# Crowdrebate: A Platform for Grouping Orders for Getting More Rebate for Customers

Nian Chen [†], Weijie Sun [†], Wangze Ni [†], Zhao Chen [†], Jiale Gu [†], Peng Cheng [*], Lei Chen [†], Xuemin Lin [#,*]
[†]*The Hong Kong University of Science and Technology, Hong Kong, China*
nchenaf@connect.ust.hk, {wsunan, wniab, zchenah, leichen}@cse.ust.hk, jialgu@ust.hk
[*]*East China Normal University, Shanghai, China*
pcheng@sei.ecnu.edu.cn
[#]*The University of New South Wales, Australia*
lxue@cse.unsw.edu.au

***Abstract*—With the flourishing development of e-commerce platforms, there are various promotion operations, such as offering coupons with a high threshold to stimulate customers to buy more products all at once. Nevertheless, customers always find that the price of the products they intend to order is below the coupons or free delivery service threshold. In order to enjoy these instant benefits, they often need to spend a lot of time looking for others with the same need to place an order together.**

**In order to address the above issues, we develop a Crowdrebate platform that collects users' requests and groups them into a set of orders qualifying to use the suitable coupons by the Crowdrebate algorithm, after that, Crowdrebate delivers products to different receivers in an order, aiming at saving users' shopping time and helping users get more rebate from online promotions. Crowdrebate is equipped with several novel techniques, including the Crowdrebate algorithm grouping orders for the maximum instant rebates, lightweight web crawling, novel items recommendation, and personalized data visualization. We introduce the sketch of the system architecture of Crowdrebate and demonstrate various scenarios via several case studies.**

## I. Introduction

With the flourishing development of e-commerce platforms, online shopping has become a part of people's lives. To further stimulate online consumption, e-commerce platforms arrange various promotion campaigns on special days, such as Black Friday [1] in Western countries and Double Eleven [2] in China [3]. Along with numerous online promotions, some intermediate platforms have emerged in the market, such as Groupon [4] and Dealmoon [5], which aim to help users sort out the hot promotions of various e-commerce platforms.

Unfortunately even some users know those promotions, they find it hard to meet the spending threshold to use a coupon or to enjoy the free delivery service. Therefore, users will intentionally seek for others, often their friends, to place the order together to meet the threshold. Because all the e-commerce platforms support only one shipping address for one order, when multiple users place an order together, one of the users will inevitably receive all the products first and then distribute them to others respectively. These complex processes are unreliable and time-consuming for consumers. What is even more damaging to consumers' motivation to shopping is that increasingly complicated e-commerce promotion activities impose extra workloads to customers, since multiple complex coupons rules may change on a daily basis [6].

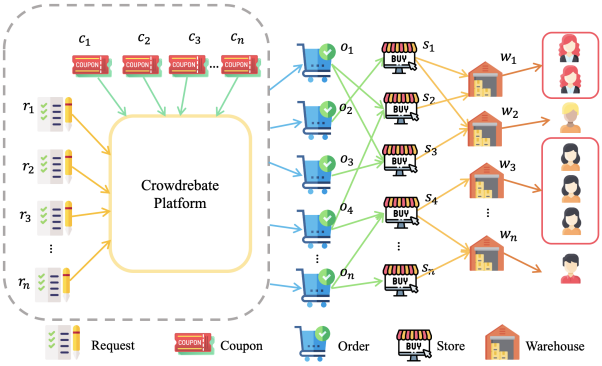



Fig. 1: The operation process of Crowdrebate

Under these circumstances, our demo-Crowdrebate firmly addresses consumers' pain points. Crowdrebate collects users' purchasing requests and group their requests even users do not know each other into orders to maximize total benefit. Meanwhile, we give the exposure of online retailers' promotions so that more users would participate in promotions. To summarize, not only do we dedicate ourselves to help users get more benefits, we also devote ourselves to help e-commerce platforms fill more orders, increase platforms' revenue, and ultimately achieve a bilateral win-win situation.

Our platform operates as follows as shown in Figure 1. Users can post their requirements corresponding to the information of the desired product. Based on the received requests, we make an order grouping and match the suitable coupons, using our *crowdrebate algorithm* [7], to maximize each order's rebate deducting the delivery costs. Because we place the orders for consumers, the products are first sent to our warehouses from the online stores, and then we will distribute the products according to the users. The site of our warehouses considers the distribution of the user loads as evenly as possible. The specific system architecture we will introduce in the next section.

Generally, Crowdrebate consists of three components. The user interface provides service for users to post requests and track data. The algorithm function supports our core features, group ordering and recommendation. The data manager han-

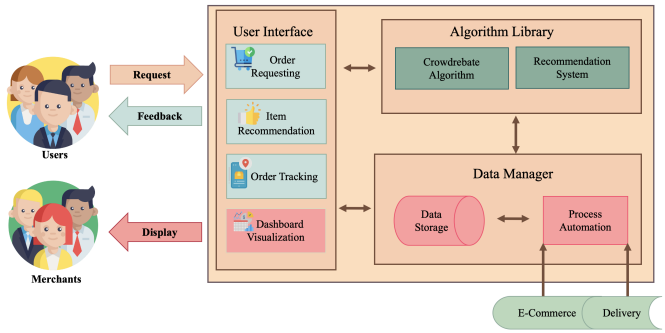



Fig. 2: The architecture of crowdrebate

dles business logic and data storage.

Thus, our demo has following contributions:

- We develop a platform for users to post for group ordering and track realtime information on mobile phones.
- We design two algorithms to combine orders for maximum rebates under different scenarios.
- We equip the platform with process automation and recommendation to handle large scale data and enrich users experience.

## II. System Architecture

Figure 2 depicts the architecture of Crowdrebate platform. The user interface module runs on the client-side, enabling both general users and merchant users to get corresponding services. The other three core modules, process automation, crowdrebate algorithm [7], and recommendation system, run on the server host and provide support for the core functionalities in Crowdrebate. We introduce the technical details of these modules in the rest part of this section.

### A. User Interface

There are two types of users interacting with the user interface module, "general users" and "merchant users". General users, who seek group buying to meet the threshold of rebates, can request and track their orders with the interface while getting a personalized recommendation. Merchant users from online shopping platforms can benefit from the data dashboard to gain a better understanding of user portraits. This module consists of the following four functions.

*1) Order Requesting:* Users in need for group shopping can post their requirements with this order requesting function. In addition to the basic product information, another critical information goes to the expected time slots and delivery addresses, impacting the rounds and delivery costs for the crowdrebate algorithm to run.

*2) Order Tracking:* The order tracking function enables users to manage all historical orders. No matter whether one's item is waiting in the pool, being processed for a group order, or already on delivery, users can track real-time status operated by the process automation module.

*3) Item Recommendation:* The item recommendation functions show users products that best fit personal demands. Crowdrebate has top coupons and recommended products on the home page, where all users share the same information of top coupons while the recommended products are personalized.

*4) Dashboard Visualization:* Crowdrebate opens API and provides a visualized data dashboard to cooperating online shopping platforms. With the visualized dashboard, merchant users can access to internal data depicting user portraits, product sales, and coupon popularity.

### B. Crowdrebate Algorithm

Crowdrebate algorithm [7] can place orders for users' best interests and works as one of the most crucial components in our platform. Considering potential combinations of coupons and delivery, the order grouping problem is proved to be NP-hard. To tackle this problem under different scenarios, Crowdrebate proposed two algorithms, the order-first algorithm and the delivery-first algorithm. Users can freely choose from these algorithms which best fits their needs, and their requests will wait in two pools to process separately.

**The Order-First Algorithm.** In most cases, coupons with strict thresholds usually guarantee tempting rebates that should way cover the delivery costs. Under such a scenario, we propose the order-first algorithm to place orders. Generally speaking, this algorithm greedily packs orders for a maximum amount of total rebates while ignoring delivery costs. For each coupon, we traverse requests in the eligible pool and calculate the differential rebate amount when compared to placing an order separately, and one of requests with the largest amount will be randomly chosen to integrate into one order until for any request the differential rebate amount fails to increase. However, the number of coupons used in a newly merged order may exceed the upper limit since their amounts are calculated separately before the merger. We introduce *dif procedure* to tackle extra coupons while computing for the differential rebate $r_d = r_{ab} - r_a - r_b$, where $r_a, r_b$ refers to the separate rebates for order $a, b$, and $r_{ab}$ refers to the total rebate after merging two orders. After packing requests, we need to place the order and assign one single address as the delivery destination. While this algorithm doesn't take the delivery into account before, it assigns a warehouse to the order to minimize the cost of delivery now.

**The Warehouse-First Algorithm.** The order-first algorithm performs well in general cases, however, it still suffers from two drawbacks. The first one is due to its lack of consideration for delivery costs, and once the delivery cost gets much higher, this algorithm performs poorly. The second defect results from the time complexity introduced by the error correction dif procedure, which can be fatal when handling peak shopping festival traffic. As a result, a more efficient approach, the warehouse-first algorithm, is proposed to tackle the large scale data and non-negligible delivery costs. Instead of directly combining requests for higher rebates, we first group those requests sharing the same warehouse. Only then will we merge requests within the same warehouse for higher rebates. The preprocessing procedure divides the integral data set into smaller groups to constrain the batch size and running time, and grouping by delivery destination guarantees relatively

small delivery costs. According to our experiment results, the warehouse-first algorithm is more efficient and can produce a better result for higher delivery costs, while the order-first algorithm stands out when rebates are profitable enough to cover delivery costs. Before users can freely choose from algorithms above, we run the order-first algorithm compulsorily for the cold start stage. This is because the request volumn is too small to be split into warehouses, and the order-first algorithm can handle such small data scale while bringing users more rebates. The experimental results can be found in our research paper [7].

### C. Process Automation

Crowdrebate uses the process automation module to complete the whole procedure from picking items to pay for our server's orders. Basically, the module comprises two sub-modules, the web crawler retrieving the information of products and coupons, and the order bot to automatically place orders.

**The Web Crawler.** Currently, two types of information will be crawled, respectively, products and coupons. For the product information, the spider bot crawls the associated coupon information and the meta data of the product for recommendation system. For the coupon information, we take the initiative to crawl information from specialized coupon websites like Etao [8] to enrich our available coupon pools.

**The Order Bot.** The crowdrebate algorithm runs in rounds periodically and produces final grouped orders each round. Thus we develop order bot to handle those orders automatically. Since some exclusive coupons in different orders may violate each other, we introduce multiple accounts to place those exclusive orders, enabling us to run scripts in parallel.

### D. Recommendation System

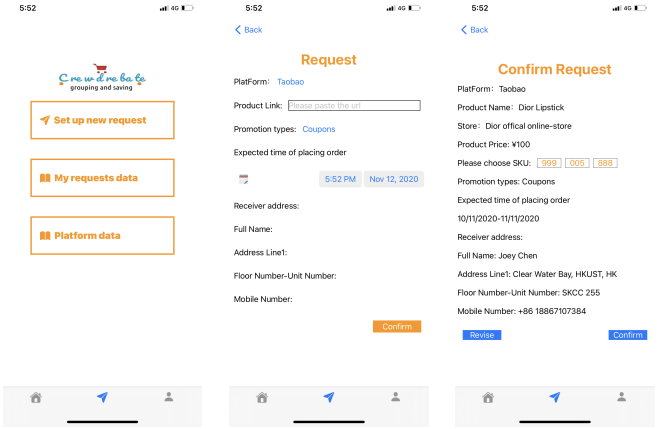
The recommendation system module is one of the critical features of Crowdrebate. Considering the online shopping scenario, the similarity of items is relatively stable to the user's interest. Therefore, we adapt the item-based collaborative filtering [9] to recommend similar products. Since users on Crowdrebate pursue higher rebates, we improve the algorithm to better meet their demands. We reserve the same process to match products with high similarities and for the ranking process, however, we rank them by the highest rebates instead of similarity.

## III. Demonstration Overview

In this section, we describe different kinds of scenarios that users commonly encountered in the Crowdrebate platform, including a new request submission, the view of top deals and products, status checking, and data analyses. A demonstrate video of Crowdrebate is accessible to watching on the YouTube under https://youtu.be/zdMbzNZkXW4.

### A. Scenarios 1: Set up request

**As a regular user** completes the registration on our platform, has the certain product intent to buy, and clicks on the




(a) Set up a request (b) Fill in information (c) Information confirm

Fig. 3: The flowchart of "Set up a request".

Set up new request button, as shown in Figure 3(a), she/he would enter the page which guides her/him to fill in the following information in Figure 3(b). At first, the user needs to select the e-commerce platform she/he wishes to order from. In order to simplify the user's task and to prevent the platform from placing the wrong order of product, the user only need copy the URL link of the product he wants to buy, and the platform will automatically crawl the product information, such as price, store, promotions, etc. If there is more than one Stock Keeping Unit SKU [10] for a product, we also crawl to all the SKU information and ask the user to make a choice. Next, the user needs to select time slot she/he wants the platform to place the order. Finally, the user fills in the delivery address. After confirming the final information twice shown in Figure 3(c), the platform will start to deal with the request at the time specified by the user. As we mentioned in the section before, we would run the *order-first algorithm* compulsorily for the cold start. In our demonstration, we adopt order-first algorithm to group requests.

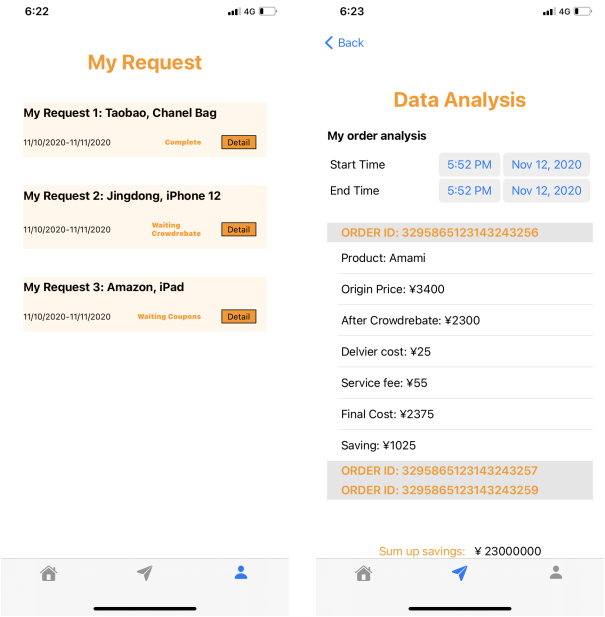
### B. Scenarios 2: Check request status and data

After completing to set up a request, the user can check all the requests she/he made on our platform on the personal page in Figure 4(a) and the status of each request.

Note that, if the user has an interest in the data of order, she/he can also see specific data for each request in the My request data page (Figure 4(b)), like the original price, the price after the platform placed the order for them, the shipping cost, the platform's service charge, and the savings after deducting various costs. The user also can choose to see the sum of the data over a period of time.

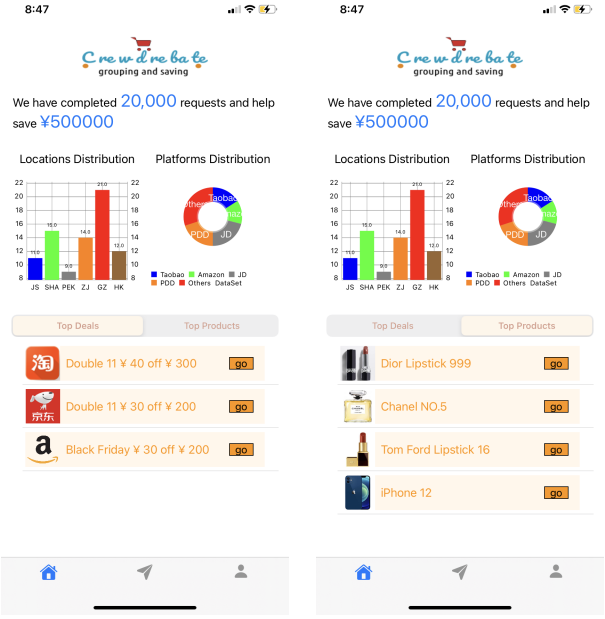
### C. Scenarios 3: View Top deals and top products

When not sure about recent promotions of e-commerce platforms, the user can use the home page in Figure 5(a) to see our daily updates of platforms' top deals to save time. While having no clear shopping target, the user can view the top products that platform makes personalized recommendations based on the previous purchase data (Figure 5(b)).




(a) Status of requests (b) Personal data

Fig. 4: View status of requests and top products.




(a) Today's deals (b) Personalized recommendations

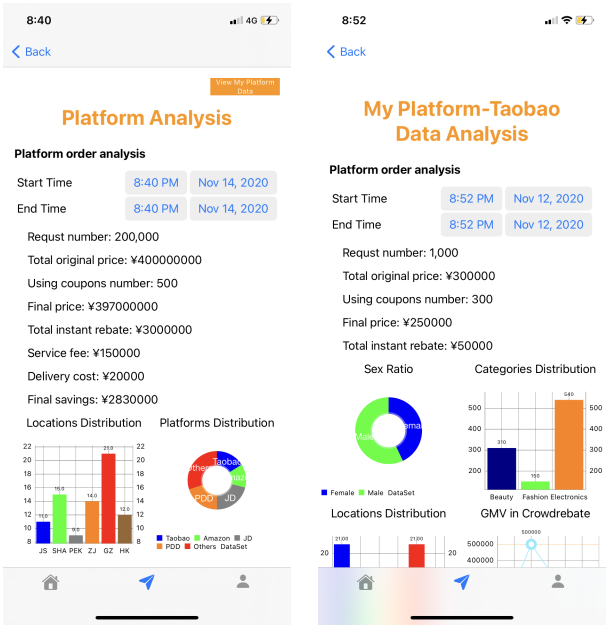
Fig. 5: View Today's deals and personalized recommendations.

### D. Scenarios 4: Dashboard of data

**As teammates** of the Crowdrebate platform or **our cooperation partners**, in addition to the basic operations mentioned above, they can also carry out the following functions:

Without being interfered with the user privacy, our teammates can enter the third button Platform data (Figure 3(a)), where they are able to see a dashboard of data related to the whole platform, such as how many orders we have received, how much money we have helped users to save, the distribution of our users by regions.(Figure 6(a)).

As for our cooperation partners of e-commerce platforms, they can see a dashboard of data related to their platforms. For example, as Taobao cooperators, they can see how many of the requests initiated by our platform users are from Taobao, the distribution of categories they like to buy, the ratio of sex, etc 6(b). This dashboard of data will help our partners to gain a more practical understanding of user preferences, design coupons, and reach a win-win collaboration mechanism.




(a) Dashboard of crowdrebate data (b) Dashboard of cooperator data

Fig. 6: Dashboard of data.

## IV. Conclusion

We present the Crowdrebate platform, which collects requests from users, groups requests into a set of orders to get more rebates and delivers products to different receivers in an order. Crowdrebate is equipped with the novel techniques including the crowdrebate algorithm grouping orders for maximum rebates, web crawling, items recommendation, and data visualization. We also introduce the system architecture and demonstrate scenarios via several case analyses.

## References

[1] E. Swilley and R. E. Goldsmith, "Black friday and cyber monday: Understanding consumer intentions on two major shopping days," *Journal of Retailing and Consumer Services*, vol. 20, no. 1, pp. 43–50, 2013.
[2] "[online] double11." https://en.wikipedia.org/wiki/Double_Eleven.
[3] G. Huang, X. Cheng, J. Wang, Y. Wang, D. He, T. Zhang, F. Li, S. Wang, W. Cao, and Q. Li, "X-engine: An optimized storage engine for large-scale e-commerce transaction processing," in *Proceedings of the 2019 International Conference on Management of Data*, pp. 651–665, 2019.
[4] "[online] Groupon." https://www.groupon.com/.
[5] "[online] dealmoon." https://www.dealmoon.com/.
[6] "[online] Must the "discount question" of Double 11 be so difficult?." https://www.tellerreport.com/business/2020-11-04-must-the-%22discount-question%22-of-double-11-be-so-difficult-.rJUQy_ulFP.html.
[7] "[online] Technical Report ." cspcheng.github.io/pdf/Crowdrebate-ICDE.pdf.
[8] "[online] etao." https://www.etao.com.
[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.
[10] "[online] SKU." https://en.wikipedia.org/wiki/Stock_keeping_unit.