SERVER AND METHOD FOR FACILITATING BATCHING ORDERS FOR ON-DEMAND SERVICE

TECHNICAL FIELD

[0001] Various embodiments relate to a server and a method for facilitating batching orders for an on-demand service.

BACKGROUND

[0002] Due to development of information technology, a consumer (who in some contexts herein may also be referred to as a "requester", a "customer", an "eater", a "passenger" or a "Pax") may request an on-demand service using a computing device associated with the consumer. The on-demand service may allow the consumer to fulfil the consumer's demand via an immediate access to goods and/or services. The consumer may request the on-demand service, for example, an item delivery service or a transport service (also referred to as an "e-hailing service" or a "car-hailing service"), using a user interface provided by an on-demand service platform and presented on the computing device associated with the consumer. To request the on-demand service, the consumer may make an order (also referred to as a "booking") for the on-demand service.

[0003] A server providing the on-demand service platform provided by an on-demand service platform provider may receive the order and allocate the order to a delivery service provider (who in some contexts herein may also be referred to as a "driver", a "driver partner", a "delivery partner", a "delivery agent" or a "Dax"). For example, if the item delivery service is requested, the server may request an item provider (who in some contexts herein may also be

referred to as a "merchant", a "food provider", a "restaurant" or a "Mex") to prepare an item for the order so that the delivery service provider may pick up the item and deliver the item to the consumer.

[0004] To improve efficiency in providing the on-demand service, the on-demand service platform provider has introduced a batching system. The batching system may allow the on-demand service platform provider to batch multiple orders (also referred to as "bookings") if at least one criterion is met, and the delivery service provider to take the batched multiple orders at a time. For example, if the multiple orders are received in a same time slot and locations relating to the multiple orders are close to each other, such multiple orders may be batched. The batching may help the on-demand service provider improve trip-per-transit-hour (TPTH) and reduce cost-to-serve (CTS), and may also help the delivery service provider earn more.

[0005] Conventionally, the batching system may use a delivery upper bound to batch the multiple orders. The delivery upper bound may refer to the latest time that the on-demand service platform provider needs to deliver the item to the consumer by the delivery service provider. The delivery upper bound may serve as a critical constraint in a batch formation, balancing fulfilment efficiency with consumer's satisfaction. While a higher upper delivery bound may enhance the fulfilment efficiency, the higher upper delivery bound may also lead to protracted actual time of arrival (ATA), and thus diminishing the consumer's satisfaction.

[0006] Conventionally, the following batching strategies may be employed to the batching system to determine the delivery upper bound:

- Dynamic Batching Delay (DBD) strategy: The DBD strategy may involve calculating a delay that is added to a direct travel time to accommodate extra time needed for detours or when delivering an order as part of a batch. The DBD strategy may be

variable, and contingent upon the number of batching attempts and current market conditions.

- Dynamic Batching Policy (DBP) strategy: The DBP strategy may serve as an evolution of the DBD strategy, setting a fixed delivery Service Level Agreement (SLA) for all (local) orders that may be aligned with prevailing market conditions.

[0007] The DBD strategy may have a high level of dynamism, but this may potentially cause a post-order delay, which may lack a guardrail for the consumer's experience and lead to ambiguity in business operations. Moreover, during periods of low allocation rates, the DBD strategy may inadvertently increase delays, inadvertently prioritising fulfilment rates over the fulfilment efficiency. Meanwhile, the DBP strategy may be clearer and provide the guardrail for the consumer's experience. The simple rule-based delivery SLA for all orders may be easy to control and have good business interpretability. However, the DBP strategy may be less efficient, because, generally, the DBP strategy may take all orders equally (for example, by using a similar delivery SLA) in a coarse granularity without sufficient consideration of specificity of each order.

[0008] The static batching strategies, including the DBD strategy and the DBP strategy, may hardly leverage information from a formed cluster, and may fail to look into reciprocity between the multiple orders, which may be critical in batching optimisation where trip efficiency is set as an objective.

[0009] Therefore, there is a need to provide a solution for facilitating batching orders for on-demand services. Specifically, there is a need to set the delivery upper bound more dynamically, by looking into a real-time demand condition by investigating a batch lovability with other orders in the same cluster, while respecting the overall estimated time of arrival (ETA) promise kept as the guardrail for the consumer's experience.

SUMMARY

[0010] According to various embodiments, there is a server for facilitating batching orders of an on-demand service, the server comprising: a memory for storing instructions; a communication interface configured to receive a plurality of orders for the on-demand service from a plurality of computing devices; and a processor for executing the stored instructions and configured to: classify the plurality of orders into one or more clusters, based on a location relating to the plurality of orders; for each cluster including two or more orders, obtain a benchmark delivery upper bound for the cluster, generate a benchmark cluster corresponding to the benchmark delivery upper bound, generate one or more cluster variants, and generate one or more delivery upper bounds each corresponding to the one or more cluster variants by adjusting the benchmark delivery upper bound; input the two or more orders of the cluster with the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants into a vehicle routing solution engine; obtain a benchmark set of candidate batches of the two or more orders for the benchmark cluster and one or more variant sets of candidate batches of the two or more orders for each of the one or more cluster variants, from the vehicle routing solution engine; compute a relative efficiency value for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches; and select one of the benchmark set of candidate batches and the one or more variant sets of candidate batches based on the relative efficiency value, as a set of finalised batches of the two or more orders for the cluster.

[0011] In some embodiments, the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute an overall time efficiency (OTE) and a sum of actual time of arrival (ATA) of the

two or more orders; and compute the relative efficiency value based on the OTE and the sum of the ATA.

[0012] In some embodiments, the processor is further configured to determine the OTE for the benchmark set of candidate batches as a benchmark OTE and the sum of the ATA for the benchmark set of candidate batches as a benchmark sum of the ATA.

[0013] In some embodiments, the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute the OTE by dividing a sum of travel time of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into a sum of a direct travel time of the two or more orders of the cluster.

[0014] In some embodiments, the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute an OTE improvement value by dividing the OTE into the benchmark OTE.

[0015] In some embodiments, the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute an ATA increasement value by dividing the sum of the ATA of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into the benchmark sum of the ATA.

[0016] In some embodiments, the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute the relative efficiency value by dividing the OTE improvement value into the ATA increasement value.

[0017] In some embodiments, the processor is further configured to select the one of the benchmark set of candidate batches and the one or more variant sets of candidate batches

with a highest relative efficiency value, as the set of finalised batches of the two or more orders for the cluster.

[0018] In some embodiments, the processor is further configured to adjust the benchmark delivery upper bound based on one or more configurations which depend on at least one of a region, a time and an order type.

[0019] In some embodiments, the processor is further configured to select one of the benchmark delivery upper bound and the one or more delivery upper bounds corresponding to the set of finalised batches, as a finalised delivery upper bound for the cluster.

[0020] According to various embodiments, there is a method for facilitating batching orders of an on-demand service, the method comprising: receiving a plurality of orders for the on-demand service from a plurality of computing devices; classifying the plurality of orders into one or more clusters, based on a location relating to the plurality of orders; for each cluster including two or more orders, obtaining a benchmark delivery upper bound for the cluster, generating a benchmark cluster corresponding to the benchmark delivery upper bound, generating one or more cluster variants, and generating one or more delivery upper bounds each corresponding to the one or more cluster variants by adjusting the benchmark delivery upper bound; inputting the two or more orders of the cluster with the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants into a vehicle routing solution engine; obtaining a benchmark set of candidate batches of the two or more orders for the benchmark cluster and one or more variant sets of candidate batches of the two or more orders for each of the one or more cluster variants, from the vehicle routing solution engine; computing a relative efficiency value for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches; and selecting one of the benchmark set of candidate batches and the one or more

variant sets of candidate batches based on the relative efficiency value, as a set of finalised batches of the two or more orders for the cluster.

[0021] In some embodiments, the method further comprises: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing an overall time efficiency (OTE) and a sum of actual time of arrival (ATA) of the two or more orders; and computing the relative efficiency value based on the OTE and the sum of the ATA.

[0022] In some embodiments, the method further comprises: determining the OTE for the benchmark set of candidate batches as a benchmark OTE and the sum of the ATA for the benchmark set of candidate batches as a benchmark sum of the ATA.

[0023] In some embodiments, the method further comprises: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing the OTE by dividing a sum of travel time of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into a sum of a direct travel time of the two or more orders of the cluster.

[0024] In some embodiments, the method further comprises: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing an OTE improvement value by dividing the OTE into the benchmark OTE.

[0025] In some embodiments, the method further comprises: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing an ATA increasement value by dividing the sum of the ATA of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into the benchmark sum of the ATA.

[0026] In some embodiments, the method further comprises: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing the

relative efficiency value by dividing the OTE improvement value into the ATA increasement value.

[0027] In some embodiments, the method further comprises: selecting the one of the benchmark set of candidate batches and the one or more variant sets of candidate batches with a highest relative efficiency value, as the set of finalised batches of the two or more orders for the cluster.

[0028] In some embodiments, the method further comprises: adjusting the benchmark delivery upper bound based on one or more configurations which depend on at least one of a region, a time and an order type.

[0029] In some embodiments, the method further comprises: selecting one of the benchmark delivery upper bound and the one or more delivery upper bounds corresponding to the set of finalised batches, as a finalised delivery upper bound for the cluster.

[0030] According to various embodiments, a data processing apparatus configured to perform the method of any one of the above embodiments is provided.

[0031] According to various embodiments, a computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments is provided.

[0032] According to various embodiments, a computer-readable medium comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments is provided. The computer-readable medium may include a non-transitory computer-readable medium.

BRIEF DESCRIPTION OF THE DRAWINGS

[0033] The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

- FIGS. 1 and 2 illustrate infrastructures of a system including a server for facilitating batching orders for an on-demand service according to various embodiments.
- FIG. 3 illustrates a block diagram of a server for facilitating batching orders for an on-demand service according to various embodiments.
- FIG. 4 illustrates an exemplary view of one or more clusters according to various embodiments.
- FIG. 5 illustrates a flowchart for a method for facilitating batching orders for an on-demand service according to various embodiments.
- FIG. 6 illustrates a data flow diagram of a server for facilitating batching orders for an on-demand service according to various embodiments.

DETAILED DESCRIPTION

[0034] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilised and structural, and logical changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0035] Embodiments described in the context of one of a server and a method are analogously valid for the other server and method. Similarly, embodiments described in the context of a server are analogously valid for a method, and vice-versa.

[0036] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0037] In the context of various embodiments, the articles "a", "an" and "the" as used with regard to a feature or element include a reference to one or more of the features or elements.

[0038] As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

[0039] Throughout the description, the term "module" may be understood as an application specific integrated circuit (ASIC), an electronic circuit, a combinational logic circuit, a field programmable gate array (FPGA), a processor which executes code, other suitable hardware components which provide the described functionality, or any combination thereof. The term of "module" may include a memory which stores code executed by the processor.

[0040] In the following, embodiments will be described in detail.

[0041] FIGS. 1 and 2 illustrate infrastructures of a system 200 including a server 100 for facilitating batching orders for an on-demand service according to various embodiments.

[0042] As shown in FIG. 1, the system 200 may include, but is not limited to, the server 100, a database system 140, a network 150, a plurality of first computing devices 160 each associated with a plurality of consumers 161 (who in some contexts herein may also be

referred to as a "requester", a "customer", an "eater", a "passenger" or a "Pax"), a plurality of second computing devices 170 (not shown) each associated with a plurality of delivery service providers 171 (who in some contexts herein may also be referred to as a "driver", a "driver partner", a "delivery partner", a "delivery agent" or a "Dax"), and a plurality of third computing devices 180 (not shown) each associated with a plurality of item providers 181 (who in some contexts herein may also be referred to as a "merchant", a "food provider", a "restaurant" or a "Mex"). In some embodiments, a consumer 161a may be the eater or the passenger for the on-demand service. In some other embodiments, the consumer 161a may be different from the eater or the passenger, and may use the on-demand service for and/or on behalf of the eater or the passenger who is a third party.

[0043] In some embodiments, the on-demand service may be a service allowing the consumer 161a to fulfil the consumer's 161a demand via an immediate access to items and/or services. The consumer 161a may request the on-demand service, such as a transport service (also referred to as an "e-hailing service" or a "car-hailing service") or an item delivery service, using a user interface screen presented on a first computing device 160a. The consumer 161a may make an order for the on-demand service using the first computing device 160a.

[0044] In some embodiments, the consumer 161a may use an application, for example, a mobile application, provided by the server 100. For example, the server 100 may provide an on-demand service platform, and may be controlled and/or managed by an on-demand service platform provider. The application may be installed in the first computing device 160a associated with the consumer 161a, to interact with the server 100 for the on-demand service.

[0045] In some embodiments, the network 150 may include, but is not limited to, a Local Area Network (LAN), a Wide Area Network (WAN), a Global Area Network (GAN), or any

combination thereof. The network 150 may provide a wireline communication, a wireless communication, or a combination of the wireline and wireless communication between the server 100 and the plurality of first computing devices 160, between the server 100 and the plurality of second computing devices 170, and between the server 100 and the plurality of third computing devices 180. As shown in FIG. 2, the network 150 may provide the wireline communication, the wireless communication, or the combination of the wireline and wireless communication between the first computing device 160a of the plurality of first computing devices 160 and a second computing device 170a of the plurality of second computing devices 170.

[0046] In some embodiments, the plurality of first computing devices 160 may be connectable to the server 100 via the network 150. In some embodiments, the plurality of first computing devices 160 may be arranged in data or signal communication with the server 100 via the network 150. In some embodiments, the plurality of first computing devices 160 may include, but is not limited to, at least one of the following: a mobile phone, a tablet computer, a laptop computer, a desktop computer, a head-mounted display and a smart watch. In some embodiments, the plurality of first computing devices 160 may be associated with the plurality of consumers 161 respectively. For example, the plurality of first computing devices 160 may belong to the plurality of consumers 161 respectively. For example, the first computing device 160a may belong to the consumer 161a who is the eater or the passenger. As another example, the first computing device 160a may belong to the consumer 161a requesting the on-demand service for the eater or the passenger who is a recipient of the on-demand service.

[0047] In some embodiments, the first computing device 160a may include a location sensor. In some embodiments, the location sensor may communicate with at least one of a global positioning satellite (GPS) server, a network server, and a Wi-Fi server, to detect a

location of the first computing device 160a. In some embodiments, the first computing device 160a may generate information about the location of the first computing device 160a.

[0048] In some embodiments, the server 100, for example, implemented by a server computer, may include a communication interface 110, a processor 120, and a memory 130 (as will be described with reference to FIG. 3).

[0049] In some embodiments, the server 100 may communicate with the plurality of first computing devices 160 via the network 150. In some embodiments, the consumer 161a may request the on-demand service, for example, the delivery service or the transport service, using the user interface screen presented on the first computing device 160a. In some embodiments, to request the on-demand service, the consumer 161a may make an order (also referred to as a "booking") for the on-demand service. In some embodiments, the first computing device 160a may receive the order from the consumer 161a for the on-demand service. The first computing device 160a may send the order to the server 100 via the network 150. In some embodiments, the order may relate to information about a pick-up location and a drop-off location for the on-demand service. In some embodiments, the order may include the information about the pick-up location and the drop-off location for the on-demand service. In some embodiments, the information may include the location of the first computing device 160a. In some embodiments, the location of the first computing device 160a may be considered as a location of the consumer 161a. In some embodiments, the location of the consumer 161a may be considered as a drop-off location (that in some contexts herein may also be referred to as a "destination" or a "delivery location") of the on-demand service (for example, the item delivery service). In some other embodiments, the location of the consumer 161a may be considered as a pick-up location (that in some contexts herein may also be referred to as a "starting point") of the on-demand service (for example, the transport service). In some other embodiments, the first computing device 160a may send

information about an address of the consumer 161a, and the address of the consumer 161a may be considered as the pick-up location or the drop-off location of the on-demand service. In some embodiments, the information may include a destination that the consumer 161a would like to go as the drop-off location. In some embodiments, the information may include information about an item provider 181a that the consumer 161a selected, and include a location of the item provider 181a as the pick-up location (as will be described below).

[0050] In some embodiments, the system 200 may further include a database 141. In some embodiments, the database 141 may be a part of the database system 140 which may be external to the server 100. The server 100 may communicate with the database 141. In some other embodiments, although not shown, the database 141 may be implemented locally in the memory 130 of the server 100.

[0051] In some embodiments, the server 100 may communicate with the plurality of second computing devices 170 via the network 150. In some embodiments, the plurality of second computing devices 170 may be arranged in data or signal communication with the server 100 via the network 150. In some embodiments, the plurality of second computing devices 170 may include, but is not limited to, at least one of the following: a mobile phone, a tablet computer, a laptop computer, a desktop computer, a head-mounted display and a smart watch. In some embodiments, the plurality of second computing devices 170 may be associated with the plurality of delivery service providers 171 respectively. For example, the plurality of second computing devices 170 may belong to the plurality of delivery service providers 171 respectively.

[0052] In some embodiments, the server 100 may receive the order from the first computing device 160a. After the server 100 receives the order from the first computing device 160a, the server 100 may allocate (assign) the order to a suitable delivery service provider 171a. In some embodiments, the second computing device 170a associated with the delivery service

provider 171a may send information about a location of the second computing device 170a to the server 100 via the network 150. The location of the second computing device 170a may be considered as a location of the delivery service provider 171a. In some embodiments, the location of the delivery service provider 171a may be considered as a current location of the delivery service provider 171a, and may change while the delivery service provider 171a moves. In some embodiments, the server 100 may provide the second computing device 170a with a map relating to a travel route from the current location of the second computing device 170a (which may be considered as the location of the delivery service provider 171a) to a next location (for example, the pick-up location or the drop-off location) for providing the on-demand service.

[0053] In some embodiments, the server 100 may communicate with the plurality of third computing devices 180 via the network 150. In some embodiments, the plurality of third computing devices 180 may be arranged in data or signal communication with the server 100 via the network 150. In some embodiments, the plurality of third computing devices 180 may include, but is not limited to, at least one of the following: a mobile phone, a tablet computer, a laptop computer, a desktop computer, a head-mounted display and a smart watch. In some embodiments, the plurality of third computing devices 180 may be associated with the plurality of item providers 181 respectively. For example, the plurality of third computing devices 180 may belong to the plurality of item providers 181 respectively.

[0054] In some embodiments, the plurality of item providers 181 may include, but are not limited to, a food provider and a goods provider, that may manufacture and/or provide items, for example, foods or goods. For example, the food provider may include, but is not limited to, a restaurant and a café As an example, the goods provider may include, but is not limited to, a store, a market, and a supermarket. In some embodiments, the server 100 may receive the order for the item delivery service with the information about the location of the

consumer 161a, and then produce a list of items, associated with at least one item provider, that may be prepared and delivered to the consumer 161a. In some embodiments, the server 100 may communicate with the plurality of third computing devices 180 to check the plurality of item providers' 181 availability. In some embodiments, the server 100 may communicate with the plurality of third computing devices 180 to aggregate information including, but not limited to, a list of available items, an estimated time of preparation of each item, and an estimated price of each item, in order to produce the list of items. In some embodiments, the first computing device 160a may display the list of items with the aggregated information of the at least one of the plurality of item providers 181 on the user interface screen. In some embodiments, after the consumer 161a makes selections on the user interface screen for the order for the item delivery service, for example, by selecting an item provider 181a and the item, the server 100 may communicate with a third computing device 180a associated with the selected item provider 181a to prepare the selected item. In some embodiments, the location of the first computing device 160a may be considered as a location of the consumer 161a. In some embodiments, the location of the consumer 161a may be considered as the drop-off location (that in some contexts herein may also be referred to as the "destination" or the "delivery location") of the item delivery service. In some other embodiments, the first computing device 160a may send information about an address of the consumer 161a, and the address of the consumer 161a may be considered as the drop-off location of the item delivery service.

[0055] In some embodiments, the delivery service provider 171a may take two or more orders which meet at least one criterion (also referred to as "batched orders"). In some embodiments, the server 100 may receive a plurality of orders from the plurality of computing devices 160. The server 100 may batch the two or more orders which meet at least one criterion. For example, if the two or more orders are received in a same time slot (for

example, a predetermined time slot (e.g. 5 minutes)) and locations (for example, pick-up locations and/or drop-off locations) of the two or more orders are close to each other (for example, within a predetermined distance), the server 100 may batch the two or more orders, and assign the batched orders to one delivery service provider 171a. The delivery service provider 171a may take the batched orders at a time.

[0056] In some embodiments, for the item delivery service, the delivery service provider 171a may pick up a first item from a first pick-up location, for example, a location of a first item provider 181a, and pick up a second item from a second pick-up location, for example, a location of a second item provider 181b, before delivering the first item to a first drop-off location, for example, a location of a first consumer 161a. The delivery service provider 171a may then deliver the first item to the first drop-off location, for example, the location of the first consumer 161a, and deliver the second item to a second drop-off location, for example, a location of a second consumer 161b. In some embodiments, for the transport service, the delivery service provider 171a may pick up a first consumer 161a at a first pick-up location, for example, a location of the first consumer 161a, and pick up a second consumer 161b at a second pick-up location, for example, a location of the second consumer 161b, before dropping off the first consumer 161a at a first drop-off location, for example, a destination of the first consumer 161a. The delivery service provider 171a may then drop off the first consumer 161a at the first drop-off location, for example, the destination of the first consumer 161a, and drop off the second consumer 161b at a second drop-off location, for example, a destination of the second consumer 161b.

[0057] FIG. 3 illustrates a block diagram of a server 100 for facilitating batching orders for an on-demand service according to various embodiments. FIG. 4 illustrates an exemplary view of one or more clusters 400 according to various embodiments.

[0058] As shown in FIG. 3, the server 100, for example, implemented by a server computer, may include a communication interface 110, a processor 120, and a memory 130.

[0059] In some embodiments, the memory 130 (also referred to as a "database") may store input data and/or output data temporarily or permanently. In some embodiments, the memory 130 may be configured to store instructions. In some embodiments, the memory 130 may store program code which allows the server 100 to perform a method 300 (as will be described with reference to FIG. 5). In some embodiments, the program code may be embedded in a Software Development Kit (SDK). The memory 130 may include an internal memory of the server 100 and/or an external memory. The external memory may include, but is not limited to, an external storage medium, for example, a memory card, a flash drive, and a web storage.

[0060] In some embodiments, the communication interface 110 may allow a plurality of first computing devices 160 to communicate with the processor 120 of the server 100 via the network 150, as shown in FIGS. 1 and 2. As shown in FIGS. 1 and 2, each of the plurality of first computing devices 160 may belong to each of consumers 161 who want to make an order for the on-demand service. In some embodiments, the communication interface 110 may transmit signals to the plurality of first computing devices 160, and/or receive signals from the plurality of first computing devices 160, via the network 150. For example, as shown in FIGS. 1 and 2, a first computing device 160a may belong to a consumer 161a who wants to make an order for the on-demand service, and the communication interface 110 may transmit signals to the first computing device 160a, and/or receive signals from the first computing device 160a via the network 150.

[0061] In some embodiments, the communication interface 110 may allow a plurality of second computing devices 170 to communicate with the processor 120 of the server 100 via the network 150, as shown in FIGS. 1 and 2. As shown in FIGS. 1 and 2, each of the plurality

of second computing devices 170 may belong to each of a plurality of delivery service providers 171 who may pick up an item from an item provider 181a at a pick-up location and deliver the item to the consumer 161a at a drop-off location and/or who may transport the consumer 161a from a pick-up location to a drop-off location. In some embodiments, the communication interface 110 may transmit signals to the plurality of second computing devices 170, and/or receive signals from the plurality of second computing devices 170, via the network 150.

[0062] In some embodiments, the communication interface 110 may allow a plurality of third computing devices 180 to communicate with the processor 120 of the server 100 via the network 150, as shown in FIG. 1. As shown in FIG. 1, each of the plurality of third computing devices 180 may belong to each of a plurality of item providers 181 who may prepare an item, for example, food, for the order. In some embodiments, the communication interface 110 may transmit signals to the plurality of third computing devices 180, and/or receive signals from the plurality of third computing devices 180, via the network 150.

[0063] The processor 120 may include, but is not limited to, a microprocessor, an analogue circuit, a digital circuit, a mixed-signal circuit, a logic circuit, an integrated circuit, a Central Processing Unit (CPU), a Graphics Processing Unit (GPU), a Digital Signal Processor (DSP), a Field Programmable Gate Array (FPGA), an Application Specific Integrated Circuit (ASIC), or any combination thereof. Any other kind of implementation of the respective functions, which will be described below in further detail, may also be understood as the processor 120.

[0064] In some embodiments, the processor 120 may be connectable to the communication interface 110. In some embodiments, the processor 120 may be arranged in data or signal communication with the communication interface 110 to transmit/receive the signals.

[0065] In some embodiments, the communication interface 110 may receive a plurality of orders (also referred to as "a plurality of bookings") for the on-demand service from the

plurality of first computing devices 160 each associated with the plurality of consumers 161. In some embodiments, the processor 120 may receive the plurality of orders for the on-demand service from the communication interface 110. For example, the communication interface 110 may receive an order (also referred to as a "booking") for the on-demand service from the first computing device 160a associated with the consumer 161a, and the processor 120 may receive the order for the on-demand service from the communication interface 110.

[0066] In some embodiments, the processor 120 may receive a request for a search for the item delivery service from the first computing device 160a. In some embodiments, the processor 120 may produce a list of items, associated with at least one item provider 181, that may be prepared and delivered to the consumer 161a. The processor 120 may then provide the list of items to the first computing device 160a, so that the consumer 161a may select an item provider 181a and an item for the item delivery service.

[0067] In some embodiments, the consumer 161a may select the item provider 181a and the item from the list of items. The first computing device 160a may generate information about the selected item provider 181a and the selected item, based on the consumer's 161a input. The processor 120 may receive the information about the selected item provider 181a and the selected item from the first computing device 160a, via the communication interface 110. In some embodiments, the processor 120 may then provide the information about the selected item to the selected item provider 181a for preparing the selected item, via the communication interface 110.

[0068] In some embodiments, the processor 120 may select a delivery service provider 171a from one or more delivery service providers 171 based on a geographical location of the one or more delivery service providers 171. In some embodiments, the processor 120 may request the selected delivery service provider 171a to pick up the selected item at a geographical

location of the selected item provider 181a (i.e. the pick-up location) and deliver the selected item to the consumer 161a (i.e. the drop off location).

[0069] In some embodiments, the processor 120 may receive a request for the transport service from the first computing device 160a. The first computing device 160a may generate information about a location of the first computing device 160a which may be considered as a location of the consumer 161a (i.e. the pick-up location) and a location of a destination (i.e. the drop-off location), for example, based on the consumer's 161a input. The processor 120 may receive the information about the pick-up location and the drop-off location from the first computing device 160a, via the communication interface 110. In some embodiments, the processor 120 may select a delivery service provider 171a from one or more delivery service providers 171 based on a geographical location of the one or more delivery service providers 171. In some embodiments, the processor 120 may request the selected delivery service provider 171a to pick up the consumer 161a at the pick-up location and drop off the consumer 161a at the drop-off location.

[0070] In some embodiments, the processor 120 may receive a plurality of orders from a plurality of first computing devices 160 each associated with a plurality of consumers 161, and the processor 120 may batch two or more orders which meet at least one criterion among the plurality of orders. For example, if the two or more orders are received in a same time slot (for example, a predetermined time slot (e.g. 5 minutes)) and locations (for example, pick-up locations and/or drop-off locations) of the two or more orders are close to each other (for example, within a predetermined distance), the processor 120 may batch the two or more orders, and assign the batched orders to the selected delivery service provider 171a. In some embodiments, for the item delivery service, the locations of the two or more orders may include the pick-up locations of the two or more orders (for example, locations of item providers 181 of the two or more orders) and/or the drop-off locations of the two or more

orders (for example, locations of consumers 161 of the two or more orders). In some embodiments, for the transport service, the locations of the two or more orders may include the pick-up locations of the two or more orders (for example, locations of consumers 161 of the two or more orders) and/or the drop-off locations of the two or more orders (for example, destinations of the two or more orders).

[0071] In some embodiments, the processor 120 may determine a travel route of the selected delivery service provider 171a. For example, the processor 120 may determine the travel route of the selected delivery service provider 171a based on the pick-up locations and the drop-off locations of the batched orders. In some embodiments, the processor 120 may provide the travel route of the selected delivery service provider 171a to a second computing device 170a associated with the selected delivery service provider 171a. In some embodiments, the processor 120 may provide the travel route of the selected delivery service provider 171a to first computing devices 160, for example, first computing devices 160a, 160b, each associated with the consumers 161, for example, a first consumer 161a and a second consumer 161b, who made each of the batched orders, for example, a first order and a second order.

[0072] In some embodiments, the selected delivery service provider 171a may take the batched orders, for example, the first order and the second order, at a time. In some embodiments, for the item delivery service, the delivery service provider 171a may pick up a first item from a first pick-up location, for example, a location of a first item provider 181a, and pick up a second item from a second pick-up location, for example, a location of a second item provider 181b, before delivering the first item to a first drop-off location, for example, a location of the first consumer 161a. The delivery service provider 171a may then deliver the first item to the first drop-off location, for example, the location of the first consumer 161a, and deliver the second item to a second drop-off location, for example, a

location of the second consumer 161b. In some embodiments, for the transport service, the delivery service provider 171a may pick up the first consumer 161a at a first pick-up location, for example, a location of the first consumer 161a, and pick up the second consumer 161b at a second pick-up location, for example, a location of the second consumer 161b, before dropping off the first consumer 161a at a first drop-off location, for example, a destination of the first consumer 161a. The delivery service provider 171a may then drop off the first consumer 161a at the first drop-off location, for example, the destination of the first consumer 161a, and drop off the second consumer 161b at a second drop-off location, for example, a destination of the second consumer 161b.

[0073] In some embodiments, in order to facilitate batching the orders for the on-demand service, the processor 120 may classify a plurality of orders into one or more clusters, after receiving the plurality of orders. In some embodiments, a candidate pool (as shown in FIG. 6) may receive the plurality of orders, and the processor 120 may classify the plurality of orders into the one or more clusters. In some embodiments, the processor 120 may classify the plurality of orders into one or more clusters based on a location relating to the plurality of orders. In some embodiments, the location relating to the plurality of orders may include at least one of a pick-up location and a drop-off location for each order of the plurality of orders.

[0074] An exemplary diagram of the one or more clusters 400 is shown in FIG. 4. As shown in FIG. 4, if the processor 120 receives 6 orders B1, B2, B3, B4, B5 and B6 (for example, a first order B1, a second order B2, a third order B3, a fourth order B4, a fifth order B5, and a sixth order B6), the processor 120 may check a location relating to the 6 orders B1, B2, B3, B4, B5 and B6, and classify the 6 orders B1, B2, B3, B4, B5 and B6 into two clusters 410, 420 based on the location relating to the 6 orders B1, B2, B3, B4, B5 and B6. In some embodiments, the processor 120 may determine a batching efficiency of the plurality of orders (for example, the 6 orders B1, B2, B3, B4, B5 and B6) based on at least one of actual

traffic data and actual road data (for example, one-way street or two-way street) in relation to the location relating to the plurality of orders (for example, the 6 orders B1, B2, B3, B4, B5 and B6), and then determine the number of the one or more clusters and how to classify the plurality of orders (for example, the 6 orders B1, B2, B3, B4, B5 and B6) into the one or more clusters based on the batching efficiency. For example, if the batching efficiency of a combination of the plurality of orders (for example, the first order B1, the second order B2, and the third order B3) is equal to or greater than a predetermined efficiency, the processor 120 may determine to classify the combination of the plurality of orders (for example, the first order B1, the second order B2, and the third order B3) into one cluster (for example, the first cluster 410). In addition, for example, if the batching efficiency of another combination of the plurality of orders (for example, the fourth order B4, the fifth order B5, and the sixth order B6) is equal to or greater than the predetermined efficiency, the processor 120 may determine to classify the another combination of the plurality of orders (for example, the fourth order B4, the fifth order B5, and the sixth order B6) into one cluster (for example, the second cluster 420). In addition, for example, if the batching efficiency of another combination of the plurality of orders (for example, the first order B1, the fifth order B5, and the sixth order B6) is less than the predetermined efficiency, the processor 120 may determine not to classify the another combination of the plurality of orders (for example, the first order B1, the fifth order B5, and the sixth order B6) into one cluster. As shown in FIG. 4, for example, the first order B1, the second order B2, and the third order B3 may belong to the first cluster 410, and the fourth order B4, the fifth order B5, and the sixth order B6 may belong to the second cluster 420.

[0075] Although not shown in FIG. 4, in some embodiments, each cluster of the one or more clusters may include one or more orders of the plurality of orders. For example, if the batching efficiency of a combination of any one of the 6 orders B1, B2, B3, B4, B5 and B6

with a seventh order B7 (not shown) is less than the predetermined efficiency, the processor 120 may determine not to classify the any combination including the seventh order B7 into one cluster, and may generate a third cluster 430 (not shown) including the seventh order B7 only. The processor 120 may then allocate the seventh order B7 to a suitable delivery service provider 171a.

[0076] Although not shown in FIG. 4, as another example, if the processor 120 receives 100 orders in a candidate pool in a city, the processor 120 may classify (divide) the 100 orders into 5 clusters (for example, a first cluster to a fifth cluster) based on pick-up locations and/or drop-off locations of each of the 100 orders. For example, the first cluster may include 10 orders, a second cluster may include 20 orders, etc.

[0077] In this manner, the processor 120 may split the plurality of orders from a city or a country into the one or more clusters, for example, multiple clusters, so that orders may be batched within the same cluster, as orders from different clusters may be too far apart to be a batch. Advantageously, by forming the one or more clusters, the processor 120 and/or a vehicle routing solution engine (for example, a Vehicle Routing Problem (VRP) solver)) (as will be described below) may explore batching possibility in a cluster scope, instead of a city scope or a country scope, and thus a workload for the batching possibility may be dramatically reduced.

[0078] In some embodiments, each of the one or more clusters may include a set of orders, for example, two or more orders of the plurality of orders. In some embodiments, after forming the one or more clusters each including the two or more orders, the processor 120 may obtain information about the two or more orders from each of the one or more cluster, including, but not limited to, pairwise pick-up and drop-off distance between orders, average travel time between pick-up and drop-off locations of the orders, and an order volume.

[0079] In some embodiments, for each cluster including the two or more orders, the processor 120 may obtain a benchmark delivery upper bound for the cluster. In some embodiments, the benchmark delivery upper bound for the cluster may be obtained from the memory 130 and/or the database 141. In some embodiments, the benchmark delivery upper bound for the cluster may be a parameter set by the on-demand service platform provider and/or the processor 120, which may be used to protect service quality. For example, the on-demand service platform provider and/or the processor 120 may reduce the benchmark delivery upper bound to obtain a solution that has better service quality, if the current solution is not efficient enough to trade against service quality.

[0080] In some embodiments, for each cluster including the two or more orders, the processor 120 may generate a benchmark cluster corresponding to the benchmark delivery upper bound. In some embodiments, the processor 120 may generate one or more cluster variants. In some embodiments, the benchmark delivery upper bound may have been obtained in relation to the cluster including the two or more orders, and the cluster including the two or more orders and corresponding to the benchmark delivery upper bound may be referred to as the "benchmark cluster". In some embodiments, one or more copies of the cluster including the two or more orders same as those of the cluster may be referred to as the "one or more cluster variants". In some embodiments, the processor 120 may determine the number of the one or more cluster variants based on one or more configurations which may depend on at least one of a region, a time and an order type (for example, a saver order or a priority order).

[0081] In some embodiments, for each cluster including the two or more orders, the processor 120 may generate one or more delivery upper bounds each corresponding to the one or more cluster variants, by adjusting the benchmark delivery upper bound. In some embodiments, the processor 120 may adjust the benchmark delivery upper bound based on

the one or more configurations which may depend on the at least one of the region, the time and the order type (for example, a saver order or a priority order). In some embodiments, an adjustment step size may be predefined from the one or more configurations when online serving (after receiving the plurality of orders). In some embodiments, the processor 120 may explore additional configurations and verify the additional configuration to be included in the one or more configurations, in offline simulations and experiments.

[0082] In some embodiments, a range for the adjustment of the benchmark delivery upper bound to generate the one or more delivery upper bounds each corresponding to the one or more cluster variants may be determined by the one or more configurations. In some embodiments, the one or more configurations may be flexible and configurable, and may vary in a different environment, for example, in a different region. For example, the processor 120 may adjust the benchmark delivery upper bound by subtracting 1 minute to generate a first delivery upper bound (i.e. the benchmark delivery upper bound – 1 minute) for a first cluster variant and adding 1 minute to generate a second delivery upper bound (i.e. the benchmark delivery upper bound + 1 minute) for a second cluster variant. In this manner, different cluster variants may have different delivery upper bounds.

[0083] In some embodiments, for each cluster including the two or more orders, the processor 120 may input the two or more orders of the cluster with the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants into a vehicle routing solution engine. In some embodiments, the vehicle routing solution engine may be a part of the processor 120. In some other embodiments, the vehicle routing solution engine may be included in the memory 130 and/or the database 141. In some other embodiments, the vehicle routing solution engine may be external to the server 100. As an example, the vehicle routing solution engine may be a

Vehicle Routing Problem (VRP) solver, for example, a conventional VRP solver which may be a kind of research problem abstract.

[0084] It may be appreciated that, conventionally, under an existing Dynamic Batching Policy (DBP) strategy, a batch of orders with a customised delivery upper bound has been presented to the VRP solver, and the VRP solver may then attempt to create efficient batches and return a set of finalised batches, where each batch may include multiple orders. In contrast, according to various embodiments, the processor 120 may submit multiple cluster variants and their relevant delivery upper bounds (for example, the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants), along with the two or more orders included in the cluster, to the VRP solver simultaneously.

[0085] In some embodiments, for each cluster including the two or more orders, the processor 120 may obtain a benchmark set of candidate batches of the two or more orders for the benchmark cluster and one or more variant sets of candidate batches of the two or more orders for each of the one or more cluster variants, from the vehicle routing solution engine.

[0086] In some embodiments, the processor 120 may receive an output of the VRP solver for an input of the benchmark cluster including the two or more orders and the benchmark delivery upper bound. In some embodiments, the VRP solver may try to form the benchmark set of candidate batches of the two or more orders (also referred to as a "trip" after converted). The trip may be a batch with multiple orders that can be dispatched to a single delivery service provider 171a. For example, the trip may be a single order, if it is impossible to form a batch with other orders in the cluster.

[0087] In some embodiments, the processor 120 may receive an output of the VRP solver for an input of the one or more cluster variants including the two or more orders and the one

or more delivery upper bounds. In some embodiments, the VRP solver may try to form the one or more variant sets of candidate batches of the two or more orders.

[0088] In some embodiments, for each cluster including the two or more orders, the processor 120 may compute a relative efficiency value for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches. In some embodiments, the relative efficiency value may be referred to as an "ROI (Return on Investment)".

[0089] In some embodiments, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, the processor 120 may compute an overall time efficiency (OTE) and a sum of actual time of arrival (ATA) of the two or more orders. In some embodiments, the processor 120 may compute the relative efficiency value based on the OTE and the sum of the ATA.

[0090] In some embodiments, the processor 120 may determine the OTE for the benchmark set of candidate batches as a benchmark OTE and the sum of the ATA for the benchmark set of candidate batches as a benchmark sum of the ATA.

[0091] In some embodiments, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, the processor 120 may compute the OTE based on a sum of travel time of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches and a sum of a direct travel time of the two or more orders of the cluster. In some embodiments, the processor 120 may compute the OTE by dividing the sum of travel time of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into the sum of a direct travel time of the two or more orders of the cluster. In some embodiments, the OTE may be obtained, according to the following mathematical equation (1):

$$OTE = \frac{\sum_{trip_k \in response_i} trip\ time_k}{\sum_{order_j \in cluster_i} direct\ trip\ time_j}$$

[0092] In some embodiments, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, the processor 120 may compute an OTE improvement value based on the OTE and the benchmark OTE. In some embodiments, the processor 120 may compute the OTE improvement value by dividing the OTE into the benchmark OTE. In some embodiments, the OTE improvement value may be obtained, according to the following mathematical equation (2):

$$OTE_{improvement} = \frac{OTE_i}{OTE_{benchmark}}$$

[0093] In some embodiments, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, the processor 120 may compute an ATA increasement value based on the sum of the ATA of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches and the benchmark sum of the ATA. In some embodiments, the processor 120 may compute the ATA increasement value by dividing the sum of the ATA of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into the benchmark sum of the ATA. In some embodiments, the ATA increasement value may be obtained, according to the following mathematical equation (3):

$$ATA_{increasement} = \frac{\sum_{order_j \in cluster_i} ATA_j}{\sum_{order_j \in cluster_{benchmark}} ATA_j}$$

[0094] In some embodiments, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, the processor 120 may compute the relative efficiency value based on the OTE improvement value and the ATA increasement value. In some embodiments, the processor 120 may compute the relative efficiency value by dividing the OTE improvement value into the ATA increasement value. In some embodiments, the

relative efficiency value may be obtained, according to the following mathematical equation (4):

$$relative\ efficiency\ ROI = \frac{OTE_{improvement}}{\mathrm{ATA}_{\mathrm{increasement}}}$$

[0095] In some embodiments, for each cluster including the two or more orders, the processor 120 may select one of the benchmark set of candidate batches and the one or more variant sets of candidate batches based on the relative efficiency value, as a set of finalised batches of the two or more orders for the cluster. In some embodiments, the processor 120 may select the one of the benchmark set of candidate batches and the one or more variant sets of candidate batches with a highest relative efficiency value, as the set of finalised batches of the two or more orders for the cluster. For example, if the processor 120 determines that the relative efficiency value of the benchmark set of candidate batches is the highest, the processor 120 may select the benchmark set of candidate batches as the set of finalised batches of the two or more orders for the cluster. As another example, if the processor 120 determines that the relative efficiency value of a second variant set of candidate batches corresponding to the second cluster variant, the processor 120 may select the second variant set of candidate batches as the set of finalised batches of the two or more orders for the cluster.

[0096] In some embodiments, the processor 120 may select one of the benchmark delivery upper bound and the one or more delivery upper bounds corresponding to the set of finalised batches, as a finalised delivery upper bound for the cluster. In some embodiments, the processor 120 may use the finalised delivery upper bound to batch the two or more orders of the cluster. For example, if the processor 120 selects the benchmark set of candidate batches as the set of finalised batches of the two or more orders for the cluster (for example, since the relative efficiency value of the benchmark set of candidate batches is the highest), the

processor 120 may select the benchmark delivery upper bound as the finalised delivery upper bound for the cluster. As another example, if the processor 120 selects the second variant set of candidate batches corresponding to the second cluster variant as the set of finalised batches of the two or more orders for the cluster (for example, since the relative efficiency value of the second variant set of candidate batches is the highest), the processor 120 may select the second delivery upper bound as the finalised delivery upper bound for the cluster.

[0097] In some embodiments, the processor 120 may select one of the benchmark cluster and the one or more cluster variants based on the relative efficiency value, as a finalised cluster (serving the set of finalised batches and the finalised delivery upper bound) for the cluster.

[0098] In some embodiments, given the benchmark delivery upper bound as $T_0 \in R^n$ for a cluster with n orders, a calculation method of the processor 120 may aim to compose k delivery upper bounds from T_1 to T_k which may be near to T_0 and used as a replacement of T_0. Then, $(k+1)$ copies of the cluster with a difference in delivery upper bounds may be sent to the VRP solver to retrieve an optimal solution (i.e. a set of candidate batches). Given a delivery upper bound T, the VRP solver may aim to provide a solution $s = f(t)$ which may optimise the OTE. By looking into the VRP solver's response, the processor 120 may obtain $(k+1)$ solutions from s_0 to s_k where $s_i = f(t_i)$. In the end, the processor 120 may select a solution that may optimise the relative efficiency value which is defined in the mathematical equation (4), taking s_0 as the benchmark set of candidate batches.

[0099] In some embodiments, the overall process of the processor 120 may behave like a local search around T_0. With the adjusted (modified) benchmark delivery upper bounds (i.e. the one or more generated delivery upper bounds), the processor 120 may identify orders that may potentially yield more efficient routing with a relaxed delivery upper bound, as well as those without such high relative efficiency value (high ROI), where the processor 120 mat

retain or further reduce the delivery upper bound. This process of merging delivery upper bound adjustments with the relative efficiency value identification may allow for immediate updates to the delivery upper bounds based on the VRP solver's output. A winning copy (i.e. selected one of the benchmark cluster and each of the one or more cluster variants), for example, identified by its highest relative efficiency value, may have its delivery upper bound adjusted accordingly for allocating the delivery service provider 171a. The one or more (predetermined) configurations may ensure an optimal balance between minimising a computational load on the VRP solver and maximising overall efficiency.

[00100] In some embodiments, the above mentioned processes (for example, from generating the one or more cluster variants to computing the relative efficiency value for each of the benchmark cluster and the one or more cluster variants) may be performed in parallel, as computation of s_i may be independent of each other. In some other embodiments, the above mentioned processes (for example, from generating the one or more cluster variants to computing the relative efficiency value for each of the benchmark cluster and the one or more cluster variants) may be performed sequentially, where the processor 120 may select (pick) T_i by utilising information obtained from s_0 , ..., $s_{(i-1)}$.

[00101] As described above, conventional batching strategies for determining a delivery upper bound, for example, a Dynamic Batching Delay (DBD) strategy and a Dynamic Batching Policy (DBP) strategy may have two primary challenges of defining an objective measurement for efficiency Return on Investment (ROI), and identifying the orders with high efficiency ROI and making real-time adjustments. To address these challenges, as described above, the various embodiments may provide a relative calculation framework, which may eschew the need for calculating absolute values that may heavily rely on predefined thresholds and lack adaptability. Advantageously, the various embodiments may show

significant overall time efficiency (OTE) and trip-per-transit-hour (TPTH) increase by adopting the adaptative upper bound adjustment strategy.

[00102] FIG. 5 illustrates a flowchart for a method for facilitating batching orders for an on-demand service according to various embodiments. According to various embodiments, the method S300 for facilitating batching the orders for the on-demand service may be provided.

[00103] In some embodiments, the method S300 may include a step S301 of receiving a plurality of orders for the on-demand service from a plurality of computing devices.

[00104] In some embodiments, the method S300 may include a step S302 of classifying the plurality of orders into one or more clusters, based on a location relating to the plurality of orders.

[00105] In some embodiments, the method S300 may include a step S303 of, for each cluster including two or more orders, obtaining a benchmark delivery upper bound for the cluster, generating a benchmark cluster corresponding to the benchmark delivery upper bound, generating one or more cluster variants, and generating one or more delivery upper bounds each corresponding to the one or more cluster variants by adjusting the benchmark delivery upper bound.

[00106] In some embodiments, the method S300 may include a step S304 of inputting the two or more orders of the cluster with the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants into a vehicle routing solution engine.

[00107] In some embodiments, the method S300 may include a step S305 of obtaining a benchmark set of candidate batches of the two or more orders for the benchmark cluster and one or more variant sets of candidate batches of the two or more orders for each of the one or more cluster variants, from the vehicle routing solution engine.

[00108] In some embodiments, the method S300 may include a step S306 of computing a relative efficiency value for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches.

[00109] In some embodiments, the method S300 may include a step S307 of selecting one of the benchmark set of candidate batches and the one or more variant sets of candidate batches based on the relative efficiency value, as a set of finalised batches of the two or more orders for the cluster.

[00110] FIG. 6 illustrates a data flow diagram of a server for facilitating batching orders for an on-demand service according to various embodiments.

[00111] In some embodiments, a processor 120 of a server 100 as described with reference to FIG. 3 may receive a plurality of (new) orders. In some embodiments, a candidate pool may receive the plurality of orders, and the processor 120 may classify the plurality of orders into the one or more clusters. In some embodiments, the processor 120 may split the plurality of orders from a city or a country into one or more clusters, for example, multiple clusters, so that orders may be batched within the same cluster, as orders from different clusters may be too far apart to be a batch. In some embodiments, each of the one or more clusters may include two or more orders. In some other embodiments, each of the one or more clusters may include one or more orders. In some embodiments, a downstream Vehicle Routing Problem (VRP) solver may explore batching possibility in a cluster scope, instead of a city scope or a country scope, and thus a workload for the batching possibility may be dramatically reduced.

[00112] As shown in FIG. 6, in some embodiments, the one or more clusters may include a first cluster (cluster 1) and an N-th cluster (cluster N). For example, the first cluster (cluster 1) may refer to one cluster of trips from a clustering operation on the candidate pool. For example, although not shown in FIG. 6, if there are 100 orders in the candidate pool in the

city, the 100 orders may be divided into 5 clusters and the first cluster (cluster 1) may include 10 orders, and a second cluster (cluster 2) (not shown) may include 20 orders, etc.

[00113] As shown in FIG. 6, in some embodiments, the processor 120 may generate a copy of the cluster including the two or more orders same as those of the cluster, and refer one or more copies of the cluster as "one or more cluster variants". For example, a first cluster variant (cluster 1 variant 1) and an M-th cluster variant (cluster 1 variant M) for the first cluster (cluster 1) may be generated. In some embodiments, the processor 120 may then adjust a benchmark delivery upper bound for the first cluster (cluster 1) (also referred to as a "benchmark cluster") to generate one or more delivery upper bounds for each of the one or more cluster variants. Different cluster variants may have delivery upper bound adjustments (for example, benchmark delivery upper bound + 1 minute or – 1 minute).

[00114] As shown in FIG. 6, in some embodiments, the processor 120 may feed the two or more orders of the first cluster (cluster 1) with the benchmark delivery upper bound of the benchmark cluster (cluster 1) and the one or more delivery upper bounds of the first cluster variant (cluster 1 variant 1) and the M-th cluster variant (cluster 1 variant M) into the VRP solver.

[00115] As shown in FIG. 6, in some embodiments, the processor 120 may receive an output from the VRP solver. In some embodiments, for the first cluster (cluster 1), the processor 120 may receive a benchmark set of candidate batches of the two or more orders (cluster 1 trips) for the benchmark cluster (cluster 1) and one or more variant sets of candidate batches of the two or more orders (cluster 1 variant 1 trips and cluster 1 variant M trips) for each of the one or more cluster variants (cluster 1 variant 1 and cluster 1 variant M).

[00116] In some embodiments, the benchmark set of candidate batches (cluster 1 trips) may refer to the output of the VRP solver with an input of the two or more orders of the first cluster (cluster 1). The VRP solver may try to form batches on the two or more orders, and

the processor 120 may call results as a "trip" after converted. The trip may be a batch with multiple orders that can be dispatched to a single delivery service provider 171a. For example, the trip may be a single order, if it is impossible to form a batch with other orders in the first cluster (cluster 1).

[00117] In some embodiments, the one or more variant sets of candidate batches of the two or more orders (cluster 1 variant 1 trips and cluster 1 variant M trips) may refer to the output of the VRP solver, and may be similar to "cluster 1 trips". Differences between the "cluster 1 trips", "cluster 1 variant 1 trips" and "cluster 1 variant M trips" may be that, different inputs may be fed to the VRP solver, and thus different results may be obtained from the VRP solver.

[00118] In some embodiments, the processor 120 may select one of the benchmark set of candidate batches (cluster 1 trips) and the one or more variant sets of candidate batches (cluster 1 variant 1 trips and cluster 1 variant M trips) based on a relative efficiency value (Return on Investment (ROI)), as a set of finalised batches of the two or more orders (cluster 1 trips winner) for the first cluster (cluster 1). In some embodiments, the set of finalised batches of the two or more orders (cluster 1 trips winner) may refer to a winner from the benchmark set of candidate batches (cluster 1 trips) and the one or more variant sets of candidate batches (cluster 1 variant 1 trips and cluster 1 variant M trips). These (M+1) instances may obtain the relative efficiency value (number) respectively, and the processor 120 may select the largest number instance as the winner which may be the highest efficiency one. For example, it may be an actual inference to one of these (M+1) instances.

[00119] In some embodiments, the processor 120 may then check if batched orders, which are batched according to the set of finalised batches of the two or more orders (cluster 1 trips winner), meet a predetermined criterion (for example, if the batched orders exceed a predetermined pooling time). If the processor 120 determines that the batched orders meet the

predetermined criterion, the processor 120 may allocate the batched orders of the first cluster (cluster 1) to delivery service provider(s) 171. If the processor 120 determines that any one of the batched orders does not meet the predetermined criterion, the processor 120 may return orders which do not meet the predetermined criterion to the candidate pool, so that the orders may be classified again.

[00120] As described above, according to various embodiments, the processor 120 may introduce an adaptative delivery upper bound adjustment strategy that may adeptly balance the consumer's experience with fulfilment efficiency. The strategy may dynamically adjust the delivery upper bound in real-time, automatically optimising equilibrium between the consumer's satisfaction and the fulfilment efficiency. Simulations of this strategy may have demonstrated significant enhancements in both overall time efficiency (OTE) and trip-per-transit-hour (TPTH), while simultaneously maintaining or even reducing average actual time of arrival (ATA).

[00121] The various embodiments may use a concept of the relative efficiency value (Return on Investment (ROI)), which may be defined as efficiency gains relative to a unit increase (for example, a unit of time (for example, 1 minute, 10 seconds, 1 second, etc.)) in the ATA. The various embodiments may enable identification of orders with high relative efficiency value in real-time, allowing for an incremental increase in the delivery upper bound to bolster efficiency. Conversely, the various embodiments may also facilitate a reduction of the delivery upper bound for orders with low relative efficiency value, thereby sustaining or even reducing the overall ATA and increasing estimated time of arrival (ETA) promise kept.

[00122] Advantageously, unlike conventional batching strategies, for example, a Dynamic Batching Delay (DBD) strategy and a Dynamic Batching Policy (DBP) strategy, the various embodiments may provide a more nuanced and adaptative strategy that may better serve both the fulfilment efficiency and the consumer's expectations for timely delivery. By

differentiating the conventional batching strategies and making different adjustments, the various embodiments may obtain more efficiency gains and achieve a better fulfilment efficiency and consumer's satisfaction balance.

[00123] While the disclosure has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A server for facilitating batching orders of an on-demand service, the server comprising:

a memory for storing instructions;

a communication interface configured to receive a plurality of orders for the on-demand service from a plurality of computing devices; and

a processor for executing the stored instructions and configured to:

classify the plurality of orders into one or more clusters, based on a location relating to the plurality of orders;

for each cluster including two or more orders, obtain a benchmark delivery upper bound for the cluster, generate a benchmark cluster corresponding to the benchmark delivery upper bound, generate one or more cluster variants, and generate one or more delivery upper bounds each corresponding to the one or more cluster variants by adjusting the benchmark delivery upper bound;

input the two or more orders of the cluster with the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants into a vehicle routing solution engine;

obtain a benchmark set of candidate batches of the two or more orders for the benchmark cluster and one or more variant sets of candidate batches of the two or more orders for each of the one or more cluster variants, from the vehicle routing solution engine;

compute a relative efficiency value for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches; and

select one of the benchmark set of candidate batches and the one or more variant sets of candidate batches based on the relative efficiency value, as a set of finalised batches of the two or more orders for the cluster.

2. The server according to claim 1, wherein the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches,

compute an overall time efficiency (OTE) and a sum of actual time of arrival (ATA) of the two or more orders; and

compute the relative efficiency value based on the OTE and the sum of the ATA.

3. The server according to claim 2, wherein the processor is further configured to determine the OTE for the benchmark set of candidate batches as a benchmark OTE and the sum of the ATA for the benchmark set of candidate batches as a benchmark sum of the ATA.

4. The server according to claim 3, wherein the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute the OTE by dividing a sum of travel time of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into a sum of a direct travel time of the two or more orders of the cluster.

5. The server according to claim 4, wherein the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute an OTE improvement value by dividing the OTE into the benchmark OTE.

6. The server according to claim 5, wherein the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute an ATA increasement value by dividing the sum of the ATA of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into the benchmark sum of the ATA.

7. The server according to claim 6, wherein the processor is further configured to, for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, compute the relative efficiency value by dividing the OTE improvement value into the ATA increasement value.

8. The server according to any one of claims 1 to 7, wherein the processor is further configured to select the one of the benchmark set of candidate batches and the one or more variant sets of candidate batches with a highest relative efficiency value, as the set of finalised batches of the two or more orders for the cluster.

9. The server according to any one of claims 1 to 8, wherein the processor is further configured to adjust the benchmark delivery upper bound based on one or more configurations which depend on at least one of a region, a time and an order type.

10. The server according to any one of claims 1 to 9, wherein the processor is further configured to select one of the benchmark delivery upper bound and the one or more delivery upper bounds corresponding to the set of finalised batches, as a finalised delivery upper bound for the cluster.

11. A method for facilitating batching orders of an on-demand service, the method comprising:

receiving a plurality of orders for the on-demand service from a plurality of computing devices;

classifying the plurality of orders into one or more clusters, based on a location relating to the plurality of orders;

for each cluster including two or more orders, obtaining a benchmark delivery upper bound for the cluster, generating a benchmark cluster corresponding to the benchmark delivery upper bound, generating one or more cluster variants, and generating one or more delivery upper bounds each corresponding to the one or more cluster variants by adjusting the benchmark delivery upper bound;

inputting the two or more orders of the cluster with the benchmark delivery upper bound of the benchmark cluster and the one or more delivery upper bounds of the one or more cluster variants into a vehicle routing solution engine;

obtaining a benchmark set of candidate batches of the two or more orders for the benchmark cluster and one or more variant sets of candidate batches of the two or more orders for each of the one or more cluster variants, from the vehicle routing solution engine;

computing a relative efficiency value for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches; and

selecting one of the benchmark set of candidate batches and the one or more variant sets of candidate batches based on the relative efficiency value, as a set of finalised batches of the two or more orders for the cluster.

12. The method according to claim 11, further comprising: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches,

computing an overall time efficiency (OTE) and a sum of actual time of arrival (ATA) of the two or more orders; and

computing the relative efficiency value based on the OTE and the sum of the ATA.

13. The method according to claim 12, further comprising: determining the OTE for the benchmark set of candidate batches as a benchmark OTE and the sum of the ATA for the benchmark set of candidate batches as a benchmark sum of the ATA.

14. The method according to claim 13, further comprising: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing the OTE by dividing a sum of travel time of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into a sum of a direct travel time of the two or more orders of the cluster.

15. The method according to claim 14, further comprising: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing an OTE improvement value by dividing the OTE into the benchmark OTE.

16. The method according to claim 15, further comprising: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing an ATA increasement value by dividing the sum of the ATA of each of the benchmark set of candidate batches and the one or more variant sets of candidate batches into the benchmark sum of the ATA.

17. The method according to claim 16, further comprising: for each of the benchmark set of candidate batches and the one or more variant sets of candidate batches, computing the relative efficiency value by dividing the OTE improvement value into the ATA increasement value.

18. The method according to any one of claims 11 to 17, further comprising: selecting the one of the benchmark set of candidate batches and the one or more variant sets of candidate batches with a highest relative efficiency value, as the set of finalised batches of the two or more orders for the cluster.

19. The method according to any one of claims 11 to 18, further comprising: adjusting the benchmark delivery upper bound based on one or more configurations which depend on at least one of a region, a time and an order type.

20. The method according to any one of claims 11 to 19, further comprising: selecting one of the benchmark delivery upper bound and the one or more delivery upper bounds corresponding to the set of finalised batches, as a finalised delivery upper bound for the cluster.