The on-demand warehousing problem

Sara Ceschia, Margaretha Gansterer, Simona Mancini & Antonella Meneghetti

Taylor & Francis
Taylor & Francis Group

The on-demand warehousing problem

Sara Ceschia [a], Margaretha Gansterer [b], Simona Mancini[b,c] and Antonella Meneghetti [a]

[a]Polytechnic Department of Engineering and Architecture, University of Udine, Udine, Italy; [b]Department of Operations, Energy, and Environmental Management, University of Klagenfurt, Klagenfurt, Austria; [c]Department of Science, Innovation and Technology, University of Eastern Piedmont, Alessandria, Italy

ABSTRACT

Warehouses are key elements of supply chain networks, and great attention is paid to increase their efficiency. Highly volatile space requirements are enablers of innovative resource sharing concepts, where warehouse capacities are traded on online platforms. In this context, our paper introduces the on-demand warehousing problem from the perspective of platform providers. The objective prioritises demand–supply matching with maximisation of the number of transactions. If there is a tie, the secondary objective maximises the number of suppliers matched with at least one customer and the number of customers that have matches within a specific threshold with respect to the minimum achievable cost. Besides the mathematical integer programming formulation, a myopic list-based heuristic and an efficient matheuristic approach are presented and benchmarked against the performance of a commercial optimisation solver. The impact of several parameters on the platform's objective is analysed. A particularly relevant finding is that the pricing flexibility on the demand side does not necessarily imply higher payments to the supply side. All data instances are made available publicly to encourage more researchers to work on this timely and challenging topic.
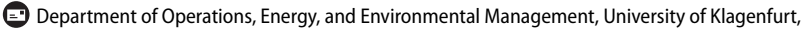
1. Introduction

Warehouses are key elements of supply chain design and management and great attention is paid to increase their sustainability (Meneghetti and Monti 2015) even by forcing renewable energy penetration into storage facilities (Meneghetti, Dal Magro, and Simeoni 2018). In a more urbanised world, space is a scarce resource, and solutions enabling vertical development and a better utilisation of floor surface such as automated storage and retrieval systems (Meneghetti, Dal Borgo, and Monti 2015) have been encouraged to deal with a tight real estate market. However, storage space is often under-utilised when companies experience low inventory levels and treats them as a sunk cost, while becoming a constrained resource during demand peaks, as underlined in recent surveys on inventory capacity utilisation and fluctuations (see Rogers et al. 2020). These variations in space utilisation open the door to concepts emerging from the sharing economy, where scare resources are used jointly by cooperating players. Hence, embracing these principles can be a viable solution to overcome such issues. Several application fields are discussed in the literature. These include joint facilities replenishment (e.g. Federgruen and Tzur 1994; Hezarkhani, Slikker, and Van Woensel 2018), production

(e.g. Gansterer and Hartl 2018), and transportation (e.g. Ke and Bookbinder 2018; Gansterer and Hartl 2020).

On-demand warehousing essentially connects companies that have available warehouse space and related services with companies that need additional storage capacity in a win–win solution (Forger 2018). An online platform acts as a third-party organisation or central mechanism, which matches supply offered with capacity requests. The value proposition consists of providing a B2B marketplace connecting renters and lenders of short-term storage and related services, which can be traded easily and with benefits for both sides (Tornese et al. 2020). From the supply side, enhanced resource utilisation and additional revenues can be gained, while from the demand side, flexibility and risk minimisation on acquiring capacity resources are enabled.

The main characteristics of on-demand warehousing with respect to traditional warehousing models (construct or outsource to a third-party logistics provider (3PL)/lease a facility) have been related by Pazour and Unnu (2018) to three main dimensions: (1) capacity granularity, (2) commitment granularity, and (3) access scalability. Capacity granularity is defined as the minimum capacity that can be acquired, which affects the

CONTACT Simona Mancini ✉ simona.mancini@aau.at 📧 Department of Operations, Energy, and Environmental Management, University of Klagenfurt, Universitätsstraße 65-67, Klagenfurt 9020, Austria

flexibility and scalability that can be gained by a company that needs additional storage space. While it is measured in full building units for the construction option and in square units for the 3PL/leasing one, capacity is measured in pallet positions in on-demand warehousing (Pazour and Unnu 2018). Commitment granularity refers to the minimum time interval the decision has to be maintained. Therefore, it can be typically associated with the payback period (at least 5 years) for the construction option, to 1–3 years for 3PL/leasing, while it drastically decreases to a weekly/monthly basis for on demand warehousing (Pazour and Unnu 2018). Access to scale is defined as the percentage of demand reachable within a given distance from resources, which impacts a company's response time to clients. While only a handful of facilities can be operated when owning them, which typically implies high transport costs to fulfil customer requests, on-demand warehousing allows access to more favourable locations in terms of distance (Pazour and Unnu 2018). The interrelation among these dimensions reflects on the cost structure of the warehouse models, moving from the large fixed costs of investments or long-term lease contracts, to the totally variable, pay-as-you-go costs of on-demand warehousing (Unnu and Pazour 2019). However, construction and lease options, if exploited at full capacity, show lower variable costs than on-demand warehousing. The latter in turn can rely on more flexibility for acquiring additional storage capacity, thus achieving spatial and temporal elasticity. This is because capacity can be scaled up and down and dynamically adjusted to meet demand when needed. Lower search costs and time for finding adequate warehouse space and faster moving goods into it should also be considered (Guillot 2018).

Moreover, in supply chain management, a mismatch between the variability of the stocking space demand and the inflexibility of the capacity supply often occurs. Indeed, both during both peak and off-peak periods, the capacity offered by traditional distribution/storage systems does not match the ideal capacity, causing extra cost or loss of potential profit. Such limits apply also to self-storage warehouses, which typically offer storage space units of a fixed size and features for one or multiple months, with material handling performed by the customers themselves, without interference of the warehouse personnel. From the demand side, storage units available per size may not fit the needs of the market, with clients being rejected or forced to rent storage units with more space than the required one. On the contrary, the smaller capacity and commitment granularity of on-demand warehousing allow customers to rent the actual required pallet positions, without assigning internal resources for material handling. From the

supply side, on-demand warehousing involves companies gaining extra-profits than their core business from available storage capacity variable over time and adjustable in size to their own internal requirements. This has motivated the development of on-line warehousing platforms, which act as centralised matching entity mediating in between supply and demand. Hence, we position our paper within the growing context of 'sharing economy'.

A typical example of a company interested in on-demand warehousing could be a start-up that does not want to invest in its own warehouse and does also not have the option to arrange long term contracts with partners to share a warehouse. These companies need warehouse space in a flexible way and on short notice. On the other hand, we observe companies running out of space due to demand fluctuations or price variability of their raw materials. This forces them to build-up safety stock and requires to rent extra space to store these exceeding materials. In these cases, companies cannot easily find partners to share warehouses since these variabilities typically occur in an unpredictable way.

By allowing warehouse locations to be closer to product destinations, on-demand warehousing can also enable additional transport savings cut down on last mile costs (FLEXE 2019), as well as the selection of more sustainable transport modes especially for fast deliveries (e.g. road, instead of air). Reducing distances and travel times can be particularly important when refrigeration is involved and further energy consumption is needed to preserve food safety and quality (Meneghetti and Ceschia 2020). Moreover, by a dynamic facility location model, it has recently been shown how hybrid networks that use on-demand in combination with traditional warehouse options reduce distribution costs by increasing capacity utilisation (Unnu and Pazour 2022). Maintaining a dynamic portfolio of warehousing options thanks to the sharing economy, enables firms to meet increasing customer service levels while making storage cost affordable (Rogers et al. 2020).

From the central mechanism's perspective, on-demand warehousing sets some challenges to be addressed (Pazour and Unnu 2018). A critical mass of both supply owners and demand requests should be achieved in a balanced way and network effects, which increase a solution value proportional to the numbers of users adopting it, should also be considered. The open nature of the market with low barriers creates a fluid set of supply owners, with different supply capabilities and characteristics varying over time. To simplify and fasten the decision process, one-size fits all, pre-determined terms of agreement are usually proposed by the platform, which sacrifices specific requests and negotiations in favour of standardisation. However, preferences on resource allocation

and management should be accommodated to induce repetitive participation of both capacity suppliers and demand requests.

The 13 operating platforms identified by an online market research completed in February 2020 have been clustered into three main groups (Tornese et al. 2020) according to the spectrum of offered services, moving from basic storage services offered by level 0 platforms to additional order fulfilment, palletisation, transportation, and inventory tracking of level 1, and finally to consultancy and KPIs monitoring offered by level 2. An evolutionary path towards more complex services is likely to be adopted by platforms in the near future to capitalise on the clients' loyalty.

While on-demand warehousing is an emerging field, to the best of our knowledge, no specific optimisation models for on demand warehousing have been developed to assist a platform on matching supply and demand requests. Moreover, it can be observed that many platforms adopt a simple matching process by revealing available space to all customers to allow for autonomy of all decision parties. This, obviously, bears the risk of overlooking beneficial matching and by this reduces the profitability of the platform. Our main research question is to investigate whether a smart matching platform can boost on-demand warehousing business simultaneously increasing customers and suppliers satisfaction.

As detailed in the literature review section, the few models dealing specifically with on-demand warehousing, in fact, adopt the perspective of either the supply side actors or the demand ones. In the latter case, models support the strategical choice of whether adopting the on-demand option for covering storage capacity requirements. In the former, the optimal storage capacity to be devolved from internal use to the on-demand business can be assessed.

Our paper closes this research gap and contributes to existing literature as follows:

(1) We introduce and mathematically formulate as an integer programming (IP) model the On-Demand Warehousing Problem (ODWP) from the perspective of platform providers. For this, we assume that the objective is to maximise current revenues while at the same time preserving future participation and profits.
(2) We provide an efficient and effective matheuristic.
(3) The assessment is conducted on both generated and real world data.
(4) We show that the proposed matheuristic approach yields very good solution quality in a short amount of time. Hence, it can be used as a decision support tool for the central mechanism.

(5) We obtain valuable managerial insights. For instance, it is shown that price flexibility on the demand side does not necessarily imply higher payments for the customers.

The paper is organised as follows. In Section 2, a literature review of models related to on-demand warehousing is provided, while in Section 3, the problem is defined and modelled. Solution methods are described in Section 4. Results for the reference scenario and the sensitivity analysis are provided in Section 5, while computational analyses are provided in Section 6. We discuss conclusions in Section 7.

2. Literature review

The literature review is organised into three sections. The first one is devoted to optimisation models applied to on-demand warehousing. The second one addresses supply matching problems arising in different contexts, whereas the third one is focused on related combinatorial optimisation problems sharing some similarity to ours (e.g. the Temporal Knapsack Problem and the Temporal Bin Packing problem).

2.1. Optimisation problems in on-demand warehousing

Despite its practical relevance, the problem of on-demand warehousing has received limited attention in the literature, from a quantitative and modelling point of view. In Unnu and Pazour (2022), the authors present a multi-period facility location model with the goal of providing the best solution among self-distribution, leasing, and on-demand warehousing. The very recent contribution by Shi, Yu, and Dong (2021) studies the decision problem of the quantity of space to allocate for on-demanding and to reserve for private usage, at each time frame. The authors propose a mathematical formulation and provide theoretical properties of the problem. The topic investigated in these papers can be seen as a prologue of the one studied in this work. In fact, we consider, the capacity allocated for on-demand warehousing or requested as input data, which is the final output of the decision problem addressed by Shi, Yu, and Dong (2021) and Unnu and Pazour (2022), respectively.

The research by Aouad and Saban (2021) addresses the online matching problem for a two-sided platform. This problem is close to ours, even if different assumptions are made. The platform operates real time, i.e. as soon as a customer request arrives, it is processed by the system, while in our case, we collect requests during a given period (e.g. a day) and perform the matching at

the end of it. Moreover, in Aouad and Saban (2021), the system provides the customer a set of possible options to choose from, while in our problem, the platform performs a direct assignment. The approach described by Aouad and Saban (2021) is more customer-centred, and this can be appreciated by customers, but can also provide suboptimal matchings, which can yield to a reduction of the platform profit and have a negative impact on both customers and suppliers satisfaction. Conversely, our approach, even if it can be considered more platform-oriented, aims at optimising the overall system performance. In fact, we maximise the profit of the platform, with customers' and suppliers' satisfaction as a secondary objective.

Interested readers are also referred to the huge body of economic literature on matching markets (e.g. Gale and Shapley 1962), where game theoretical aspects such as stability are discussed. In our study, we tackle the problem from an Operations Research perspective. We analyse how valuable matches can be found within huge solutions spaces, and how on-demand warehousing platforms might be influenced by changes in endogenous or exogenous parameters.

Zhong et al. (2020) study the case of an on-demand service platform, in which two types of customers are considered: (i) those who are congestion-sensitive (i.e. whose willingness to submit their request to the platform is influenced by the level of congestion of the platform itself that could create a delay in response time and in perceived quality of service) and (ii) those who are not. During the acceptance decision process, this sensitivity is considered by the platform. The authors propose and compare different platforms' behavioural strategies all aiming at maximising platform profit, but do not actually provide an optimisation model to analyse it.

Our research can also be related to self-storage warehousing; this industry collects companies (usually international corporations such as Shurgard) whose core business is to rent storage units for some periods (measured by months) to both private and business customers, who handle storage operations by themselves (Gong et al. 2013). At the beginning of a planning horizon (which typically ranges between one and four months) warehouse managers decide which new orders to accept or reject, based on the currently available storage units. The main difference between on-demand warehousing and self-storage warehousing is twofold. First, in self-storage, capacity is offered in blocks of a predetermined size, and this will force customers with small demands to rent a space much larger than that actually needed, while in ODW customers can rent exactly the space they need. Second, while self-storage warehousing concept aims at offering space to private persons

and companies, on-demand warehousing involves companies gaining extra-profits from available storage capacity which is variable over time and adjustable in size to their individual requirements. In Zhang et al. (2016), self-storage warehousing is modelled as a deterministic scheduling problem and solved by a branch-and-price algorithm. Differently from us, they consider a single warehouse offering storage units of various types, but each one can serve only one request in a period. In addition, the customers' satisfaction is not taken into account, given that the only objective is to maximise the provider's revenue. Gong et al. (2013) and Zhou, Gong, and Koster (2016) study the tactical problem of determining the facility design that better fits the matching between storage design (in terms of types and units per type) and market demand such that the final revenue of the self-storage provider is maximised. Another paper related to ours is Shi, Yu, and Dong (2021), which develops a dynamic and stochastic warehouse revenue management model for a retail platform (such as Amazon) that has to allocate its warehouse space for self-use and rent. However, in their model, the retail platform's revenue is generated from selling their own goods and storing and delivering external ones.

2.2. Supply matching problems

Recently, Boysen, Briskorn, and Schwerdfeger (2019) dedicated a survey to optimisation-based matching of demand and supply in a sharing economy focusing on a static and deterministic environment. They propose a classification scheme based on the tuple-notation, such that any matching problem can be described by three fundamental elements: supply (i.e. the features of the shared resources), demand (i.e. the features of customer requests), and the objective to optimise. Then, each basic element can be further described by some sub-attributes: supply can be detailed in terms of *movability*, *capacity*, and *availability*; the demand attributes concern *matching restrictions*, *dependent requests*, *sharing duration* and *time windows*; lastly, the natural objective, which is to maximise the total profit, is distinguished if the request's profit is always equal, or individual for each request, or depends on the resource assigned to it, or a more elaborate alternative objective is considered.

Following their notation, the OWDP can be described by the $|\text{IM}, k^{\lambda}, \text{AC}|\text{group}|\sum w|$tuple, where IM stands for immobile resources, k^{λ} describes a resource that can satisfy simultaneously many requests starting and ending at different times, AC denotes that a resource may not be available for sharing during some predefined periods, group specifies that there is a compatibility between

Table 1. Features of different supply/demand matching problems in a sharing economy. A star (*) indicates a joint usage of the shared resource.

| Business | Resource | | | Demand | | Ex. operators |
	Movability	Mult.	Type	Matching restrictions	Real-time/ reserv.	
Hotel booking	Immobile	1:1	std	indiv.	reserv.	Booking
Apartment renting	Immobile	1:1	indiv.	indiv.	reserv.	Airbnb
Car sharing	Mobile	1:1	std	indiv.	reserv.	ShareNow
Car pooling	Self moving	1:*	std	Group	reserv.	BlaBlaCar
Ride sharing	Self moving	1:*	std	Group	r-t/reserv.	Uber
Car parking	Immobile	1:*	std	Range	reserv.	Ampido
ODW	Immobile	1:*	std	Group	reserv.	OneVASTWarehouse

customers and resources, $\sum w$ represents the objective that is to maximise the number of serviced requests.

Table 1 shows the main features of different supply–demand matching problems in the spectrum of sharing applications, using some categories introduced by Boysen, Briskorn, and Schwerdfeger (2019). The first three columns refer to the characteristics of the resources shared, distinguishing between immobile, mobile, and self-moving (a resource can move even if no request is currently serviced), the multiplicity of the matching relation (exclusive or jointly usage) and the typology (standard or individual). Indeed, optimisation-based matching requires an exact and simple specification of each request to identify perfectly all available offers. As a consequence, the resource that is shared must be standard (e.g. storage space or bedrooms). On the contrary, the acceptability of an offer about an individual and complex resource, such as holiday apartment (e.g. Airbnb), may involve personal judgments that cannot be translated into a few attributes (Boysen, Briskorn, and Schwerdfeger 2019). The following two columns specify the attributes of the requests made by customers: matching restrictions (individual preferences, group preferences, customer's maximum acceptability range), and real-time or advance reservation. Finally, the last column reports some real operators currently involved in the corresponding business. According to BlaBlaCar, the world's leading community-based travel network, ride sharing, car sharing, and car pooling implement different business models. The car sharing operators offer the rental of a car owned by a third part, for short term, usually in urban context (Jorge and Correia 2013). On the contrary, car pooling assumes that the drivers share a vehicle for a trip with other passengers to split the travel costs (so without profit). Finally, the term 'ride sharing' concerns a generic sharing of car rides, including the purpose of making a profit. In addition, ride sharing systems can be real time when they support an automatic ride-matching process between participants on very short notice or even en-route (Furuhata et al. 2013). Our problem is more similar to the a car parking application (Shao

et al. 2016). However, in the ODWP there are specific matching restrictions such that supply and demand may not fit each other. In addition, suppliers may be available only during predefined time intervals.

2.3. Related combinatorial optimisation problems

The problem addressed in this paper can be modelled as an extension of the temporal knapsack problem (TKP). The TKP is a generalisation of the well-known Knapsack problem (KP) introduced by Bartlett et al. (2005). It arises in many applications ranging from logistics and production to telecommunications. With respect to the classical KP, in TKP, a temporal aspect is considered. Time is discretised in time slots and items are associated with profits and occupy a certain amount of capacity only for a specific subset of time-slots. Despite its practical relevance, this problem has generated a limited interest in the literature. Most of the papers addressing TKP present theoretical properties of the problem and focus on exact methods. In Caprara, Furini, and Malaguti (2013), the authors proposed a Dantzig–Wolf reformulation (DWR). An improvement of this algorithm has been published in Caprara et al. (2016), where a multi-level DWR is applied in a recursive fashion. A stabilised column generation procedure exploiting dual-optimal inequalities has been presented in Gschwind and Irnich (2017). Recently, Clautiaux, Detienne, and Guillot (2021) introduced an iterative dynamic programming approach to efficiently address TKP.

All the above-mentioned papers deal with the single-knapsack version of the problem and study the classical KP objective function dealing with collected profit maximisation. To the best of our knowledge, we are the first to address the multi-knapsack extension of the problem, to consider item-knapsack incompatibilities. Furthermore, we provide a hierarchical objective function, in which, in addition to the classical profit maximisation, a secondary objective is included, which reflects customer and supplier satisfaction. This objective is perfectly suitable for the on-demand warehousing problem, but can

also be adapted to describe a generalised service quality optimisation problem, which arises in many different applications.

A problem which is strictly related to the TKP is the temporal bin packing problem (TBPP), where the goal is to minimise the sum over all available time slots used in order to accommodate all assignable items. The TBPP was first introduced by De Cauwer, Mehta, and O'Sullivan (2016), where an application of workload management in data centres is discussed. Different formulations are proposed and a comparison of their performances is provided. A column generation-based matheuristic and a new objective function are proposed in Furini and Shen (2018). The authors benchmark its performances against various heuristic methods. In Furini and Shen (2018), the goal is to minimise the number of bins used, discarding the number of time slots for which it is actually used. However, while the objective function proposed by De Cauwer, Mehta, and O'Sullivan (2016) is more suitable to describe applications, where bins represent resources that can be rented for a limited amount of time (e.g. virtual machines in data centres), the one proposed by Furini and Shen (2018) represents cases in which a resource (e.g. vehicle or container) can be rented only for the whole time period, and partial renting is not permitted. This is a reasonable assumption in the field of production and logistics. In Dell'Amico, Furini, and Iori (2020), the authors proposed a branch-and-price approach and provided tight lower and upper bounds. A multi-objective extension of the TBBP is introduced in Aydin, Muter, and Birbil (2020). This problem is motivated by an application in cloud computing, where knapsacks represent virtual machines and items represent tasks to be executed. Each time a machine is activated, a fire-up cost occurs. The goal is to maximise the weighted sum of number of bins used and fire-up costs. The authors provide a mathematical formulation, an efficient heuristic, which iteratively solves smaller versions of the problem, and a column generation-based approach. Preprocessing techniques to reduce the number of variables involved in the model and an alternative formulation for the TBBP with fire-ups are presented in Martinovic, Strasdat, and Selch (2021). It is worth noting that the fire-ups issue does not apply in our on-demand warehousing problem, since the warehouses' capacity made available by suppliers is, generally, only a portion of their total capacity, which is temporarily unused. Hence, no activation costs arise.

To the best of our knowledge, we are the first to formally describe and solve the ODWP from the perspective of an on-demand warehousing platform provider, where user satisfaction of both the supply and the demand side is taken into account.

3. Problem definition

In this section, we formally describe the ODWP, which is based on a TKP as it is motivated in Section 2. There are three basic notions for the investigated problem:

Capacity supplier (CS): A company that has under-utilised resources in terms of storage capacity and decides to make them available for rent to derive additional profit. Each owner specifies a supply with the following features: the location of the warehouse, the period in which the space is available (in weeks), its capacity (in palletised unit loads), some additional features (e.g. temperature control for frozen/chilled products) or services offered (e.g. security, packing), and the storage fee, split into a component for storage (per pallet and per week) and for handling (per pallet).

Customer: A company that needs stocking space to fulfil its own requests quickly. Each customer has a position and a demand expressed in terms of required warehouse capacity (in palletised units) for a fixed time frame (in days with a minimum of a week), possibly with some required characteristics or services.

Online platform: A digital service that works as a marketplace by collecting capacity offers from suppliers and requests by the customers. It manages the interactions between them (such as contracts or services) and facilitates the matching between demand and supply. Its revenue mostly comes from a platform fee which depends on the number of transactions.

The ODWP consists of (i) assigning stocking space available at suppliers to customers, (ii) maximising the total number of requests satisfied, and (iii) both the satisfaction of customers and suppliers. It is subjected to the following constraints:

Complete request: The stocking space assigned to a customer must be sufficient to completely fulfil her demand for a number of consecutive days equal to her timeframe.

Warehouse capacity: The total requests assigned to a supplier must not exceed her warehouse capacity in each day of her availability period.

Customer–Supplier compatibility: There is a compatibility relation between each couple of a customer and a supplier. It states that the customer requirements (i.e. additional features or services) can be satisfied by the suppliers.

Minimum customer satisfaction: For each customer who has a processed request, a minimum level of

satisfaction must be guaranteed. This depends on the location and storage and handling fees of the assigned supplier.

The planning horizon is split into a set T of homogeneous time slots. A set K of capacity suppliers is considered. Each supplier submits the information regarding the stocking space available in each time slot, Q_{kt} to the platform. For each CS are also known the storage cost per pallet per day, $\hat{c}_k^s$, and the handling cost per pallet, $\hat{c}_k^h$. A set of customers, submitting a request of stocking space, I, is defined. Each customer i is characterised by a single request q_{it}, which is equal to its demand $\hat{q}_i$ during the time frame $t_i \in [t_i^{start}, t_i^{end}]$ and 0 otherwise. The length of the storage period requested by a customer i is identified as τ_i.

A customer–supplier compatibility matrix is available, such that customers i can be assigned to supplier k, if and only if $\phi_{ik} = 1$. Requirements on the warehousing space (e.g. refrigeration for chilled and frozen products) or on additional services (e.g. palletisation and transportation) are considered when creating the customer-supplier compatibility matrix ϕ.

Each assignment of a customer to a supplier is associated with a transportation cost c_{ik}^r computed from the platform that depends on the mutual distance, a handling cost, $c_{ik}^h = \hat{c}_k^h \hat{q}_i$ and a storage cost, $c_{ik}^s = \hat{c}_k^s \hat{q}_i \tau_i$. The global cost related to the assignment is computed as the sum of transportation, handling, and storage cost, $c_{ik} = c_{ik}^r + c_{ik}^h + c_{ik}^s$. We define the minimum assignment cost for customer i as $c_i^{min} = \min_{k \in K} c_{ik}$. A customer i is classified as satisfied if it is assigned to a supplier k such that $c_{ik} \leq (1 + \beta)c_i^{min}$, where β is named a satisfaction tolerance. The higher its value, the lower its satisfaction threshold. Based on this definition, a full satisfaction matrix can be computed in a pre-processing phase, in which σ_{ik} is equal to 1, if the assignment to supplier k fully satisfies customer i and 0 otherwise.

A supplier is considered to be satisfied when she is matched at least with one customer. However, it should be underlined that other satisfaction measures can easily be adopted. For instance, a supplier is satisfied if a minimum percentage of his capacity is fulfilled. The secondary term of the hierarchical objective function, in fact, aims at maximising the number of suppliers (customers) satisfied, whichever is the metric adopted to declare the supplier (customer) satisfaction.

If the same physical customer submits several disjoint requests, these are treated as separate customers. The centralised platform decides whether to accept a customer or not. The goal of the problem is to first maximise the number of transactions, i.e. successful matching between customers and suppliers, and second, a convex

Table 2. Notation.

Symbol	Description
T	Set of homogeneous timeslots
K	Set of suppliers
I	Set of customers
Q_{kt}	Stocking space available at supplier k on timeslot t
$\hat{c}_k^s$	Storage cost per pallet per day for supplier k
$\hat{c}_k^h$	Handling cost per pallet for supplier k
$\hat{q}_i$	Request of customer i
q_{it}	Request of customer i on timeslot t
$[t_i^{start}, t_i^{end}]$	Timeframe of customer i
τ_i	Length of the storage period for customer i
ϕ_{ik}	$\phi_{ik} = 1$ if customer i is compatible with supplier k, 0 otherwise
c_{ik}^r	Transportation cost for customer i if assigned to supplier k
c_{ik}^h	Handling cost for customer i if assigned to supplier k
c_{ik}^s	Storage cost for customer i if assigned to supplier k
c_{ik}	Global cost for assigning customer i to supplier k
c_i^{min}	Minimum assignment cost for customer i
β	Satisfaction tolerance
σ_{ik}	$\sigma_{ik} = 1$ if the assignment to supplier k fully satisfy customer i, 0 otherwise
y_{ik}	$y_{ik} = 1$ if customer i is assigned to supplier k, 0 otherwise
x_i	$x_i = 1$ if customer i is served, 0 otherwise
z_i	$z_i = 1$ if customer i is satisfied, 0 otherwise
u_k	$u_k = 1$ if supplier k is satisfied, 0 otherwise

combination of the percentage of customers and suppliers satisfied.

Table 2 summarises the notation introduced.

3.1. Mathematical formulation

The mathematical model can be formalised as follows:

$$\max \sum_{i \in I} x_i + \left[\alpha \frac{\sum_{i \in I} z_i}{|I|} + (1 - \alpha) \frac{\sum_{k \in K} u_k}{|K|} \right] \quad (1)$$

$$\sum_{i \in I} q_{it} y_{ik} \leq Q_{kt} \quad \forall k \in K \quad \forall t \in T \quad (2)$$

$$\sum_{k \in K} y_{ik} = x_i \quad \forall i \in I \quad (3)$$

$$y_{ik} \leq \phi_{ik} \quad \forall i \in I \quad \forall k \in K \quad (4)$$

$$z_i \leq \sigma_{ik} y_{ik} \quad \forall i \in I \quad \forall k \in K \quad (5)$$

$$u_k \leq \sum_{i \in I} y_{ik} \quad \forall k \in K \quad (6)$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in I \quad \forall k \in K; \quad (7)$$

$$x_i \in \{0, 1\} \quad \forall i \in I \quad (8)$$

$$z_i \in \{0, 1\} \quad \forall i \in I \quad (9)$$

$$u_k \in \{0, 1\} \quad \forall k \in K \quad (10)$$

The objective function reported in (1) is a hierarchical function in which the first goal is to maximise the number of executed transactions, i.e. the number of customers' requests fulfilled, the second goal is to maximise a performance indicator that can take values between 0 and 1. This indicator is a convex combination of two

sub-objective, i.e. the percentage of satisfied customers and of satisfied suppliers. This way, the second term of the objective function in squared brackets assumes a value between 0 and 1 and therefore is always subordinated to the first term. In fact, according to this objective function, a solution in which more customers are served will be always preferred to one in which the system serves less customers, even if the latter has a higher percentage of fully satisfied customers and/or suppliers. This means that the objective function first aims at maximising the number of customers accepted by the system, and among those solutions obtaining the optimal number of served customers, looks for a solution that maximises the number of actors (suppliers or customers) satisfied. Constraints (2) ensure that warehouses capacity is respected at each time-slot. Constraints (3) and (4) state that if a customer request is accepted, it must be assigned to one and only one warehouse, among those compatible. Constraints (5) check if the customer is fully satisfied by the solution provided by the platform or not. Finally, constraints (6) state if a supplier is satisfied

or not, i.e. if at least one customer has been assigned to it. Finally, constraints (7)–(10) specify variables' domain.

Note that we have scaled the secondary objective such that it can only have values between 0 and 1. This way, since the primary objective can only have integer values, the secondary objective intervenes only in the event of a tie on the primary objective. The adoption of a secondary objective allows us to differ among solutions having the same value of the primary objective, i.e. the same number of transactions. To better clarify this concept and the role of the secondary objective, we depict, in Figure 1, an example with four different solutions, having the same value for the primary objective but different values for the secondary objective.

In Figure 1(a) we depict a solution in which nine customers are covered (all satisfied except customer 6) and only two suppliers are satisfied (A and C). In subfigure (b), customers are all satisfied, but only suppliers A and C are satisfied, since B is not participating in any transaction. In (c), all the three suppliers are involved
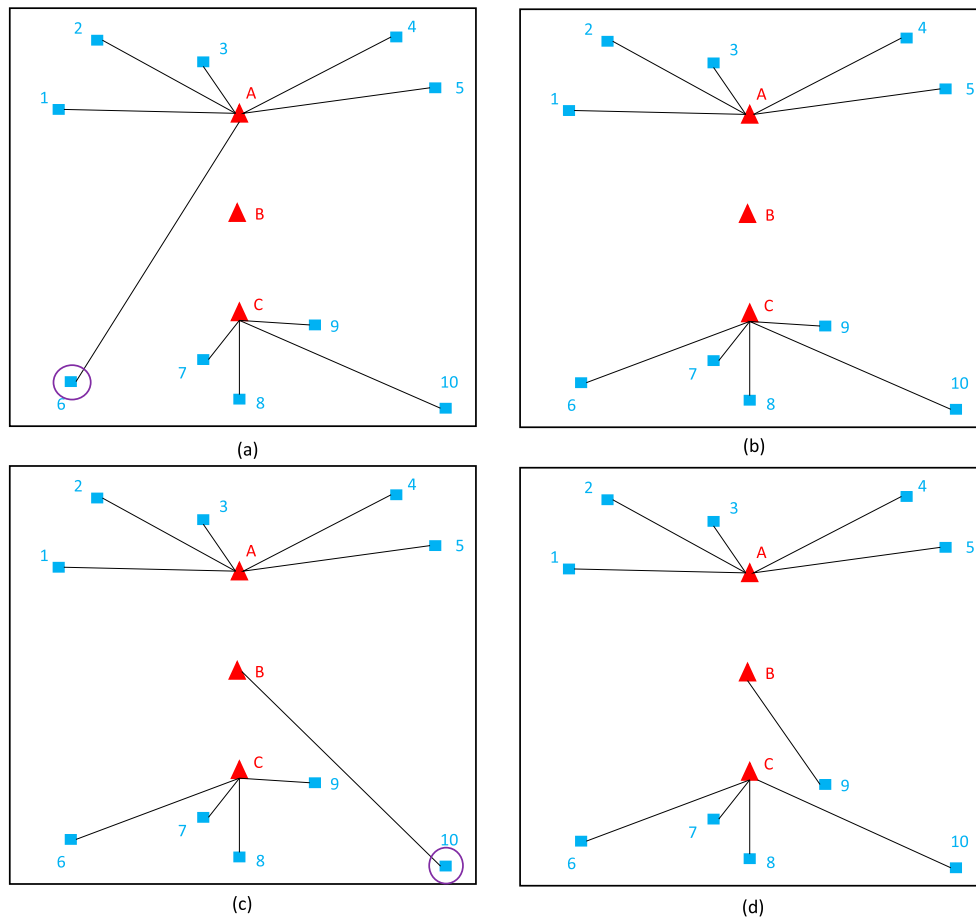
Figure 1. Solutions with the same value for the primary objective (# transactions executed) but different values for the secondary objective (suppliers' and customers' satisfaction)

(and then satisfied), while only nine customers are satisfied, since 10 is assigned to warehouse B which is very far from its location. Finally, in (d) we depict a solution in which all customers and all suppliers are satisfied. In fact, even if 9 is not assigned to its nearest warehouse, it is served by B which is located within a threshold distance. Hence, the customer can be considered satisfied. Clearly, the solution depicted in (d) dominates all the other solutions. It is worth to note that considering only the first objective, all these solutions will be considered optimal and therefore equivalent among each others. The secondary objective, instead, allows us to differ among those solutions, indicating as optimal only the one depicted in (d), which guarantees a total satisfaction of both customers and suppliers. For any value of parameter α, which indicates the level of importance of customers' satisfaction respect to suppliers' satisfaction, (d) would be always ranked as first among the 4 solutions, since it strongly dominates all the others. Also, (a) would be always ranked as fourth, while the dominance of (b) with respect to (c) depends on the value chosen for parameter α.

4. Solution methods

Following the analysis of computational complexity provided in Boysen, Briskorn, and Schwerdfeger (2019), the IP model proposed in Section 3.1 is strongly $\mathcal{NP}$-hard. In addition, the problem is modelled as a Temporal Bin Packing problem with side constraints, an extension of the Knapsack problem, which has been proved to be $\mathcal{NP}$-hard, too. Thus this justifies the usage of heuristic methods.

Beside the adoption of a commercial solver for IP optimisation problems, two different solutions methods have been analysed. In particular, a greedy algorithm is presented in Section 4.1, while a matheuristic is explained in Section 4.2.

4.1. Greedy algorithm

The greedy algorithm assigns the compatible supplier with the minimum global cost among those with sufficient capacity to each customer. In detail, for each customer i, the algorithm selects the *best* supplier k to assign to it, computing the global cost c_{ik} and considering only compatible suppliers. When supplier k is selected for customer i, its residual capacity is decreased by the demand $\hat{q}_i$ for the time frame $[t_i^{\text{start}}, t_i^{\text{end}}]$. If no supplier is able to completely satisfy the request, the customer is not served, and the greedy procedure goes to the next customer. Customers are considered in their original order, given that it reproduces the sequence of registration of the customer

requests in the online platform. The procedure finishes when all the customers have been evaluated.

This algorithm reflects a greedy and myopic assignment strategy as it is observed on many real-world matching platforms. That is, the demand side gets a list of possible suppliers and is free to select any of them according to subjective preferences.

4.2. Matheuristic

The broad family of *Matheuristics* comprises all solution approaches which hybridise exact methods and heuristic techniques (Maniezzo, Boschetti, and Stützle 2021). Within this family, we can identify a smaller class named *model-based heuristics* (MH), in which the mathematical model of the problem is exploited to explore very large neighbourhoods. Starting from an initial solution, at each iteration, a destroy operator is applied to partially destroy the current solution. Starting from the remaining solution, the model is exploited to optimally reconstruct it, looking for the best feasible solution obtainable. The destroy operator can be always the same, or can be picked, following predetermined selection criteria, from a pool of available operators. Such operators may be randomised or deterministic and their probability of being picked can be fixed or can vary based on their previous performance following a self-learning mechanism. MHs have been introduced in Mancini (2016) where an Adaptive Large Neighbourhood Search (ALNS), with several operators and performance-dependent operators' selection probabilities, has been proposed to solve a real-life multi-depot multi-period vehicle routing problem. Thereafter, MHs have been successfully applied to solve several combinatorial optimisation problems, in the field of vehicle routing (Marques et al. 2020; Mancini, Gansterer, and Hartl 2021; Mancini and Gansterer 2021), timetabling (Lindahl, Sorensen, and Stidsen 2018) and scheduling (Abu-Marrul et al. 2021), among others.

While in Mancini (2016), Lindahl, Sorensen, and Stidsen (2018), Marques et al. (2020) and Abu-Marrul et al. (2021), several operators are defined, the other papers deal with a single destroy operator, which is deterministic in Mancini, Gansterer, and Hartl (2021) while it is randomised in Mancini and Gansterer (2021). In the latter papers, it is shown that a very good performance can be achieved even using a random destroy mechanism.

In this paper, we propose an MH based on a single random destroy operator. The MH framework can be described as follows. We start from an initial solution S^0, which is computed by running the IP with a short time-limit, TL^{init}. This solution is kept as current best solution (CB). Then, at each iteration, we randomly select a subset

P of suppliers. All the customers who have been assigned, in the current solution, to suppliers included in P, are forced to remain assigned to the same supplier. However, all customers assigned to suppliers, which are not included in P, and all customers which were not served by any supplier, can be assigned to any supplier, if served, or can be rejected. We define $yfix_{ik}$ as the value that the corresponding decision variable, y_{ik}, takes in the current solution. The forced assignment can be imposed by adding the following constraints:

$$y_{ik} = 1 \quad \forall i \in I, \quad \forall k \in P : yfix_{ik} = 1 \qquad (11)$$

The resulting version of the IP is then run with a short time-limit, TL. If the best solution obtained so far (S^{iter}) is better than CB, it is kept as current solution. The algorithm terminates after a fixed number of iterations N_{iter}. The current best solution, CB, is kept as best solution. A pseudocode of MH is reported in Algorithm 1.

Algorithm 1 MH pseudocode

$S^0 \leftarrow$ run the model with a time limit TL^{init}
$CB \leftarrow S^0$
while $iter \leq N_{iter}$ **do**
    Randomly select a subset P of suppliers
    Add the related constraints 11 to the model
    $S^{iter} \leftarrow$ run the resulting version of the model with a time limit TL
    if $S^{iter} \geq CB$ **then**
        $CB \leftarrow S^{iter}$
    end if
end while

---

5. Results

In this section, the performance of the proposed demand–supply matching model is analysed from the platform's perspective. To this end, a basic scenario has been created as described in Section 5.1. A sensitivity analysis is then performed in Section 5.2, to understand how the central matching mechanism can be affected by the weights in the objective function, which can translate the platform's preferences for a given type of clients. The impact of supply–demand balance, as well as the flexibility on customer satisfaction levels are also investigated. The set of instances is composed by a basic scenario obtained by collecting the suppliers' data from an existing online platform, and a data set of generated instances representing other typical application scenarios. All instances are available at https://bitbucket.org/sceschia/on-demand-warehousing-optimization-problem/ for future inspection and comparison.

5.1. The basic scenario

The basic scenario has been created by collecting all the data related to suppliers (location, available space, minimum pallets for booking, handling and storage costs, services) from the online platform OneVASTWarehouse (OneVASTWarehouse 2021) simulating a customer request of 100 palletised units in London from April 1st, 2021, to May 26, 2021. Thus considering that a day corresponds to a timeslot, the planning period is equal to 8 weeks ($T = 56$). We obtained a list of 18 suppliers scattered all over Great Britain, with available capacity ranging from 100 to 9000 pallets and handling and weekly storage costs from 1.25 to 3 [£/pallet].

Unfortunately, the data about customers' requests were not available; as a consequence, we decided to generate it, trying to reproduce the peculiar features of the on demand warehousing market. For each customer i, the demand $\hat{q}_i$ was randomly chosen with probability $p_q = 0.05$ between 10 and 100 pallets, with $p_q = 0.70$ between 101 and 1000 pallets, and with $p_q = 0.25$ between 1001 and 2500 pallets. Regarding the customer timeframe, we first defined the length of the storage period τ_i, and subsequently the starting timeslot t_i^{start}. The value of τ_i is randomly selected between 2 and 4 weeks with probability $p_\tau = 0.6$, between 4 and 6 weeks with $p_\tau = 0.2$, and with $p_\tau = 0.1$ for a timeframe shorter than 2 weeks or longer than 6 weeks. The minimum booking period was fixed to 7 timeslots (a week) according to the real suppliers' data. Once the storage duration was set, the first day of a request t_i^{start} is randomly picked between $[0, T - \tau_i]$.

To compute the distances between each couple of customers and suppliers, the position of each customer was generated, drawing a point in a square with latitude included in $[50.735247, 55.811775]$ and longitude included in $\in [-3.000000, 1.000000]$. The transportation cost between each customer i and supplier k is then computed as $c_{ik}^r = 2\texttt{distance\_cost} \cdot d_{ik} \frac{\hat{q}_i}{\texttt{pallets\_per\_vehicle}}$, where $\texttt{distance\_cost}$ is the unitary transportation cost per km, d_{ik} is the distance between customer i and supplier k in km, $\texttt{pallets\_per\_vehicle}$ is the number of palletised units per vehicle, which was set equal to 30. Finally the compatibility matrix was randomly built, imposing a density of about 0.9.

Note that all random values are drawn from a discrete uniform distribution.

5.2. Sensitivity analysis

In this section, we analyse the impact of different parameters on the optimal solution for the basic scenario. The analysis concerns three parameters:

(1) α: the weight associated with the supplier/customer satisfaction within the secondary objective (see Section 5.3);

(2) ρ: the ratio between the suppliers' total offer and the customers' total demand (see Section 5.4);

(3) β: the customer' satisfaction tolerance, measuring the distance of the customer' cost in the optimal solution, from the minimum cost matching among compatible suppliers (see Section 5.5). Note that alternative measures (e.g. observed supplier quality) can easily be integrated as well but are not included in this computational study.

Three values for α, specifically [0.01, 0.5, 0.99], are considered. The first value represents the case in which priority is given to suppliers' satisfaction, the third one corresponds to a case in which priority is given to customers' satisfaction, while the value 0.5 corresponds to a situation in which the same importance is assigned to suppliers' and customers' satisfaction.

For what concerns the parameter ρ, we consider four values: [0.5,1,1.5,2]. Since this parameter represents the balance between total supply and demand, lower values of ρ correspond to excess of demand, while higher values to excess on the supply side.

Finally, five values of β varying from 0.1 to 0.5 are considered. A value of 0.1 means that customers are fully satisfied by all options whose cost is at most 10% higher than the cost corresponding to the minimum-cost assignment. Similarly, $\beta = 0.5$ indicates that all the assignments, costing up to 50% more than the cheapest option, are considered satisfactory for the customer. Hence, the lower the value of β, the lower the flexibility of the customers.

Overall, 12 versions of the basic scenario are considered, each one characterised by a variation of a single parameter. Results are reported in Table 3. Each row represents a variation of the reference scenario, for which we report (i) number of suppliers (K), (ii) number of customers (I), (iii) values of parameters α, ρ and β, (iv) number of satisfied suppliers, (v) satisfied customers, and (vi) total number of transactions executed.

5.3. Analysing the secondary objective weight

In this section, the role of the weight α, which affects the secondary objective of our hierarchical objective function (see Section 3.1), is investigated.

With α equal to 0.1 and 0.5, we obtain exactly the same results, with 24 proposed transactions, 15 fully satisfied customers and 11 suppliers used. Instead, focusing on customers' satisfaction ($\alpha = 0.99$) it is possible to increment the number of customers fully satisfied up to 17, with a decrease of the satisfied suppliers from 11 to 9. The fact that the obtained optimal solution, which gives priority to suppliers, is equal to the optimal solution in which suppliers and customers are equally important, is due to the fact that the satisfaction is measured in percentage. Since the number of customers is much higher than the number of suppliers, the benefit achieved satisfying an extra customer is lower in percentage, than that obtainable if one extra supplier is satisfied.

The proposed model can act as a decision support tool for the platform, who can vary the value of α according to seasonality. In fact, in a low-demand season, when the capacity offered by the suppliers is much greater than the customer demand, the platform can prioritise customers' satisfaction. Conversely, in a high-demand season, when the capacity offered by suppliers is lower with respect to the demand, it might be more important to satisfy suppliers, since they are becoming a scarce resource. In both situations, by acting on parameter α, the platform can build loyalty with customers or suppliers to incentivise them to use the platform again in the future. It's worth noticing that prioritising one of the two groups of actors involved (customers and suppliers), does not imply a large decrease in the satisfaction of the other group. This means that the two objectives

Table 3. Optimal results on the basic scenario for different values of α, β, ρ, being K the number of suppliers and I the number of customers.

Param. variation	K	I	α	ρ	β	Satisfied suppliers $\sum_k u_k$	Satisfied customers $\sum_i z_i$	Transactions executed $\sum_i x_i$
α	18	65	0.01	1	0.5	11	15	24
	18	65	0.5	1	0.5	11	15	24
	18	65	0.99	1	0.5	9	17	24
ρ	18	150	0.5	0.5	0.5	14	27	41
	18	65	0.5	1.0	0.5	11	15	24
	18	46	0.5	1.5	0.5	8	8	15
	18	34	0.5	2.0	0.5	7	6	12
β	18	65	0.5	1	0.1	11	2	24
	18	65	0.5	1	0.2	11	7	24
	18	65	0.5	1	0.3	11	10	24
	18	65	0.5	1	0.4	11	12	24
	18	65	0.5	1	0.5	11	15	24

are not completely in conflict. Therefore, it is possible to achieve good satisfaction levels for both groups. We would like to remark that the primary goal for the platform is to maximise its own profit, i.e. the number of potentially executed transactions, which is always pursued by the hierarchical objective function. In fact, the value of α does not affect the number of total executed transactions, which is always maximised, but allows to keep the profit for the platform fixed while maximising the actors' satisfaction with consequent long term beneficial effects for the platform itself. To further confirm these results, we perform simulations varying α also on the 20 generated instances used for the computational analysis (see Section 6.1 for a detailed description of the generated instances). Such instances differ from the basic scenario for the number of time slots T in the planning horizon, of suppliers K and of customers I, and the average ratio between the daily demand and the daily supply (see Table 5).

Table 4 shows the results obtained by the IP solver with a time limit of 1 hour. For each instance and for each possible value of α, we report the disaggregated value of the objective function in terms of the primary and secondary objectives. That is, (i) the total number of transactions executed ($\sum_i x_i$) and (ii) the secondary objectives, i.e. the number of satisfied suppliers ($\sum_k u_k$) and the number of satisfied customers ($\sum_i z_i$).

As already noticed for the basic scenario, we find that in many cases, the total number of satisfied suppliers remains the same for $\alpha = 0.01$ and $\alpha = 0.5$, while the number of satisfied customers slightly increases. Only for $\alpha = 0.99$, the number of satisfied suppliers drops in favour of the customer satisfaction.

Table 4. Results on generated instances for $\alpha \in [0.01, 0.5, 0.99]$.

Instance	Satisfied suppliers $\sum_k u_k$			Satisfied customers $\sum_i z_i$			Transactions exec. $\sum_i x_i$		
α	0.01	0.5	0.99	0.01	0.5	0.99	0.01	0.5	0.99
test-4-50	6	6	5	21	22	23	41	41	41
test-4-100	12	12	12	23	23	23	79	79	79
test-4-250	33	33	33	93	116	118	205	205	205
test-4-500	62	62	54	289	294	297	445	445	445
test-4-1000	130	130	123	470	471	478	890	890	891
test-8-50	6	6	6	24	24	24	27	27	27
test-8-100	11	11	11	34	35	35	61	61	61
test-8-250	22	22	19	60	64	67	152	152	152
test-8-500	56	56	53	132	135	139	376	376	376
test-8-1000	110	110	101	334	334	339	740	741	741
test-12-50	4	4	4	19	20	20	36	36	36
test-12-100	10	10	7	7	7	10	22	22	22
test-12-250	18	18	16	64	65	67	144	144	144
test-12-500	50	50	46	184	196	200	334	334	334
test-12-1000	91	91	78	293	300	315	689	689	688
test-16-50	4	4	4	30	30	30	33	33	33
test-16-100	8	8	8	17	17	17	46	46	46
test-16-250	15	15	15	60	63	63	153	153	153
test-16-500	37	37	36	145	152	154	411	411	411
test-16-1000	80	80	79	286	292	290	736	736	737

Table 5. Features of the generated instances.

Instance	T	K	I	γ
test-4-50	28	7	50	0.797
test-4-100	28	16	100	0.759
test-4-250	28	37	250	0.760
test-4-500	28	69	500	0.742
test-4-1000	28	136	1000	0.746
test-8-50	56	8	50	0.842
test-8-100	56	13	100	0.717
test-8-250	56	28	250	0.815
test-8-500	56	61	500	0.709
test-8-1000	56	124	1000	0.716
test-12-50	84	5	50	0.576
test-12-100	84	12	100	0.878
test-12-250	84	24	250	0.725
test-12-500	84	50	500	0.666
test-12-1000	84	95	1000	0.692
test-16-50	112	5	50	0.363
test-16-100	112	9	100	0.683
test-16-250	112	19	250	0.680
test-16-500	112	37	500	0.661
test-16-1000	112	81	1000	0.675

It should also be noted that the number of transactions executed is almost insensitive to the changes of α, given that this parameter controls the balance between the secondary objectives. Only for instances with 1000 customers, the values of the total number of transactions executed are slightly different. However, it should be underlined that no proven optimal solution can be found within the given time limit for the IP optimisation solver for such instances.

We can conclude that the experiments conducted on different generated instances confirm the observations about the weight α derived for the basic scenario.

5.4. Analysing the supply–demand balance

Increasing parameter ρ (i.e. the ratio of supply and demand) yields a variation on the number of customers (keeping fixed suppliers' availability) from 34 to 150. It should be noted that even if in the case with 34 customers only 12 can be served, this does not mean that the other 22 customers are rejected because the capacity of suppliers is totally used, but because there is no match between the temporal availability of capacity and the periods in which the requests have to be fulfilled. In fact, when increasing the number of customers, we can note a significant increment on the number of transactions that can be executed. Trends of variation for the number of transactions, customers and suppliers satisfied, with the increment of the number of customers, are reported in Figure 2.

As it can be noticed from the figure, the increment of transactions is more than proportional to the increment of customers, as well as the increment of satisfied customers, which shows a very similar trend. Conversely, the increment of satisfied suppliers shows – proportional
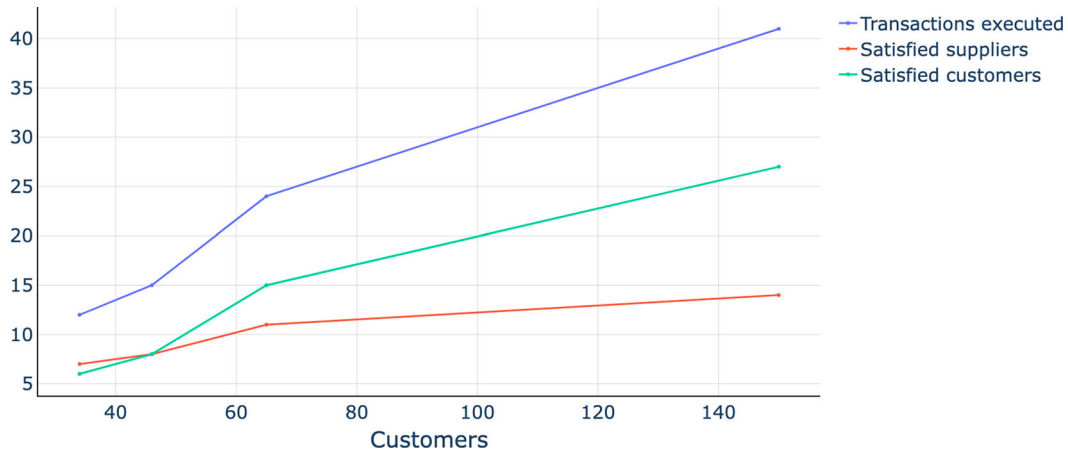
Figure 2. Results on the basic scenario for increasing number of customers. Results on the basic scenario for increasing number of customers, showing a more than proportional increment on the number of transactions that can be executed.
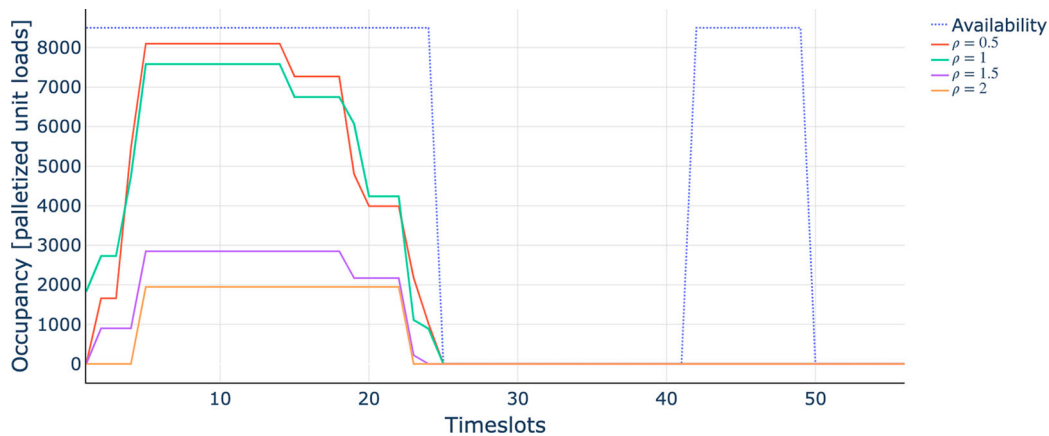
Figure 3. Occupancy profile for warehouse 14 of the scenario, for different values of ρ on the planning horizon. Occupancy profile for warehouse 14 of the scenario, for different values of rho on the planning horizon. For some time periods, the occupancy level is null because currently there are no compatible customers.

to the number of customers – very low growth. The limited number of suppliers involved in the transactions may be due to the fact that some of them are strictly dominated by other suppliers, because they are located very far from the customers' area or because they are not offering competitive prices. The reason could also be related to the capacity availability's profile of the supplier. In fact, in some cases, for suppliers having a very limited capacity or a larger capacity but available only for a very limited period, it is very difficult to find compatible customers. To demonstrate this effect, in Figure 3, we report the occupancy profile, for different values of ρ, for supplier 14. This supplier has two availability windows – from days 1 to 25 and from days 41 to 50. We can notice that, in the first window, for lower values of ρ, i.e. a larger number of customers, the occupancy rate significantly increases, getting close to total occupancy. Conversely, although there is space available over time frame [41, 50], the occupancy level is null, for each value of ρ, because currently there are no compatible customers, whose requests can

be completely fulfilled within this period. This is due to the limited duration of the second availability window.

Another frequent cause of null occupancy recorded in a period, for a supplier, is the highly global capacity availability in that period, which generates competition among suppliers. In fact, suppliers with higher storage and loading/unloading costs, or located very far from the customers area, may result to be not competitive. To overcome this issue, suppliers who are not matched with any customer could be invited by the platform to provide more competitive prices in order to stay in the market.

5.5. Analysing the satisfaction tolerance

In this section, we investigate how a variation of the customer satisfaction tolerance affects the supply–demand matching performance of the proposed optimisation model. The parameter β defines the maximum allowed increment, with respect to the minimum feasible cost, to consider a customer fully satisfied, as described in
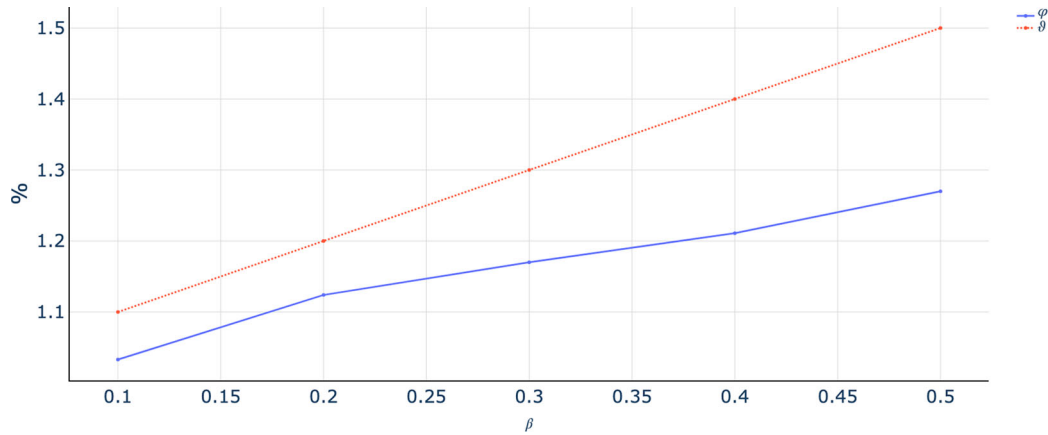
Figure 4. Graphics of the average increment of cost for the fully satisfied customers respect to the average minimum cost achievable, φ, (blue) for different values of β, in comparison with the threshold ϑ (red). Average increment of cost for the fully satisfied customers respect to the average minimum cost achievable, for different values beta, in comparison with the threshold theta.
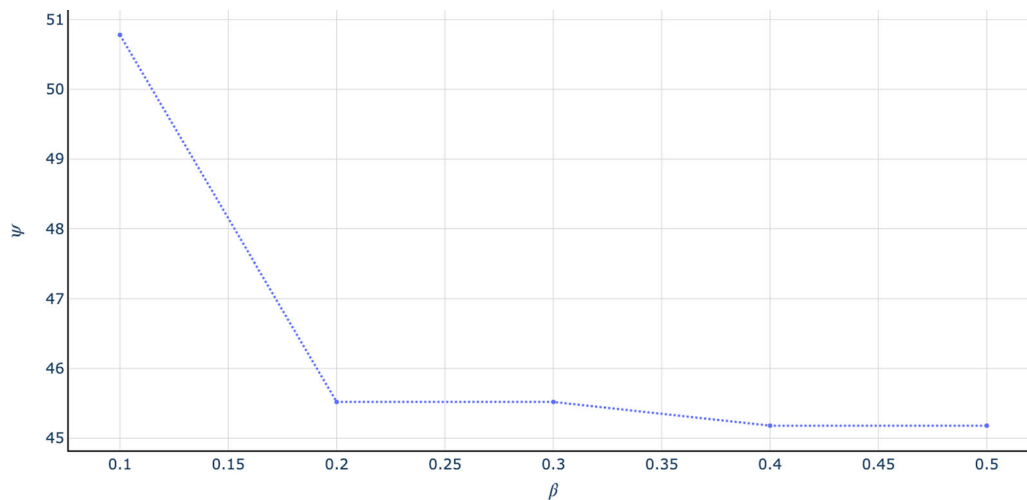
Figure 5. Graphics of the average increment of cost for all the accepted customers respect to the average minimum cost achievable (ψ), for different values of β. Average increment of cost for all the accepted customers respect to the average minimum cost achievable, for different values of beta. It shows that increasing the value of beta, the cost paid by the fully satisfied customers tends to increase, while the average cost paid by all the customers decrease.

Section 3. From β, the values of parameters σ in Equation 5, which identifies satisfied customers, are derived.

Obviously, the larger this value, the larger the flexibility on demand–supply matching and the larger the number of fully satisfied customers. What is interesting to point out is that a larger flexibility does not necessarily imply a linearly proportional increment of costs for the customers and this is a strong insight.

In Figure 4, we report the plot of: (i) φ, which indicates the average increment of cost observed by fully satisfied customers with respect to the minimum cost achievable, for each customer, ($c_i^{\min}$), and (ii) ϑ, which is the maximum increment of cost allowed for considering customers fully satisfied and corresponds to the average over all the customers i $(1 + \beta c_i^{\min})$. It can be observed from the graphics that the distance between the

two plots significantly increases with the growth of β, since φ rises only slightly (as represented by the slope of the blue line), whereas ϑ linearly increase with β. The managerial insight we can derive from this, is that even if the platform demands a high flexibility level of customers (in terms of willingness to accept a higher percentage increment of cost with respect to the cheapest option), the average increment of cost for fully satisfied customers, in the optimal matching, is quite low, since only very few of them receive an expensive matching. Another interesting insight is shown in Figure 5, where the average increment of cost for all accepted customers (fully satisfied and not) with respect to the average minimum cost achievable, ψ, is plotted. We can show that while φ is increasing with the increase of β, ψ shows an opposite behaviour. This means that, increasing the value of β, and the cost paid by the fully satisfied customers tends to increase, while the

average cost paid by all the customers decrease. Hence, the platform can choose if it prefers to have few customers with a very high level of satisfaction at the cost of a lower generalised satisfaction level, or to achieve a higher average level of global satisfaction but without peaks. The comparison of the graphics of the behaviour of φ and Ψ at the variation of β, shows that the best compromise, in this case, would be to choose an intermediate value of β, i.e. $\beta = 0.2$, such that Ψ is very near to the lowest value, and simultaneously, φ is still quite low. However, β could be seen as a tool for the platform, to influence the matching mechanism to achieve secondary goals which can depend on the specific situation of the market and seasonality, while keeping constant the primary objective, which is the number of transactions executed.

6. Computational analysis

The proposed solution methods are compared in terms of solution quality and computational times. For the IP solver, the default FICO Xpress configuration (v. 8.11.1) is used with a time limit of 3600 seconds. The matheuristic is implemented in the Xpress-Mosel environment and the MIP models within are run with the default FICO Xpress configuration (v. 8.11.1). For each one of these models, a time limit of 7 seconds is imposed. The greedy algorithm is written in C++ and compiled using the GNU C/C++ compiler (v. 4.9.3) under Ubuntu Linux (v. 18.04). All experiments were run on an Intel® i7-7700 (3.60 GHz) machine.

6.1. Generated instances

With the aim to analyse our model and solution methods on scenarios with different size and features, we developed a parametrised instance generator that is able to create artificial test cases, which can be considered an extension of the basic scenario. The generator receives as input the number of timeslots T, the number of customers I, and the ratio ρ between the total supply and the total demand, and it is able to produce an instance with the desired characteristics. The data about customers was generated as described for the basic scenario in Section 5.1.

Regarding the capacity suppliers, the generator simulates 'spotted' space availability, resulting from under-utilised warehouse space for short time. Suppliers are created one at a time, up to the threshold value ρ is reached. In detail, for each new supplier and each week of the planning period, we first decide if some space is available that week with probability equal to 0.5, and then we possibly choose at random a space available between 5000 and 10,000 pallets. Storage and handling costs are randomly set according to the minimum and maximum values collected for the basic scenario. The supplier locations are randomly generated in the same area defined in Section 5.1 for customers.

We generated 20 instances whose main features are reported in Table 5: number of timeslots T, suppliers K, customers I, and average ratio, γ, between the daily demand and the daily supply ($\gamma = \frac{1}{T} \sum_t \frac{\sum_i q_{it}}{\sum_k Q_{kt}}$). The name of each instance follows this pattern: test-w-I, where w is the number of weeks (e.g. $w = 4$, if $T = 28$) and I is the number of customers.

6.2. Comparison with the greedy algorithm

We compare the performance of the IP solver with those of the greedy procedure described in Section 4.1. Table 6 reports the comparative results between the exact solver and the greedy algorithm in terms of (i) value of the objective function (of $_{IP}$ and of $_{Greedy}$), (ii) time for convergence to the best solution (tf), (iii) total computational time in seconds (time), (iv) and the percentage gap computed as $\Delta = \frac{(\text{of}_{IP} - \text{of}_{Greedy})}{\text{of}_{IP}}$. For the greedy algorithm, the computational time is not reported, given that it is negligible (some milliseconds). The value of the parameter α was fixed to 0.5, to balance between suppliers' and customers' satisfaction. The second column reports also the upper bound (ub) computed by the exact solver. Proven optimal solutions are highlighted in bold.

Table 6 indicates that the value of the objective function grows with the size of the instances, which depends on the number of customers. On the contrary, if the planning horizon (T) is enlarged and made equal to the number of customers, the instance becomes easier because

Table 6. Comparison between IP model and greedy algorithm.

Instance	IP model				Greedy	
	ub	of	tf	time	of	Δ
test-4-50	41.324	**41.324**	0.01	0.01	39.115	5%
test-4-100	79.245	**79.245**	0.07	98.47	67.055	15%
test-4-250	205.339	**205.339**	50.06	50.23	175.078	15%
test-4-500	445.37	**445.37**	88.14	88.14	399.091	10%
test-4-1000	894.011	890.357	497.18	3600.00	776.069	13%
test-8-50	27.308	**27.308**	0.00	0.00	26.115	4%
test-8-100	61.299	**61.299**	0.02	0.02	51.088	17%
test-8-250	153.077	152.26	75.69	3600.00	117.055	23%
test-8-500	376.461	376.297	61.68	3600.00	298.053	21%
test-8-1000	741.518	741.305	1227.20	3600.00	617.058	17%
test-12-50	36.3	**36.3**	0.01	0.01	31.095	14%
test-12-100	22.226	**22.226**	0.01	0.01	19.025	14%
test-12-250	144.253	**144.253**	0.03	0.04	118.055	18%
test-12-500	335.166	334.348	51.13	3600.00	312.082	7%
test-12-1000	690.515	689.314	2509.25	3600.00	557.06	19%
test-16-50	33.35	**33.35**	0.00	0.00	33.135	1%
test-16-100	46.265	**46.265**	0.00	0.01	42.04	9%
test-16-250	153.26	**153.26**	0.63	4.91	124.047	19%
test-16-500	412.869	411.326	1355.46	3600.00	336.07	18%
test-16-1000	739.394	736.32	313.52	3600.00	623.056	15%

the same number of requests insists on a more extended period, reducing the competition between customers. It should be noted that all instances with up to 250 customers (except test-8-250) can be optimally solved by the IP model in less than 1 minute. Overall, the IP model can rapidly provide near-optimal solutions also for medium and large cases, given that the maximum loss with respect to the upper bound is less than 1%. Comparing the results obtained by the IP solver with those of the greedy algorithm, it turns out that the adoption of the optimisation model allows to obtain an average improvement of 14% of the objective function compared to the application of the greedy procedure. It should be noted that the greedy algorithm reflects a simple list-based solution as basic matching mechanism that it is typically applied by online platforms in the logistics industry.

Table 6 highlights that for larger instances, i.e. those with 500 or 1000 customers, the IP model is not able to find the optimal solution in 1 hour or the converge time is longer than 100 seconds. As a consequence, for each of these instance families called *hard*, we generated a set composed of 10 additional instances to perform further comparative experiments between different solution methods.

6.3. Comparison with the matheuristic on hard instance sets

A summary of the results on hard instances is provided in Table 7. The scores reported are average values (objective function and computational time) obtained on each instance set (marked with the capital letter Test-∗) composed of 10 different instances. Note that in some cases (Test-8-500, Test-12-500), even if the average running time of the IP model is inferior to the timeout of 3600 s, the best values are obtained by the matheuristic. This happens because the value reported is the average computational time on 10 different instances, and the corresponding value of the objective function is the average between optimal and near optimal solutions over all the instance set.

Given that the matheuristic is not deterministic, we run 10 repetitions for each instance and we collected the worst, average, and best result over the whole set. The last column reports the gap between the IP and MH methods ($\Delta = \frac{(\text{of}_{\text{IP}} - \text{avg}_{\text{MH}})}{\text{of}_{\text{IP}}}$).

We can observe that MH provides robust and high-quality solutions. The performance gap with respect to the exact solver does not exceed 0.04% in any of the sets. In particular, MH obtains even better results than the IP solver for families Test-8-1000 and Test-16-1000, although the gap of IP with respect to the upper bound is always very small (max 0.85%). In addition, MH reduces the computational time by 98%, with an average converge time of 37 seconds.

In conclusion, the numerical results obtained on the generated data sets demonstrate that the proposed solution method is able to efficiently and effectively solve practical problems in real world size.

7. Conclusion

This paper is the first to formally describe and solve the on-demand warehousing problem (OWDP) from the perspective of a platform provider. The problem is embedded in the emerging field of operations management in the sharing economy, where scarce resources are shared among different players of supply networks. The OWDP's objective is to maximise the platform's number of transactions. If there is a tie, a secondary objective maximises that suppliers are matched with at least one customer and that users have matches that meet a minimum cost tolerance. Besides the mathematical formulation, two (math)heuristic solution methods were presented and benchmarked against the performance of a commercial optimisation solver. The experimental results on generated instances showed that the solver provides optimal solutions for small- and medium-scale cases, and near-optimal solutions for large-scale problems within a time limit of 1 hour. However, the proposed matheuristic method was able to obtain robust and

Table 7. Comparison between the IP model and the matheuristic ($\alpha = 0.5$) on hard instance sets. Computational times are given in seconds.

Instance set	IP model			Matheuristic					
	of	tf	time	worst	avg	best	tf	time	Δ
Test-4-500	417.736	441	3600	417.311	417.581	418.211	42	75	0.04%
Test-4-1000	857.340	441	3600	856.016	857.065	858.015	42	79	0.03%
Test-8-500	357.404	925	3240	357.189	357.319	357.49	26	66	0.02%
Test-8-1000	776.410	718	3600	775.799	776.478	777.298	41	79	−0.01%
Test-12-500	364.116	522	2542	363.806	364.076	364.206	29	57	0.01%
Test-12-1000	738.121	632	3600	737.312	737.972	738.512	46	79	0.02%
Test-16-500	343.723	563	2340	343.511	343.611	343.711	27	49	0.03%
Test-16-1000	768.730	972	3600	768.317	768.907	769.618	42	78	−0.02%
Avg.	577.948	652	3265	577.408	577.876	578.383	37	70	0.02%

high quality solutions for all test cases, while reducing the computational time of 98% with respect to the exact method. The greedy heuristic approach, which reflects a basic list-based matching procedure as it is typically observed on, e.g. platforms in the logistics sector, yields considerably worse results than both alternative approaches. This allows us to answer our main research question, which is to investigate whether a smart matching platform can boost on-demand warehousing business simultaneously increasing customers and suppliers satisfaction. Indeed, comparing the results of the Greedy approach, representing a myopic list-based matching procedure used in practice, with the solutions of the proposed optimisation approach, we can observe that by exploiting a smart strategy it is possible to strongly improve the performances of the matching system by up to 21%.

To provide managerial insights, an additional computational study using a basic scenario was conducted. The impact of several parameters on the platform's objective is analysed. A particularly relevant finding is that the pricing flexibility on the demand side does not necessarily imply higher payments to the supply side. All data instances were made publicly available in order to encourage more researchers to work on this timely and challenging topic.

Future research is recommended to analyse the problem in a dynamic context, where customer requests and supplier availability are associated with a registration date and the matching has to be adjusted periodically in response to real-time events. In such settings, it might be crucial to automatically learn from past observations to anticipate future customer demands. In addition to that, splitting demand over multiple suppliers might be a viable approach to increase the rate of accepted requests. Furthermore, it would be interesting to tackle the problem as a more decentralised bi-level matching problem allowing for more autonomy by both customers and suppliers. In this model, first a list of potential suppliers is presented to each customer, from whom she selects the preferred one; then, each of the supplier reviews the received requests and chooses whether to accept a request and complete a transaction.

Finally, it would be interesting to embed the uncertainty of some input parameters (e.g. supply and demand) into the formulation and devise a robust optimisation method.

Data availability statement

The data that support the findings of this study are openly available at https://bitbucket.org/sceschia/on-demand-warehousing-optimization-problem/ for future inspection and comparison.

Disclosure statement

Funding

Notes on contributors

Sara Ceschia is Assistant Professor of Operations Research at Polytechnic Department of Engineering and Architecture, University of Udine, Italy. Her current research focuses on local search-based meta-heuristics for the solution of combinatorial optimisation problems in the fields of logistics, timetabling, healthcare, scheduling and engineering. She is also interested in the development of hybrid optimisation methods and automatic algorithm configuration techniques.
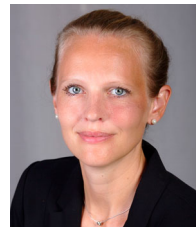
Margaretha Gansterer is Full Professor of Business Administration with a focus on Logistics and Operations Management at the University of Klagenfurt. Since October 2019, she is Head of the Department of Operations, Energy, and Environmental Management and since January 2022 Dean of the Faculty of Management and Economics. She has published about 30 peer-reviewed scientific publications in the field of Operations Management. Her research interests lie in efficient solution approaches for complex decision problems as well as in mechanisms for decentrally planned collaborations in production and logistics.

Simona Mancini is Assistant Professor in Computer Science since 2021 at the University of Eastern Piedmont, Italy. She teaches as External Lecturer at the University of Klagenfurt since 2021. She is author of about 30 papers in the field of Operations Research. Her current research interest focuses on the design of exact and matheuristic methods for complex optimisation problems arising in different fields, such as Transportation, Logistics, Last-mile delivery, Healthcare and Tourism management.

Antonella Meneghetti is Associate Professor of Industrial Systems Engineering at Polytechnic Department of Engineering and Architecture, University of Udine. She got her Ph.D. degree in Industrial Engineering from University of Padua. She has several publications in the field of sustainable logistics, with a particular focus on optimisation of energy efficient automated warehouses, refrigerated transports and renewable energy integration; sustainable production planning; industrial and urban symbiosis.

ORCID

Sara Ceschia http://orcid.org/0000-0003-1191-1929
Margaretha Gansterer http://orcid.org/0000-0002-0039-4519
Antonella Meneghetti http://orcid.org/0000-0002-9475-0763

References

Abu-Marrul, V., R. Martinelli, S. Hamacher, and I. Gribkovskaia. 2021. "Matheuristics for a Parallel Machine Scheduling Problem with Non-anticipatory Family Setup Times: Application in the Offshore Oil and Gas Industry." *Computers & Operations Research* 128.

Aouad, A., and D. Saban. 2021. "A Framework for An Efficient Implementation of Logistics Collaborations." *SSRN Electronic Journal,* 1–33. doi:10.2139/ssrn.3712553.

Aydin, N., I. Muter, and S. I. Birbil. 2020. "Multi-objective Temporal Bin Packing Problem: An Application in Cloud Computing." *Computers & Operations Research* 121.

Bartlett, M., A. M Frisch, Y. Hamadi, I. Miguel, S. A. Tarim, and C. Unsworth. 2005. "The Temporal Knapsack Problem and Its Solution." In *CPAIOR 2005: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, edited by R. Bartak and M. Milano, Vol. 3524, 34–48. Springer.

Boysen, Nils, Dirk Briskorn, and Stefan Schwerdfeger. 2019. "Matching Supply and Demand in a Sharing Economy: Classification, Computational Complexity, and Application." *European Journal of Operational Research* 278 (2): 578–595.

Caprara, A., F. Furini, and E. Malaguti. 2013. "Uncommon Dantzig–Wolfe Reformulation for the Temporal Knapsack Problem." *INFORMS Journal on Computing* 25 (3): 560–571.

Caprara, A., F. Furini, E. Malaguti, and E. Traversi. 2016. "Solving the Temporal Knapsack Problem Via Recursive Dantzig–Wolfe Reformulation." *Information Processing Letters* 116 (5): 379–386.

Clautiaux, F., B. Detienne, and G. Guillot. 2021. "An Iterative Dynamic Programming Approach for the Temporal Knapsack Problem." *European Journal of Operational Research* 293 (2): 442–456.

De Cauwer, Milan, Deepak Mehta, and Barry O'Sullivan. 2016. "The Temporal Bin Packing Problem: An Application to Workload Management in Data Centres." In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, 157–164. San Jose, CA, USA.

Dell'Amico, M., F. Furini, and M. Iori. 2020. "A Branch-and-Price Algorithm for the Temporal Bin Packing Problem." *Computers & Operations Research* 114.

Federgruen, Awi, and Michal Tzur. 1994. "The Joint Replenishment Problem with Time-Varying Costs and Demands: Efficient, Asymptotic and ϵ-Optimal Solutions." *Operations Research* 42 (6): 1067–1086.

FLEXE. 2019. *The State of On-Demand Warehousing.* Technical Report. FLEXE. https://www.flexe.com/whitepaper/2019-state-on-demand-warehousing.

Forger, Gary. 2018. "Big Picture: On-Demand Warehousing Prepares for TakeOff." *Modern Materials Handling* https://www.mmh.com/article/on_demand_warehousing_prepares_for_take_off.

Furini, F., and X. Shen. 2018. "Matheuristics for the Temporal Bin Packing Problem." In *Recent Developments in Metaheuristics*, edited by Yalaoui F. Amodeo L., Talbi EG., 333–345. Springer.

Furuhata, Masabumi, Maged Dessouky, Fernando Ordó nez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. 2013. "Ridesharing: The State-of-the-Art and Future Directions." *Transportation Research Part B: Methodological* 57: 28–46.

Gale, D., and L. S. Shapley. 1962. "College Admissions and the Stability of Marriage." *The American Mathematical Monthly* 69 (1): 9–15.

Gansterer, Margaretha, and Richard F. Hartl. 2018. "Centralized Bundle Generation in Auction-based Collaborative Transportation." *OR Spectrum* 40 (3): 613–635.

Gansterer, M., and R. F. Hartl. 2020. "Shared Resources in Collaborative Vehicle Routing." *TOP* 28: 1–20.

Gong, Yeming, René B. M. de Koster, J. B. G Frenk, and Adriana F Gabor. 2013. "Increasing the Revenue of Self-storage Warehouses by Facility Design." *Production and Operations Management* 22 (3): 555–570.

Gschwind, T., and S. Irnich. 2017. "Stabilized Column Generation for the Temporal Knapsack Problem Using Dual-Optimal Inequalities." *OR Spectrum* 39: 541–556.

Guillot, Craig. 2018. "Flexible Warehouse Schemes Emerge to Meet Supply Chain Demands." *Supply Chain Dive* https://www.supplychaindive.com/news/pop-up-supply-chains-retail-fulfillment/543049/.

Hezarkhani, Behzad, Marco Slikker, and Tom Van Woensel. 2018. "Collaborative Replenishment in the Presence of Intermediaries." *European Journal of Operational Research* 266 (1): 135–146.

Jorge, Diana, and Gonçalo Correia. 2013. "Carsharing Systems Demand Estimation and Defined Operations: A Literature Review." *European Journal of Transport and Infrastructure Research* 13 (3): 201–220.

Ke, Ginger Y., and James H. Bookbinder. 2018. "Coordinating the Discount Policies for Retailer, Wholesaler, and Less-Than-Truckload Carrier Under Price-Sensitive Demand: A Tri-Level Optimization Approach." *International Journal of Production Economics* 196 (Supplement C): 82–100.

Lindahl, M., M. Sorensen, and Thomas R. Stidsen. 2018. "A Fix-and-Optimize Matheuristic for University Timetabling." *Journal of Heuristics* 24: 645–665.

Mancini, S. 2016. "A Real-Life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search Based Matheuristic." *Transportation Research Part C: Emerging Technologies* 70: 100–112.

Mancini, S., and M Gansterer. 2021. "Vehicle Routing with Private and Shared Delivery Locations." *Computers & Operations Research* 133 (1Article ID 105361.

Mancini, S., M Gansterer, and Richard F. Hartl. 2021. "The Collaborative Consistent Vehicle Routing Problem with Workload Balance." *European Journal of Operational Research* 293 (3): 955–965.

Maniezzo, V., Marco A. Boschetti, and T. Stützle. 2021. *Matheuristics – Algorithms and Implementation*. EURO Advanced Tutorials on Operational Research. Springer.

Marques, A., R. Soares, M. J. Santos, and P. Amorim. 2020. "Integrated Planning of Inbound and Outbound Logistics with a Rich Vehicle Routing Problem with Backhauls." *Omega* 92: 1–18.

Martinovic, J., N. Strasdat, and M. Selch. 2021. "Compact Integer Linear Programming Formulations for the Temporal Bin Packing Problem with Fire-Ups." *Computers & Operations Research* 132.

Meneghetti, A., and S. Ceschia. 2020. "Energy-Efficient Frozen Food Transports: The Refrigerated Routing Problem." *International Journal of Production Research* 58 (14): 4164–4181.

Meneghetti, A., E. Dal Borgo, and L. Monti. 2015. "Rack Shape and Energy Efficient Operations in Automated Storage and Retrieval Systems." *International Journal of Production Research* 53 (23): 7090–7103.

Meneghetti, A., F. Dal Magro, and P. Simeoni. 2018. "Fostering Renewables Into the Cold Chain: How Photovoltaics Affect Design and Performance of Refrigerated Automatedware-houses." *Energies* 11 (5): 1029.

Meneghetti, A., and L. Monti. 2015. "Greening the Food Supply Chain: An Optimisation Model for Sustainable Design of Refrigerated Automated Warehouses." *International Journal of Production Research* 53 (21): 6567–6587.

OneVASTWarehouse. 2021. "https://onevastwarehouse.com/." Visited on March 8, 2021.

Pazour, J. A., and K Unnu. 2018. "On the Unique Features and Benefits of On-Demand Distribution Models." In *15th IMHRC Proceedings (Savannah, Georgia. USA)*, 1–9.

Rogers, Z. S., S. Golara, Y. Abdusalam, and D. S. Rogers. 2020. "A Sharing-Enabled Portfolio Approach to Distribution." *Transportation Journal* 59 (2): 99–128.

Shao, Chaoyi, Hai Yang, Yi Zhang, and Jintao Ke. 2016. "A Simple Reservation and Allocation Model of Shared Parking Lots." *Transportation Research Part C: Emerging Technologies* 71: 303–312.

Shi, Y., Y. Yu, and Y. Dong. 2021. "Warehousing Platform's Revenue Management: A Dynamic Model of Coordinating Space Allocation for Self-use and Rent." *European Journal of Operational Research* 293: 167–176.

Tornese, F., K Unnu, M. G. Gnoni, and J. A. Pazour. 2020. "On-demand warehousing: main features and business models." In *XXV Summer School "Francesco Turco" – Industrial Systems Engineering*, 1–23.

Unnu, K., and J. A. Pazour. 2019. "Analyzing Varying Cost structures of Alternative Warehouse Strategies." In *IISE Annual Conference and Expo 2019*, 480–485. Orlando, Florida.

Unnu, Kaan, and Jennifer Pazour. 2022. "Evaluating On-Demand Warehousing Via Dynamic Facility Location Models." *IISE Transactions* 1–16. doi:10.1080/24725854.2021.2008066.

Zhang, Xiandong, Yeming (Yale) Gong, Shuyu Zhou, René de Koster, and Steef van de Velde. 2016. "Increasing the Revenue of Self-storage Warehouses by Optimizing Order Scheduling." *European Journal of Operational Research* 252 (1): 69–78.

Zhong, Y., Q. Pan, W. Xie, T. C. E. Cheng, and X. Lin. 2020. "Pricing and Wage Strategies for An On-Demand Service Platform with Heterogeneous Congestion-Sensitive Customers." *International Journal of Production Economics* 230.

Zhou, Shuyu, Yeming Gong, and René de Koster. 2016. "Designing Self-storage Warehouses with Customer Choice." *International Journal of Production Research* 54 (10): 3080–3104.