Lemma 4.3. *The Greedy algorithm is a* $1/2$*-approximation algorithm for the* Fairness Maximization *problem with category constraints.*

Proof. We will prove the theorem using induction on the steps of the Greedy algorithm. Let P_k denote the first k items selected by Greedy, for $k \leq K$. By definition, the k items must belong to k distinct categories. Let P_k^* denote the items selected by the optimal algorithm for the corresponding k categories. Also, let $L_k = \text{SAT}_G(P_k)$ and $L_k^* = \text{SAT}_G(P_k^*)$ denote the corresponding sets of users in G that are satisfied by the items in P_k and P_k^*. We will show that $|L_k^* \setminus L_k| \leq |L_k|$ for all $1 \leq k \leq K$. Since $|L_k^* \setminus L_k| \geq |L_k^*| - |L_k|$, it follows that $|L_k| \leq \frac{1}{2}|L_k^*|$, which proves our claim when $k = K$.

For $k = 1$ our claim is trivially true, since $|L_1^*| \leq |L_1|$, by definition of the Greedy algorithm. Assume that it is true for $k = j$. Let N_{j+1} denote the users that are satisfied by the item selected by the greedy algorithm for category $j+1$, and let N_{j+1}^* denote the corresponding set of users for the item selected in the optimal solution. We have that $L_{j+1}^* = L_j^* \cup N_{j+1}^*$. Therefore,

$$\begin{aligned} |L_{j+1}^* \setminus L_{j+1}| &= |(L_j^* \setminus L_{j+1}) \cup (N_{j+1}^* \setminus L_{j+1})| \\ &\leq |L_j^* \setminus L_{j+1}| + |N_{j+1}^* \setminus L_{j+1}| \\ &\leq |L_j^* \setminus L_j| + |N_{j+1}^* \setminus L_j| \\ &\leq |L_j| + |N_{j+1} \setminus L_j| \\ &= |L_{j+1}| \end{aligned}$$

The last inequality follows from the inductive hypothesis and the property of the Greedy algorithm that it always selects the item that maximizes the additional number of users that are satisfied by the package. $\square$

For the general case where we select k_j items from each input category C_j, we create k_j replicas C_j^l, $l = 1, \ldots, k_j$, for each input category C_j. We then run Greedy on this dataset. Note that the same item cannot be selected multiple times since it will have coverage zero.

4.2.2 Distance constraints

We now consider Problem 3, where we want the items in the package to satisfy distance constraints. We can still adapt the basic Greedy algorithm for this case, by considering as candidate items only the items that when added to the existing solution satisfy the distance constraints. We cannot prove any guarantees for this algorithm. Actually, there are cases when the Greedy algorithm may terminate before finding K items. We now propose two heuristics, and one exact algorithm that take advantage of spatial partitioning and indexing to reduce the search space.

A space-partitioning approach. For this algorithm, we divide the space by a grid, such that each cell has width and height $\epsilon/\sqrt{2}$, that is, a diagonal of length ϵ. Thus, any two items inside the same cell are within the ϵ-distance threshold. For each cell, we then run one instance of the Greedy algorithm, considering only the items that appear in the cell, and report the best solution. This approach greedily solves one local problem per cell, however, it fails to consider packages that include items from different cells. We refer to this algorithm as PartitionGreedy.

Grid-based greedy algorithm. The second approach is again a greedy method based on a space partitioning. We partition the space by a grid again, but this time each cell has side length ϵ. Then, we run again the Greedy algorithm for each cell, but this time, we allow the search to extend neighboring cells if necessary. Let L_j denote the j-th cell that is examined by the algorithm. The first item i in L_j is selected greedily. We then take advantage of the grid to reduce the number of candidate items to consider. It is easy to see that any item in L_j can only form valid packages with items inside L_j, or its direct neighbors (at most 9 cells in total). Furthermore, as more items are selected, the 9-cell search space is further reduced.

Figure 1 shows an example with a 4×4 grid. Suppose the Greedy algorithm on cell L_6 selects i_1 as the first item. For the second item, it will consider as candidates only items that fall in L_6 or in one of the 8 cells surrounding L_6 (i.e., the cells $\{L_{1\text{–}3}, L_{5\text{–}7}, L_{9\text{–}11}\}$). If the second item selected is i_2, which falls in L_2, the third item cannot be selected from cells $L_{9\text{–}11}$ because all items in them are further than ϵ from i_2. Therefore, for the third item, we only have to consider cells $\{L_{1\text{–}3}, L_{5\text{–}7}\}$. If the third item is i_3 in L_7, the fourth item can only be selected from cells $\{L_{2\text{–}3}, L_{6\text{–}7}\}$, and so on. Therefore, having partitioned the items based on the grid, we can dynamically prune the search space of candidate cells and items. We will refer to this algorithm as GridGreedy.

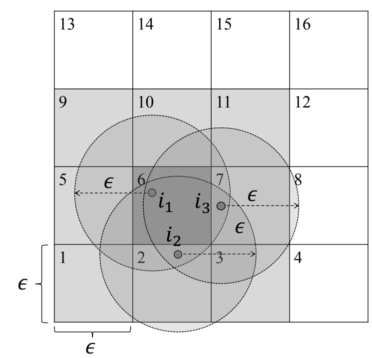

Figure 1: Grid Example

Grid-based backtracking algorithm. It is possible to change the GridGreedy algorithm to an exhaustive search optimal algorithm that generates all valid item combinations based on the distance constraint. This algorithm uses the grid to minimize the cost of generating item combinations by early-eliminating items that are too far from the current (partial) combination. The *grid-based backtracking algorithm* runs one search for each item i in $\mathcal{I}$, by taking i as the first item in the combination. Then, it considers all possible items to add next to the package (thus, generating a search tree rooted at i), by examining only the neighboring cells of the one that includes i. For each item added, the search space for the next ones to add is restricted. As soon as a complete combination is formed, its fairness score is measured, and finally the best package overall is returned. We refer to this algorithm as GridOptimal.

5. EXPERIMENTS

The goals of the experiments are three-fold. First, to understand the tradeoff between fairness and quality. We want to quantify the effect on quality when optimizing for fairness and vice versa, and understand how the hardness of the in-

Acknowledgements

This work was supported by grant 17205015 from Hong Kong RGC and by Marie Curie Reintegration Grant projects titled JMUGCS and LBSKQ which have received research funding from the European Union.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] S. Amer-Yahia, F. Bonchi, C. Castillo, E. Feuerstein, I. Méndez-Díaz, and P. Zabala. Composite retrieval of diverse and complementary bundles. *IEEE TKDE*, 26(11):2662–2675, 2014.

[3] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *PVLDB*, 2(1):754–765, 2009.

[4] S. J. Brams. Fair division. In *Computational Complexity*, pages 1073–1080. Springer, 2012.

[5] C. Chekuri and A. Kumar. *Maximum Coverage Problem with Group Budget Constraints and Applications*, pages 72–83. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[6] T. Deng, W. Fan, and F. Geerts. On the complexity of package recommendation problems. *SIAM J. Comput.*, 42(5):1940–1986, 2013.

[7] J. Gorla, N. Lathia, S. Robertson, and J. Wang. Probabilistic group recommendation via information matching. In *WWW*, pages 495–504, 2013.

[8] N. G. Hall and D. S. Hochbaum. A fast approximation algorithm for the multicovering problem. *Discrete Applied Mathematics*, 15(1):35 – 40, 1986.

[9] D. K. Herreiner and C. D. Puppe. Envy freeness in experimental fair division problems. *Theory and Decision*, 67:65–100, 2009.

[10] D. S. Hochbaum. Approximation algorithms for np-hard problems. *SIGACT News*, 28(2):40–52, June 1997.

[11] Q. Hua, Y. Wang, D. Yu, and F. C. M. Lau. Set multi-covering via inclusion-exclusion. *Theor. Comput. Sci.*, 410(38-40):3882–3892, 2009.

[12] Q. Hua, D. Yu, F. C. M. Lau, and Y. Wang. Exact algorithms for set multicover and multiset multicover problems. In *ISAAC*, pages 34–44, 2009.

[13] A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 596–627, 2007.

[14] K. Li, W. Lu, S. Bhagat, L. V. S. Lakshmanan, and C. Yu. On social event organization. In *KDD*, pages 1206–1215, 2014.

[15] X. Liu, Y. Tian, M. Ye, and W.-C. Lee. Exploring personal impact for group recommendation. In *CIKM*, pages 674–683, 2012.

[16] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of user. In *ECSCW*, pages 199–218, 2001.

[17] A. G. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM TOIS*, 29(4):20, 2011.

[18] S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. Recommending packages to groups. In *International Conference on Data Mining (ICDM)*, 2016.

[19] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[20] S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. Exploiting group recommendation functions for flexible preferences. In *ICDE*, pages 412–423, 2014.

[21] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[22] H. Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948.

[23] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, Jan. 2009.

[24] P. Tsaparas, A. Ntoulas, and E. Terzi. Selecting a comprehensive set of reviews. In *KDD*, pages 168–176, 2011.

[25] S. Vargas, P. Castells, and D. Vallet. Intent-oriented diversity in recommender systems. In *SIGIR*, pages 1211–1212, 2011.

[26] M. Xie, L. V. S. Lakshmanan, and P. T. Wood. Breaking out of the box of recommendations: from items to packages. In *RecSys*, pages 151–158, 2010.

[27] M. Xie, L. V. S. Lakshmanan, and P. T. Wood. Generating top-k packages via preference elicitation. *PVLDB*, 7(14):1941–1952, 2014.

[28] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *ICDE*, pages 1299–1302, 2009.

[29] Q. Yuan, G. Cong, and C.-Y. Lin. Com: A generative model for group recommendation. In *KDD*, pages 163–172, 2014.

[30] T. Zhu, P. Harrington, J. Li, and L. Tang. Bundle recommendation in ecommerce. In *SIGIR*, pages 657–666, 2014.

[31] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, pages 22–32, 2005.