

Offline Matching Approximation Algorithms in Exchange Markets

Zeinab Abbassi
 Department of Computer Science
 University of British Columbia
 2366 Main Mall
 Vancouver, Canada
 zeinab@cs.ubc.ca

Laks V. S. Lakshmanan
 Department of Computer Science
 University of British Columbia
 2366 Main Mall
 Vancouver, Canada
 laks@cs.ubc.ca

ABSTRACT

Motivated by several marketplace applications on rapidly growing online social networks, we study the problem of efficient offline matching algorithms for online exchange markets. We consider two main models of one-shot markets and exchange markets over time. For one-shot markets, we study three main variants of the problem: one-to-one exchange market problem, exchange market problem with short cycles, and probabilistic exchange market problem. We show that all the above problems are NP-hard, and propose heuristics and approximation algorithms for these problems. Experiments show that the number of items exchanged will increase when exchanges through cycles are allowed. Exploring algorithms for markets over time is an interesting direction for future work.

Categories and Subject Descriptors

F.2.2 [Theory of Computation]: Algorithms & Complexity—Nonnumerical Algorithms; H.m [Information Systems]: [Miscellaneous]

General Terms

Algorithms, Performance

Keywords

Social Networks, Exchange Markets, Recommendation Algorithms, Set Packing, Edge Partitioning

1. INTRODUCTION

Online social networks are rapidly developing. Users spend increasing amounts of time on social network websites like MySpace and Facebook. A social network is a good framework to implement a market among users for exchanging items. Examples of existing exchange markets on the Web are peerflix.com and readitswapit.co.uk.

Motivated by the above and in order to improve the quality of user experience in exchange markets, we study optimal matching algorithms for online exchange markets. In particular, we propose two main models of exchange markets:

(i) *One-shot markets* in which each user receives an item in return for the item which has given away in the same time; and (ii) *Over-time markets* in which a user may not receive an item in return for her given item, but earn some points

that can be spent on another item later. In these systems, our goal is to find a collection of exchanges amongst users that maximize the number of items exchanged.

2. MODEL

In an online exchange market a transaction happens at the same time as the user declares her intent to acquire the item. However, in an offline market, the user states her intent to the system by listing her wishes in a wishlist, and the system notifies her about a possible exchange later through an offline message.

One-Shot exchange markets. Let U be a set of users and I be a set of items. For each user $u \in U$, let S_u (a.k.a. item list) be the set of items that user u owns and W_u (a.k.a. wish list) be the set of items that user u requires.

Simple exchange markets (SIMPLEMARKET). In a simple exchange market, we only match users one-by-one, i.e. in each transaction, two users u and v can exchange two items i and j . In an instance of the the simple market problem, we are given a set of users U with the item lists S_u and wish lists W_u for each user $u \in U$, and our goal is to find the maximum cardinality set of exchanges $[(u, i), (v, j)]$ where $i \in S_u, j \in W_u, j \in S_v$ and $i \in W_v$.

Exchange markets through cycles. In an offline exchange market, we can find cycles of size larger than 2. Clearly, we can exchange more number of items through these types of cycles compared to the one-to-one exchanges. We also assume that each user has at most one copy of each item i in their item list. Similarly, each user wishes for at most one copy of any item. In the CYCLEMARKET problem, our goal is to find a set of conflict-free cycles: $[(u_1, i_1), (u_2, i_2), (u_3, i_3), \dots, (u_k, i_k)]$ where $i_1 \in S_{u_1}, i_1 \in W_{u_2}, i_2 \in S_{u_2}, i_2 \in W_{u_3} \dots, i_k \in S_{u_k}, i_k \in W_{u_1}$ such that $[(u, i), (*, *)]$ appears at most once in the set of cycles to avoid conflicts. Our goal here is to maximize the number of items involved in exchanges, thus maximizing the number of transactions. Note that a transaction over a cycle can happen if all of its exchanges happen. As a result, it is desirable to discover short cycles, and solve the short cycle exchange market problem, denoted by SHORTCYCLEMARKET problem. In this problem, given a number $p \geq 3$, we want to find a conflict-free set of exchange cycles of at most p exchanges each.

Probabilistic exchange markets (PROBMARKET). A more refined model for exchange takes into account the probability of a user being involved in an exchange, perhaps based on the reputation of the user she must trans-

act with, their location information, friendship, etc. As a result, we have a graph with some probability on each edge. Assuming that the probability of realizing each exchange is independent of other exchanges, in the PROBMARKET problem, our goal is to find a set of cycles with the maximum expected number (or expected value) of edges that are covered. Note that the probability of a cycle is the product of the probability of its edges.

3. HARDNESS RESULT

The above exchange market problems are related to the kidney exchange problems studied in [1], however the slight difference which will be explained, makes the problem much harder. The property that discriminates the kidney exchange from our problem is that each patient needs exactly one kidney and each donor wants to donate one kidney.

We show that despite the fact that the kidney exchange problem with cycles of size 2 is polynomial-time solvable, the SIMPLEMARKET problem is NP-hard. To prove this hardness result, we give a reduction from a variant of edge disjoint partitioning problem 4-partite graphs to the SIMPLEMARKET problem. We first show the NP-hardness of that variant using a reduction from the edge-partitioning of tripartite graphs [4] and then give a reduction from 4CYCEDGEPART to the SIMPLEMARKET problem. Using similar methods, we can also show NP-hardness of the SHORTCYCLEMARKET and PROBMARKET problems.

4. OUR ALGORITHMS

The one-shot exchange market problem is a special case of the weighted k -set packing problem, in which, given a collection of sets, each of which has an associated weight and contains at most k elements drawn from a finite base set, our goal is to find a collection of disjoint sets of maximum total weight. The restriction to sets of size at most k properly includes multi-dimensional matching, which is a generalization of the ordinary graph matching problem. Here, Using ideas from known algorithms for the k -set packing problem, we design heuristics and approximation algorithms for the above exchange market problem (including the PROBMARKET problem). To formalize the one-shot exchange market problem as a special case of the set packing problem, we consider sets as the exchange cycles and define the elements of the sets to consist of the user, the item, and the act of giving or wishing. The *weight* of each set in the given problem is the number of items exchanged on the corresponding cycle, or the total expected value of the exchanges on the cycle. Now, using ideas from algorithms for the set packing problem, we consider the following heuristic algorithms, some of which have provable performance guarantee for all of the above problems.

A Greedy Algorithm. Given the hardness result, we look for approximation algorithms for the problem. One approach is the following greedy algorithm: at each step, we find the best exchange cycle C with the maximum weight. In order to find the best cycle, we can try all short cycles and then pick the cycle with the maximum weight. Then add C to the collection of cycles. Remove all the edges that are in conflict with C . In [3], it is shown that the performance of the above greedy approach to the weighted k -set packing problem is a $2k$ -approximation.

Maximal/Greedy Algorithm. To improve the running time of the greedy algorithm, we design the following heuristic. In this approach, in order to find the cycles, we can perform the breadth first search(BFS) algorithm from each node v , and find a cycle C_v by identifying backward edges in the BFS algorithm. Then we pick the best cycle (i.e. cycle C_u with the maximum weight) and add it to the set of conflict-free cycles. We repeat this process until we get a maximal collection of cycles.

Local Search Algorithm. In this algorithm, we iteratively attempt to replace a small subset of the current solution by some set of cycles that result in a larger total weight. More precisely, for any exchange cycle C that is not already selected, we try to add C and remove all the conflicting edges. If the total weight of the collection increases by a factor β , add C to the collection and update the collection of cycles by removing all conflicting cycles from it. We perform this procedure until no local improvement is possible. In [2], the authors prove that the local search algorithm gives a $k - 1$ -approximation algorithm.

Greedy/Local Search. Another heuristic algorithm is to run the greedy algorithm to find a collection of cycles and then run the local search algorithm starting from this collection. In [3], it is proved that this algorithm is a $2(2k + 1)/3$ -approximation algorithm and show that this is asymptotically tight.

5. EXPERIMENTAL RESULTS

Experiments are done on a set of synthetic data. We randomly generated the data so that the number of items in the item lists and wish list follow a power law distribution. We observed that the number of users involved in transactions increases when cycles of length larger than 2 are allowed. This increase for the maximal/greedy algorithm is 4% to 6%, and for the local search algorithm is 6% to 10%. The experimental results are much better than the worst-case approximation factors.

6. FUTURE WORK

In this paper, we study various models for the exchange market problem. Two interesting research directions are designing improved approximation algorithms for these problems and comparing the quality of these algorithms on some real data sets. Finally, designing efficient algorithms for over time markets is left as an interesting future work. These markets can be modeled as repeated variants of the one-shot exchange markets. In particular, the use of *virtual points* (that can be gained or redeemed by users in exchange of items) can decrease the expected waiting time of users. Exploring these ideas is an interesting research direction.

7. REFERENCES

- [1] D. J. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *ACM Conference on Electronic Commerce*, pages 295–304, June 13-16 2007.
- [2] E. M. Arkin and R. Hassin. On local search for weighted k -set packing. In *ESA 1997*, pages 13–22, 1997.
- [3] B. Chandra and M. Halldorsson. Greedy local improvement and weighted set packing approximation. In *SODA 1999*, pages 169–176, 1999.
- [4] I. Holyer. The np-completeness of some edge-partition problems. *SIAM Journal of Computing*, 10(4):713–717, November 1981.