A Combination of Cell-Based and Graph-Based Techniques for Clustering Big Datasets

Presenter: Mr. Duong Van Hieu, a PhD student at KMUTNB
Email: dvhieu@gmail.com
Advisor: Associate Professor Dr.Phayung Meesad

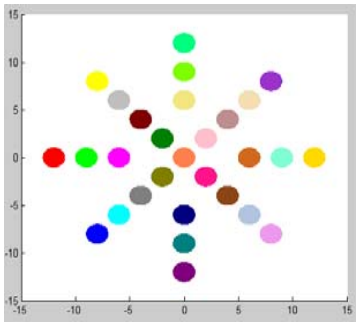
Agenda

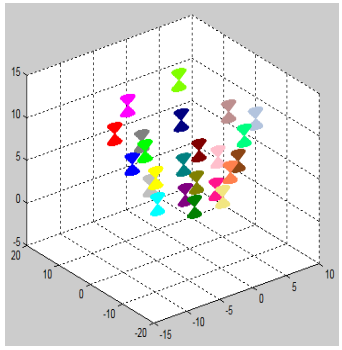
1. Problem Setting and related literatures
2. Proposed Solutions
   - Solution 1: Outlier detection
   - Solution 2: Clustering

1. Problem Setting and literatures

1.1. Problems

- To cluster a big dataset $\mathbf{X}=\{x_{ij}\}$ with $i=1,\ldots, N$ and $j=1,\ldots, D$ into a number of clusters satisfying
  1. Outliers should be identified and eliminated
  2. An optimal cluster number should be evaluated automatically
  3. All processes should be done on a personal computer
  4. Proposed solutions should outperform some previous solutions



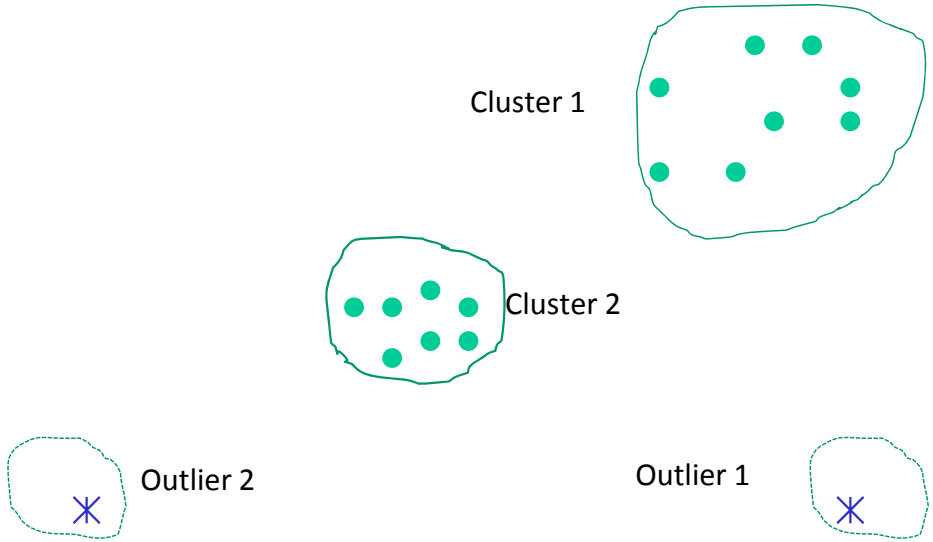


1. Problem Setting and literatures

1.2. Literatures:

Outliers

- *"An outlier is an observation which deviates so much from the other observations as to arouse suspicious that it was generated by a different mechanism"*



1. Problem Setting and literatures

1.2. Literatures:

Outlier Detection

1. Objects can be marked as outliers or non-outliers.
2. Objects can be assigned outlier scores and sorted in a descending order based on scores => **the top *k* objects are considered as *k* outliers**

Object IDs	Outlier Scores
122	1,429.75
18	211.79
207	120.34
177	96.93
192	82.82
91	70.01
…	…

1. Problem Setting and literatures

1.2. Literatures:

Outlier Detection Algorithms

1. **LOF** and **COF** algorithms may produce poor results when outliers located between low and high density cluster.
2. **INFLO** algorithm may produce incorrect outliers because it assumes that all neighbors of an object have the same density
3. Results of the **RBDA**, **ODMR**, **ODMRD** algorithms might be negatively affected by local irregularities of datasets

…

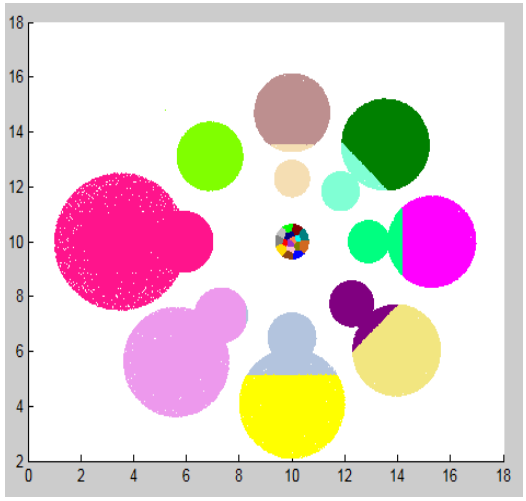
4. **The DROS algorithm can produce very high precision results compared to the previous algorithms. However, executing time is too long.**

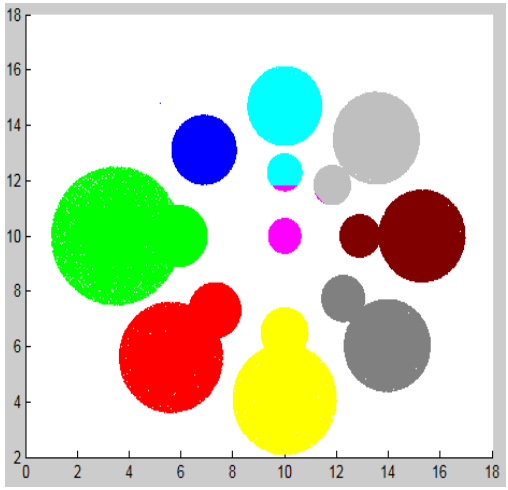
1. Problem Setting and literatures

1.2. Literatures:

Clustering

- To estimate an optimal cluster number K, initialize K centers
- To cluster objects into K groups and obtain K final centers





1. Problem Setting and literatures

1.3. Literatures:

Cluster Number Estimation Algorithms

1. **Group 1:** Similarity matrix, Decision-theoretic rough set, Graph-based K-Means, MST-based, etc. methods require a large amount of memory
2. **Group 2:** Locally adaptive clustering, weighted gap statistic, etc. methods are time consumption when N is a large number

   …

3. **The newly proposed Quantization error modeling method can produce very good results. However, it is not appropriate to work with very large datasets due to long executing time**

2. Proposed Solutions

Solution 1:

A Fast Outlier Detection Algorithm for Big Datasets

- To identify and eliminate *n* outliers from a big dataset **X** having *N* objects and *D* attributes.

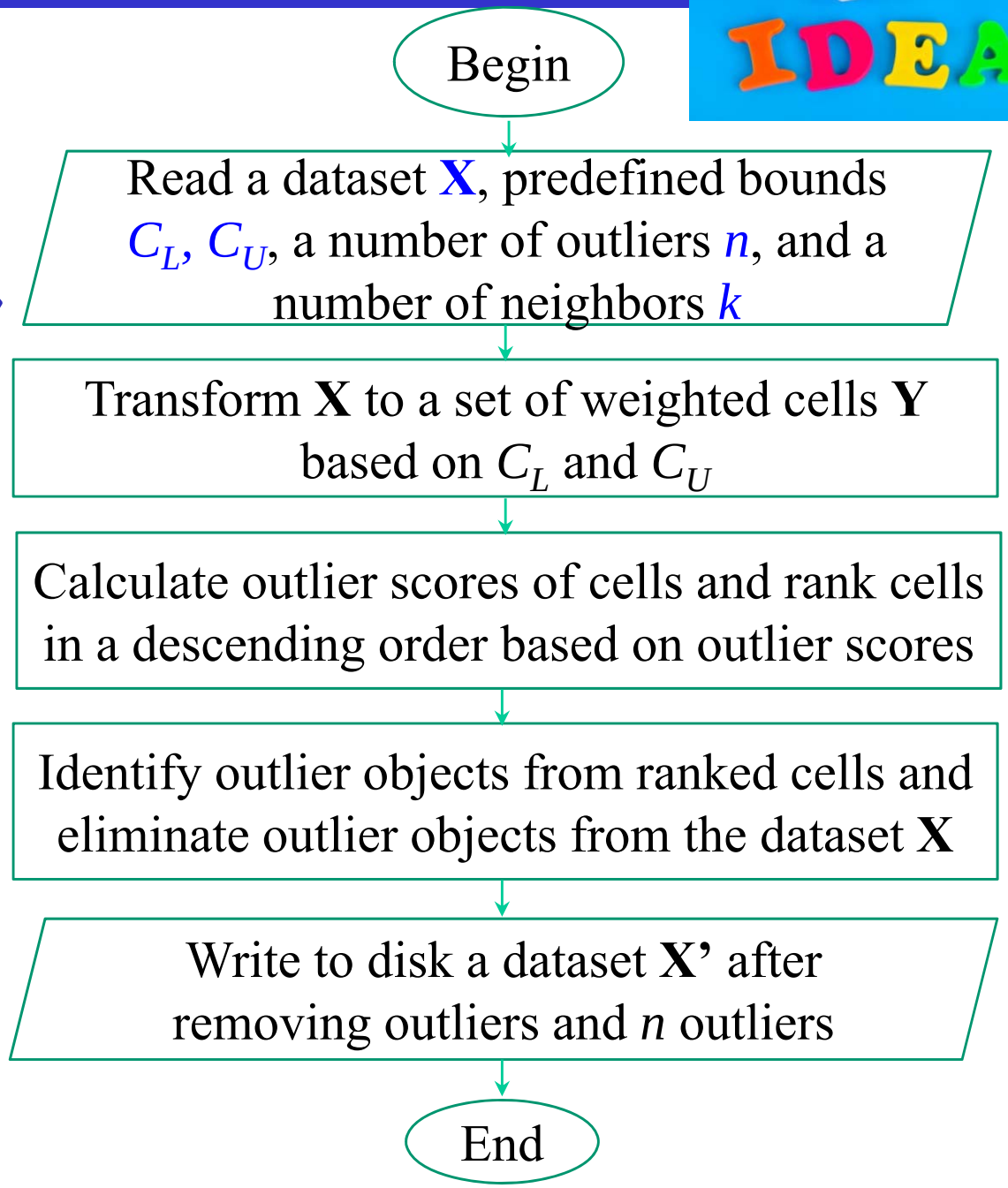
- **To produce almost the same results compared to the DROS algorithm**

- **To reduce a large part of executing time of the DROS algorithm**

Solution 1: Methodology

Cell-RDOS: A Fast Outlier Detection Algorithm for Big Datasets.

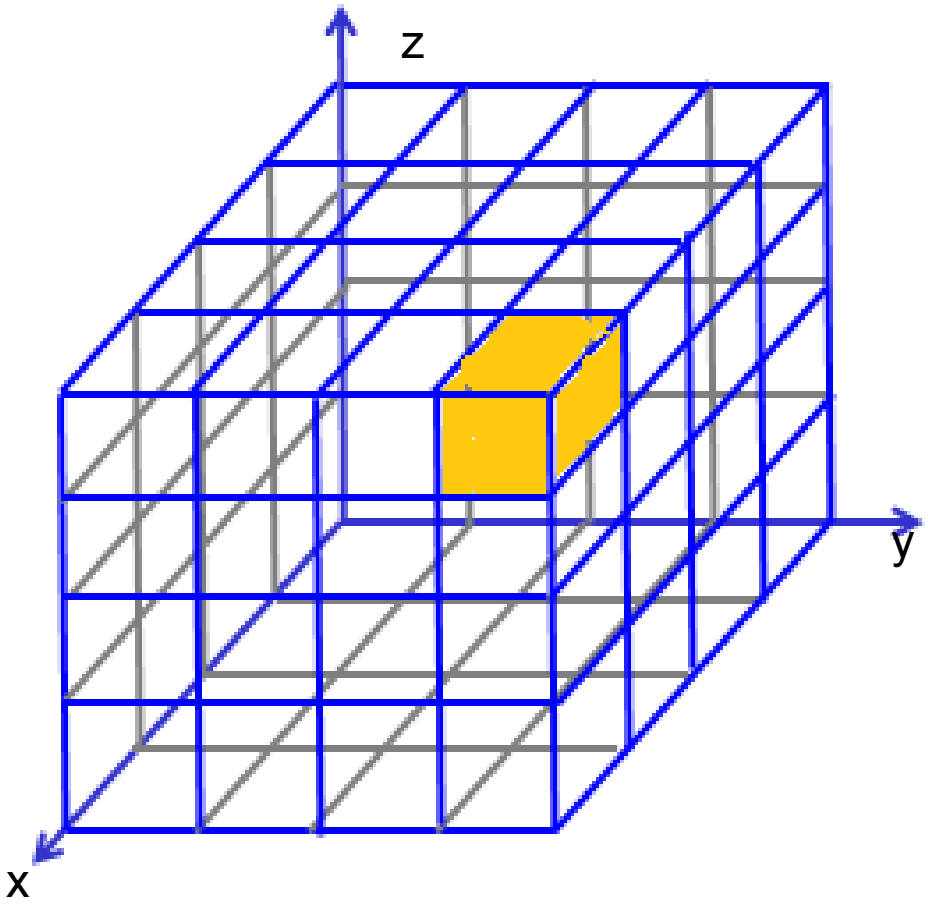
- **Accepted at IC²IT 2016 Conference**
- **An extended paper was accepted by IWBDA 2016 workshop**



Solution 1: Methodology

Step 1: To transform a big dataset to a small set of cells

- M is initialized by $1 + (C_L + C_U)^{1/D}$
- **X** is divided in to M^D cells
- Calculate a number of unempty cells, $C_{unempty}$



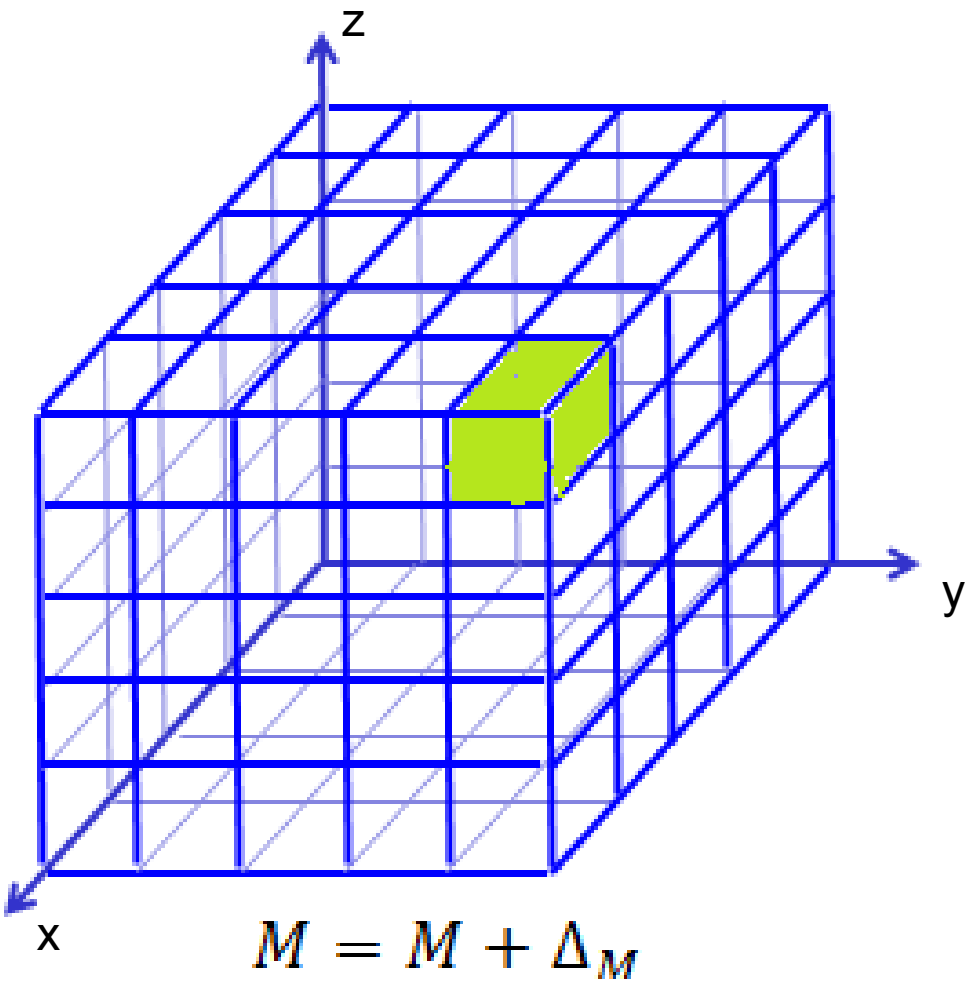
- Weight of an unempty cell is a number of objects contained in this cell

Solution 1: Methodology

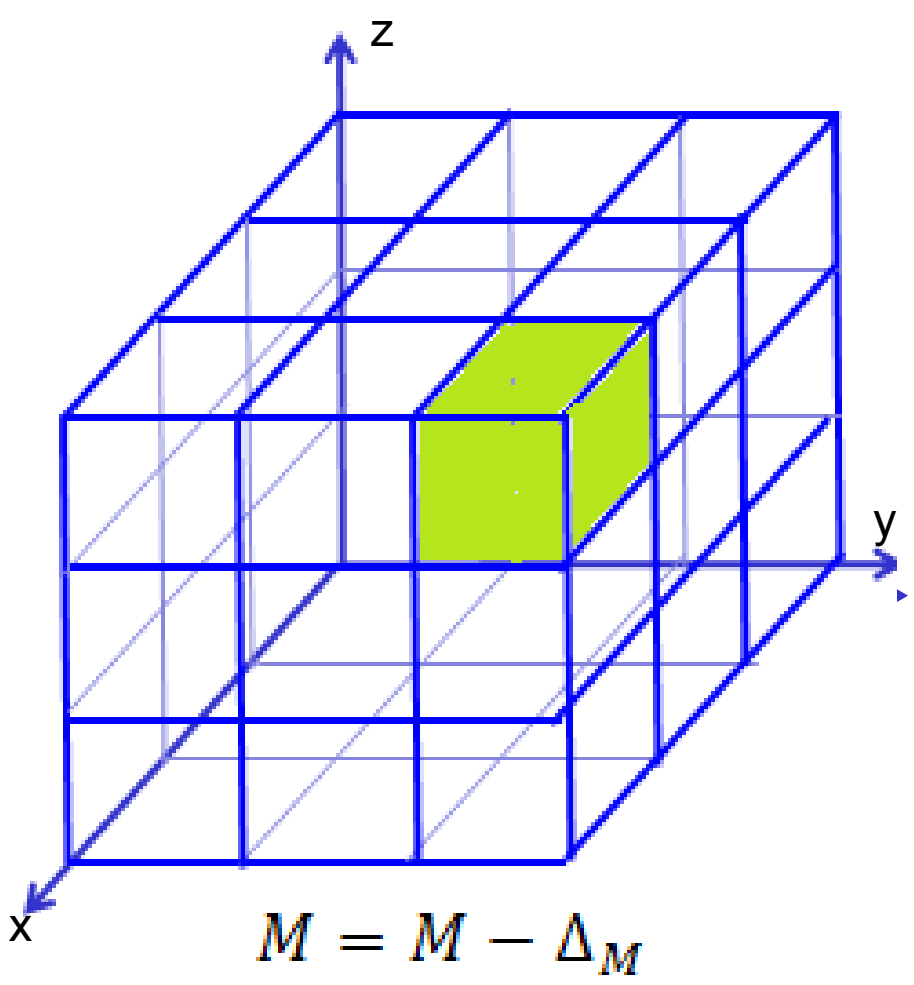
Step 1: To transform a dataset X to a set of weighted cells Y

- M is adjusted until a stopping condition is satisfied

$$(C_L \leq C_{unempty} \leq C_U) \text{ or } ((M^- = M^+) \text{ or } (n_{loop} \geq n_{max}))$$



$$M = M + \Delta_M$$

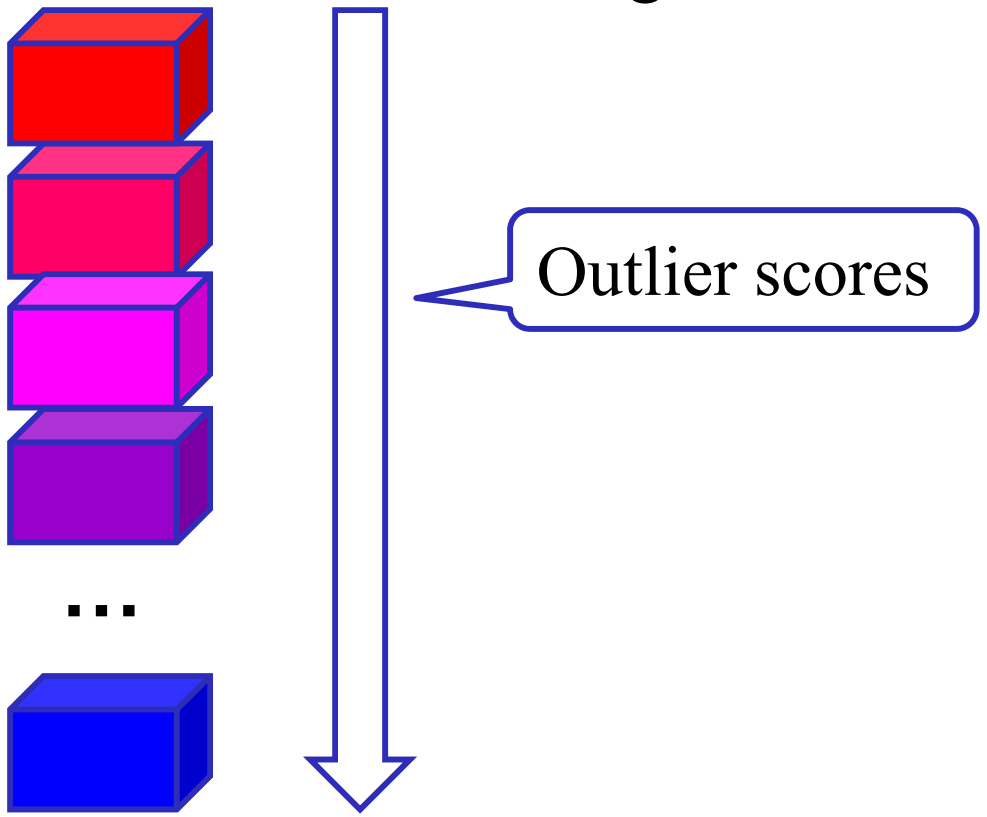


$$M = M - \Delta_M$$

Solution 1: Methodology

Step 2: To apply the RDOS algorithm on a set of cells

- Calculate outlier score of each cell
- Sort cells based on their scores in a descending order

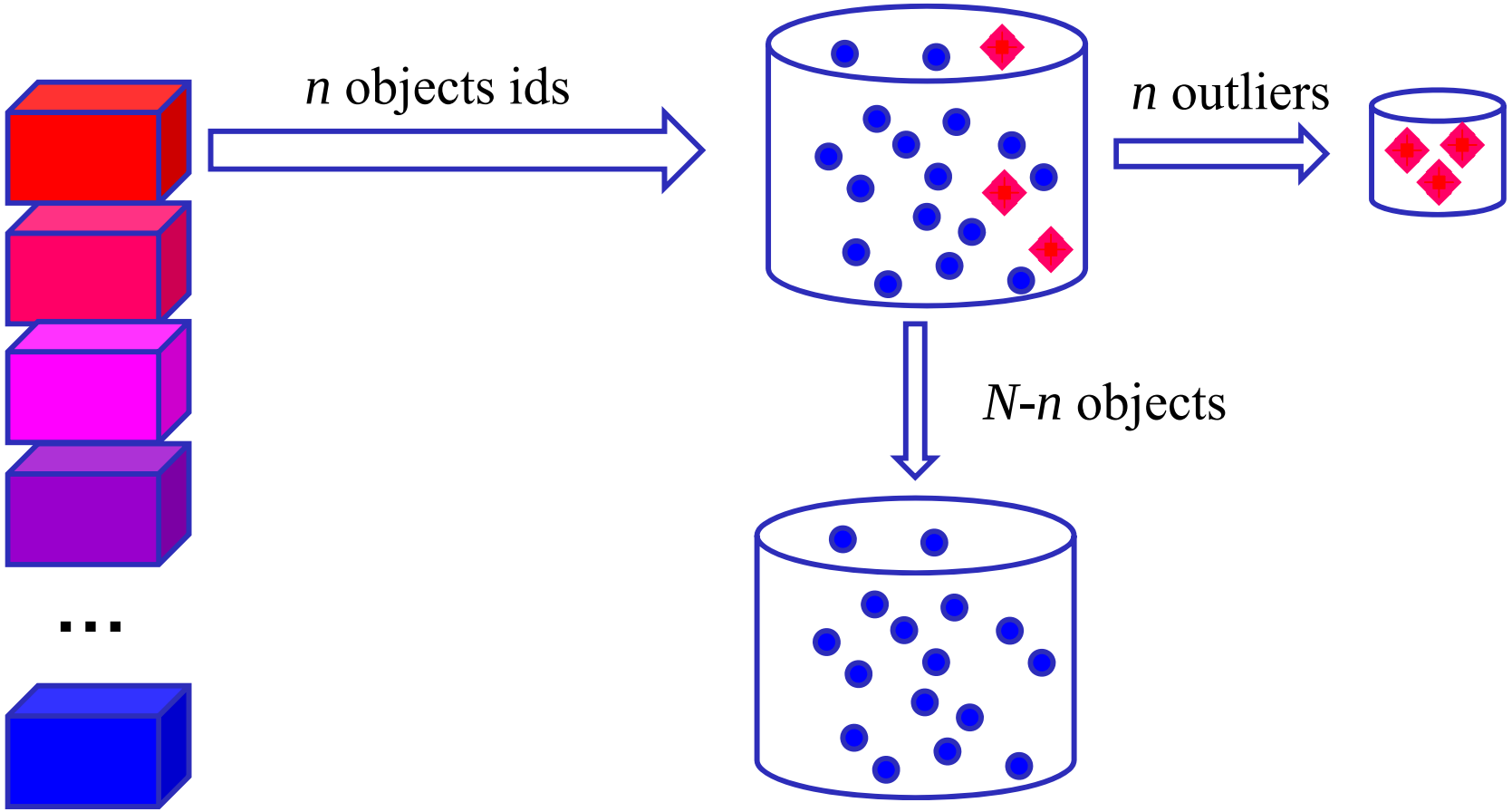


Solution 1: Methodology

Step 3: To obtain outlier objects from ranked cells

- Retrieve ids of *n* objects which are considered as outliers in the top ranked cells

- Eliminate *n* ouliers in the provided dataset



Solution 1: Results

- **Comparison of matching results** between the proposed method (Cell-DROS) and the DROS algorithms

Datasets	No. of objects	No. of detected outliers	No. of matching detected objects	Matching
Activity Recognition	162,501	10	9	90%
Person Activity	164,860	10	9	90%
TDrive Trajectory	176,424	7	6	85%
2010_09_10_jam_+40	584,760	10	10	100%
Dataset2D	800,059	100	100	100%

Solution 1: Results

- **Comparison of executing time** between the proposed method and the DROS algorithm

Datasets	No. of objects	Executing time (in minutes)		Reduced time
		RDOS	Cell-RDOS	
Activity Recognition	162,501	509.99	8.33	98.37%
Person Activity	164,860	533.93	4.46	99.16%
TDrive Trajectory	176,424	611.85	398.55	34.86%
2010_09_10_jam_+40	584,760	12,086.84	8.85	99.93%
Dataset2D	800,059	9,807.62	12.05	99.88%

Good

Solution 1: Conclusions

- When working with very large datasets:

1) **Accuracy level of the proposed Cell-RDOS algorithm matches the accuracy level of the RDOS algorithm,**
2) **The Cell-RDOS algorithm can reduce up to 99% of executing time compared to the RDOS algorithm.**

Solution 2:

A Cell-MST-Based Clustering Algoritm

- To estimate an optimal K, initialize centroids, do clustering process on a personal computer with limited memory

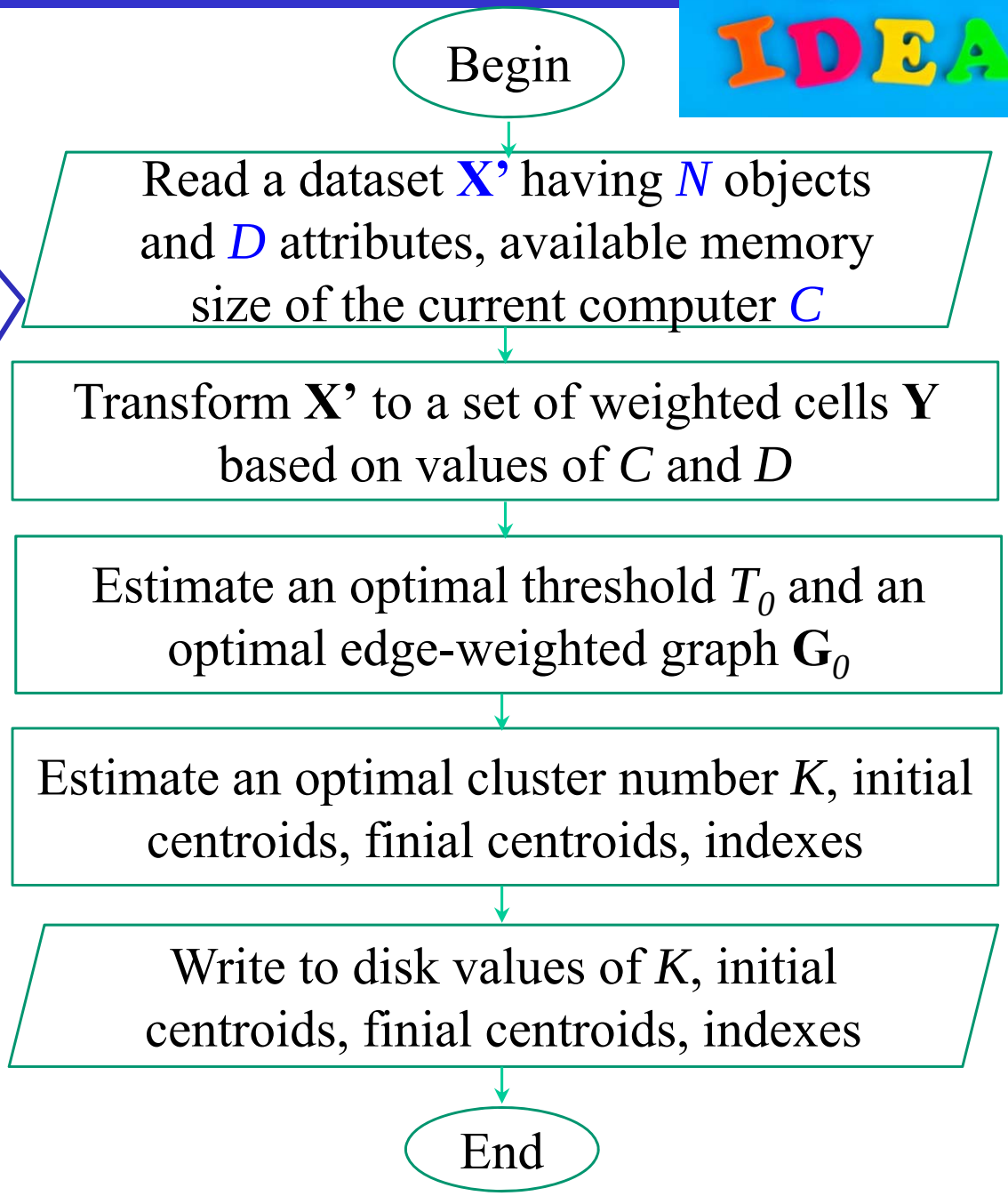
- To overcome the memory overwhelming problem of the previous methods.

- To be faster and more accurate than using the QEM method

Solution 2: Methodology

Cell-MST-Based Clustering algorithms for Big Datasets on Computers with Limited Memory

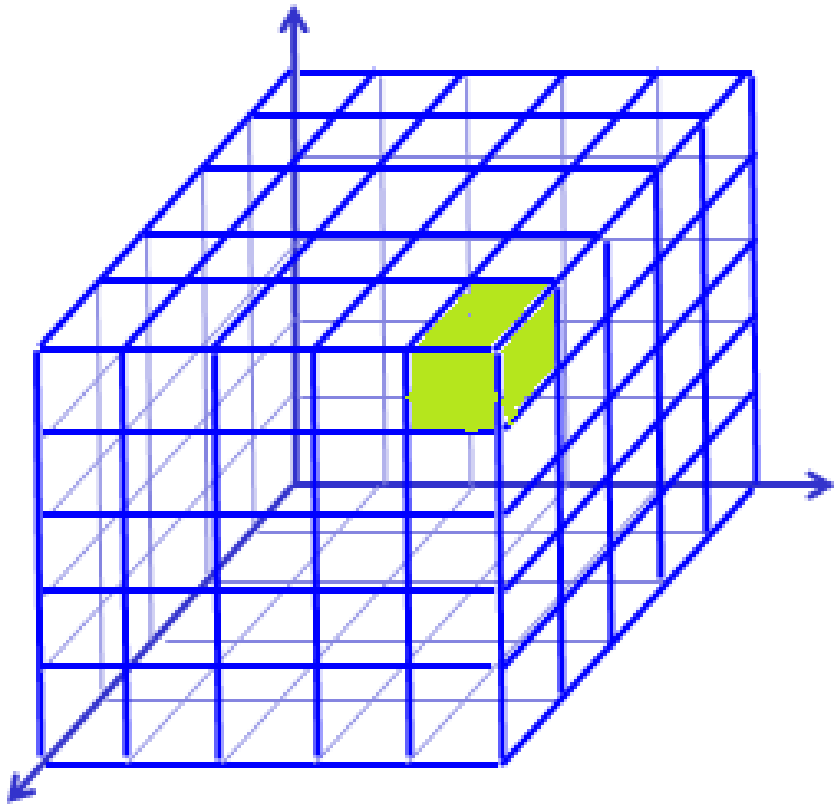
- **Presented at ICITEE 2015 conference**.



Solution 2: Methodology

Step 1: To transform a big dataset to a small set of cells

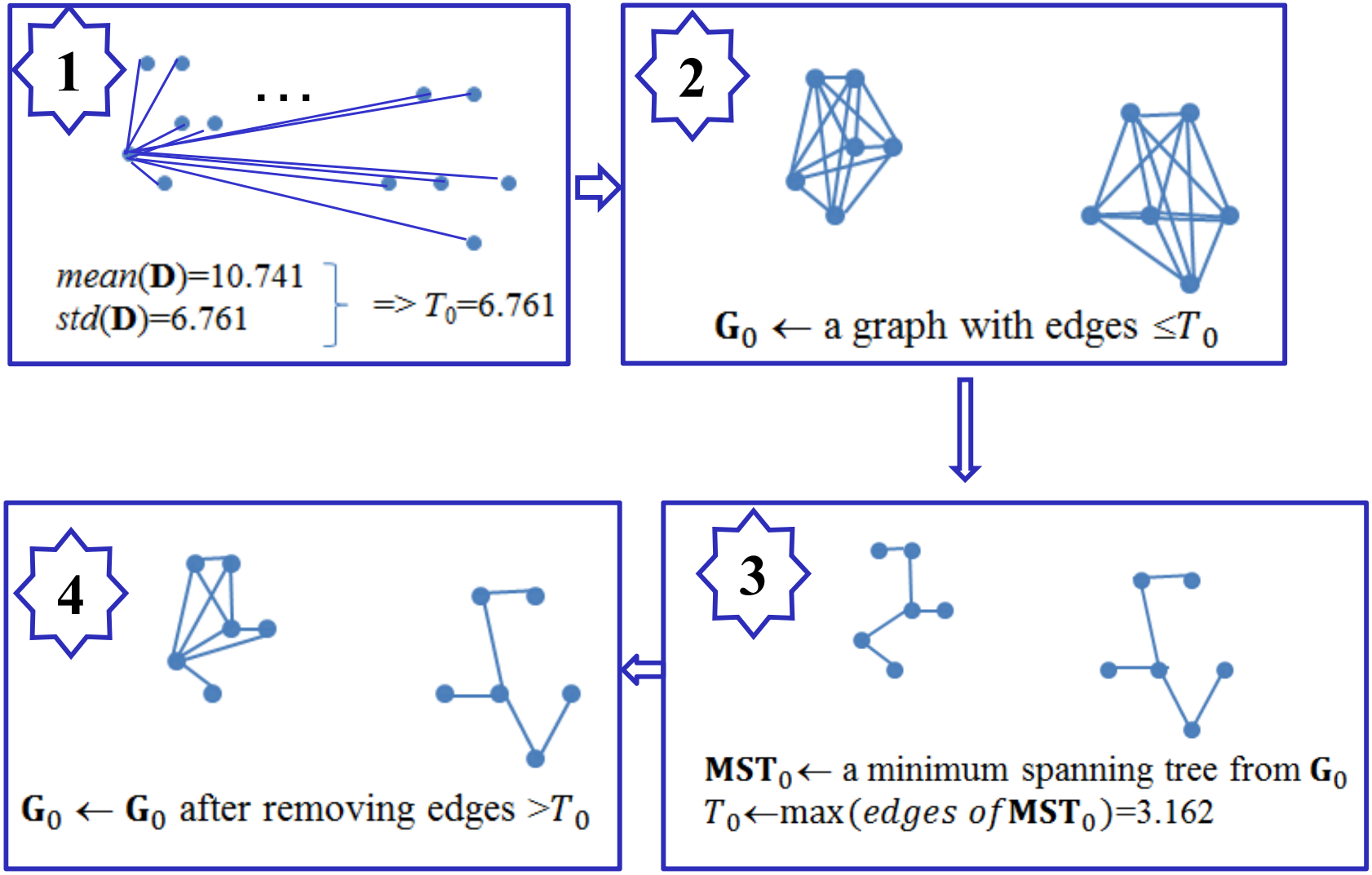
- M is initialized by available memory size and data type used to store distances between cells
- $\mathbf{X}$ is divided in to M^D cells
- Calculate a number of unempty cells, $C_{unempty}$



Solution 2: Methodology

Step 2: To estimate optimal threshold T_0 and graph G_0

A set of cells	
x	y
30	13
32	10
25	10
27	13
30	8
28	10
10	11
15	12
13	14
12	10
11	14
13	12



Solution 2: Methodology

Step 3: To estimate a cluster number and do clustering

1. Build various **forests** with various threshold T_i from T_0 to 1 from the graph $\boldsymbol{G}_0$
2. Compare the **forests** => the best forest => an optimal cluster number K_i
3. Run the K-mean clustering algorithm

Solution 2: Results

Comparison of required memory

Datasets	SIM, MST methods (in GB)	Running on the Machine 1		
		Unempty cells	Cell-MST (in GB)	Reduced memory (%)
Simulation2D1	1,192.14	20,988	0.8204	99.93
Simulation2D2	1,192.14	18,345	0.6268	99.94
Simulation2D3	2,463.35	11,068	0.2282	99.99
Simulation2D4	3,916.21	6,536	0.0796	99.99
Simulation3D1	228.17	5,177	0.0499	99.97
Simulation3D2	256.38	7,345	0.1005	99.96
Simulation3D3	2,463.35	1,459	0.0040	99.99
Simulation3D4	3,916.21	527	0.0005	99.99
Transactions70k	824.87	1,207	0.0027	99.99
Transactions90k	1,362.81	1,142	0.0024	99.99
3D road networks	352.25	4,974	0.0461	99.98
Power consumption	8,021.85	358	0.0002	99.99
TDriveTrajectory	587,562.50	1,116	0.0023	99.99

Reduce more than 99% of required memory sizes compared to using the original MST-based method

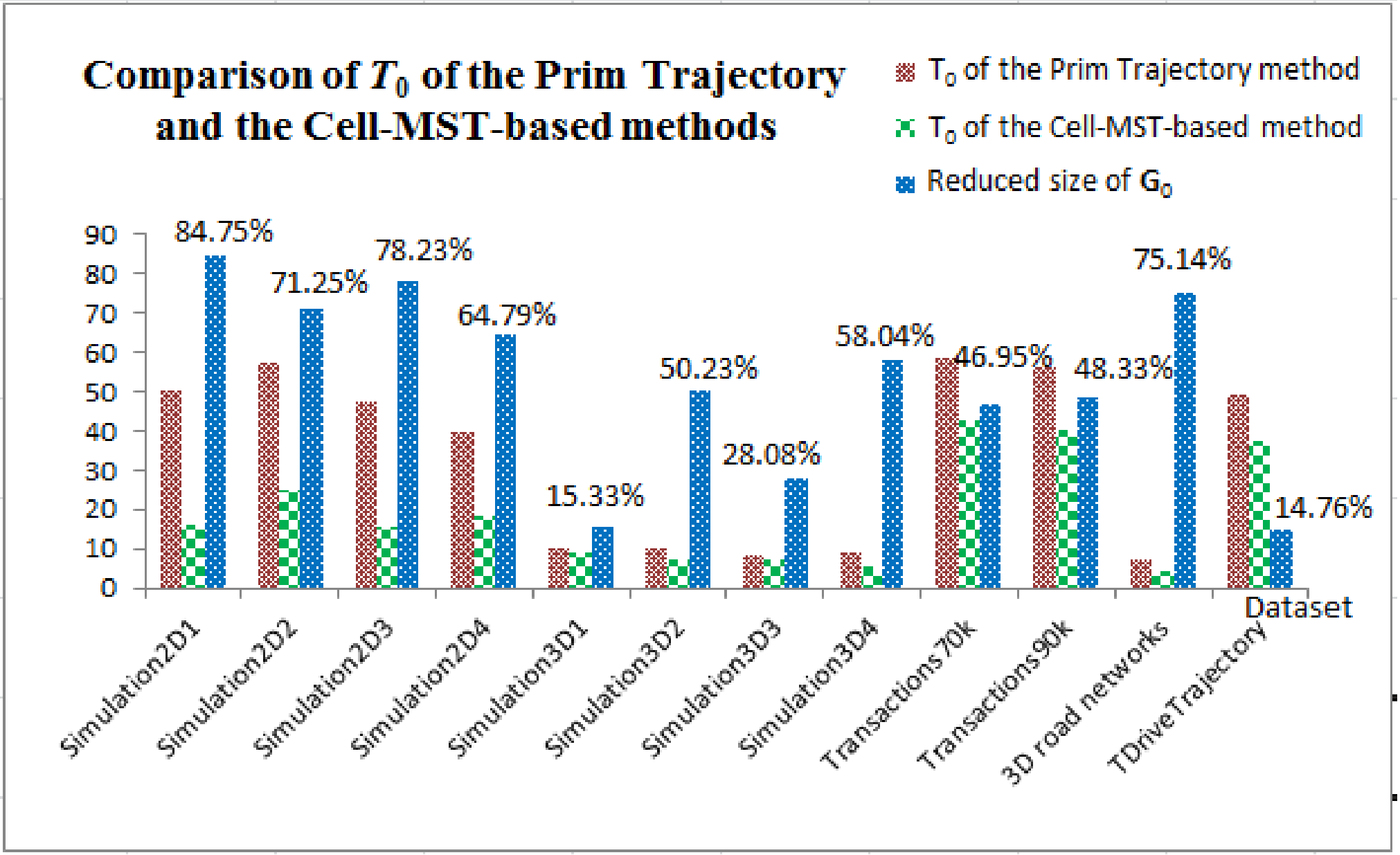
Solution 2: Results

- Comparison values of T_0

Datasets	Prim Trajectory		Cell-MST-based method		
	T_0	No. of Selected Edges	T_0	No. of Selected Edges	Reduced Edges (%)
Simulation2D1	50.57	41,359,158	16.12	6,305,392	84.75
Simulation2D2	57.31	34,418,067	24.99	9,896,801	71.25
Simulation2D3	47.53	10,477,300	15.40	2,280,602	78.23
Simulation2D4	39.85	2,809,222	18.60	989,048	64.79
Simulation3D1	10.08	2,273,135	8.91	1,924,607	15.33
Simulation3D2	9.98	4,444,453	6.93	2,211,789	50.23
Simulation3D3	8.61	72,571	7.00	52,194	28.08
Simulation3D4	9.05	19,757	5.69	8,291	58.04
Transactions70k	58.57	106,053	42.57	56,256	46.95
Transactions90k	56.54	97,437	40.59	50,348	48.33
3D road networks	6.89	1,355,799	3.96	337,088	75.14
TDriveTrajectory	49.36	252,839	37.57	215,513	14.76
Power consumption	0.95	0	2.97	29.112	No comparison

Solution 2: Results

- Comparison values of T_0



Solution 2: Results

- Comparison of cluster numbers

Datasets	No. of clusters	QEM method runs on Machine 1		Cell-MST-based method runs on Machine 1	
		No. of clusters	Intra_Inter Ratio	No. of clusters	Intra_Inter Ratio
Simulation2D1	10	29	3.519966	9	0.444574
Simulation2D2	16	29	3.931455	16	0.293210
Simulation2D3	22	26	1.241363	20	0.278007
Simulation2D4	29	21	0.375361	29	0.176896
Simulation3D1	7	18	0.796333	7	0.313435
Simulation3D2	7	29	0.897971	5	0.388929
Simulation3D3	23	22	2.663111	23	0.202289
Simulation3D4	25	29	3.331492	25	0.224338
Transactions70k	4	4	0.100067	4	0.100069
Transactions90k	4	2	0.249952	4	0.099972
3D road networks	#	20	0.516063	6	0.301752
Power consumption	#	28	1.045296	2	0.044454
TDriveTrajectory	#	29	0.891278	4	0.004215

Better estimated cluster number compared to the QEM method

Solution 2: Results

- Comparison of Executing time

Datasets	The averages of executing time running on the Machine 1(in minutes)						
	QEM method m_{Max}=29	Cell-MST method, α=12.5%		Cell-MST method, α=10%		Cell-MST method, α=7.5%	Cell-MST method, α=5%
		Running time	Reduced time (%)	Running time	Reduced time (%)		
Simulation2D1	11.54	2.13	81.53	3.92	66.00		
Simulation2D2	12.76	2.51	80.35	3.47	72.82		
Simulation2D3	34.71	1.53	95.58	1.74	94.99		
Simulation2D4	23.14	1.33	94.25	1.59	93.14		
Simulation3D1	7.76	0.42	94.56	0.47	93.96		
Simulation3D2	10.49	1.02	90.29	1.20	88.60		
Simulation3D3	28.01	1.72	93.86	2.06	92.65		
Simulation3D4	19.34	1.85	90.44	2.36	87.77		
Transactions70k	2.79	0.45	84.04	0.68	75.47		
Transactions90k	2.80	0.39	86.09	0.45	83.77		
3D road networks	159.59	0.94	99.41	0.97	99.39		
Power consumption	232.55	0.61	99.74	0.68	99.71		
TDriveTrajectory	767.45	20.23	97.36	29.04	96.22		

Much faster than the QEM method

Solution 2: Conclusions

1. The required memory is extremely smaller than the required memory of the previous similarity-based and MST-based algorithms (**reduce 99% of required memory**)
2. **Better** than using the Prim Trajectory algorithms
3. **Extremely faster** than using the quantization error modeling method
4. **Results are considerably better** than using the quantization error modeling method

Thank you
for your attention

Presenter: Mr. Duong Van Hieu
Email: dvhieu@gmail.com
duongvanhieu@tgu.edu.vn
duongvan.hieu@email.kmutnb.ac.th

3. Experiments and Analysis

3.1. Experiment Implementation

- Using the C programming language, compiled by the TDM-GCC 4.8.1 64 bit release associated with the Dev C++ 5.9.2.
- The 4-byte float data type is used to store edge lengths.
- A laptop computer configured with an Intel processor core i5-3230M CPU 2.60 GH, 6GB of RAM, Windows 8.1.
- A desktop computer configured with an Intel processor core i5-24000 CPU 3.10 GHz, 8GB of RAM, and Windows 7.

References of the Proposed Outlier Detection Algorithm

1. Chandola, V., Banerjee, A. and Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys, 41, 15:1-15:58 (2009)
2. Hawkins, D. M.: Introduction. In: Hawkins, D. M. (eds.) Identification of Outliers, pp. 1-9, Chapman & Hall (1980)
3. Aggarwal, C. C.: Outlier Analysis. In: Aggarwal, C. C. (eds.) Data Mining, pp. 237-263. Springer International Publishing Switzerland (2015)
4. Hodge, V. J.: Outlier Detection in Big Data. In: J. Wang (eds.) Encyclopedia of Business Analytics and Optimization, vol. 5, pp. 1762-1771. IGI Global (2014)
5. Bhattacharya, G., Ghosh, K. and Chowdhury, A. S.: Outlier detection using neighborhood rank difference. Pattern Recognition Letters, 60–61, 24-31 (2015)
6. Shaikh, S. and Kitagawa, H.: Top-k Outlier Detection from Uncertain Data. International Journal of Automation and Computing, 11, 128-142 (2014)
7. Breunig, M. M., Kriegel, H.P., Raymond, T. Ng, and Sander, J.: LOF: identifying density-based local outliers. ACM SIGMOD Record, 29, 93-104 (2000)
8. Tang, J., Chen, Z., Fu, A.W.C. and Cheung, D.W.L.: Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In: the Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (2002)
9. Jin, W., Tung, A. H., Han, J. and Wang, W.: Ranking Outliers Using Symmetric Neighborhood Relationship. In:Ng,W.K., Kitsuregawa, M. , Li, J. and Chang, K. (eds.) Advances in Knowledge Discovery and Data Mining,vol. 3918, pp. 577-593. Springer Berlin, Heidelberg (2006)

References the Proposed Outlier Detection Algorithm

10. Huang, H., Mehrotraa, K. and Mohana,C. K.: Rank-based outlier detection. Journal of Statistical Computation and Simulation, 83, 518-531(2013)
11. Huang, H., Mehrotra,K. and Mohan,C.: Algorithms for Detecting Outliers via Clustering and Ranks. In:Jiang,H., Ding, W., Ali, M. and Wu, X. (eds.) Advanced Research in Applied Artificial Intelligence,vol. 7345, pp. 20-29. SpringerBerlin, Heidelberg (2012)
12. Ha, J.,Seok, S. and Lee, J.S.: A precise ranking method for outlier detection. Information Sciences, 324, 88-107 (2015)
13. Hodge, V. J.: Outlier Detection in Big Data. In: Wang, J.(eds.) Encyclopedia of Business Analytics and Optimization, vol. 5, pp. 1762-1771. Business Science Reference, Hershey (2014)
14. Hieu, D. V. and Meesad, P.: A Cell-MST-Based Method for Big Dataset Clustering on Limited Memory Computers. In: 7th International Conference on Information Technology and Electrical Engineering, pp. 632-637, Chiang Mai, Thailand (2015)
15. Yuan, J., Zheng, Y., Xie, X. and Sun, G.: Driving with knowledge from the physical world. In: the Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, San Diego, California, USA (2011)
16. Lichman, M.: Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html

References of the proposed Cell-MST-based clustering algorithm

[1] P. Kokol, "Data Mining and Knowledge Discovery, Introduction to," in Encyclopedia of Complexity and Systems Science, R. A. Meyers, Ed., ed: Springer New York, 2015, pp. 1-3.

[2] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," Information Sciences, vol. 275, pp. 314-347, 8/10/ 2014.

[3] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, et al., "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," Emerging Topics in Computing, IEEE Trans. on, vol. 2, pp. 267-279, 2014.

[4] D. Van Hieu and P. Meesad, "Fast K-Means Clustering for Very Large Datasets Based on MapReduce Combined with a New Cutting Method," in Knowledge and Systems Engineering. vol. 326, V.-H. Nguyen, A.-C. Le, and V.-N. Huynh, Eds., ed: Springer International Publishing, 2015, pp. 287-298.

[5] A. Ben Ayed, M. Ben Halima, and A. M. Alimi, "Survey on clustering methods: Towards fuzzy clustering for big dataa," in Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of, 2014, pp. 331-336.

[6] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," Expert Systems with Applications, vol. 40, pp. 200-210, 1// 2013.

[7] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," presented at the Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, Louisiana, 2007.

[8] A. Hui and L. Wei, "K-means initial clustering center optimal algorithm based on estimating density and refining initial," in Information Science and Service Science and Data Mining (ISSDM), 2012 6th International Conference on New Trends in, 2012, pp. 603-606.

[9] G. Tzortzis and A. Likas, "The MinMax k-Means clustering algorithm," Pattern Recognition, vol. 47, pp. 2505-2516, 7// 2014.

[10] L. Han, W. Qingfeng, D. Huailin, W. Shuangshuang, C. Qing, and M. Zhuo, "A method for automatically determining The number of clusters of LAC," in Computer Science & Education, 2009. ICCSE '09. 4th International Conference on, 2009, pp. 1907-1910.

[11] X. Shao, J. Pi, and l. Liu, "A method of dynamically determining the number of clusters and cluster centers," in Computer Science & Education (ICCSE), 2013 8th International Conference on, 2013, pp. 283-286.

[12] H. Yu, Z. Liu, and G. Wang, "An automatic method to determine the number of clusters using decision-theoretic rough set," International Journal of Approximate Reasoning, vol. 55, pp. 101-115, 1// 2014.

[13] A. Kolesnikov, E. Trichina, and T. Kauranne, "Estimating the number of clusters in a numerical data set via quantization error modeling," Pattern Recognition, vol. 48, pp. 941-952, 3// 2015.

References of the proposed Cell-MST-based clustering algorithm

[14] T. Ishioka, "Extended K-means with an Efficient Estimation of the Number of Clusters," in Intelligent Data Engineering and Automated Learning — IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents. vol. 1983, K. Leung, L.-W. Chan, and H. Meng, Eds., ed: Springer Berlin Heidelberg, 2000, pp. 17-22.
[15] M. Yan and K. Ye, "Determining the Number of Clusters Using the Weighted Gap Statistic," Biometrics, vol. 63, pp. 1031-1037, 2007.
[16] K. Stokes, "Graph k-Anonymity through k-Means and as Modular Decomposition," in Secure IT Systems. vol. 8208, H. Riis Nielson and D. Gollmann, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 263-278.
[17] C. Zhong, M. Malinen, D. Miao, and P. Fränti, "A fast minimum spanning tree algorithm based on K-means," Information Sciences, vol. 295, pp. 1-17, 2/20/ 2015.
[18] D. Reddy, D. Mishra, and P. Jana, "MST-Based Cluster Initialization for K-Means," in Advances in Computer Science and Information Technology. vol. 131, N. Meghanathan, B. Kaushik, and D. Nagamalai, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 329-338.
[19] L. Galluccio, O. Michel, P. Comon, and A. O. Hero Iii, "Graph based k-means clustering," Signal Processing, vol. 92, pp. 1970-1984, 9// 2012.
[20] M. Piao Tan and C. A. Floudas, "Determining the optimal number of clusters Determining the Optimal Number of Clusters," in Encyclopedia of Optimization, C. A. Floudas and P. M. Pardalos, Eds., ed: Springer US, 2009, pp. 687-694.
[21] S. Ray and R. H. Turi, "Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation," presented at the Proceedings of the 4th international conference on advances in pattern recognition and digital techniques (ICAPRDT '99), Calcutta, India, 1999.
[22] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 28, pp. 301-315, 1998.
[23] R. Sedgewick and K. Wayne, "Graph," in Algorithms, ed: Pearson Education, Inc., 2011, pp. 515-694.
[24] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, et al., "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," Journal of Multiple-Valued Logic and Soft Computing vol. 17, pp. 255-287, 2011.
[25] M. Lichman. (2015, 3-June-2015). {UCI} Machine Learning Repository. Available: http://archive.ics.uci.edu/ml/datasets.html