Integrated Production Inventory Routing Planning for Intelligent Food Logistics Systems

Yantong Li, Feng Chu, *Senior Member, IEEE*, Chenpeng Feng, Chengbin Chu, and MengChu Zhou, *Fellow, IEEE*

***Abstract*— An intelligent logistics system is an important branch of intelligent transportation systems. It is a great challenge to develop efficient technologies and methodologies to improve its performance in meeting customer requirements while this is highly related to people's life quality. Its high efficiency can reduce food waste, improve food quality and safety, and enhance the competitiveness of food companies. In this paper, we investigate a new integrated planning problem for intelligent food logistics systems. Two objectives are considered: minimizing total production, inventory, and transportation cost and maximizing average food quality. For the problem, a bi-objective mixed integer linear programming model is formulated first. Then, a new method that combines an ϵ-constraint-based two-phase iterative heuristic and a fuzzy logic method is developed to solve it. Computational results on a case study and on 185 randomly generated instances with up to 100 retailers and 12 periods show the effectiveness and efficiency of the proposed method.**

***Index Terms*— Bi-objective optimization, ϵ-constraint-based two-phase iterative heuristic, food quality, intelligent food logistics system, integrated planning.**

I. Introduction

THE wide deployment of intelligent transportation systems (ITS) will significantly impact our life and society [1]. According to Crainic *et al.* [2], ITS development focuses on the following three major directions: 1) vehicle and infrastructure; 2) information and communication technologies (ICT)- related hardware and software; and 3) methodologies, e.g., models and algorithms. For a few decades, the development of the vehicle, infrastructure, and ICT has been improving ITS efficiency and safety and helping us alleviate environmental impact. Compared to the first two directions, methodology innovation is to some extent lagging behind and represents a great challenge for industry and academia. Intelligent logistics system (ILS), as an important area of ITS, strongly needs advanced ICT and novel method to enhance system performance and competitiveness [3]. It is realized that operations research-based decision-aid tools play a central role in doing so [4].

In traditional logistics systems, production, storage, and distribution activities are often managed sequentially based on their upstream activities. Facing the rapid progress of globalization, such a strategy has become less competitive. With the development of ICT, decision makers hope that decision-aid tools could give more support by simultaneously taking all actors' activities into consideration such that they are able to make optimal integrated decisions. A vendor-managed inventory (VMI) system in which a vendor manages the inventory of its retailers is a good example and has been successfully applied to Wal-Mart, Shell Chemical, etc. [5]. It can increase the overall revenue and profit of both vendors and retailers. An inventory routing problem (IRP), which arises from VMI applications, integrates inventory control and vehicle routing activities. It has been extensively studied by researchers and widely adopted by many companies. A production inventory routing problem (PIRP), as a generalization of IRP, has been attracting increasing attention of both researchers and practitioners. Its objective is to provide an optimal and integrated plan in which production, inventory, and routing activities are jointly optimized. The integration of these formerly decoupled activities can significantly improve system efficiency through more efficient resource utilization. The work [6] indicates that great cost reduction of logistics systems can be achieved by such integrated planning.

A food logistics system (FLS) is a specific logistics system that aims to provide final consumers with high-quality food products. Despite FLS shares some common characteristics with general logistics systems, it has its own features, e.g., 1) most food has a relatively short shelf life and starts deteriorating once being produced; 2) the process of food production, storage and transportation needs to respect strict conditions and regulations; 3) food supply and price fluctuate severely [7]; and 4) final consumers have high requirements

Manuscript received June 29, 2017; revised March 1, 2018; accepted May 6, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 71571061, Grant 71431003, Grant 71601062, and Grant 71701049, and in part by the PHC Cai Yuanpei Program under Grant 34644SB. The Associate Editor for this paper was K. Boriboonsomsin. *(Corresponding author: Chengbin Chu.)*

Y. Li is with IBISC, University of Evry, Université Paris-Saclay, 91025 Evry, France (e-mail: yantong.li@univ-evry.fr).

F. Chu is with IBISC, University of Evry, Université Paris-Saclay, 91025 Evry, France, and also with the School of Economics and Management, Fuzhou University, Fuzhou 350116, China (e-mail: feng.chu@univ-evry.fr).

C. Feng is with the School of Management, Hefei University of Technology, Hefei 230001, China (e-mail: cpfeng@hfut.edu.cn).

C. Chu is with the School of Management, Northwestern Polytechnical University, Xi'an 710065, China, and also with ESIEE Paris, Université Paris-Est, 93162 Noisy-le-Grand Cedex, France (e-mail: chengbin.chu@esiee.fr).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with the Renewable Energy Research Group, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2835145

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

for food quality, safety and traceability. These features complicate FLS. In fact, existing models and methods for the general logistics system cannot be directly applied. Operations research-based decision-aid tools for intelligent food logistics system (IFLS) can boost the coordination of all activities in FLS, save system-wide cost, and improve food quality and traceability. Embedding appropriate planning models and algorithms that cover intrinsic food characteristics into a decision-aid tool can significantly enhance system intelligence and efficiency.

Existing research on perishable food (item) logistics planning mainly focuses on the following problems: 1) production-scheduling-routing problem (PSRP); 2) production-inventory-direct delivery problem (PIDDP); and 3) IRP. These studies focus on different applications with a common ground where food quality is considered. However, the first problem excludes the inventory consideration and the second one does not consider routing while the third one fails to integrate production planning into its consideration. To the best of our knowledge, our group is the first to investigate the integrated PIRP considering food quality deterioration to maximize the total revenue [8], [9]. Yet our previous studies [8], [9] did not set food quality as an objective.

In this paper, we investigate a new integrated PIRP for IFLS considering two objectives, i.e., to minimize the total production, inventory and routing cost and to maximize the average food quality. The problem consists of establishing an integrated food production, inventory, and distribution plan while taking into account the customer preference for buying fresher food. Firstly, we formulate the integrated food production inventory routing problem (FPIRP) with a bi-objective mixed integer linear programming (MILP) model. Then a new approach that combines an ϵ-constraint-based two-phase iterative heuristic (ϵ-CTIH) and a fuzzy logic method is developed to solve the problem. Thirdly, a case study is presented to illustrate that the proposed approach is able to solve real-life instances, and computational experiments on 185 randomly generated instances are conducted to evaluate its performance. Computational results show that commercial software CPLEX can only solve small-size instances with up to 20 retailers, 5 periods and 1 vehicle within 14400 CPU seconds while our algorithm can solve larger-size instances with up to 100 retailers, 12 periods and 10 vehicles within 4000 CPU seconds on average. For all small-size instances, our algorithm can save much computation time while providing competitive Pareto solutions compared with CPLEX; for medium- and large-size instances, our algorithm is able to obtain a reasonable number of approximate Pareto solutions within an acceptable time. In summary, the main contributions of this study include: 1) studying a new bi-objective food production inventory routing problem; 2) proposing a bi-objective MILP model for the problem; and 3) developing an efficient approach that combines an ϵ-CTIH and a fuzzy logic method to solve it.

Section II reviews the related literature. Section III describes the problem. In Section IV, a combined method is presented. Computational experiments on an illustrative case and randomly generated instances are conducted in Section V. Finally, Section VI concludes the paper.

II. Literature Review

In this section, we first review the literature on PIRP for general logistics systems, and then report related research on FLS with food perishability consideration. Finally, we briefly review the multi-objective optimization methods.

A. PIRP for General Logistics Systems

In this subsection, we review research works on PIRP for general logistics systems without considering food-related specific features. PIRP, as a complex combinatorial optimization problem, was first studied in 1993 by Chandra and Fisher [6]. Since then, PIRP has been widely studied for its efficient solution methods. Exact solution methods for PIRP are relatively scarce because of its NP-hardness. To the best of our knowledge, branch-and-cut algorithms have been developed for PIRP with a single un-capacitated vehicle [10], a single capacitated vehicle [11] or multiple capacitated vehicles [12], [13]. An exact solution method has the advantage in proving optimality of solutions whereas it is extremely time-consuming and can thus only solve small-size instances. Meta-heuristics exploit solution space by a guided search procedure with accumulated search experience to avoid getting trapped into local optima. Several meta-heuristics have been developed for PIRP, e.g., greedy randomized adaptive search procedure and path relinking approach [14], memetic algorithm [15], Tabu search algorithm [16], [17], and adaptive large neighborhood search algorithm [18]. They can find near-optimal solutions and can be easily adapted to solve similar problems. However, the quality of the obtained solutions often needs to be evaluated by other techniques. Mathematical programming techniques enable standard MILP solvers, e.g., CPLEX, to be applied to find optimal or near-optimal solutions. Relaxation and decomposition techniques are often combined to solve complex combinatorial optimization problems efficiently. Brahimi and Aouam [19] develop a hybrid heuristic that combines a relax-and-fix and local search heuristic. Zhang *et al.* [20] propose an iterative MILP-based heuristic in which a MILP model is solved iteratively with a restricted set of updated candidate routes. Decomposition-based heuristics are widely used to decompose PIRP into production and distribution subproblems that are quickly solvable to obtain near-optimal solutions [21]. Absi *et al.* [22] propose a MILP-based two-phase iterative approach in which the original problem is decomposed into a lot-sizing subproblem and a routing subproblem. These two subproblems are solved sequentially and iteratively. Solyali and Haldun [23] propose a mathematical programming-based multi-phase heuristic. Chitsaz *et al.* [24] develop a three-phase decomposition matheuristic. For more details of PIRP, please see the excellent review [25].

B. Planning Problems for Perishable Food Logistics System

Research on logistics systems handling perishability focuses on the aforementioned three types of problems, i.e., PSRP, PIDDP, and IRP. For PSRP, Chen *et al.* [26] formulate a nonlinear model in which the production schedules and vehicle routes are determined simultaneously. Devapriya *et al.* [27] propose an evolutionary algorithm-based heuristic to solve a practical PSRP for a perishable product. Ahuja *et al.* [28] develop a greedy heuristic for PIDDP. Amorim *et al.* [29] study a bi-objective PIDDP to minimize the total cost and maximize the freshness. For IRP, Le *et al.* [30] develop a MILP model and propose a column generation-based solution approach. Mizaei and Seifi [31] propose a hybrid meta-heuristic with lost sale consideration. Soysal *et al.* [32] investigate an IRP with truckload-dependent distribution cost, service level consideration and demand uncertainty. Coelho and Laporte [33] study an IRP with age-based product price to maximize the total revenue. However, none of these studies has integrated production, inventory and routing for better system performance. Li *et al.* [8] propose the first MILP model for FPIRP with food quality deterioration to maximize the total revenue. The model is solved by CPLEX within 7200s. Fifty four instances with up to 15 retailers, 5 periods, 2 vehicles and 3 quality levels are tested to evaluate the effectiveness of the model. Then Li *et al.* [9] further study an FPIRP with time windows. They formulate a new MILP model and 45 instances with up to 40 retailers, 3 periods, 2 vehicles and 2 quality levels are directly solved by CPLEX within 7200s. Recenctly, Qiu *et al.* [34] study a real-world variant of PIRP with perishable inventory and develop a branch-and-cut algorithm to solve it.

C. Multi-Objective Optimization Methods

Generally, methods for solving multi-objective optimization problems can be classified into preference-based methods and generating methods [35], [36]. The former; e.g., goal programming, goal-attainment, and global criterion methods, take into consideration preferences of a decision maker during a solution process. They provide a single solution for a decision maker, which is not flexible and lacks information about alternative solutions. In contrast, generating methods; e.g., weighted sum, ϵ-constraint, and evolutionary methods etc. [37], aim to generate a set of Pareto optimal solutions without any preference from a decision maker. These methods have the advantages to provide a set of solutions that can be selected flexibly by a decision maker and have been adopted in many real-world applications [38], [39]. Among these methods, weighting methods have several shortcomings. For instance, many combinations of weights may result in the same solution, objective functions have to be scaled to a common scale before forming the weighted sum, and it is hard to control the expected number of Pareto solutions. Evolutionary methods [40] provide a set of approximate Pareto solutions that may be very far from the optimal Pareto solutions. The ϵ-constraint method is well-known as one of the most effective approaches, especially for bi-objective optimization problems [41]. It transforms an original multi-objective problem into a set of mono-objective problems that can be solved to obtain a set of Pareto solutions. It is superior to the weighting method because it is able to produce non-extreme efficient solutions, does not require scaling, and needs less effort to obtain an expected number of Pareto solutions [42].

Our study aims to develop a decision-aid tool for an intelligent food logistics system. The tool will provide a decision maker with a set of Pareto solutions, each corresponding to an integrated production inventory routing plan, and help select a solution if necessary. In particular, food quality that greatly impacts customer satisfaction is considered as an objective for the first time in such an integrated planning problem. Thus the studied problem is a bi-objective FPIRP that simultaneously minimizes the total cost and maximizes the average food quality.

III. Problem Description and Formulation

Consider a complete digraph $G=\{N,A\}$ with a set of vertices $N=\{0, 1, \dots, n\}$ and a set of arcs $A=\{(i, j) : i, j \in N, i \neq j\}$. A plant with limited production and storage capacities is located at vertex 0. It has a fleet of homogeneous vehicles $K=\{1, , 2, \dots, |K|\}$, each having capacity V. A set of n retailers $R = \{1, 2, \dots, n\}$ with a limited storage capacity are geographically located at vertices $\{1, \dots, n\}$. Consider a time horizon $T = \{1, 2, \dots, |T|\}$, the customer demand at each retailer is assumed to be deterministic and time varying. A bi-objective food production inventory routing problem (BFPIRP) consists of simultaneously determining production, storage, delivery, and routing planning to satisfy customer demand with food quality deterioration consideration. Note that food quality can be distinguished by a set of quality levels $Q = \{0, 1, \dots, |Q|\}$, where 0 represents the freshest food. The two objectives of BFPIRP are to simultaneously minimize the total production, inventory and routing cost and maximize the average quality of food provided to final customers. The decisions to be made for each period are: 1) how much to produce at the plant; 2) how much to replenish each retailer; 3) how to arrange the transportation routes for the planned deliveries; and 4) how to fulfill customer demand at each retailer.

The study is conducted under the following assumptions: 1) each vehicle's route starts and ends at the plant, and each vehicle can perform at most one trip within each period; 2) each retailer can be visited at most once within each period; i.e., split delivery is not allowed; 3) food quality degrades by one quality unit per period; and 4) once the quality goes beyond $|Q|$, food products can no longer be used to meet customer demand and should be directly turned into waste. The last two assumptions are realistic and are related to food characteristics. The first two assumptions are commonplace in IRP and PRP. They can be justified by the fact that it is always possible to reduce to length of the periods for the assumptions to hold. No split delivery assumption is further justified by the fact that deliveries disturb regular operations at the retailers who therefore prefer few deliveries. To ensure the consistence, we assume that the demand of any retailer in any period does not exceed the sum of the storage and vehicle capacities. To formulate the problem, the following parameters and variables are defined:

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Parameters:

- c_{ij} travel cost on arc $(i, j) \in A$;
- a_t unit production cost in period $t \in T$;
- b_t production setup cost in period $t \in T$;
- C production capacity;
- g_i^q initial inventory at $i \in N$ with $q \in Q$;
- D_i^t customer demand at retailer $i \in R$ in period $t \in T$;
- U_i inventory capacity of $i \in N$;
- h_i^q unit inventory holding cost per period at $i \in N$ with $q \in Q$;
- V vehicle capacity;

Decision Variables:

- p_t production quantity in period t;
- w_t binary variable equal to 1 if $p_t > 0$; otherwise 0;
- I_i^{qt} inventory level with quality q of retailer i at the end of period t;
- y_i^{qkt} delivery quantity with quality q to retailer i by vehicle k in period t;
- d_i^{qt} quantity of food used to fulfill customer demand with quality q at retailer i in period t;
- v_i^{kt} binary variable equal to 1 if retailer i is visited by vehicle k in period t; otherwise 0;
- x_{ij}^{kt} binary variable equal to 1 if arc (i, j) is traversed by vehicle k in period t; otherwise 0.

With the above description and notation, BFPIRP can be formulated as follows:

$$\min f_1 = \sum_{t\in T}(a_t p_t + b_t w_t) + \sum_{i\in N}\sum_{q\in Q}\sum_{t\in T} h_i^q I_i^{qt} + \sum_{(i,j)\in A}\sum_{k\in K}\sum_{t\in T} c_{ij} x_{ij}^{kt} \quad (1)$$

$$\min f_2 = \sum_{i\in R}\sum_{q\in Q}\sum_{t\in T} d_i^{qt} q \Big/ \sum_{i\in R}\sum_{t\in T} D_i^t \quad (2)$$

$$\text{s.t. } I_i^{q0} = g_i^q, \quad \forall i \in N, q \in Q \quad (3)$$

$$I_0^{qt} = I_0^{q-1,t-1} - \sum_{i\in R}\sum_{k\in K} y_i^{qkt}, \quad \forall q \in Q \setminus \{0\}, t \in T \quad (4)$$

$$I_0^{0t} = p_t - \sum_{i\in R}\sum_{k\in K} y_i^{0kt}, \quad \forall t \in T \quad (5)$$

$$I_i^{qt} = I_i^{q-1,t-1} + \sum_{k\in K} y_i^{qkt} - d_i^{qt}, \quad \forall i \in R, q \in Q \setminus \{0\}, t \in T \quad (6)$$

$$I_i^{0t} = \sum_{k\in K} y_i^{0kt} - d_i^{0t}, \quad \forall i \in R, t \in T \quad (7)$$

$$p_t \le C w_t, \quad \forall t \in T \quad (8)$$

$$\sum_{q\in Q} I_i^{qt} \le U_i, \quad \forall i \in N, t \in T \quad (9)$$

$$\sum_{q\in Q} d_i^{qt} = D_i^t, \quad \forall i \in R, t \in T \quad (10)$$

$$\sum_{i\in R}\sum_{q\in Q} y_i^{qkt} \le V, \quad \forall k \in K, t \in T \quad (11)$$

$$\sum_{q\in Q} y_i^{qkt} \le V v_i^{kt}, \quad \forall i \in R, k \in K, t \in T \quad (12)$$

$$\sum_{k\in K} v_i^{kt} \le 1, \quad \forall i \in R, t \in T \quad (13)$$

$$\sum_{j\in N\setminus\{i\}} x_{ij}^{kt} = \sum_{j\in N\setminus\{i\}} x_{ji}^{kt} = v_i^{kt}, \quad \forall i \in R, k \in K, t \in T \quad (14)$$

$$\sum_{i\in R} x_{0i}^{kt} \le 1, \quad \forall k \in K, t \in T \quad (15)$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij}^{kt} \le |S| - 1, \quad \forall S \subseteq R, |S| \ge 2, k \in K, t \in T \quad (16)$$

$$p_t \ge 0, \quad \forall t \in T \quad (17)$$

$$w_t \in \{0, 1\}, \quad \forall t \in T \quad (18)$$

$$I_i^{qt} \ge 0, \quad \forall i \in N, q \in Q, t \in T \quad (19)$$

$$d_i^{qt} \ge 0, \quad \forall i \in R, q \in Q, t \in T \quad (20)$$

$$y_i^{qkt} \ge 0, \quad \forall i \in R, q \in Q, k \in K, t \in T \quad (21)$$

$$v_i^{kt} \in \{0, 1\}, \quad \forall i \in R, k \in K, t \in T \quad (22)$$

$$x_{ij}^{kt} \in \{0, 1\}, \quad \forall (i, j) \in A, k \in K, t \in T \quad (23)$$

Objective function (1) minimizes the total cost, in which the first summation denotes the fixed and variable production cost, the second and third ones are total inventory cost and total routing cost, respectively. Objective function (2) maximizes the average food quality level. Note that the smaller the objective value, the higher the average food quality. Constraints (3) are the inventory initialization at the plant and retailers. Constraints (4)-(7) indicate the food flow balance at the plant and retailers. Constraints (8) and (9) are the production and inventory capacity constraints. Constraints (10) indicate that customer demand at each retailer must be satisfied. The vehicle capacity constraint is imposed by (11). Constraints (12) allow positive delivery quantity to a retailer only if it is visited. Constraints (13) forbid split delivery. Constraints (14) correspond to the vehicle flow conservation. Constraints (15) denote that one vehicle can perform at most one route in each period. Constraints (16) eliminate all subtours. Constraints (17)-(23) are nonnegative and integer conditions on decision variables. The considered BFPIRP is NP-hard since it contains a VRP that is well known to be NP-hard [43]. Due to the complexity of the studied problem, a new approach that combines an ϵ-constraint-based two-phase iterative heuristic and a fuzzy logic method is developed in next section.

IV. Solution Method

In this section, we briefly introduce some notation and principles of bi-objective optimization and ϵ-constraint methods. Then we present the ϵ-CTIH to solve the considered BFPIRP. Finally, a fuzzy decision method for selecting a preferred solution is introduced.

A. Bi-Objective Optimization

For the sake of brevity, we introduce bi-objective optimization with the considered BFPIRP, which can be simply

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

represented as follows:

$$\min f_1 = \varphi(\boldsymbol{x}) \tag{24}$$

$$\min f_2 = \omega(\boldsymbol{x}) \tag{25}$$

$$\text{s.t. } \boldsymbol{x} \in \mathcal{X} \tag{26}$$

where $\boldsymbol{x}$ represents a vector of all decision variables; $\varphi(\boldsymbol{x})$ and $\omega(\boldsymbol{x})$ denote cost and quality, respectively. $\mathcal{X}$ is the feasible region of $\boldsymbol{x}$ defined by (3)-(23). A feasible solution $\boldsymbol{x} \in \mathcal{X}$ is said to cover another feasible solution $\boldsymbol{x}' \in \mathcal{X}$ if $\varphi(\boldsymbol{x}) \leq \varphi(\boldsymbol{x}')$ and $\omega(\boldsymbol{x}) \leq \omega(\boldsymbol{x}')$, and $\boldsymbol{x}$ is said to dominate $\boldsymbol{x}'$ ($\boldsymbol{x} \prec \boldsymbol{x}'$) if and only if $\boldsymbol{x}$ covers $\boldsymbol{x}'$ and at least one of the two inequalities is strict. Similarly, $\boldsymbol{x} \in \mathcal{X}$ is said to be non-dominated if there is no feasible solution $\boldsymbol{x}' \in \mathcal{X}$ such that $\boldsymbol{x}' \prec \boldsymbol{x}$. If a feasible solution is non-dominated, then we say that it is Pareto-optimal. Then its corresponding objective values $(\varphi(\boldsymbol{x}), \omega(\boldsymbol{x}))$ form a Pareto point. The Pareto-optimal solution set is defined as $P_s = \{\boldsymbol{x} \in \mathcal{X} | \boldsymbol{x} \text{ is Pareto-optimal}\}$, and the Pareto front is defined as $P_f = \{(\varphi(\boldsymbol{x}), \omega(\boldsymbol{x})) | \boldsymbol{x} \in P_s\}$.

B. ϵ-Constraint Method

The basic idea of an ϵ-constraint method is to transform an initial bi-objective problem (say problem defined by (24)-(26)) into a sequence of mono-objective problems with one principal objective (say minimize $\varphi(\boldsymbol{x})$) by transforming the other objective (say minimize $\omega(\boldsymbol{x})$) as a constraint bounded by a parameter ϵ: $\omega(\boldsymbol{x}) \leq \epsilon$. For a given value of ϵ, the (ϵ-parameterized) mono-objective problem can be written as:

$$\min \{\varphi(\boldsymbol{x}) | \omega(\boldsymbol{x}) \leq \epsilon, \boldsymbol{x} \in \mathcal{X}\} \tag{27}$$

after solving this problem, we obtain a solution $\boldsymbol{x}^*(\epsilon) \in \mathcal{X}$, $f_1(\epsilon) = \varphi(\boldsymbol{x}^*(\epsilon))$ and $f_2(\epsilon) = \omega(\boldsymbol{x}^*(\epsilon)) \leq \epsilon$. It can easily be proved that for any $\boldsymbol{x} \in P_s$, there is an ϵ such that $\boldsymbol{x} = \boldsymbol{x}^*(\epsilon)$. As a consequence, by considering all possible values of ϵ and solving the corresponding mono-objective problems (27), we can generate the set of all Pareto solutions P_s.

For our problem, we transform the second objective (food quality) into a constraint, thus yielding the mono-objective model (27) denoted by FPIRP(ϵ).

In order to construct the set of Pareto-optimal solution, we need to know the set of all possible values of ϵ, which is actually an interval. This interval can be determined by obtaining an ideal point (f_1^I, f_2^I) and a nadir point (f_1^N, f_2^N). The ideal and nadir points define lower and upper bounds on the objective values of Pareto-optimal solutions, respectively [41]. They are obtained by exactly solving the following mono-objective problems:

$$f_1^I = \min\{\varphi(\boldsymbol{x}) | \boldsymbol{x} \in \mathcal{X}\} \tag{28}$$

$$f_2^I = \min\{\omega(\boldsymbol{x}) | \boldsymbol{x} \in \mathcal{X}\} \tag{29}$$

$$f_1^N = \min\{\varphi(\boldsymbol{x}) | \omega(\boldsymbol{x}) = f_2^I, \boldsymbol{x} \in \mathcal{X}\} \tag{30}$$

$$f_2^N = \min\{\omega(\boldsymbol{x}) | \varphi(\boldsymbol{x}) = f_1^I, \boldsymbol{x} \in \mathcal{X}\} \tag{31}$$

the value of ϵ can be bounded by interval $[f_2^I, f_2^N]$. Then a step size Δ should be fixed to explore values of ϵ and form a series of mono-objective problems that are solved to obtain the Pareto front or its approximation. The solution to each mono-objective FPIRP(ϵ) for a given value of parameter ϵ, if not dominated, is a Pareto-optimal solution to the original problem. The objective values of all Pareto solutions $(\varphi(\boldsymbol{x}), \omega(\boldsymbol{x}))$ form a Pareto front.

Ideally, we expect to generate the exact Pareto front. This would require solving a large number of mono-objective problems FPIRP(ϵ) due to the continuous nature of parameter ϵ, which is impractical and unnecessary for decision makers. In addition, it is computationally challenging to optimally solve a complex mono-objective FPIRP(ϵ). In practice, decision makers may expect some representative Pareto solutions within a reasonable amount of computation time. This motivates us to design a new method called ϵ-CTIH to obtain an approximate Pareto front. Method ϵ-CTIH consists of solving a set of $M = \{1, 2, \ldots, |M|\}$ mono-objective FPIRP(ϵ) for $|M|$ values of parameter ϵ. Next, we present a two-phase iterative heuristic (TIH) to solve the m^{th} mono-objective problem FPIRP(ϵ_m) with $m \in M$.

C. Two-Phase Iterative Heuristic (TIH) for FPIRP(ϵ_m)

The key to ϵ-CTIH is to efficiently solve a series of mono-objective FPIRP(ϵ_m) by decomposing it into a lot-sizing subproblem and some independent vehicle-routing problems (VRPs). This decomposition is necessary due to the complexity of FPIRP(ϵ_m) with realistic size. More specifically, FPIRP(ϵ_m) is first decomposed into a variant of lot-sizing problem PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) and VRP($\epsilon_m$), where $\boldsymbol{vc}^m$ is a matrix with entries vc_{it}^m ($i \in R$ and $t \in T$) and $\boldsymbol{\lambda}^m$ is a vector with entries λ_t^m.

PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) determines the quantity of food of each quality level to deliver to each customer in each period without considering the detailed routes of individual vehicles. These delivery quantities are then disaggregated into the quantity to be transported to each retailer in each period by each individual vehicle by solving the vehicle routing problem VRP(ϵ_m). In order that the decomposition leads to a solution close to an optimal one and to ensure existence of a feasible disaggregation, the lot-sizing problem must take into account, in an aggregated way, the delivery cost and the transportation capacity. The delivery cost is calculated by estimating the cost of visiting each customer $i \in R$ in each period $t \in T$, namely vc_{it}^m. The aggregate transportation capacity is calculated by taking into account the fill rate of the fleet in each period $t \in T$. This is done by introducing parameter λ_t^m. On the other hand, in order for the aggregation to be realistic, these two subproblems PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) and VRP($\epsilon_m$) are solved sequentially and iteratively until a near-optimal solution is found. The sets of parameters $\boldsymbol{vc}^m$ and $\boldsymbol{\lambda}^m$ are updated at each iteration using the solution to VRP(ϵ_m) obtained in the previous iteration. The updating of $\boldsymbol{vc}^m$ is described in detail later. Parameter $\lambda_t^m \in (0, 1]$ for each $t \in T$ is initially set to 1, and is decreased when the routing phase is infeasible. If there is no feasible solution to VRP(ϵ_m) for some period t, then λ_t^m is decreased by $\tau \in (0, 1)$. Note that this idea has been used in [16], [18], and [22].

To be specific, PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) is obtained from FPIRP(ϵ_m) by eliminating route-related variables x_{ij}^{kt}'s and constraints (14)-(16) and (23). Vehicle-related index k is dropped from vehicle-related variables y_i^{qkt} and v_i^{kt} by considering an aggregated vehicle with aggregated capacity. PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) can be formulated as follows:

$$\min f_1 = \sum_{t\in T}(a_t p_t + b_t w_t) + \sum_{i\in N}\sum_{q\in Q}\sum_{t\in T} h_i^q I_i^{qt} + \sum_{(i,j)\in A}\sum_{t\in T} vc_{it}^m v_i^t \tag{32}$$

s.t. (3), (8) − (10), (17) − (20), and

$$\sum_{i\in R}\sum_{q\in Q}\sum_{t\in T} d_i^{qt} q \,/ \sum_{i\in R}\sum_{t\in T} D_i^t \le \epsilon_m \tag{33}$$

$$I_0^{qt} = I_0^{q-1,t-1} - \sum_{i\in R} y_i^{qt}, \forall q \in Q\setminus\{0\}, t\in T \tag{34}$$

$$I_0^{0t} = p_t - \sum_{i\in R} y_i^{0t}, \forall t \in T \tag{35}$$

$$I_i^{qt} = I_i^{q-1,t-1} + y_i^{qt} - d_i^{qt}, \quad \forall i\in R, q\in Q\setminus\{0\}, t\in T \tag{36}$$

$$I_i^{0t} = y_i^{0t} - d_i^{0t}, \forall i \in R, t\in T \tag{37}$$

$$\sum_{i\in R}\sum_{q\in Q} y_i^{qt} \le \lambda_t^m |K| V, \forall t\in T \tag{38}$$

$$\sum_{q\in Q} y_i^{qt} \le V v_i^t, \forall i\in R, t\in T \tag{39}$$

$$y_i^{qt} \ge 0, \forall i\in R, q\in Q, t\in T \tag{40}$$

$$v_i^t \in \{0,1\}, \forall i\in R, t\in T \tag{41}$$

In this formulation, decision variable $v_i^t = 1$ if retailer i is visited in period t, and 0 otherwise. Variable y_i^{qt} is the delivery quantity with quality q to retailer i in period t. Other variables keep the same meaning as in BFPIRP. Objective function (32) minimizes the sum of production, inventory and approximate visiting costs. Constraints (33) mean that the average quality is restricted by ϵ_m. Constraints (34)-(37) are similar to (4)-(7). Constraints (38) ensure that the total delivery quantity to all retailers within one period cannot exceed the total vehicle capacity with the expected fill rate. Constraints (39) ensure the delivery quantity to be 0 if a retailer is not visited. Nonnegative and binary variables are defined by (40) and (41). For each iteration of TIH, once PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) is solved, we obtain values of all variables except those related to vehicle routes.

From the obtained solution to PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$), we can form a set of retailers $\mathcal{V}^t$ to be visited in period $t\in T$; i.e., $\mathcal{V}^t = \{i\in R | v_i^t = 1\}$. For retailer $i\in\mathcal{V}^t$, let $\mathcal{D}_i^t$ denote the corresponding delivery quantity; i.e., $\mathcal{D}_i^t = \sum_{q\in Q} y_i^{qt}$. Then solving VRP($\epsilon_m$) is equivalent to solve a series of independent VRPs, one for each period $t\in T$ such that with $\mathcal{V}^t \ne \emptyset$. Such a problem is denoted VRP(t, ϵ_m). By definition, VRP(t, ϵ_m) can be formulated as follows:

$$\min \sum_{(i,j)\in A}\sum_{k\in K} c_{ij} x_{ij}^k \tag{42}$$

$$\text{s.t. } \sum_{i\in R} \mathcal{D}_i^t v_i^k \le V, \quad \forall k\in K \tag{43}$$

$$\sum_{k\in K} v_i^k = 1, \quad \forall i\in\mathcal{V}^t \tag{44}$$

$$\sum_{j\in\mathcal{V}^t\cup\{0\}\setminus\{i\}} x_{ij}^k = \sum_{j\in\mathcal{V}^t\cup\{0\}\setminus\{i\}} x_{ji}^k = v_i^k, \quad \forall i\in\mathcal{V}^t, k\in K \tag{45}$$

$$\sum_{i\in\mathcal{V}^t} x_{0i}^k \le 1, \quad \forall k\in K \tag{46}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij}^k \le |S| - 1, \forall S\subseteq \mathcal{V}^t, |S|\ge 2, k\in K \tag{47}$$

$$v_i^k \in\{0,1\}, \quad \forall i\in\mathcal{V}^t, k\in K \tag{48}$$

$$x_{ij}^k \in\{0,1\}, \quad \forall i\in\mathcal{V}^t, j\in\mathcal{V}^t\setminus\{i\}, k\in K \tag{49}$$

Objective function (42) minimizes the total routing cost. Constraints (43)-(47) are the vehicle routing constraints. Constraints (48) and (49) define binary variables. PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) and VRP($\epsilon_m$) are solved iteratively by TIH. The solutions to these two problems form a feasible solution to FPIRP(ϵ_m).

Then the visiting costs $\boldsymbol{vc}^m$ are updated as follows: if retailer $i\in R$ is visited in period t, $vc_{it}^m = c_{i^-i} + c_{i^+i} - c_{i^-i^+}$, where i^- and i^+ denote the predecessor and successor nodes of i in the solution to the vehicle routing problem VRP(t, ϵ_m) obtained in the previous iteration, respectively. Otherwise, vc_{it}^m is updated as the minimum insertion cost to one of the routes performed in period t. If the method falls into local optima, we introduce a diversification mechanism to restart the algorithm when the incumbent solution is not improved after N^D iterations. The number of iterations of TIH is limited to N^T. Once the algorithm stops, we output the best incumbent solution, its objective value f_1^m, the corresponding value of f_2^m, and the corresponding value of the two sets of parameters $\boldsymbol{vc}^{*m}$ and $\boldsymbol{\lambda}^{*m}$. Algorithm 1 describes the detailed steps of TIH.

In the main loop (line 3 to 19), PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) and VRP($\epsilon_m$) are solved sequentially. A diversification mechanism is introduced to re-start the algorithm (line 15 to 17). Other lines in Algorithm 1 are self-explained.

D. ϵ-Constraint-Based Two-Phase Iterative Heuristic (ϵ-CTIH)

In this subsection, we depict the general framework of ϵ-CTIH to solve BFPIRP. Firstly, objective f_2 is transformed into a constraint with an upper bound ϵ to form FPIRP(ϵ). To determine the range of ϵ, we should first compute the interval $[f_2^I, f_2^N]$. Preliminary tests show that it is very time consuming to exactly solve (28)-(30) . TIH is used to solve (28)-(30) to obtain near-optimal solutions and the corresponding objective values are denoted as f_1^{AI} , f_2^{AI} and f_1^{AN}, respectively. Because we use TIH which is a heuristic to solve FPIRP(ϵ), we can only obtain an approximate value of f_1^I, denoted as f_1^{AI}. Therefore, the constraint $\varphi(\boldsymbol{x}) = f_1^I$ cannot be taken into account to calculate f_2^N. Thus we use an alternative way to obtain the value f_2^{AN} by calculating it directly with (2) based on the obtained solution to (28). The approximate ideal point (f_1^{AI}, f_2^{AI}), approximate nadir point (f_1^{AN}, f_2^{AN}), and interval $[f_2^{AI}, f_2^{AN}]$ of ϵ are formed.

By varying the value of ϵ in $[f_2^{AI}, f_2^{AN}]$, a set of $|M|$ mono-objective FPIRP(ϵ_m) are solved by calling TIH. As mentioned

Algorithm 1 TIH for Solving FPIRP(ϵ_m)

1. Initialize $f_1^m \leftarrow +\infty$, $j \leftarrow 0$
2. Input N^T, N^D, $\epsilon_m, \boldsymbol{\lambda}^m$, τ and $\boldsymbol{vc}^m$
3. **While** $(j < N^T)$ **do**
4. Solve PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) to obtain total production and inventory cost PC , get $\mathcal{V}^t$ and $\mathcal{D}_i^t$
5. Solve VRP(t, ϵ_m) for all $t \in T$ such that $\mathcal{V}^t \neq \emptyset$ to get route $\mathcal{R}^{kt}$, for all $k \in K$, number of routes $\mathcal{N}^t$, and total routing cost RC
6. **If** there exists $\mathcal{N}^t > |K|$ **then**
7. Set $\lambda_t^m = \lambda_t^m - \tau$ and goto step 4
8. **Else if** $PC + RC < f_1^m$ **then**
9. Set $\boldsymbol{vc}^{*m} \leftarrow \boldsymbol{vc}^m$, $\boldsymbol{\lambda}^{*m} \leftarrow \boldsymbol{\lambda}^m$, $f_1^m = PC + RC$
10. Calculate f_2^m with (2) and set $l = 0$
11. **Else**
12. set $l = l + 1$
13. **End if**
14. Update $\boldsymbol{vc}^m$
15. **If** $l \geq N^D$ **then**
16. Diversify by setting $vc_{it}^m \leftarrow \theta(c_{0i} + c_{i0})$ for $i \in R$ and $t \in T$, set $l \leftarrow 0$
17. **End if**
18. Set $j \leftarrow j + 1$
19. **End while**
20. Output $f_1^m, f_2^m, \boldsymbol{vc}^{*m}$ and $\boldsymbol{\lambda}^{*m}$

above, for the m^{th} iteration, TIH consists of sequentially and iteratively solving two subproblems PIDDP($\epsilon_m, \boldsymbol{vc}^m, \boldsymbol{\lambda}^m$) and VRP($\epsilon_m$) to obtain a near-optimal solution to FPIRP(ϵ_m). The sets of parameters $\boldsymbol{vc}^m$ and $\boldsymbol{\lambda}^m$ play a central role in TIH. Good initial values for these parameters may greatly improve the performance of TIH. In ϵ-CTIH, the set of mono-objective FPIRP(ϵ_m) differs from each other only in (33) by changing the value of ϵ with a step size Δ. Therefore, the proper parameter settings of FPIRP(ϵ_m) may also be used in solving FPIRP(ϵ_{m+1}). For this purpose, in addition to outputing the best feasible solution to FPIRP(ϵ_m), TIH outputs the corresponding values of f_2^{*m}, $\boldsymbol{vc}^{*m}$ and $\boldsymbol{\lambda}^{*m}$. In the $(m+1)^{th}$ iteration of ϵ-CTIH to solve FPIRP(ϵ_{m+1}), the initial values are set as follows: $\epsilon_{m+1} = f_2^{*m} - \Delta$, $\boldsymbol{vc}^{m+1} = \boldsymbol{vc}^{*m}$ and $\boldsymbol{\lambda}^{m+1} = \boldsymbol{\lambda}^{*m}$. Note that these settings contain useful information of FPIRP(ϵ_m) to avoid redundant iterations when solving FPIRP(ϵ_{m+1}) and serve as a good starting point for FPIRP(ϵ_{m+1}). Our heuristic links FPIRP(ϵ_m) to FPIRP(ϵ_{m+1}) with ϵ, $\boldsymbol{vc}$, and $\boldsymbol{\lambda}$ while classic ϵ-constraint methods only use ϵ as a link. The general framework of ϵ-CTIH is outlined in Algorithm 2.

E. Selection of a Preferred Solution

As a set S of approximate Pareto solutions numbered from 1 to $|S|$ are obtained, decision makers may need to choose a best-compromised solution according to their preference. We adopt a fuzzy decision method to help them select a preferred solution since such a method can take into account their preference and indicate the optimality degree of the selected solution [44], [45]. For the i^{th} objective and the s^{th}

Algorithm 2 ϵ-CTIH for Solving BFPIRP

1. Initialize $m \leftarrow 0, vc_{it}^0 \leftarrow c_{0i} + c_{i0}, \forall i \in R, \forall t \in T$, $\lambda_t^0 \leftarrow 1, \forall t \in T$
2. Call TIH to solve (28) to get f_1^{AI}, $\boldsymbol{vc^{*0}}$ and $\boldsymbol{\lambda^{*0}}$, and calculate f_2^{AN} with the solution to (28)
3. Call TIH to solve (29) and (30) to get f_2^{AI} and f_1^{AN}
4. Set $\mathcal{A}_F \leftarrow \{(f_1^{AI}, f_2^{AN}), (f_1^{AN}, f_2^{AI})\}$, $\epsilon_m \leftarrow f_2^{AN}$
5. **While** $(\epsilon_m > f_2^{AI})$ **do**
6. Set $m \leftarrow m + 1$, $\epsilon_m \leftarrow \epsilon_{m-1} - \Delta$, $\boldsymbol{vc}^m \leftarrow \boldsymbol{vc}^{*m-1}$, $\boldsymbol{\lambda^m} \leftarrow \boldsymbol{\lambda}^{*m-1}$
7. Call TIH to solve FPIRP(ϵ_m) with $\boldsymbol{vc}^m$ and $\boldsymbol{\lambda^m}$, and output (f_1^m, f_2^m), $\boldsymbol{vc}^{*m}$ and $\boldsymbol{\lambda}^{*m}$
8. Set $\mathcal{A}_F \leftarrow \mathcal{A}_F \cup (f_1^m, f_2^m)$
9. **End while**
10. Remove the dominated points from $\mathcal{A}_F$ and return $\mathcal{A}_F$

solution, where $s \in S$, a linear membership function $\delta_i(f_i^s)$ is defined for each of the two objectives as follows:

$$\delta_i(f_i^s) = \begin{cases} 1, & f_i^s \leq f_i^I \\ \dfrac{f_i^N - f_i^s}{f_i^N - f_i^I}, & f_i^I < f_i^s < f_i^N, i = 1, 2; 1 \leq s \leq S \\ 0, & f_i^s \geq f_i^N \end{cases} \tag{50}$$

where f_i^I and f_i^N denote the lower and upper limits of the i^{th} objective function, respectively; f_i^s is the i^{th} objective value of the s^{th} Pareto solution. The membership degree δ^s of the s^{th} solution is calculated as follows:

$$\delta^s = \sum_{i=1}^{2} w_i \delta_i(f_i^s) \tag{51}$$

where w_i denotes the weight of the i^{th} objective with $\sum_{i=1}^{2} w_i = 1$. It can be determined based on decision makers' preference. The solution with the maximum value of δ^s is selected as their most preferred solution.

Note that this selection process presents some advantages over weighted sum method. Although both approaches require modeling decision makers' preference, the meaning of weights is different. In a weighted sum method, a weight is directly attached to each objective, while these objectives are often of different units and scales. Modeling decision makers' preference among such objectives is challenging. Furthermore, if decision makers change their mind, a new set of weights has to be estimated to reflect their new preference and a new problem has to be solved again. The whole process thus may be very time consuming. In a fuzzy decision approach, however, a weight is attached to the membership degree of each objective and these membership degrees are already normalized to a scalar between 0 and 1 and the set of Pareto optimal solutions is already available at the selection stage. Such weights are easier to estimate in order to model decision makers' preference. More importantly, when decision makers change their mind, it is sufficient to estimate a new set of weights and select a new solution among the set of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Pareto-optimal solutions. This makes the selection process much shorter.

V. Computational Experiments

This section presents the computational experiments conducted on a case study to illustrate the performance of our algorithm and on 185 randomly generated instances to show the effectiveness and efficiency of the proposed approach. We compile the algorithms in C++ by using Microsoft Visual Studio 2010 linked with CPLEX version 12.6.0. All runs are performed on a personal computer with CORE CPU 2.5 GHz and 8 GB RAM.

A. Case Study

A case is derived from a real-life fresh-meat product logistics network. A food company produces and distributes a single type of fresh-meat product to its 40 owned retail stores with 3 homogeneous vehicles. Detailed data of all parameters can be found in the Appendix. Decision makers have to make an integrated plan for the next week (7 days) indicating how much to produce on each day, how much to deliver to each retailer on each day, how to arrange proper routes for the planned deliveries, and at the same time, guide the retailers to sell meat products with different quality to their final customers.

Currently, the company applies a three-phase heuristic to generate the production and distribution plan for the following week. The daily production quantity is first determined based on the demand from each retailer to minimize the production cost by assuming products are stored only at the production site. The distribution plan is then calculated to minimize the total inventory cost. Vehicle routes are generated last based on the planned deliveries. The three-phase heuristic generates a plan with a total cost of 309,260 RMB and an average quality level of 2.00. Note that the quality level is an average value of all product quality sold to final customers. In the case study, the quality level can be interpreted as the average number of days a product has been stored before it is sold.

1) Model Solution: CPLEX is unable to solve the studied case with guaranteed optimality due to its large size. Therefore, we use the proposed model and approach to obtain a set of Pareto solutions for decision makers. The obtained Pareto solutions (dominated solutions have been removed) are shown in Fig. 1, where the horizontal and vertical axes denote the cost and quality, respectively. Having a set of Pareto solutions at hand, we can directly provide all these solutions to decision makers. Besides, we can also help them select a best-compromised solution. As can be seen from Fig. 1 that we can obtain 29 Pareto solutions for the studied case. The computation time is 1741.7s. The obtained Pareto front shows a clear conflict between these two objectives. The two points formed by the approximate lower and upper bounds of the two objectives are shown in solid squares. The solution generated by the currently-used three-phase heuristic is shown as a solid triangle. It is obvious that the solution provided by the currently three-phase heuristic is dominated by some solutions obtained by our heuristic. The solution (f_1^{AI}, f_2^{AN}) obtained

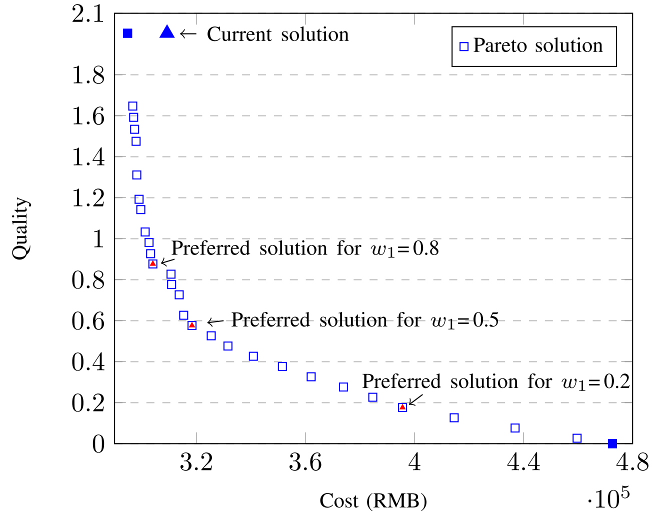

Fig. 1. Approximate Preto front for the case.

by our ϵ-CTIH, as compared to the solution obtained by the three-phase heuristic, can reduce the total cost by 10.77%.

2) Results and Discussion: To help decision makers select their preferred solutions, we assume the decision makers have 3 different preferences based on different situations. In the first situation, they focus more on cost and the weight of the first objective is set to $w_1 = 0.8$. In the second situation, they treat two objectives equally, i.e., $w_1 = 0.5$. While in the last situation, they focus more on quality, i.e., $w_1 = 0.2$. The three selected solutions are shown in Fig. 1, from which we can see that different weights may lead to different Pareto solutions. Detailed results of the selected solutions are shown in Table I, where the first four columns are self-explained. Column Δf_1 gives the cost increase compared with f_1^{AI}; i.e., $\Delta f_1 = (F_1 - f_1^{AI})/f_1^{AI}$ and Δf_2 denotes the quality improvement compared with f_2^{AN}, i.e., $\Delta f_2 = (f_2^{AN} - F_2)/f_2^{AN}$. The next three columns correspond to the production, inventory and transportation costs, respectively. The last column indicates the number of production setups.

TABLE I
Detailed Information for the Selected Solutions

w_1	δ^{max}	F_1	F_2	$\Delta f_1(\%)$	$\Delta f_2(\%)$	$Pcost$	$Icost$	$Tcost$	S
0.8	0.87	304050	0.88	3.14	56.21	281942	2162	19946	3
0.5	0.79	318404	0.58	8.01	71.21	289511	1516	27377	5
0.2	0.82	395631	0.18	34.21	91.19	363580	2614	29437	6

From Table I, we can see that the membership values range from 0.79 to 0.87, which are relatively high. If decision makers focus more on cost, the selected solution shows 3.14% cost increase and 56.21% quality improvement. If they focus more on quality, the selected solution shows 34.21% cost increase and 91.19% quality improvement. We can observe from the last four columns that as they focus more on quality, the production cost increases because the production should be set up more frequently; the transportation cost increases since the products should be delivered to retailers more frequently to ensure quality; the inventory cost decreases since less inventory can be held because of the high quality requirement. Note that the inventory cost increases from 1516 RMB to 2614 RMB when the quality improves from 0.58 to 0.18. This is because the initial inventory can no longer be all used to fulfill the demand when the quality requirement is high, and some of the initial inventory is held till the end of its shelf life. It is also shown in the last column that when the quality level improves from 0.88 to 0.18, the production setup cost doubles.

B. Instance Generation

To thoroughly evaluate the proposed model and solution method, we randomly generate 37 sets of instances with 5 for each, totaling 185 instances. These instances are divided into two groups characterized by the number of retailers, vehicles and periods. The first group of instances, small-size instances, are generated as follows: the number of retailers n is 10, 15, or 20; the length of the planning horizon $|T|$ is 3, 4, or 5 for $n = 10$, 15, or 20 with 1 vehicle; and it is 3, 6, 9, or 12 for $n = 20$ with 2 vehicles. The second group of instances are medium- and large-size which are generated as follows: the number of retailers n is 30, 40, 50, 60, 80, or 100; the length of the planning horizon $|T|$ is 3, 6, 9 or 12; the number of vehicles $|K|$ is $n/10$. For all instances, the number of quality levels $|Q|$ is set as: 2 for $|T| = 3$; 3 for $|T| = 4, 6$; 4 for $|T| = 5, 9$; and 5 for $|T| = 12$. All parameters concerning inventory routing are generated according to the rules of [33]. The initial inventory g_i^q for all $i \in N$ and $q \in Q$ is set to 0. Remaining parameters concerning production are generated based on [46], as stated below: the production capacity C is set to be $\frac{\sum_{i\in R}\sum_{t\in T} D_i^t}{|T|}\beta$, where β is randomly generated from interval [2, 4]; the inventory capacity of the plant U_0 is set to be γC, where γ is randomly generated from interval [1.5, 2]; the unit production cost a_t is randomly generated from interval [4, 7]; the fixed setup cost b_t is set to be μC, where μ is randomly generated from interval [0.3, 0.5].

In TIH, PIDDP($\epsilon_m, \boldsymbol{vc^m}, \boldsymbol{\lambda^m}$) is solved by CPLEX with default settings and a time limit of 10 seconds. VRP(t, ϵ_m) is solved by VRPH package [48]. Parameters N^T and N^D are set to 20 and 5, respectively. The value of θ is randomly generated from [0.5, 1.5] for each $i \in R, t \in T$. The aggregate vehicle capacity parameters λ_t^m and τ are set to 1 and 0.05, respectively. According to our preliminary test, the step size parameter Δ is set to 0.05.

C. Performance Evaluation of ϵ-CTIH

To evaluate the performance of ϵ-CTIH, we compare the approximate Pareto front $\mathcal{A}_F$ obtained by ϵ-CTIH with a reference set $\mathcal{R}_F$ obtained by another ϵ-constraint method (ϵ-CC) that solves the BFPIRP by applying the ϵ-constraint method framework and adopts CPLEX to solve each of the mono-objective problem FPIRP(ϵ_m) instead of TIH. The general framework of ϵ-CC is similar to that in Algorithm 2. However, CPLEX is called to solve FPIRP(ϵ_m) with a time limit of 14400s, i.e., 2 hours and only ϵ is used to link FPIRP(ϵ_m) and FPIRP(ϵ_{m+1}). Since the number of subtour elimination constraints (16) increases exponentially when the number of nodes increases, they are added only when they are violated. In detail, we first solve the full model FPIRP(ϵ_m) with (16) being relaxed. At the end of each iteration, if there exist one or more subtours in the solution, then we add the subtour elimination constraints concerning these subtours. Then the model is re-solved. This process repeats until no subtour can be found. Then we get the optimal solution. If the time limit is reached and there are still subtours in the solution, then CPLEX fails to provide any feasible solution.

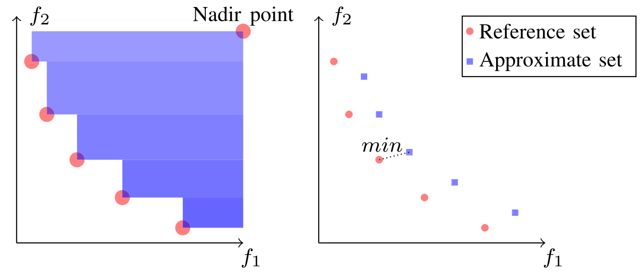

Fig. 2. Hypervolume indicator and e-dominance indicator.

1) Performance Evaluation Metrics: To compare the performance of ϵ-CTIH and ϵ-CC, we use the four widely used indicators [49], namely the cardinality $|\mathcal{A}_F|$ and $|\mathcal{R}_F|$, hypervolume ratio $\mathcal{H}$, average e-dominance $\mathcal{D}$, and computation time $\mathcal{T}$. Note that $\mathcal{A}_F$ and $\mathcal{R}_F$ give the Pareto solution sets obtained by the evaluated method (i.e., ϵ-CTIH) and the reference method (i.e., ϵ-CC), respectively.

Cardinality is the number of solutions in an obtained Pareto solution set. If $|\mathcal{A}_F| > |\mathcal{R}_F|$, we say ϵ-CTIH performs better than ϵ-CC from cardinality perspective.

Hypervolume ratio is the ratio of hypervolumes of $|\mathcal{A}_F|$ and $|\mathcal{R}_F|$ with the denotation of $\mathcal{H}_\mathcal{A}$ and $\mathcal{H}_\mathcal{R}$, respectively. $\mathcal{H}_\mathcal{A}$ indicates the objective space covered by set $\mathcal{A}_F$. As shown in Fig. 2, each Pareto point in the set forms a rectangle shown by the shaded area with respect to a reference point (generally the Nadir Pareto point), and the hypervolume of the solution set is the area of the union of all rectangles. The larger the hypervolume ratio, the better the algorithm. If $\mathcal{H} > 1$, then ϵ-CTIH performs better than ϵ-CC.

Indicator e-dominance denotes the average distance between $\mathcal{A}_F$ and $\mathcal{R}_F$. An approximate Pareto point $(\varphi(\boldsymbol{x}), \omega(\boldsymbol{x})) \in \mathcal{A}_F$ is said to e-dominate $(\varphi(\boldsymbol{x}'), \omega(\boldsymbol{x}')) \in \mathcal{R}_F$ if $\varphi(\boldsymbol{x}) \leq e(\boldsymbol{x}')\varphi(\boldsymbol{x}')$ and $\omega(\boldsymbol{x}) \leq e(\boldsymbol{x}')\omega(\boldsymbol{x}')$, where the e-dominance indicator $e(x')$ for a given Pareto point $((\varphi(\boldsymbol{x}'), \omega(\boldsymbol{x}'))$ is calculated as:

$$e(x') = \min_{(\varphi(\boldsymbol{x}),\omega(\boldsymbol{x}))\in\mathcal{A}_F} max\{\frac{\varphi(\boldsymbol{x})}{\varphi(\boldsymbol{x}')}, \frac{\omega(\boldsymbol{x})}{\omega(\boldsymbol{x}')}\} \tag{52}$$

$e(x') < 1$ indicates that $(\varphi(\boldsymbol{x}'), \omega(\boldsymbol{x}'))$ is dominated by $(\varphi(\boldsymbol{x}), \omega(\boldsymbol{x}))$. The average e-dominance indicator is calculated as:

$$\mathcal{D} = \frac{1}{|\mathcal{R}_F|} \sum_{(\varphi(\boldsymbol{x}'),\omega(\boldsymbol{x}'))\in\mathcal{R}_F} e(x') \tag{53}$$

The closer $\mathcal{D}$ is to 1, the closer$\mathcal{A}_F$ is to $\mathcal{R}_F$.

2) Results for Small-Size Instances: Firstly, we present the computational results on 65 small-size instances in Table II, where the first three columns denote the numbers of retailers,

TABLE II

COMPARISON RESULTS FOR SMALL-SIZE INSTANCES

n	$\|K\|$	$\|T\|$	$\|\mathcal{R}_F\|$	$\|\mathcal{A}_F\|$	$\mathcal{H}$	$\mathcal{D}$	$\mathcal{T}_{\mathcal{R}}$(s)	$\mathcal{T}_{\mathcal{A}}$(s)	$\#\mathcal{T}_{\mathcal{R}}$(s)	$\#\mathcal{T}_{\mathcal{A}}$(s)
10	1	3	10.8	10.2	0.986	1.004	260	25	667	28
		4	12.4	12.2	0.993	1.002	880	38	2899	57
		5	17.4	16.8	0.995	1.001	2325	72	7265	103
15	1	3	10.6	10.6	0.984	1.002	1954	31	4958	39
		4	7.8	12.8	3.532	1.008	10081	48	14400	54
		5	11	12.4	0.990	1.003	6179	65	14400	101
20	1	3	7	10.6	2.197	1.007	2963	31	14400	37
		4	9.6	9.8	1.154	1.003	5779	48	14400	50
		5	6.2	13.8	1.024	1.011	9647	65	14400	95
20	2	3	0	10.4	-	-	14400	36	14400	60
		6	0	13.8	-	-	14400	164	14400	309
		9	0	18	-	-	14400	553	14400	809
		12	0	16.4	-	-	14400	780	14400	918
Average			7.1	12.9	1.428	1.005	4452	150	9754	204

TABLE III

COMPUTATIONAL RESULTS FOR LARGE-SIZE INSTANCES

n	$\|K\|$	$\|T\|$	$\|\mathcal{A}_F\|$	$\mathcal{T}_{\mathcal{A}}$(s)	$\#\mathcal{T}_{\mathcal{A}}$(s)	n	$\|K\|$	$\|T\|$	$\|\mathcal{A}_F\|$	$\mathcal{T}_{\mathcal{A}}$(s)	$\#\mathcal{T}_{\mathcal{A}}$(s)
30	3	3	8.8	33	45	60	6	3	10.8	70	132
		6	14.4	247	421			6	11.6	446	552
		9	16.4	741	1274			9	13.4	1353	2215
		12	16.2	1203	1442			12	21.2	3441	5060
40	4	3	11.6	50	94	80	8	3	9.6	81	160
		6	15.8	356	624			6	10.8	558	659
		9	17.4	1230	1519			9	19.8	2228	3890
		12	16	1424	2038			12	16.4	3125	4078
50	5	3	7.2	42	62	100	10	3	7.8	82	206
		6	12.8	384	419			6	11	626	933
		9	14.8	1238	1508			9	13.8	2288	4050
		12	15.8	1896	2590			12	18.6	3726	5881
Average									13.8	1119	1660

vehicles and periods, respectively. Cardinality $|\mathcal{R}_F|$ and $|\mathcal{A}_F|$ are shown in columns 4 and 5. Indicators $\mathcal{H}$ and $\mathcal{D}$ are represented in columns 6 and 7, respectively. Columns $\mathcal{T}_{\mathcal{R}}$ and $\mathcal{T}_{\mathcal{A}}$ correspond to the average computation times (seconds) of ϵ-CC and ϵ-CTIH, respectively. Each value in column 4 to 9 is an average value over 5 instances with the same size. Finally, the last two columns show the longest computation time for each group of 5 instances by ϵ-CC and ϵ-CTIH, respectively.

We can observe from cardinality columns (columns 4 and 5) that ϵ-CTIH can generate almost as many Pareto solutions as ϵ-CC does for instances with 10 retailers and 1 vehicle, which indicates the performances of ϵ-CTIH are comparable with ϵ-CC. Algorithm ϵ-CTIH performs better than ϵ-CC by providing more Pareto solutions for other instances (instances with 15 retailers or more). It is worth noting that ϵ-CC cannot even obtain a Pareto feasible solution within 14400s for instances with 20 retailers and 2 vehicles. In terms of hypervolume ratio, all values are either close to or greater than 1, which means that ϵ-CTIH performs equally or better than ϵ-CC. Moving to the e-dominance indicator, all values of the e-dominance are very close to 1, which implies that the obtained Pareto solutions by the two methods are quite close to each other. As for the computation time, the average computation times for ϵ-CTIH and ϵ-CC are 150s and 7513 s, respectively. The former needs only 2% computation time of the latter. The same trend can also be observed from the last two columns which show the longest computation time of the two methods. The last results indicate that ϵ-CTIH is relatively stable since the average of the longest computation time over all instances is only 54s longer than that of the average computation time. It shows, however, a great difference between the average computation time and the longest computation time of ϵ-CC. In addition, the computation time of ϵ-CC increases exponentially while that of ϵ-CTIH increases relatively slowly.

3) Results for Medium and Large-Size Instances: To further evaluate the performance of ϵ-CTIH, we next conduct experiments on 120 instances that are considered as medium- and large-size instances of the considered problem. For them, ϵ-CC cannot provide a feasible solution within 14400s. Therefore, we only report the results obtained by ϵ-CTIH and evaluate its performance in terms of cardinality and computation time. Detailed results are given in Table III.

As can be seen from Table III, ϵ-CTIH is able to solve all instances and obtain a reasonable number of approximate Pareto solutions within acceptable computation time. The average number of Pareto solutions obtained is 13.8 and the average and longest computation times are 1119s and 1660s, respectively. It is shown that the computation time of ϵ-CTIH increases quickly with the number of periods $|T|$, and it increases slightly with the number of retailers n. Take instances with 100 retailers as an example. As the number of periods increases from 3 to 12, the average computation time increases from 82s to 3726s. If we fix the number of periods as 3, the average computation time increases from 33s to 82s as the number of retailers increases from 30 to 100. In terms of the longest computation time, some of the instances show a relatively large difference from the average, but the computation time is still acceptable. The longest computation time is 1660s on average over the 12 sets of instances. The computation time varies even for instances with the same size because the number of Pareto solutions obtained is different, which may result in solving different number of mono-objective problems.

VI. CONCLUSION

This paper investigates a new bi-objective production inventory routing problem for an intelligent food logistics system. We first propose a novel bi-objective mixed integer linear programming model for the problem. Then a combined approach is developed to solve it. The approach is evaluated on a case study and on 185 randomly generated instances with up to 100 retailers and 12 periods. Computational results on the case study indicate that the proposed model and method are able to solve a real-world case with 40 retailers and 7 periods. Our approach can provide decision makers with a set of Pareto solutions and help them select a desired solution. Compared with an existing three-phase heuristic, our approach can reduce the total cost by 10.77%. It also shows that a decision maker can greatly improve the average quality provided to final customers with a slight cost increase by using the proposed approach. Computational results on randomly generated instances show the effectiveness and efficiency of the proposed approach.

The model can be extended to multi-product and/or multi-plant settings. More efficient algorithms, e.g., [40]–[43], [47] should be developed such that larger-size instances can be

TABLE IV

DETAILED DATA FOR THE CASE

ID	I_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	ID	I_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
1	177	80	72	64	79	74	79	69	21	171	73	86	79	75	74	80	81
2	188	61	67	73	84	74	80	82	22	144	82	61	79	82	84	82	66
3	163	80	70	72	81	79	65	67	23	174	86	81	86	77	77	78	73
4	151	67	62	80	62	80	70	73	24	169	76	66	90	74	79	78	81
5	174	76	64	78	71	84	80	82	25	154	89	72	72	69	81	74	85
6	185	64	67	64	74	70	67	71	26	161	85	85	90	72	73	76	72
7	154	77	79	73	71	71	77	67	27	173	69	87	65	82	86	67	86
8	183	87	82	70	81	73	72	80	28	158	67	85	77	76	75	77	77
9	187	88	78	75	71	76	64	89	29	168	78	69	74	71	83	65	65
10	170	60	88	68	70	82	63	62	30	157	76	70	76	75	89	70	80
11	154	85	73	76	77	71	63	82	31	148	73	70	73	80	73	75	71
12	178	81	81	77	77	74	76	78	32	178	65	65	77	68	81	64	83
13	184	77	82	76	84	70	71	87	33	181	88	67	82	70	78	73	62
14	147	62	70	73	81	75	78	71	34	168	68	83	78	77	72	75	73
15	146	84	83	90	87	78	80	74	35	185	69	77	82	71	72	70	65
16	147	79	79	77	68	74	75	90	36	180	71	67	73	77	81	64	71
17	177	78	84	75	76	83	87	61	37	176	63	73	77	72	80	78	81
18	155	71	79	63	83	76	64	73	38	177	83	71	62	67	82	66	66
19	143	65	64	75	77	88	74	66	39	167	64	87	76	66	77	70	76
20	178	82	76	74	81	71	74	68	40	188	67	80	64	83	76	66	75

Production Capacity: 60000 Kg;
Inventory capacity at Production site: 40000 Kg;
Initial inventory at production site: 500 Kg;
Vehicle capacity: 6000 Kg;
Fixed production setup cost: 2000 RMB;
Variable production cost: 20 RMB/Kg;
Unit inventory cost at production site: 0.08 RMB/Kg/day;
Unit inventory cost at all retailers: 0.12 RMB/Kg/day;
Unit transportation cost: 30 RMB/Km;
Quality level range: [0, 3].

solved within an acceptable time. In addition, the demand in the food market is often uncertain. Thus the future study may also include demand uncertainty to make the problem more realistic. The simultaneous optimization of food preservation methods and operational planning activities may be conducted, e.g., to include temperature control and packaging selection when planning the food production and distribution activities, because different preservation methods may lead to different food shelf life and cost [50], [51]. Finally, it is meaningful to apply the proposed model to some real world cases.

APPENDIX

See Table IV.

REFERENCES

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[2] T. G. Crainic, M. Gendreau, and J.-Y. Potvin, "Intelligent freight-transportation systems: Assessment and the contribution of operations research," *Transp. Res. C, Emerg. Technol.*, vol. 17, no. 6, pp. 541–557, 2009.

[3] D. McFarlane, V. Giannikas, and W. Lu, "Intelligent logistics: Involving the customer," *Comput. Ind.*, vol. 81, pp. 105–115, Sep. 2016.

[4] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[5] A. M. Campbell and M. W. P. Savelsbergh, "A decomposition approach for the inventory-routing problem," *Transp. Sci.*, vol. 38, no. 4, pp. 488–502, 2004.

[6] P. Chandra and M. L. Fisher, "Coordination of production and distribution planning," *Eur. J. Oper. Res.*, vol. 72, no. 3, pp. 503–517, 1994.

[7] O. Ahumada and J. R. Villalobos, "Application of planning models in the agri-food supply chain: A review," *Eur. J. Oper. Res.*, vol. 196, no. 1, pp. 1–20, 2009.

[8] Y. Li, F. Chu, Z. Yang, and R W. Calvo, "A production inventory routing planning for perishable food with quality consideration," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 407–412, 2016.

[9] Y. Li, F. Chu, C. Chu, W. Zhou, and Z. Zhu, "Integrated production inventory routing planning with time windows for perishable food," in *Proc. IEEE 19th Int. Conf. Intell. Trans. Syst. (ITSC),*, Nov. 2016, pp. 2651–2656.

[10] M. Ruokokoski, O. Solyalı, J.-F. Cordeau, R. Jans, and H. Süral, "Efficient formulations and a branch-and-cut algorithm for a production-routing problem," HEC Montreal, Montreal, QC, Canada, Tech. Rep. G-2010-66, 2010

[11] C. Archetti, L. Bertazzi, G. Paletta, and M. G. Speranza, "Analysis of the maximum level policy in a production-distribution system," *Comput. Oper. Res.*, vol. 38, no. 12, pp. 1731–1746, 2011.

[12] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems," *INFORMS J. Comput.*, vol. 26, no. 1, pp. 103–120, 2014.

[13] Y. Qiu, L. Wang, X. Xu, X. Fang, and P. M. Pardalos, "Formulations and branch-and-cut algorithms for multi-product multi-vehicle production routing problems with startup cost," *Expert Syst. Appl.*, vol. 98, pp. 1–10, May 2018.

[14] M. Boudia, M. A. O. Louly, and C. Prins, "A reactive GRASP and path relinking for a combined production–distribution problem," *Comput. Oper. Res.*, vol. 34, no. 11, pp. 3402–3419, 2007.

[15] M. Boudia and C. Prins, "A memetic algorithm with dynamic population management for an integrated production–distribution problem," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 703–715, 2009.

[16] J. F. Bard and N. Nananukul, "The integrated production–inventory–distribution–routing problem," *J. Scheduling*, vol. 12, no. 3, pp. 257–280, Jun. 2009.

[17] V. A. Armentano, A. L. Shiguemoto, and A. Løkketangen, "Tabu search with path relinking for an integrated production–distribution problem," *Comput. Oper. Res.*, vol. 38, no. 8, pp. 1199–1209, 2011.

[18] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Optimization-based adaptive large neighborhood search for the production routing problem," *Transp. Sci.*, vol. 48, no. 1, pp. 20–45, 2014.

[19] N. Brahimi and T. Aouam, "Multi-item production routing problem with backordering: A MILP approach," *Int. J. Prod. Res.*, vol. 54, no. 4, pp. 1076–1093, 2016.

[20] Q. Zhang, A. Sundaramoorthy, I. E. Grossmann, and J. M. Pinto, "Multiscale production routing in multicommodity supply chains with complex production facilities," *Comput. Oper. Res.*, vol. 79, pp. 207–222, Mar. 2017.

[21] F. Fumero and C. Vercellis, "Synchronized development of production, inventory, and distribution schedules," *Transp. Sci.*, vol. 33, no. 3, pp. 330–340, 1999.

[22] N. Absi, C. Archetti, S. Dauzére-Pères, and D. Feillet, "A two-phase iterative heuristic approach for the production routing problem," *Transp. Sci.*, vol. 49, no. 4, pp. 784–795, 2014.

[23] O. Solyali and H. Süral, "A multi-phase heuristic for the production routing problem," *Comput. Oper. Res.*, vol. 87, pp. 114–124, Nov. 2017.

[24] M. Chitsaz, J.-F. Cordeau, and R. Jans, "A unified decomposition matheuristic for assembly, production and inventory routing," HEC Montréal, Canada, Montreal, QC, Tech. Rep. GERAD G–2017–03, 2017.

[25] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "The production routing problem: A review of formulations and solution algorithms," *Comput. Oper. Res.*, vol. 55, pp. 141–152, Mar. 2015.

[26] H.-K. Chen, C.-F. Hsueh, and M.-S. Chang, "Production scheduling and vehicle routing with time windows for perishable food products," *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2311–2319, 2009.

[27] P. Devapriya, W. Ferrell, and N. Geismar, "Integrated production and distribution scheduling with a perishable product," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 906–916, 2017.

[28] R. K. Ahuja, W. Huang, H. E. Romeijn, and D. R. Morales, "A heuristic approach to the multi-period single-sourcing problem with production and inventory capacities and perishability constraints," *Informs J. Comput.*, vol. 19, no. 1, pp. 14–26, 2007.

[29] P. Amorim, H.-O. Günther, and B. Almada-Lobo, "Multi-objective integrated production and distribution planning of perishable products," *Int. J. Prod. Econ.*, vol. 138, no. 1, pp. 89–101, 2012.

[30] T. Le, A. Diabat, J.-P. Richard, and Y. Yih, "A column generation-based heuristic algorithm for an inventory routing problem with perishable goods," *Opt. Lett.*, vol. 7, no. 7, pp. 1481–1502, 2013.

[31] S. Mirzaei and A. Seifi, "Considering lost sale in inventory routing problems for perishable goods," *Comput. Ind. Eng.*, vol. 87, pp. 213–227, Sep. 2015.

[32] M. Soysal, J. M. Bloemhof-Ruwaard, R. Haijema, and J. G. van der Vorst, "Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty," *Int. J. Prod. Econ.*, vol. 164, pp. 118–133, Jun. 2015.

[33] L. C. Coelho and G. Laporte, "Optimal joint replenishment, delivery and inventory management policies for perishable products," *Comput. Oper. Res.*, vol. 47, pp. 42–52, Jul. 2014.

[34] Y. Qiu, J. Qiao, and P. M. Pardalos, "Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory," *Omega*, 2018, doi: 10.1016/j.omega.2018.01.006.

[35] E. Demir, T. Bektas, and G. Laporte, "The bi-objective pollution-routing problem," *Eur. J. Oper. Res.*, vol. 232, no. 3, pp. 464–478, 2014.

[36] K. Deb, "Multi-objective optimization," in *Search Methodologies*. Berlin, Germany: Springer, 2014, pp. 403–449.

[37] C. C. Coello, G. B. Lamont, D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. Berlin, Germany: Springer, 2007.

[38] G. Tian, M. Zhou, P. Li, C. Zhang, and H. Jia, "Multiobjective optimization models for locating vehicle inspection stations subject to stochastic demand, varying velocity and regional constraints," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1978–1987, Jul. 2016.

[39] G. Tian, Y. Ren, and M. Zhou, "Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution and particle swarm optimization combined algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3009–3021, Nov. 2017.

[40] Q. Kang, S. W. Feng, M. C. Zhou, A. C. Ammari, and K. Sedraoui, "Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary algorithms," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2557–2568, Sep. 2017.

[41] J.-F. Bérubé, M. Gendreau, and J.-Y. Potvin, "An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits," *Euro. J. Oper. Res.*, vol. 194, no. 1, pp. 39–50, 2009.

[42] G. Mavrotas, "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems," *Appl. Math. Comput.*, vol. 213, no. 2, pp. 455–465, 2009.

[43] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Manage. Sci.*, vol. 40, no. 10, pp. 1276–1290, 1994.

[44] M. Esmaili, H. A. Shayanfar, and N. Amjady, "Multi-objective congestion management incorporating voltage and transient stabilities," *Energy*, vol. 34, no. 9, pp. 1401–1412, 2009.

[45] Z. Zhou, F. Chu, A. Che, and M. Zhou, "ε-constraint and fuzzy logic-based optimization of hazardous material transportation via lane reservation," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 847–857, Jun. 2013.

[46] M. Boudia, M. A. O. Louly, and C. Prins, "Combined optimization of production and distribution," in *Proc. CD-ROM Int. Conf. Ind. Eng. Syst. Manage. (IESM)*, Marrakech, Morocco, May 2005.

[47] B. Wang, X. Xia, H. Meng, and T. Li, "Bad-scenario-set robust optimization framework with two objectives for uncertain scheduling systems," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 1, pp. 143–153, Jan. 2017.

[48] C. Groër, B. Golden, and E. Wasil, "A library of local search heuristics for the vehicle routing problem," *Math. Program. Comput.*, vol. 2, no. 2, pp. 79–101, 2010.

[49] T. Okabe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2, Dec. 2003, pp. 878–885.

[50] A. Rong, R. Akkerman, and M. Grunow, "An optimization approach for managing fresh food quality throughout the supply chain," *Int. J. Prod. Econ.*, vol. 131, no. 1, pp. 421–429, 2011.

[51] Q. Li, P. Yu, and X. Wu, "Shelf life extending packaging, inventory control and grocery retailing," *Prod. Oper. Manage.*, vol. 26, no. 7, pp. 1369–1382, 2017.

Yantong Li received the B.S. degree in traffic and transportation from Beijing Jiaotong University, Beijing, China, in 2011, and the M.S. degree in transport planning and management from Military Transportation University, Tianjin, China, in 2013. He is currently pursuing the Ph.D. degree in automation with University of Evry, Université Paris-Saclay, Evry, France.

His research interests include operations research, intelligent logistics systems, and combinatorial optimization.

Feng Chu (M'07–SM'12) received the B.S. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 1986, the M.S. degree in metrology, automatic control, and electrical engineering from National Polytechnic Institute of Lorraine, Lorraine, France, in 1991, and the Ph.D. degree in automatic control, computer science, and production management from University of Metz, Metz, France, in 1995.

She is currently a Full Professor of operations research with University of Evry, Université Paris-Saclay, Evry, France, and a Co-Leader of the Algorithmic, Operations Research, Bioinformatics and Statistical Learning Group, Computing, Integrative Biology, and Complex Systems Laboratory. Her research interests include the modeling, analysis, and optimization of complex systems, such as intelligent transportation systems and logistic and production systems based on combinatorial optimization, operations research, and Petri nets.

Chenpeng Feng received the B.S. degree in information management and information system and the Ph.D. degree in management science and engineering from University of Science and Technology of China, Hefei, China, in 2010 and 2015, respectively.

He is currently a Lecturer with Hefei University of Technology, Hefei. His research interests include performance evaluation and resource allocation.

Chengbin Chu received the B.Sc. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 1985 and the Ph.D. degree in computer science from University of Metz, Metz, France, in 1990.

He was with the National Research Institute in Computer Science and Automation, France, as a Research Officer (chargé de recherche), from 1987 to 1996. He was a Professor with University of Technology of Troyes, France, from 1996 to 2008, where he was also the Founding Director of the Industrial Systems Optimization Laboratory. From 2008 to 2017, he held a Chair Position in supply chain management with the CentraleSupélec, Université Paris-Saclay, France, sponsored by multinational companies. He is currently with ESIEE, Université Paris-Est, France. He is interested in research areas related to operations research and modeling, analysis, and optimization of supply chain and production systems.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of electrical and computer engineering.

His interests include Petri nets, intelligent automation, Internet of Things, big data, Web services, and intelligent transportation. He was a recipient of the Humboldt Research Award for U.S. Senior Scientists from the Alexander von Humboldt Foundation, the Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society.

He is a Life Member of the Chinese Association for Science and Technology, USA, and served as its President in 1999. He is a Fellow of International Federation of Automatic Control, American Association for the Advancement of Science, and the Chinese Association of Automation.