Machine Learning and Generative AI for Large-Scale Traveling Salesman Problems

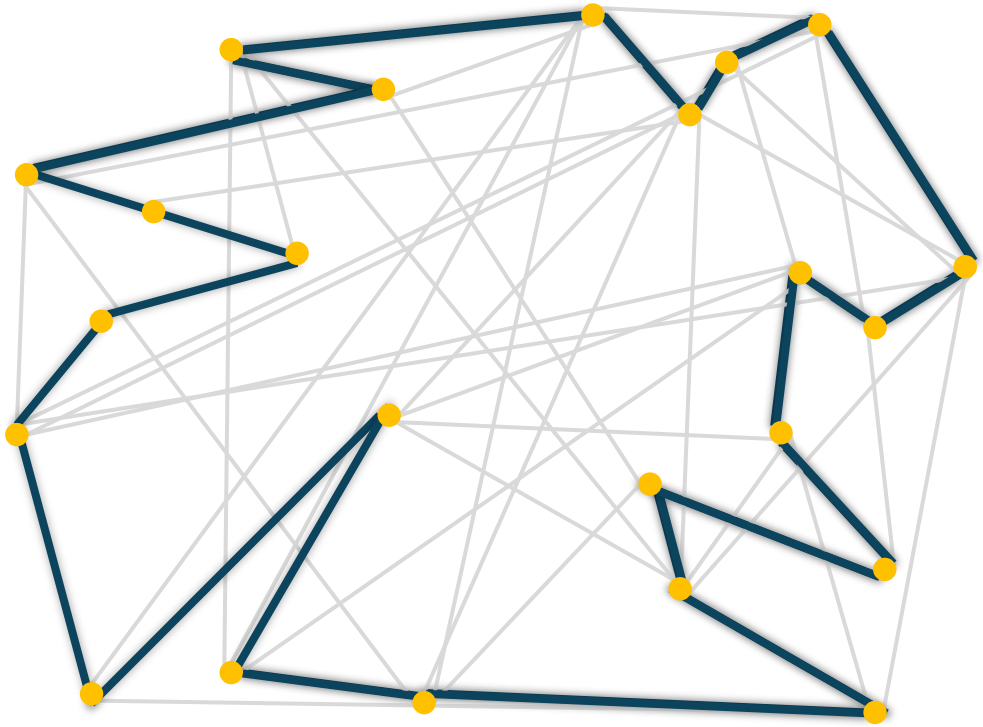
Bocheng Lin
Supervisors: Yi Mei, Mengjie Zhang
Victoria University of Welington

2024.12

1. Introduction
2. Motivations
3. Research Objectives
4. Preliminary Works
5. Proposed Contributions and Research Plan

1. Introduction
2. Motivations
3. Research Objectives
4. Preliminary Works
5. Proposed Contributions and Research Plan

Traveling Salesman Problem



a complete graph $G = (V, E)$

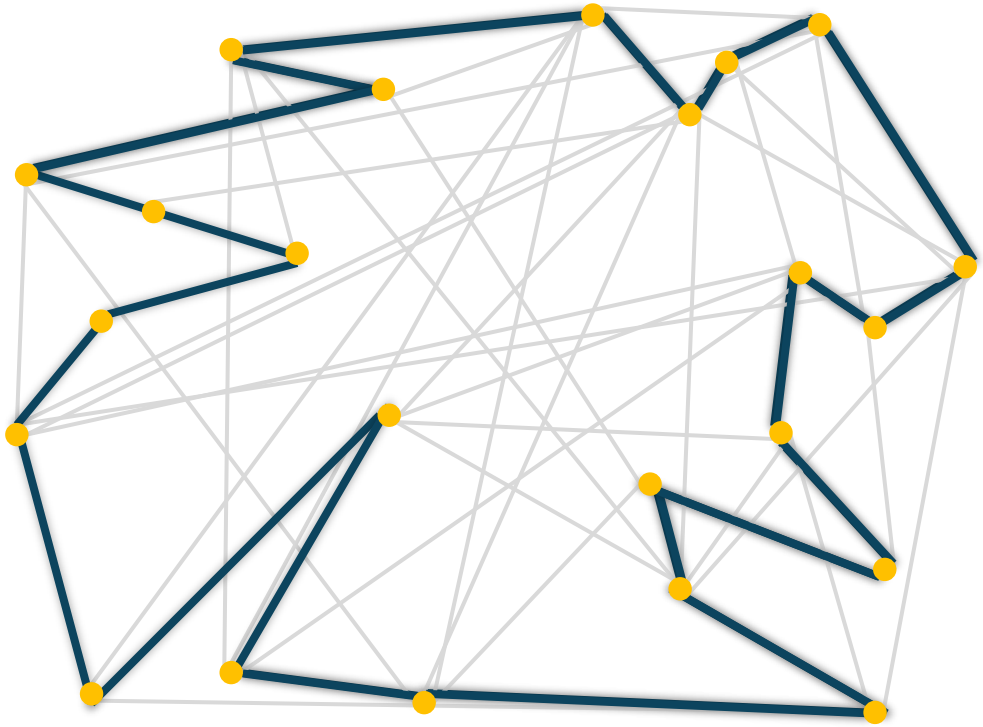
vertices V representing cities

edges E representing the paths between cities

$w(i, j)$ reflects the cost or distance between cities i and j

The objective is to find the shortest possible route that visits each city exactly once and returns to the starting point

Traveling Salesman Problem



NP-Hard

Large-Scale Traveling Salesman Problem

Exact Methods	find provably optimal solutions through exhaustive or branch-and-bound searches, albeit often with prohibitive computational costs. Algorithms: MILP, Constraint Programming, cut-and-price, Concorde, etc.
Heuristic Methods and (Hyper-Heuristics) Evolutionary Machine Learning	produce near-optimal solutions efficiently by exploiting problem-specific insights, though without guaranteeing global optimality Algorithms: Greedy Search, A*, Tabu Search, LNS, LKH-3, ACO, GPHH, GP-ACO, etc.
Neural Combinatorial Optimization (NCO)	leverages neural networks, often paired with reinforcement learning, to learn and solution patterns from data Algorithms: Ptr-Net, AM, POMO, GCN, GNN, Equity-Transformer, Sym-NCO, etc.
Generative AI (Diffusion Models)	data-driven generative models to directly produce high-quality solutions, integrating global structural information for more holistic route designs Algorithms: DIFUSCO, IDEQ, TSPDiffuser, DISCO, Fast t2t, SymmetricDiffusers, etc.

Large-Scale Traveling Salesman Problem

Exact Methods	**In theory, optimal solution**	**Unacceptable Computation Time**
Heuristic Methods and (Hyper-Heuristics) Evolutionary Machine Learning	High quality solution in reasonable time Good Generalization ability	Hard to capture global information Trade off between quality and efficiency Low speed (improvement heuristics)
Neural Combinatorial Optimization (NCO)	High potential to capture global information with GNNs High speed Inference with GPU	Low Generalization ability High Computation resources
Generative AI (Diffusion Models)	**High quality solutions** High speed Inference with GPU	Huge Computation resources Low Generalization ability Huge data

In the context of the Traveling Salesman Problem, “global information” encompasses the holistic structural and relational data regarding all cities and their interconnecting distances, enabling comprehensive evaluation and optimization beyond solely local considerations.

Machine Learning
is to automatically find a function from data.

Set scope

Define the set of candidate functions

Deep Learning (CNN, Transformer…), Decision Tree, etc.

Establish criteria

Define the criteria for evaluating the quality of functions

Supervised Learning, Semi-supervised Learning, Reinforcement Learning, etc.

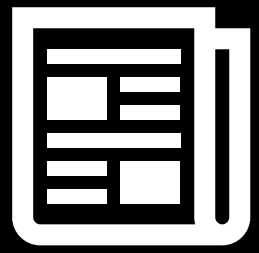
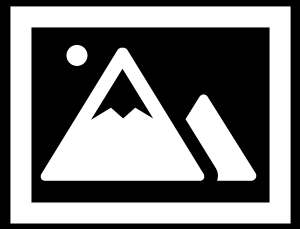
Achieve the goal

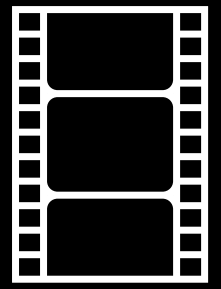
Find the best function → Optimization

Gradient Descent, Genetic Algorithm, etc.

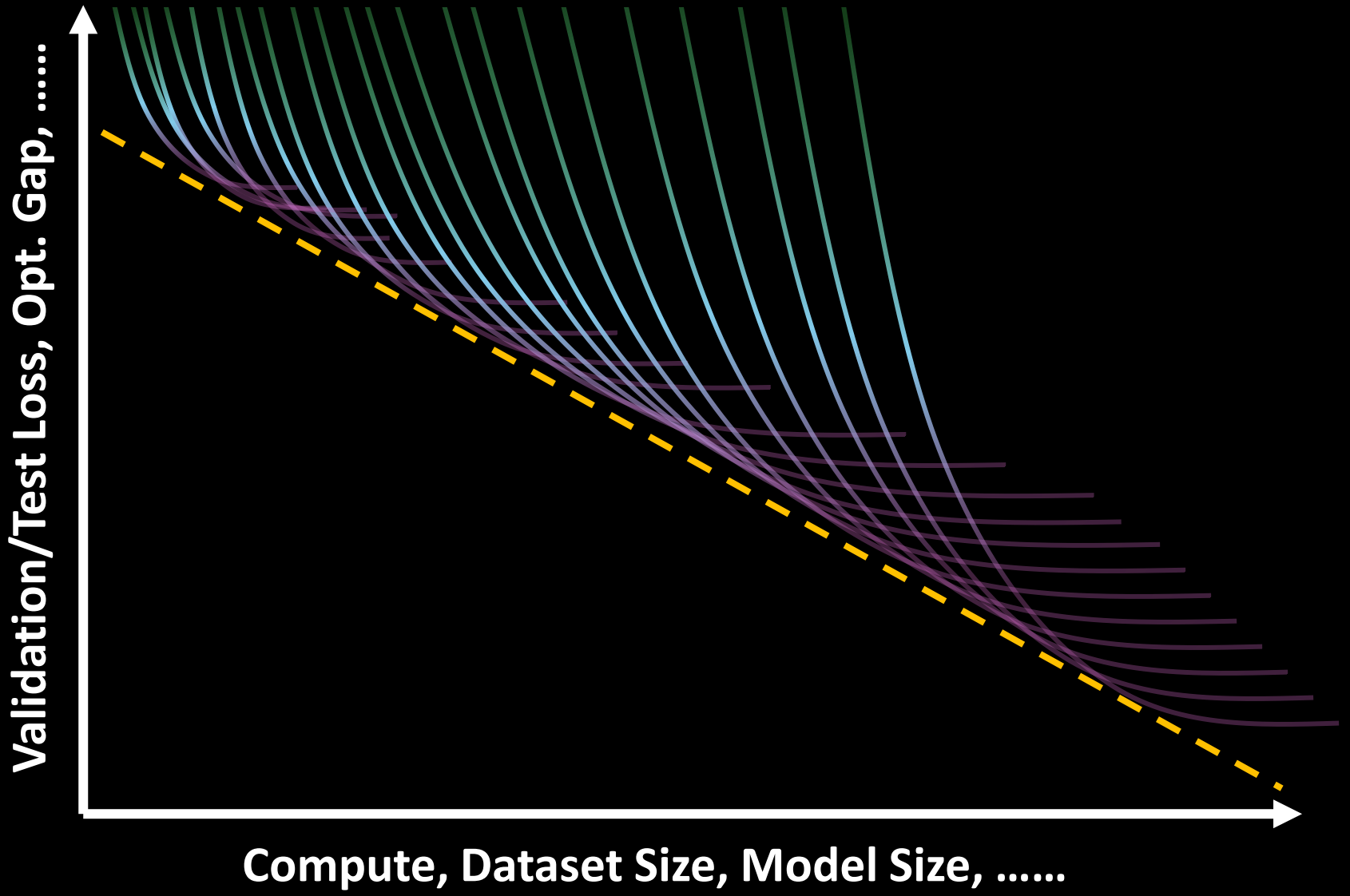
Generative AI

is to enable machines to generate complex and structured things.

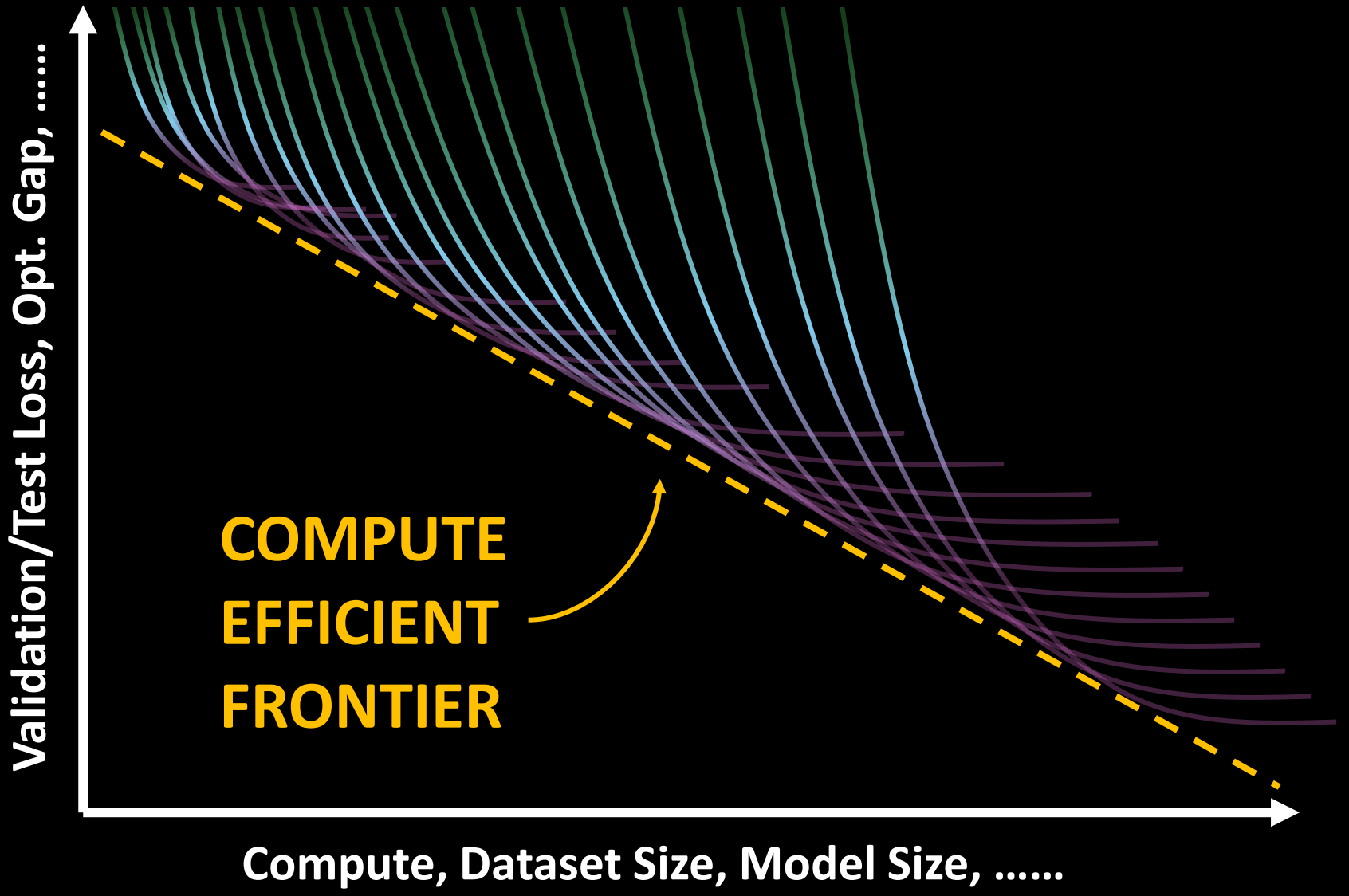





Scaling Laws



Scaling Laws



Large-Scale Traveling Salesman Problem

Exact Methods	**In theory, optimal solution**	**Unacceptable Computation Time**
Heuristic Methods and (Hyper-Heuristics) Evolutionary Machine Learning	High quality solution in reasonable time Good Generalization ability	Hard to capture global information Trade off between quality and efficiency Low speed (improvement heuristics)
Neural Combinatorial Optimization (NCO)	High potential to capture global information with GNNs High speed Inference with GPU	Low Generalization ability High Computation resources
Generative AI (Diffusion Models)	**High quality solutions** High speed Inference with GPU	Huge Computation resources Low Generalization ability Huge data

In the context of the Traveling Salesman Problem, “global information” encompasses the holistic structural and relational data regarding all cities and their interconnecting distances, enabling comprehensive evaluation and optimization beyond solely local considerations.

1. Introduction
2. Motivations
3. Research Objectives
4. Preliminary Works
5. Proposed Contributions and Research Plan

Current Limitations

1 Insufficient Understanding of Scaling Laws

2 NCOs' Generalization Ability and Heuristics' Global Information Processing

3 Insufficient Adaptability and Customization in Generative AI and Heuristic Methods

4 Insufficient Research on Transfer Learning

Current Limitations

1 Insufficient Understanding of Scaling Laws

With the problem size increases, Heuristics and NCOs need more resources
Computing (Evaluation Times), Training data, Model size …

But, how much?

Compute Efficient Frontier?

How to allocation?

Current Limitations

2 NCOs' Generalization Ability and Heuristics' Global Information

Exact Methods	**In theory, optimal solution**	**Unacceptable Computation Time**
Heuristic Methods and (Hyper-Heuristics) Evolutionary Machine Learning	High quality solution in reasonable time Good Generalization ability	Hard to capture global information Trade off between quality and efficiency Low speed (improvement heuristics)
Neural Combinatorial Optimization (NCO)	High potential to capture global information with GNNs High speed Inference with GPU	Low Generalization ability High Computation resources
Generative AI (Diffusion Models)	**High quality solutions** High speed Inference with GPU	Huge Computation resources Low Generalization ability Huge data

In the context of the Traveling Salesman Problem, "global information" encompasses the holistic structural and relational data regarding all cities and their interconnecting distances, enabling comprehensive evaluation and optimization beyond solely local considerations.

3 Insufficient Customization in Generative AI and Heuristics

Generative AI (Diffusion Models) ***end-to-end***

cannot adjust model according to specific instance

high computation cost of inference

Heuristics ***fixed strategy***

cannot adjust strategy according to specific instance

low computation cost (constructive heuristics)

4 Insufficient Research on Transfer Learning

When facing new variant of TSPs

NCOs

Re-collect training data, re-train

Heuristics

Re-design heuristics manually

1. Introduction
2. Motivations
3. Research Objectives
4. Preliminary Works
5. Proposed Contributions and Research Plan

Overall Goal

to develop scalable and efficient solutions for large-scale TSP by leveraging machine learning and generative AI techniques

Objective 1

Exploring Scaling Laws in Solving Large-Scale TSP

Current Limitations

1 Insufficient Understanding of Scaling Laws

With the problem size increases, Heuristics and NCOs need more resources
Computing (Evaluation Times), Training data, Model size …

But, how much? **Prediction**

Compute Efficient Frontier?

How to allocation?

Objective 1

Scaling Laws

1.1 Analysis the Scaling Behavior of Heuristic Methods

1.2 Analysis the Scaling Behavior of NCO Methods

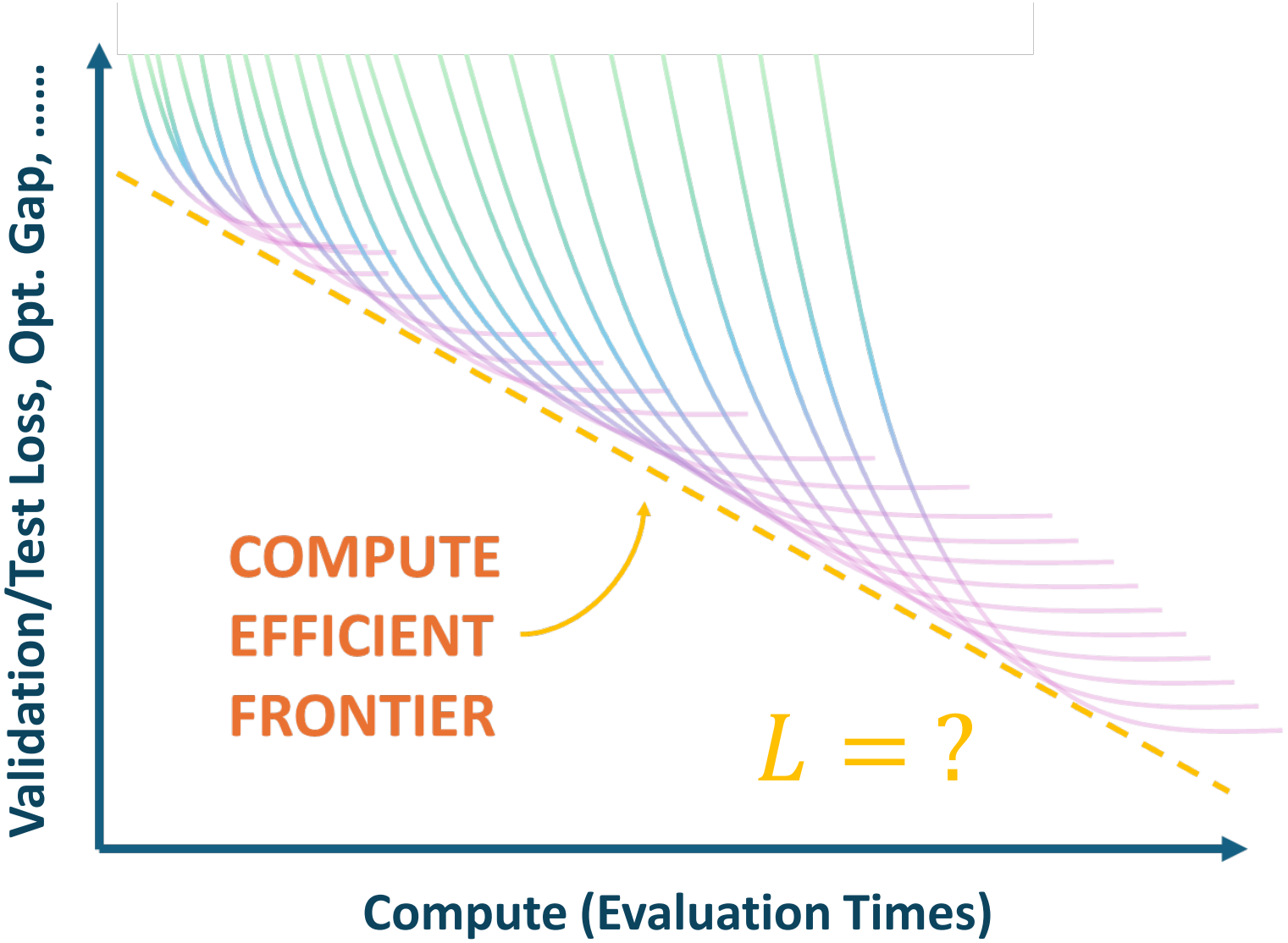
1.3 Formulate Scaling Laws to Guide Resource Allocation

Optional

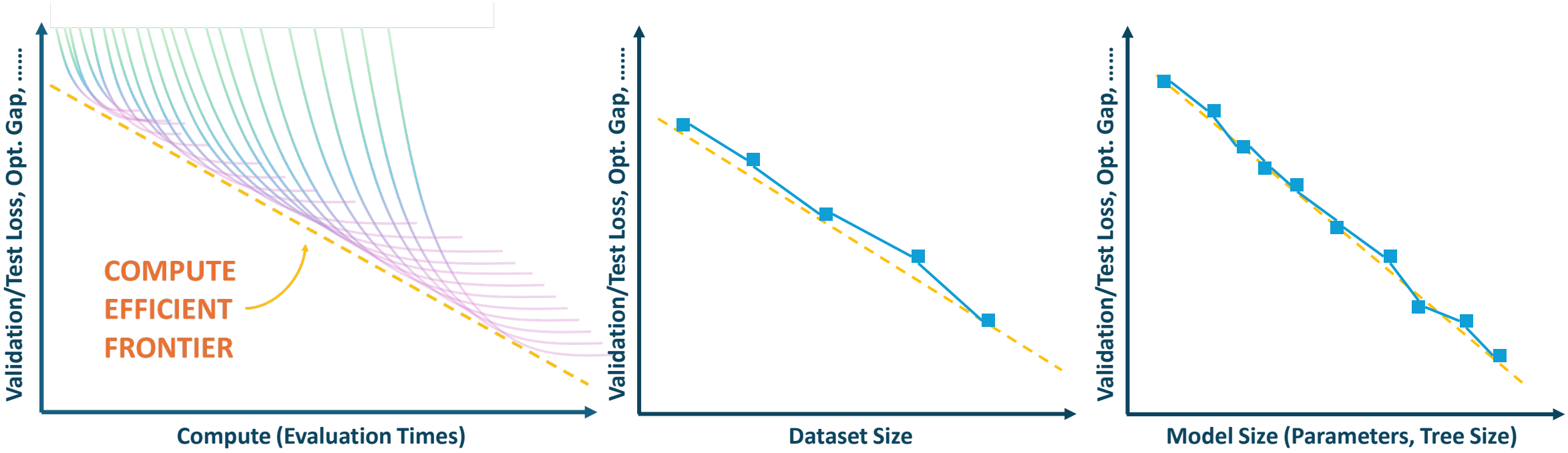
1.4 Develop Theoretical Models to Explain Observed Scaling Behaviors

Objective 1

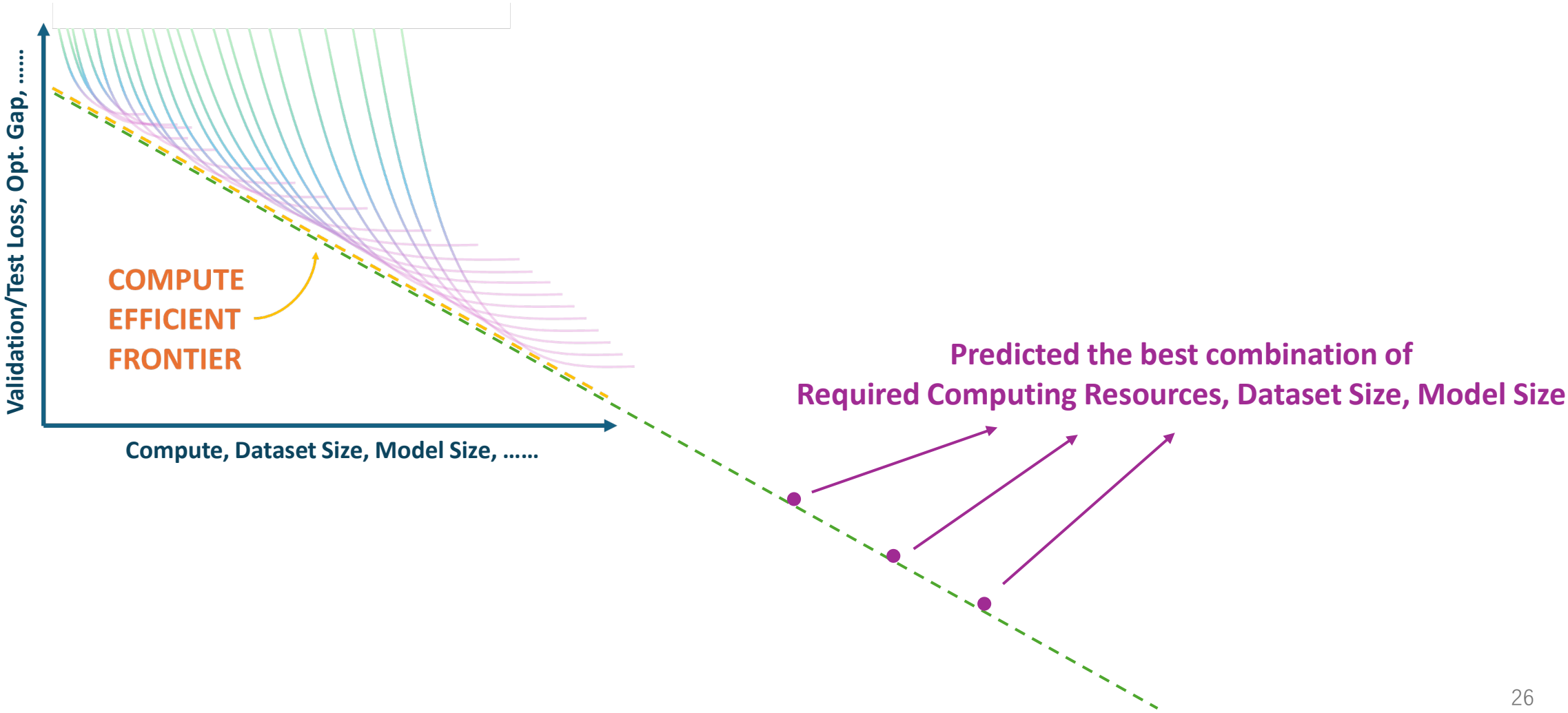
1.1 & 1.2 Analysis the Scaling Behavior of Heuristic & NCO Methods



Objective 1
1.1 & 1.2 Analysis the Scaling Behavior of Heuristic & NCO Methods

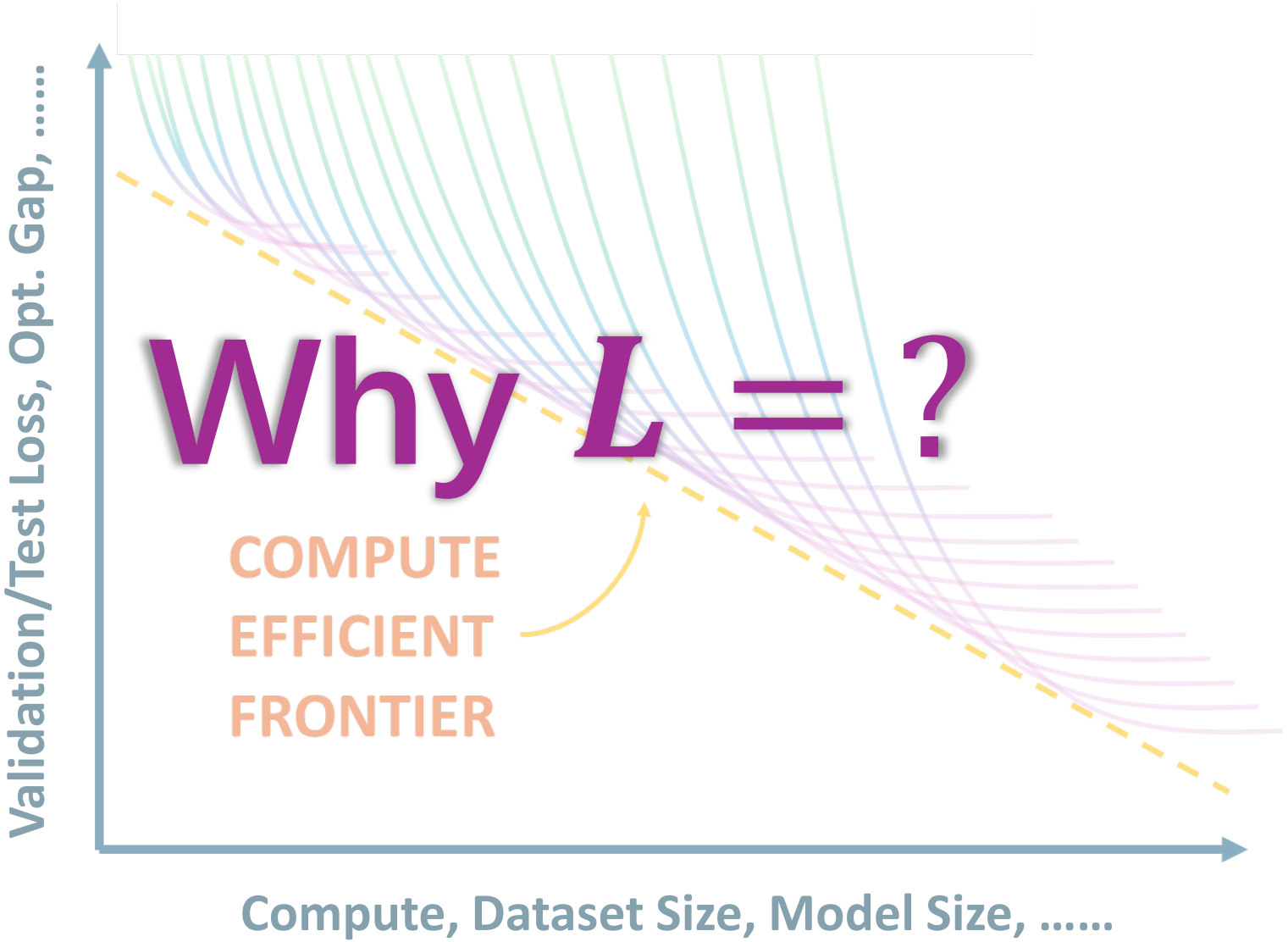


Objective 1
1.3 Formulate Scaling Laws to Guide Resource Allocation



Objective 1

1.4 Develop Theoretical Models to Explain Observed Scaling Behaviors



Objective 2

Integrating NCO with Hyper-Heuristic Approaches

Current Limitations

2 NCOs' Generalization Ability and Heuristics' Global Information

Heuristic Methods and (Hyper-Heuristics) Evolutionary Machine Learning	High quality solution in reasonable time Good Generalization ability	Hard to capture global information Trade off between quality and efficiency Low speed (improvement heuristics)
Neural Combinatorial Optimization (NCO)	High potential to capture global information with GNNs High speed Inference with GPU	Low Generalization ability High Computation resources

In the context of the Traveling Salesman Problem, "global information" encompasses the holistic structural and relational data regarding all cities and their interconnecting distances, enabling comprehensive evaluation and optimization beyond solely local considerations.

Current Limitations

2 NCOs' Generalization Ability and Heuristics' Global Information

Heuristic Methods
and (Hyper-Heuristics)
Evolutionary Machine Learning

High quality solution in reasonable time
Good Generalization ability

Hard to capture global information
Trade off between quality and efficiency
Low speed (improvement heuristics)

Neural Combinatorial Optimization (NCO)

High potential to capture global information with GNNs
High speed Inference with GPU

Low Generalization ability
High Computation resources

In the context of the Traveling Salesman Problem, "global information" encompasses the holistic structural and relational data regarding all cities and their interconnecting distances, enabling comprehensive evaluation and optimization beyond solely local considerations.

Current Limitations

2 NCOs' Generalization Ability and Heuristics' Global Information

Heuristic Methods
and (Hyper-Heuristics)
Evolutionary Machine Learning

High quality solution in reasonable time
Good Generalization ability

Hard to capture global information
Trade off between quality and efficiency
Low speed (improvement heuristics)

Neural Combinatorial Optimization (NCO)

High potential to capture global information with GNNs
High speed Inference with GPU

Low Generalization ability
High Computation resources

In the context of the Traveling Salesman Problem, "global information" encompasses the holistic structural and relational data regarding all cities and their interconnecting distances, enabling comprehensive evaluation and optimization beyond solely local considerations.

Objective 2

NCO & Hyper-Heuristics

2.1 Enhance the Function Set of GP

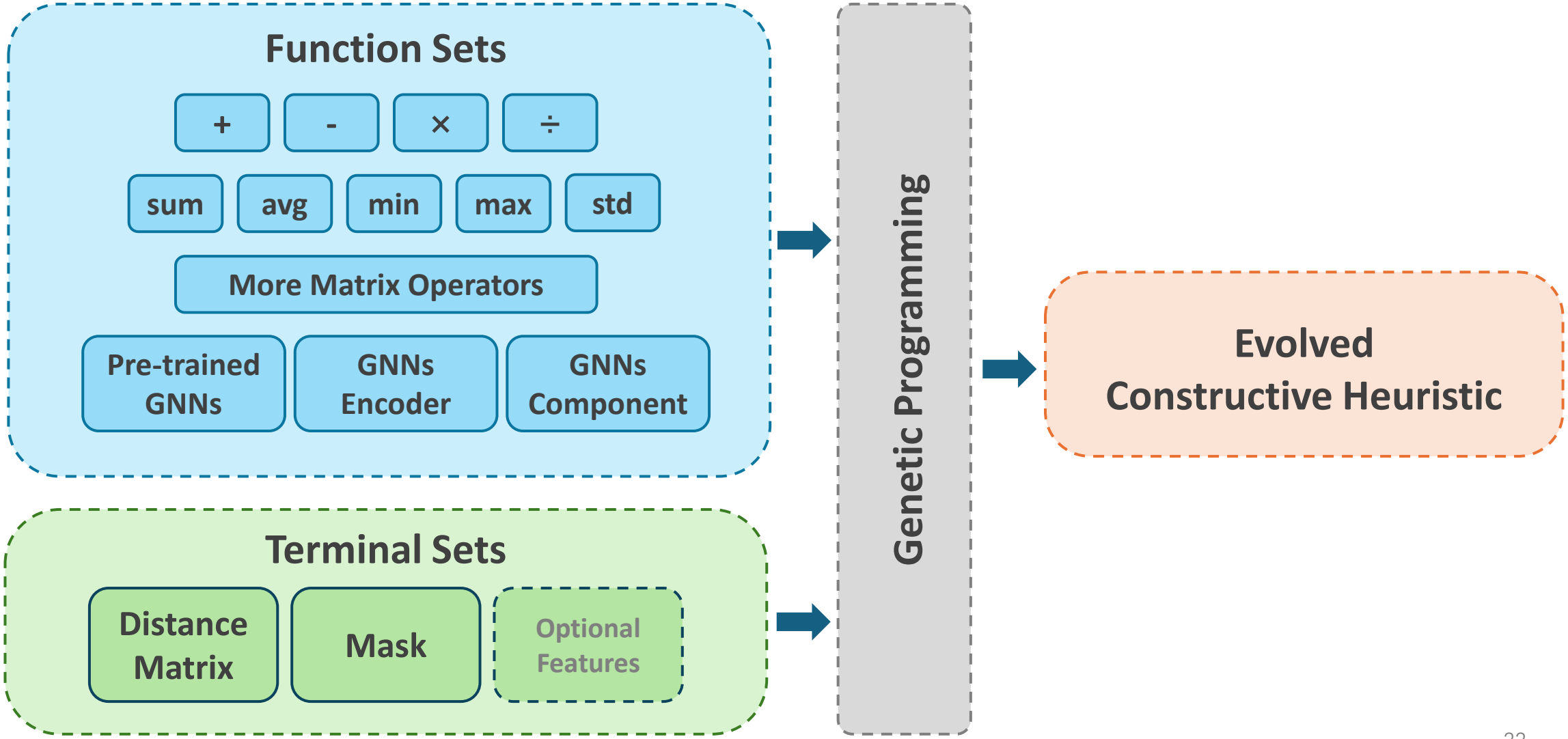
2.2 Optimize Sampling Strategy in Heuristic Search

2.3 Leveraging Historical Search Data to Improve Performance

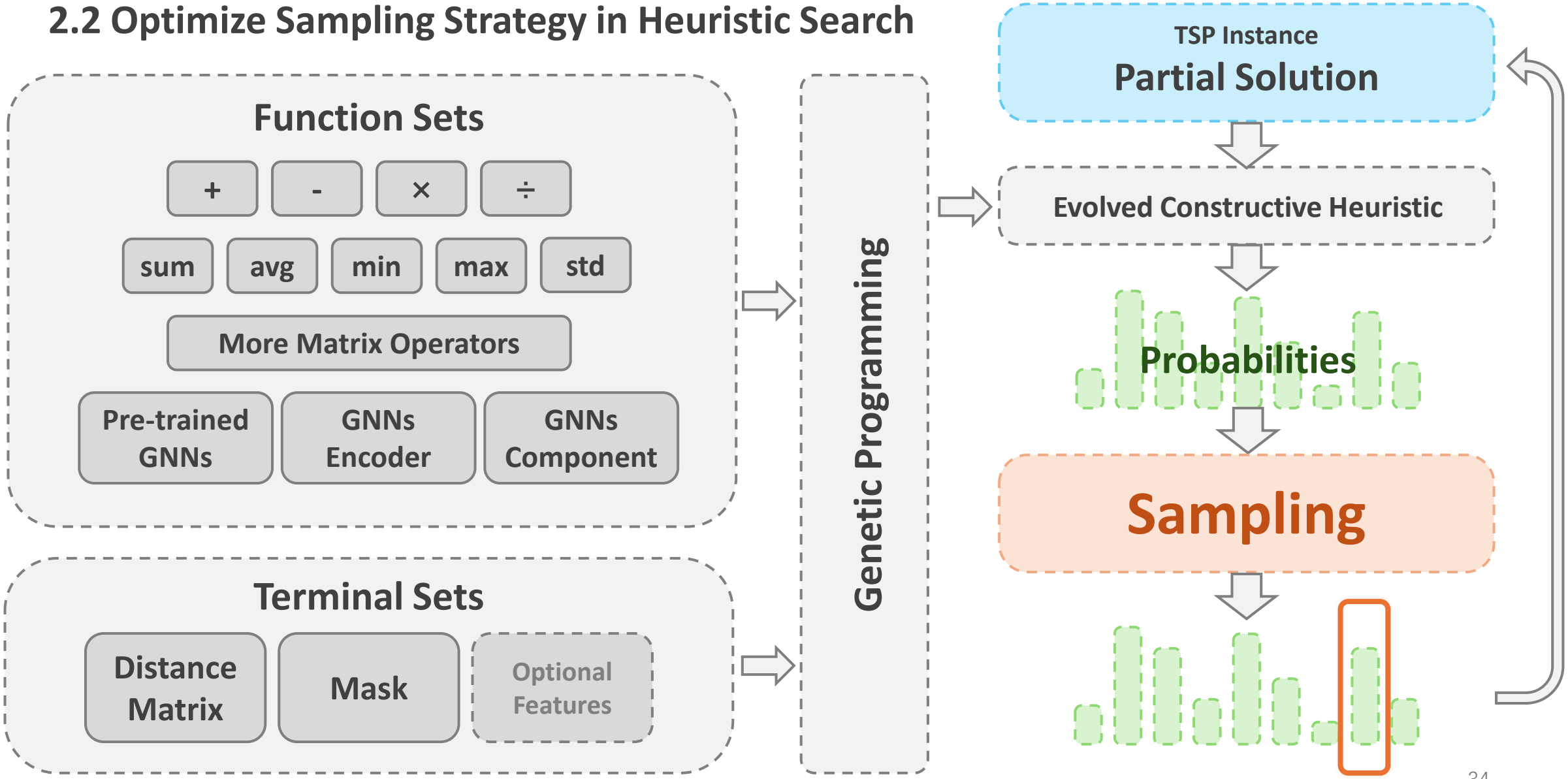
for Objective 3

2.4 Prepare Structure Data to Support Generative AI Integration

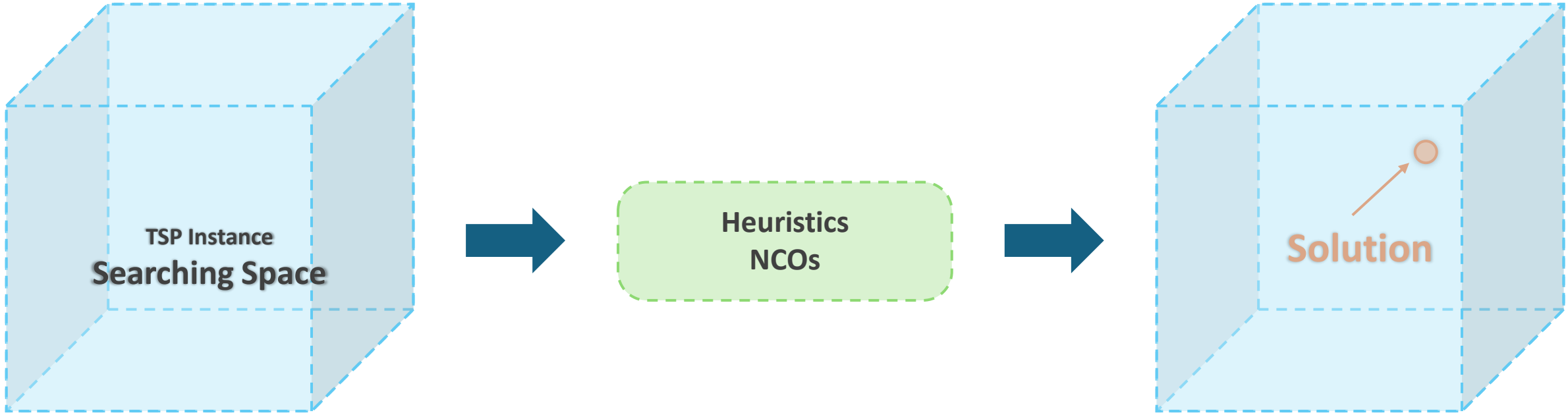
Objective 2
2.1 Enhance the Function Set of GP



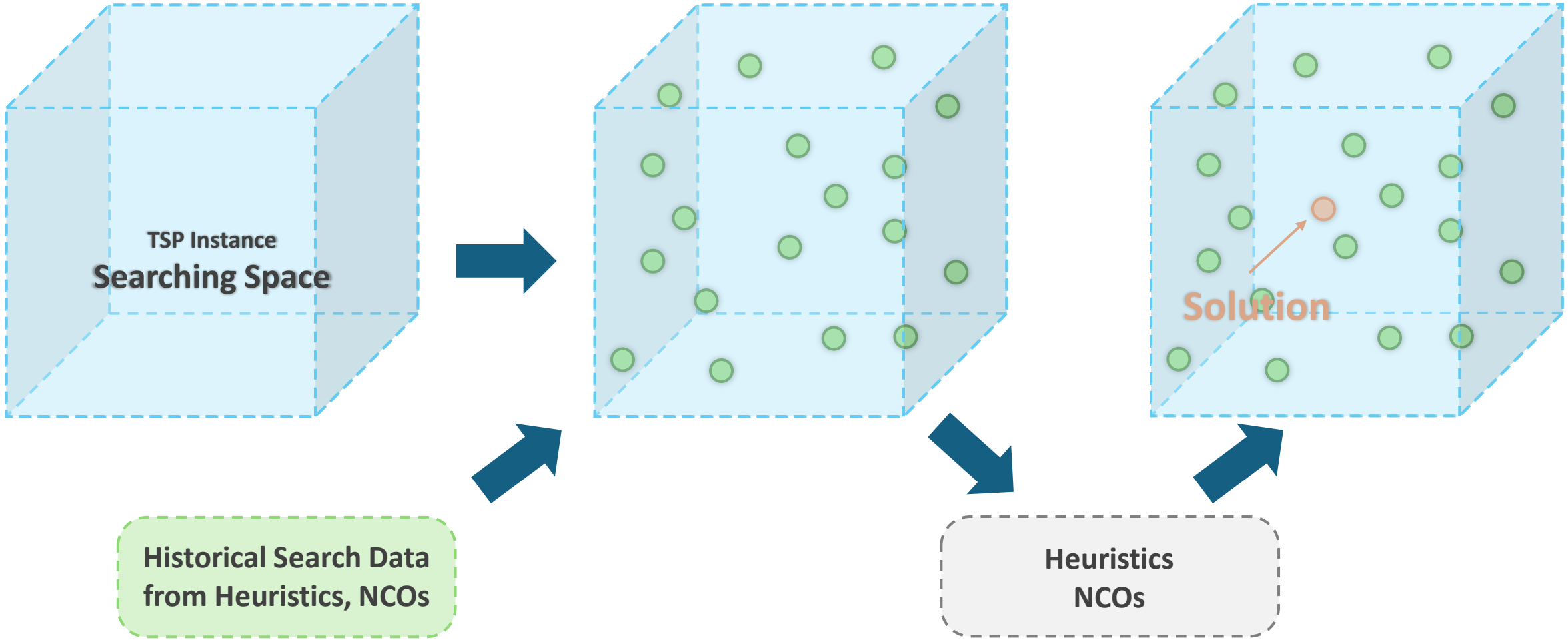
Objective 2
2.2 Optimize Sampling Strategy in Heuristic Search



Objective 2
2.3 Leveraging Historical Search Data to Improve Performance

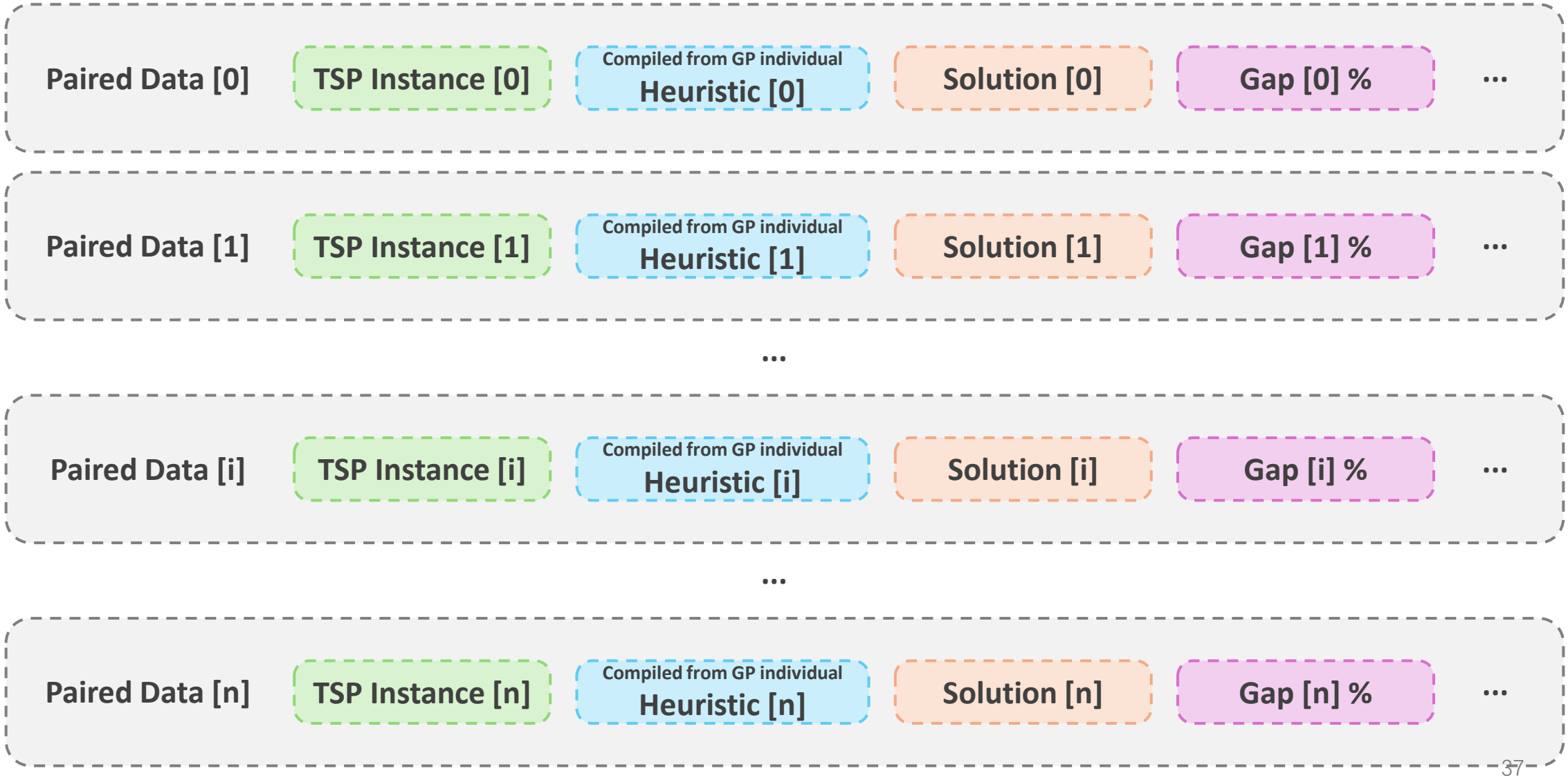


Objective 2
2.3 Leveraging Historical Search Data to Improve Performance



Objective 2

2.4 Prepare Structure Data to Support Generative AI Integration



Objective 3

Develop Generative AI Models for Heuristic Generation

Current Limitations

3 Insufficient Customization in Generative AI and Heuristics

Generative AI (Diffusion Models) *end-to-end*

cannot adjust model according to specific instance

high computation cost and inference

Generate Heuristics

Heuristics *fixed strategy*

cannot adjust strategy according to specific instance

low computation cost (constructive heuristics)

Objective 3

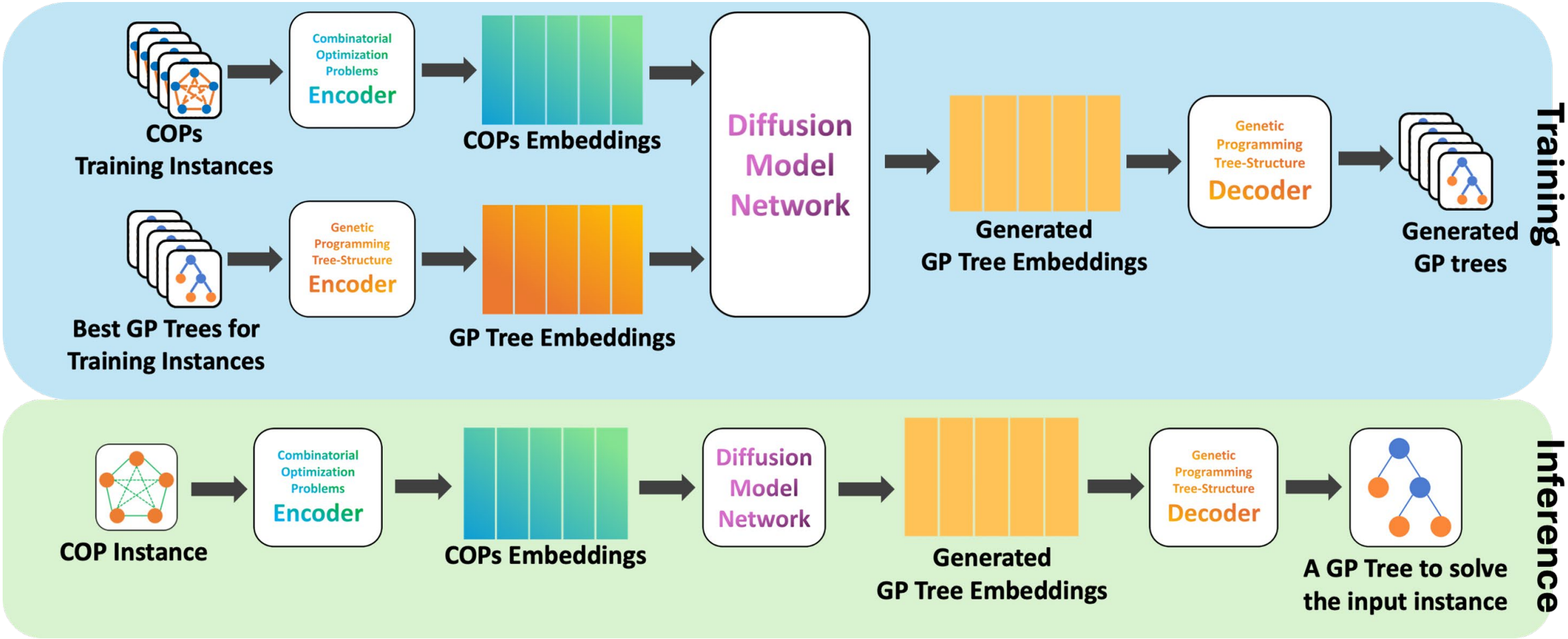
Generative AI for Heuristic Generation

3.1 Diffusion Models for Instance-level Heuristic Generation

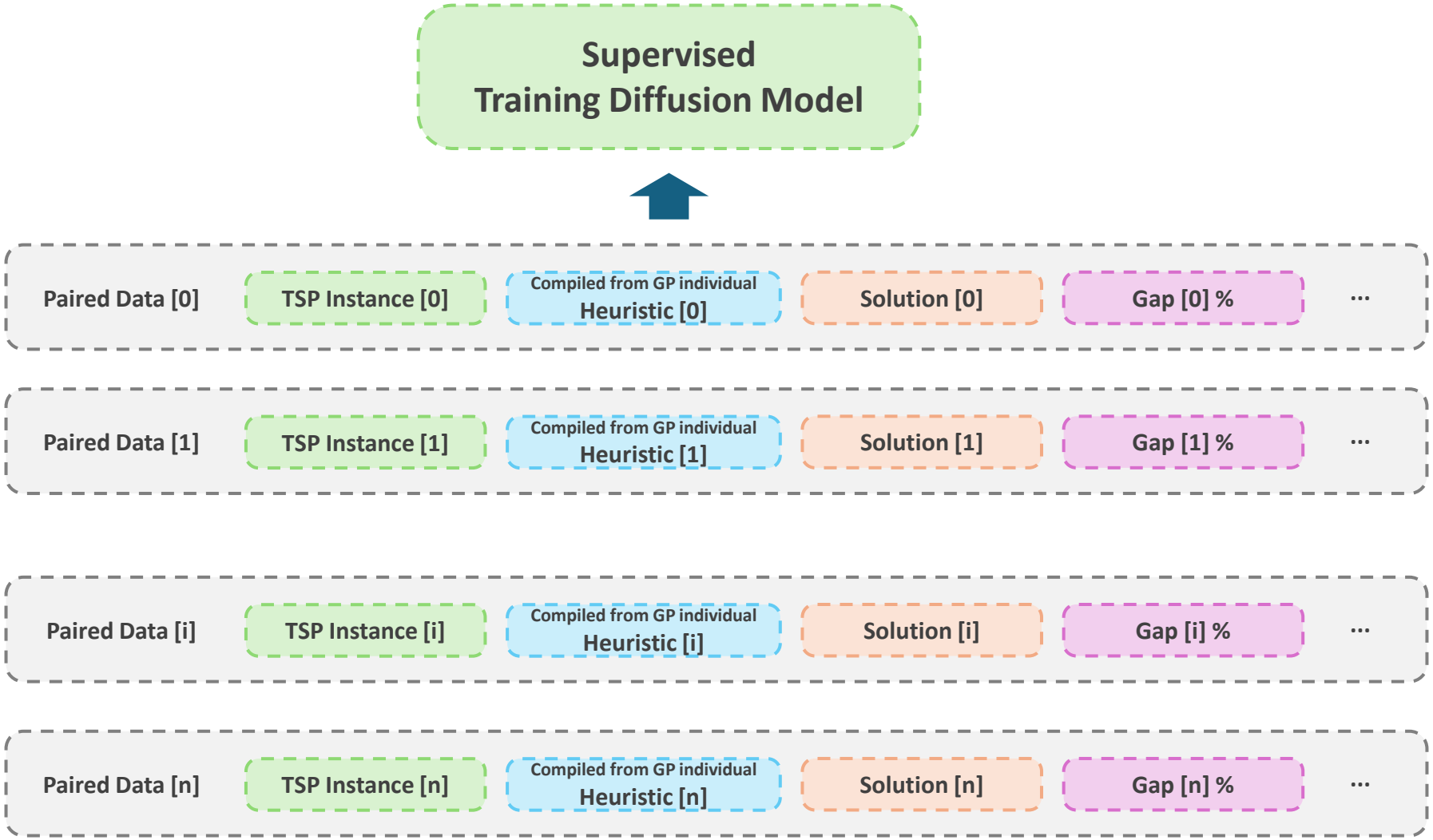
3.2 Explore Self-Supervised Training Methods

3.3 Analysis Limitations and Explore Improvement Methods

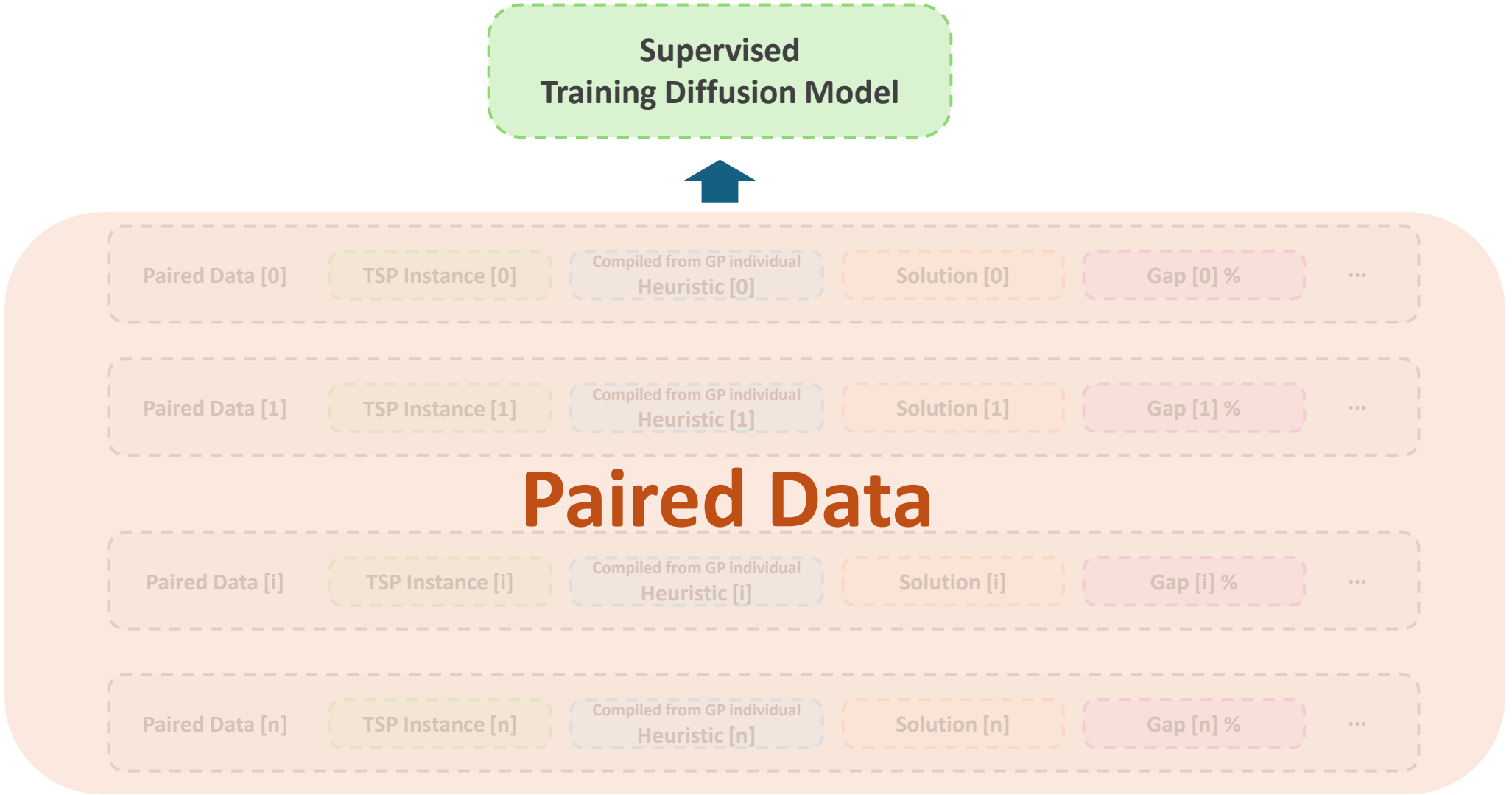
Objective 3
3.1 Diffusion Models for Instance-level Heuristic Generation



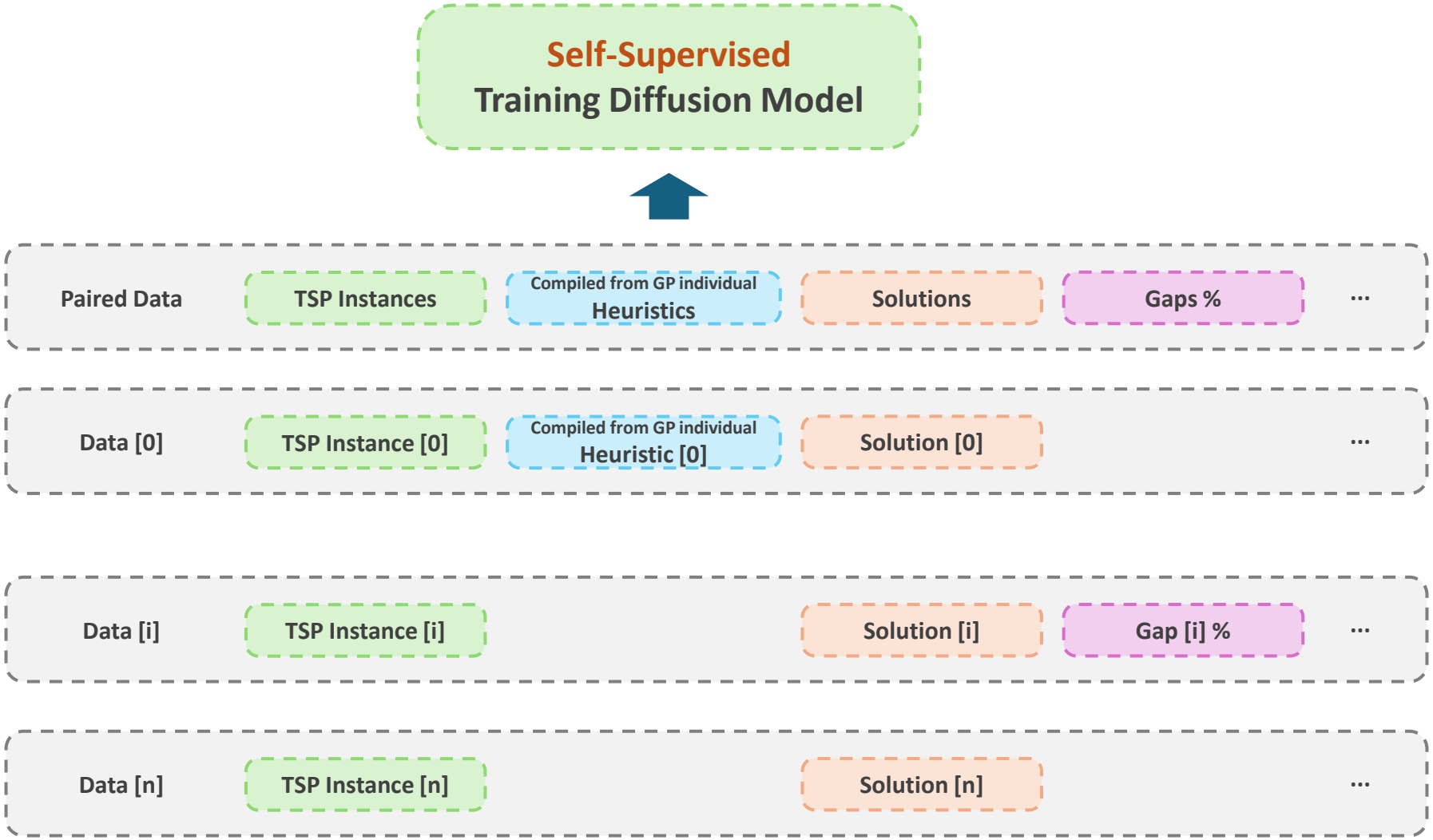
Objective 3
3.2 Explore Self-Supervised Training Methods



Objective 3
3.2 Explore Self-Supervised Training Methods



Objective 3
3.2 Explore Self-Supervised Training Methods



Optional

Objective 4

Enhancing GP and NCO Methods on Large-Scale and Variant TSP Problems through Transfer Learning and Domain Adaptation

Current Limitations

4 Insufficient Research on Transfer Learning

When facing new variant of TSPs

NCOs

Re-collect training data, re-train

Heuristics

Re-design heuristics manually

Focusing on differences between variants

Optional Objective 4

Transfer Learning and Domain Adaptation

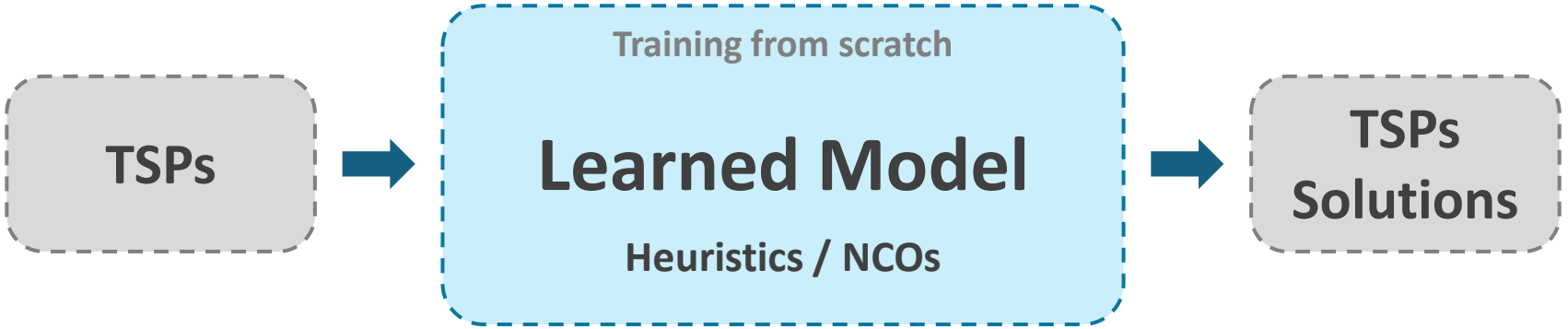
4.1 Develop a Transfer Learning Framework for GP and NCO

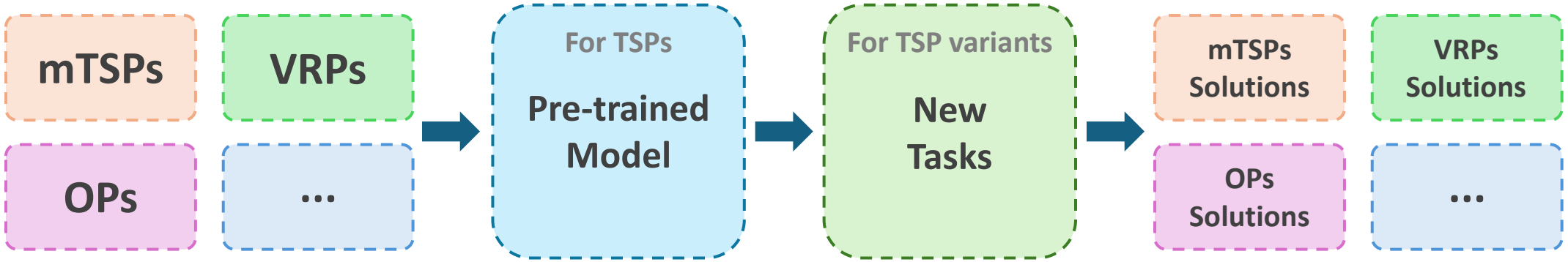
4.2 Domain Adaptation for Variant Problems

4.3 Transfer Learning for Large-Scale & Variant Problems

Optional Objective 4

Enhancing GP and Neural NCO Methods on Large-Scale and Variant TSP Problems through Transfer Learning and Domain Adaptation



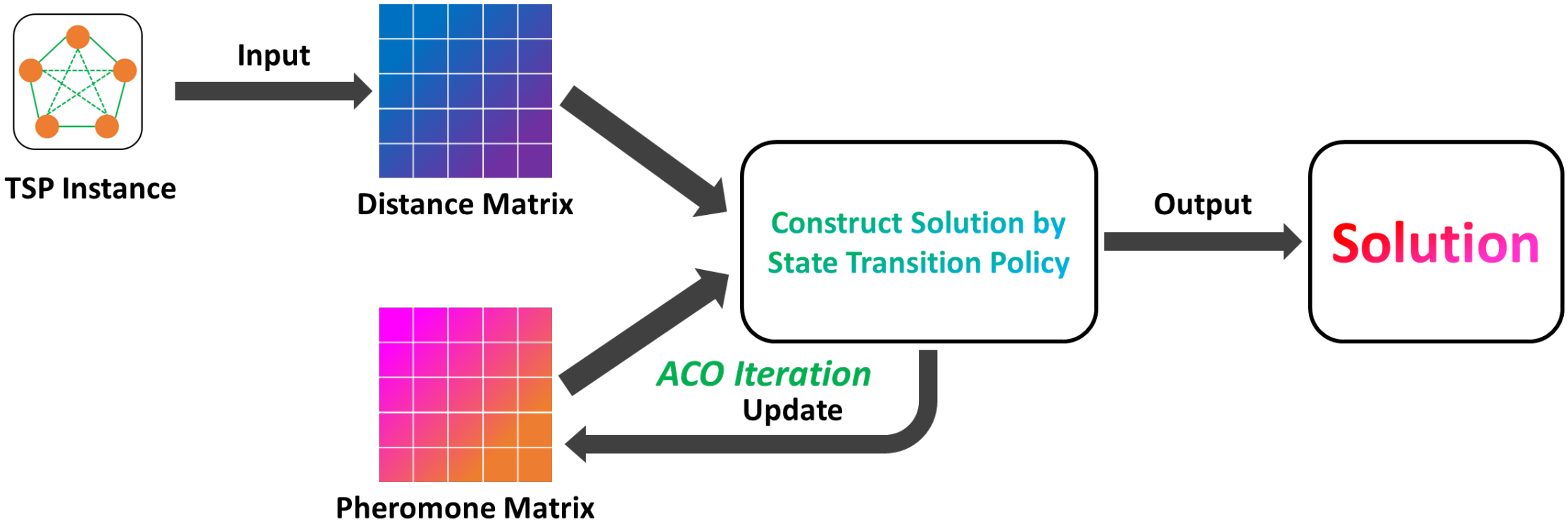


1. Introduction
2. Motivations
3. Research Objectives
4. Preliminary Works
5. Proposed Contributions and Research Plan

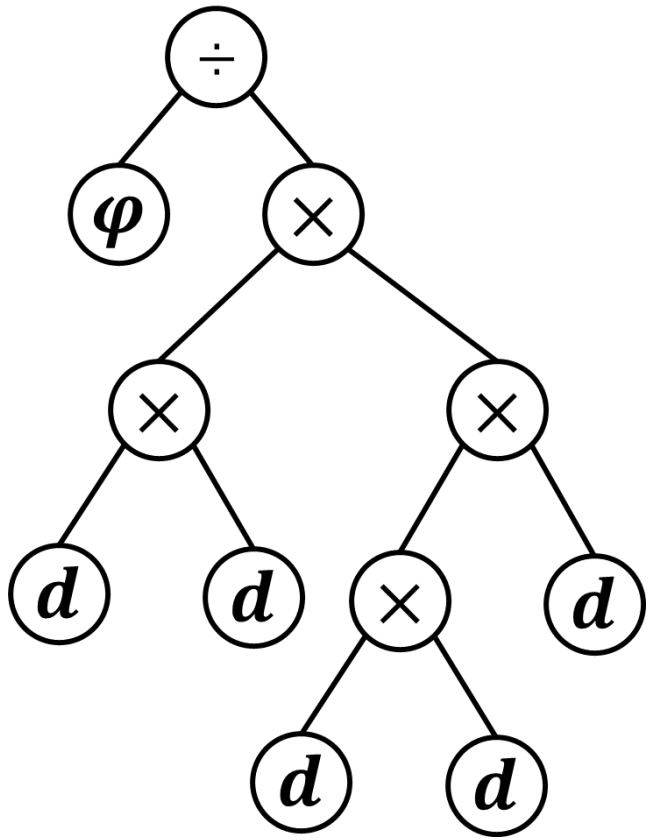
Automated Design of State Transition Rules in Ant Colony Optimization by Genetic Programming: A Comprehensive Investigation

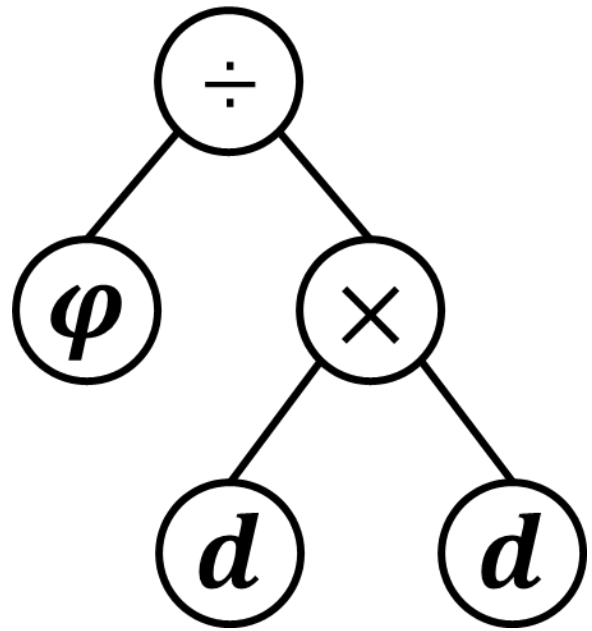
Bocheng Lin, Yi Mei*, Mengjie Zhang

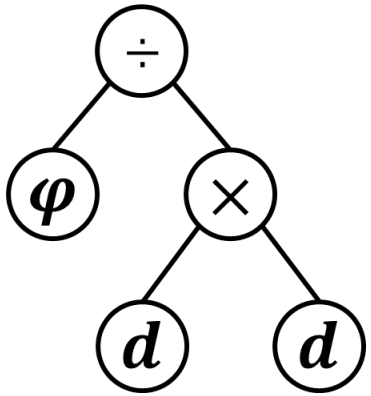
2024 IEEE Conference on Artificial Intelligence (IEEE CAI 2024) Workshop
Advances in Computational Intelligence and Machine Learning: Applications and Innovations
Special Issue of Memetic Computing Journal

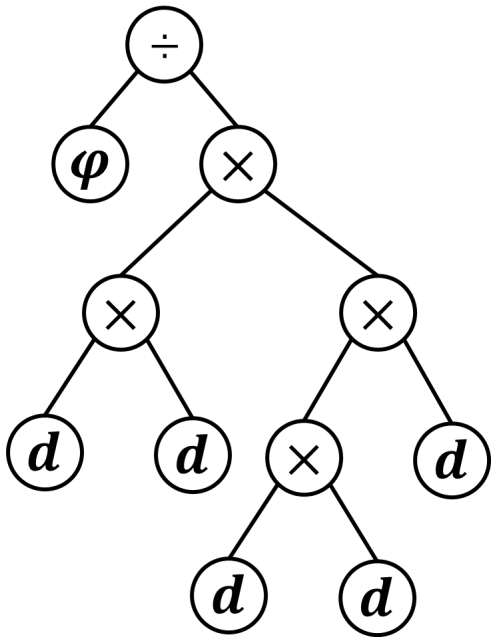

Input
TSP Instance
Distance Matrix
Construct Solution by
State Transition Policy
Output
Solution
ACO Iteration
Update
Pheromone Matrix

$$p_{ij} = \frac{\varphi_{ij}^{\alpha} / d_{ij}^{\beta}}{\sum_{l \in I_s} \varphi_{il}^{\alpha} / d_{il}^{\beta}}$$









Function Set

÷ × − +

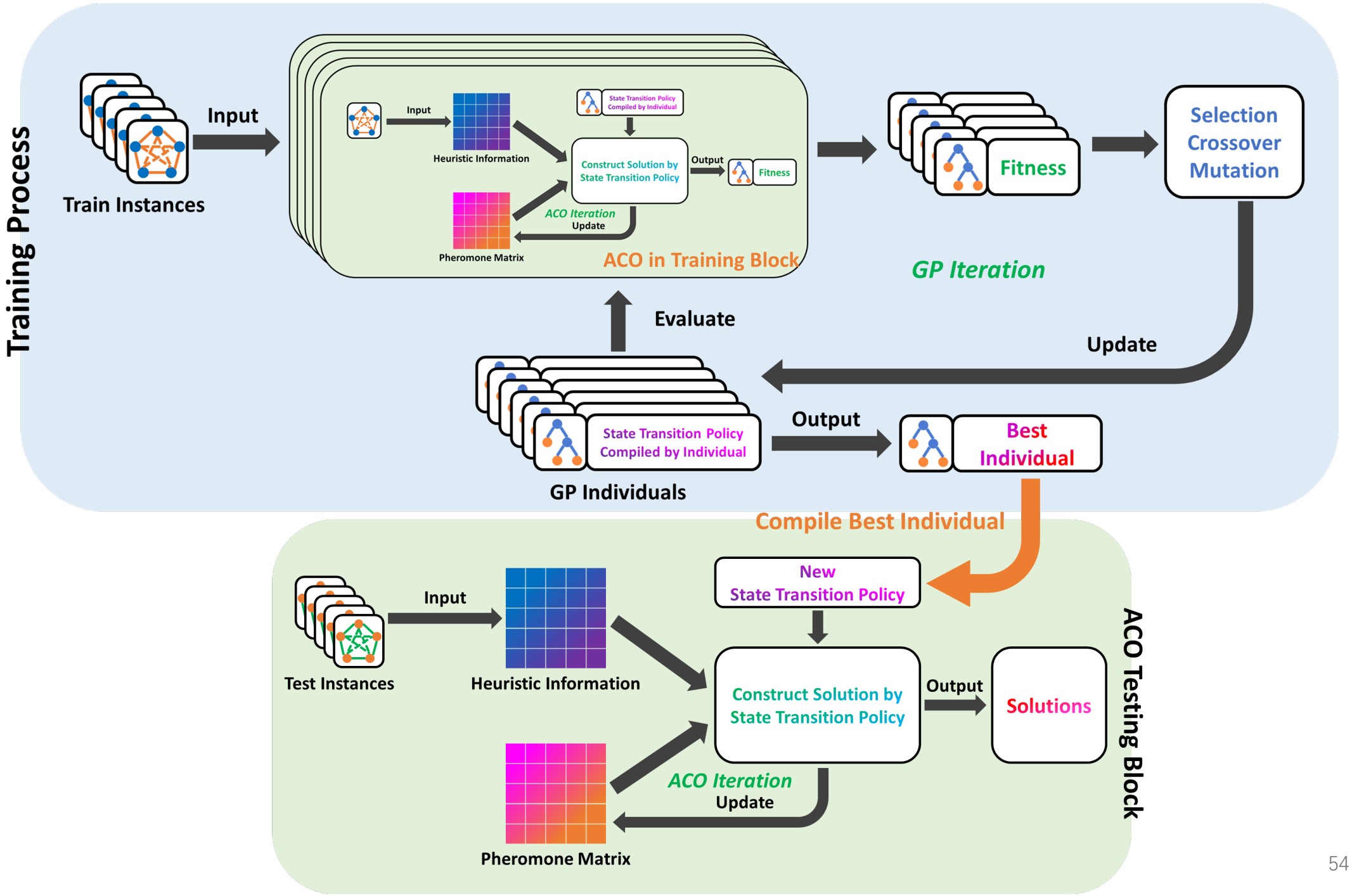
Terminal Set

φ d

Input

New
State Transition Policy

[2] Runka A. Evolving an edge selection formula for ant colony optimization[C]//Proceedings of the 11th Annual conference on Genetic and evolutionary computation. 2009: 1075-1082..


Training Process
Input
Train Instances
Input
Heuristic Information
State Transition Policy Compiled by Individual
Construct Solution by State Transition Policy
Output
Fitness
ACO Iteration
Update
Pheromone Matrix
ACO in Training Block
Fitness
Selection Crossover Mutation
GP Iteration
Update
Evaluate
State Transition Policy Compiled by Individual
GP Individuals
Output
Best Individual
Compile Best Individual
New State Transition Policy
Input
Test Instances
Heuristic Information
Construct Solution by State Transition Policy
Output
Solutions
ACO Iteration
Update
Pheromone Matrix
ACO Testing Block

Runka's Work in 2009

- Only trained on 1 instance and test on 2 instance
- Only Ant System
- No local search
- No global information

[2] Runka A. Evolving an edge selection formula for ant colony optimization[C]//Proceedings of the 11th Annual conference on Genetic and evolutionary computation. 2009: 1075-1082..

Questions:

1. Does GP-ACO exhibit generality when the distributions of training datasets and testing datasets are the same?
2. How do ACO variants influence the learning capabilities of GP-ACO?
3. Does the incorporation of local search weaken GP-ACO‘s learning effectiveness?
4. Would incorporating additional global information enhance GP-ACO’s learning capability?
5. Does GP-ACO remain effective when the distributions of training datasets and testing datasets are different?
6. How interpretable is GP-ACO?

Experiments to answer Questions:

Experiment 1: To answer Question 1 and Question 2

1. Does GP-ACO exhibit generality when the distributions of training datasets and testing datasets are the same?
2. How do ACO variants influence the learning capabilities of GP-ACO?

We apply the GP-ACO framework to AS and ACS and MMAS, without considering local search.

Train and test on three scales of TSP problems (TSP20, TSP20-100, TSP100)

Experiments to answer Questions:

Experiment 2: To answer Question 3

3. Does the incorporation of local search weaken GP-ACO‘s learning effectiveness?

We introduce local search to all the ACO variants in Experiment 1 while keeping the training and testing conditions unchanged.

Experiments to answer Questions:

Experiment 3: To answer Question 4

4. Would incorporating additional global information enhance GP-ACO's learning capability?

Building on Experiments 1 and 2, add four terminals containing global information to GP-ACO, which we refer to as xGP-ACO, while maintaining the same training and testing conditions.

Experiments to answer Questions:

Experiment 4: To answer Question 5

5. Does GP-ACO remain effective when the distributions of training datasets and testing datasets are different?

We directly use the state transition rules trained by xGP-ACO on the TSP100 dataset from Experiment 3, test on the publicly available TSPLIB dataset.

Datasets & Algorithm Settings

- **Datasets**

  Training: TSP20, TSP20-100, TSP100 *50 instances*

  Testing: TSP20, TSP20-100, TSP100 *10 instances*

  TSPLIB *15 instances*

- **Algorithm Settings**

GP

50 generations, 100 individuals,
0.8 crossover rate, 0.15 mutation rate,
0.05 reproduction rate,
10 elite individuals, 4 tournament size
5 maximum tree depth
30 independent training

ACO

20 (only TSP20) or 50 ants,
100 generations,
α = 1, β = 2 or 5 (only AS)
ρ = 0.9 or 0.98 (only MMAS)
q_0 = 0.9 (only ACS)
2-opt steps = cities/4

Each instance is generated using the portgen generator, which was previously used to create testbeds for the 8th DIMACS Implementation Challenge.
The portgen generator produces TSP instances, known as *rue* instances, by uniformly and randomly placing points on a two-dimensional plane.

Runka's Work in 2009

- Only trained on 1 instance and test on 2 instance
- Only Ant System
- No local search
- No global information

Before experiment, thinking:

Why Runka did not do these?

Why only Runka did this work without anyone following this?

[2] Runka A. Evolving an edge selection formula for ant colony optimization[C]//Proceedings of the 11th Annual conference on Genetic and evolutionary computation. 2009: 1075-1082..

Runka's Work in 2009

- Only trained on 1 instance and test on 2 instance
- Only Ant System
- No local search
- No global information

Execution Time

More than 50 hours

Table 1: GP parameters settings

Parameter	Value
Max Tree Depth	15
Generations	50
Population Size	200
Elitism	1 individual
Initialization	Ramped Half-and-half [2,6]
Selection	Tournament (size 4)
Crossover Type	Subtree Crossover
Crossover Percent	80%
Mutation Type	Subtree Mutation
Mutation Percent	20%
Reproduction	0%
Terminal Node Selection	10%

Table 2: AS parameters settings

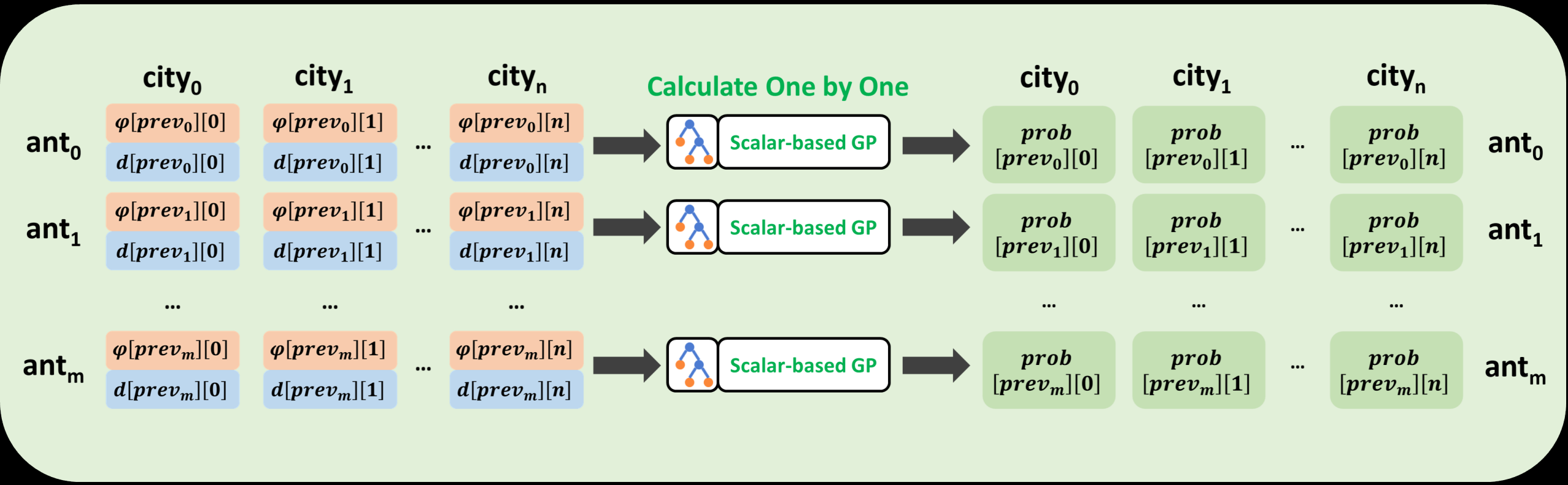
Parameter	Value
Duration	100
Evaporation Rate (ρ)	0.1
Number of ants	n (problem size)
α (for default formula)	2
β (for default formula)	1
Tournament Size (when needed)	7

[2] Runka A. Evolving an edge selection formula for ant colony optimization[C]//Proceedings of the 11th Annual conference on Genetic and evolutionary computation. 2009: 1075-1082..

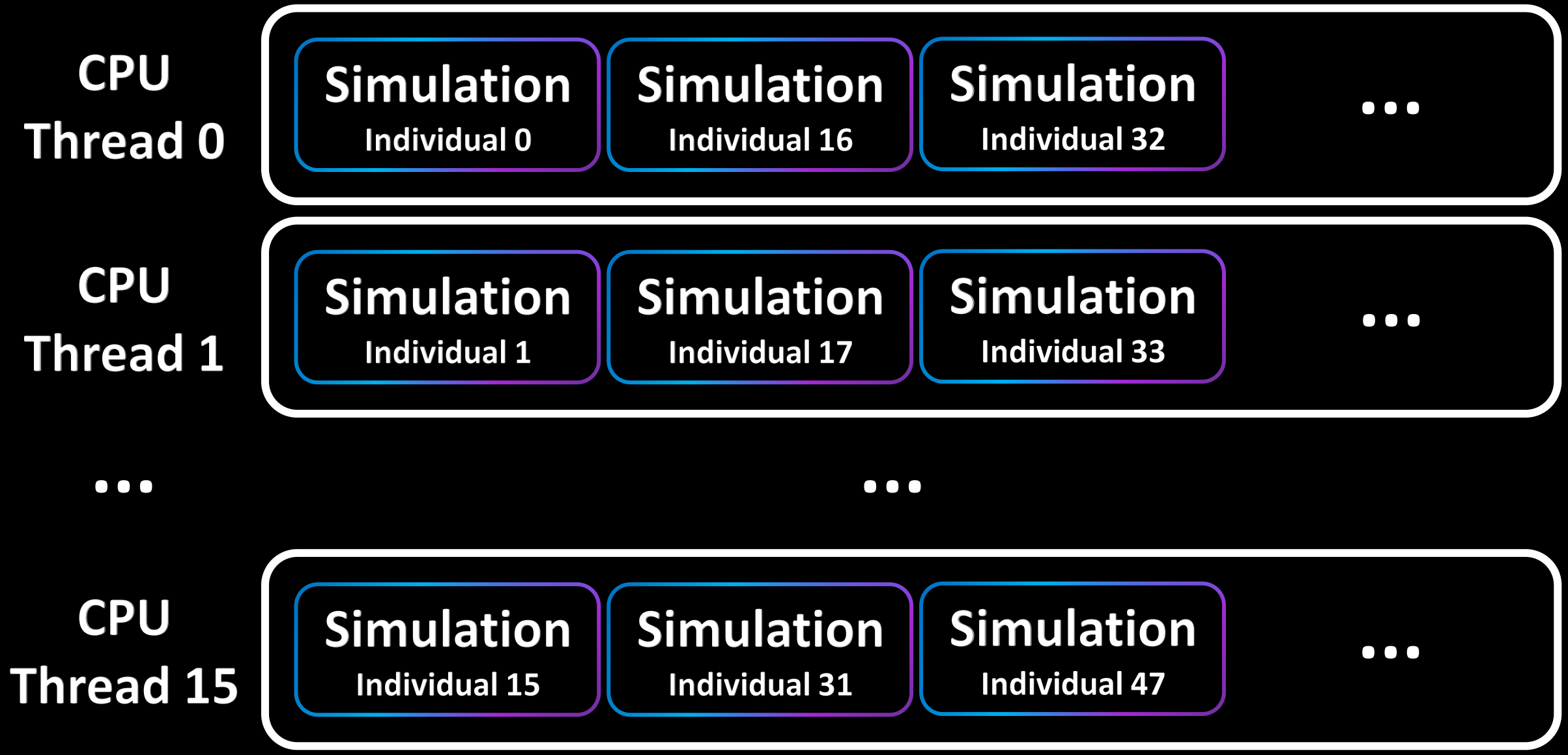
for *ant* **in** *all_ants*:

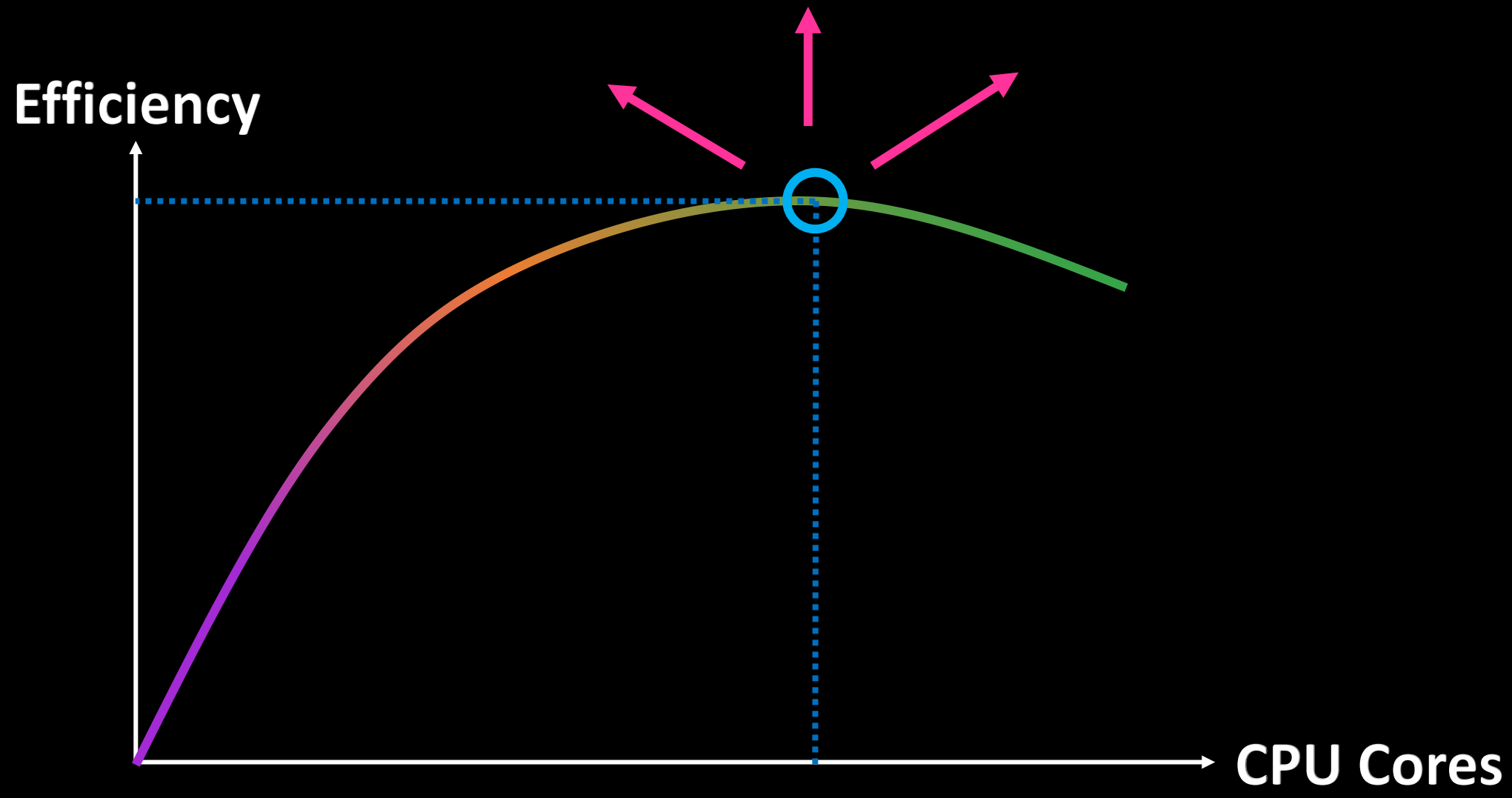
while *unvisited_cities* **is not empty:**

for *cities* **in** *unvisted_cities*:

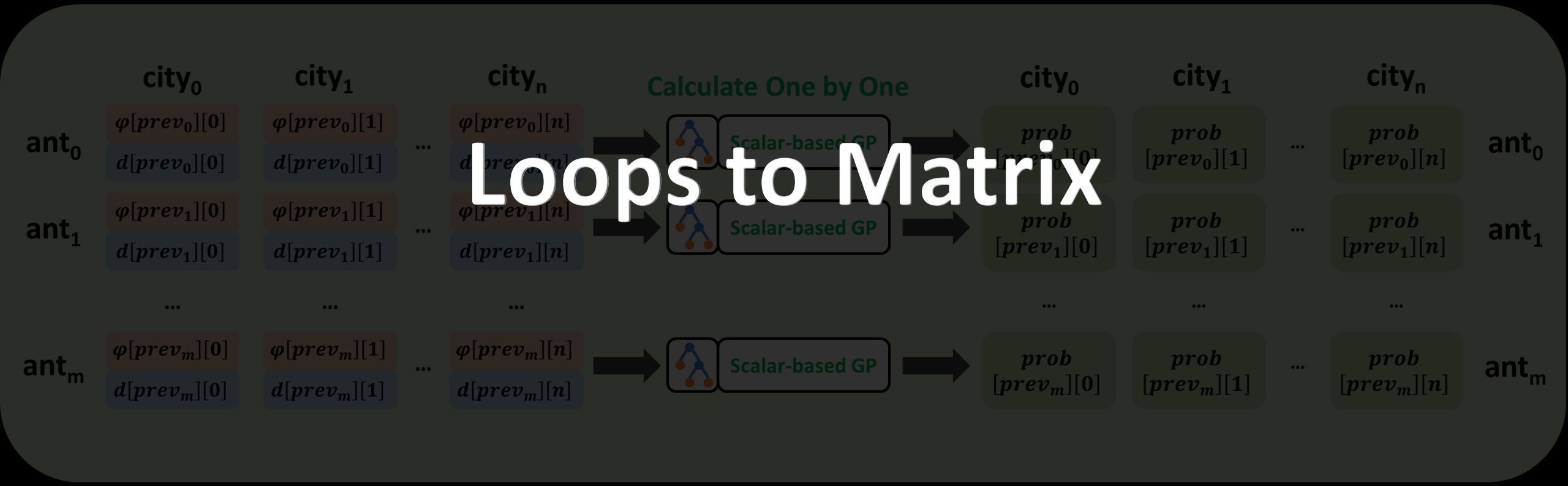


Parallel Simulation/Evaluation

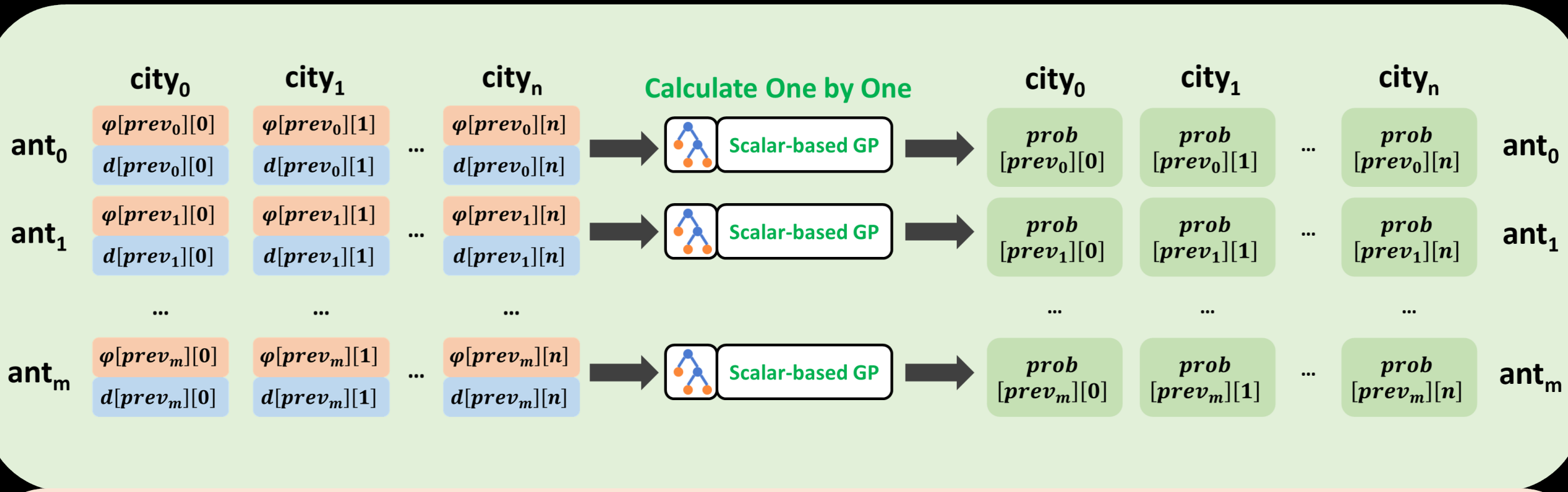



Efficiency
CPU Cores

for *ant* **in** *all_ants*:
while *unvisited_cities* **is not empty**:
for *cities* **in** *unvisted_cities*:

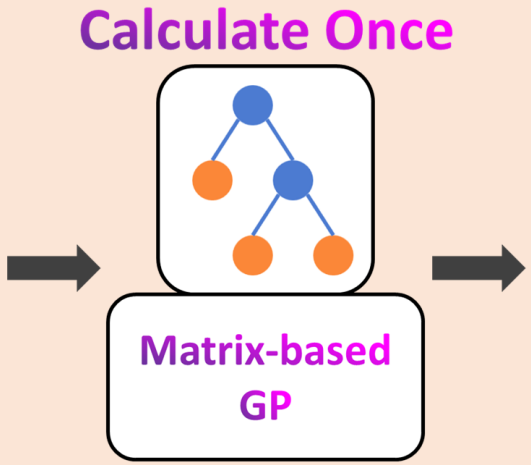


Loops to Matrix



$$\begin{bmatrix} \varphi[prev_0][0] & \cdots & \varphi[prev_0][n] \\ \vdots & \ddots & \vdots \\ \varphi[prev_m][0] & \cdots & \varphi[prev_0][n] \end{bmatrix}$$

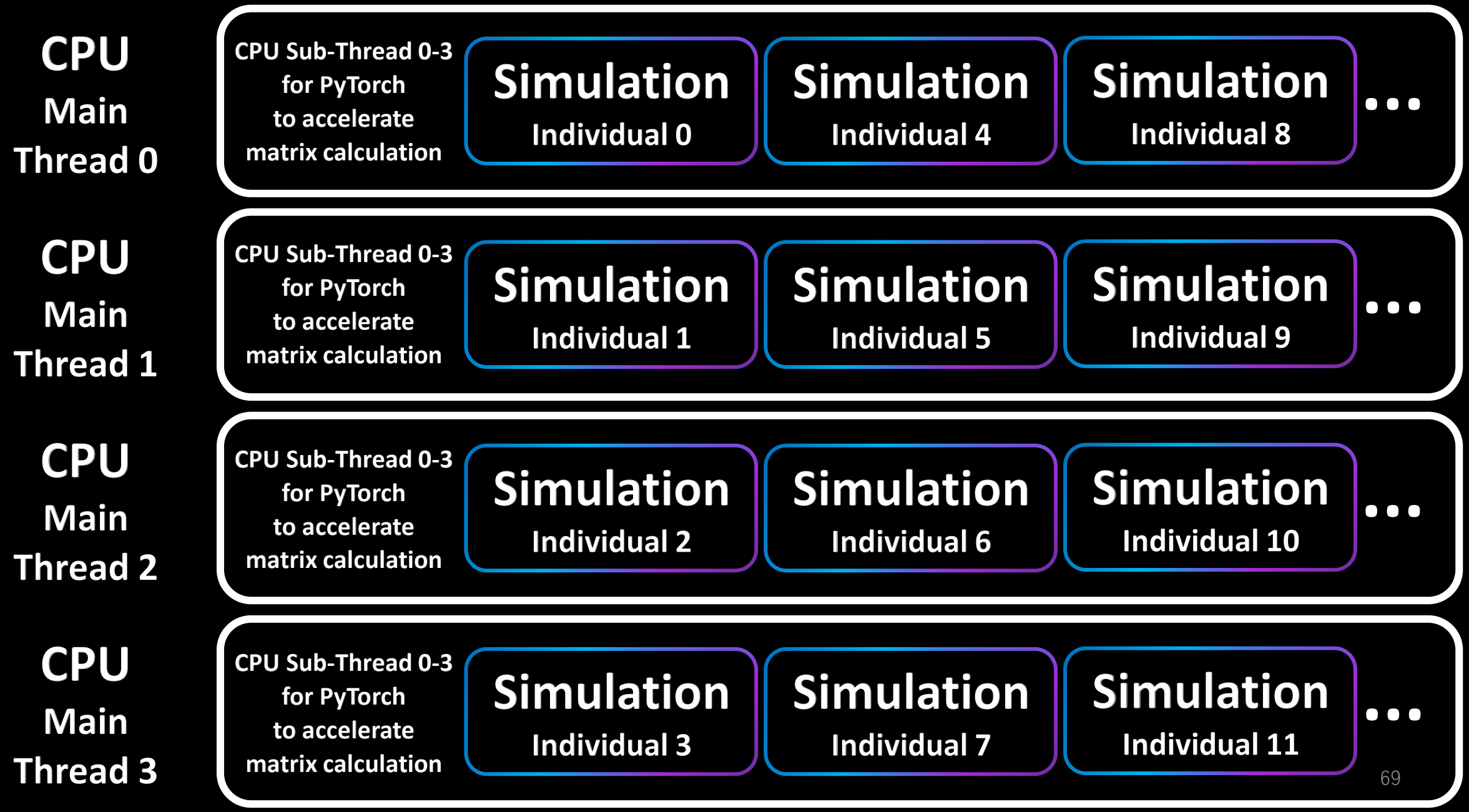
$$\begin{bmatrix} d[prev_0][0] & \cdots & d[prev_0][n] \\ \vdots & \ddots & \vdots \\ d[prev_m][0] & \cdots & d[prev_0][n] \end{bmatrix}$$

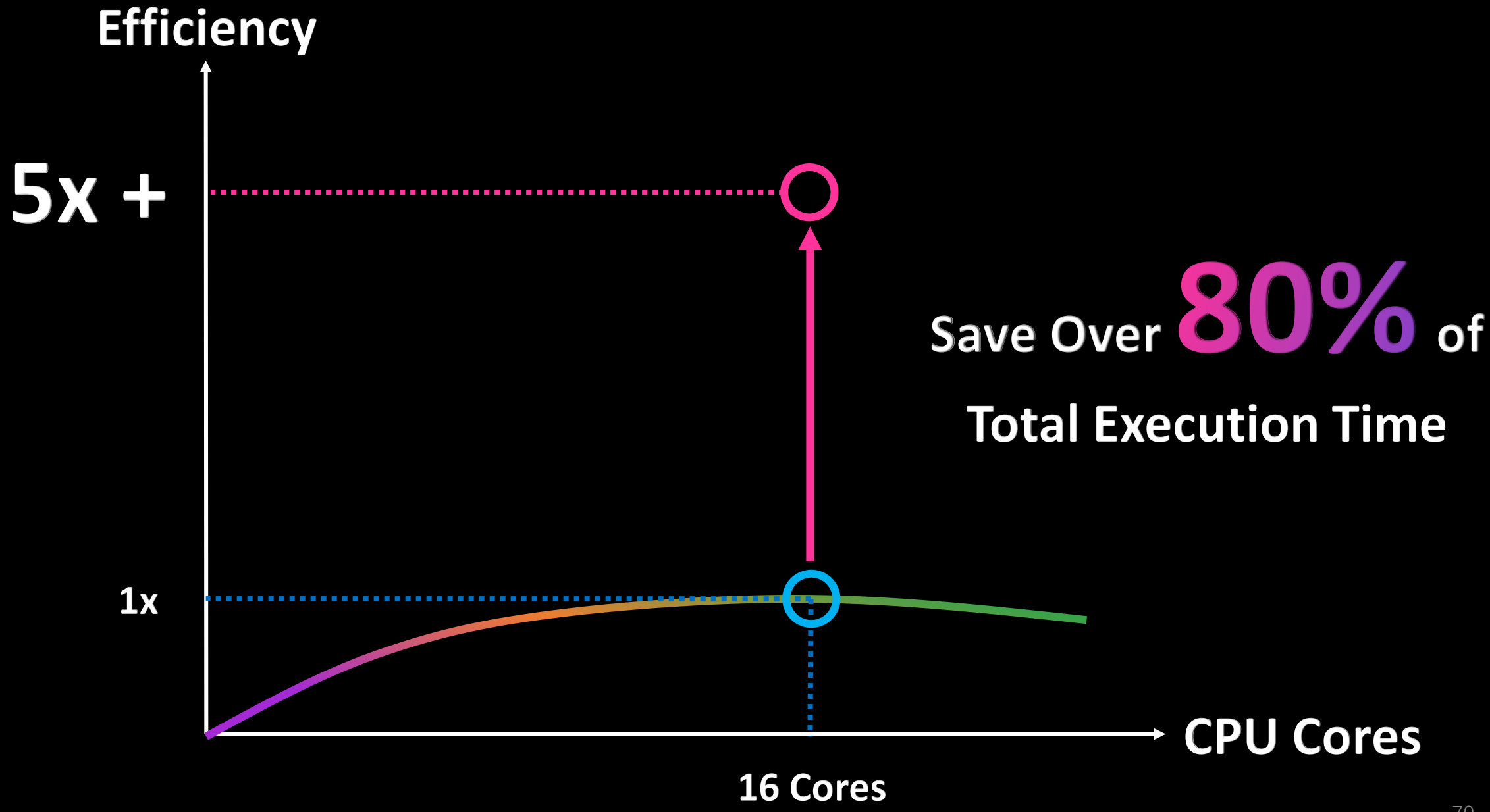


Accelerated by PyTorch

$$\begin{bmatrix} prob[prev_0][0] & \cdots & prob[prev_0][n] \\ \vdots & \ddots & \vdots \\ prob[prev_m][0] & \cdots & prob[prev_0][n] \end{bmatrix}$$

Parallel inside Simulation/Evaluation




Efficiency
5x +
1x
Save Over 80% of
Total Execution Time
CPU Cores
16 Cores

Experiment 1: To answer Question 1 and Question 2

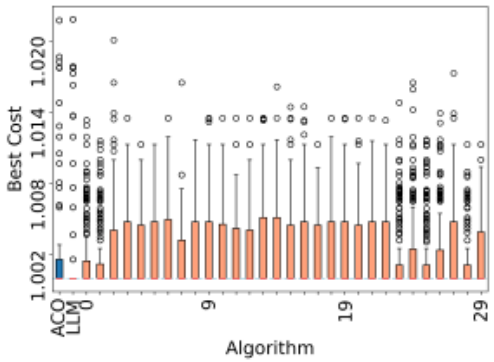
1. Does GP-ACO exhibit generality when the distributions of training datasets and testing datasets are the same?
2. How do ACO variants influence the learning capabilities of GP-ACO?

We apply the GP-ACO framework to AS and ACS and MMAS, without considering local search.

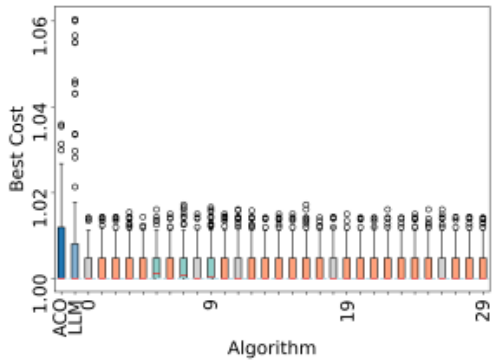
Train and test on three scales of TSP problems (TSP20, TSP20-100, TSP100)

Table 1: The normalized costs of GP-ACO, ACO baselines, and LLM-ACO on the test datasets.

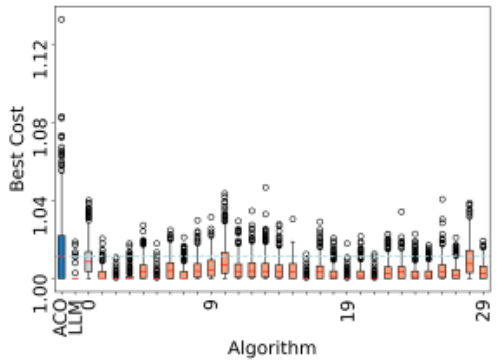
TSP		AS			ACS			MMAS		
		baseline	LLM	GP	baseline	LLM	GP	baseline	LLM	GP
20	mean	1.0028	1.0020	1.0019	1.0063	1.0074	**1.0025**	1.0164	**1.0031**	1.0039
		+	+		+	+		+	-	
	std	0	0	0.0001	0	0	0.0002	0	0	0.0002
	min	1	1	1	1	1	1	1	1	1
	max	1.0217	1.0218	1.0201	1.0359	1.0604	1.0172	1.1327	1.0186	1.0467
20-100	mean	1.0364	1.0271	**1.0265**	1.0499	1.0418	**1.0306**	1.0632	1.0076	1.0254
		+	+		+	+		+	-	
	std	0	0	0.0004	0	0	0.0004	0	0	0.0001
	min	1	1	1	1.0006	1.0016	1	1	1	1
	max	1.1042	1.0747	1.3813	1.1445	1.0831	1.1660	1.2101	1.0440	1.1212
100	mean	1.0784	1.0533	1.0105	1.0867	1.0721	**1.0446**	1.1248	**1.0144**	1.0400
		+	+		+	+		+	-	
	std	0	0	0.0001	0	0	0.0005	0	0	0.0001
	min	1.0359	1.0127	1.0005	1.0310	1.0148	1.0162	1.0481	1	1.0071
	max	1.1346	1.0962	1.0274	1.1451	1.1154	1.0799	1.2604	1.0680	1.0852



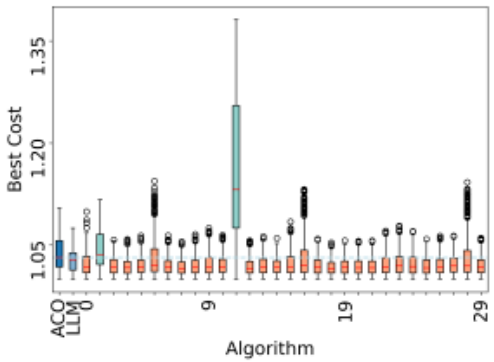
(a) TSP20 - GP-AS



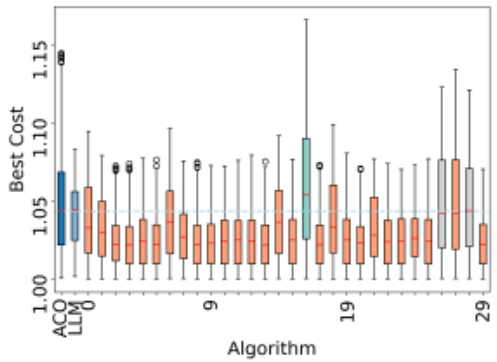
(b) TSP20 - GP-ACS



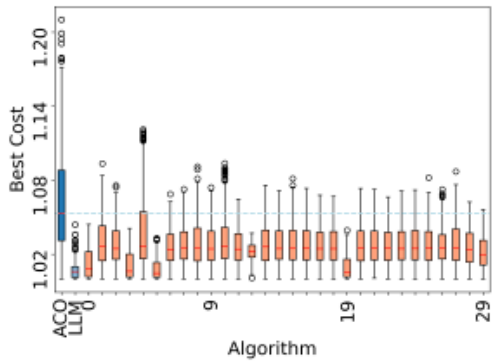
(c) TSP20 - GP-MMAS



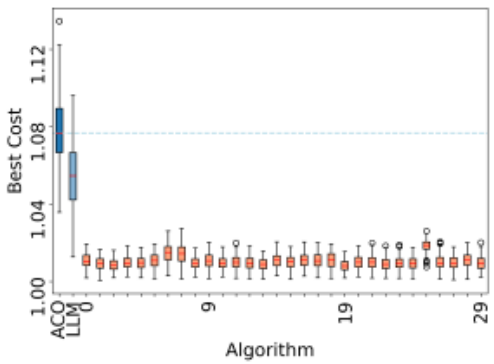
(d) TSP20-100 - GP-AS



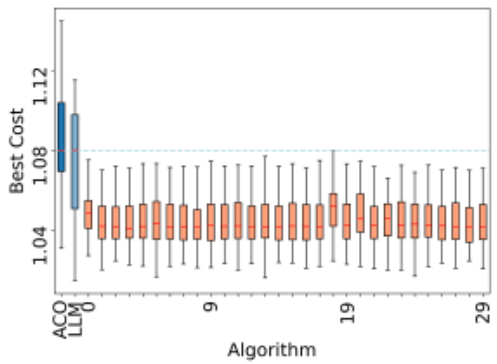
(e) TSP20-100 - GP-ACS



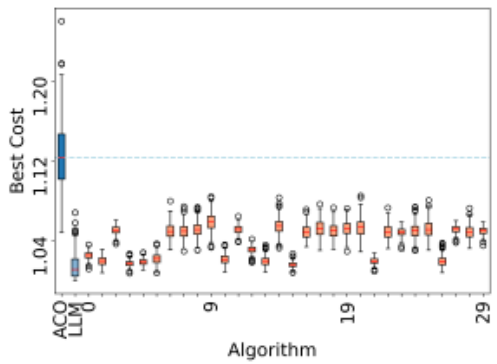
(f) TSP20-100 - GP-MMAS



(g) TSP100 - GP-AS



(h) TSP100 - GP-ACS



(i) TSP100 - GP-MMAS

Experiment 1: To answer Question 1 and Question 2

1. Does GP-ACO exhibit generality when the distributions of training datasets and testing datasets are the same?
2. How do ACO variants influence the learning capabilities of GP-ACO?

Answer 1: The GP-ACO demonstrates generality.

Across three scales of the TSP problem, the trained GP-ACO consistently outperforms the ACO baseline, and this advantage becomes more pronounced as the problem size increases.

Answer 2: Different ACO frameworks do not affect the learning capability of the GP-ACO.

In three variants of the ACO, the GP-ACO significantly surpasses the ACO baseline in performance. Additionally, we discovered that the state transition rules designed in GP-AS and GP-ACS notably outperform those in LLM-ACO, whereas the reverse is true for GP-MMAS.

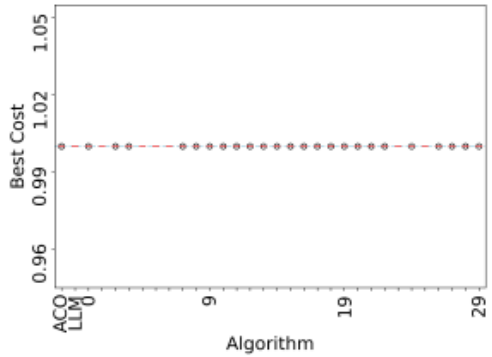
Experiment 2: To answer Question 3

3. Does the incorporation of local search weaken GP-ACO's learning effectiveness?

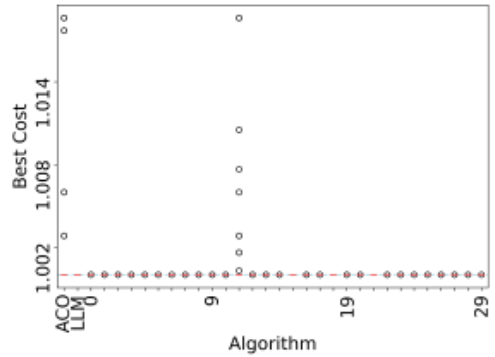
We introduce local search to all the ACO variants in Experiment 1 while keeping the training and testing conditions unchanged.

Table 2: The normalized costs of GP-ACO+2-opt, ACO+2-opt baselines, and LLM-ACO+2-opt on the test datasets.

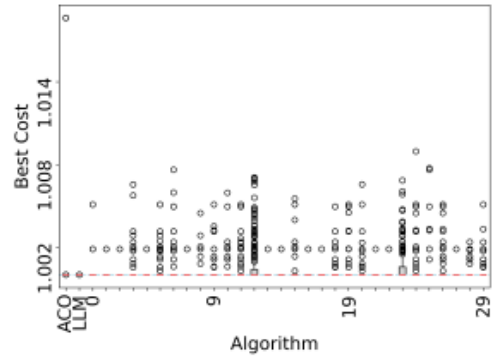
TSP		AS+2-opt			ACS+2-opt			MMAS+2-opt		
		baseline	LLM	GP	baseline	LLM	GP	baseline	LLM	GP
20	mean	1	1	1	1.0003	1	1	1.0002	1	1.0001
		=	=		+	-		+	-	
	std	0	0	0	0	0	0	0	0	0
	min	1	1	1	1	1	1	1	1	1
	max	1	1	1	1.0186	1.0000	1.0186	1.0186	1	1.0090
20-100	mean	1.0018	1.0037	1.0008	1.0021	1.0046	1.0008	1.0007	1.0008	1.0008
		+	+		+	+		-	+	
	std	0	0	0	0	0	0.0001	0	0	0.0001
	min	1	1	1	1	1	1	1	1	1
	max	1.0101	1.0162	1.0452	1.0097	1.0173	1.0123	1.0102	1.0084	1.0495
100	mean	1.0045	1.0091	1.0021	1.0045	1.0096	1.0017	1.0017	1.0011	1.0013
		+	+		+	+		=	-	
	std	0	0	0.0001	0	0	0.0001	0	0	0.0001
	min	1	1	1	1	1	1	1	1	1
	max	1.0184	1.0269	1.0136	1.0211	1.0281	1.0117	1.0112	1.0185	1.0276



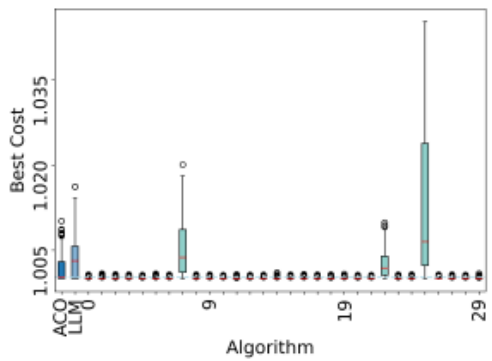
(a) TSP20 - GP-AS+2-opt



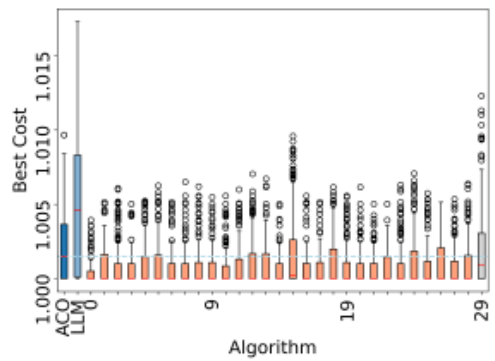
(b) TSP20 - GP-ACS+2-opt



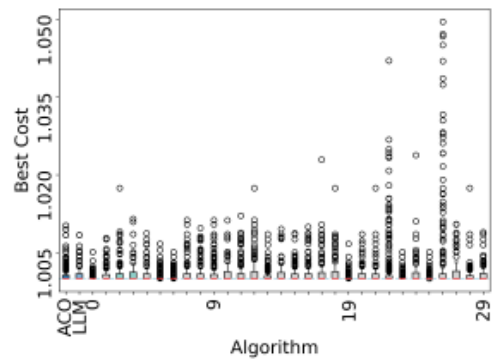
(c) TSP20 - GP-MMAS+2-opt



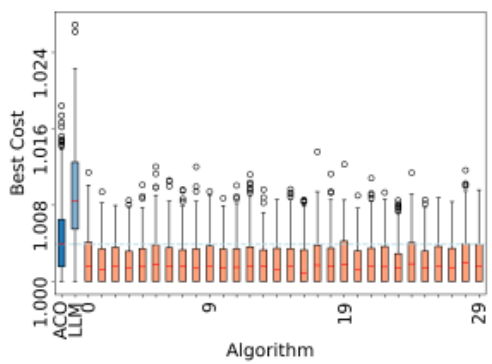
(d) TSP20-100 - GP-AS+2-opt



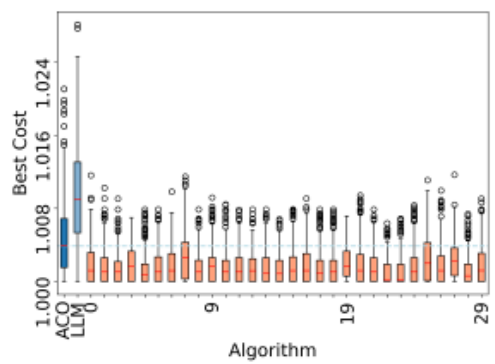
(e) TSP20-100 - GP-ACS+2-opt



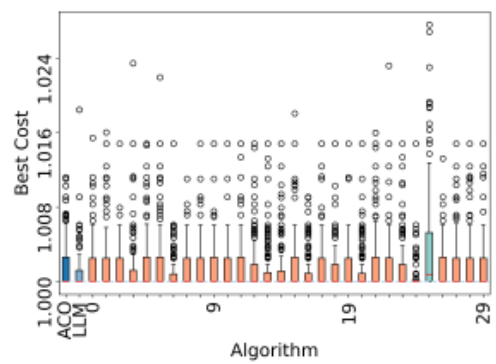
(f) TSP20-100 - GP-MMAS+2-opt



(g) TSP100 - GP-AS+2-opt



(h) TSP100 - GP-ACS+2-opt



(i) TSP100 - GP-MMAS+2-opt

Experiment 2: To answer Question 3

3. Does the incorporation of local search weaken GP-ACO's learning effectiveness?

Answer 3: The introduction of local search does not impact the learning effectiveness of GP-ACO in the GP-AS+2-opt and GP-ACS+2-opt scenarios,
but it diminishes the learning capabilities of GP-ACO in the GP-MMAS+2-opt scenario.

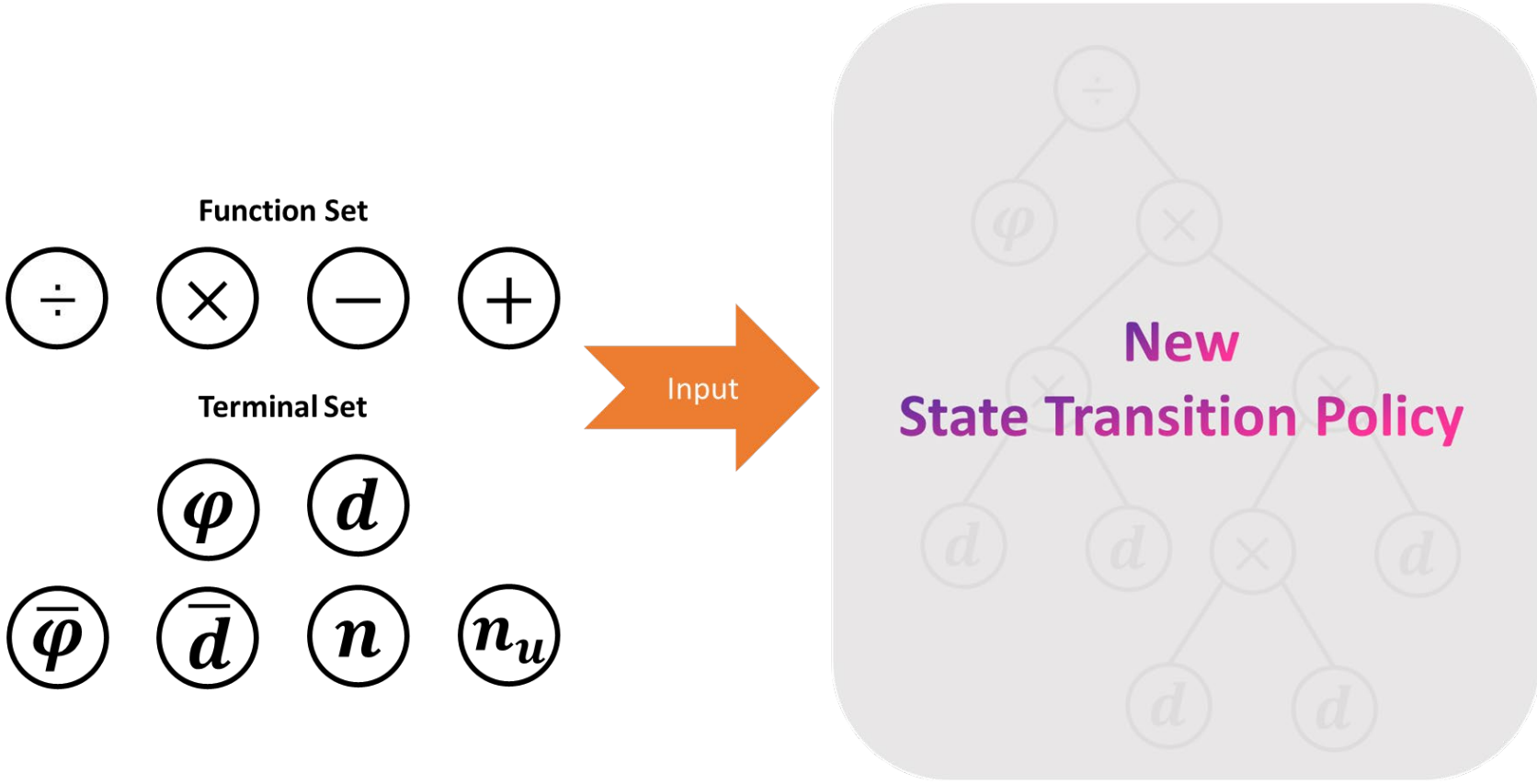
Enhancing Answer 1: GP-ACO+2-opt continues to exhibit generality.
However, in the case of GP-MMAS+2-opt, its performance in generality did not meet expectations.

Enhancing Answer 2: ACO variants that rely more on state transition rules are almost unaffected by the introduction of 2-opt local search regarding the learning capabilities of GP-ACO.
Conversely, ACO variants that depend more heavily on pheromone updates show a reduction in GP-ACO's learning abilities when local search is implemented.

Experiment 3: To answer Question 4

4. Would incorporating additional global information enhance GP-ACO's learning capability?

Building on Experiments 1 and 2, add four terminals containing global information to GP-ACO, which we refer to as xGP-ACO, while maintaining the same training and testing conditions.



average pheromone to the candidate node from the current node
average distance to the candidate node from the current node
total number of nodes
number of candidate nodes

TSP		AS					ACS					MMAS				
		baseline		LLM		GP	xGP	baseline		LLM		GP	xGP	baseline		LLM		GP	xGP
20	mean	1.0028	+	1.0020	+	1.0019	+ **1.0014**	1.0063	+	1.0074	+	**1.0025**	+ 1.0027	1.0164	+	1.0031	+	1.0039	+ **1.0015**
	std	0		0		0.0001	0.0001	0		0		0.0002	0.0002	0		0		0.0002	0.0002
	min	1		1		1	1	1		1		1	1	1		1		1	1
	max	1.0217		1.0218		1.0201	1.0159	1.0359		1.0604		1.0172	1.0172	1.1327		1.0186		1.0467	1.0403
20-100	mean	1.0364	+	1.0271	+	1.0265	+ **1.0235**	1.0499	+	1.0418	+	1.0306	+ **1.0293**	1.0632	+	1.0076	+	1.0254	+ **1.0029**
	std	0		0		0.0004	0.0004	0		0		0.0004	0.0004	0		0		0.0001	0.0001
	min	1		1		1	1	1.0006		1.0016		1	1	1		1		1	**0.9940**
	max	1.1042		1.0747		1.3813	1.1882	1.1445		1.0831		1.1660	1.1177	1.2101		1.0440		1.1212	1.0579
100	mean	1.0784	+	1.0533	+	1.0105	+ **1.0079**	1.0867	+	1.0721	+	1.0446	+ **1.0440**	1.1248	+	1.0144	+	1.0400	+ **1.0039**
	std	0		0		0.0001	0.0001	0		0		0.0005	0.0005	0		0		0.0001	0.0001
	min	1.0359		1.0127		1.0005	1.0008	1.0310		1.0148		1.0162	1.0156	1.0481		1		1.0071	**0.9881**
	max	1.1346		1.0962		1.0274	1.0194	1.1451		1.1154		1.0799	1.0797	1.2604		1.0680		1.0852	1.0400

TSP		AS+2-opt								ACS+2-opt								MMAS+2-opt							
		baseline		LLM		GP		xGP		baseline		LLM		GP		xGP		baseline		LLM		GP		xGP	
20	mean	**1**	-	**1**	=	**1**	=	**1**		1.0003	+	**1**	=	**1**	=	**1**		1.0002	+	**1**	-	1.0001	-	**1**	
	std	0		0		0		0		0		0		0		0		0		0		0		0	
	min	1		1		1		1		1		1		1		1		1		1		1		1	
	max	1		1		1		1.0063		1.0186		1.0000		1.0186		1		1.0186		1		1.0090		1.0103	
20-100	mean	1.0018	+	1.0037	+	1.0008	+	**1.0002**		1.0021	+	1.0046	+	**1.0008**	+	**1.0008**		**1.0007**	=	1.0008	+	1.0008	+	**1.0007**	
	std	0		0		0		0		0		0		0.0001		0.0001		0		0		0.0001		0.0001	
	min	1		1		1		1		1		1		1		1		1		1		1		1	
	max	1.0101		1.0162		1.0452		1.0053		1.0097		1.0173		1.0123		1.0080		1.0102		1.0084		1.0495		1.0728	
100	mean	1.0045	+	1.0091	+	1.0021	+	**1.0017**		1.0045	+	1.0096	+	**1.0017**	-	1.0018		1.0017	+	**1.0011**	-	1.0013	=	1.0013	
	std	0		0		0.0001		0.0001		0		0		0.0001		0.0001		0		0		0.0001		0.0001	
	min	1		1		1		1		1		1		1		1		1		1		1		1	
	max	1.0184		1.0269		1.0136		1.0142		1.0211		1.0281		1.0117		1.0141		1.0112		1.0185		1.0276		1.0276	

Experiment 3: To answer Question 4

4. Would incorporating additional global information enhance GP-ACO's learning capability?

Answer 4: The introduction of additional global information in xGP-ACO without local search proves highly effective,
significantly enhancing learning capabilities compared to GP-ACO.
With local search, incorporating more global information significantly improves the learning capabilities of GP-AS+2-opt
but weakens those of xGP-MMAS+2-opt, which has less dependency on state transition rules.
Enhancing Answer 2 & 3: After introducing more global information, and excluding the consideration of local search, **variations in ACO frameworks do not reduce the learning capabilities of xGP-ACO.**
However, when the local search is introduced, **the lesser dependence on state transition rules in MMAS variants diminishes the learning capabilities of xGP-ACO+2-opt.**

Experiment 4: To answer Question 5

5. Does GP-ACO remain effective when the distributions of training datasets and testing datasets are different?

We directly use the state transition rules trained by xGP-ACO on the TSP100 dataset from Experiment 3, test on the publicly available TSPLIB dataset.

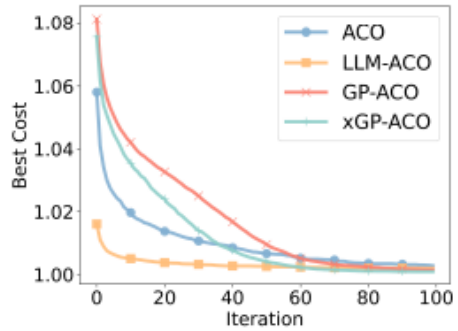
TSP instances	AS				ACS				MMAS			
	baseline	LLM	xGP	xGP-best	baseline	LLM	xGP	xGP-best	baseline	LLM	xGP	xGP-best
bier127	10.75	**5.52**	6.62	8.83	17.84	6.60	7.83	**6.56**	6.80	2.06	5.45	**0.54**
ch130	9.84	5.57	0.88	**0.75**	20.59	**6.19**	10.11	9.63	7.98	2.70	**0.71**	1.53
d493	21.23	11.31	11.20	**3.15**	32.53	**10.90**	16.25	13.97	11.22	8.42	**4.16**	4.98
eil51	7.57	4.01	1.11	**0.78**	13.20	**5.04**	10.47	9.29	5.21	1.71	1.11	**0.31**
fl417	16.90	14.46	4.26	**1.81**	26.12	15.11	12.68	**12.18**	9.98	9.29	**2.53**	4.83
kroA150	13.76	9.26	3.47	**2.23**	23.55	**10.02**	13.67	11.62	8.83	4.51	1.92	**1.05**
kroB100	7.03	5.46	**1.35**	1.36	16.36	7.41	**5.76**	6.30	7.43	1.42	1.46	**0.26**
kroC100	6.92	4.68	1.25	**1.08**	17.05	**5.49**	7.75	7.22	7.96	1.14	1.39	**0.18**
lin318	17.34	10.21	11.25	**6.40**	26.70	**9.65**	16.51	15.74	10.73	6.50	**3.87**	3.93
pr226	15.00	**7.28**	10.33	9.30	17.58	**9.89**	12.01	10.27	8.74	**2.20**	5.62	2.24
pr264	13.62	8.92	8.62	**7.52**	25.30	**9.30**	12.75	11.49	9.82	5.99	3.81	**3.30**
pr299	20.97	10.36	13.36	**7.92**	33.72	**11.21**	19.04	17.28	12.23	8.13	4.23	**3.69**
pr439	23.52	**10.94**	21.67	16.86	36.07	**13.35**	19.71	16.19	12.27	7.31	7.53	**4.45**
rat99	9.00	6.50	1.18	**1.01**	15.58	**8.60**	10.54	10.76	8.49	3.61	**1.85**	1.97
ts225	9.25	**3.70**	14.82	19.71	23.17	**3.74**	8.91	5.56	12.08	3.10	8.71	**0.70**
Avg. opt. gap	13.51	7.88	7.43	**5.91**	23.02	**8.83**	12.27	10.93	9.32	4.54	3.62	**2.26**

TSP instances	AS+2opt				ACS+2opt				MMAS+2opt			
	baseline	LLM	xGP	xGP-best	baseline	LLM	xGP	xGP-best	baseline	LLM	xGP	xGP-best
bier127	0.85	0.74	0.99	**0.51**	1.26	0.71	0.38	**0.26**	3.60	0.07	2.05	**0.03**
ch130	0.94	1.25	**0.03**	0.25	1.89	1.40	**0.94**	1.01	4.00	0.22	0.20	**0.14**
d493	2.30	2.60	1.82	**1.42**	3.30	2.25	2.07	**1.82**	7.96	1.12	4.41	**0.58**
eil51	0.95	0.90	**0.39**	0.68	1.50	0.94	0.72	**0.71**	4.38	0.70	**0.41**	0.69
fl417	1.13	1.67	**0.35**	0.83	1.74	1.62	1.07	**1.05**	2.60	0.59	5.49	**0.52**
kroA150	1.30	1.81	**0.52**	0.69	2.04	2.11	0.67	**0.56**	4.60	0.21	1.41	**0.04**
kroB100	0.52	0.65	0.27	**0.26**	0.81	0.40	0.18	**0.21**	3.06	0.09	0.25	**0.08**
kroC100	0.28	0.67	0.22	**0.008**	0.65	0.81	**0.04**	**0.04**	2.56	0.05	0.26	**0.03**
lin318	2.38	2.41	**1.26**	1.72	3.29	2.06	1.65	**1.03**	5.74	0.58	5.74	**0.45**
pr226	0.40	0.86	0.55	**0.26**	0.89	1.03	**0.51**	0.52	1.14	0.03	8.63	**0.007**
pr264	1.04	0.73	0.89	**0.21**	1.43	2.49	0.27	**0.16**	5.45	**0.007**	5.49	0.07
pr299	1.62	1.85	1.66	**1.22**	2.43	1.96	1.18	**1.02**	7.60	0.33	6.35	**0.18**
pr439	1.49	1.84	3.50	**1.26**	2.20	1.63	1.34	**1.00**	7.17	0.55	9.36	**0.22**
rat99	1.04	1.44	**0.38**	0.71	2.09	1.16	**0.74**	0.75	7.35	0.72	0.46	**0.69**
ts225	0.54	0.74	3.29	**0.48**	0.72	1.37	0.32	**0.29**	3.34	0.33	7.61	**0.002**
Avg. opt. gap	1.12	1.34	1.08	**0.70**	1.75	1.46	0.80	**0.70**	4.70	0.37	3.87	**0.25**

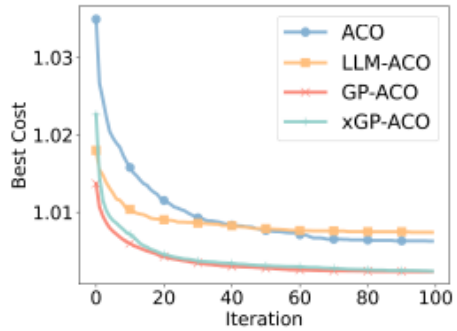
Experiment 4: To answer Question 5

5. Does GP-ACO remain effective when the distributions of training datasets and testing datasets are different?

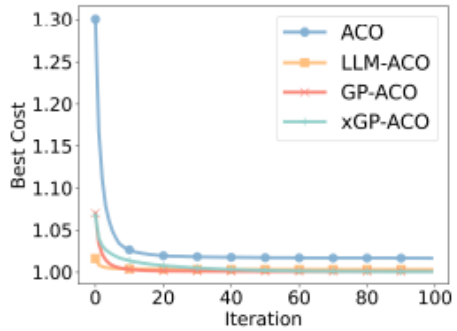
Answer 5: **The state transition rules learned by xGP-ACO and xGP-ACO+2-opt demonstrate strong performance across maps of varying scales and distributions.** **This indicates that the generality capabilities of xGP-ACO extend beyond training scenarios, adapting effectively to diverse and larger-scale environments.**



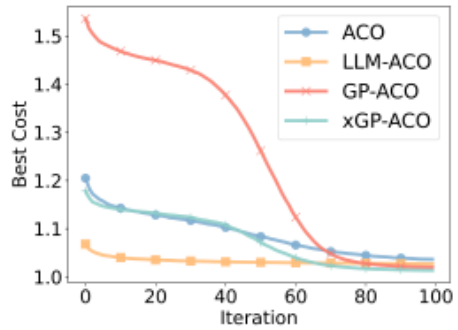
(a) TSP20-AS



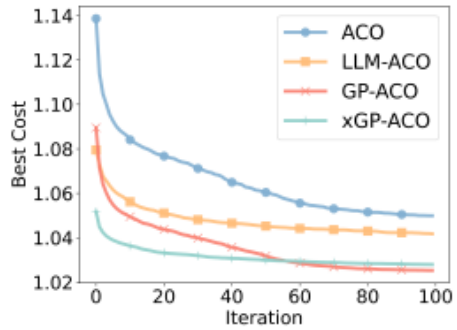
(b) TSP20-ACS



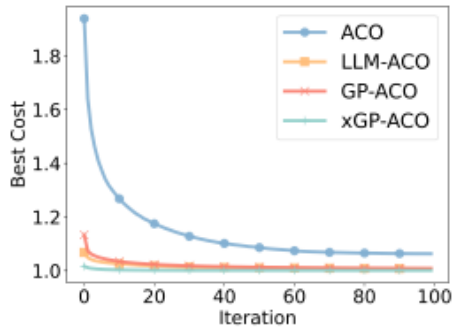
(c) TSP20-MMAS



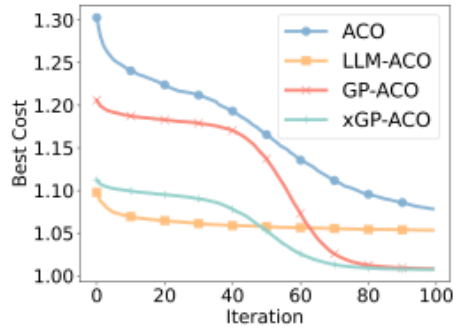
(d) TSP20-100-AS



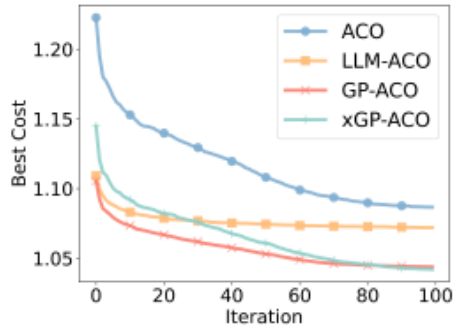
(e) TSP20-100-ACS



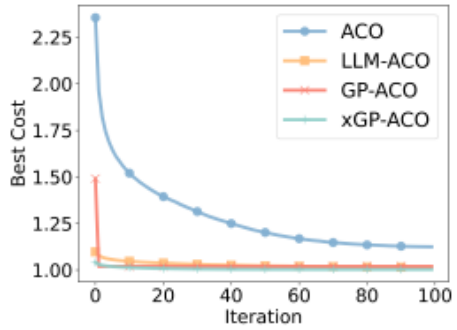
(f) TSP20-100-MMAS



(g) TSP100-AS



(h) TSP100-ACS



(i) TSP100-MMAS

Fig. 7: Convergence of ACO baseline, LLM-ACO, GP-ACO and xGP-ACO

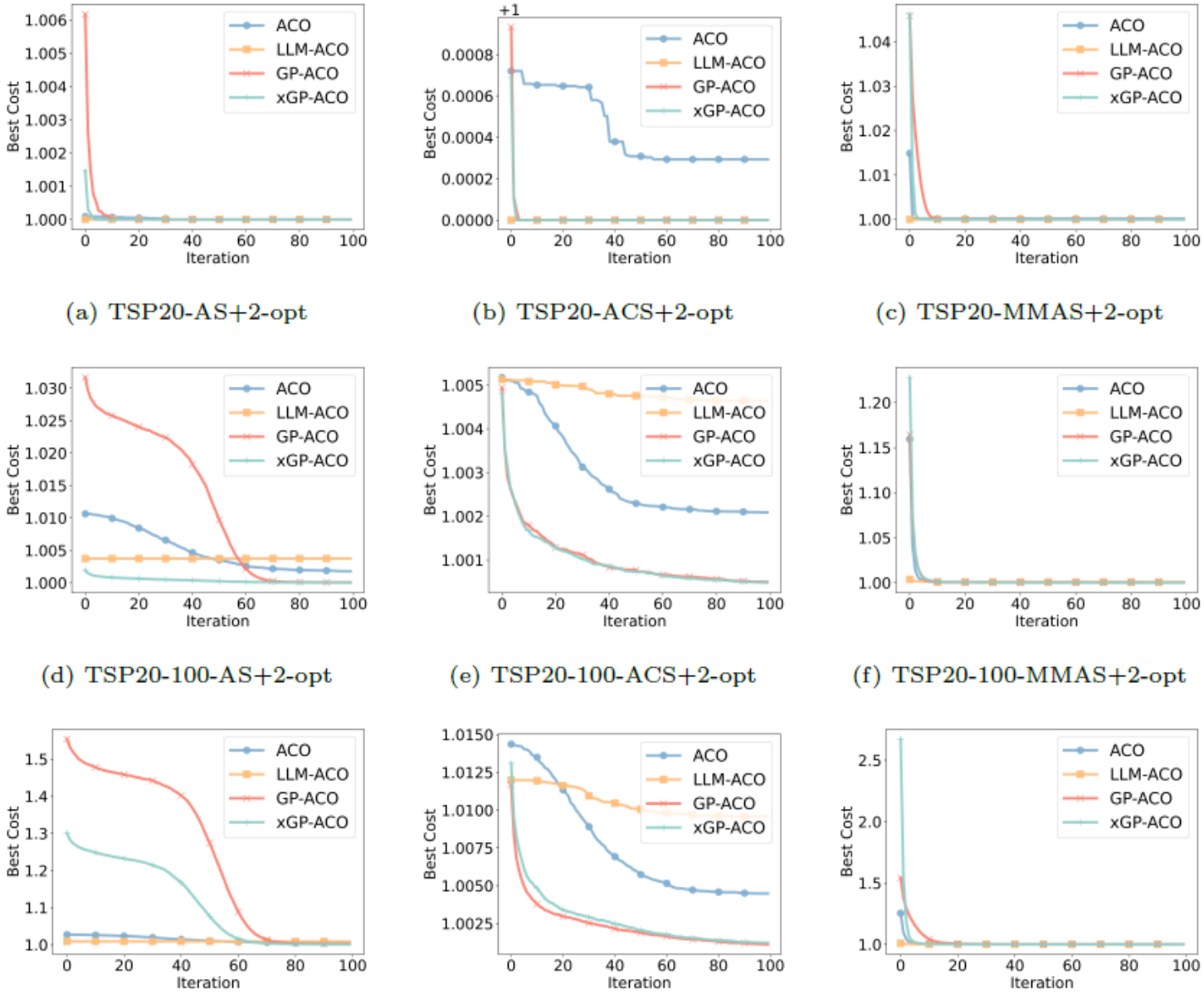


Fig. 8: Convergence of ACO+2-opt baseline, LLM-ACO+2-opt, GP-ACO+2-opt and xGP-ACO+2-opt

Table 7: New State Transition rules learned by GP-ACO.

Algorithm	TSP20	TSP20-100	TSP100
GP-AS	$\frac{\varphi}{(d-2\varphi)d^2} - \varphi$	$\frac{\varphi}{(\varphi+d^4)d^5}$	$\frac{2\varphi}{(2\varphi+d^2)(2\varphi+d^4)d^3}$
GP-ACS	$\varphi d^3 - \frac{d^3}{\varphi} + 1$	$\frac{\left(d^3-1\right)\varphi}{(\varphi-d)d^4} - d$	$\frac{2\varphi}{(\varphi+d)^2 d} - 2d^2 + 1$
GP-MMAS	$-\varphi d - \frac{2d^2}{\varphi+d}$	$-\frac{2\varphi}{(\varphi+\varphi d+2d^2)d}$	$\frac{\varphi+\varphi(\varphi+d)d^2}{(\varphi+d)^2 d^4}$

Table 8: New State Transition rules learned by xGP-ACO.

Local Search	**xGP-AS**	**xGP-ACS**	**xGP-MMAS**
None	$\frac{n^2\varphi^2\bar{d}^2}{\varphi d\bar{d}+d^2}$	$\left(\frac{\bar{d}}{d}\right)^3 - \frac{\varphi}{d^2}$	$\frac{2\bar{\varphi}+\varphi d}{\bar{\varphi}^4+d^8}$
2-opt	$\frac{n^3\varphi^3\bar{d}^2}{(n\varphi+1)(n+\varphi)d}$	$\frac{\varphi^2\bar{d}}{d^2}$	$-\varphi^2 + (n - n_u - 2)\,\varphi - n_u - \bar{\varphi}$

To answer Question 6

6. How interpretable is GP-ACO?

Answer 6: GP-ACO and xGP-ACO exhibit excellent interpretability.

The primary goals of this paper are to answer four research questions that were formulated concerning

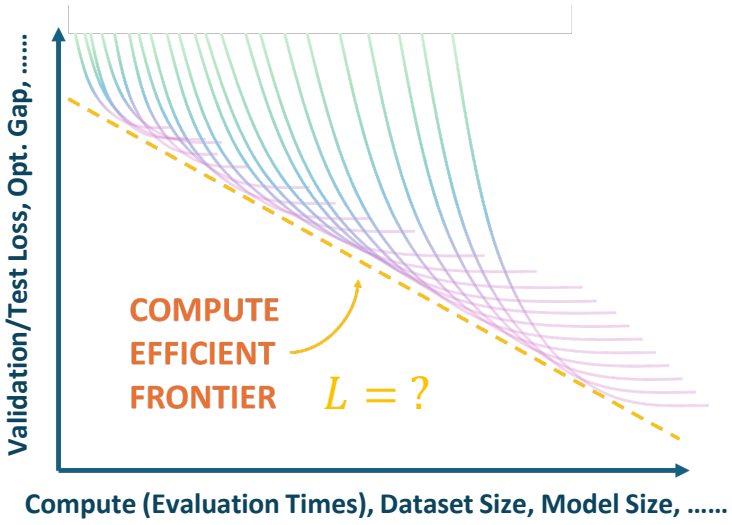
1. generality,
2. the impact of different ACO variants on GP-ACO,
3. the influence of the 2-opt local search on GP-ACO,
4. and the effects of incorporating more global information into GP-ACO.

From the experimental results, we observe that:

1. GP-ACO exhibits robust generality and performs well across problems of different scales and distributions;
2. Different ACO variants do not affect the learning capability of GP-ACO;
3. The introduction of 2-opt local search diminishes the learning capabilities of GP-ACO in the MMAS+2-opt variant, which has a reduced dependency on state transition rules;
4. Incorporating additional global information (xGP-ACO) significantly enhances the learning capabilities of GP-ACO;
5. GP-ACO and xGP-ACO show excellent interpretability.

1. Introduction
2. Motivations
3. Research Objectives
4. Preliminary Works
5. Proposed Contributions and Research Plan

Proposed Contribution 1
Development of Mathematical Scaling Laws for Heuristic and NCO Methods in TSP



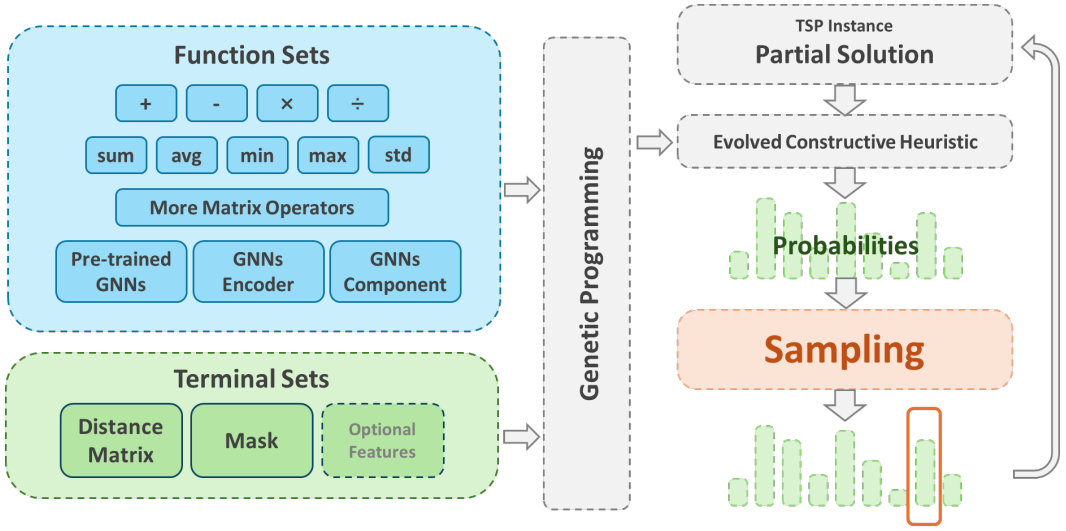
Key Focus:

- Characterize performance scaling with problem size, dataset size, model size, and …
- Predict performance and resource needs for large-scale TSP instances

Expect:

Enable efficient resolution of large TSP problems with improved energy efficiency and predictive performance.

Proposed Contribution 2
Integration of NCO Methods with GP for Enhanced Heuristic Algorithms



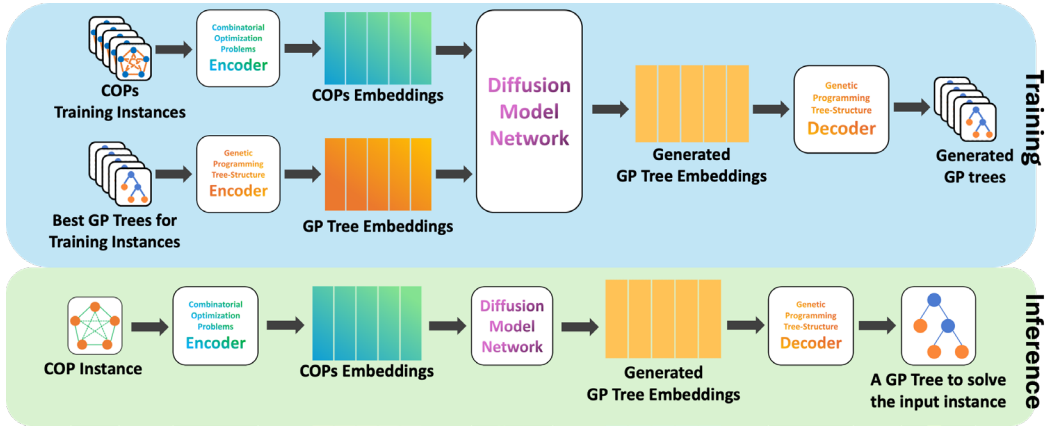
Key Focus:

- Augment GP function set with advanced operators and NCO techniques
- Improve generalization and scalability

Expect:

Outperform state-of-the-art heuristic approaches and deliver efficient and adaptable solutions for large-scale TSP.

Proposed Contribution 3
Development of Generative AI Models for Instance-Level Heuristic Generation in TSP



Key Focus:

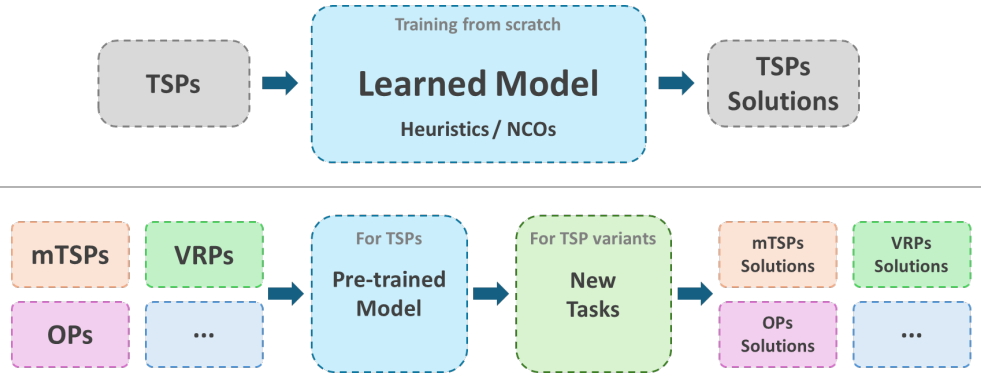
- First to apply generative AI (diffusion models) for heuristic generation in combinatorial optimization
- Adapt heuristics to instance-specific characteristics

Expect:

Enhanced solution quality and computational efficiency; Improved generalization and robustness

Optional Proposed Contribution 4

Enhancing GP and NCO Methods through Transfer Learning and Domain Adaptation for Large-Scale and Variant TSPs



Key Focus:

- Apply transfer learning to adapt models trained on small/simple instances to larger, complex instances, and different TSP variants

Expect:

Improve scalability and adaptability; Enhance generalization and robustness of GP and NCO methods

Research Plan

Phase	Task	Duration
1	Literature Review, overall design, and writing the proposal	12 (completed)
2.1	Develop Scaling Laws for NCO methods	10
2.2	Develop Scaling Laws for population-based heuristics methods	10
3.1	Extending the function set of GP	10
3.2	Develop a new sampling strategy	6
3.3	Prepare and structure data for training generative models	12
4.1	Develop a diffusion model for generating heuristics	12
4.2	Explore self-supervised training methods	6
5	Writing the thesis	4
2.3	Develop theoretical models to explain observed scaling laws	6 <optional>
6.1	Develop a transfer learning framework for GP and NCO methods	6 <optional>
6.2	Develop a domain adaptation method	6 <optional>

Timeline

Phase	Task	Time in Months											
		2	4	6	8	10	12	14	16	18	20	22	24
N/A	Literature Review	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.1	Develop Scaling Laws for NCO methods	✓	✓	✓	✓	✓							
2.2	Develop Scaling Laws for population-based heuristics methods	✓	✓	✓	✓	✓							
3.1	Extending the function set of GP		✓	✓	✓	✓	✓						
3.2	Develop a new sampling strategy				✓	✓	✓						
3.3	Prepare and structure data for training generative models		✓	✓	✓	✓	✓	✓					
4.1	Develop a diffusion model for generating heuristics for specific instances					✓	✓	✓	✓	✓	✓		
4.2	Explore self-supervised training methods							✓	✓	✓			
5.1	Write the first fraft of the thesis											✓	✓
5.2	Edit the final draft												✓
2.3	Develop theoretical models to explain observed scaling laws <optional>										✓	✓	✓
6.1	Develop a transfer learning framework for GP and NCO methods <optional>								✓	✓	✓		
6.2	Develop a domain adaptation method <optional>									✓	✓	✓	

Thesis Outline

Chapter 1: Introduction
Chapter 2: Literature Review
Chapter 3: Scaling Laws for Heuristics and NCO in TSP
Chapter 4: Integrating NCO with GP
Chapter 5: Diffusion Models for Generating Heuristics
<optional> Chapter 6: Transfer Learning and Domain Adaptation for GP and NCO
Chapter 7: Conclusion and Future Work

Thank you for your listening!

Bocheng Lin 2024.12

Shot at Tongariro National Park