Conflict-Awareness Event Handling

Rahul B. CHougule[1], Ajay P. Jadhav[2], Arjun V. Patil[3], Sameed S. Bondre[4], Deepak S. Nandaygol[5],Praveen A. Patel[6]
[1,2,3,4,5]Students, Bharati Vidyapeeth's College of Engineering, Shivaji University, Kolhapur
Email: sameedbondre@gmail.com
[6] Assistant Professor, Bharati Vidyapeeth's College of Engineering, Shivaji University, Kolhapur
Email: praveenpatel7@gmail.com

ABSTRACT

With the rapid developing of Web 2.0 and Online To Offline merchandising model, various online case -based mixer web EBSNs are getting popular. An important task of EBSNs is to facilitate the most satisfactory event-player system for both sides, i.e., event enroll more players and players are arranged with personally interesting case. Existing overture usually focusing on the arrangement of each single event to a set of potential users, or ignore the confidence trick lists between different events, which leads to infeasible or redundant organization. In this paper, to address the shortcomings of existing approaches, we first identify a more general and useful event-participant arrangement trouble , called Global Event-participant Agreement with Conflict and Capacitance (GEACC) trouble , focusing on the conflicts of different events and making event-participant arrangements in a global thought We find that the GEACC trouble is NP-hard due to the conflicts among events. Thus, we Figure two approximation algorithms with provable approximation ratios and an exact algorithm with pruning technique to address this problem. In addition, we propose an online setting of GEACC, called Online GEACC, which is also practical in real-world scenarios. We further design an online algorithm with provable execution warrant. Finally, we verify the effectiveness and efficiency of the proposed method acting through extensive experiments on real and synthetic datasets.

Keywords – Greedy Algorithm, Minimum cost flow, Maximum Cost flow, Final Agreement

I. INTRODUCTION

The prevalence of Web a pair of. On-line To Offline (O2O) selling model has LED to the boom of numerous on-line event-based social networks EBSNs. For example, Groupon1 collects cluster purchase events and recommends these group discounts to users, and Meetup2 receives information on accomplishment of attendees in offline events, such as gatherings, sports activities, etc., and sends such information to users. Such EBSNs facilitate organizing social events and ease the recruitment of cluster activity participants. Note that "participant" and "user" are used interchangeably in this paper. However, most existing EBSNs only offer a public open event info sharing platform, where strategic organization and international event-participant arrangement are absent. Imagine the following scenario. Bob is a sport enthusiast and typically attends sports activities organized on Meet up. In a Saturday evening, Bob faces a dilemma since Meet up recommends him 3 conflicting sport activities on Sunday: a hiking trip from 8:00 a.m. to 12:00 p.m., a badminton game from 9:00 a.m. to 11:00 a.m., and a basketball game from 11:30 a.m. to 1:30 p.m. on a basketball court that's one-hour away by car from the court game sports stadium. Though Bob is interested in all 3 sports, he can solely attend at most one in all them. In fact, several users typically encounter the same problem: they need to confront with a confusing selection from many conflicting events. Besides resolving conflicts of events, it is appealing to own an event-participant arrangement strategy that globally optimizes the benefits of each event organizers and users, e.g., for organizing a carnival or a film festival. Particularly, [2], [3], [4], [5] are the recent studies on such event arrangement downside in static eventualities, a.k.a. offline scenarios, i.e., information of events and users is absolutely given. In addition to the above static setting, the online setting of EBSN platforms, which has not however been studied, is also vital. That is, users can dynamically login EBSN platforms at any time and register for events in a first-come, first-served way. Since an EBSN platform cannot recognize in advance whether or not users UN agency are a lot of inquisitive about an exact event can return later or not, it has to create decisions exclusively for this user. Imagine the following scenario. Carol is music fan and wants to attend a concert. At the time she logs in, only a pop and a jazz music concert are on the market with one spot left, respectively. Carol decides to join the pop music concert. Two days later, David, who is solely inquisitive about pop, logs in the platform. However, since the pop music concert has no quota left, David cannot find any interesting event to attend.

II. GUIDELINES OF THE PAPER

In this paper, to address the inadequacies of existing methodologies, we first distinguish a more broad and helpful occasion member course of action issue, called Global Event-member Arrangement with Conflict and Capacity () issue, concentrating on the contentions of various occasions and making occasion member game plans in a worldwide perspective. We find that the GEACC issue is NP-hard because of the contentions among occasions. Accordingly, we outline two guess calculations with provable estimate proportions and a careful calculation with pruning system to address this issue. What's more, we propose an internet setting of GEACC, called Online GEACC, which is likewise viable in genuine situations. We promote plan an online calculation with provable execution ensure. At last, we confirm the adequacy and proficiency of the proposed techniques through broad tests on genuine and manufactured datasets.

With the fast advancement of Web 2.0 and Online To Offline show casing model, different online occasion based informal communities are getting prevalent. An essential errand of EBSNs is to encourage the most attractive occasion member course of action for both sides, i.e., occasions select more members and members are organized with specifically fascinating occasions. Existing methodologies typically concentrate on the course of action of every single occasion to an arrangement of potential clients, or disregard the contentions between various occasions, which prompt infeasible or excess game plans. "member" and "client" are utilized reciprocally as a part of this paper. Be that as it may, most existing EBSNs just give an open/open occasion data sharing stage, where vital association and worldwide occasion member game plan are truant. Envision the accompanying situation. Actually, numerous clients for the most part experience the same issue: they need to face with confounding decision from numerous clashing occasions. Other than determining clashes of occasions, it is speaking to have an occasion member game plan system that universally streamlines the advantages of both occasion coordinators and clients, e.g., for sorting out a fair or a film celebration. Especially are the late studies on such occasion plan issue in static situations, a.k.a. disconnected situations, i.e., data of occasions and clients is completely given

III. PROPOSED WORK

We identify a new event-participant arrangement problem with extensive real-life applications, and propose a formal definition of global event-participant arrangement with conflict and capacity problem. we prove that GEACC is np-hard and design two approximation algorithms, min cost flow-GEACC and greedy. min cost flow-GEACC has 1an approximation ratio, where a is the maximum of users' capacities, but is not scalable due to its quartic time complexity. we further develop a more efficient greedy-based approximation algorithm, which guarantees 1 1þa worst-case approximation ratio. we also present an exact algorithm that utilizes an effective pruning rule to reduce redundant search space. we propose an online setting of the GEACC problem and design an online algorithm, online greedy- GEACC, with provable competitive ratio. we verify the effectiveness and efficiency of the proposed offline and online methods with extensive experiments on real and synthetic datasets.

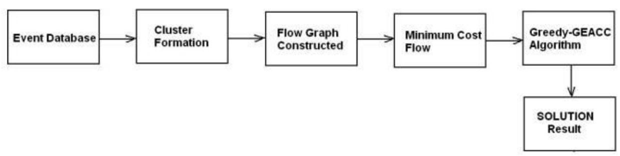

Figure 1: Proposed System Architecture

MODULES-

Event Database

Inputs area unit taken as per demand of user and keep in information. Information is often any event. These inputs area unit used for additional process for cluster formation. A knowledge event may be a relevant state transition outlined in a happening schema. Typically, event schemata area unit represented by pre- and post-condition for one or a collection of knowledge things. In distinction to Event condition action, that considers a happening to be a symbol, the information event not solely refers to the modification (signal), however describes specific state transitions, that area unit said in ECA as conditions. Considering information events as relevant information item state transitions permits shaping advanced event-reaction schemata for a information. Shaping information event schemata for relative databases is restricted to attribute and instance events. Object-oriented databases additionally support assortment properties that permit shaping changes in collections as information events, too.

Cluster Formation

Event database contains events and event manager Collection. But there are various types of events and different types of event manager that handles those events. So in cluster part all event and event manager are separate out with similar functionalities and work. In file systems, a cluster or computer memory unit could be a unit of disc space allocation for files and directories. to scale back the overhead of managing on-disk information structures, the file system doesn't allot individual disk sectors by default, however contiguous teams of sectors, referred to as clusters. On a disk that uses 512-byte sectors, a 512-byte cluster contains one sector, whereas a 4-kibibyte (KB) cluster contains eight sectors.

A cluster is that the smallest logical quantity of disc space that may be allotted to carry a file. Storing little files on a file system with massive clusters can thus waste area disc house space; such wasted disk space is named slack space. For cluster sizes that square measure little versus the common file size, the wasted house per file are statistically regarding 1/2 the cluster size; for giant cluster sizes, the wasted house can become larger. However, a bigger cluster size reduces accounting overhead and fragmentation, which can improve reading and writing speed overall. Typical cluster sizes vary from one sector (512B) to 128 sectors (64 KB).

A cluster needn't be physically contiguous on the disk; it should span over one track or, if sector interleaving is employed, could even be discontinuous at intervals a track. This could not be confused with fragmentation, because the sectors square measure still logically contiguous.

Flow Graph Constructed

All event and event managers are separate out with similar functionality and work generate all combinations between event and event managers. A flow graph could be a style of letter of the alphabet related to a group of linear pure mathematics or differential equations:[1][2] "A signal flow graph could be a network of nodes (or points) interconnected by directed branches, representing a group of linear pure mathematics equations. The nodes in an exceedingly flow graph area unit accustomed represent the variables, or parameters, and therefore the connecting branches represent the coefficients relating these variables to at least one another. The flow graph is related to variety of straightforward rules that alter each potential answer [related to the equations] to be obtained." "The word is much from standardized, and not standardization are often expected within the predictable future."

A designation "flow graph" that features each the Mason graph and therefore the Coates graph, and a range of alternative kinds of such graphs[8] seems helpful, and agrees with Abrahams and Coverlet's and with Henley and Williams' approach.

Minimum Cost Flow

Separate out with similar functionality and work with Common attribute like event type and event functionality. The minimum-cost flow drawback (MCFP) is associate optimization and call drawback to search out the most cost effective potential approach of causing a definite quantity of flow through a flow network. A typical application of this drawback involves finding the simplest delivery route from a mill to a warehouse wherever the road network has some capability and price associated. The minimum price flow drawback is one among the foremost elementary among all flow and circulation issues as a result of most different such issues is forged as a minimum price flow drawback and conjointly that it is resolved terribly with efficiency mistreatment the network simplex algorithmic program.

Greedy-Algorithm

The main idea of Greedy is to greedily add the most similar unmatched pair that does not conflict with existing matched pair into current matching at each iteration. Unlike MinCost- Flow-GEACC that resolves conflicts after obtaining temporary results, Greedy avoids conflicts from the first beginning.

Approximate Solutions For GEACC

In this section, we present two approximation algorithms for the GEACC problem. The first one is based on the minimum cost flow problem but is not scalable for large datasets. We then propose the second more efficient approximation algorithm

Minimum Cost Flow

Separate out with similar functionality and work with Common attribute like event type and event functionality.

The main idea of Greedy is to greedily add the most similar unmatched pair that does not conflict with existing matched pair into current matching at each iteration. Unlike MinCost- Flow-GEACC that resolves conflicts after obtaining temporary results, Greedy avoids conflicts from the first beginning

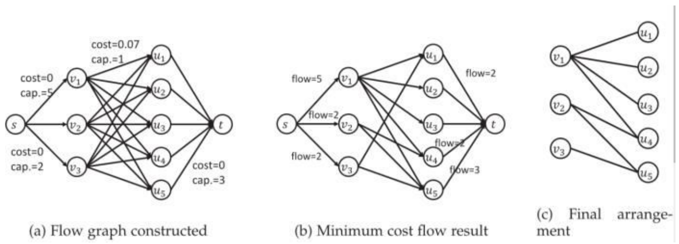

Figure 2: MinCost Flow GEACC.

Greedy Algorithm

MinCostFlow-GEACC could be inefficient when the scale of data is large. In this section, we present a more efficient set of computer instructions, Greedy. The main idea of Greedy GEACC is to greedily add the most almost the same the very best pair fv;ug that does not conflicts with existing matched pairs into the current matching at each cycle. Unlike MinCostFlow-GEACC that resolves conflicts after getting an (only lasting for a short time) result, Greedy avoids conflicts from the first beginning.

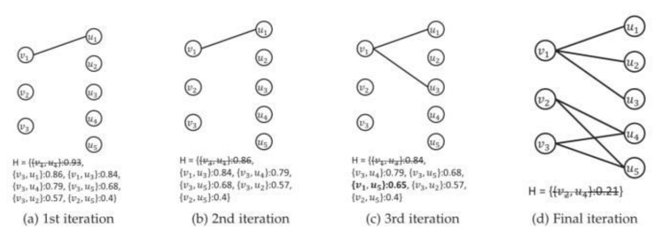

Figure 3: Illustrated example of Greedy

Algorithm 1. MinCostFlow-GEACC

input:V;U;fcvg;fcug;fl lvg;fl lug;CF

output: A feasible arrangement M

construct GF ¼ðNF;AFÞ;

foreach D Dmin to Dmax do

FD MinCostFlow(GF;D);

construct MD ; accordingly;

if MaxSumðMD ;Þ > MaxSumðM;Þthen

M; MD ; ;

foreach u 2 U do

L sorted list offvjm;ðv;uÞ¼1gin non-increasing order of simðl lv;l luÞ;

for i 1 tojLjdo

vL i the i-th element of L ;

if vL i does not conflict with u's matched events in M then

mðv;uÞ 1;

return M

Algorithm 2. Greedy

input: V;U;fcvg;fcug;fl lvg;fl lug;CF

output: A feasible arrangement M

H ; ;

foreach v 2 V do

unn v's first NN in U;

pushfv;unnginto H;

foreach u 2 U do

vnn u's first NN in V;

iffvnn;ug = 2 H then

pushfvnn;uginto H;

heapify H;

mðv;uÞ 0;8v 2 V;u 2 U;

while H 6¼;do

extract the most similar pairfv;ugfrom H;

if (cv > 0) and (cu > 0) and (v does not conflict with u's matched events) then

mðv;uÞ 1;

decrease cv;cu by 1;

if cv > 0 then

unn v's next feasible unvisted NN;

if unn9andfv;unng = 2 H then

pushfv;unnginto H;

if cu > 0 then

vnn u's next feasible unvisited NN;

if vnn9andfvnn;ug = 2 H then

23 pushfvnn;uginto H;

24 return M

EVALUATION FOR GEACC-

MinCostFlow-GEACC is not efficient enough according to our previous experiment results. Thus, we study the scalability of Greedy in this part. The results are shown in Figures. 5a and 5b. Specifically, we set jVj¼100;200;500;1000 respectively, and vary the size of jUj.Since jUj is relatively large, we set max cv to 200.Theother parameters are set to default. We observe that the memory cost of Greedy grows linearly with the size of data and is relatively small subtracting those consumed by input data. Also, the time cost of Greedy grows nearly linearly with the size of data. The results show that GreedyGEACC is scalable in both time and space. Effectiveness of approximate solutions. We next study the effectiveness of our approximate solutions, whose results are presented in Figures. 5c and 5d. Notice that since we need to find the exact solutions in this part of evaluation and Prune-GEACC is infeasible on large dataset, we set jVj¼5, jUj¼15 and cv Uniform½1;10. The other parameters are set to default. In Figure. 5c, we compare the approximated Max Sums returned by MinCostFlow-GEACC and GreedyGEACC with the optimal MaxSum. We first observe that when jCFj¼; MinCostFlow-GEACC returns the optimal matching, which is reasonable. We also observe that the MaxSums returned by Greedy are quite close to the optimal ones, indicating that Greedy returns quite good results in practice. Figure. 5d shows the running time of different algorithms. The results indicate that the two approximate solutions are very efficient compared with the exact solution. Therefore, in overall, our approximate solutions are both effective and efficient.

IV. EXPERIMENTAL RESULTS

1.Manage User Details - This window appears only for the admins console, i.e. only admin can add the event managers details and create his account by giving him his id or password. The admin gets the notification for the activities of the event managers. In this window the admin and edit and delete the updated event managers details by clicking edit or delete option.

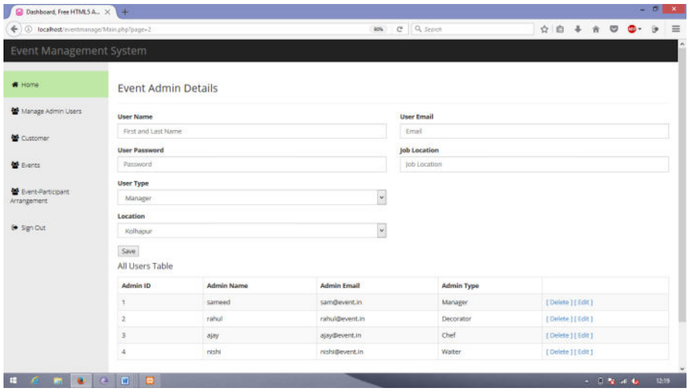

Figure 5: Manage User Details

2.Customer Details - This page is used to add the details of customer like name, address, contact number, email id and also his/her location. In this page the customer can also edit and delete his/her records if necessary

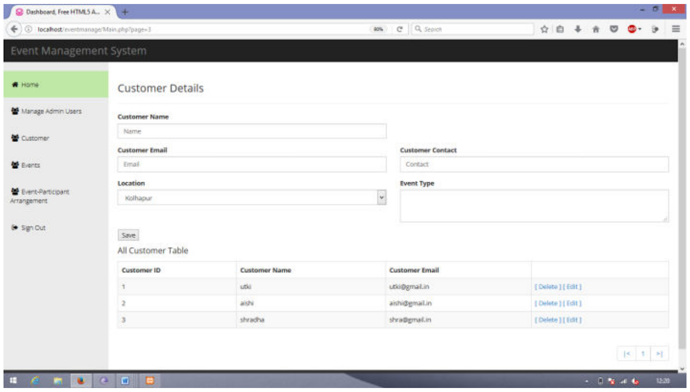

Figure 6: Customer Details

3. Event Details - This page is used to add the details of the events that the customer wants to attend and the location of the event and also the time of the event. In this page the customer can also add the work load.

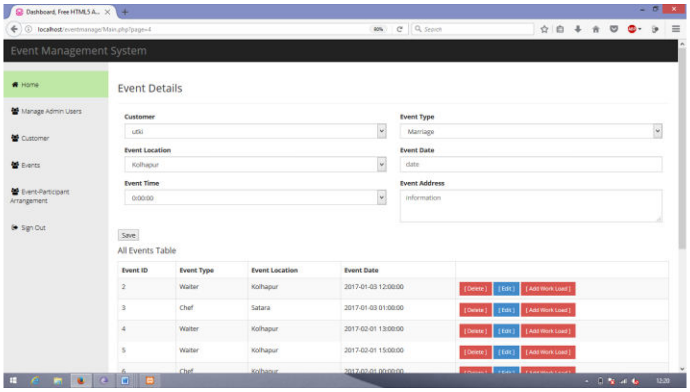

Figure 7: Event Details

4. Minimum Cost Flow -This page gives the final result by eliminating the conflicts among all the events held at same time and place so that workers can get work equally so single worker has to work beyond limits.

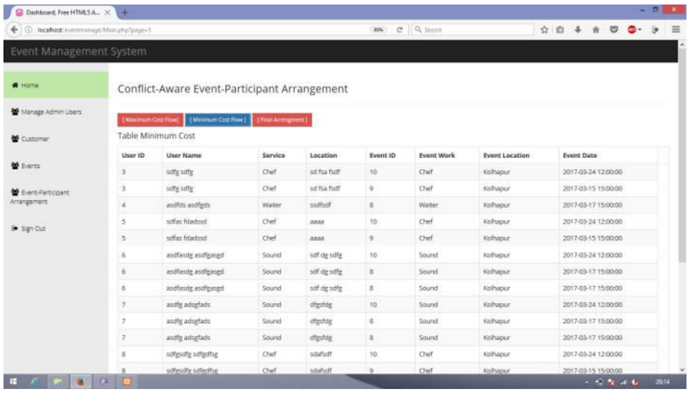

Figure 8: Minimum Cost Flow

5. Final Agreement -This page gives the final result by eliminating the conflicts among all the events held at same time and place so that workers can get work equally so single worker has to work beyond linits.

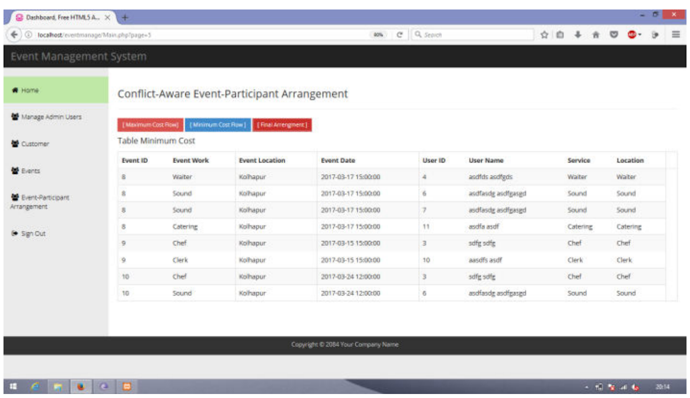

Figure 9: Final Aggreement

V. CONCLUSION AND FUTURE WORK

In this paper, we tend to determine a completely unique event-participant arrangement downside known as international Event-participant Arrangement with Conflict and capability. We tend to first analyze our variations compared with ancient matching issues and prove the NP hardness of our downside. Then, we tend to style a precise algorithmic program and 2 approximation algorithms. the precise algorithmic program is efficient for tiny datasets by means that of a pruning rule. The MinCostFlow-GEACC approximation algorithmic program isn't ascendible to giant datasets,and the Greedy approximation algorithmic program runs significantly quicker than MinCostFlowGEACC whereas guarantees a similar order of approximation magnitude relation. In addition to the offline setting, we conjointly study {the online|the we tend tob|the net} state of affairs of GEACC, called OnlineGEACC, wherever users arrive on the EBSN platforms in an internet approach. We tend to any propose a competitive-ratio-guaranteed on-line algorithmic program for OnlineGEACC, called OnlineGreedy. We conduct intensive experiments that verify the efficiency, effectiveness and quantifiability of the projected approaches.

Conflict-Awareness Event-Handling can be futher made on android app so that it can be used by any user or customer anywhere anytime as now a days android phones are used daily and almost everyone has an android smartphone.This project can be also used in hospitals where this project can inform whether the doctors are available or particular staff are available or not. This project can also be used in small and large scale industries.

REFERENCES

[1] J. She, Y. Tong, L. Chen and C. C. Cao, "Conflict-Aware Event-Participant Arrangement and Its Variant for Online Setting," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2281-2295, Sept. 1 2016.

[2] K. Li, W. Lu, S. Bhagat, L. V. S. Lakshmanan, and C. Yu, "On social event organization," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2014, pp. 1206–1215.

[3] J. She, Y. Tong, L. Chen, and C. C. Cao, "Conflict-aware eventparticipant arrangement," in Proc. 31st IEEE Int. Conf. Data Eng., 2015, pp. 735–746.

[4] J. She, Y. Tong, and L. Chen, "Utility-aware social eventparticipant planning," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2015, pp. 1629–1643.

[5] Y. Tong, J. She, and R. Meng, "Bottleneck-aware arrangement over event-based social networks: the max-min approach," World Wide Web J., to be published.