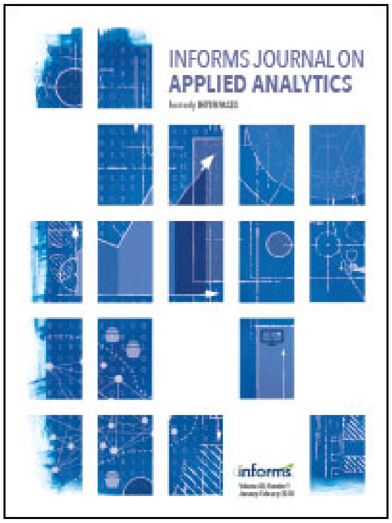


INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

JD.com: Operations Research Algorithms Drive Intelligent Warehouse Robots to Work

Hengle Qin, Jun Xiao, Dongdong Ge, Linwei Xin, Jianjun Gao, Simai He, Haodong Hu, John Gunnar Carlsson

informs
http://pubsonline.informs.org/journal/inte
INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 52, No. 1, January–February 2022, pp. 42–55
ISSN 2644-0865 (print), ISSN 2644-0873 (online)

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

JD.com: Operations Research Algorithms Drive Intelligent Warehouse Robots to Work

Hengle Qin,[a,b] Jun Xiao,[b] Dongdong Ge,[a] Linwei Xin,[c] Jianjun Gao,[a] Simai He,[a] Haodong Hu,[a] John Gunnar Carlsson[d]

[a] Research Institute for Interdisciplinary Sciences, Shanghai University of Finance and Economics, Shanghai 200433, China; [b] Department of AI and Big Data, JD Logistics, JD.com, Beijing 100176, China; [c] Booth School of Business, University of Chicago, Chicago, Illinois 60637; [d] Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, California 90089

Contact: qinhengle@jd.com (HQ); xiaojun@jd.com (JX); ge.dongdong@shufe.edu.cn (DG); linwei.xin@chicagobooth.edu, https://orcid.org/0000-0002-8160-6877 (LX); gao.jianjun@shufe.edu.cn (JG); simaihe@mail.shufe.edu.cn, https://orcid.org/0000-0003-2701-9265 (SH); hu.haodong@shufe.edu.cn (HH); jcarlsso@usc.edu (JGC)

Accepted: July 20, 2021

https://doi.org/10.1287/inte.2021.1100

Abstract. JD.com pioneered same-day delivery as a standard service in China's business-to-consumer e-commerce sector in 2010. To balance the urgent need to meet growing demands while maintaining high-quality logistics services, the company built intelligent warehouses that use analytics to significantly improve warehouse efficiency. The brain of the intelligent warehouse system is the dispatching algorithm for storage rack-moving robots, which makes real-time dispatching decisions among robots, racks, and workstations after solving large-scale integer programs in seconds. The intelligent warehouse technology has helped the company decrease its fulfillment expense ratio to a world-leading level of 6.5%. The construction of intelligent warehouses has led to estimated annual savings of hundreds of millions of dollars. In 2020, JD.com delivered 90% of its first-party-owned retail orders on the same day or on the day after the order was placed. The agility of such intelligent warehouses has allowed JD.com to handle 10 times the normal volume of orders during peak sales seasons and has also helped the company respond quickly to COVID-19 and ensure the rapid recovery of production capabilities.

Funding: D. Ge is partially supported by the National Natural Science Foundation of China [Grants 11831002, 11471205, and 72150001]. J. Gao is partially supported by the National Natural Science Foundation of China [Grants 71971132 and 61573244]. S. He is partially supported by the National Natural Science Foundation of China [Grant 71825003].

Supplemental Material: The electronic companion is available at https://doi.org/10.1287/inte.2021.1100.

Keywords: intelligent warehouse • robotic system • automatic guided vehicle (AGV) • integer program • cutting planes • dispatching • e-commerce • order picking • order fulfillment • Edelman Award

Introduction

In the past decade, China's domestic warehouse area increased from 550 million square meters to 1.12 billion square meters, and the number of employees increased from 511,000 to 1.18 million. Labor costs increased from $3,100 to $13,900 per person per year, an increase of 4.5 times. In addition, the annual express parcel volume skyrocketed from 3.67 billion to 83 billion, a nearly 23-fold increase. Furthermore, China has a nearly 10-fold increase in e-commerce parcel volume during promotion days, such as 618 Grand Promotion (June 18) and Singles Day (November 11), among many other promotional campaigns throughout the year. Both demand and labor costs increase yearly, thus causing challenges for companies to balance the urgent need to meet growing demand while managing costs. Although the boom of e-commerce and the expectations of shorter and shorter delivery times are not unique to the Chinese market and are witnessed globally, these challenges are particularly severe in China because its e-commerce demand accounts for approximately 70% of the total express-market demand. These challenges have made the Chinese logistics industry realize productivity cannot rely primarily on human labor for the future development of business.

In response to these developing needs, companies have recently explored the use of conventional stacker and automated conveyor equipment. Although the efficiency can be improved and the logistics cost can be controlled effectively, the installation of automated equipment is extremely time consuming and the equipment is difficult to maintain. Additionally, the automated equipment lacks flexibility in production, making adapting to the characteristics of the peaks and troughs of the Chinese e-commerce business difficult. Many companies believe flexible and intelligent warehouses equipped with mobile robots represent the future of the industry.

JD.com, also known as Jingdong, has been committed to the use of logistics assets and technology. As a leading Chinese business-to-consumer (B2C) e-commerce company, JD.com dabbled in the field of e-commerce in 2004. Its net revenues and operations income, based on other than generally accepted accounting principles, for 2020 were $114.3 billion and $2.4 billion, respectively. As an online retailer and a marketplace for third-party sellers, JD.com has been distinguishing itself from its competitors by a strong commitment to customer service. Its "211 program," which provides same-day delivery for orders submitted prior to 11 a.m. and next-day delivery (before 3 p.m.) for orders submitted before 11 p.m., set a new standard in China's B2C e-commerce sector. In 2020, approximately 90% of the total online retail orders processed through JD.com's logistic network were delivered on the same day or the day after the order was placed, with over 60% of the total online retail orders covered by the 211 program. JD.com currently serves over 471 million active customers and manages the flow of millions of stock-keeping units (SKUs) through China's largest nationwide fulfillment network.

JD.com is a major innovator in the use of autonomy in logistics. It made a strategic decision to invest in in-house logistics instead of relying on third-party logistics. Starting in 2007, it began to establish its self-operated, nationwide logistics forces. In 2012, it registered its own logistics company. JD.com also set logistics automation as a strategic goal. On April 25, 2017, JD Logistics was officially established (JD.com 2017b). This new business unit focuses on innovating the next generation of intelligent logistics technologies, with a particular emphasis on automated fulfillment capabilities, including automated warehouse technologies, autonomous delivery vehicles, and drones (Huang 2017, Lin and Singer 2017). With the mission of reducing logistics costs for the entire Chinese logistics industry, JD Logistics is committed to sharing with society the infrastructure, management experience, and professional technology it has accumulated and becoming a global supply chain infrastructure service provider. At present, JD Logistics has six comprehensive logistics networks covering small- to medium-sized items, large-sized items, cold chain, B2B, cross-border, and crowdsourcing. With the global coverage of these six large networks and the application of big data, analytics, cloud computing, and intelligent equipment, JD Logistics encompasses an intelligent supply chain service system, including warehousing, transportation, and distribution. With control over fulfillment and delivery, JD.com has been able to offer same-day or next-day deliveries on most orders, a competitive strategy that had won customers over the years.

The company is aware of the importance of operations research and advanced analytics to manage warehouses. Research scientists from JD Logistics and several universities have conducted joint research to support various operations in its intelligent warehouses, one of its major recent innovations. JD.com's intelligent warehouse technology has significantly boosted warehouse efficiency and reduced operational costs. Robots were introduced into its production process in 2016, and intelligent warehouses have become common within the company. A highlight of JD.com's intelligent warehouse network is the development in 2017 of a fully automated warehouse located at its Asia No. 1 logistics center (Asia No. 1 refers to JD.com's large fulfillment centers, which are used for both stocking and e-commerce order fulfillment) in Shanghai's Jiading District, which covers a combined floor area of more than 400,000 square feet and stores 60,000 boxes of goods (China Daily 2017). All operations in the warehouse are fully automated, including receiving goods, storage, packaging, and sorting. It is equipped with nearly 1,000 robots with varying functions, including three types of six-axis mechanical arms used for storage and packaging and three types of automatic guided vehicles (AGVs) for sorting and order picking. They can sort 3,600 items per hour and the warehouse can process over 200,000 orders a day. We refer interested readers to JD.com (2017a) for a video about this warehouse.

Problem Description

One of the major challenges in operating an intelligent warehouse is the management and dispatching of AGVs. In this section, we provide a high-level overview of this problem. In a conventional picker-to-parts warehouse system, workers walk or drive along the storage aisles, pick up requested items from storage racks, and return to accumulate, sort, and pack the orders. As distinguished from the conventional picker-to-parts warehouse system, in a modern parts-to-picker intelligent warehouse system, storage racks of inventoried items are transported to workstations where the picking occurs. In an intelligent warehouse, SKUs are initially stored on storage racks in the picking area. A large number of AGVs, which pick up racks and transport them in the warehouse, surround the storage racks. Figure 1(a) shows a typical rack that an AGV (the small disk-shaped robot at the bottom) carries. Figure 1(b) shows a corner of the warehouse. Approximately 70%–80% of these racks have two sides to store

Figure 1. (Color online) (a) Storage Rack and an AGV Carrying It and (b) a Small Section of the Warehouse

products; the remainder of the racks have only one side (usually for oversized products). Each side of a rack has 10–20 storage grids, and each grid stores only one type of SKU. A typical warehouse handles hundreds of SKUs. The distribution of SKUs is intentionally designed such that one type of SKU can be placed on many different storage racks.

Workstations that receive the orders and pick the SKUs from the rack to fulfill the orders are on the edge of the picking area. Each workstation contains several berths for the AGVs. The orders are periodically sent to the workstations. A typical order consists of various types of SKUs. After receiving an order, the dispatch center examines the types and the number of SKUs in that order and sends commands to one or multiple AGVs to pick up and transport racks to the workstation. The candidate racks include the ones in the picking area and the ones already being carried by AGVs. Because the number of candidate racks is typically large, the dispatch center must determine how to make optimal dispatching decisions.

If a rack has been assigned to a workstation, it needs to be transported by an AGV. In particular, if a rack is currently not moving, an idle AGV will be assigned to pick it up and deliver it to the target workstation. When an AGV carrying a rack arrives at the workstation, it will park in a berth near the workstation. Based on the order received, either a human picker or robot arm of the workstation will select the required SKUs from the racks. Depending on the type of the warehouse, this process can be either fully automated or semiautomated, in which case the rack storage and retrieval operations are automated, whereas the picking process is still manual. The majority of JD.com's intelligent warehouses are semiautomated. Once the desired products have been picked from the rack, the AGV will carry the rack back to the picking area. If the workstation has no empty berth at the moment, no rack is assigned to this workstation. Furthermore, because of the technical difficulties, when a double-sided rack arrives at a workstation, the robot arm can select only the products from the chosen side. Therefore, when assigning a rack to a workstation, the central dispatching system must also decide which side of the rack is assigned.

The previous problem description applies to warehouses that stock small SKUs, where multiple items are usually consolidated into one package before shipping. The operation is different in a warehouse that holds medium-sized SKUs. These items can be picked by robotic arms, and one package usually includes only one item (i.e., multiple-item orders are split and shipped separately). Finally, we emphasize that the previous description applies to only the dispatching aspect of the warehouse operating system. In this paper, we do not discuss other important and challenging problems, such as which racks to stow received inventory, how to allocate orders to workstations, how to balance the workload among disjoint zones, where to store a paired AGV and rack when they leave a workstation, and how to route AGVs without collision.

Generally speaking, compared with the picker-to-parts system, the parts-to-picker system can boost the picker productivity (Wurman et al. 2008, Wulfraat 2012). The parts-to-picker system also provides flexibility in adding and removing robots, which is particularly important in an e-commerce environment with volatile demand. This revolutionary system was first patented in the United States by Kiva Systems Inc. (Mountz et al. 2008), which was later acquired by Amazon. Increasingly more providers are competing in this growing market (Banker 2016). The parts-to-picker system is also referred to as the robotic mobile fulfillment

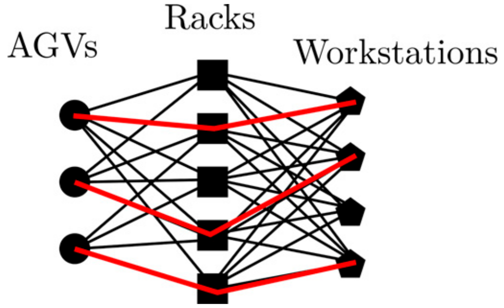
(RMF) system in the academic literature. The performance of such RMF systems has had little scientific study (Azadeh et al. 2019). Although recent investigations, including Boysen et al. (2017), Yuan et al. (2018), Yuan et al. (2019), Cezik et al. (2021), and Wang et al. (2021), have been conducted on related operations planning and control problems, the real-time dispatching problem in a large-scale intelligent warehouse, as we describe in this paper, has not been documented in the academic literature. We refer the interested readers to a comprehensive review of robotized and automated warehouse systems (Azadeh et al. 2019). We also emphasize that simply installing Kiva-type robots does not necessarily lead to the desired benefits, and operations research concepts and analytics play a key role in achieving the benefits.

Framing AGV Dispatching as a Matching Assignment Problem

To run an efficient intelligent warehouse-management system, we aim to effectively dispatch AGVs to racks and workstations and to minimize the operating cost and order-fulfillment time. To achieve this goal, we build and solve a complex online tripartite matching problem. The three components of the flow network are (1) AGVs, (2) racks, and (3) workstations (also including those for restocking and charging). At a high level, AGVs are assigned to racks and workstations by using a tripartite network flow paradigm (Figure 2). Rather than explicitly minimizing order-fulfillment time, in our model, we minimize the volume of unsatisfied demand as a surrogate for minimizing order-fulfillment time. Given the demand for each type of SKU at any time, the warehouse management system makes decisions such as to execute the following AGV actions:

- An AGV picks up a rack within the picking area and brings it to a workstation, where some requested SKUs are retrieved.
- An AGV picks up a partially empty rack within the picking area and brings it to a restocking station, where it is stocked with additional SKUs.

Figure 2. (Color online) Tripartite Network Flow Problem Is to Assign AGVs to Racks and then to Workstations



- An AGV retrieves a partially empty rack from a workstation and returns it to the picking area.
- An AGV retrieves a partially empty rack from a workstation and brings it to a restocking station.
- An idle AGV travels to a charging station, where its battery is recharged.

We focus only on the first bullet (i.e., an AGV picks up a rack within the picking area and brings it to a workstation) to emphasize the key concepts; however, the solution procedure, as we describe, generalizes to the other cases. We model the overall problem as a bicriteria optimization problem, where the goal is to minimize the total travel distance of the AGVs, subject to the constraint that as many orders as possible must be met in a timely fashion. We face additional constraints that complicate the problem beyond a mere minimum-cost flow problem, because the same SKU type may be located in many different racks. For example, if we require 100 units of a specific SKU type, we must decide whether to use two AGVs to collect two nearby racks with 50 units each or use one AGV to collect a single rack that contains 100 units but is farther away. These constraints essentially correspond to set-covering style constraints that couple the decision variables in a way that makes the constraint matrix lose total unimodularity.

The matching problem we describe here must be solved in each operating period, because the numbers of idle AGVs, racks in the picking area, available berths, and inventory of each rack (side) change over time. In our system, one operating period is less than five seconds. Thus, the matching problem must be solved within three seconds. Given the problem size, the major challenge is the constraint on the computational time. As such, we focus only on a static problem for practical reasons and do not consider a more complex dynamic version of the problem incorporating demand forecasting.

We solve this problem by separating it into two subproblems by introducing Lagrange multipliers. After the decomposition, one subproblem is a much smaller integer program. We relax its integrality constraints to create a linear programming model that we enhance with cutting planes to improve the solution quality. The resulting solution is rounded to the nearest integer values. We also adopt a preprocessing procedure to reduce the subproblem size, for example, by eliminating from consideration the possibility of an AGV traveling a considerable distance to a rack side unless its inventory can completely satisfy the required quantity of a workstation's SKU. The other subproblem is an unbalanced assignment problem and is equivalent to a linear program, which we solve via the Hungarian algorithm. Please see the appendix for further details of the model formulation and solution methodology.

It is plausible to attack this problem by first determining the matching between racks and workstations and then determining the matching between AGVs and racks, that is, by decoupling the problem. Although this approach results in an algorithm that runs much faster, the quality of the solution is not guaranteed. For example, when ignoring the locations of AGVs, this algorithm could match workstations with racks that are further away from AGVs.

Integration of Algorithms with Management Systems

The architecture of the intelligent warehouse system contains four layers—the integrated management layer, warehouse management layer, intelligent dispatching layer, and device control layer—which we discuss briefly below. The integration management layer interfaces with various external business systems (e.g., enterprise resource planning and the warehouse management system) and decouples the intelligent warehouse system from the external systems. It is mainly responsible for receiving business requests from the external systems and converting the messages into standard formats. The warehouse management layer includes several functional modules closely related to warehouse management, such as inventory center, order center, and location center. It receives messages in standard formats from the integrated management layer and implements various warehousing operations, such as inbound/outbound activities and inventory checking. The intelligent dispatching layer is the brain of the intelligent warehouse system. It uses different algorithms based on different purposes. For example, when receiving a customer order, it determines how to optimally fulfill the order based on the information in the order (e.g., merchandise, shelf inventory distribution, workstations). When the intelligent dispatching layer determines which resources to use to complete a task, it sends commands to the device control layer, which then dispatches AGVs to complete the task. The device control layer is mainly responsible for coordinating and dispatching different devices to complete tasks together. It includes two subsystems: the task center and the AGV console. The device control layer communicates with a single AGV through the AGV console to control the action of a single AGV. The AGV receives a task issued through the controller and converts it into robot instructions to drive the internal executors (e.g., servo motors) to complete the task. The design ensures that the replacement and upgrade of devices have the least impact on the system to ensure that the system scales well and responds quickly to demands.

When designing an algorithm, it is important to consider its compatibility with the existing hardware. For example, our dispatching model does not consider the possibility of reassigning an occupied AGV although doing so may lead to an improved solution because of the following practical reason. Frequent rewriting of tasks involves more input/output (I/O) communication, which increases the chances of system errors. AGVs operate in a wireless communication environment and must maintain high-frequency I/O communication with the central dispatching system at all times. Frequent rewriting of tasks amplifies the amount of communication, thereby increasing the probability of communication errors and resulting in fatal order fulfillment and inventory errors. Such errors are undesirable, especially in an e-commerce environment that requires a high picking accuracy. As such, the project team ruled out this design for system-stability reasons.

Managerial, Political, and Financial Implementation Challenges

Rome was not built in one day. Having conducted research to determine the best solution, the project team's most important task was to find a warehouse to verify the efficiency of the solution. In the real world, JD.com's warehouses are concentrated in seven regions: north, east, south, southwest, central, northeast, and northwest. To deploy a new warehouse, the company's central warehouse planning department had to first communicate with the seven operation centers to confirm the new warehouse requirements and determine a target warehouse. Then, the team proposed a plan for the joint review of the warehouse planning department and the region. If approved, the team went to the site to deploy and implement the plan. Two major challenges arose: (1) Could the team convince the warehouse planning department to approve this new warehouse model? (2) Was any region willing to carry out the pilot study and build the very first model?

The project team solved the first problem by persuading the warehouse planning department to conduct a simulation comparison. The team modeled the layout of a conventional warehouse under the jurisdiction of the North China Operations Center in a computer-simulated environment. At the same time, the team also built an intelligent warehouse solution under the same conditions in the computer-simulated environment. Then, it compared the two approaches based on one month of real data from the conventional warehouse, including daily inventory and the numbers of online orders and pickers. The results showed the intelligent warehouse saved a considerable amount of human labor and also met the requirement for timeliness in meeting order deadlines. In addition, based on the results of a stress test, the peak order volume processed by the intelligent warehouse was much higher than the volume in the conventional warehouse. Moreover, based on calculating the return on investment (ROI), the team concluded that the intelligent warehouse would be able to recover the investment cost within five years provided that the daily order volume reached a certain threshold. Therefore, the warehouse planning department approved the project team's attempt to promote the intelligent warehouse model and contacted the manager responsible for the East China Operations Center to deploy the first intelligent warehouse.

Within JD.com the regional operations centers are critical decision makers; thus, building the first intelligent warehouse required their support. To solve the second problem, the team not only demonstrated the benefits of the intelligent warehouse model by simulations, but also discovered the true concern of the East China Operation Center management through frequent communication: a large number of robots deployed in this model would put great pressure on the depreciation cost of the region. To address this concern, the project team made a proposal: if the region would agree to deploy an intelligent warehouse, the depreciation costs of the equipment would be linked to the project team and not be associated with the region for the first two years. After resolving this issue, the East China Operations Center agreed to deploy the first intelligent warehouse.

In addition to building a new intelligent warehouse from scratch, converting existing conventional warehouses is an important way to deploy intelligent warehouses. The first intelligent warehouse converted from an existing, conventional one was in Beijing under the jurisdiction of the North China Operations Center. It was a warehouse for computer, consumer electronics, and communication (3C) products. Additionally, the conversion process of an existing warehouse differed from the deployment process of a new warehouse. For a new warehouse, the project team needed only to polish the ground so that it was smooth enough for robots to run and then complete routine steps, such as labeling, shelf placement, workstation installation, robot entry, and inventory allocation to shelves. However, for a conversion, before polishing the ground, additional steps were required, for example, shutting down the warehouse operation, temporarily relocating inventory to a nearby warehouse, and removing old equipment. An emerging challenge to converting a conventional warehouse to an intelligent one was the assignment (or reassignment) of the warehouse employees because the intelligent warehouse model would need fewer operating personnel. The agreement reached between the project team and managers of the operation center was to retrain existing employees to become local equipment maintainers and thus improve their skills and eventually improve their productivity. Layoffs were not an option. This solution not only allowed most employees to keep their jobs but also solved the problem of providing a local maintenance team for the equipment. Because of the nature of converting the conventional warehouses, the team was confronted with the challenge of converting an existing warehouse into an intelligent one without compromising normal warehouse operations and the stability of the employees. Fortunately, the team meticulously thought out its plan, addressed various concerns from different perspectives, implemented the plan, and eventually completed the conversion of this conventional warehouse. We refer interested readers to CCTV-4 (2017) for a video showing media coverage of this warehouse after its conversion; in particular, see the section between 0:50 and 4:47 of the video.

JD.com had built more than 20 intelligent warehouses by the end of 2017; however, the deployments did not always go smoothly. According to a summary report at the end of that year, the processing efficiency of some intelligent warehouses did not meet the planned targets and the average daily order volume of intelligent warehouses did not reach the threshold for recovering investment returns. The expansion was temporally suspended because of the concerns about the operating efficiency of intelligent warehouses, and the team was urged to improve the efficiency of existing intelligent warehouses. After the analysis, the team found that the inefficiency was due to (1) insufficient inventory (e.g., some warehouses did not stock enough inventory because the warehouse managers were skeptical about the new technology and therefore too conservative) and (2) the lack of optimization on many details of the operation (e.g., to pick items on upper levels of racks, raising the height of the workstation is preferable to using ladders). The team then spent an entire quarter on adjusting and improving the processes. At the company's first quarterly meeting in 2018, the data suggested that the overall processing efficiency had reached the planned level. Management once again affirmed the role of the intelligent warehouse model for the company's long-term development and restarted the project. Subsequently, the intelligent warehouse project progressed faster than ever.

With the ongoing work of the project team, the maturity of the method and deployment efficiency have gradually improved. Today the team can build a new intelligent warehouse within one month and convert an existing warehouse within two months, thus greatly improving the deployment and efficiency of intelligent warehouses within JD.com. Furthermore, every intelligent warehouse must undergo a rigorous ROI analysis before implementation. A key criterion for approving a

Table 1. Relevant Statistics Gathered During the Nanxiaoying Stress Test

Panel A

	Orders	Items	SKU types	Workstations	Workers	AGVs	Area (sq. ft)	Time (h)
Automated	24,268	56,283	424	8	38	64	30,139	16
Conventional	49,101	109,962	3,001	38	98	NA	86,111	16

Panel B

	Orders/hour/workstation	Items/hour/workstation	Items/hour/worker	Items/sq. ft
Automated	190	440	93	1.87
Conventional	81	181	70	1.28
Ratio	2.3:1	2.4:1	1.3:1	1.5:1

deployment plan is whether building an intelligent warehouse will recover the investment cost within five years. In the calculation process, JD.com first determines the cost of fulfilling a single order under the conventional warehouse operation mode and then calculates the corresponding cost under the intelligent warehouse operation mode. If the five-year cost saving, that is, the cost difference per order multiplied by the estimated orders handled by the warehouse over the ensuing five years, is greater than the cost of investment (including construction, equipment, labor, management, energy consumption, and packaging), it will be considered as a good investment. Otherwise, the plan will not be implemented.

Finally, the team also encountered a challenge caused by the cash-flow constraint when rapidly promoting the intelligent warehouse technology. To resolve this challenge, when the technology was mature enough, the intelligent warehouse model was commercialized and made available for sale to external clients. Through revenues from external sources, the team was able to self-finance the project and raise startup capital for the deployment of intelligent warehouses and effectively achieve strong internal and external circulations. Internal circulation refers to the expanded use of intelligent warehouses within JD.com and the use of JD Logistics scenarios to further advance the technology. The continuous accumulation of practice and experience in the internal circulation enables JD.com to be a trusted partner for its external clients. The external circulation refers to the commercialization of JD.com's intelligent warehouse technology. The revenues generated by the commercialization can further accelerate the expansion of intelligent warehouses in internal circulation. The driving effect of the two-part circulation promotes the continuous and robust development of the intelligent warehouse solution.

Stress Tests

To conduct an apples-to-apples comparison, JD.com conducted two real-life stress tests to compare an automated warehouse with a conventional warehouse that relies on human labor. Both tests took place on November 11, 2017, one in Nanxiaoying and one in Gu'an (both of which are located near Beijing). Note that November 11 is China's most important annual promotion day and also the world's largest online shopping festival. The results of the two tests are summarized in Tables 1 and 2, suggesting that the improvement automation provided is dramatic by all relevant metrics. Specifically, in the Nanxiaoying test, the automated warehouse was 30,139 square feet and processed 24,268 orders (including 56,283 items and 424 SKU types) within 16 hours by employing eight workstations, 38 workers (including workers other than pickers), and 64 AGVs. By contrast, the conventional warehouse was 86,111 square feet and processed 49,101 orders (including 109,962 items and 3,001

Table 2. Relevant Statistics Gathered During the Gu'an Stress Test

Panel A

	Orders	Items	SKU types	Workstations	Workers	AGVs	Area (sq. ft)	Time (hr)
Automated	8,918	26,086	198	3	20	26	21,528	20
Conventional	18,705	26,950	6,415	18	72	NA	161,459	20

Panel B

	Orders/hour/workstation	Items/hour/workstation	Items/hour/worker	Items/sq. ft
Automated	149	435	65	1.21
Conventional	52	75	19	0.17
Ratio	2.9:1	5.8:1	3.4:1	7.1:1

SKU types) within 16 hours by employing 38 workstations and 98 workers. The automated warehouses stocked only fast-moving items, whereas the conventional warehouses also stocked slow-moving items. Their ratios of the number of processed orders per hour per workstation, the number of processed items per hour per workstation, the number of processed items per hour per worker, and the number of processed items per square feet were 2.3, 2.4, 1.3, and 1.5 times, respectively. The improvement was even more significant in the Gu'an test for many reasons, including number of SKU types, size and layout of the warehouse, and workers' proficiency. Finally, compared with the claim in the literature (Wurman et al. 2008, Wulfraat 2012) that pick rates of 600 items per hour per workstation are achievable under the assumption that the Kiva system can deliver a new rack face every six seconds by keeping a small queue of work at the picking station, our pick rates were only 440 and 435, respectively, during the stress tests. However, we found that the rates depended heavily on workers' proficiency and could exceed 600 as workers became more experienced with the new system.

As a result of the success of the stress tests, JD.com subsequently built a warehouse that can handle 200,000 orders a day but employs only four people, whose jobs are centered around servicing the AGVs (Palmer 2018). Optimization combined with modern robotics technology resulted in tremendous improvements in picking efficiency and warehouse space utilization.

Estimated Impact

As part of the company's strategic plan, JD.com has been aggressively expanding its warehouse network. We refer the reader to Table 3 for a summary of the growth in warehouses and warehouse space experienced since the company went public on Nasdaq in 2014. In particular, it is currently operating over 900 warehouses in which it largely applies intelligent warehouse technology. Some of the intelligent warehouses can process more than 1.3 million orders per day during peak seasons. JD.com's application of intelligent technologies is the most extensive in the field of e-commerce logistics in China. Armed with advanced operations research concepts and analytics, its

Table 3. Growth of JD.com's Warehouse Network Since 2014, Including the Number of Warehouses (Both Conventional and Intelligent) and the Warehouse Gross Floor Areas (Million Square Meters)

	2014	2015	2016	2017	2018	2019	2020
Number of warehouses	123	213	256	486	550	700	900
Gross floor area	2.2	4.0	5.6	10.0	12.0	16.9	21.0

intelligent warehouse technology has significantly improved warehouse efficiency. It also has helped the company lower its number of inventory turnover days to 33.3 and to decrease its fulfillment expense ratio from 7.2% in 2016 to a world-leading level of 6.5%. The construction of intelligent warehouses has led to an estimated hundreds of millions of dollars in annual savings. In 2020, 90% of JD.com's first-party retail orders were delivered on the same day or on the day after they were placed, raising the industry's bar for customer satisfaction.

The agility of such intelligent warehouses not only enables JD.com to handle 10 times the normal volume of orders during peak sales seasons, such as 618 Grand Promotion (June 18) and Singles Day (November 11), but also helped the company respond quickly to the many complexities associated with the COVID-19 pandemic and ensure the rapid recovery of production capability. The sudden outbreak of COVID-19 swept through the world in 2020, and crowds were perfect hotbeds for the virus to spread. When conventional warehouses were shut down to prevent the spread of the virus, JD.com's intelligent warehouse played a key role in this battle against COVID-19. Intelligent warehouses effectively prevented warehouse workers from gathering, functioned normally and efficiently in virus-ridden areas, and ensured supplies were distributed to their destinations in a timely manner.

As the company that pioneered same-day delivery as a standard service in China's B2C e-commerce sector 10 years ago, JD.com continues to pursue balancing the urgent need to alleviate pressure on logistics workers while maintaining the high-quality logistics services that define the company's brand. Before the deployment of intelligent warehouses, logistics workers across the industry faced unprecedented pressure in the form of physical labor. For example, warehouse employees often managed heavy loads, causing serious joint problems; pickers walked the length of almost a marathon each day; and workers in cold chain warehouses transitioned frequently between normal temperatures and temperatures as low as −30°C. JD.com's intelligent warehouse technology team has been exploring ways to address these issues and improve working conditions.

As part of its Retail-as-a-Service strategy, in addition to improving its own operations, JD.com became a retail infrastructure service provider and began providing the integrated technology services of both software and automation solutions to its logistics peers and a wide range of industries in 2018. It has built a brand image for its intelligent warehouse technology in over a hundred opening-up programs (i.e., programs to commercialize the technology for external clients) across industries, including 3C, apparel, industrial products, education, fast-moving consumer goods, retail, automotive, fresh produce, and manufacturing, facilitating the optimization of warehousing and replenishment. For example, Company L, the 3C electronics industry manufacturing giant, anticipates that JD.com's intelligent warehouse solutions can help create without interruption a downstream production line from an upstream raw material storage warehouse. Company L has a production line on the first floor of a production site and has a raw material storage warehouse above it. According to the production line schedule and economies of scale, the raw materials on the second floor must be supplied periodically to the first floor. JD.com designed a solution by using AGVs via a hoist to replenish raw materials from the second floor to the first floor for manual picking. At present, JD.com exports thousands of AGVs annually through its intelligent warehouse solutions.

JD.com's intelligent warehouse technology has received intense media coverage since its deployment, including a highly publicized featured appearance on the April 21, 2018, broadcast on Xinwen Lianbo, China Central Television's nightly news program (CCTV 2018). This program is shown simultaneously by all local TV stations in mainland China, making it one of the world's most-watched programs. Furthermore, as a representative of contemporary Chinese science and technology showing the world the achievements of China's development in the new era, JD.com was also featured during the closing ceremony of the PyeongChang 2018 Winter Olympics in an eight-minute promotional video for the 2022 Winter Olympics, which will be held in Beijing; see JD.com (2018) for a summary of the coverage.

With the application of intelligent technologies, China's ratio of social logistics cost to GDP has steadily decreased from 18% to 14.2% in the past decade. JD.com's management believes the entire industry will see benefits long into the future, and logistics automation will lead in global efficient circulation and sustainable development.

Milestones

We highlight the following milestones for JD.com's intelligent warehouse technology.

- 2010: JD.com began to develop intelligent storage technology, including the application of operations research and analytics in storage.
- 2015: In June, 14 Asia No. 1 warehouses were completed, marking the success of the first stage of self-developed intelligent storage technology. In October, JD.com began to invest in the research and development of next-generation warehousing technology: intelligent warehouse technology mainly involving robotics and data-driven decision making.
- 2016: In June, an engineering team started to work on an AGV control model and algorithm in intelligent warehouses. The engineering team included 10 algorithm engineers and 50 development engineers. In December, the first intelligent warehouse project was completed.
- 2017: In October, JD.com completed construction of a flagship intelligent warehouse located at the Asia No. 1 logistics center in Shanghai's Jiading District, achieving fully unattended operations in the processes of receiving, storing, order picking, packaging, and sorting. Subsequently, this warehouse was able to handle more than 200,000 orders a day.

At present, JD.com's intelligent warehouse team has been expanded to hundreds of engineers, including warehouse design engineers, algorithm engineers, software development engineers, hardware engineers, and implementation engineers. They have completed the research and development of six types of storage robots and three core intelligent storage systems.

Lessons for Other Organizations

For organizations wishing to implement intelligent warehouse solutions, JD.com's practice provides several lessons. First, 100% automation is impractical and not ideal and JD.com does not aim for full automation. Compared with manual picking, the types of commodities that can be processed by intelligent warehouses are relatively limited, and restrictions on intelligent warehouse locations and layout are in place. As a result, the processing efficiency of intelligent warehouses has an upper limit. By contrast, manual picking is more flexible in terms of efficiency and responsiveness to complex scenarios. Ideally, the proportion of intelligent and conventional warehouses should be adjusted dynamically to allow companies to cope better with situations such as large e-commerce promotions and the handling of parts with special shapes. Second, in the process of upgrading conventional warehouses to intelligent warehouses, planning ahead and continuing to employ existing workers (e.g., retraining them to perform emerging jobs) is critical. Third, when dealing with optimization and decision-making problems, relying completely on complex models and sophisticated algorithms is not ideal. Sometimes in-depth business analysis and simple strategies can greatly simplify the complexity of the problem, and excellent outcomes can be achieved without particularly complex models and algorithms. Fourth, a gap always exists between a theoretically ideal solution and reality. A perfect solution in the model may not work in reality simply because of unexpected problems, and repeatedly testing and then making adjustments is necessary. No perfect plan exists—only continuous improvement of the implemented model.

Future Focus

Current intelligent warehouse solutions only solve the problem of efficient picking and fulfillment operations and not the problem of high-density storage. Paying more attention to the balance between high-density storage and efficient picking/fulfillment operations is necessary. Furthermore, investing in technologies other than warehousing, such as drones and driverless cars, is important. Finally, the long-term social impact of the intelligent warehouse technology must be considered and evaluated.

Acknowledgments

The authors thank Sudharshana Apte and Gul Ege for constructive comments and feedback when verifying their submission for the 2021 Franz Edelman Award competition. They are grateful to Goutam Dutta and Gul Ege for voluntarily coaching their team when preparing for the final presentation. The opinions expressed in this paper are those of the authors and do not necessarily represent the views of JD.com.

Appendix. Model Formulation and Solution Methodology

We explain our approach in more detail in this appendix. As we explain previously, our problem can be modeled as an online tripartite network flow problem with additional constraints involving demand for SKUs. Below, we describe our integer programming formulation and solution method. In each operating period, we solve an online tripartite matching problem for AGVs, racks, and workstations. Note that the model/algorithm we describe here is the basic version developed when the team first explored this problem. The currently implemented model/algorithm is a variant of the basic version.

Model Parameters

- Let $\mathcal{I} = \mathcal{I}_a \cup \mathcal{I}_b$ be the index set of all AGVs in the picking area, where $\mathcal{I}_a$ denotes the set of idle AGVs and $\mathcal{I}_b$ denotes the set of occupied AGVs. We use the letter i to indicate the index of the AGVs.
- Let $\mathcal{J} = \mathcal{J}_a \cup \mathcal{J}_b$ be the index set of every side of the racks in the picking area. Here, $\mathcal{J}_a$ denotes the set of all rack sides on the stationary racks and $\mathcal{J}_b$ denotes the set of all sides on the moving racks (i.e., the racks that are currently carried by AGVs). We use the letter j to indicate the side of the rack.
- Let $\mathcal{T}$ be the index set of all the racks in the picking area. We use t to denote the index of a rack. We know the mapping between the sides and the racks; that is, given the index of the rack, we know whether it is double sided or single sided. In addition, given the index of one particular side, we know the index of the rack to which this side belongs.
- Let $\mathcal{K}$ be the index set of all workstations. We use the letter k to indicate the workstation.
- Let $\mathcal{S}$ be the index set of SKU types. We use the letter s to indicate the product type.
- Let $\{c^1_{ij}\}_{i\in\mathcal{I},j\in\mathcal{J}}$ and $\{c^2_{jk}\}_{j\in\mathcal{J},k\in\mathcal{K}}$ be the travel-distance cost matrices. More specifically, c^1_{ij} is the travel distance between the current location of AGV $i \in \mathcal{I}$ and the rack side $j \in \mathcal{J}$, whereas c^2_{jk} is the travel distance between the rack side $j \in \mathcal{J}$ and the workstation $k \in \mathcal{K}$. We note previously that robot arms can select SKUs from only one side of the rack. When AGV i is carrying rack t and rack side j_1 on rack t is the one ready for selecting, we set $c^1_{ij_1} = 0$. However, because of technical difficulties, a distance cost will arise if an AGV wants to turn the rack around. Therefore, if the rack on AGV i is double sided and the other side is j_2, we have $c^1_{ij_2} > 0$.
- Let $\{O_{ks}\}_{k\in\mathcal{K},s\in\mathcal{S}}$ be the requirement (i.e., demand) matrix of SKUs from workstations. The element O_{ks} denotes the number of the type $s \in \mathcal{S}$ of SKUs required by the workstation $k \in \mathcal{K}$. Here, we treat O_{ks} as an input of our matching problem. Allocating orders/items to workstations is an important and challenging problem; however, for brevity, we omit the discussion.
- Let $\{q_{js}\}_{j\in\mathcal{J},s\in\mathcal{S}}$ be the inventory matrix of SKUs on the sides of racks. The element q_{js} denotes the number of the type $s \in \mathcal{S}$ of SKUs on the side $j \in \mathcal{J}$.
- Let $\{B_k\}_{k\in\mathcal{K}}$ be the vector of numbers of available berths for all the workstations; that is, no more than B_k racks can be parked at the k-th workstation. Note that in our matching problem, we only consider workstations with positive B_k; that is, if $B_k = 0$ for some k, the associated workstation will be removed from $\mathcal{K}$ in this round of matching.
- Let α_1, α_2, and α_3 be weighting parameters: α_1 represents the weight assigned to the total cost of moving AGVs to racks; α_2 represents the weight assigned to the total cost of moving racks to workstations; and α_3 represents the weight assigned to the total amount of unsatisfied demands. Here, α_1 and α_2 can be different, because it costs more energy when AGVs are moving with racks than without racks.

We also have the following information:

- For each rack $t \in \mathcal{T}$, we have a subset $\mathcal{J}_t \subset \mathcal{J}$ that consists of the index (indices) of the rack side(s) on rack t. For one-sided rack t, $|\mathcal{J}_t| = 1$. For double-sided rack t, $|\mathcal{J}_t| = 2$, we have $|\mathcal{J}| \le 2|\mathcal{T}|$.
- For each AGV $i \in \mathcal{I}_b$ that is currently transporting a rack, we know the index t_i of the rack. $\mathcal{J}_{t_i}$ is then the set of indices (index) of the sides on this rack.

Decision Variables

To find the best tripartite matching between the AGVs, sides (racks), and workstations in our problem, we introduce decision variables for a matching between the AGVs and sides (rack), a matching between the sides and workstations, and the number of SKUs representing the unsatisfied demands. These definitions are as follows:

- Let $X := \{x_{ij} \in \{0,1\}\}_{i\in\mathcal{I},j\in\mathcal{J}}$ be the matching variables between the AGVs and sides. It is $x_{ij} = 1$ if the AGV $i \in \mathcal{I}$ is assigned to the side $j \in \mathcal{J}$ and $x_{ij} = 0$ otherwise.
- Let $Y := \{y_{jk} \in \{0,1\}\}_{j\in\mathcal{J},k\in\mathcal{K}}$ be the matching variables between the sides and workstations. It is $y_{jk} = 1$ if the side $j \in \mathcal{J}$ is assigned to the workstation $k \in \mathcal{K}$ and $y_{jk} = 0$ otherwise.
- Let $Z := \{z_{ks} \in \mathbb{Z}_+\}_{k\in\mathcal{K},s\in\mathcal{S}}$ be the variables representing the unsatisfied demand required for the product type $s \in \mathcal{S}$ at the workstation $k \in \mathcal{K}$.

The total number of variables equals $|\mathcal{I}| \times |\mathcal{J}| + |\mathcal{J}| \times |\mathcal{K}| + |\mathcal{K}| \times |\mathcal{S}|$.

Objective Function

Our target is to minimize the total travel-distance cost while fulfilling the order requirements as much as possible. Minimizing the travel distance is a common objective in the literature, because order picking is generally considered to be the most critical function and typically accounts for the majority of warehouse operating costs (Eisenstein 2008, Bartholdi and Hackman 2016). The total travel-distance cost can be divided into two parts: the distance cost of AGVs picking up racks and the distance cost of the selected racks being transported to workstations. We do not explicitly model the travel-distance cost from workstations to the rack-storage area upon the completion of picking tasks, because the storage area is divided into zones and racks are assigned to the zones based on their velocity (e.g., fast-moving and slow-moving zones) and racks are randomly stored in their zones upon the completion of tasks. How well the order requirements are met is measured by the total number of the unsatisfied pieces of demand. Thus, we have the following objective function:

$$\mathcal{P} \quad \min \ \alpha_1 \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c^1_{ij} x_{ij} + \alpha_2 \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} c^2_{jk} y_{jk} + \alpha_3 \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}} z_{ks}.$$

Constraints

Our warehouse system has the following constraints:

$$\sum_{j\in\mathcal{J}} x_{ij} \le 1, \quad \forall i \in \mathcal{I}, \tag{A.1}$$

$$\sum_{i\in\mathcal{I}} x_{ij} \le 1, \quad \forall j \in \mathcal{J}, \tag{A.2}$$

$$\sum_{k\in\mathcal{K}} y_{jk} \le 1, \quad \forall j \in \mathcal{J}, \tag{A.3}$$

$$\sum_{j\in\mathcal{J}} y_{jk} \le B_k, \quad \forall k \in \mathcal{K}, \tag{A.4}$$

$$\sum_{i\in\mathcal{I}} x_{ij} \ge \sum_{k\in\mathcal{K}} y_{jk}, \quad \forall j \in \mathcal{J}, \tag{A.5}$$

$$\sum_{j\in\mathcal{J}} y_{jk} \cdot q_{js} \ge O_{ks} - z_{ks}, \quad \forall k \in \mathcal{K},\ s \in \mathcal{S}, \tag{A.6}$$

$$x_{ij} = 0, \quad \forall i \in \mathcal{I}_a,\ j \in \mathcal{J}_b, \tag{A.7}$$

$$x_{ij} = 0, \quad \forall i \in \mathcal{I}_b,\ j \notin \mathcal{J}_{t_i}, \tag{A.8}$$

$$\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_t} x_{ij} \le 1, \quad \forall t \in \mathcal{T}, \tag{A.9}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{I},\ j \in \mathcal{J}, \tag{A.10}$$

$$y_{jk} \in \{0,1\}, \quad \forall j \in \mathcal{J},\ k \in \mathcal{K}, \tag{A.11}$$

$$z_{ks} \ge 0,\ z_{ks} \in \mathbb{Z}_+, \quad \forall k \in \mathcal{K},\ s \in \mathcal{S}. \tag{A.12}$$

We interpret these constraints as follows:

- Constraint (A.1) requires that each AGV can be assigned to at most one side.
- Constraint (A.2) requires that each side can be picked up by at most one AGV.
- Constraint (A.3) requires that each side can be assigned to at most one workstation.
- Constraint (A.4) requires that the number of rack sides assigned to a workstation cannot exceed the number of available berths in that workstation.
- Constraint (A.5) requires that whenever an AGV is assigned to a rack side, the rack side must also be assigned to a workstation. Otherwise, if a rack side is not assigned to an AGV, it should not be assigned to any workstation.
- Constraint (A.6) requires that when some rack sides are assigned to a workstation, the SKUs on these rack sides are used toward fulfilling the order requirement in that workstation, while any unsatisfied requirements are recorded in z_{ks}.
- Constraint (A.7) requires that an idle AGV should never be assigned to a rack side in movement.
- Constraint (A.8) requires that an occupied AGV cannot be assigned to rack sides except the one or two rack sides on the rack that the AGV is currently transporting. As we discuss previously, for system-stability reasons we do not reassign an occupied AGV.
- Constraint (A.9) requires that on a double-sided rack, only one side can be chosen when we want to assign an AGV to pick up the rack. Therefore, only one AGV will be assigned to a rack and only one side of the rack could be ready for selecting by the robot arms each time, regardless of whether the rack is double sided.
- Constraints (A.10), (A.11), and (A.12) define the domain of the decision variables.

Constraints (A.5) and (A.9) collectively also guarantee that both sides of a double-sided rack will not be assigned to workstations at the same time.

Solution Method

Problem ($\mathcal{P}$) is a large-scale integer programming problem, which is hard to solve for the exact solution in practice. Furthermore, this tripartite matching problem needs to be solved in an extremely limited computational time; for example, the operating time for one period is only three seconds. To give some approximate figures, in a typical warehouse, we deploy $|\mathcal{I}| = 250$ AGVs, 1,800 racks (with $|\mathcal{J}| = 3{,}300$ rack sides), and $|\mathcal{K}| = 50$ workstations to handle more than $|\mathcal{S}| = 2{,}000$ different types of products. Thus, if we use model ($\mathcal{P}$) to guide our dispatching decisions, Problem ($\mathcal{P}$) has approximately $O(10^6)$ decision variables and $O(10^6)$ constraints, where $O(\cdot)$ is the "Big O" notation and indicates order of magnitude. To find a good solution (although possibly not optimal given the time constraint), we mainly adopt the idea of the "divide and conquer" approach and separate the main problem into two smaller ones and solve them separately.

In Problem ($\mathcal{P}$), because only Constraint (A.5) links the decision variables $\{x_{ij}\}$ and $\{y_{jk}\}$, we divide the problem into two subproblems by introducing the Lagrange multipliers associated with Constraint (A.5). Let $\Lambda := \{\lambda_j \ge 0, j \in \mathcal{J}\}$ be the Lagrange multipliers. We then consider the following partially relaxed problem ($\hat{\mathcal{P}}(\Lambda)$):

$$\hat{\mathcal{P}}(\Lambda) \quad \min_{X,Y,Z} \ \alpha_1 \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c^1_{ij} x_{ij} + \alpha_2 \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} c^2_{jk} y_{jk} + \alpha_3 \sum_{k\in\mathcal{K}} \sum_{s\in\mathcal{S}} z_{ks} + \sum_{j\in\mathcal{J}} \lambda_j \left(\sum_{k\in\mathcal{K}} y_{jk} - \sum_{i\in\mathcal{I}} x_{ij} \right)$$

$$\text{(s.t.)} \quad \{X, Y, Z\} \quad \text{satisfy Constraints (A.1)–(A.12).}$$

Below, we use $V(\cdot)$ to denote the optimal objective value of problem $(\cdot)$. According to the duality theory, for any Λ, the optimal objective value $V(\hat{\mathcal{P}}(\Lambda))$ provides a lower bound of the original problem ($\mathcal{P}$), that is, $V(\hat{\mathcal{P}}(\Lambda)) \le V(\mathcal{P})$. Generally speaking, we can strengthen such a lower bound by finding the optimal dual variable Λ^*, that is, $\Lambda^* = \arg\max_\Lambda V(\hat{\mathcal{P}}(\Lambda))$. However, searching for the optimal dual variable Λ is still a difficult problem. At this stage, we assume we have some known Λ for Problem ($\hat{\mathcal{P}}(\Lambda)$). For any fixed Λ, we can rewrite the objective function of Problem ($\hat{\mathcal{P}}(\Lambda)$) as

$$\min_{X,Y,Z} \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}(\alpha_1 c^1_{ij} - \lambda_j)x_{ij} + \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}(\alpha_2 c^2_{jk} + \lambda_j)y_{jk} + \alpha_3 \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}} z_{ks}.$$

Based on the previous formulation, we observe that the decision variables $\{x_{ij}\}$ and the variables $\{y_{jk}, z_{ks}\}$ are totally separated in Problem ($\hat{\mathcal{P}}(\Lambda)$). Thus, solving Problem ($\hat{\mathcal{P}}(\Lambda)$) is equivalent to solving two subproblems ($\mathcal{P}^1(\Lambda)$) and ($\mathcal{P}^2(\Lambda)$):

$$\mathcal{P}^1(\Lambda): \quad \min_X \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}(\alpha_1 c^1_{ij} - \lambda_j)x_{ij}$$

(s.t.) X satisfies Constraints (A.1), (A.2), (A.7), (A.8), (A.9), and (A.10),

and

$$\mathcal{P}^2(\Lambda): \quad \min_{Y,Z} \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}(\alpha_2 c^2_{jk} + \lambda_j)y_{jk} + \alpha_3 \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}} z_{ks}$$

(s.t.) $\{Y, Z\}$ satisfy Constraints (A.3), (A.4), (A.6), (A.11), and (A.12).

In general, the first problem represents the matching between rack sides and AGVs and the second problem represents the matching between rack sides and workstations.

Note that solving the previous two subproblems separately may not provide a feasible solution to the original problem ($\mathcal{P}$), because Constraint (A.5) may not be satisfied. A better way to use the previous decomposition structure is to solve the subproblem ($\mathcal{P}^2(\Lambda)$) first and use the solution of the subproblem ($\mathcal{P}^2(\Lambda)$) as the input for the subproblem ($\mathcal{P}^1(\Lambda)$). More specifically, after we achieve the solution $\{y^*_{jk}\}$ from Problem ($\mathcal{P}^2(\Lambda)$), we solve the modified version of the subproblem ($\mathcal{P}^1(\Lambda)$) as follows:

$$\mathcal{P}^1: \quad \min_X \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c^1_{ij} x_{ij}$$

$$\text{(s.t.)} \quad \sum_{i\in\mathcal{I}} x_{ij} \ge \sum_{k\in\mathcal{K}} y^*_{jk}, \quad \forall j \in \mathcal{J},$$

X satisfies Constraints (A.1), (A.2), (A.7), (A.8), (A.9), and (A.10).

Compared with Problem ($\mathcal{P}^1(\Lambda)$), Problem ($\mathcal{P}^1$) has an additional constraint $\sum_{i\in\mathcal{I}} x_{ij} \ge \sum_{k\in\mathcal{K}} y^*_{jk}$ to ensure the feasibility of the solution. Here, $\{y^*_{jk}\}$ provides the information of rack sides that are chosen to match the demand in different workstations. Using $\{y^*_{jk}\}$ as the input in Problem ($\mathcal{P}^1$) gives the matching solution $\{x_{ij}\}$ of the AGVs for these rack sides. We also found that changing the coefficients in the objective function from $\alpha_1 c^1_{ij} - \lambda_j$ to c^1_{ij} does not impact the quality of the solution. Hence, we use $\{c^1_{ij}\}$ in the objective function of Problem ($\mathcal{P}^1$).

The key issue is to now find a good value for the dual variable Λ. After solving the continuous relaxation of the original problem ($\mathcal{P}$), we can get the correspondent dual variables $\hat{\Lambda}$ of Constraint (A.5). We then use $\hat{\Lambda}$ as a surrogate of the optimal Λ. However, the solution to the continuous relaxation of Problem ($\mathcal{P}$) is not fast enough to match the operating period. Fortunately, updating Λ in each operating period is not necessary. We update Λ only if doing so is necessary, for example, computing Λ in a longer time interval. In summary, we adopt the following steps to solve Problem ($\mathcal{P}$) in each operating period:

1. Update Λ if doing so is necessary;
2. Solve Problem ($\mathcal{P}^2(\Lambda)$) for solution $\{y^*_{jk}\}$ and $\{z^*_{ks}\}$;
3. Solve Problem ($\mathcal{P}^1$) by using $\{y^*_{jk}\}$ as the input for the solution $\{x^*_{ij}\}$. Below, we discuss in detail how to solve Problems ($\mathcal{P}^2(\Lambda)$) and ($\mathcal{P}^1$).

Solutions for Two Subproblems ($\mathcal{P}^1$) and ($\mathcal{P}^2(\Lambda)$)

Let us focus first on the second subproblem ($\mathcal{P}^2(\Lambda)$). The subproblem ($\mathcal{P}^2(\Lambda)$) is still an integer programming problem, although a smaller one than the original problem ($\mathcal{P}$). Our main idea is to solve the corresponding linear relaxation problem (e.g., relax $y_{jk} \in \{0,1\}$ to $y_{jk} \in [0,1]$ for all $i \in \mathcal{I}, j \in \mathcal{J}$ and $z_{js} \in \mathbb{Z}_+$ to $z_{js} \in \mathbb{R}_+$) and round the continuous solutions to the integer solutions. We use ($\mathcal{P}^2_{lp}(\Lambda)$) to denote the relaxed problem for Problem ($\mathcal{P}^2(\Lambda)$). We adopt the following heuristic to enhance the quality and efficiency of the previous solution procedure.

We add some cutting planes in the relaxed problem ($\mathcal{P}^2_{lp}(\Lambda)$) to tighten the constraints, which force the integer solution to expose itself at the extreme points. We mainly add two types of cuts: the *lifting cuts* and the *minimum cover cuts*. We refer the interested reader to Sierksma and Zwols (2015) for a detailed discussion of these cuts. We introduce these two types of cuts to strengthen the demand-satisfaction Constraint (A.6) and illustrate in the next sentence how to add the cuts. For some $k \in \mathcal{K}$ and $s \in \mathcal{S}$, Constraint (A.6) essentially says $\sum_{j\in\mathcal{J}} y_{jk} \cdot q_{js} \ge O_{ks}$ (here we ignore the auxiliary variable z_{ks} for illustrative purposes).

We first talk about lifting cuts. If we can identify some j^* such that $q_{j^*,s} \ge O_{ks}$, we may introduce a new constraint $\sum_{j\in\mathcal{J}} y_{jk} \cdot q_{js} - y_{j^*,k} \cdot s \ge O_{ks}$, where $s = q_{j^*,s} - O_{ks}$. Substituting s in the previous constraint gives

$$\sum_{j\in\mathcal{J}, j\neq j^*} y_{jk} \cdot q_{js} + y_{j^*,k} \cdot O_{ks} \ge O_{ks}. \tag{A.13}$$

We can thus see that Constraint (A.13) is tighter than the original one. Thus, in our implementation, if we can find some j^*, we replace Constraint (A.6) by Constraint (A.13). This procedure has the similar effect of adding a valid inequality but avoids adding new constraints. If multiple j^* exist, repeat the same procedure for all j^*.

We next discuss minimum cover cuts. A minimum cover cut is usually referred to the cut designed for the knapsack problem. We slightly misuse the name here because our method is developed in the same spirit. We sort $\{q_{js}\}$ with respect to j in a descending order, namely, $q_{j_1,s} \ge q_{j_2,s} \ge \ldots$ If $\ell^* \ge 1$ exists such that $\sum_{u=1}^{\ell^*} q_{j_u,s} \ge O_{ks}$, we may add a constraint $\sum_{j\in\mathcal{J}} y_{jk} \ge \ell^*$, interpreted as requiring at least ℓ^* rack sides to satisfy the workstation-SKU combination (k, s). This constraint is particularly useful when all the racks have relatively low inventory levels (i.e., q_{js} are all small and ℓ^* is large). By contrast, if j exists such that $q_{js} > O_{ks}$, then $\ell^* = 1$ and the constraint $\sum_{j\in\mathcal{J}} y_{jk} \geq 1$ becomes trivial and less useful.

Note that adding cuts does not guarantee an integer solution by solving the relaxed problem. The purpose of adding these cuts is to enhance the quality of the solution derived from the relaxed problem ($\mathcal{P}^2_{lp}(\Lambda)$), namely, to achieve a better outcome after the rounding procedure. We also found that even a small number of cuts is sufficient to find a high-quality (albeit suboptimal) solution.

In addition to the technique of strengthening the constraints, we adopt the preprocessing procedure to reduce the problem size before solving the relaxed problem. Intuitively speaking, for each workstation k, if a rack side is far away from this workstation and its inventory is not sufficient to satisfy the requested SKU, we may eliminate this rack side. More specifically, for each workstation k, we introduce the cost-benefit value for each rack side as follows:

$$V_{kj} = (\alpha_2 c^2_{jk} + \lambda_j) + \alpha_3 \sum_{s\in\mathcal{S}} \max\{0,\ O_{ks} - q_{js}\},\ j \in \mathcal{J}.$$

We then set up a cost-benefit threshold: if V_{kj} is greater than the threshold, we eliminate the corresponding rack side from the candidate set.

When we solve the linear programming problem ($\mathcal{P}^2_{lp}(\Lambda)$), we adopt the dual simplex method (Bersimas and Tsitsiklis 1997), because the number of constraints is far fewer than the number of decision variables. Adopting the dual simplex method significantly speeds up the solution process.

In the rounding process, we only need to round the solutions of $\{y^*_{jk}\}$. Then, $\{z^*_{ks}\}$ can be computed as below:

$$z^*_{ks} = \max\left\{0,\ O_{ks} - \sum_{j\in\mathcal{J}} y^*_{jk} \cdot q_{js}\right\},\ k \in \mathcal{K}, s \in \mathcal{S}.$$

Next, we move to the subproblem ($\mathcal{P}^1$), which is an *unbalanced assignment problem*. Such a problem is known to be equivalent to its linear programming relaxation, which can be solved efficiently by, for example, the Hungarian algorithm (e.g., Bersimas and Tsitsiklis 1997, Burkard and Çela 1999). The complexity of this algorithm is approximately $O(\max\{|\mathcal{I}|, |\mathcal{J}|\}^4)$. In our implementation, we further reduce the size of Problem ($\mathcal{P}^1$) by a decision-variable replacement. In Problem ($\mathcal{P}^1$), we need to match the AGVs with the chosen racks but not AGVs with the rack sides. Once we know $\{y^*_{jk}\}$, we have the information about which racks are chosen. Thus, in Problem ($\mathcal{P}^1$), we replace the original variables that represent the match between the AGVs and the rack sides with the new variables that represent the match between AGVs with racks. Such a replacement may help us reduce almost half of the decision variables in Problem ($\mathcal{P}^1$).

In summary, we decompose the original problem ($\mathcal{P}$) into two subproblems ($\mathcal{P}^1$) and ($\mathcal{P}^2$), where ($\mathcal{P}^1$) is essentially a linear programming problem and ($\mathcal{P}^2$) is a much smaller integer program. For example, in a typical warehouse with 250 AGVs, 1,800 racks (with 3,300 rack sides), 50 workstations, and 2,000 SKUs, Problem ($\mathcal{P}$) has approximately $O(10^6)$ decision variables. After the decomposition, Problem ($\mathcal{P}^1$) is still large (but it is essentially a linear program), whereas the size of the integer programming problem ($\mathcal{P}^2$) is reduced significantly (by approximately 75%).

To evaluate the performance, we compare our algorithm with the commercial mixed-integer programming solver. For details on this performance, please see the electronic companion.

References

Azadeh K, De Koster R, Debjit R (2019) Robotized and automated warehouse systems: Review and recent developments. *Transportation Sci.* 53(4):917–945.

Banker S (2016) Robots in the warehouse: It's not just Amazon. Accessed January 22, 2021, https://www.forbes.com/sites/stevebanker/2016/01/11/robots-in-the-warehouse-its-not-just-amazon/?sh=1ddf1ae440b8.

Bartholdi JJ, Hackman ST (2016) *Warehouse & Distribution Science: Release 0.97* (Supply Chain and Logistics Institute, Georgia Institute of Technology, Atlanta).

Bersimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization* (Athena Scientific, Nashua, NH).

Boysen N, Briskorn D, Emde S (2017) Parts-to-picker based order processing in a rack-moving mobile robots environment. *Eur. J. Oper. Res.* 262(2):550–562.

Burkard RE, Çela E (1999) Linear assignment problems and extensions. Du D-Z, Pardalos P, eds. *Handbook of Combinatorial Optimization* (Springer, Boston), 75–149.

CCTV (2018) Intelligence drives modernization, JD.com's unmanned warehouse begins digital China. Accessed January 22, 2021, https://www.youtube.com/watch?v=vW7GF_RvaL8.

CCTV-4 (2017) Across China. Accessed January 22, 2021, https://www.youtube.com/watch?v=SOR0hy30GGo.

Cezik T, Graves SC, Liu A (2021) Velocity-based stowage policy for semi-automated fulfillment system. Preprint, submitted February 18, https://dx.doi.org/10.2139/ssrn.3784731.

China Daily (2017) JD.com sets up first unmanned warehouse in Jiading. Accessed January 22, 2021, https://www.chinadaily.com.cn/m/beijing/zhongguancun/2017-10/16/content_33328438.htm.

Eisenstein DD (2008) Analysis and optimal design of discrete order picking technologies along a line. *Naval Res. Logist.* 55(4):350–362.

JD.com (2017a) JD.com fully automated warehouse in Shanghai. Accessed January 22, 2021, https://www.youtube.com/watch?v=RFV8IkY52iY.

JD.com (2017b) JD.com announces establishment of JD Logistics business group. Accessed January 22, 2021, https://www.globenewswire.com/news-release/2017/04/25/970694/0/en/JDcom-Announces-Establishment-of-JD-Logistics-Business-Group.html.

JD.com (2018) Media coverage of JD.com's unmanned warehouse. Accessed January 22, 2021, https://youtu.be/CW9ts-sxSHY.

Huang E (2017) In China, a robot has started delivering packages to people. Accessed January 22, 2021, https://qz.com/1009155/chinas-second-largest-ecommerce-company-jd-jd-just-used-a-robot-to-deliver-packages/.

Lin J, Singer PW (2017) In China, an e-commerce giant builds the world's biggest delivery drone. Accessed January 22, 2021, http://www.popsci.com/jd-com-builds-worlds-biggest-delivery-drone.

Mountz MC, D'Andrea R, LaPlante JA, Lyons DP, Mansfield PK, Amsbury BW (2008) Inventory system with mobile drive unit and inventory holder. US Patent 7,402,018 B2, filed October 14, 2004, issued July 22, 2008.

Palmer A (2018) Chinese e-commerce company is running a nearly autonomous warehouse with almost ZERO human employees. Accessed June 28, 2021, https://www.dailymail.co.uk/sciencetech/article-5845805/Chinese-e-commerce-company-JD-com-running-nearly-autonomous-warehouse.html.

Sierksma G, Zwols Y (2015) *Linear and Integer Optimization: Theory and Practice*, 3rd ed. (Chapman and Hall/CRC, Boca Raton, FL).

Wang Z, Sheu J-B, Teo C-P, Xue G (2021) Robot scheduling for mobile-rack warehouses: Human-robot coordinated order picking systems. *Production Oper. Management*, ePub ahead of print March 8, https://doi.org/10.1111/poms.13406.

Wulfraat M (2012) A supply chain consultant evaluation of Kiva systems (Amazonrobotics). Accessed January 22, 2021, https://mwpvl.com/html/kiva_systems.html.

Wurman PR, D'Andrea R, Mountz M (2008) Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine* 29(1):9–19.

Yuan R, Cezik T, Graves SC (2018) Stowage decisions in multi-zone storage systems. *Internat. J. Production Res.* 56(1–2):333–343.

Yuan R, Graves SC, Cezik T (2019) Velocity-based storage assignment in semi-automated storage systems. *Production Oper. Management* 28(2):354–373.

Hengle Qin is a director of logistics technology at JD Logistics, JD.com. Since joining JD in 2011, he has been playing a key role in the company's Asia No.1 fulfillment center projects. He started to lead JD's unmanned warehouse project in 2015, where he and his team achieved significant accomplishments in scheme design, algorithm design, and efficiency improvement of unmanned warehouse technologies. He is currently pursuing a PhD at the Shanghai University of Finance and Economics.

Jun Xiao is a vice president at JD.com and the president of JD-X, JD Logistics. Since joining JD in 2007, he has led the design and research and development of logistics systems. He is the founder of JD's warehousing and distribution systems. He built a team from scratch, led the team to exert innovation and entrepreneurial spirit, and quickly achieved significant results in unmanned warehouses, drones, unmanned vehicles, unmanned supermarkets, and artificial intelligence.

Dongdong Ge is a professor and the dean of the Research Institute for Interdisciplinary Sciences at the Shanghai University of Finance and Economics. He received his PhD degree from MS&E at Stanford University. His main research interests lie in large-scale optimization algorithms and its applications in supply chain and intelligent industry. He is also the chief scientist and a cofounder of Cardinal Operations (shanshu.ai) and leads the development of mathematical programming solver COPT.

Linwei Xin is an assistant professor of operations management at Booth School of Business, University of Chicago. His primary research interest is supply chain and inventory management. His work on stochastic inventory theory by using asymptotic analysis has been recognized with several INFORMS paper competition awards, including the Applied Probability Society Best Publication Award (2019) and first place in the George E. Nicholson Student Paper Competition (2015).

Jianjun Gao is an associate professor at Shanghai University of Finance and Economics, China. He got his BE from the University of Science and Technology of China and PhD from the Chinese University of Hong Kong. His research interests include optimization and optimal control theory with applications in management and finance. He has conducted several funded research projects from the NSFC and also participated in several industry-based consulting projects in various areas.

Simai He is a professor at the Research Institute for Interdisciplinary Sciences, Shanghai University of Finance and Economics. He receives BSc in mathematics from the University of Science and Technology of China and PhD in operations research from the Chinese University of Hong Kong. His research focuses on optimization theory, operations research, and supply chain management. He has received the Award for Young Scientists from the Operations Research Society of China in 2014.

Haodong Hu is an associate professor at Shanghai University of Finance and Economics. He holds a PhD in CS from Stony Brook University and his BA in mathematics from Nankai University. His major research interests are application of operations research, cache-oblivious structures, and their application in big data. He worked on Windows and Bing Search Platform at Microsoft. He has more than 14 years of practical experience in software and algorithm development.

John Gunnar Carlsson is the Kellner Family Associate Professor of Industrial and Systems Engineering at the University of Southern California. He received a PhD in computational mathematics from Stanford University in 2009 and an AB in music and mathematics from Harvard College in 2005. He serves as an associate editor for *Operations Research, Management Science, Transportation Science,* and *Computers and Operations Research.*