Optimal Scheduling of Granulation, Packaging, and Routing Operations in Make-to-Order Fresh Produce Supply Chain

Mu Du[1,2], Nan Kong[2*], Xiangpei Hu[1]

[1]*Institute of Systems Engineering, Dalian University of Technology, Dalian, China*

[2]*Weldon School of Biomedical Engineering, Purdue University, United States*

Mu Du: dumu@mail.dlut.edu.cn. No.2 Linggong Rd., Dalian University of Technology, Dalian, Liaoning, China. 116024. +86 411 84707336

Nan Kong *(corresponding author): nkong@purdue.edu. 206 S. Martin Jischke Dr. West Lafayette, IN 47907, United States. +1 765 4962467

Xiangpei Hu: drhxp@dlut.edu.cn. No.2 Linggong Rd., Dalian University of Technology, Dalian, Liaoning, China. 116024. +86 411 84707336

Optimal Scheduling of Granulation, Packaging, and Routing Operations in Make-to-Order Fresh Produce Supply Chain

Farm-to-home online retailing is a fast-growing business model for fresh produce in China. However, Chinese online retailers often fail to make profit due to discordant scheduling among their three major distribution center (DC) operations, namely, granulation, packaging, and routing. The lack of cooperation has resulted in substantial increase in order processing cost. Specifically motivated by the operational inefficiency of our industry partner, we propose the make-pack-route problem (MPRP) to devise schedules for the three DC operations, which incorporates flexible order composition. To solve the MPRP, we develop an iterative sequential scheduling method, which solves three scheduling problems sequentially, corresponding to the three DC operations, in an iterative manner. For the resultant delivery routing and granulation operation scheduling problems, we further develop clustering based construction heuristic methods. Our numerical studies verify its viability of solving large instances (e.g., up to 200 orders and 30 minimal sales units) in practically acceptable time duration (e.g., 1 hour), and show a minimal 1.5% total operational cost reduction compared to a commonly used ad-hoc sequential scheduling method. Our studies also suggest the algorithmic superiority becomes more noticeable with increased order number, variety, average quantity and demand variety in an order.

Keywords: Supply chain management; manufacturing management; Production distribution integrated scheduling; make-to-order supply chain; local search heuristic

1 Introduction

In recent years, farm-to-home online retailing of fresh produce has emerged as an attractive and competitive business model in China’s metropolitan areas. Presently, China has more than 4,000 fresh produce online retailers, whose total market shares top 6 billion US dollars. This make-to-order business model simplifies traditional fresh produce supply chain by connecting the farmers directly to the customers. It also caters the need of Chinese metropolitan residents, who have strict request on food freshness but hardly have any time for daily grocery shopping due to long commuting. However, most Chinese online retailers, including our Beijing-based industry partner, face challenges in making profit due to difficulties in daily production and distribution planning in their distribution centers (DCs).

Here are the major challenges. First, out of the satisfaction concern, online retailers usually allow customers to choose freely the type and quantity of the fresh-produce in their orders. This requires differentiating the steps of fresh produce granulation and intermediate order assembly in the production process but making coordinated decisions. Second, without retailing facilities at the community level, online retailers must deliver individual customer orders to geographically disperse households with tight due time through heavy urban traffic. Considering these practical challenges, efficient scheduling is needed for the three major valued-added DC operations: fresh produce granulation, intermediate product assembly, and finished product delivery.

There are two major challenges. First, out of the satisfaction concern, online retailers usually allow customers to choose freely the type and quantity of the fresh-produce in their orders. This requires differentiating the steps of fresh produce granulation and intermediate order assembly in the production process but making coordinated decisions. This clearly makes the scheduling for production in the DCs more challenging. Second, without retailing facilities at the community level, online retailers must deliver customer orders individually to geographically disperse households and with tight due time through heavy urban traffic. Considering the above challenges and requirements in practice, efficient and effective scheduling is needed for the three major valued-added DC operations: fresh produce granulation, intermediate product assembly, and finished product delivery.

In reality, conflicts often arise among the scheduling decisions of the three departments, especially when each department solely focuses on its own performance improvement. With allowance of making self-interest decisions, the department in charge of fresh produce granulation (i.e., *MSU-making* department) normally prefers using its granulation machines to prepare minimal sales units (MSUs) of the same fresh produce type as continuously as possible in order to minimize the cost incurred by the machine reconfiguration. On the other hand, the department in charge of order delivery (i.e., *order-delivery* department) typically prefers grouping orders based on customer geographic proximity. In addition, routing decisions allow delivery workers to maximize their familiarity to the delivery routes, and thus ensure scheduling convenience. It is clear that batching of fresh produce and clustering of customers can hardly be well aligned, which is the main reason conflicts occur. Taking a closer look at the production process, the department in charge of packing various fresh produces (i.e., *order-packing* department) to meet diverse ordering requirements is often positioned awkwardly to either align itself with the upstream MSU-making department or the downstream order-delivery department. If taking a reactive mode, the order-packing department would have to procure more machines and hire more people to reduce its processing time and help coordinate the operations on both sides. These planning decisions, however, often intensify the discordance between the MSU-making and order-delivery departments, which tends to result in higher inventory levels of both intermediate and finished products, and subsequently, lead to increased duration of the make-to-order production-delivery process on a daily basis. Therefore, scheduling of the three DC departments must be made in an integrative manner.

In this paper, we term our integrated scheduling decision problem the *make-pack-route* problem (MPRP). We develop a sequential scheduling heuristic, knowing that the proposed NP-hard problem is computationally intractable in real-world cases (e.g., up to 200 orders and 30 MSUs). Our method solves the three scheduling problems related to routing, packaging, and granulation, separately and iteratively in that order. Although the order of solving the three problems is similar to what is currently being done in practice, we consider a more holistic objective function in the MPRP. Moreover, iterative heuristic methods are applied in solving the routing and granulation scheduling problems, given that they demand the majority of the solution time.

The main contributions of this paper are twofold. First, we propose an iterative sequential scheduling method with local-search heuristic, which captures in greater detail the operational decisions and requirements from all three differentiable DC departments and incorporates key features of a make-to-order fresh produce supply chain, e.g., flexible combination in order placement. Second, we generate computational insights on the performance of the proposed heuristic with variable characteristics of the supply chain, e.g., differing in demand quantity and variety, as well as in holding cost.

2. Literature Review

For integrated production and distribution scheduling, the production-routing problem (PRP), which aims to coordinate plans for the production stage and its subsequent distribution stage with finished product storage facilities, is most relevant to our work. The PRP can be expressed as the integrated production and outbound distribution scheduling problem (IPODSP) considering batch delivery to multiple customers with routing decisions to make (Chen, 2010). Although Potts (1980) introduced the integrated production and distribution scheduling problem in early 1980s, the integrated production and routing decision problem was not studied until early 1990s by Chandra (1993) and Chandra and Fisher (1994). By considering routing in making the production and delivery scheduling decisions, both makespan and total cost would be reduced significantly. Since then, numerous PRP models have been proposed with consideration of diverse objectives. For example, minimization of the total cost was considered by

Boudia, Louly, and Prins (2007), Boudia and Prins (2009), Adulyasak, Cordeau, and Jans (2012), Amorim, Belo-Filho, Toledo, Almeder, and Almada-Lobo (2013), and Low, Chang, Li, and Huang (2014), maximization of the number of delivered orders was considered by Armstrong, Gao, and Lei (2008), Viergutz and Knust (2014), minimization of the makespan was considered by Geismar, Laporte, Lei, and Sriskandarajah (2008), and Gao, Qi, and Lei (2015), and minimization of the number of vehicles and delivery cost was considered by Chen and Vairaktarakis (2005), and Russell, Chiang, and Zepeda (2008). In addition, numerous local search heuristic methods (Boudia et al., 2007; Armstrong et al., 2008; Viergutz et al., 2014; Adulyasak, et al., 2015; Low et al., 2014; Absi, , Archetti, , Dauzère-Pérès & Feillet , 2015), metaheuristics (Boudia et al., 2009; Geismar et al., 2008; Russell et al., 2008; Fahimnia, Luong & Marian, 2012; Belo-Filho, Amorim, & Almada-Lobo, 2015), and branch-and-bound exact solution algorithms (Armstrong et al., 2008; Viergutz et al., 2014; Archetti, Bertazzi, Paletta, & Speranza, 2011) were proposed. Several comprehensive reviews on formulations and solution algorithms are available for the PRP (Chen, 2010; Adulyasak, Cordeau, & Jans, 2015; Lei, Oppenheim, Qi, Dong, Lee, & Wang, 2015).

From the literature review above, we conclude that the majority of the literature is focused on models with only one production stage. Thus, when considering two production stages (i.e., granulation and packaging), existing PRP models do not apply. We present a detailed comparison on the decision variables in Table 1. Consequently, algorithms of the PRP, cannot be implemented directly to solve the MPRP.

[Insert Table 1 here]

Table 1. Comparison between the MPRP and the relevant problems in the literature.

Moreover, to the best of our knowledge, Farahani, Grunow and Günther (2012) is the only paper that considers the scheduling problem with all three DC departments. The authors developed an iterative decision scheme that attempts to coordinate production and distribution planning. They reported that instances with 200 orders, 3 products, and 5 fixed combinations, could be solved in a reasonable amount of time. The main difference of their paper from ours is twofold. First, they considered a problem with limited numbers of product combinations. Thus their model does not involve detailed characterization of potential fresh produce orders. In our paper, we allow flexible combination of fresh produce in each order, which increases complexity both in the modeling and solution.

3. An Optimization Model for the MPRP

We start this section with an overview of the DC workflow and related operations, rooted at observations on our industry partner; see Figure A.1 in appendix A. In their practice, the DC stops receiving orders of the next day at 3 pm. Then sufficient quantities of fresh produce are procured from the farm and transported to the DC within the next two hours. Meanwhile, the DC generates their schedules according to the received orders and their capacities. Finally, the three DC departments finish their jobs based on the respective scheduling requirements. In this way, all the orders could be delivered before 7 am the next day. However, significant holding cost might be incurred due to lack of coordination among the three departments.

We next provide a detailed description of the DC operations; see Figure A.2 in appendix A. In the MSU-making department, fruits and vegetables are granulated into MSUs with automatic machinery. Typically, there are four types of granulation modes: binding, incising, palletizing and boxing. Appendix B shows a list of fruits and vegetables available in the partnering retailers' store and their corresponding granulation modes. Machines in this department are parallel and can be switched between the four modes with some cost. Once granulated, the MSUs are stored in the buffers and transferred to the order-packing department. The order-packing department has several packaging machines in parallel, which are equipped with MSU shelves. The finished order packages are stored in the buffer before distribution. Finally, the packages are loaded on trucks and delivered before the due time. Thereby, the proposed mathematical formulation for the MPRP includes the following decision variables:

- MSU-making department: production-sequencing and lot-sizing decisions for each item on each granulation machine at each period.
- Order-packing department: machine utilization decision at each period; worker employment decision in each day; and order-packing sequencing decision on each packaging machine at each period.
- Order-delivery department: deliver-sequencing decisions for each vehicle on each delivery tour, including delivery routing and departure time.

Now we turn our attention to the MPRP formulation. We first present the model assumptions. One, machines in the MSU-making department and order-packing department are parallel. Two, granulation machines can switch between the four processing modes and idle mode with no-zero switching time and monetary cost, which both obey the triangle inequality. Three, both time and monetary cost of the inbound logistics are ignored. Four, perishability is only addressed implicitly with a penalty much higher than the holding cost, since all the goods must be delivered within 24 hours of being picked from the field. Five, temporary workers are hired to hedge the fluctuations in the ordering. We assume that permanent and temporary labors have different salary rates and working efficiencies. Six, for simplification, we consider all the delivery vehicles homogeneous in the MPRP. The MPPR with heterologous vehicle could be modeled by adding a few decision variables and constraints. We present a framework of the MPRP model in Figure 1 and briefly describe the formulation in the following paragraph. The mathematical formulation is detailed in Appendix C.

[Insert Figure 1 here]

Figure 1. A schematic of the MPRP model.

In the MPRP formulation, the objective function aims at minimizing the total make-to-order processing cost, which includes MSU-making cost, order-packing cost, order-delivery cost, and holding costs of MSUs and finished products (packed orders ready for delivery). MSU-making constraints include those ensuring the MSU-making time assigned to each machine in each period would not exceed the length of the period and sufficient MSUs are made for all orders. Order-packing constraints include the following: constraints ensuring all the orders to be packed; constraints allowing each packing machine to be used in a certain period only if it is turned on in that period; and constraints ensuring sufficient machinery and workforce to be employed for order-packing operations. Order-delivery constraints include standard constraints for the capacitated vehicle routing problem.

The MPRP is an NP-hard problem since it is composed of a 2-stage flow-shop machine scheduling problem and a vehicle routing problem. We implemented the MPRP model in Gurobi by a PC with an Intel Xeon E3-1231 3.40 gigahertz processor and 8 gigabyte RAM. The instances with 10-15 orders could be solved in 2000s. However, when facing large instances, e.g. 100 orders, a standard PC could not even load the problem due to the memory limit. Thus, a heuristic method is needed.

4. An Iterative Sequential Scheduling Heuristic

In the current practice, an ad-hoc sequential scheduling method is widely used for scheduling the operations of the three DC departments. In this section, we first describe the ad-hoc scheduling method and then propose an improved heuristic, namely iterative sequential scheduling heuristic, which can solve large MPRP instances more efficiently.

4.1. The Ad-hoc Sequential Scheduling Method

In a nutshell, this ad-hoc sequential scheduling method implies making decisions on order routing, product assembly, and MSU granulation sequentially in that order (see Figure 2). More specifically, the order-delivery department first makes the delivery routing decisions with an objective of minimizing the delivery cost subject to given order delivery dues. Then the order-packing department schedules workforce and equipment for intermediate product assembly according to the order-delivery department decisions. Its goal is to minimize the order-packing (i.e., final assembly labor and machine setup) cost and order-holding cost. Finally, the MSU-making department schedules raw material granulation operations to satisfy the requests from the order-packing department with an objective of minimizing the MSU-making (i.e., MSU making and machine switching) cost and MSU-holding cost.

[Insert Figure 2 here]

Figure 2. A schematic of the ad-hoc sequential scheduling method.

This ad-hoc method is expected to ensure the coordination of the production-and-delivery processes to some degree by passing downstream decisions to upstream departments. Moreover, it alleviates the computational challenge in solving the MPRP. The method is known to the practitioners dealing with production and distribution operations scheduling in make-to-order supply chains (Van der Laan, Salomon, & Dekker, 1999). However, the ad-hoc sequential scheduling method only solves each of the three smaller scheduling problems once. In other words, it only realizes one-way decision information flow. As an improvement, we propose a local search heuristic, which coordinates the three types of the scheduling decisions in an iterative manner.

4.2 An Iterative Heuristic Sequential Scheduling Method

The proposed iterative sequential scheduling heuristic follows the schematic of the previous ad-hoc method. However, different from the previous method, our proposed method applies local search on the order-delivery schedule and improves the overall solution over iterations. Further, it involves two clustering-based heuristic algorithms to alleviate the computational burdens of the iterative MSU-making and initial order-delivery problems. Finally, when generating the initial order-delivery plan, we take into account the scheduling preference of the MSU-making department to improve the coordination effect.

Figure 3 depicts a schematic of the improved heuristic method. For simplicity, we use X_k, Y_k, and Z_k to denote the solutions of MSU-making, order-packing and order-delivery problems at iteration k, respectively. We first construct an initial solution (X_0, Y_0, Z_0). In particular, we construct the initial delivery plan Z_0 heuristically via an insertion-based heuristic. Then we compute Y_0 exactly via a standard IP solver, and compute X_0 with an MUS-batching-and-sequencing heuristic. We set (X_0, Y_0, Z_0) as the initial incumbent solution and calculate its objective function value for the total supply chain cost. Then at each iteration k, we generate the neighborhood candidates of Z_k with a set of neighborhood structures and some level of randomization. We then search through the generated neighborhood candidates to seek a better solution to replace the incumbent one. Due to the randomization, we may not see any improvement at some of the iterations. Even without the improvement, we move to the next iteration to randomly generate a new set of neighbors and compare them with the incumbent solution. The above procedure is repeated until either no improvement is made over a pre-specified period of iterations or the solution time reaches a pre-specified threshold.

[Insert Figure 3 here]

Figure 3. A schematic of the iterative heuristic sequential scheduling.

As illustrated, our local search heuristic involves four key components, which are (i) an order-batching-and-sequencing heuristic for the initial order-delivery problem; (ii) an MSU-batching-and-assignment heuristic for iterative MSU-making problems; (iii) neighborhood structures to generate local search moves on the order-delivery solution; and (iv) a greedy local search strategy. In the following paragraphs, we will explain each component in details.

We first describe an order-batching-and-sequencing heuristic algorithm (see Figure 4), which is intended to construct the initial order delivery plan efficiently. We select orders from an order pool O sequentially according to the similarities of both order contents and shipping addresses, and thus form batches of orders $\{R_1, R_2, \ldots\}$ for further consideration. To form each batch R_b, we sequence the orders in the pool non-increasingly according to their distances to DC to form an order list π, and then we insert the first order of π into R_b. We next sequence the remaining orders in the current order pool O non-increasingly based on the similarities of order contents and geographic distances to the orders in R_b to form an order list σ, and then we select the orders from σ sequentially until R_b exceeds the vehicle capacity or delivery due. Here, we introduce a weighting parameter α to balance the interests of improving production and delivery operations. A bigger α implies that more consideration is given to the order-delivery department and thus the resultant scheduling plan places more emphasis on delivery cost reduction. Next for each batch considered, we solve a TSP with the implementation of the subtour-elimination algorithm (Derochers & Laporte, 1991) to sequence the orders to be delivered and check whether the final delivery time for the batch exceeds the due time. If the final delivery time has exceeded the due time, we discard the last order from the batch and place it back in the order pool for the next-round selection. The remaining batch is then used to form the delivery plan based on the existing TSP solution. Once a TSP tour is generated for each batch, we essentially solve a vehicle routing problem heuristically and generate an initial delivery plan.

[Insert Figure 4 here]

Figure 4. A flowchart of the order-batching-and-sequencing heuristic.

Next we describe the MSU-batching-and-assignment heuristic algorithm (see Figure 5), which is intended to alleviate the computational burden of solving the MSU-making problem. Given the quantity of each MSU needed in each period t_s, which is known in advance at each iteration, we make the batching and slot assignment decisions for the MSU-making jobs at each t_s. We first put the MSUs of identical grocery type into the same set to form the set J. Then we sequence the un-done jobs non-increasingly into the set ω, according to their total processing times. We also sequence the machines in M non-increasingly into the set γ, according to their available slots S_m. Next, we assign the first jobs in ω to the first machine in γ, and then update the un-done job set and available slots of machines. The jobs could be split into smaller ones by the restriction of the machine slots. In this algorithm, we assign each MSU set to several slots of some granulation machine at that period so that we can reduce the MSU-holding cost effectively subject to the requirement of the order-packing department at the following period. Meanwhile, we make the assignment according to the granulation requirement of each MSU set so as to minimize the switching cost of each machine.

[Insert Figure 5 here]

Figure 5. A flowchart of the order-batching-and-sequencing heuristic.

The third essential component of the proposed heuristic method is the neighborhood structures used to update the incumbent order-delivery plan Z. To perform the local search, we consider two local search schemes adapted from the vehicle routing literature, namely order exchange and node move (Zachariadis, Tarantilis & Kiranoudis, 2009; Polat, Kalayci, Kulak & Günther, 2015). To utilize the first local search scheme, we select a pair of tours (i.e., order batches). For the selected pair, we exchange certain identical portion of the orders from the two tours and generate solutions of the new TSPs corresponding to the new tours. We consider three optional values for the identical portion of the orders to be exchanged. They are 30%, 50% and 80%. According to the portion given, we uniformly select a subset of orders from all the orders on each tour in the tour pair. This specification leads to three different neighborhood structures, which are labeled as OE-3, OE-5 and OE-8, respectively. To utilize the second local search scheme, we select a subset of nodes and rearrange their positions within the tour (i.e., intra-tour moves) or between the tours (i.e., inter-tour moves). In the MPRP, we consider each community as a node and aggregate all orders from the same community to have the same delivery address. Among both intra-tour moves and inter-tour moves, we specifically employ three types, namely swap, shift, and 2-opt. Thus this specification leads to six different neighborhood structures; see Figure 6 for graphic illustrations of the node move scheme. In summary, we employ nine different neighborhood structures to generate the candidate neighbor.

[Insert Figure 6 here]

Figure 6. Neighborhood search strategies considered.

The last key component of the heuristic is the local search strategy. We randomly sort the nine neighborhood structures described earlier. To check the feasibility of Z_k^c, we verify the constraints on vehicle capacity and order delivery due time for the newly generated set of candidate tours. If Z_k^c is infeasible, we discard it and move to the next neighborhood structure. If Z_k^c is feasible, we use it to solve the corresponding Y_k^c and X_k^c. Likewise, a neighborhood candidate that fails to find corresponding feasible Y_k^c and X_k^c will also been discarded. With feasible (X_k^c, Y_k^c, Z_k^c), we compare it with the incumbent solution. Once improvement is identified in terms of the total cost, we replace the incumbent and move to the next iteration. Otherwise, we move down the list of the candidate neighbors until we exhaust the whole list. If still no improvement found with all the neighbors, we will move to the next iteration without replacing the incumbent solution. With randomization in the neighbor generation, it is plausible that the incumbent solution is improved at the next iteration. The entire solution procedure terminates when either of the two pre-specified conditions is satisfied.

5 Numerical study

In this section, we report our numerical studies to show the viability of our proposed heuristic. With these studies, we also investigated the effects of several model parameters, including total order quantity, order variation, and available MSU number on the superiority of the heuristic. In addition, we used these studies to evaluate the performance of the heuristic with different MSU holding costs, period lengths, and delivery due times.

We randomly generated two sets of test instances: those with 10-15 orders (termed small instances), and those with 100-200 orders (termed large instances) that are closely based on the real setting of our industry partner in Beijing. In the small instance set, we considered 5, 10 or 15 types of available MSUs, and generated the available MSU quantity of each order following respective normal distributions $N(5, 2)$, $N(10, 2)$, and $N(15, 2)$. In addition, we considered a total of 6 periods with the length of each period being 900s, and set the requested delivery due time to be 9,000s. In the large instance set, we considered 10, 20 or 30 types of available MSUs, and generated the MSU quantity in each order following respective normal distributions $N(15, 2)$, $N(15, 4)$, and $N(15, 6)$. In addition, we extended the length of each period to 1800s and thus set the requested delivery due time to be 14400s. We randomly generated the available MSUs with equal probability from 62 common types of fruits and vegetables in Chinese grocery (see Table B.1 in the appendix). The detailed information on the production and delivery parameters is listed in Table D in the appendix. Finally, we considered 30 residential communities in Beijing and acquired their location and transportation information via the Baidu map (see Figure D.3 and Table D.7 and D.8 in the appendix). These communities are regular clients of our partner. We implemented the heuristics with Python 2.7, and solved encountered IPs with the MIP solver of Gurobi 6.0.5 whenever appropriate. All programs were run on a PC with an Intel Xeon E3-1231 3.40 gigahertz processor and 8 gigabyte RAM.

In our numerical studies, we first used the classes of small instances to tune the weighting parameter α in the iterative sequential scheduling (ISS) heuristic method. We considered 18 instances sets, each of which contained 5 instances. These instance sets differ by the number of orders (O), the number of MSUs available for ordering (M), and the average quantity of MSUs in each order (I). Varying O and M implies different business scales; varying M typically reflects the different seasons when fresh produces have different throughputs; and varying I may indicate different types of customers who have different purchase behaviors. We varied α from 0.1 to 0.9 with a step size of 0.1. As illustrated in Figure E.4 in the appendix, our comparative results suggest the relationship between the total cost and α is not entirely clear. However, we could obtain the lowest total cost in 63 out of 90 instances when setting the value of α to be between 0.6-0.9 in the heuristic. Moreover, the heuristic with α = 0.7 could yield the best solutions in 22 instances, we thus set α=0.7 for the ISS method in the following experiments.

After fixing α, we evaluated the computational performance of the ISS method on the 18 classes of small instances. To set benchmarks for the computational comparison, we used the exact solution of the MPRP via the MIP solver and the heuristic solution via the ad-hoc sequential scheduling (AHSS) method with encountered three IPs being solved via the MIP solver. For the comparison, we also set an identical solution time for all three methods. Intuitively, the AHSS method takes less time than the other two solution methods. We thus ran the AHSS method and took the solution time as the time limit for the other two solution methods. We next took into account the fact that the decision time requested to a real-world online retailer is typically less than 1 hour on processing orders and generating corresponding operational schedules. We noticed from our preliminary experiments that solving the order-packing problem took nearly negligible amount of time compared to the other two IPs (i.e., MSU-packing and order-delivery). We thus set the CPU time limit on those two IPs to be 1800 seconds each in the AHSS method.

We present the comparative study results on small instances in Table 2. In the table, the 18 instance sets are indexed by (O, M, I). Note that relatively large instances of the 3 classes couldn't be loaded in the memory allocated to Gurobi on a standard PC used for the experiments. In the table, we report the average total cost (termed Ave_TC) in RMB with each solution method over the 5 test instances of each class. We also report the unified solution time for all three methods. With the MIP solver, we obtained a solution gap on each test instance after 1-hour solution. We used the relative solution gap and the feasible solution to calculate the final lower bound obtained by the solver. We then used it to calculate an upper bound on the relative solution gap given the result obtained by the ISS method. These upper bounds are reported in the column entitled Ave_Gap_UB.

[Insert Table 2 here]

Table 2. Comparison of optimal schedules generated by solving the MPRP via the MIP solver, and the two heuristic methods.

Table 2 suggests that with the same amount of computational time, our proposed ISS method would yield much better schedules than the commonly used AHSS method and better schedules than the exact solution obtained directly via a standard MIP solver. Our results suggest that the ISS's relative gaps are reasonably small. Moreover, our results suggest that our method is relative robust to the increase on each of the three parameters (i.e., order number, order variety, and average MSU quantity in each order), as opposed to the other two methods. Note that the MIP solver even could not load the IPs for instances of (15,10,10) and (15,10,15). Although such comparison cannot be carried out for larger instances, we are confident the superiority of the heuristic over the MIP solver will be extended as it is expected to scale up better. In our studies, we also recorded the makespan of the production and distribution process. We also observed that ISS could reduce the makespan compared to AHSS. But its makespan reduction could not compete with the MIP solver.

Next we report the numerical study results on large test instances to assess the outperformance of the ISS method over the AHSS method with varied model parameters. When solving these large instances, ISS could always yield a solution within 1 hour, which justifies its real-world applicability. On the other hand, these instances could not even be loaded to the memory allocated to Gurobi on a standard PC. We tested 9 instance classes, each of which contains five instances. Due to the computational limit, we only compared ISS with AHSS with encountered MSU-making problem and order-delivery problem being solved with the proposed corresponding heuristics. For these instances, we again set the total time limit to be 1 hour for all three methods. Note that with the heuristic methods on the two computational expensive IPs, one hour was sufficient to solve these large instances. In Table 3, we present the total cost saving, operational cost savings of the order-packing and order-delivery departments, and order holding cost saving. The cost saving of MSU making department is relatively small (ranging from 1.59% to -0.02% with an average of 0.2%), so we omit it from Table 3.

[Insert Table 3 here]

Table 3. Comparison between ISS and AHSS on solving large instances.

Table 3 confirms that ISS could yield lower total cost than AHSS. This confirms our intent of making algorithmic improvement over the AHSS method. Intuitively, through multiple iterations in ISS, we could increase the level of coordination and thus generate an overall smoother production-distribution process. At a more detailed level, significant reduction arises in the order holding costs between production completion and order shipment; meanwhile only slight operational cost increase is incurred at the order-packing and order-delivery departments.

Furthermore, our results show that the total cost saving has clearly a positive correlation with the number of available MSUs, and somewhat a positive correlation with the number of orders and the variation on the MSU quantities. These results may suggest that with the iterative improvement, the ISS method is more capable to deal with increasing commodity variety and combination flexibility. Intuitively, with decisions made only once at each DC department with the AHSS method, increases in the aforementioned model parameters force the MSU-making and order-packing departments to utilize larger storage space and/or allow longer buffer time to prevent flow interruptions along the production-delivery process. To contrary, the ISS method can reduce these unnecessary costs because it helps realize iterative interactions between the production and the delivery phases in the algorithmic sense.

Finally, we report in Table 4 our numerical investigation on the relationship of the outperformance of the ISS method with respect to the order holding cost. Here, we set O=150, M=20, I=15, but varied the holding cost from the original baseline value (i.e., H-1, which is calculated based on the MSU and order holding costs listed in Table B.1 and Table D.6 of the appendix) to 10 times of the baseline value. We set the time limitation for ISS to be 1 hour again. Our results suggest that for commodities that require high holding costs, the retailer could see significant improvement on the total production-distribution cost. Intuitively, with increased holding cost, the AHSS method would have already paid more attention to the storage reduction of intermediate and finished products. Thereby, the ISS method could not make much further improvement with later iterations, since ISS is designed to focus more on the coordination of operational scheduling required at each DC department.

[Insert Table 4 here]

Table 4. The performance evaluation of ISS under different operational parameters.

6 Conclusions and Future Research

In this paper, we investigate the coordination of DC operational scheduling for make-to-order fresh produce supply chains, which was inspired by real practice of a fresh produce online retailer in Beijing, China. We propose an iterative sequential scheduling heuristic method for the problem of make-pack-route decision making at DC, which helps realize decision interaction and coordination among the three involved DC decision department, in a progressive manner. To accommodate the decision making under tight timeline (usually less than an hour), we develop local search based heuristics algorithms for the MSU-making and order-delivery problems, the two problems that present significant computational burden.

Our numerical studies verify the superiority of our proposed heuristic on reducing the total operational cost. Compared to a commonly used ad-hoc sequential scheduling method, the proposed heuristic employs several local search strategies to make progressive improvement and employs a more global optimization objective to coordinate the independent decisions. Compared to the exact solution method, the proposed algorithm decomposes the integrative scheduling problem into smaller IPs. As a result, our heuristic has shown great economical potential of improving the production and distribution scheduling for make-to-order supply chains by ensuring a robust total cost saving of 1.5%, which means an annual cost saving of $50,000 for a make-to-order fresh produce online retailer like our industry partner in Beijing. Further, our sensitivity analyses indicate that the proposed heuristic can better accommodate customer demand diversity and incorporate operational-level parameters along the production-delivery process. We believe our algorithm is not limited to the fresh produce online retailing sector. It is expected to be widely applicable to problems arising in make-to-order production and distribution.

We will conduct our future research in both modeling and solution aspects. For the modeling, we plan to consider the perishable nature of fresh produce by incorporating a freshness loss parameter in the MPRP model. In addition, we plan to model inventory capacities for both intermediates and finished goods, and capacity differences among delivery vehicles. For more customized solution improvement, we plan to request comprehensive history data of operations from our partner and similar retailers, and use the data to tune our heuristic method, especially on designs of neighborhood structure and local search strategy.

References:

Adulyasak, Y., Cordeau, J. F., & Jans, R. (2012). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1), 20- 45.

Adulyasak, Y., Cordeau, J. F., & Jans, R. (2015). The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55, 141-152.

Amorim, P., Belo-Filho, M. A. F., Toledo, F. M. B., Almeder, C., & Almada-Lobo, B. (2013). Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, 146(1), 208-218.

Armstrong, R., Gao, S., & Lei, L. (2008). A zero-inventory production and distribution problem with a fixed customer sequence. *Annals of Operations Research*, 159(1), 395- 414.

Archetti, C., Bertazzi, L., Paletta, G., & Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12), 1731-1746.

Belo-Filho, M. A. F., Amorim, P., & Almada-Lobo, B. (2015). An adaptive large neighborhood search for the operational integrated production and distribution problem of perishable products. International Journal of Production Research, 53(20), 6040-6058.

Boudia, M., Louly, M. A. O., & Prins, C. (2007). A reactive GRASP and path relinking for a combined production distribution problem. *Computers & Operations Research*, 34(11), 3402-3419.

Boudia, M., & Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production distribution problem. *European Journal of Operational Research*, 195(3), 703-715.

Chandra, P. (1993). A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society*, 681-692.

Chandra, P., & Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3), 503-517.

Chen, Z. L., & Vairaktarakis, G. L. (2005). Integrated scheduling of production and distri- bution operations. *Management Science*, 51(4), 614-628.

Chen, Z. L. (2010). Integrated production and outbound distribution scheduling: review and extensions. *Operations Research*, 58(1), 130-148.

Desrochers, M., & Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1), 27-36.

Farahani, P., Grunow, M., & Günther, H. O. (2012). Integrated production and distribution planning for perishable food products. *Flexible services and manufacturing journal*, 24(1), 28-51.

Fahimnia, B., Luong, L., & Marian, R. (2012). Genetic algorithm optimisation of an integrated aggregate production–distribution plan in supply chains. *International Journal of Production Research*, 50(1), 81-96.

Gao, S., Qi, L., & Lei, L. (2015). Integrated batch production and distribution scheduling with limited vehicle capacity. *International Journal of Production Economics*, 160, 13- 25.

Geismar, H. N., Laporte, G., Lei, L., & Sriskandarajah, C. (2008). The integrated production and transportation scheduling problem for a product with a short lifespan. *INFORMS Journal on Computing*, 20(1), 21-33.

Lei, L., Oppenheim, R., Qi, L., Dong, H., Lee, K., & Wang, S. (2015). Integrated production/distribution/routing planning for supply chain networks: a review. *Supply Chain Management and Logistics: Innovative Strategies and Practical Solutions*, 43.

Low, C., Chang, C. M., Li, R. K., & Huang, C. L. (2014). Coordination of production scheduling and delivery problems with heterogeneous fleet. *International Journal of Production Economics*, 153, 139-148.

Polat,O., Kalayci, C.B., Kulak,O., & Günther, H. O. (2015). A perturbation based variable neighborhood search heuristic for solving the Vehicle Routing Problem with simultaneous pickup and delivery with time limit. *European Journal of Operational Research*, 242(2), 369-382.

Potts, C. N. (1980). Technical note-analysis of a heuristic for one machine sequencing with release dates and delivery times. *Operations Research*, 28(6), 1436-1441.

Russell, R., Chiang, W. C., & Zepeda, D. (2008). Integrating multi-product production and distribution in newspaper logistics. *Computers & Operations Research*, 35(5), 1576- 1588.

Van der Laan, E., Salomon, M., & Dekker, R. (1999). An investigation of lead-time effects in manufacturing/remanufacturing systems under simple PUSH and PULL control strategies. *European Journal of Operational Research*, 115(1), 195-214.

Viergutz, C., & Knust, S. (2014). Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research*, 213(1), 293-318.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195(3), 729-743.

Table 1. Comparison between the MPRP and the relevant problems in the literature.

Table 2. Comparison of optimal schedules generated by solving the MPRP via the MIP solver, and the two heuristic methods.

Table 3. Comparison between ISS and AHSS on solving large instances.

Table 4. The performance evaluation of ISS under different operational parameters.

Figure 1. A schematic of the MPRP model.

Figure 2. A schematic of the ad-hoc sequential scheduling method.

Figure 3. A schematic of the iterative heuristic sequential scheduling.

Figure 4. A flowchart of the order-batching-and-sequencing heuristic.

Figure 5. A flowchart of the order-batching-and-sequencing heuristic.

Figure 6. Neighborhood search strategies considered.