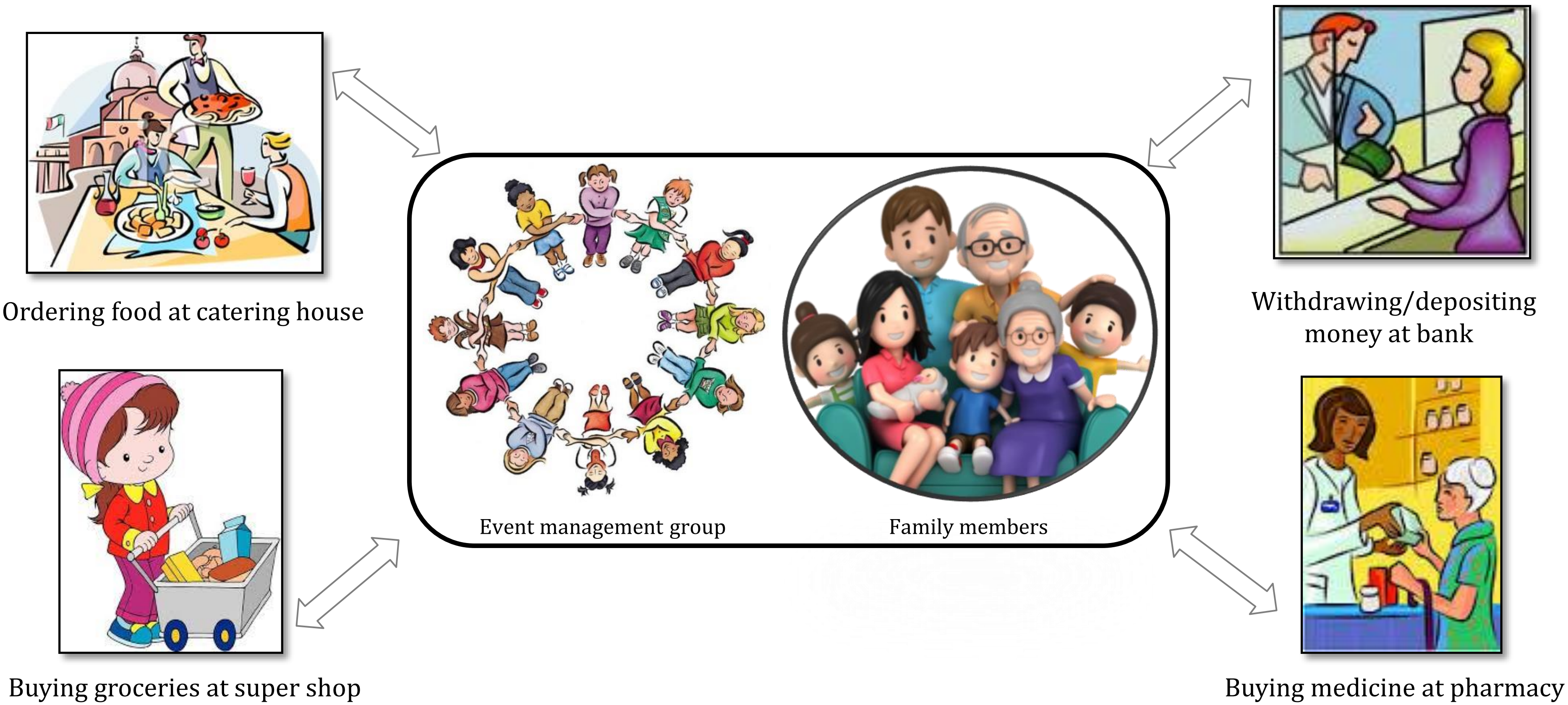
Efficient Scheduling of Generalized Group Trips in Road Networks

Student : Yeasir Rayhan (1305111)
Supervisor: Dr. Tanzima Hashem

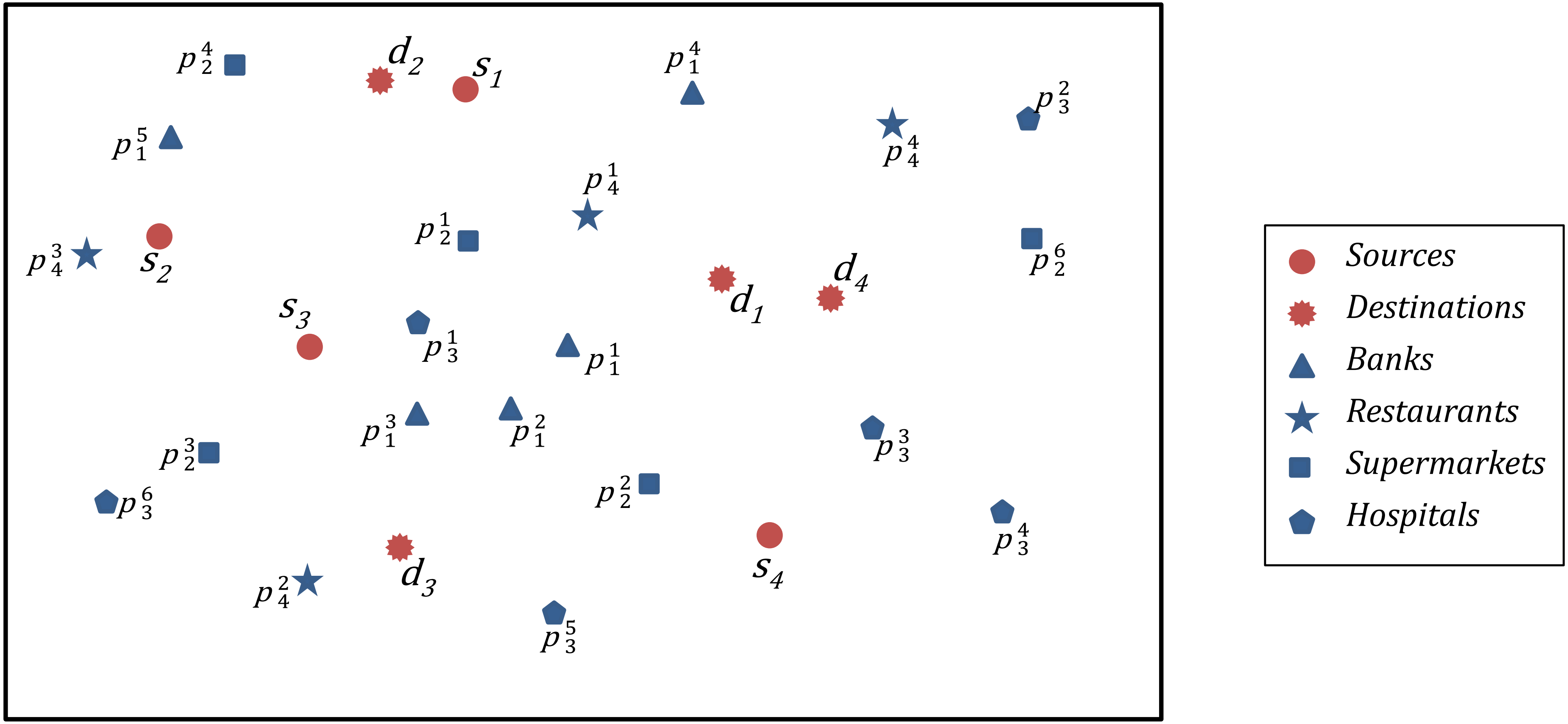
Motivation / Application

- Sharing multiple tasks in a family
- Event Management

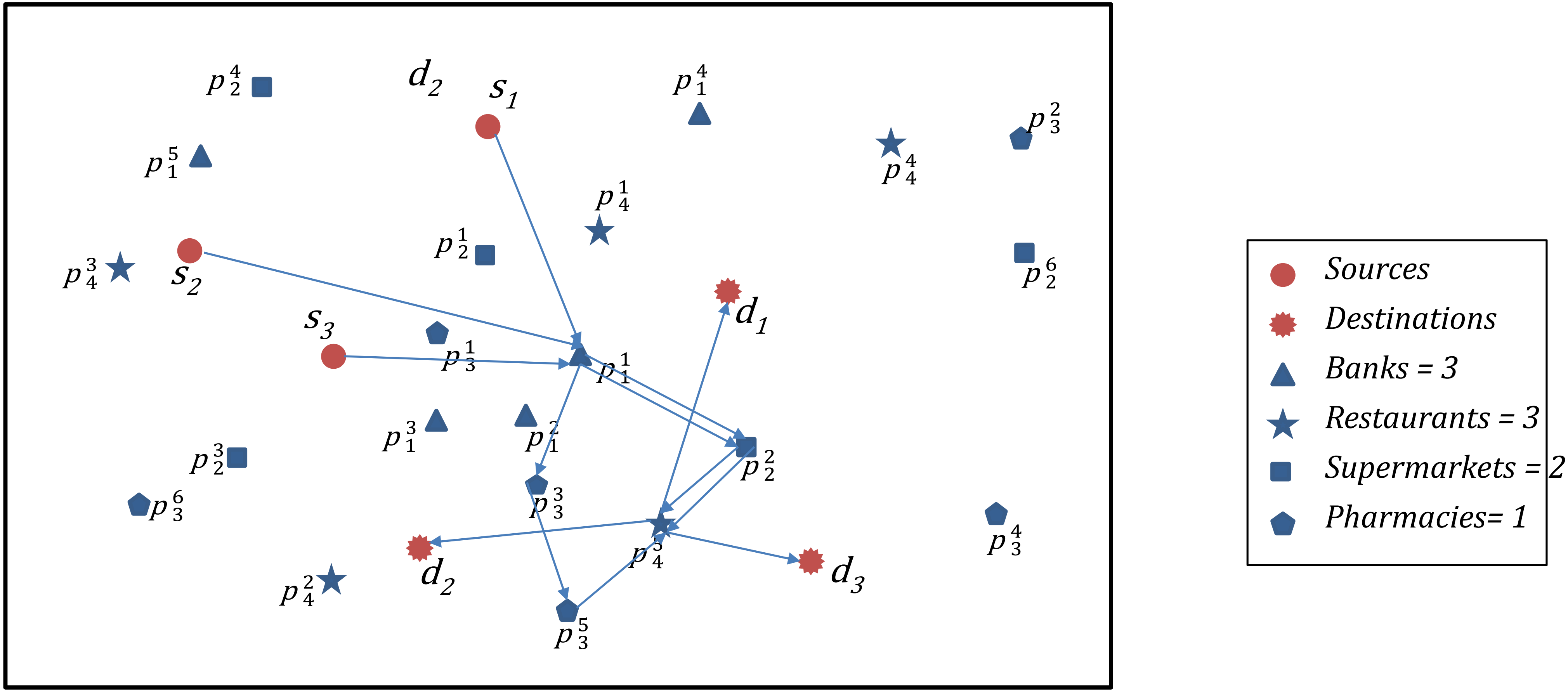


Generalized Group Trip Scheduling (GGTS) Queries

Generalized Group Trip Scheduling (GGTS) Queries



Generalized Group Trip Scheduling (GGTS) Queries

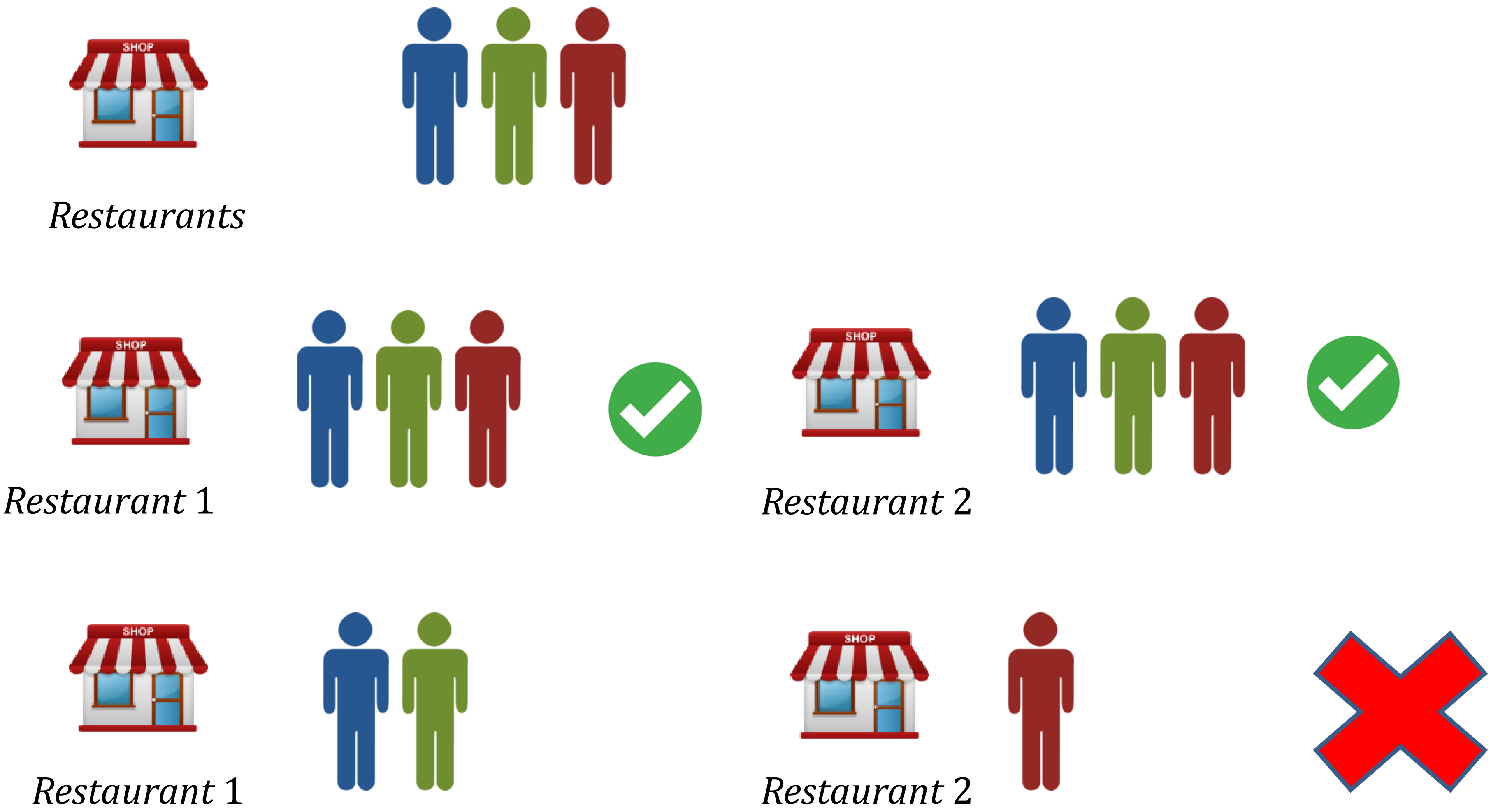


$s_1 \rightarrow p_1^1 \rightarrow p_2^2 \rightarrow p_4^5 \rightarrow d_1$

$s_2 \rightarrow p_1^1 \rightarrow p_2^2 \rightarrow p_4^5 \rightarrow d_2$

$s_3 \rightarrow p_1^1 \rightarrow p_3^3 \rightarrow p_4^5 \rightarrow d_3$

Generalized Group Trip Scheduling (GGTS) Queries



Generalized Group Trip Scheduling (GGTS) Queries

- Inputs:
  - Source locations of n users : $s_1, s_2, s_3, \ldots, s_n$
  - Destination locations of n users : $d_1, d_2, d_3, \ldots, d_n$
  - m POI types : $c_1, c_2, c_3, \ldots, c_m$
  - Expected no of people to visit those POI types : $n_1, n_2, n_3, \ldots, n_m$ and $1 \leq n_i \leq n$, n = no of group members

- Output:
  - N trips for n no of users, $T_1, T_2, T_3, \ldots, T_n$

    Here, $T_i = \{ p_1, p_2, p_3, \ldots, p_k \}$: A set of POIs and $| T_i | = 0\text{-}m$

- Objective:
  - Minimize the Trip Overhead Distance for the group.

Generalized Group Trip Scheduling (GGTS) Queries

- Conditions:

  - Each user may visit (0-m) number of of POI types.

  - When a POI type c_i is expected to be visited by n_i number of users, where $n_i > 1$ then those n_i users will visit the same POI location p^j_i of that type c_i .

  - $T_1 \cup T_2 \cup T_3 \ldots \cup T_n = \{ p_1, p_2, p_3, \ldots, p_m \}$ and $| T_1 \cup T_2 \cup T_3 \ldots \cup T_n | = m$

  - $T_i \cap T_j = \emptyset$ or $T_i \cap T_j = \{ p_1, p_2, p_3, \ldots, p_k \}$

  - $\sum_{i=1}^{n} T_n = n_1 + n_2 + n_3 + \ldots + n_m$

Related Works

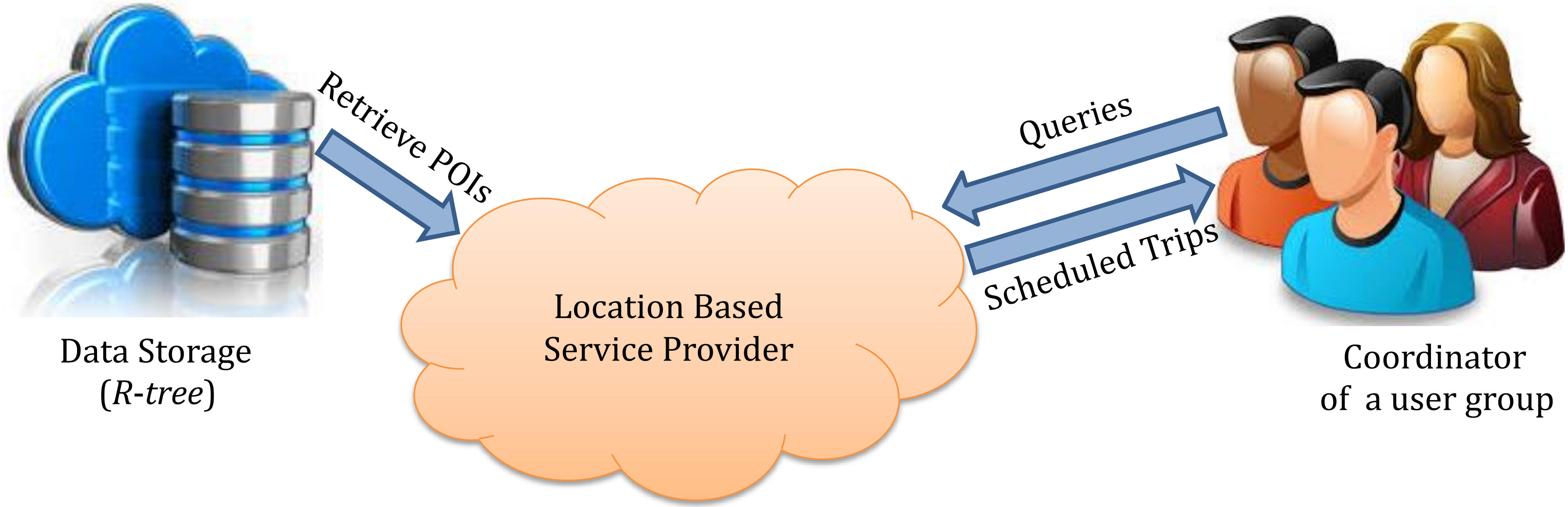
- Algorithms for group trip planning queries[4]
  - Consider a group of users to plan best possible trips.
  - In a GTP query, all users go through a same set of POIs.

- Algorithms for group trip scheduling queries[5].
  - Consider a group of users to plan best possible trips so that one POI type is visited by one and only one user and no POI type is visited by two different member.

We propose the first approach to evaluate GGTS queries in spatial databases

[1] Li *et al.*, 2005, [2] Chen *et al.*, 2008 , [3] Sharifzadeh *et al.*, 2008
[4] Hashem *et al.*, 2013 [5] Jahan *et al.*, 2017

System Architecture Overview



- Queries :
  - *n* predefined source-destination pairs , *m* specific POI types and expected no of people to visit that certain POI type.
- Data Storage :
  - POI information is indexed using *R-tree*
- Location Based Service Provider :
  - Retrieve POIs, compute and schedule trips and return the resultant trips to the group coordinator

Research Challenge

- Finding appropriate set of POIs from Large POI Database.
  - California data set: About 87,635 POIs of 63 different POI types, on average 1300 POIs for each type
  - If the no of required POI type is 6, the number of possible set of POIs :

$$\prod_{j=1}^{m}(No.of\ POIs\ for\ j\ th\ POI\ type)\ ;\ \ where\ ,m = No.of\ required\ POI\ Type$$

$$= 1300 * 1300 * \ldots * 1300$$

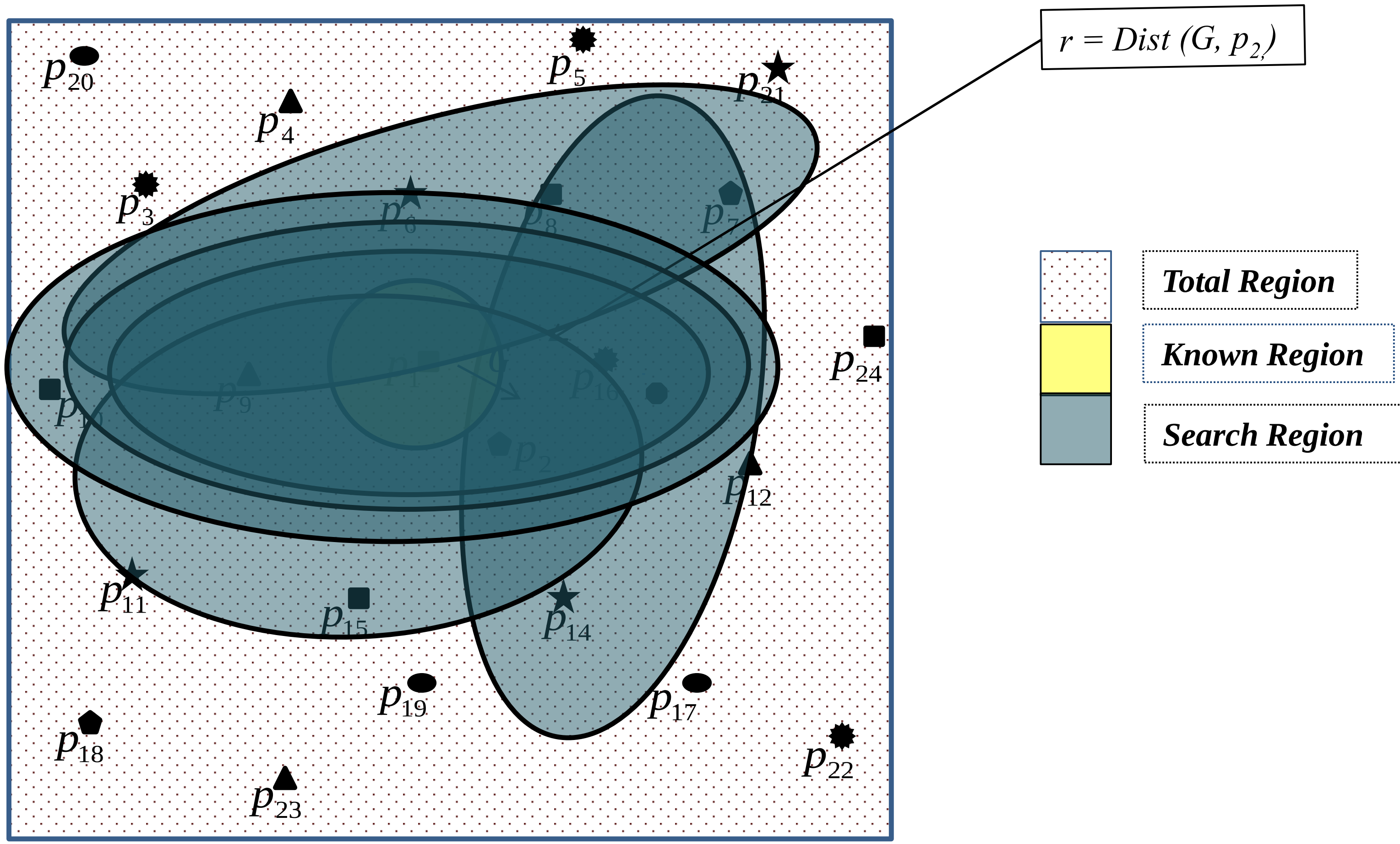
$$= (1300)^6 = \ 4.83\ e^{+18}$$

Huge amount !!!!!!

- Efficiency of GGTS queries depends on
  - Refinement of the search space
  - Scheduling of the Trips
    - Total Trip Distance is minimum.
    - When more than one member is expected to visit a POI type they visit the same POI location of that type.

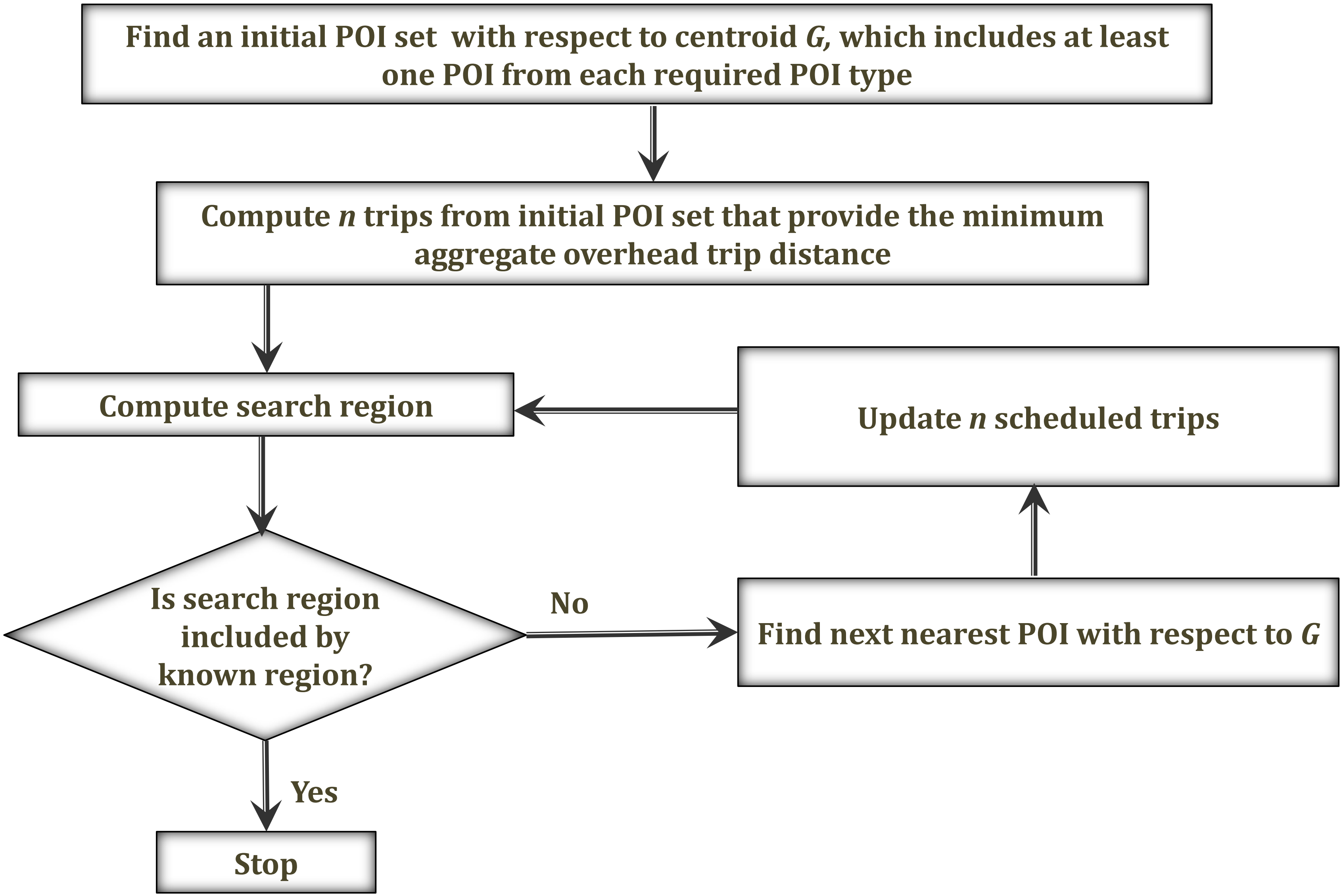
Thesis Goal

- Introduction of a *GGTS* Query
- Refinement of the Search Space
- Scheduling of the Trips
- Experimentation with Real Data Set



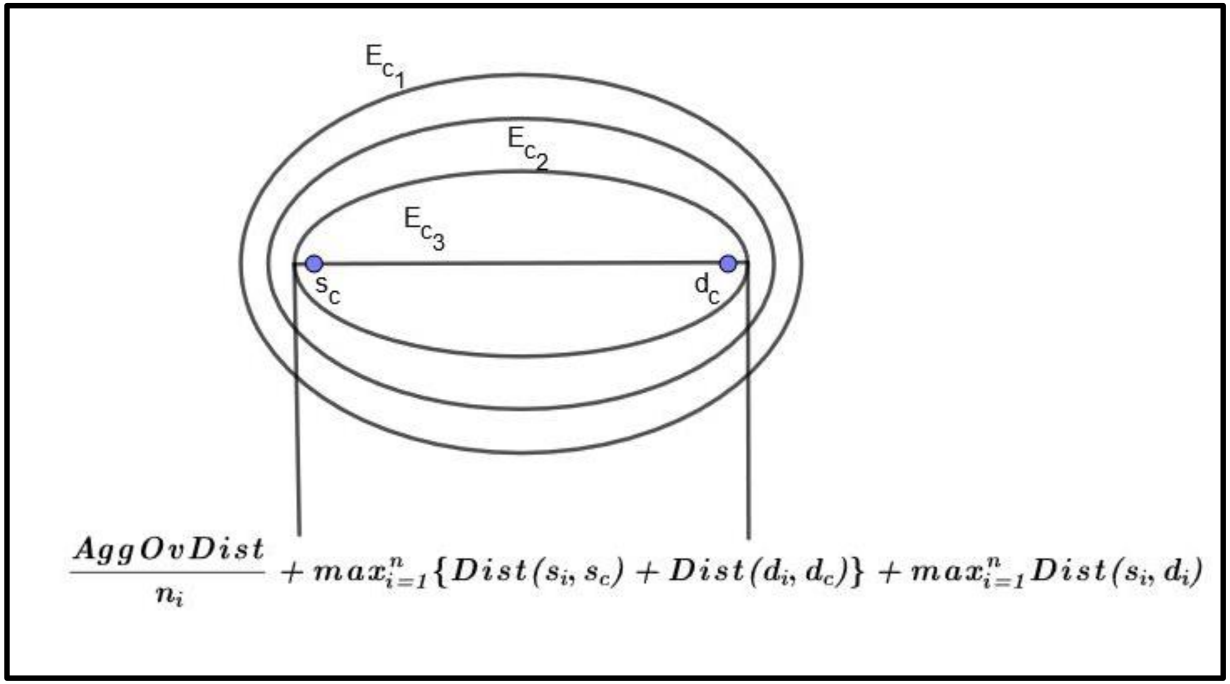
- **Known Region :** The region which has been already explored
- **Search Region :** The region which we need to explore for the optimal solution

Optimal Approach



First Refinement Technique (Type Based)

- $TripDist_i$: The current minimum trip distance of user u_i among the scheduled trips



$$S_c = \frac{S_1 + S_2 + \ldots + S_n}{n}$$

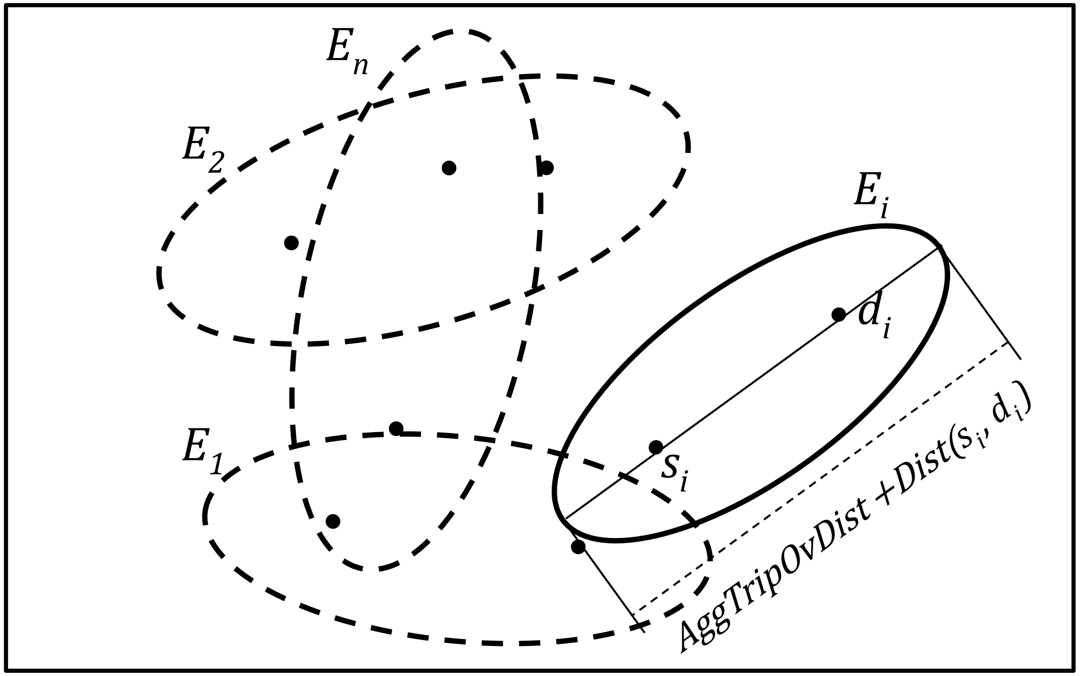
$$d_c = \frac{d_1 + d_2 + \ldots + d_n}{n}$$

Aggregate trip overhead distance : $\sum_{i=1}^{n} (TripDist_i) - \sum_{i=1}^{n} Dist(s_i, d_i)$

- m ellipses : $E_{c_1}, E_{c_2}, \ldots, E_{c_m}$
  - The foci of Ellipse $\boldsymbol{E_{c_m}}$ are at $\boldsymbol{S_c}$ and $\boldsymbol{d_c}$
  - Major axis of the ellipse : $\boldsymbol{\frac{AggTripOvDist}{n_i} + max_{i=1}^{n}\{Dist(s_i, s_c) + Dist(d_i, d_c)\} + max_{i=1}^{n} Dist(s_i, d_i)}$

Refinement Technique (User Based)

- $TripDist_i$: The current minimum trip distance of user u_i among the scheduled trips
- $AggTripOvDist$: The current minimum aggregate trip overhead distance of the group



Aggregate trip overhead distance:
$$\sum_{i=1}^{n} (TripDist_i) - \sum_{i=1}^{n} Dist\,(s_i, d_i)$$

- n ellipses, $E_1, E_2, \dots, E_n$
  - The foci of Ellipse E_i are at $\boldsymbol{s_i}$ and $\boldsymbol{d_i}$
  - Major axis of the ellipse : $\boldsymbol{AggTripOvDist + Dist\,(s_i, d_i)}$

Combining Refinement techniques

- From *Type* based refinement techniques we find the search region for a POI type c_i: $\boldsymbol{E_{c_i}}$
- From *User* based refinement techniques we find the search region:

$$\boldsymbol{E_1} \cup \boldsymbol{E_2} \cup \ldots \cup \boldsymbol{E_n}$$

❑ For any POI type c_i the search region :

$$\boldsymbol{E_{c_i}} \cap (\boldsymbol{E_1} \cup \boldsymbol{E_2} \cup \ldots \cup \boldsymbol{E_n})$$

Optimal Trip Scheduling

- Constructing a combination table
  - Each row contains a possible combination of users visiting a certain sequence of POI types satisfying all the conditions

u_1 : $c_1 \rightarrow c_2 \rightarrow c_4$

u_2 : $c_1 \rightarrow c_2 \rightarrow c_4$

u_3 : $c_1 \rightarrow c_3 \rightarrow c_4$

- Constructing an index table
  - Each row contains a sequence of POI types a user might visit satisfying all the conditions and the corresponding combination number that contains such a sequence

$c_1 \rightarrow c_4$

$c_1 \rightarrow c_2 \rightarrow c_4$

$c_1 \rightarrow c_3 \rightarrow c_4$

$c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow c_4$

Optimal Trip Scheduling

Construct the combination and index table

↓

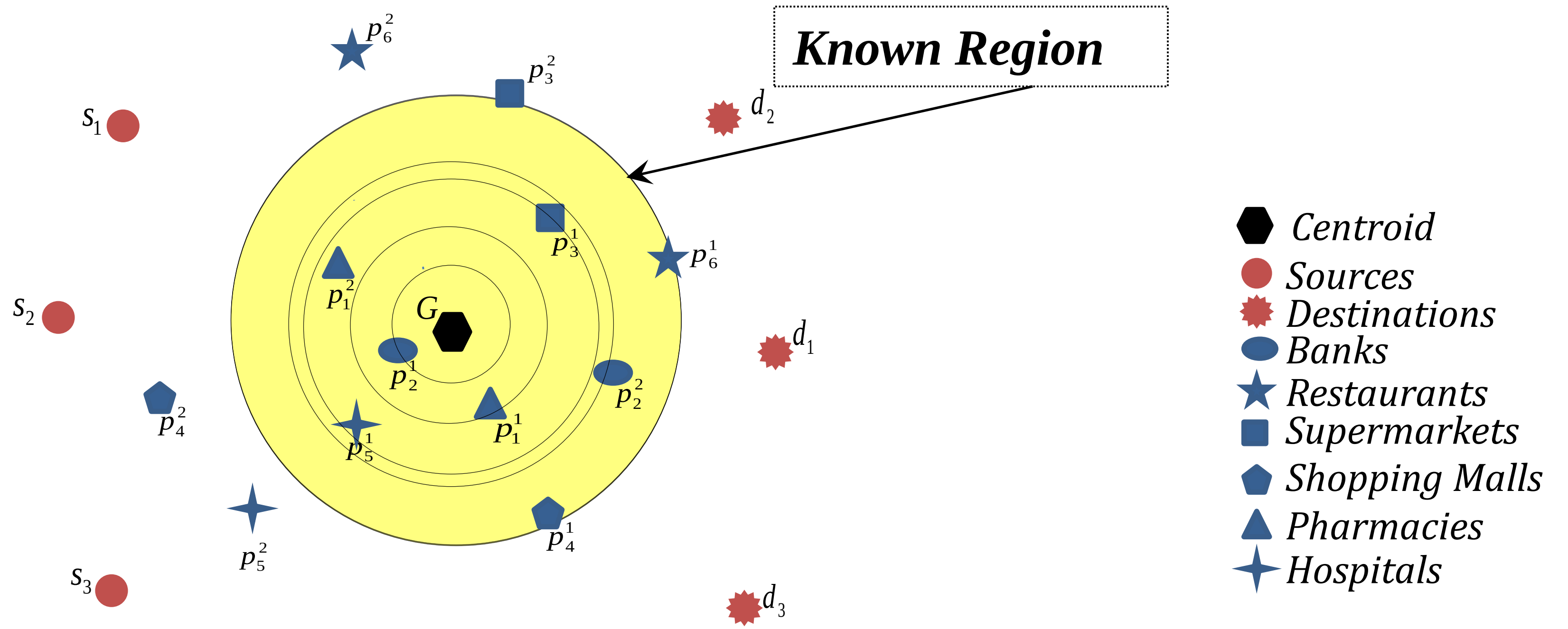
Calculate the trip distances for each combination with *initial POI set*

↓

Check with combinations in the combination table after the retrieval of a POI *p* required POI type to update the trip distances

Retrieve Initial Set of POIs

- The POIs are stored using *R*-tree* in the database
- Retrieve nearest POI from *G* incrementally
  - *G*: Geometric centroid of all group members' source and destination locations.



Initial POI set:
$\{p_1^1, p_1^2, p_2^1, p_3^1, p_4^1\}$

Figure: Initial known region after finding at least one POI of all required POI types

Compute Initial Scheduled Trips

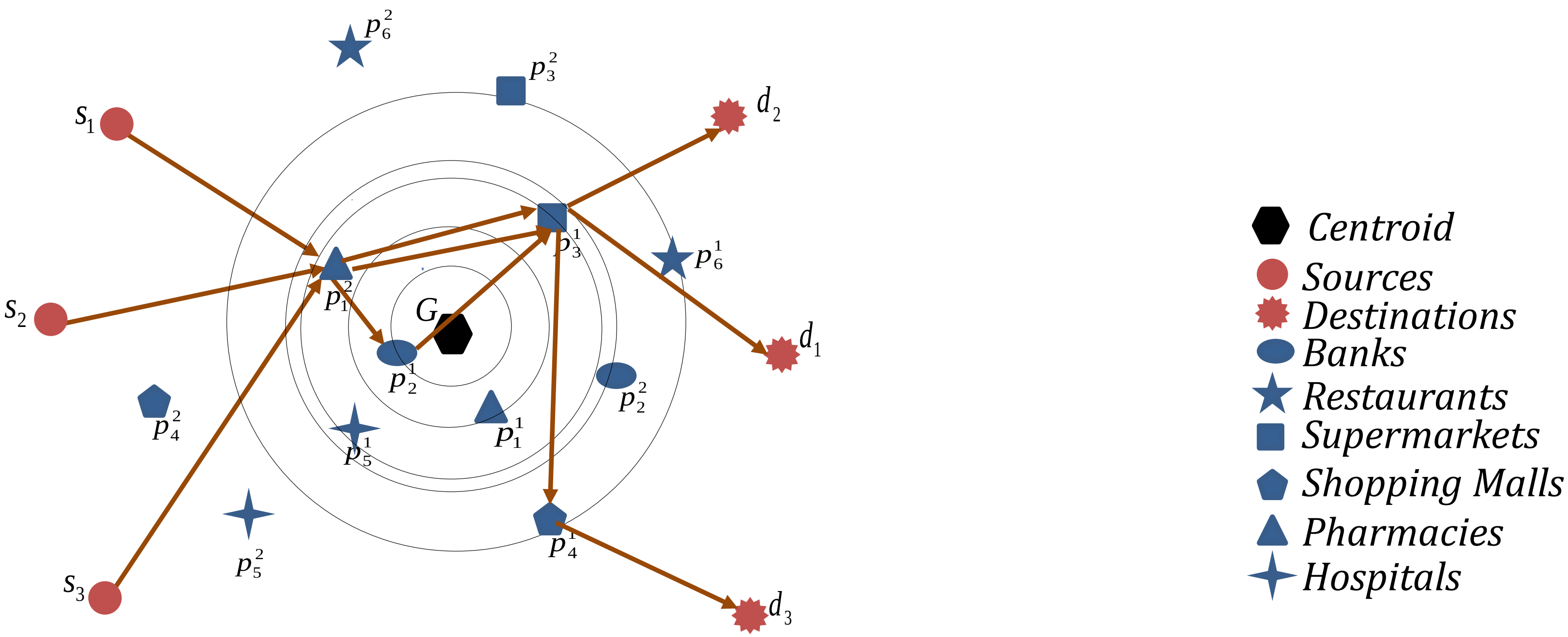


Figure: Initial scheduled optimal routes

$$s_1 \rightarrow p_1^2 \rightarrow p_2^1 \rightarrow p_3^1 \rightarrow d_1$$

$$s_2 \rightarrow p_1^2 \rightarrow p_3^1 \rightarrow d_2$$

$$s_3 \rightarrow p_1^2 \rightarrow p_3^1 \rightarrow p_4^1 \rightarrow d_3$$

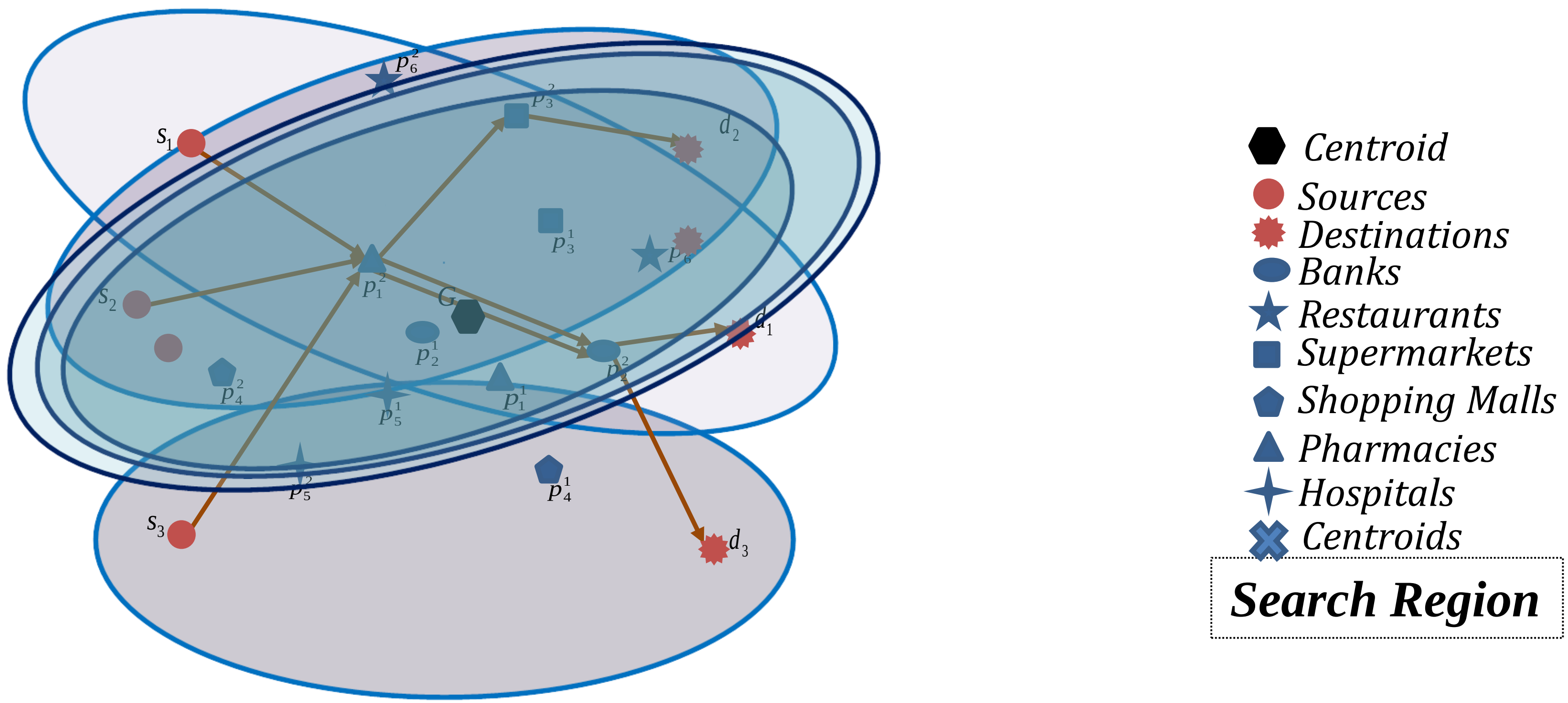
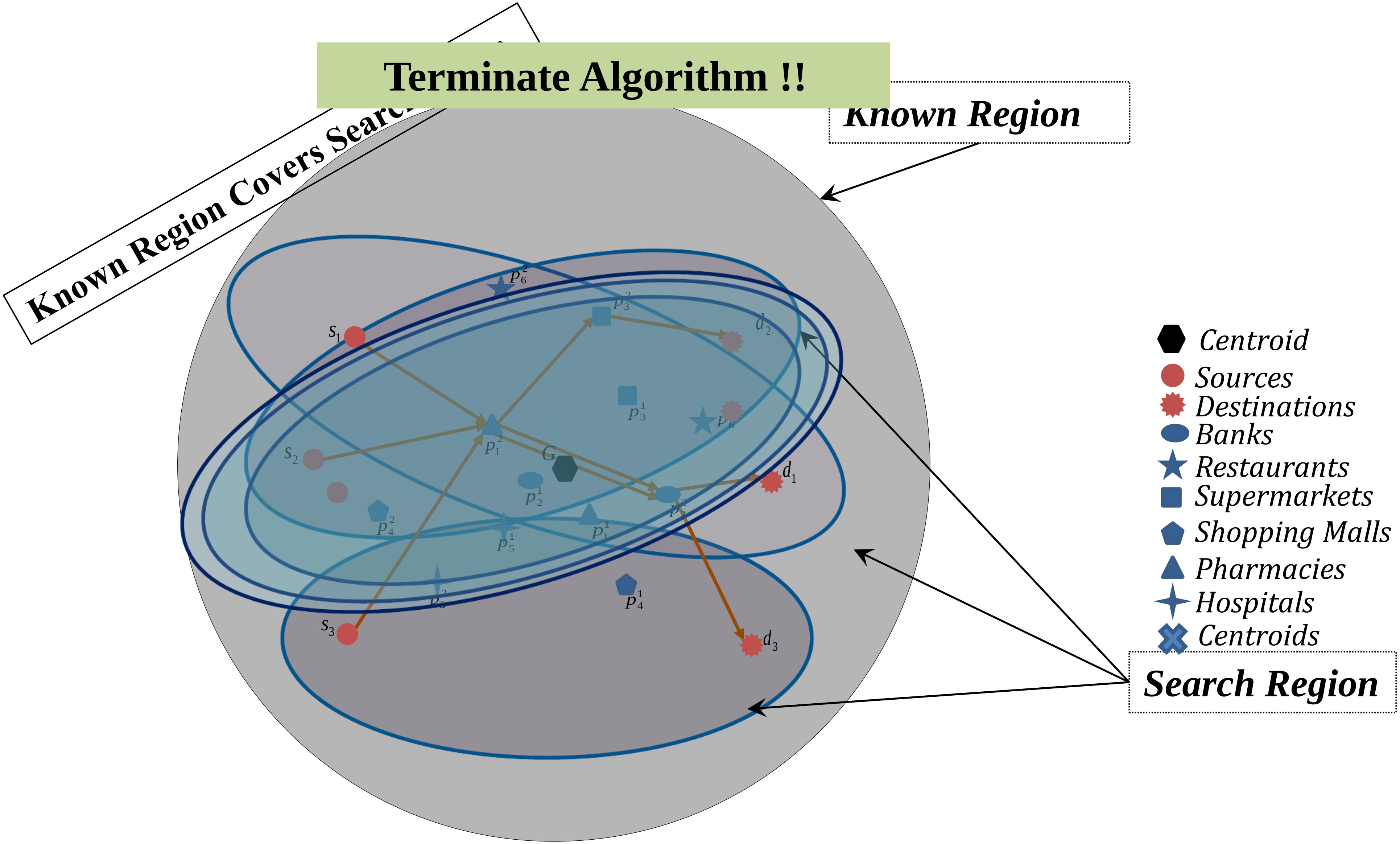


Figure: Updated Search Region

Check Terminating Condition



Heuristic Approaches (TSH-GGTS)

Construct the combination and index table

↓

Calculate the trip distances for each combination with *initial POI set*

↓

Update the combination table with best *t* combination for which trip distance is minimum

↓

Check with combination in the updated combination table after the retrieval of a POI *p* of required POI type to update the best ten trip distances

Heuristic Approaches (SRH-GGTS)

Find at least one POI of all required POI types for each member

↓

Calculate the trip distances for each pattern with *candidate POI set*

↓

Check with patterns in the pattern table after the retrieval of a POI *p* of required POI type to update the trip distances

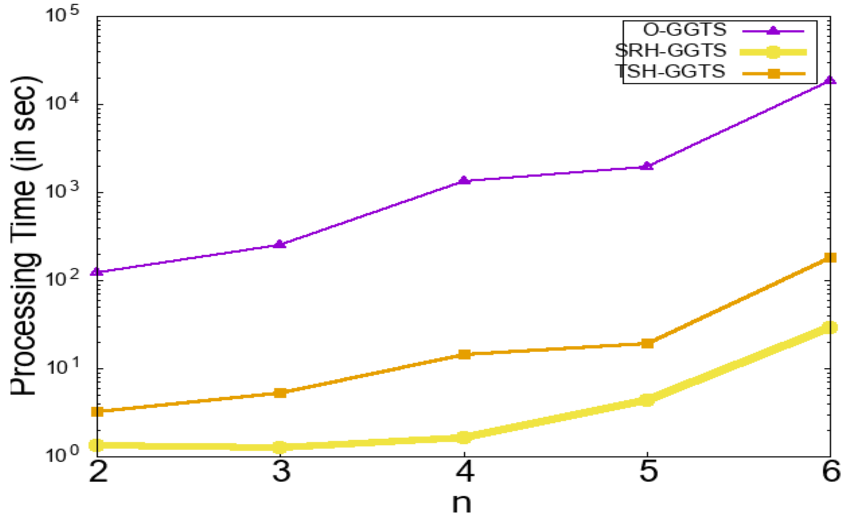
Experiment Setup

Data Set	• Real data set [California dataset] - 87635 POIs with 63 POI types - Road network : 21048 nodes with 21693 edge
Parameters	• The number of required POI types • The number of users of a group • The query area
Measurement terms	• Processing time • I/O access

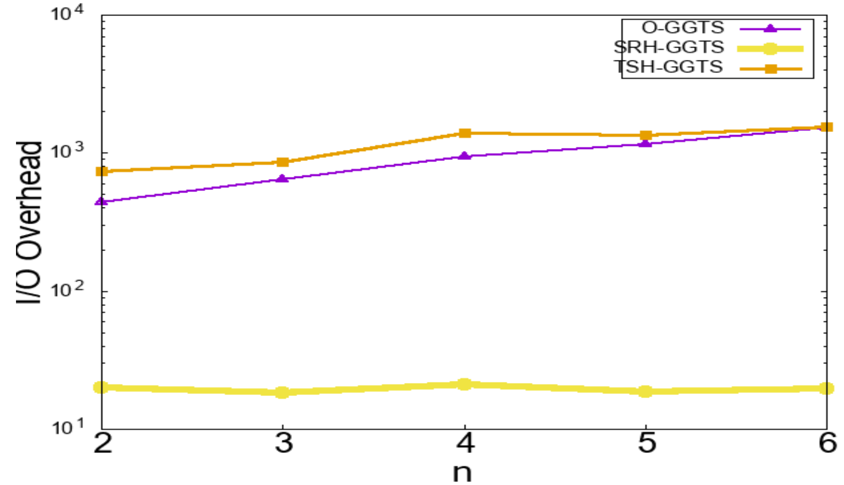
Parameter Settings

Parameter	Range	Default
Group size (n)	2, 3, 4, 5, 6	3
Number of POI types (m)	2, 3, 4, 5, 6	3
Query area (A) (in sq. units)	50×50, 100×100, 150×150, 200×200, 250×250, 300×300	100×100

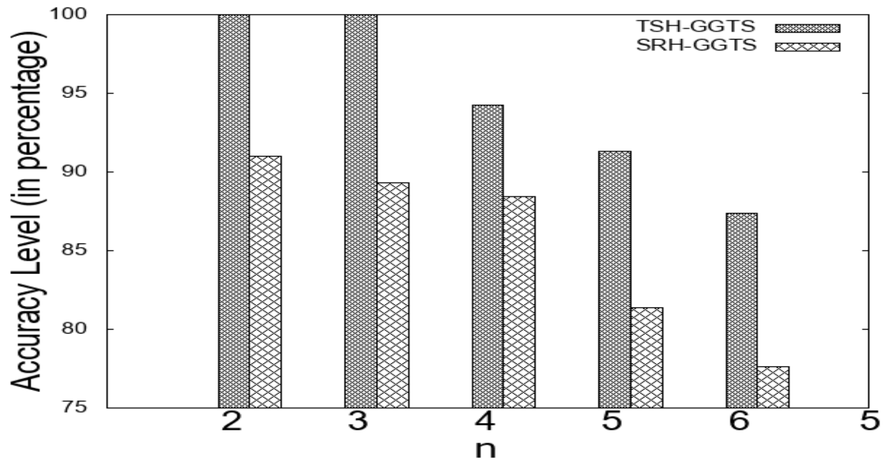
Performance Analysis (Effect of Group Size *n*)



Processing time vs Group size

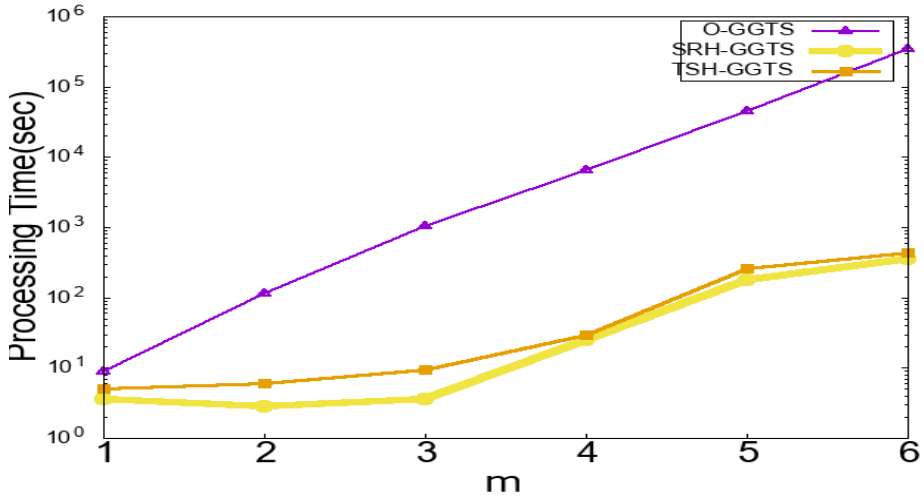


I/O Overhead vs Group size

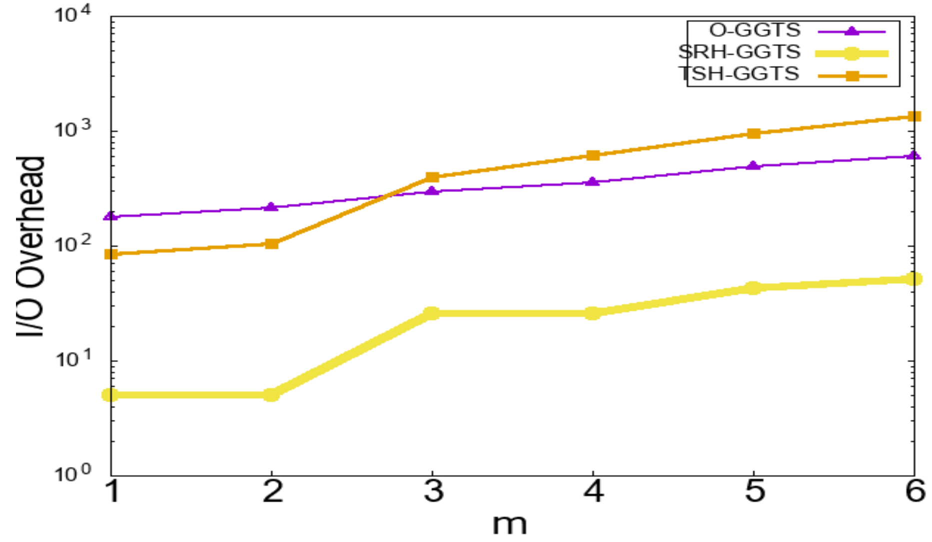


Accuracy level of heuristic approaches with varying ***n***

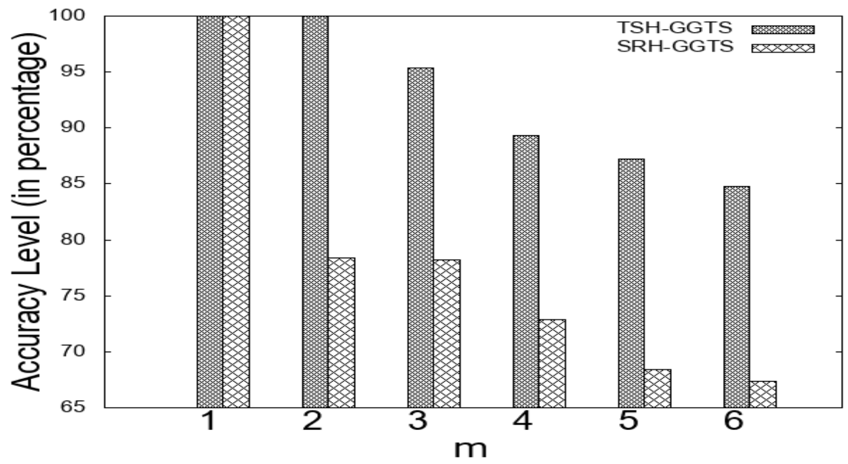
Performance Analysis (Effect of number of POI types *m*)



Processing time vs No of POI types

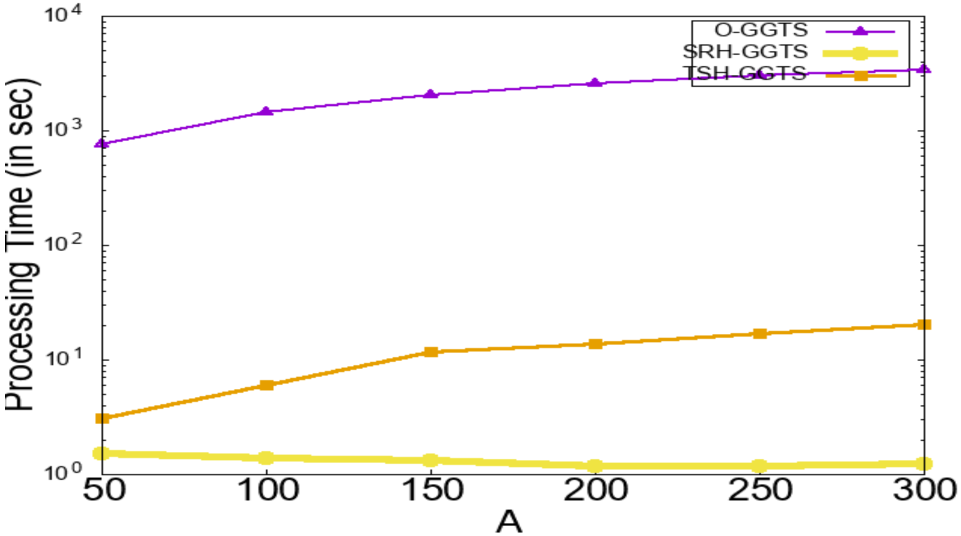


I/O Overhead vs No of POI types

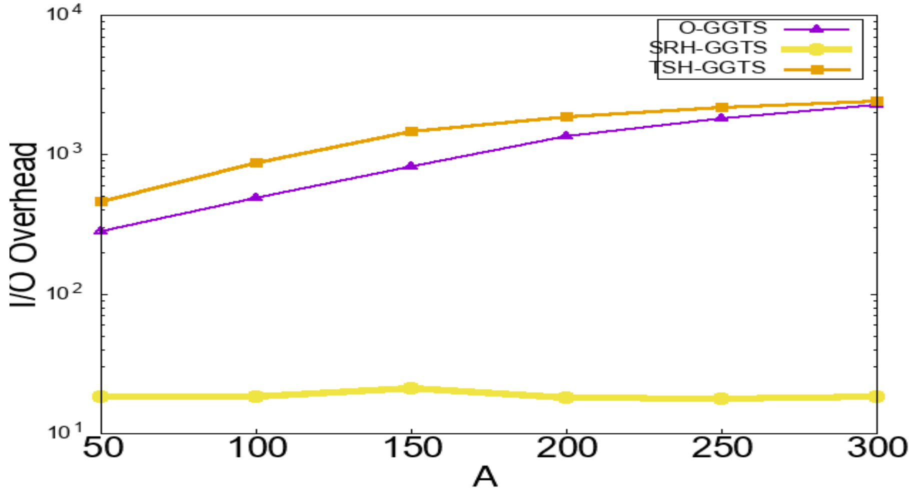


Accuracy level of heuristic approaches with varying *no of POI types*

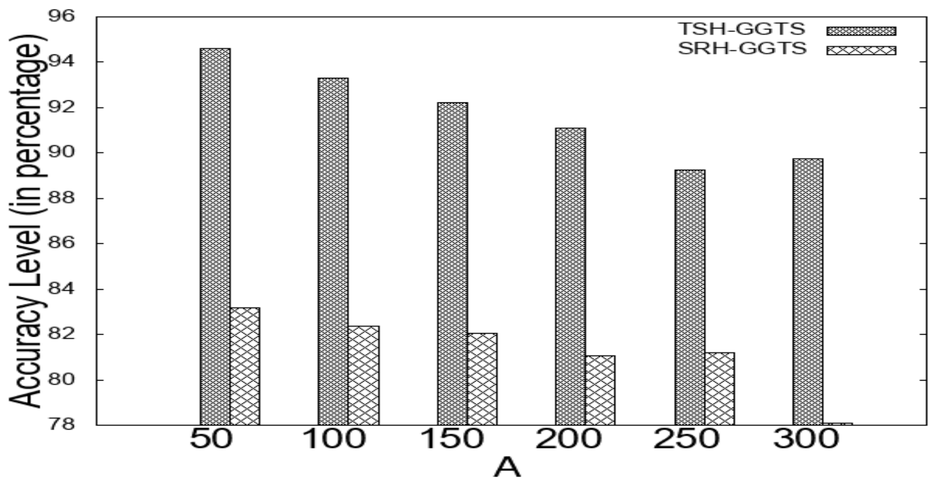
Performance Analysis (Effect of Query Area *A*)



Processing time vs Area size



I/O Overhead vs Area size



Accuracy level of heuristic approaches with varying *area size*

Optimal approach:

- Small parameter settings

Heuristic approaches:

- Large parameter settings

TSH-GGTS:

- Processing time 23299.64 times less than Optimal-GGTS
- Accuracy on average 92.95%

SRH-GGTS:

- Processing time 23330.07 times less than Optimal-GGTS
- Accuracy on average 77.26 %

Conclusions

- Introduce the first Generalized Group Trip Scheduling (GGTS) Queries in the spatial database
- Develop algorithms to compute GGTS queries almost accurately and efficiently
- Perform extensive experiments using real and data sets

References

- Li F., Cheng D., Hadjieleftheriou M., Kollios G. and Teng S-H, On trip planning queries in spatial databases, In SSTD, 273–290, 2005.
- Chen H., Ku W. S., Sun M. T. and Zimmermann R., The multi-rule partial sequenced route query, In GIS, 10, 2008.
- Sharifzadeh M., Kolahdouzan M. and Shahabi C., The optimal sequenced route query, In VLDB journal, 17(4): 765–787, 2008.
- Ohsawa Y., Htoo H., Sonehara N. and Sakauchi M., Sequenced route query in road network distance based on incremental euclidean restriction, In DEXA, 484-491, 2012.
- Malviya N., Madden S. and Bhattacharya A., A continuous query system for dynamic route planning, In ICDE, pages 792-803, 2011.
- Geisberger R., Kobitzsch M. and Sanders P., Route planning with flexible objective functions. In ALENEX, 124-137, 2010.
- Geisberger R., Rice M. N., Sanders P. and Tsotras V. J., Route planning with flexible edge restrictions, In JEA, 1.2:1.1-1.2:1.20, 2012.

References

- Hashem T., Hashem T., Ali M. E. and Kulik L., Group Trip Planning Queries in Spatial Databases, In SSTD, 259-276, 2013.
- Mahmud H., Amin A. M., Ali M. E., Hashem T. and Nutanong S., A group based approach for path queries in road networks, In SSTD, 367-385, 2013.
- Xu Z. and Rodrigues B., A 3/2-approximation Algorithm for Multiple Depot Multiple Traveling Salesman Problem, In SWAT, 127-138, 2010.
- Gutin G. and Karapetyan D., A memetic algorithm for the generalized traveling salesman problem, In Natural Computing, 9(1):47-60, 2010.
- Pop P. C., Matei O. and Sabo C., A new approach for solving the generalized traveling salesman problem, In Hybrid Metaheuristics, 62-72, 2010.

Thank you

Questions?