2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

# *CF*-Cluster: Clustering Bike Station based on Common Flows

Liangxu Liu
*school of electronic and information engineer*
Ningbo University of Technology
Ningbo, China
luransh@ 126.com

Dayao Gong *
*school of electronic and information engineer*
Ningbo University of Technology
Ningbo, China
527250255@qq.com

Bo Guan
*school of electronic and information engineer*
Ningbo University of Technology
Ningbo, China
guanbo@nbut.cn

Junyan Xiao
*school of electronic and information engineer*
Ningbo University of Technology
Ningbo, China
979712564@qq.com

***Abstract*—Along with the rapid development of green travel of city bike sharing, how to mine moving patterns from dataset of sharing bike have gradually become hot point of bike sharing research (e.g., bike scheduling, city computing, and so on). Stations clustering is the base of these research directions. Existing literature was clustered the stations by their scalar data, such as, the location, the number of bike lent, the number of bike returned, and so on. Obviously, these clusters didn't own similar features of bicycle flows because of no taking the relations between stations into account. In this paper, we propose an algorithm of clustering analyses, called** Cluster **analysis based on** Com**mon** Flow **(*CF-Cluster* for short), based on similar relations between stations. In *CF*-Cluster, the clusters are defined as the station subset, in which the ratio of common relations (Common Flow Ration) is exceeds the threshold. According to the feature of common flow in subset, *CF-Cluster* divides into two phases. The first is to discovering candidate station subsets through the idea of Apriori, which is classic algorithm of association rules. The second phase is to eliminates overlapped clusters in candidate subsets to obtain station clusters. Finally, Empirical evaluation proves that our algorithm owns availability and effectiveness. Moreover, scale-up experiments show the affects in the number and size of clusters.**

***Keywords—sharing bike, station clustering, Common Flows, vehicle scheduling***

## I. INTRODUCTION

"Tide Problem", where no bike in high demand area for bike renting and no more space in high demand area for bike returning, increasingly becomes a major problem restricting the further development of shared *bike*s. Although many researchers and practitioners have been working to solve the problem of tides, this problem has become more and more serious with the rapid development of sharing bike industry. In this case, huge number of historical data, which collected by sharing bike system has become one of new researching focus to resolve this problem.

In recent years, sharing bikes, whether with pile or without pile, have developed rapidly. "Tide" is the problem that both of them are faced. In this paper, we focus on resolving the "Tide" in sharing bikes system with pile. Bike-sharing system with pile is a way to rent or return sharing-bike from fixed station piles around the city. In the system with pile, resolving 'Tide problem' involves research on station layout, vehicle scheduling strategy, bike flow analysis, and so on.

Cluster analysis of the station is the basis of these researching directions. There are a lot of literatures of sharing-bike cluster analysis. Most of them divided into two categories. The first [1][2][3] is using classic cluster algorithm (such as k-means) to cluster the station, in which the feature of node is defined as those count scalars, such as, rental bike number, return bike number, available bike, available piles, and so on. Its defect is that this solution neglects the relations between station pairs. The second is using fixed model (such as Petri Net) to train abstracting sharing-bike system [15] [16]. But perfect training dataset is difficult to obtain.

The most relevant research in this paper are the former. But count scalars only show amount of changes, neglecting of the changes' coming from. Obviously, the clusters, based on these resolutions, is coarse. In this paper, the cluster is defined as station set with similar common inflow, that is to say, each entry owns that a certain rate of bikes come from similar source stations. And then ***Common Flow Rate*** (***CFR***) is introduced to describe similar source stations, which is the minimum ratio of bike number from common station of the subset to the total at each entry. Next, a novel clustering algorithm, Clustering Algorithm based on Common Flow (*CF-Cluster*, for short), is proposed. In *CF-Cluster*, clustering process is divided two phases. The former is that candidate station subsets (***p-Maximum Subset***) are mined by Appriori. The latter is Pruning station clusters from candidate sets by eliminating overlapped clusters.

## II. REATED WORK

Research areas related to "Tide Problem" involves several research areas, such as station layout, vehicle scheduling strategy, bike flow analysis, and so on. However, most of them is based on or start from cluster analysis of bike stations. In order to discovery more available station clusters, a lot of researchers try to pursuit this topic. For Example, Froehlich et al. discovered station clusters from Barcelona *Sharing bike* System according to available *Bike*s number, empty pile

This work is supported by the Zhejiang Provence Natural Science Foundation of China (Y14F020044, Y14F020045), Ningbo Natural Science Foundation of China (2014A610072)..

978-1-5386-1956-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.75

number, the position, and other the interaction between stations per five-minute interval[4][5]. In which, station clusters are discovered through EM algorithm and Gaussian Mixture model. Lathia et al analyzed the space and time spent on the station to get the impact the change of user's strategy to Barclays *Bike* rental model in London[6]. Borgnat et al took the count of arriving and living as station feature to determine the similarity of stations[7][8][9]. Vogel et al extracted the feature vector of the station from the counting series which represents the number of the *Bike* rented and borrowed per hour, and clustered by K-Means and Gaussian Mixture model based on EM and sIB[10][11]. Etienne et al. processed the counting series to create a mobility pattern by the Poisson mixture model and the station scale factor used to balance the differences between stations[12]. They took weekday and weekend into account. Another study, proposed by Chardondeng, was quite different[13]. They established Day Aggregation Model, Station Aggregation Model, and Station Aggregation Model according to the travel statistics, to realize the redistribution of station classification and the number of station level. In these literatures, the similarity between stations is mostly based on static characteristic, such as location and capacity, or simple dynamic factors(rental and return count series). And then classic clustering algorithm, like K-Means, were employed for clustering.

However, On the whole, Sharing *Bike* system is direct graph that stations is regarded as the node, and rented- returned record is the connection. Based on this feature, the similarity between stations is based more on the links than on the nodes. some researchers hope to discover available station clusters based on the links. For example, L. Chen et al [14] dynamically grouped neighboring stations into clusters according to context, so that the stations in the same cluster have similar *Bike* usage patterns. And then the weight of each link is calculated according to associated common and opportunistic contextual factors, and merge them together to construct the network, in which *Bike* stations is regard as nodes, and the link is connecting two stations if they are geographically close to each other. Neighboring stations with similar *Bike* usage patterns are grouped into clusters. These clusters can be considered as communities that are densely connected internally and loosely connected between each other. Finally, a Geographically-Constrained Label Propagation (GCLP) method was proposed to solve this problem.

Moreover, Kadri[15] and Labadi[16] regarded Sharing *Bike* system as a Petri-net, and then trained it according to the idea of Petric Net. Time, control arc, variable arc and weight feature in Petri Net enable it could be used to model the Sharing *Bike* system with performance evaluation and discrete event. However, Petri Net is black box, it is difficult to obtain a perfect training dataset to ensure system running.

Another related research field is association rule. Many algorithms for generating association rules have been proposed. Some well-known algorithms are Apriori, FP-Growth. Apriori uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support [17]. FP stands for frequent pattern [18]. In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure by inserting instances. If many instances share most frequent items, FP-tree provides high compression close to tree root.

## III. The Concept of *CF*-Cluster

Current popular resolution of vehicle scheduling and route planning is combining global and local scheduling. In this case, station clusters based on bike flows are more attractive than based on count scalar data. Therefore, current mainstream solution, which clusters the stations with position and pure count scalar (such as station rental and return counts, station's available bike or pile number), doesn't fit to be applied to scheduling. In this paper, station clustering based on Common Flows are proposed.

### *A. Basic Concept*

For the sake of simplifying the description, some basic concepts are provided as follows.

**Definition 1**: Given directed Graph $G_t=(V, E)$, $V$ is the set of bike station, and E is the set of bike usage records. $v_i(v_i \in V)$ is bike station. $e_{ij}=(v_i, v_j, k)(e_{ij} \in E)$ is the count of bike usage records, $|e_{ij}|=k$ is the weight of $e_{ij}$, which is bike number that rented bike from $v_i$ and returned it to $v_j$.

**Definition 2**: Given station set $f=\{v_1, \ldots, v_n\}$ is subset of $V$, defined as **Definition 1**. Its ***Source Set*** is defined as follows:

$$SS(f) = \{SS(v_1) \cap \ldots \cap SS(v_n)\} \quad (1)$$

Where $SS(v_i) = \{v_x \mid e_{xi}>0\}$

That is to say, Source Set of subset $f$ is station set, where no bike usage records is not exist between its entry and each entry of $f$.

**Property 1**: $f$ is the station collection. $f_1$ is a subset of $f$. So, $SS(f)$ is a subset of $SS(f_1)$.

Prove: Assuming $f=f_1 \cup f_1'$.

$SS(f)= SS(f_1 \cap f_1')= SS(f_1) \cap SS(f_1')$;

Obviously, $SS(f)$ is the subset of $SS(f_1)$.

**Definition 3**: Provided $f=\{v_1, \ldots, v_n\}$ is a subset of $G_t$. ***Common Flows Ratio*** $CFR(f)$ is defined as ***Formula* (2)**.

$$CFR(f) = min(\frac{\sum e_{xi} \mid v_x \in SS(f)}{S(v_i)}) \quad (2)$$

*CFR* of Subset $f$ is minimum value of each entry, which is the ratio of return bike number from $SS(f)$ to that of all .

**Property 2:** if $f1$ is the subset of $f$, $CFR(f_1) \geq CFR(f)$.

**Prove:** according to ***Property* 1**, if $f1$ is the subset of $f$, $SS(f)$ is the subset of $SS(f_1)$. So $CFR(f_1) \geq CFR(f)$.

### B. Station Clustering

In this paper, station cluster is used in vehicle scheduling to optimize the route. The cluster is regarded as station set, which entry own A certain proportion of common flows. Obviously, station cluster is station set which size is as large as possible. For ease of expression, *p*- **Maximum Subset** are introduced.

**Definition 4**: Given subset $f$ of $G_t$ with $CRF(f)=p$, $f$ is ***p*-Maximum Subset**, if and only if there is no subset $f' \in \boldsymbol{MS_p}(G_t)$, satisfied with $f' \supset f$ . Denoted by $\boldsymbol{f \in MS_p}(G_t)$. $\boldsymbol{MS_p}(G_t)$ is the set of ***p*-Maximum Subset** in $G_t$.

According to sharing-bike system, $\boldsymbol{MS_p}(G_t)$ owns lots of overlapping subsets(following experiments prove this result). For Example, $s_1$, $s_2$, $s_3$, $s_4$, $s_5$, $s_6$, $s_7$ are seven stations in same dense region, subset $f_1=\{s_1, s_2, s_3, s_4\}$ is a ***p*-Maximum Subset** maybe mean that similar subset of $f_1$ (such as $\{s_1, s_2, s_3, s_7\}$, $\{s_1, s_2, s_3, s_5\}$,$\{s_1, s_3, s_4, s_6\}$, $\{s_1, s_3, s_4, s_7\}$) are ***p*-Maximum Subset** too. In order to avoid this disturb, ***p,q*-Maximum Subset** are defined as follows.

**Definition 5**: Given $\boldsymbol{f \in MS_p}(G_t)$, $\boldsymbol{f}$ is ***p, q*-Maximum Subset, denoted by $\boldsymbol{f \in MS_{p,q}}(G_t)$,** if and only if there is no subset $f'$, which satisfied:

(1) $CFR(f) \geq CFR(f')$;

(2) more than $q$ of stations in $f'$ belong to $f$.

Obviously, both of $\boldsymbol{MS_p}(G_t)$, $\boldsymbol{MS_{p,q}}(G_t)$ are station clusters. the subset of, which delete most overlapped entries. There are lots of overlapped clusters in $\boldsymbol{MS_p}(G_t)$ , $\boldsymbol{MS_{p,q}}(G_t)$ is the subset of $\boldsymbol{MS_p}(G_t)$, which delete many clusters with higher overlapped stations.

## IV. Implementation of Station Cluster

According to above analysis, discovery station clusters with Common Flows equate to discover $\boldsymbol{MS_p}(G_t)$ and $\boldsymbol{MS_{p,q}}(G_t)$. the process of Clustering Algorithm based on Commom Flows involves two phases. The first is to discover all $\boldsymbol{MS_p}(G_t)$ as candidate clusters, implemented by function *CandidateMS-gen*. The second is to Prune station clusters from candidate clusters that obtain from the first phase, implemented by function *MS-gen*. Figure 1 shows its pseudo code.

```
1 MS_p = CandidateMS-gen(p, G_t);
2 MS_pq = MS-gen(p, MS_p);
```

Figure 1 Pseudo code of *CF-Cluster* Algorithm

### A. Discover candidate Station Clusters

The target of first phase is select all *p*-**Maximum Subsets** (candidate station clusters). Native solution is calculating Common Flows Ratio of all subsets to determine whether it is *p*-**Maximum Subset**. Its Computational complexity is too expensive(O(2^n)). n is the number of station.

From ***Property* 2**, ***Common Flow Ratio*** of set $f$ must be smaller than or equal to that of its subset. Therefore, Apriori is introduced to optimize the first phase. Figure 2 shows its pseudo code. The process divided into three steps.

Firstly, all subset with two station, which Common Flows Ratio isn't less than $p$ (User specified), to form candidate set $L_2$ (with two stations).

Secondly, the candidate set $C_r$ is generated from set $L_{r-1}$ (with $r$-1 stations), this step is implemented by function *apriori-gen*(),detailed in [17].

Thirdly, for each entry of $C_r$ is executed two steps. The first step is $r$-th entry, which Common Flows Ratio exceeds $p$ (user specified), is inserted into $L_r$ . The second is all its subset in $L_{r-1}$ are deleted.

Repeat step 2 and step 3 until no entry in $C_k$. the result is the collection of $L_k$.

```
1 L_2={(v_i,v_j)|CFR(v_i,v_j)>=p};
2 for(r=3; L_{r-1}<>∅ ;r++)
3   C_r = Apriori-gen(L_{r-1});
4   computeCR(C_r);
5   L_r= {f_i∈ C_r | CR(f_i)≥p}
6 Answer = ⋃_r L_r
```

Figure 1 Pseudo code of *CandidateMS-gen*

### B. Prune Station Clusters

From above analysis, candidate station clusters, which are generated in first phase, are higher overlapped. The target of second phase is eliminating high overlapped station clusters. The idea of this phase is that station cluster would be eliminated if another candidate station cluster, which satisfied ***Definition 5***, exists. Figure 3 shows its pseudo code. The process is simple. The algorithm just check whether candidate station cluster is overlapped with the other with more size or more higher value of *CRF*. If true, eliminating this entry.

```
1 for (r=3; L_{r-1}<>∅ ; r++)
2   For each entry sc in L_r;
3     if isOverlapped(sc, MS_p, q);
4       L_r = L_r -{sc}
5 Answer = MS_p
```

Figure 3 Pseudo code of *MS-gen*

## V. Experiments

To validate the efficiency and effectiveness of our proposed method, extensive experiments are performed on dataset from real world. In this section, Data representation is described firstly. Next, experimental results are analyzed in running time and cluster number, and then compared using different dataset. Finally, the results from different parameter p are analyzed.

TABLE I.

HE DESCRIPTION OF DATASETS

| Data set | ***10min*** | ***1houe*** | ***2hours*** | ***4hours*** | ***8hours*** |
|---|---|---|---|---|---|
| #records | 75287 | 186521 | 455612 | 747433 | 1509659 |
| #stations | 581 | | | | |
| #Date duration | 2016-01-01~2016-12-31 | | | | |
| #City | Chicago | | | | |

*A. Data Representation and experimental envionment*

We evaluate our framework Sharing bike System in Chicago. Experimental dataset is sharing bike data (01/01/2016 – 12/31/2016). The data processing details are as follows. one years' bike trip historical records are collected from data portals of Chicago. We extract the data format (record id, rental station, rental time, return station, return time) from trip record. Based on this data, we count the vector (station ID, return number) of the intervals is calculated (station ID is source station ID, the number is the number of bikes that rent from source station and return to target station), and then target station's multidimensional vector is formed for clustering. We have set six di□erent sampling periods to analyze the . They are: 8 hours, 4 hours, 2 hours, 1 hour, 30 minutes, and Start time is 7:00am. Dataset description is presented in ***Table*** □.

All Experiments are carried out at a PC with Intel Core i5-6500 CPU, and 8GB RAM. Operation System is 64bit Win 7. Algorithm is developed by C++, and map display is implemented by Java web.

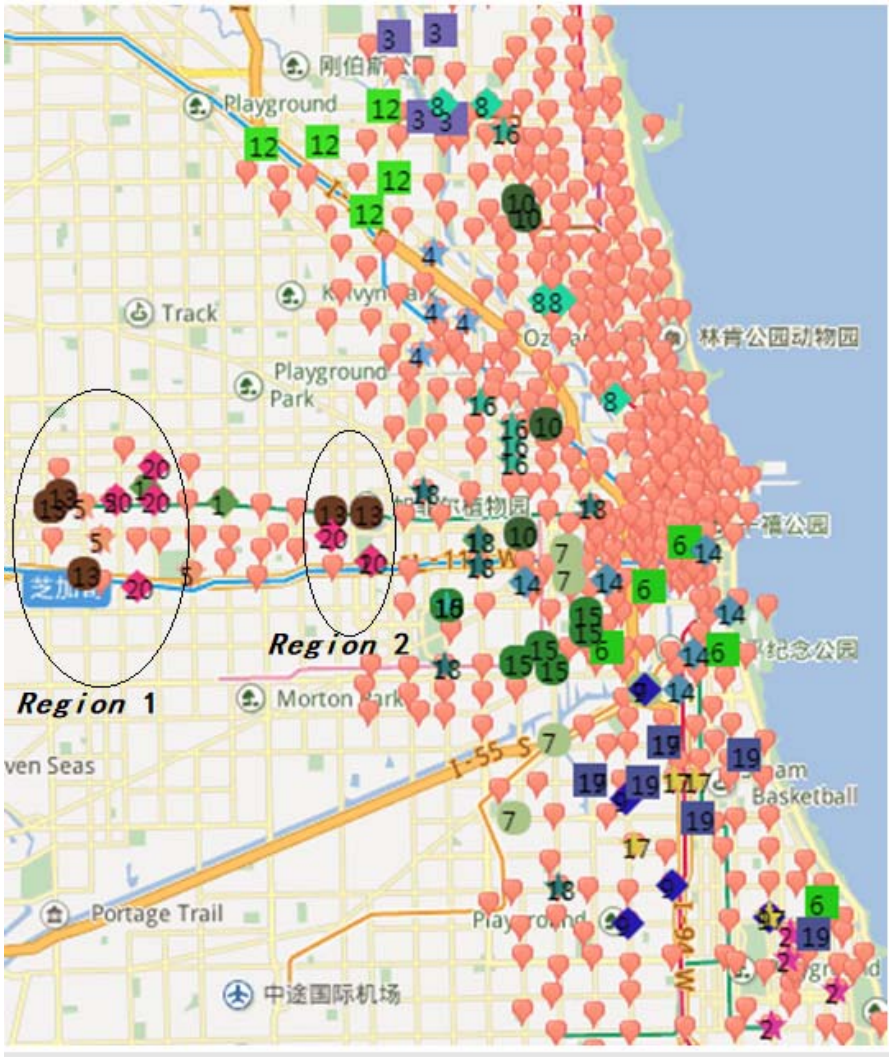


Figure 4 Station Clusters by *CF*-Cluster($p$=0.2, $q$=0.5 )

*B. Results Analysis*

Figure 4 shows some clusters from *CommFlow-Cluster* ($p$=0.2, $q$=0.5), and experimental data is using 2 hours. For clarity, the clusters is partial. From the Figure, there are several different styles of clusters. most clusters own strong regional feature. Some of them are in the suburbs (such as, 2#, 3#, 12#, 15#, and so on), which station could be the controlling points of this region in vehicle scheduling. the other are near the central(such as, 6#, 15#, 7#, 14#, and so on), in which a small enclosed region maybe be under cover. Moreover, there are some novel clusters that have great area. For example, 13# and 20# clusters divided west area of Chicago into two regions(***Region 1*** and ***Region 2*** in ***Figure* 4**). This phenomena means that there are certain Residential area between two regions. The Similarity is clusters collection of (19#, 6#), (8#,

10#, 16#). Obviously, these results couldn't obtain by others algorithms.

TABLE II. EXPERIMENTAL RESULTS WITH DIFFERENT DATASET

| Data set | Before Pruned | | After Pruned | |
|---|---|---|---|---|
| | *Running Time* | *Clusters* | *Running Time* | *Clusters* |
| 30 minutes | 549 | 48 | 550 | 28 |
| 1 hour | 702 | 33 | 703 | 17 |
| 2 hours | 1004 | 17 | 1005 | 11 |
| 4 hours | 1488 | 19 | 1489 | 8 |
| 8 hours | 1974 | 7 | 1975 | 4 |

***Table*** □ shows experimental results (running time and clusters number) before and after eliminated with different dataset($p$=0.30, $q$=0.5). From the Table, we can find that computation time becomes larger and larger as time span of the dataset increases. This is because that bike usage records' becoming large would make vector dimension of station higher. However, to our surprise, The number of clusters does not increase as time span of the dataset increases. The reason is that higher dimension of station vector makes clusters' *CRF* less and less. Meantime, we can find out that cluster number drop substantially. This phenomena show there are lots of overlapped clusters. To our surprise, less clusters are overlapped after pruning.

### C. Effectiveness of Parameters

The parameters, which affect *CF-Cluster*, involve the threshold of common flow ratio $p$ and Merging factor $q$. In this section, we show experimental results along with the changes of them.

TABLE III. EXPERIMANTAL RESULTS WITH DIFFERENT *P*

| *p* | Before Pruned | | After Pruned | |
|---|---|---|---|---|
| | *Running Time* | *Clusters* | *Running Time* | *Clusters* |
| 0.1 | 1869 | 519 | 1923 | 59 |
| 0.2 | 1073 | 91 | 1074 | 28 |
| 0.3 | 1042 | 17 | 1043 | 11 |
| 0.4 | 1038 | 8 | 1039 | 6 |
| 0.5 | 1030 | 4 | 1031 | 4 |

Table □ shows the changes of running time and cluster number with parameter $p$ from 0.1 to 0.5($q$ = 0.5), experimental dataset is 2 hours interval. From Table, we could find out that both of running time drops largely along with the increase of $p$. the reason is that candidate subsets will reduce rapidly with the increase of parameter $p$. Meantime, cluster number is reduced.

TABLE IV. EXPERIMANTAL RESULTS WITH DIFFERENT *Q*

| q | Before Pruned | | After Pruned | |
|---|---|---|---|---|
| | *Running Time* | *Clusters* | *Running Time* | *Clusters* |
| 0.2 | 1126 | 42 | 1127 | 15 |
| 0.4 | 1126 | 42 | 1086 | 17 |
| 0.6 | 1126 | 42 | 1080 | 18 |
| 0.8 | 1126 | 42 | 1038 | 30 |

Table □ shows the changes of running time and cluster number with parameter $q$ from 0.2 to 0.8 ($p$=0.25), experimental dataset is 2 hours interval. From Table, we could find out that running time and cluster number before Pruned doesn't change along with $q$ from 0.2 to 0.8. because q is the parameter in pruned phase. In pruned phase, running time increase a few along with q from 0.2 to 0.8, the reason is that candidate subset is eliminated more easily. And cluster number would become larger and larger along with p from 0.2 to 0.8. the reason is obviously.

## VI. CONCLUSION

Aimed to the defects of existed *Bike* station clustering analysis, this paper proposes a novel station clustering algorithm, which computes the similarity between station based on Common Flow. Based on this, this paper proposes a novel station Cluster algorithm based on Common Flows (CF-Cluster for shortly), which divides clustering process into two phases. The first phase aims to discovering candidate station subsets which Common Flows Ratio exceeds certain value. The second phase is to eliminating overlapped clusters from candidate subsets to obtain station clusters. Experimental results show that the clusters from *CF-Cluster* is effectiveness and available.


## ACKNOWLEDGMENT

This work is supported by the Zhejiang Provence Natural Science Foundation of China (Y14F020044, Y14F020045), Ningbo Natural Science Foundation of China (2014A610072).


## REFERENCES


[1] M. Benchimol, et al. Balancing the stations of a self- service bike hire system. RAIRO-Operations Research, 45(1):37–61, January 2011.

[2] Gaspero L D, Rendl A, Urli T. Balancing sharing bike systems with constraint programming[J]. Constraints, 2016, 21(2):318-348.

[3] R. Nair, E. Miller-Hooks, R. C. Hampshire, and A. Bu˘si´c. Large-Scale Vehicle Sharing Systems: Analysis of V´elib'. International Journal of Sustainable Transportation, 7(1):85–106, April 2012.

[4] J. Froehlich, J. Neumann, and N. Oliver. Measuring the pulse of the city through shared bicycle programs. In UrbanSense08, pages 16–20, 2008.

[5] J. Froehlich, J. Neumann, and N. Oliver. Sensing and predicting the pulse of the city through shared bicycling. In 21st International Joint Conference on Artificial Intelligence, IJCAI'09, pages 1420–1426. AAAI Press, 2009.

[6] Neal Lathia, A. Saniul, and L. Capra. Measuring the impact of opening the London shared bicycle scheme to casual users. Transportation Research Part C: Emerging Technologies, 22:88–102, June 2012.

[7] P. Borgnat, E. Fleury, C. Robardet, and A. Scherrer. Spatial analysis of dynamic movements of V´elo'v, Lyon's shared bicycle program. In Francois Kepes, editor, European Conference on Complex Systems, ECCS'09. Complex Systems Society, September 2009.

[8] P. Borgnatet al. Shared Bicycles in a City: A Signal processing and Data Analysis Perspective. Advances in Complex Systems, 14(3):1–24, June 2011.

[9] P. Borgnatet al. Dynamics On and Of Complex Networks, Volume 2, chapter A Dynamical Network View of Lyon's V´elo'v Shared Bicycle System. Springer Berlin Heidelberg, 2013. URL http://liris.cnrs.fr/publis/?id=5713.

[10] P. Vogel and D.C. Mattfeld. Strategic and operational planning of bike-sharing systems by data mining - a case study. In ICCL, pages 127–141. Springer Berlin Heidelberg, 2011.

[11] P. Vogel, T. Greiser, and D.C. Mattfeld. Understanding bike-sharing systems using data mining: Exploring activity patterns. Procedia - Social and Behavioral Sciences, 20(0):514 – 523, 2011.

[12] Etienne COME, Latifa Oukhellou. Model-based count series clustering for Bike Sharing System usage mining, a case study with the Velib system of Paris. ACM Transactions on Intelligent Systems and Technology, 2014, 27p.

[13] Chardon CMD, Caruso G. Estimating bike-share trips using station level data[J]. Transportation Research Part B Methodological, 2015, 78:260-279.

[14] L Chen et al. Dynamic cluster-based over-demand prediction in bike sharing system [J]. ACM International Joint Conference. 2016:841-852

[15] A Kadri, K Labadi, I Kacem. An integrated Petri net and GA based approach for performance optimization of bicycle sharing systems [J]. European J of Industrial Engineering, 2015, 9(5)

[16] K Labadi et al. Stochastic Petri Net Modeling, Simulation and Analysis of Public Bicycle Sharing Systems. IEEE Transactions on Automation Science & Engineering, 2015, 12(4): 1380-1395.

[17] Agrawal Rakesh, Srikant Ramakrishnan. Fast algorithms for mining association rules in large databases, in Bocca, Jorge B.; Jarke, Matthias; and Zaniolo, Carlo; editors, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, pages 487-499.

[18] Han (2000). Mining Frequent Patterns Without Candidate Generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD '00: 1–12. doi:10.1145/342009.335372.