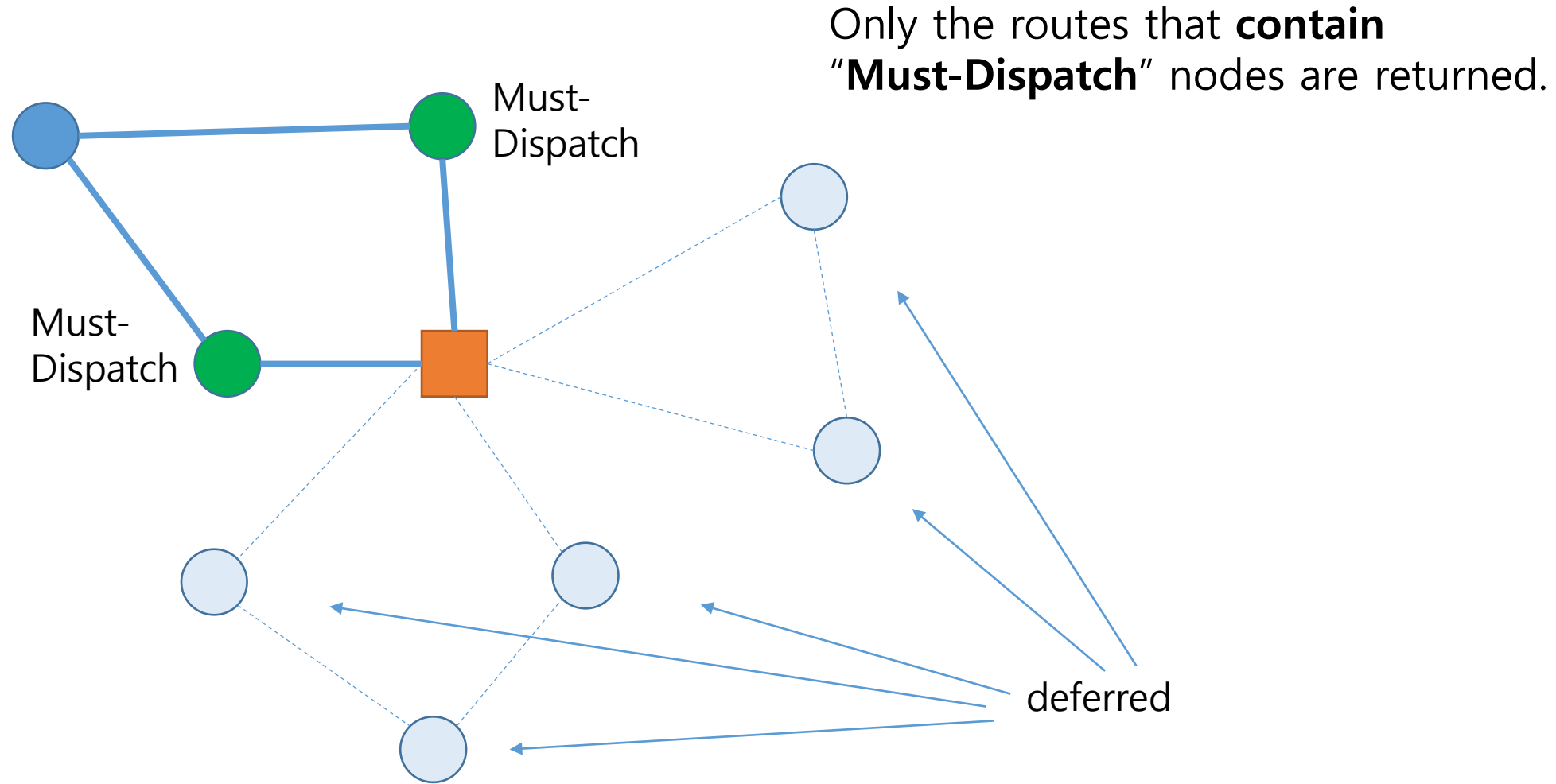
< Dynamic Problems >

**Modified HGS Cost Function
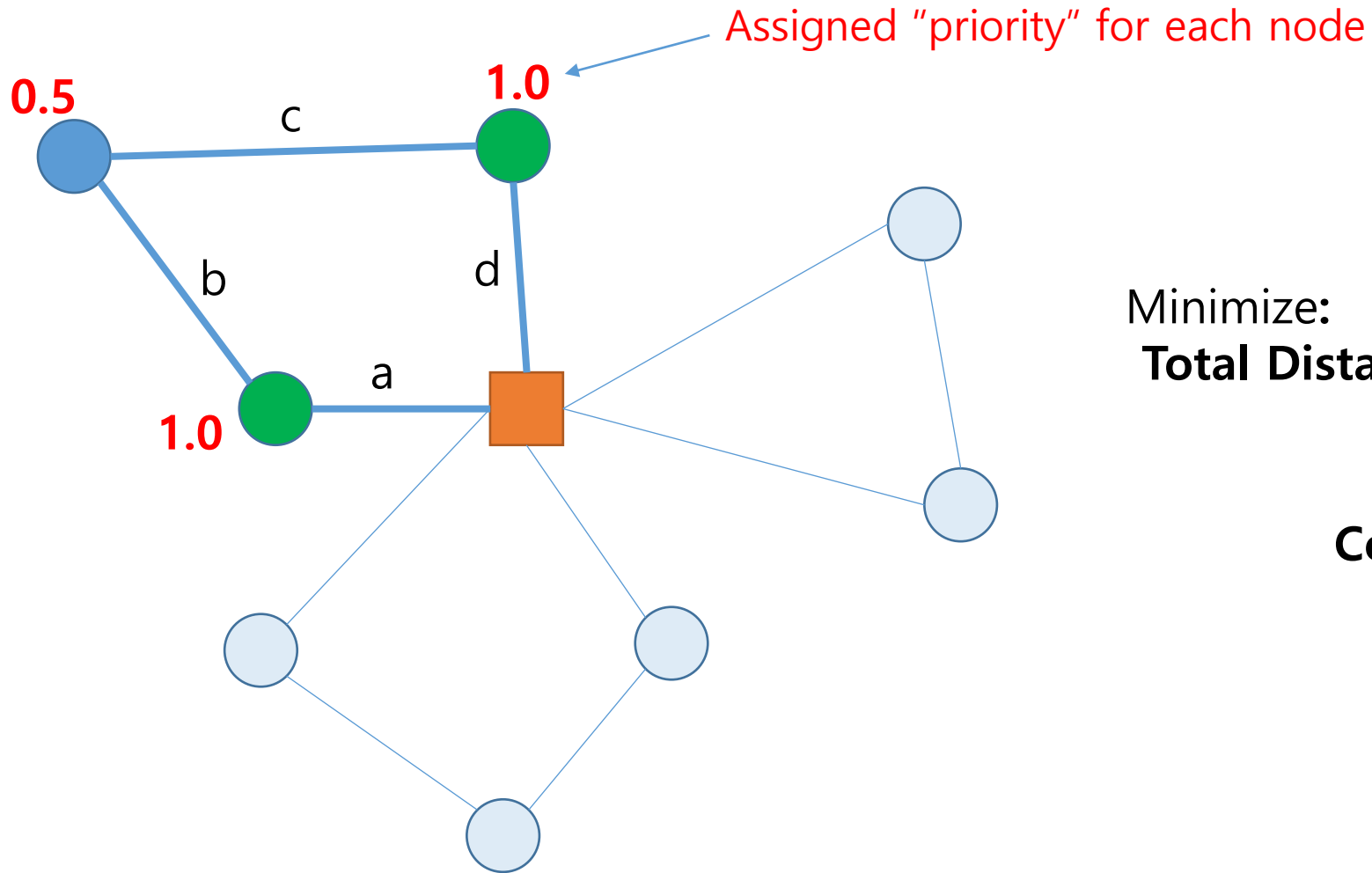+
Lazy Dispatch**

[v0.0] Use HGS to Generate Routes, and Dispatch Lazily.

Generate Routes with **all nodes**, just like in Static Problems

[v0.0] Use HGS to Generate Routes, and Dispatch Lazily.

Only the routes that **contain** "**Must-Dispatch**" nodes are returned.

Must-Dispatch

Must-Dispatch

deferred

[v2.0] Modified HGS Cost
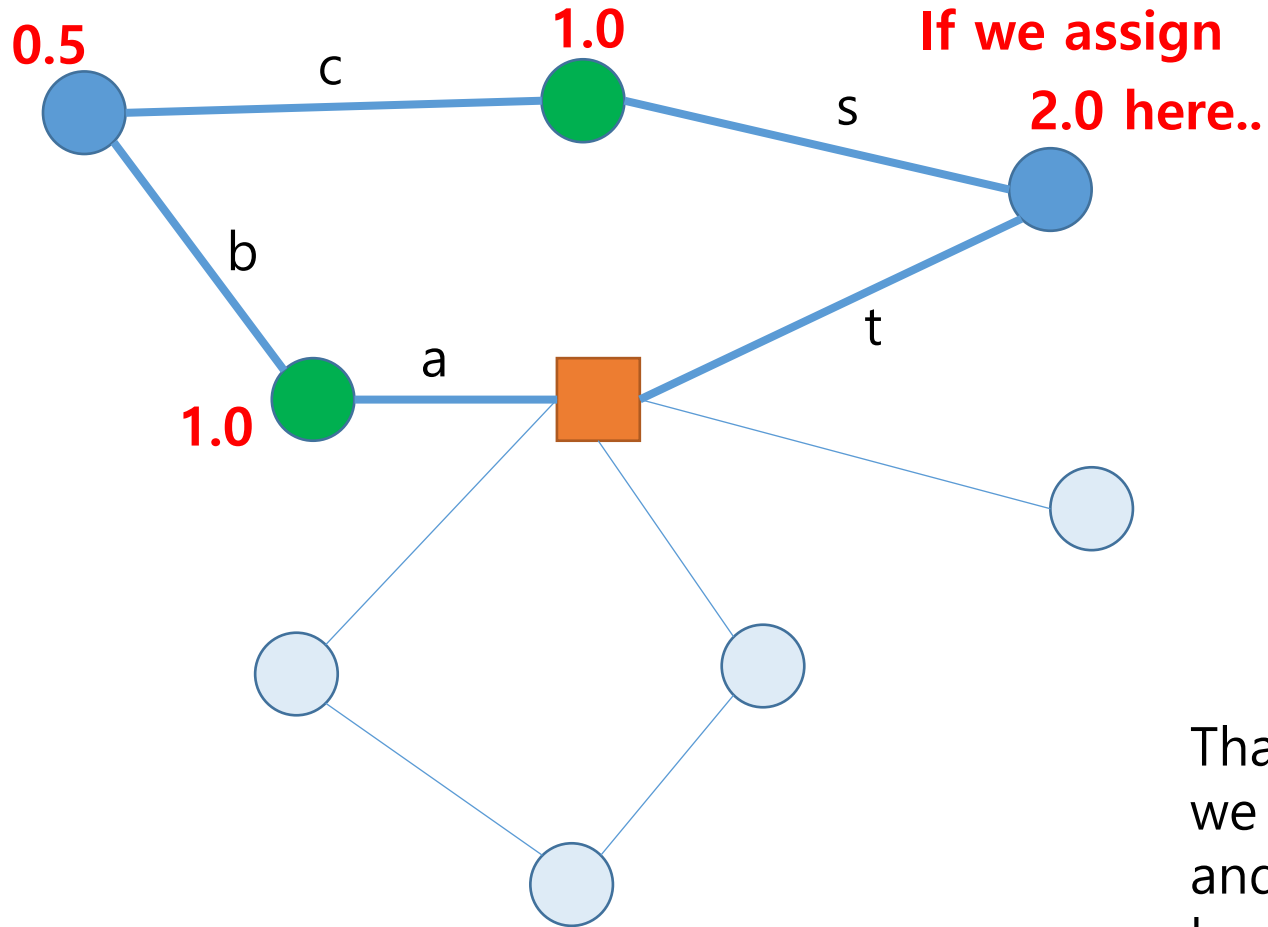

Assigned "priority" for each node

0.5   c   1.0

b   d

1.0   a

Minimize:
Total Distance over Sum of "Priorities"

$$\text{Cost} = \frac{a + b + c + d}{1.0 + 0.5 + 1.0}$$

[v2.0] Modified HGS Cost

0.5

c

1.0

s

**If we assign
2.0 here..**

b

t

a

1.0

Minimized_Cost =

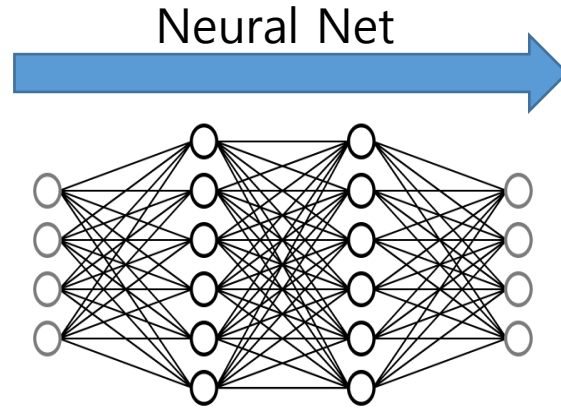$$\frac{a + b + c + s + t}{1.0 + 0.5 + 1.0 + 2.0}$$

That is,
we can **control** which nodes to dispatch,
and which nodes to defer
by assigning customized "**Priorities**"

So what's the good formula for "priorities"?

- We don't know, and we are too lazy.

- We let the AI figure it out by itself.

**Information of
a node and the current epoch**

- when it opens
- when it closes
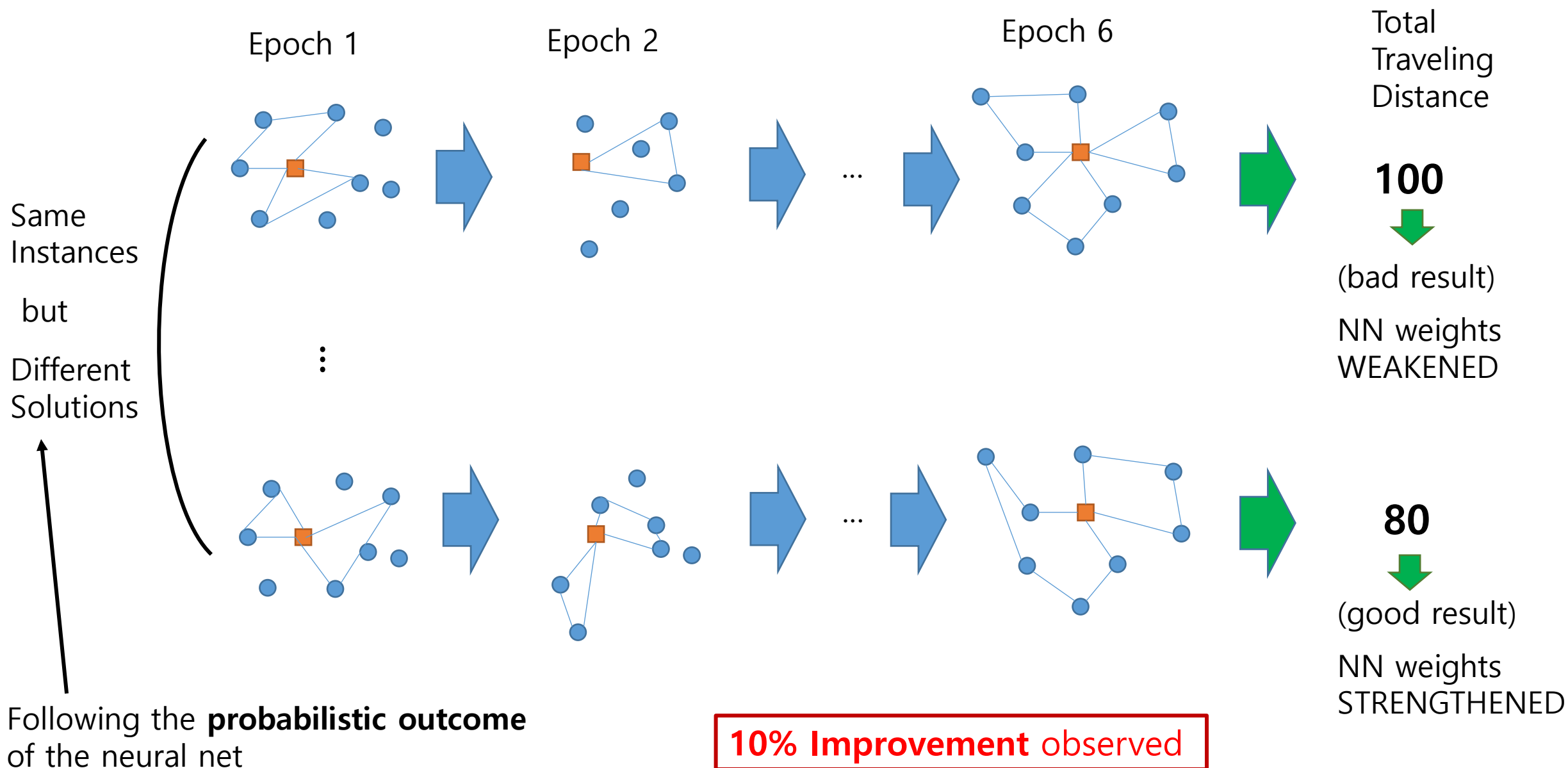- when it becomes "must-dispatch"
- how many epochs left
- ...

Neural Net

Probabilities
for

{ Priority = 0.2
Priority = 0.4
...
Priority = 1.0
... **}**

Select a priority,

according to the
probabilities.

Reinforcement Learning

Buy 4 REINFORCE Samples, Get a Baseline for Free!
Wouter Kool, Herke van Hoof, Max Welling

Epoch 1 Epoch 2 Epoch 6

Total Traveling Distance

Same Instances but Different Solutions

100

(bad result)

NN weights WEAKENED

80

(good result)

NN weights STRENGTHENED

Following the **probabilistic outcome** of the neural net

10% Improvement observed

Thank You
For Listening!