



European J. of Industrial Engineering

ISSN online: 1751-5262 - ISSN print: 1751-5254

<https://www.inderscience.com/ejie>

Lagrangian-based solutions for the multi-level production-inventory problem in iron and steel production with reverse logistics

Guoli Liu

DOI: [10.1504/EJIE.2025.10063416](https://doi.org/10.1504/EJIE.2025.10063416)

Article History:

Received:	06 February 2023
Last revised:	11 February 2024
Accepted:	25 February 2024
Published online:	02 June 2025

Lagrangian-based solutions for the multi-level production-inventory problem in iron and steel production with reverse logistics

Guoli Liu

School of Mechanical Engineering,
Shenyang University of Technology,
No.111, Shenhao West Road,
Shenyang 110870, China
Email: lg_li1978@126.com

Abstract: This research deals with the production-inventory problem originating from the ironmaking production system in Shanghai Baoshan Iron and Steel Complex. A mixed integer programming (MIP) model based on the minimisation of total related costs including production/purchasing costs, inventory costs and setup costs is proposed to determine the production and inventory quantities of all materials in each time period under material-balance and capacity constraints. To solve the problem, a decomposition approach based on Lagrangian relaxation (LR) is developed. A solution property is introduced to speed up the solving process. Heuristic strategies are applied to improve the upper bound. In order to further improve the quality of the solutions, an alternative Lagrangian relaxation algorithm based on variable splitting is derived. A detailed numerical evaluation based upon the actual production data from Baosteel is performed. The computational results reveal that the proposed algorithms can obtain good quality solutions within a reasonable time. [Submitted: 6 February 2023; Accepted: 25 February 2024]

Keywords: combinatorial optimisation; Lagrangian relaxation; reverse logistics; production-inventory planning.

Reference to this paper should be made as follows: Liu, G. (2025) 'Lagrangian-based solutions for the multi-level production-inventory problem in iron and steel production with reverse logistics', *European J. Industrial Engineering*, Vol. 19, No. 4, pp.502–538.

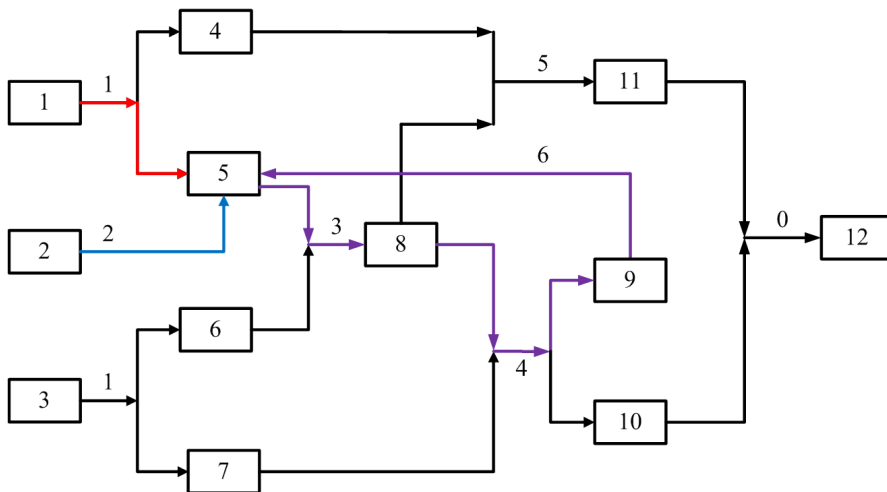
Biographical notes: Guoli Liu received her PhD in Systems Engineering from the Northeastern University, China. She is currently a Professor at the Shenyang University of Technology, China. Her current research interests include combinatorial optimisation theory and its applications in operations management and operations research. She has published several academic articles in *Applied Mathematics and Computation*, *European Journal of Operational Research*, *Journal of the Operational Research Society*, and *International Journal of Operational Research*, among others.

This paper is a revised and expanded version of a paper entitled 'Model and solution for the multilevel production-inventory system before ironmaking in Shanghai Baoshan Iron And Steel Comple', presented at The 2006 International Conference in Computational Science and Its Applications, Glasgow, UK, 8–11 May 2006.

1 Introduction

Since carrying inventories for various reasons can cost steel companies anywhere from 20% to 40% of their value a year, the top management of many companies has recognised the critical importance of integrating production and inventory operations. Effective multi-level production-inventory planning that helps companies improve profitability by simultaneously optimising production and inventory quantities is therefore vital and becomes the focus of research efforts. In this paper, we investigate the problem of making production and inventory decisions for ironmaking materials which should be manufactured through several stages in Baosteel, and then find that it is virtually a multi-item multi-level capacitated lot-sizing problem with reverse logistics that combines features from both assembly and chemical production systems. The problem aims to minimise the sum of setup, production and inventory holding costs on the condition that the given demand in each time period is fulfilled without backlogging. The importance of production-inventory planning and the increasing concern about reverse logistics motivate this study.

Figure 1 An example of special product structures with divergent, convergent and cyclic features (see online version for colours)



However, making an efficient production-inventory plan for this production system is no easy task. Chen and Thizy (1990) proved that the multi-item capacitated lot-sizing problem is strongly NP-hard. Given that the multi-item capacitated lot-sizing problem is a special case of the problem at hand, it becomes evident that the problem we are dealing with is NP-hard. Furthermore, the fact that demands of intermediate products and recyclable quantities of by-products are dependent on the decisions for production quantities at succeeding stages makes the problem even more challenging. Additionally, the special product structure with splitting, mixing, and recycling features introduces further complexity. Splitting is divergent, meaning that one product is decomposed into at least two others. Mixing is convergent, implying that one product is made by some others. Recycling is cyclic, indicating that one product is recovered for reuse. An example of such special structures is presented in Figure 1, where different production

routings of each material are pointed out and the operation numbers are given over the line. In the figure, it is evident that material 5 can be directly provided by processing materials 1, 2 and 9, respectively. Additionally, the cycle composed of materials 5, 8 and 9 is essentially a typical reverse logistics that represents the process by which organisations recover by-products and residuals for reuse, resale, remanufacturing, recycling or disposal (Johnson, 1998).

This paper makes three significant contributions. Firstly, a more practical and applicable model for capacitated lot-sizing problems is introduced. At present, most related research simplifies the actual production process to achieve optimal solutions under various theoretical conditions. As a result, those findings are usually difficult to directly apply to guide actual production. Secondly, an under-researched reverse logistics problem in steel production planning is investigated. This problem really exists in the actual steel production process, and its impact on the cost and profit of enterprises cannot be ignored. Thirdly, effective modelling and solution methods are designed for the capacitated lot-sizing problem with splitting, mixing, and recycling features in product structure, which can find high-quality solutions in a reasonable time.

The remainder of this paper is organised as follows. Section 2 reviews several related works as well as the current state of research concerning production planning with reverse logistics. Section 3 discusses the production-inventory problem that originates from Baosteel, and formulates a novel mixed integer programming model. To solve the problem, section 4 proposes a decomposition LR algorithm and a heuristic strategy. Section 5 reconsiders the problem based on variable splitting and introduces another heuristic strategy to improve the quality of the solutions further. The performance of all the algorithms is evaluated on a set of 1,750 randomly generated test problems of sizes varying between 15 and 75 items and with 6 to 30 time periods in the planning horizon. The computational results are reported. After the analysis of the results, Section 6 concludes the paper and presents a discussion of future work.

2 Literature review

For the reason of cost saving and environmental protection, production planning with reverse logistics has attracted considerable attention from both academics and industry in recent years. Various quantitative models with deterministic demands occurred. For example, Shih (2001) proposed a MIP model to minimise the total cost, which consists of transportation cost, operating cost, fixed cost for new facilities, final disposal cost and landfill cost, as well as the sale revenue of reclaimed materials. Hu et al. (2002) presented a discrete-time linear analytical model to minimise the total operational cost of a multi-time-step, multi-type hazardous-waste reverse logistics system. Dobos (2003) represented the optimal inventory problem in a reverse logistics system as an optimal control model with two state variables and three control variables. Kim et al. (2006) proposed a MIP model to maximise the total cost savings by optimally deciding the quantity of parts to be processed at each remanufacturing facilities, the number of purchased parts from subcontractor. Demirel and Gökçen (2008) developed a MIP model for a remanufacturing system including multi-phase and multi-product forward and reverse distribution. Mutha and Pokharel (2009) established an integer programming model for the design of a reverse logistics network. Sasikumar et al. (2010) provided a mixed integer nonlinear programming model for a multi-echelon reverse logistics

network to realise the maximisation of its profit. Das and Chowdhury (2012) investigated an integrated, reverse logistics supply chain planning process with modular product design and formulated it as a MIP model. Lee et al. (2013) formulated a facility location problem in an integrated forward and reverse distribution network as a nonlinear bi-objective model to minimise the total shipping cost and time simultaneously. Subulan and Tasan (2013) developed a nonlinear mixed integer programming model for tactical planning in a conceptual closed-loop supply chain with remanufacturing option. Garcia-Sabater et al. (2013) introduced a novel formulation technique for mathematically modelling the relationships between operations and materials, utilising the concept of 'stroke'. Mar-Ortiz et al. (2014) proposed a mathematical programming model based on the concept of dedicated cells and a variable neighbourhood search algorithm for the disassembly cell formation problem with demand variability. Bing et al. (2014) analysed the strategic alternatives of reverse logistics network design by scenario study approach. Bouras (2015) adopted optimal control theory to derive optimal production rate, optimal remanufacturing rate, as well as advertisement rate and optimal disposal rate in a reverse supply chain. Kilic et al. (2015) created a mixed integer linear programming model with consideration of recycling rates for each product category to determine optimum locations and flows for 10 scenarios with different collection rates. Govindan et al. (2016) put forward a multi-objective mixed integer mathematical problem for a generic closed loop supply chain network. Yu and Solvang (2016) presented a multi-objective mixed integer programming model for network design of a general four echelon reverse logistics system. Demirel et al. (2016) developed a mixed integer linear programming model to optimise the costs associated with opening facilities, recovery processes, transportation of end-of-life vehicles and its components/materials through the network and revenues associated with selling reusable/remanufacturable parts and scrap metals. Hong et al. (2016) used game theory to optimise pricing, advertising, and used-product collection strategies for a two-level hybrid manufacturing-remanufacturing closed-loop supply chain. Bouras et al. (2016) adopted optimal control theory to find out the explicit expression of the optimal production rate, the optimal remanufacturing rate, and the optimal disposal rate in a reverse supply chain. Taskhiri et al. (2016) proposed a mixed-integer linear programming model for a transportation problem to minimise the logistics cost for wood flow, including the harvesting, chipping, storage (drying) and transportation costs. Cunha and Melo (2016) presented a multicommodity reformulation and a Wagner-Whitin-based formulation with new valid inequalities for the economic lot-sizing problem with remanufacturing. Jung et al. (2016) studied a supply planning problem in a manufacturing system to minimise the total cost at the remanufacturer. Bazan et al. (2016) provided an extensive literature review on the mathematical modelling of inventory with return flows that were developed in economic order/production quantity and joint economic lot size settings. Kannan et al. (2017) designed a mixed integer nonlinear programming model for the planning of reverse logistics networks from the perspective of the third party while integrating the disassembly line balancing in the recovery process. Cunha et al. (2017) used an effective reformulation based on valid inequalities to solve the multi-item economic lot-sizing problem with remanufacturing and uncapacitated production. Moreover, Panagiotidou et al. (2017) studied the problem of optimising the returns classification process under partial quality information in a hybrid manufacturing/remanufacturing facility. Desport et al. (2017) proposed a mixed-integer programming model and a metaheuristic approach

for a tactical inventory planning problem concerned with optimally planning supplies and repairs based on demand forecasts and in the face of conflicting business objectives. Suzanne et al. (2020) proposed a comprehensive state-of-the-art review around the topic of circular economy and reverse logistics with a particular emphasis on mid-term production planning under discrete time settings. Reddy et al. (2022) investigated the multi-facility green reverse logistics network design problem. Chen and Lin (2022) devised a production-inventory model for a closed-loop supply chain to jointly determine sale and collection prices and production time points. Ghanbarzadeh-Shams et al. (2022) introduced a novel fuzzy multi-objective programming model with deterministic demands and chance constraints to address the production planning problem in the carpet industry, aiming to optimise both total cost efficiency and sustainability attributes.

Another stream of research in the literature is dedicated to providing efficient approximate algorithms. In this regard, we refer the readers to the following articles. Pishvaei et al. (2010) applied a simulated annealing algorithm with dynamic neighbourhood search mechanisms to find near optimal solutions for the proposed model designed to minimise the transportation and fixed opening costs in a multistage reverse logistics network. Lee et al. (2015) devised a multi-objective hybrid genetic algorithm using a priority-based chromosome and adaptive weight approach to solve a reverse logistics network problem. Wang et al. (2016) employed a new cross-entropy-based solution methodology to solve the NP-hard problem of designing and planning a closed-loop supply chain. Sahling (2016) proposed a new model formulation subject to emission constraints for an integrated production and remanufacturing planning problem with vendor selection and devised a combination of column generation and a period-oriented fix-and-relax heuristic, as well as an adapted fix-and-optimize heuristic to solve it. Jing et al. (2016) designed an approach based on self-adaptive genetic algorithm to solve a multi-factory production planning problem with remanufacturing and back-ordering. Fang et al. (2016) established an integrated three-stage model based on the internet of things technology for acquisition process, production planning with returned product recovery, and raw material procurement. An efficient particle swarm optimisation algorithm based on two heuristic methods was proposed to solve the problem. Jiang et al. (2016) introduced a dual-objective remanufacturing process optimisation method to maximise the reliability and minimise process cost for the first time, where a genetic algorithm was applied. Jauhar and Pant (2016) made a comprehensive survey on the application of genetic algorithms on supply chain management with reverse logistics. Guo et al. (2017) developed a two-stage forward/reverse logistics network and route planning model, and validated its effectiveness using genetic and particle swarm optimisation algorithms. Coelho and Mateus (2017) proposed a Benders' decomposition algorithm and a combination of Benders' method and tests for fixing variables in a single two-steps algorithm for a capacitated plant location problem in reverse logistics. Kumar et al. (2017) put forward a model of a multi-period, multi-echelon, vehicle routing, forward-reverse logistics system to maximise the total expected profit, and resolved it using artificial immune system and particle swarm optimisation algorithms. Qiu et al. (2018) developed a solution method of branch-and-cut guided search algorithm for a production routing problem with reverse logistics and remanufacturing. Barzinpour and Taki (2018) designed artificial immune system algorithm based on CLONALG, genetic and memetic algorithms for solving a capacitated single-objective single-product mathematical model developed for a dual channel supply chain network. Chung et al. (2018) developed a modified genetic

algorithm with crossing date heuristic to maximise the reuse of packaging and simultaneously minimise the total operating cost. Khorshidian et al. (2019) utilised a hybrid of the improved version of the augmented e-constraint method and TOPSIS to solve the bi-objective model formulated for truck scheduling and transportation planning in a forward/reverse logistics network. Chan et al. (2020) proposed a single-vendor multi-buyer synchronised cycles model with reverse logistics and developed a genetic algorithm to solve this synchronised supply chain problem. Afra and Behnamian (2021) designed Lagrangian heuristic algorithms for solving the green multi-product production routing problem with reverse logistics and remanufacturing. In addition, Dumez et al. (2023) designed a decomposition-based matheuristic to solve the 2-echelon multi-trip vehicle routing problem with capacitated satellites and reverse flows. Zhou et al. (2023) designed a multi-echelon reverse logistics network for collecting eco-packages and developed a hybrid genetic algorithm-tabu search to efficiently solve the resulting capacity location-routing problem. Tasoglu and Ilgin (2024) proposed a simulation-based genetic algorithm approach for the joint optimisation of reverse logistics network design and disassembly line balancing with sequencing decisions. Lei et al. (2024) presented a mixed-integer linear programming model for a collection-disassembly-delivery problem in a reverse logistics network and improved the two-phase iterative heuristic approach and adaptive large neighbourhood search heuristic to solve the problem.

Although a lot of researches concerning reverse logistics have been conducted, most of them only focus on the remanufacturing problems in external supply chains. The reverse logistics problems in internal production systems, especially in intricate iron and steel production system, are seldom studied. Different from the existing literatures, in this paper, we investigate a real production-inventory problem with reverse logistics originating from an ironmaking production system in Baosteel and formulate it as a deterministic mixed integer programming model in consideration of practical constraints. Considering the large-scale variables and constraints in the model, an efficient Lagrangian relaxation approach is designed. Illustrative numerical examples based upon the actual production data from Baosteel are given to demonstrate the effectiveness of the proposed approach. The details are explained below.

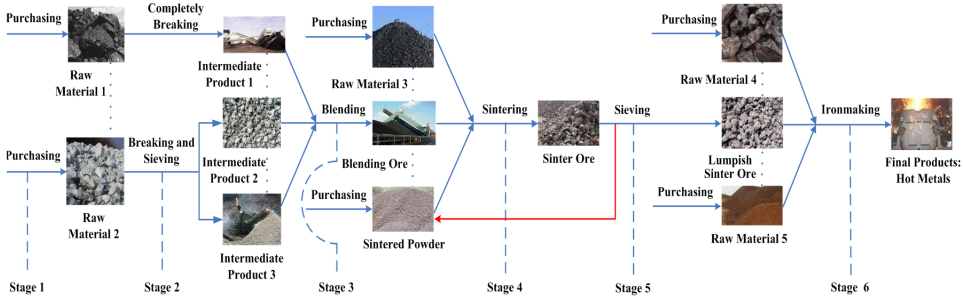
3 Preliminaries and mathematical formulation of the problem

3.1 Problem description

After an investigation on the multilevel production-inventory system before ironmaking in Baosteel has been made, we summarise the process of making hot metals for the steel plant as six stages which is illustrated in Figure 2. A brief description of the process is as follows. At stage 1, raw materials are purchased. At stage 2, the purchased raw materials are processed on breaking operation, sieving operation or completely breaking operation. The intermediate products obtained are directly used as the input materials of ‘blending’ at stage 3. After ‘blending’ and ‘sintering’, one of the main ironmaking materials, sinter ore, is produced at stage 4. In order to improve the quality of the hot metals, the sinter ore is disposed by ‘sieving’ at stage 5. During this course, a useable by-product, sintered powder, is generated and usually recovered to use in stage 3. It means that the sintered powder used in stage 3 can be supplied by both purchasing and reverse logistics. At stage 6, raw materials and intermediate products are processed by ‘ironmaking’ and

finally converted into final products, ‘hot metals’. Notice that, here, every raw material and every intermediate product can be kept temporarily in its work-in-process storage, which incurs an inventory cost. But final products cannot be stored.

Figure 2 The flowsheet of the ironmaking production system in Baosteel (see online version for colours)



3.2 Model formulation

3.2.1 Definitions

1 Raw material:

Raw materials stand for the items that can be purchased from external markets.

2 Final product:

One item is called final product only if it has external demands.

3 Intermediate product:

Intermediate products refer to the items that can be produced by others, excluding final products.

4 Operation:

An operation refers to a production facility, i.e., a machine or a processor or a production line. To facilitate modelling, purchasing and reverse supplying are also regarded as operations in this paper.

5 Divergent operation:

The operations that convert raw materials into multiple products are defined as divergent operations and are classified by the number of the materials used in this process. The set of divergent operations is therefore divided into two parts, i.e., the set of one-to-many operations and the set of many-many operations.

6 One-to-many operation:

The operation that needs only one kind of raw material to produce multiple products is regarded as a one-to-many operation. See Figure 3(a) as an example.

7 *Many-many operation:*

The operations that process multiple raw materials into multiple products are considered as many-many operations. See Figure 3(b) as an example.

8 *Convergent operation:*

The operation that converts raw materials into a unique product is defined as a convergent operation. The set of convergent operations is separated into two subsets, i.e., the set of one-one operations and the set of many-to-one operations, based on the number of the materials to be processed.

9 *One-one operation:*

The operation that belongs to the set of convergent operations and needs only one kind of raw material is viewed as a one-one operation. See Figure 4(a) as an example.

10 *Many-to-one operation:*

The operations that belong to the set of convergent operations and need multiple raw materials are viewed as many-to-one operations. See Figure 4(b) as an example.

11 *Core product and by-product:*

The core product represents the only chief product of a one-to-many or many-many operation, while by-products correspond to the subordinate products. Notice that, for each operation, the yield of the core product and its by-products is in fixed proportions.

Figure 3 Examples of divergent operations, (a) one-to-many (b) many-many (see online version for colours)

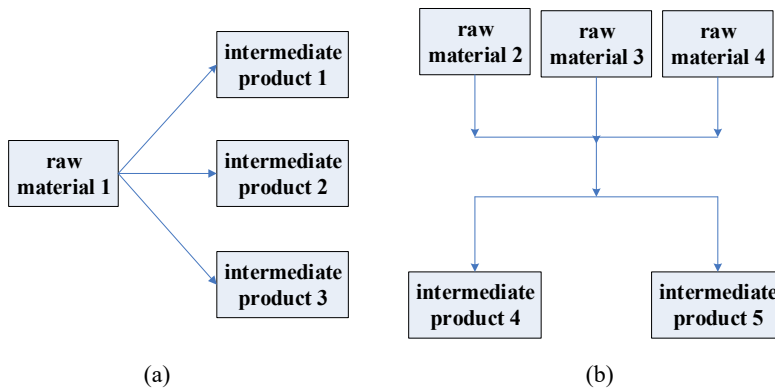
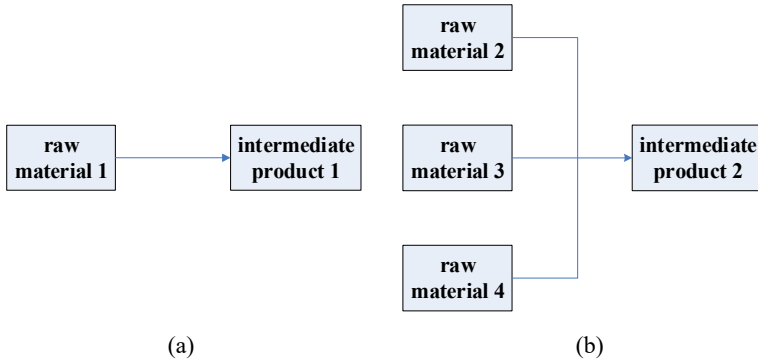


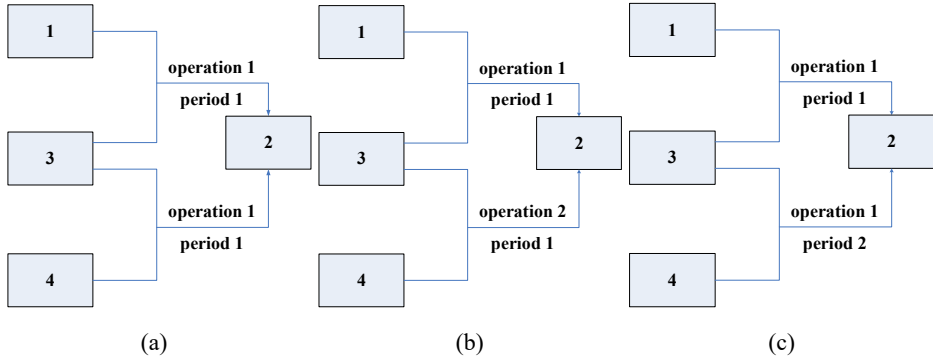
Figure 4 Examples of convergent operations, (a) one-to-one (b) many-to-one (see online version for colours)



3.2.2 Assumptions

- Production of a given item in any given time period incurs a production cost and a fixed setup cost.
- JIT production is required for each final product, which means that backordering is not allowed and no inventories of final products are permitted.
- In each time period, intermediate products are permitted to be produced at different operations and allowed to reversely supply other items.
- Different recipes could be adopted to produce an item at one operation but in different time period. That is, the recipe for producing an item in one time period at an operation must be unique. Figure 5 is given as an example.
- The external demands of final products are positive real numbers which are given and change with time.
- The operation of purchasing is considered as a one-one operation.
- Final products are not allowed to reversely supply other items.
- The costs related to reverse supplying are neglected here.
- In each time period, there exists only one reverse supplying operation in which one item is provided by another.
- No capacity limit for reverse supplying.
- The initial inventory level of each item is set to be zero.
- 'Ironmaking' is the only operation used to directly produce final products.

Figure 5 Explanation of assumption 4, (a) not allowed (b) allowed (c) allowed (see online version for colours)



3.2.3 Parameters

- K The number of stock yards.
- T Set of indices for time periods, i.e., the decision horizon.
- N Set of indices for items in the production process, excluding final products.
- N_r Set of raw materials, $N_r \subset N$.
- N_I Set of intermediate products, $N_I \subset N$.
- N_f Set of final products.
- L_1 Set of divergent operations in L , including one-to-many and many-many operations.
- L_2 Set of convergent operations in L , including one-one and many-to-one operations, $L_2 \cap L_1 = \Phi$.
- L Set of all operations, excluding reverse supplying and ironmaking, $L = L_1 + L_2$.
- L_p The operation standing for purchasing, $L_p \subset L_2$.
- L_r The operation standing for reverse supplying.
- L_f The operation standing for ironmaking.
- A_l^t Set of items which can be processed at operation l in period t , i.e., the material set of operation l in period t , for $l \in L, t \in T$.
- F_l^t Set of core products of operation l in period t , for $l \in L_1, t \in T$.
- B_{il}^t Set of by-products corresponding to core product i of operation l in period t , for $l \in L_1, t \in T, i \in F_l^t$.
- J_l^t Set of the products of operation l in period t , for $l \in L_2, t \in T$.
- d_i^t Demand for final product i in period t , for $i \in N_f, t \in T$.

- S_i^t Set of immediate successors of item i in forward logistics in period t , for $i \in N$, $t \in T$.
- h_i The holding cost for one unit of item i in a period, for $i \in N$.
- E_i^t Set of immediate predecessors of item i in forward logistics in period t , for $i \in N$, $t \in T$.
- pr_i^t Set of operations that can provide item i in period t in forward logistics, including purchasing and ironmaking, for $i \in N \cup N_f$, $t \in T$, where $pr_j^t = \{L_f\}$, $j \in N_f$, $t \in T$.
- rs_i Set of items that could be provided by item i via operation L_r , for $i \in N$, where $rs_j = \Phi$, $j \in N_r$.
- H_{ij}^t Set of operations that can use item i to produce item j in period t in forward logistics, for $i \in N$, $j \in S_i^t$, $t \in T$, i.e., $H_{ij}^t = \{l \mid l \in pr_j^t, j \in S_i^t, i \in N, t \in T\}$, where $H_{ij}^t = \{L_f\}$, for $j \in N_f$, $i \in E_j^t$, $t \in T$.
- bk_i Set of items which could provide item i via operation L_r , for $i \in N$.
- ld_{ij} The lead time of item j provided by item i via operation L_r , for $i \in N$, $j \in rs_i$.
- LI_i^t Set of items that could be provided by item i via operation L_r in period t and used within the decision horizon, i.e., $LI_i^t = \{j \mid j \in rs_i, t + ld_{ij} \in T, i \in N\}$.
- ρ_{ij} The total quantity of item i used to provide one unit of item j , for $i \in N$, $j \in rs_i$.
- sc_{il}^t The setup cost for the production/purchasing of item i at operation l in period t , for $i \in N$, $l \in L$, $t \in T$.
- pc_{il}^t The unit production/purchasing cost for item i at operation l in period t , for $i \in N$, $l \in L$, $t \in T$.
- r_{ijl}^t The quantity of item i needed per unit production of item j at operation l in period t , for $l \in L_1$, $t \in T$, $i \in A_l^t$, $j \in F_l^t \cap S_i^t$.
- $r_{ijl}^t = 0$, for $l \in L_1$, $t \in T$, $i \in A_l^t$, $k \in F_l^t$, $j \in B_{kl}^t \cap S_i^t$.
- r_{ijl}^t The quantity of item i needed per unit production of item j at operation l in period t , for $l \in L_2$, $t \in T$, $i \in A_l^t$, $j \in S_i^t$.
- r_{ijl}^t The corresponding quantity of item j produced per unit production of item i at operation l in period t , for $l \in L_1$, $t \in T$, $i \in F_l^t$, $j \in B_{il}^t$.
- r_{i,j,L_f}^t The quantity of item i needed per unit production of final product j at operation L_f in period t , for $j \in N_f$, $t \in T$, $i \in E_j^t$.

P_{it}	The price of raw material i in period t , for $i \in N_r, t \in T$.
p_{il}^t	The capacity absorption coefficient of item i in period t for operation l , where $p_{i,L_p}^t = P_{it}$, for $l \in L_2, t \in T, i \in J_1^t$.
p_{il}^t	The capacity absorption coefficient of item i in period t for operation l , for $l \in L_1, t \in T, i \in F_l^t$.
Bg_t	The purchasing budget in period t , for $t \in T$.
cap_l^t	The capacity limit for operation l in period t , where $cap_{L_p}^t = Bg_t$, for $l \in L, t \in T$.
V_k	The largest inventory capacity of stock yard k , for $k = 1, \dots, K$.
U_k	Set of the items that should be stored in yard k , for $k = 1, \dots, K$.
M	A large enough positive number.

3.2.4 Decision variables

x_i^t	The production quantity of item i in period t , for $i \in N, t \in T$.
I_i^t	The inventory level of item i at the end of period t , for $i \in N, t \in T$.
z_{il}^t	The production quantity of item i at operation l in period t , for $i \in N \cup N_f, t \in T, l \in pr_i^t$.
y_{il}^t	The binary variable that is one when the production of item i has been set up in period t at operation l , for $i \in N, t \in T, l \in pr_i^t$.
w_{ij}^t	the total quantity of item j , which is provided by item i via operation L_r and can be used in period t , for $i \in N, j \in rs_i, t \in T$.

The lot-sizing problem in this multi-level production system is to determine the lot sizes for all materials which satisfy the external demands for final products. Using the above definitions, the problem can now be described as a MIP model. The details are presented below.

3.2.5 Mathematical formulation

3.2.5.1 The objective function

The objective function expressed by the following equation is to minimise the sum of setup, production/purchasing and inventory holding costs for all items over all time periods.

$$\begin{aligned}
(\text{CP}) \text{ Minimise } C, \text{ with } C \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\
& + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\
& + \sum_{i \in N} \sum_{t \in T} \frac{1}{2} \cdot h_i \cdot (I_i^{t-1} + x_i^t + I_i^t)
\end{aligned} \tag{1}$$

3.2.5.2 Constraints

a Material balance constraints

The dependent demands are fulfilled from inventory, production and reverse logistics, which is guaranteed by the following equations:

$$I_i^{t-1} + x_i^t - I_i^t - \sum_{j \in S_i^t} \sum_{l \in H_{ij}^t} r_{ijl}^t \cdot z_{jl}^t - \sum_{j \in LI_i^t} \rho_{ij} \cdot w_{ij}^{t+ld_{ij}} = 0, \quad \text{for } i \in N, t \in T \tag{2}$$

Each item may be supplied by multiple operations including producing, reverse supplying and purchasing, which can be represented as:

$$\sum_{l \in pr_i^t} z_{il}^t + \sum_{j \in bk_i} w_{ji}^t = x_i^t, \quad \text{for } i \in N, t \in T \tag{3}$$

The independent demands of final products in the decision horizon are satisfied by introducing the undermentioned constraints.

$$z_{j, pr_j^t}^t = z_{j, L_f}^t = d_j^t, \quad \text{for } j \in N_f, t \in T \tag{4}$$

Each core product and its by-products are produced in proportion, which can be represented as:

$$z_{jl}^t = r_{ijl}^t \cdot z_{il}^t, \quad \text{for } l \in L_1, t \in T, i \in F_l^t, j \in B_{il}^t \tag{5}$$

b Variable consistency constraints

The limitations set by the following inequalities are appended to assure the incidence of a setup cost when a lot size is produced.

$$z_{il}^t \leq M \cdot y_{il}^t, \quad \text{for } i \in N, t \in T, l \in pr_i^t \tag{6}$$

c Production/purchasing capacity constraints

The available capacity used for production/purchasing is limited and therefore not allowed to exceed its upper bound. That can be described as:

$$\sum_{i \in F_l^t} p_{il}^t \cdot z_{il}^t \leq cap_l^t, \quad \text{for } l \in L_1, t \in T \tag{7}$$

$$\sum_{i \in J_l^t} p_{il}^t \cdot z_{il}^t \leq cap_l^t, \quad \text{for } l \in L_2, t \in T \tag{8}$$

d *Inventory capacity constraints*

To avoid saturation of depots, the inventory quantities of the materials are restricted by:

$$\frac{1}{2} \cdot \sum_{i \in U_k} (I_i^{t-1} + x_i^t + I_i^t) \leq V_k, \quad \text{for } k = 1, \dots, K, t \in T \quad (9)$$

e *Value range constraints*

The value ranges of the continuous and binary decision variables are respectively defined as follows:

$$x_i^t, I_i^t \geq 0, \quad \text{for } i \in N, t \in T \quad (10)$$

$$z_{il}^t \geq 0, \quad \text{for } i \in N \cup N_f, t \in T, l \in pr_i^t \quad (11)$$

$$w_{ij}^t \geq 0, \quad \text{for } i \in N, j \in rs_i, t \in T \quad (12)$$

$$y_{il}^t \in \{0, 1\}, \quad \text{for } i \in N, t \in T, l \in pr_i^t \quad (13)$$

Based on the analysis of constraints (2), a property is proposed to tighten the lower bound of problem (CP):

Property 1: The production of item i must be set up in period t at operation l_1 , i.e., $y_{i l_1}^t = 1$, if the following conditions are satisfied:

- 1 Item i is the only immediate predecessor of item j in period t in forward logistics, i.e., $E_j^t = \{i\}$.
- 2 Item i can only be produced on operation l_1 in period t in forward logistics, i.e., $pr_i^t = \{l_1\}$.
- 3 Item j can only be produced on operation l in period t in forward logistics, i.e., $pr_j^t = \{l\}$.
- 4 Item i cannot be provided by operation L_r , i.e., $bk_i = \Phi$.
- 5 The inventory level of item i at the end of period $t - 1$ is zero, i.e., $I_i^{t-1} = 0$.
- 6 The production quantity of item j at operation l in period t is larger than zero, i.e., $z_{jl}^t > 0$.

Proof: According to the definition of r_{ijl}^t , we have that $r_{ijl}^t = 0$ if and only if $l \in L_1$, $i \in A_i^t$, $k \in F_i^t$, $j \in B_{kl}^t \cap S_i^t$. Otherwise, $r_{ijl}^t > 0$. In light of the fixed proportional relationship between the yields of core products and its by-products, if $z_{jl}^t > 0$ for $l \in L_1$, $i \in A_i^t$, $k \in F_i^t$, $j \in B_{kl}^t \cap S_i^t$, it implies that $z_{kl}^t > 0$ for $l \in L_1$, $k \in F_i^t$. Therefore, in any case, the following result holds:

$$\sum_{j \in S_i^t} \sum_{l \in H_{ij}^t} r_{ijl}^t \cdot z_{jl}^t = r_{ijl}^t \cdot z_{jl}^t + r_{ikl}^t \cdot z_{kl}^t + \sum_{j' \in S_i^t \setminus \{j, k\}} \sum_{l \in H_{ij}^t} r_{ij'l}^t \cdot z_{j'l}^t > 0.$$

According to the definitions of parameters and decision variables, we have that

$$I_i^t \geq 0, \sum_{j \in LI_i^t} \rho_{ij} \cdot w_{ij}^{t+ld_{ij}} \geq 0,$$

By (2) and 5, we get that

$$x_i^t = I_i^t - I_i^{t-1} + \sum_{j \in S_i^t} \sum_{l \in H_{ij}^t} r_{ijl}^t \cdot z_{jl}^t + \sum_{j \in LI_i^t} \rho_{ij} \cdot w_{ij}^{t+ld_{ij}} > 0.$$

By 2 and 4, obviously, we have that

$$\sum_{l \in pr_i^t} z_{il}^t = z_{ih}^t, \sum_{j \in bk_i} w_{ji}^t = 0.$$

Consequently, by (3), we obtain that

$$z_{ih}^t = \sum_{l \in pr_i^t} z_{il}^t = x_i^t - \sum_{j \in bk_i} w_{ji}^t = x_i^t > 0.$$

By (6), we have that $y_{ih}^t = 1$. □

Given that Property 1 can improve the efficiency of solution methodology and tighten the lower bound, a series of effective algorithms integrating Property 1 have been designed to provide high quality solutions for problem (CP) within a reasonable time. The details are expatiated in what follows.

4 Lagrangian relaxation

Due to their inherent combinatorial complexity, mixed integer programming problems are typically challenging to solve. It is well-known that Lagrangian relaxation is an efficient method to obtain near-optimal solutions for such problems. Representative studies include the works of Diabat et al. (2013, 2015) and Hamdan and Diabat (2020). Therefore, Lagrangian relaxation is employed in this research. In this approach, coupling constraints are first relaxed and introduced into the objective function weighted by Lagrangian multipliers. The resulting relaxed problem is easier to solve than the original one. Based on the optimal solution of the relaxed problem, specialised heuristics are then employed to generate a feasible solution satisfying all constraints. Subsequently, the multipliers are updated using subgradient optimisation. The procedure iterates until further improvement is unlikely. During these iterations, the dual function is maximised (minimised), and its objective function values provide lower bounds (upper bounds) for the original minimisation (maximisation) problem.

With careful investigations on the nature of the production-inventory problem, a solution strategy composed of Lagrangian relaxation, linear programming and heuristics is developed. The details for the strategy are elaborated below.

The decomposition approach for the production-inventory problem considered in this paper hinges on the observation that when we disregard constraints (6), the Lagrangian relaxation problem can be decomposed into easily solvable subproblems for each kind of variable. With these facts, we relax constraints (6) by means of Lagrangian multipliers $u_{il}^t > 0$. The Lagrangian relaxed problem then reads:

$$\begin{aligned}
 (\mathbf{LR}) \text{ Minimize } C_{LR}, \text{ with } C_{LR} \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\
 & + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\
 & + \sum_{i \in N} \sum_{t \in T} \frac{1}{2} \cdot h_i \cdot (I_i^{t-1} + x_i^t + I_i^t) \\
 & + \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_i^t} u_{il}^t \cdot (z_{il}^t - M \cdot y_{il}^t)
 \end{aligned} \tag{14}$$

Subject to constraints (2)–(5), (7)–(13).

We decompose the relaxed problem into two smaller subproblems, each involving one type of variable. The subproblems are:

$$\begin{aligned}
 (\mathbf{LR}_1) \text{ Minimise } C_{LR_1}, \text{ with } C_{LR_1} \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} sc_{il}^t \cdot y_{il}^t + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} sc_{il}^t \cdot y_{il}^t \\
 & - M \cdot \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_i^t} u_{il}^t \cdot y_{il}^t
 \end{aligned} \tag{15}$$

Subject to constraints (13).

$$\begin{aligned}
 (\mathbf{LR}_2) \text{ Minimise } C_{LR_2}, \text{ with } C_{LR_2} \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} pc_{il}^t \cdot z_{il}^t + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} pc_{il}^t \cdot z_{il}^t \\
 & + \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_i^t} u_{il}^t \cdot z_{il}^t \\
 & + \sum_{i \in N} \sum_{t \in T} \frac{1}{2} \cdot h_i \cdot (I_i^{t-1} + x_i^t + I_i^t)
 \end{aligned} \tag{16}$$

Subject to constraints (2)–(5), (7)–(12).

The Lagrangian dual of problem (CP), denoted by **LD**, becomes:

$$(\mathbf{LD}) \text{ Maximise } C_D(u_{il}^t), \text{ with } C_D(u_{il}^t) \equiv \min C_{LR} \tag{17}$$

Subject to constraints (2)–(5), (7)–(13) and

$$u_{il}^t \geq 0, \quad \text{for } i \in N, t \in T, l \in pr_i^t \tag{18}$$

In view of the above models, a Lagrangian relaxation algorithm outlined below is designed to solve the production-inventory problem studied in this paper.

In the first step, the subproblems of the relaxed problem, i.e., **LR**₁ and **LR**₂, are optimally solved for a given set of multipliers. This yields an optimal solution to the relaxed problem, which is generally infeasible for problem (CP) because of the violation

of constraints (6). In the second step, a heuristic method constructs a feasible plan based on the relaxed solution. The value of the dual function serves as a lower bound to the optimal feasible cost, while the objective value of the feasible solution supplies an upper bound. The relative duality gap between the best upper bound and lower bound provides a measure to judge the solution quality. In the third step, new multiplier values are generated following standard conventions for subgradient optimisation in an attempt to eliminate any constraint violations in the next iteration. A new relaxed problem with these updated multipliers is then constructed. This completes one iteration. In our implementation, this process continues until the maximum iteration number or the minimum duality gap is reached. The solution of the subproblems, the construction of feasible solutions, and the subgradient algorithm are explained in detail in the following sections.

4.1 Subproblem solution

$$\begin{aligned}
 (\mathbf{LR}_1) \text{ Minimise } C_{LR_1}, \text{ with } C_{LR_1} &\equiv \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} sc_{il}^t \cdot y_{il}^t + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} sc_{il}^t \cdot y_{il}^t \\
 &\quad - M \cdot \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_i^t} u_{il}^t \cdot y_{il}^t \\
 &= \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} (sc_{il}^t - M \cdot u_{il}^t) \cdot y_{il}^t \\
 &\quad + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} (sc_{il}^t - M \cdot u_{il}^t) \cdot y_{il}^t \\
 &\quad - M \cdot \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in B_{il}^t} u_{il}^t \cdot y_{il}^t
 \end{aligned} \tag{19}$$

Subject to constraints (13).

Recall that y_{il}^t is a zero-one variable. A simple but fast polynomial algorithm is applied to solve the subproblem optimally.

Step 1 Compute the coefficient of each variable.

Step 2 For each variable y_{il}^t , if its coefficient is positive, set $y_{il}^t = 0$; otherwise, let $y_{il}^t = 1$.

As noted previously, problem $(\mathbf{LR}_2)$ is a typical linear programming model. So, it can be directly solved by the standard linear programming software package OSL.

The results obtained from the above provide the basis for the construction of a feasible solution to the original problem, the topic discussed in the following part.

4.2 Construction of a feasible solution to the original problem

Because the discrete decision variables and the continuous decision variables are calculated separately in different subproblems, the consistency among them defined by constraints (6) is generally violated. To construct a feasible solution based on the optimal solution of problem $(\mathbf{LR}_2)$, a heuristic method is adopted to restore feasibility. The algorithmic steps of the heuristics are:

- Step 1 Take the optimal solution of problem **(LR₂)** as the initial solution. For each $i \in N, t \in T, l \in pr_i^t$, set $y_{il}^t = 0$.
- Step 2 Search for the operation that satisfies the condition: $z_{il}^t = 0$, where $i \in N, t \in T, l \in pr_i^t$. Set $y_{il}^t = 1$.
- Step 3 If there exists no such operation, go to Step 4; otherwise, return to Step 2.
- Step 4 Stop. A feasible solution $(I_i^t, x_i^t, z_{il}^t, y_{il}^t, w_{ij}^t)$ is obtained.

4.3 Heuristic strategy 1

If the current optimal solution is not improved during the last two iterations, a heuristic strategy will come into play. The heuristic strategy is developed based on the best solution and the best Lagrangian multipliers. The basic steps are:

- Step 1 Initialisation: set $flag = 1$.
- Step 2 In the current optimal solution, search for an unselected item that has been produced in two consecutive time periods. If there exist no such item, set $flag = 0$, go to Step 7.
- Step 3 In the latter time period, if the setup cost of the selected item is higher than its holding cost, set the corresponding production quantity to zero. Otherwise, return to Step 2.
- Step 4 Introduce the constraint that sets the production quantity of the selected item to zero in the latter time period into problem **(LR₂)**, thereby forming a tighter problem **(LR₃)**.
- Step 5. Use standard software package OSL to solve problem **(LR₃)**. If the problem has no feasible solution, set $flag = 0$, go to Step 7.
- Step 6. Construct a feasible solution based on the solution obtained from Step 5.
- Step 7 Stop. If $flag = 1$, a feasible solution $(I_i^t, x_i^t, z_{il}^t, y_{il}^t, w_{ij}^t)$ is obtained. If the objective value is less than the upper bound, update the current optimal solution and the upper bound. If $flag = 0$, the heuristic is not effective for this problem.

4.4 Subgradient algorithm (**LR_p**) for solving problem (**LD**)

The most popular method for optimising the Lagrangian dual problem has been subgradient optimisation. It is commonly used to solve non-differentiable problems by taking a fixed step size in the direction of a subgradient. In order to solve the Lagrangian dual problem with respect to (17), a subgradient method is constructed in what follows. Here $t_{(m)}$ is the step size for updating the multipliers, and $\beta_{(m)}$ is a factor for adjusting the step size at the m^{th} iteration to update the Lagrangian multipliers. The procedure for the subgradient algorithm is:

Step 1 *Initialise*

- 1 Use Property 1 to determine the values of specific variables and introduce them as equality constraints into the original problem (CP), forming a new problem (CP1).
- 2 Perform the linear relaxation of problem (CP1) by relaxing every binary variable, y_{il} , as a continuous variable between 0 and 1.
- 3 Solve the linear relaxed problem. Let C_1 denote the optimal objective value.
- 4 Use the heuristics introduced in Subsection 4.2 to construct a feasible solution to the original problem based on this LP relaxed solution. Denote the corresponding objective value as C_2 .
- 5 Set $m = 0, flag = 0; C^U = C_2; C^L = C_1; u_{il}^{t(0)} = 0$, for $i \in N, t \in T, l \in pr_i^t$, where m is the iteration index, $\{u_{il}^{t(m)}\}$ represents the non-negative Lagrangian multipliers at the m^{th} iteration, C^U and C^L are the upper bound and lower bound of function C , respectively.

Step 2 *Solve Lagrangian relaxed problem*

- 1 For given values of the Lagrangian multipliers, the optimal solution to the relaxed problem can be calculated by optimally solving all the subproblems using Property 1 and the heuristic algorithms described in Section 4.1.
- 2 If the optimal objective value $C_D(u_{il}^{t(m)}) > C^L$, let $C^L = C_D(u_{il}^{t(m)})$.

Step 3 *Find a feasible solution*

- 1 Based on the solution of Step 2, a feasible solution to the original problem is constructed by using the heuristics presented in subsection 4.2.
- 2 If the objective value $C^{(m)} < C^U$, set $C^U = C^{(m)}, flag = 0$; otherwise, let $flag = flag + 1$.

Step 4 *Improve the upper bound*

- 1 If $flag \geq 2$, heuristic strategy I is called.
- 2 If the upper bound is updated, set $flag = 0$.

Step 5 *Convergence check*

If one of the following conditions is satisfied, then stop; otherwise go to Step 6.

- 1 $(C^U - C^L)/C^U < \zeta$, where $\zeta > 0$ is a very small number.
- 2 $m >$ the maximum iteration number.

Step 6 *Update the Lagrangian multipliers and return to Step 2*

In iterative steps, the multipliers are updated according to the following formulas which result in a new set of multipliers $\{u_{il}^{t(m+1)}\}$:

- 1 $\theta_{il}^{t(m)}(u_{il}^{t(m)}) = z_{il}^t - M \cdot y_{il}^t$, for $i \in N, t \in T, l \in pr_i^t$
- 2 $t_{(m)} = \beta_{(m)} \cdot (C^U - C_D(u_{il}^{t(m)})) / \|\theta^{(m)}(u_{il}^{t(m)})\|^2$
- 3 $u_{il}^{t(m+1)} = \text{Max}\{0, u_{il}^{t(m)} + t_{(m)} \cdot \theta_{il}^{t(m)}(u_{il}^{t(m)})\}$, for $i \in N, t \in T, l \in pr_i^t$

- 4 $\eta_{(m)} = 0.15 \cdot [m/20]$
- 5 $\beta_{(m+1)} = \beta_{(m)} \cdot \exp(-0.5 \cdot \eta_{(m)}^2)$.

4.5 Experiment A

In this experiment, we consider the ironmaking production system in Baosteel described in Figure 2. The algorithms described in previous sections are coded in C++ language and implemented on a personal computer with 16 GB of RAM and an Intel Core I7-4600U CPU at 2.10 GHz.

Based on the real production data from Baosteel, we randomly generate 1750 test instances. The largest scale of these instances corresponds to the product structure network shown in Figure 6, which reflects the maximum scale observed in actual production process. To investigate the impact of product structure complexity on algorithm efficiency, we further construct six additional product structures with varying degrees of complexity, as depicted in Figures 7–12. The primary parameters of the problem instances are generated as follows.

- 1 The number of items in N : ranging from 15 to 75, i.e., $|N| \in \{15, 25, 35, 45, 55, 65, 75\}$.
- 2 The number of the periods in the decision horizon T : ranging from 6 to 30, i.e., $|T| \in \{6, 12, 18, 24, 40\}$.
- 3 The setup cost for the production/purchasing of item i at operation l in period t : sc_{il}^t is randomly selected from one of the following ranges: $R_1 = [30,000, 1,000,000]$, $R_2 = [100,000, 800,000]$, $R_3 = [800,000, 1,500,000]$, $R_4 = [1,500,000, 2,200,000]$, or $R_5 = [2,200,000, 2,700,000]$.
- 4 The unit purchasing cost for item i at operation l in period t : $pc_{il}^t \in [100, 300]$.
- 5 The unit production cost for item i at operation l in period t : $pc_{il}^t \in [10, 50]$.
- 6 Demand for final product i in period t : $d_i^t \in [207, 604, 316, 333]$.
- 7 The holding cost for one unit of item i in a period: $h_i \in [8, 25]$.
- 8 The inventory level of item i at the end of period t : $I_i^{t-1} = 0$.
- 9 The large enough positive number: $M = \sum_{i \in N_f} \sum_{t \in T} 5 \cdot d_i^t$.
- 10 The quantity of item i needed per unit production of item j at operation l in period t : $r_{ijl}^t \in [0, 3]$.
- 11 The largest inventory capacity of stock yard k : $V_k \in [\sum_{t \in T} \max\{d_i^t\}, 10 \cdot \sum_{t \in T} \max\{d_i^t\}]$.
- 12 The capacity limit for operation l in period t : $cap_l^t \in [100, 10 \cdot \sum_{t \in T} \max\{d_i^t\}]$.

From the above, it is evident that there are 35 combinations of N and T . For each such combination, we generate 10 instances. In the stopping criterion, the small number, ζ , is set to 0.001 and the maximum iteration number is set to 100. Given that Lagrangian relaxation algorithms cannot guarantee the attainment of optimal solutions, the relative dual gap $(C^{UB} - C^{LB})/C^{UB}$ is used as a measure of solution optimality, where C^{UB} is the best upper bound of the original problem and C^{LB} is the best lower bound. Optimal performance and running times of algorithm $\mathbf{LR}_P$ against different problem sizes are reported in Tables 1 and 2, respectively. From the computational results, the following observations can be made.

- 1 The average duality gap is 4.981326% and the average running time is 100.3489 seconds. The fact indicates that the algorithm can obtain near optimal solutions in a short computation time for all cases and therefore has a good prospect of application.
- 2 The duality gaps increase when setup costs increase, but these duality gaps are relatively stable when the number of materials or periods increases. The average growth rate of the duality gaps for different ranges of setup costs = $((1.969007 - 0.198590) + (5.135075 - 1.969007) + (7.397384 - 5.135075) + (10.20658 - 7.397384))/4 = (10.20658 - 0.198590)/4 = 2.5019975$.
- 3 When the number of materials or the number of periods increases, the problem size becomes larger and the computational time increases significantly. Because the number of the variables involved in the problem increases and the capacity constraints are tighter, which make the problem more difficult to solve.
- 4 When the setup costs are relatively small, they have little influence upon the objective function. Hence, the problem (CP) can be approximately optimally solved.

To further reduce the dualities, variable splitting is adopted to reanalyse the problem. Full details of the analysis are given in the next part.

Figure 6 The product structure with the largest size ($|N| = 75$)

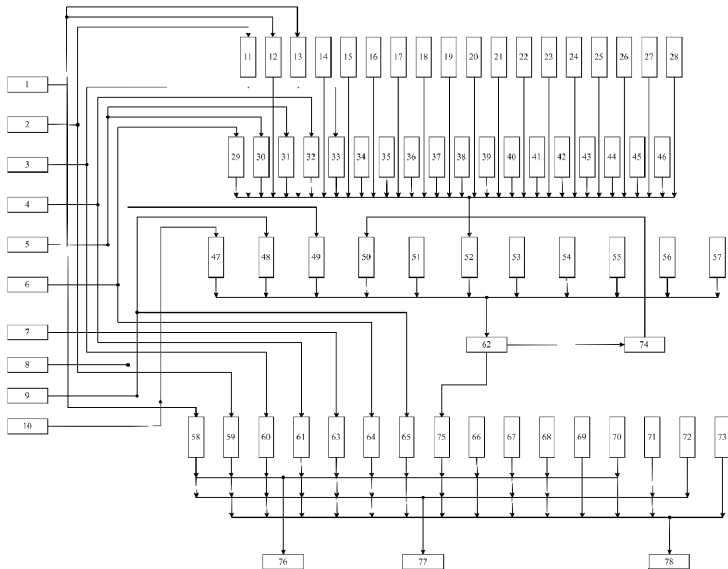


Figure 7 The product structure for $|N| = 15$

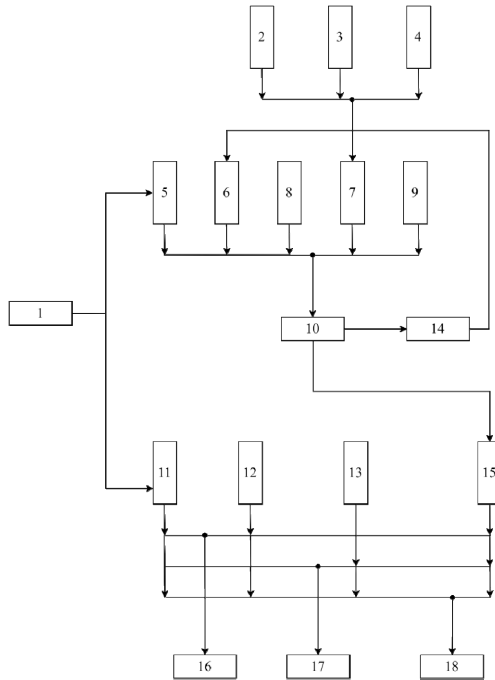


Figure 8 The product structure for $|N| = 25$

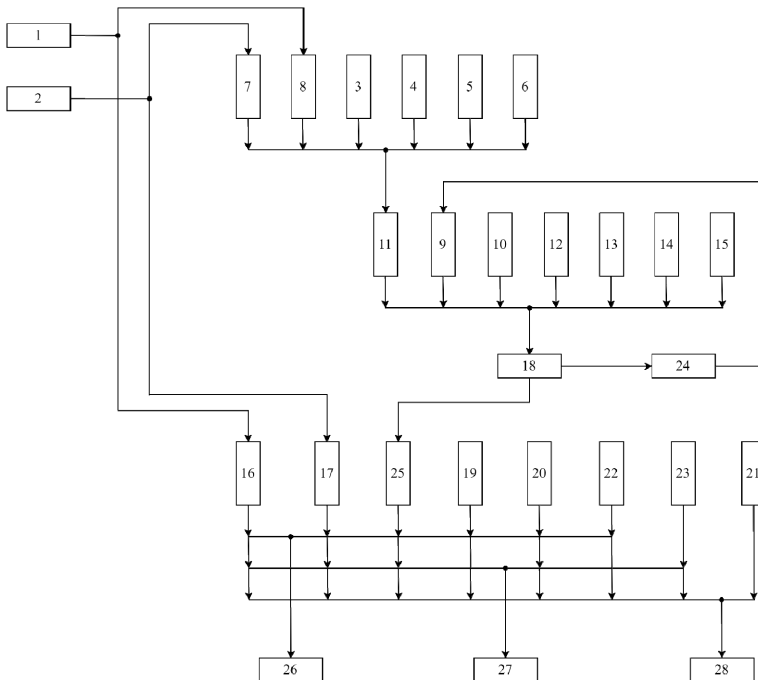


Figure 9 The product structure for $|N| = 35$

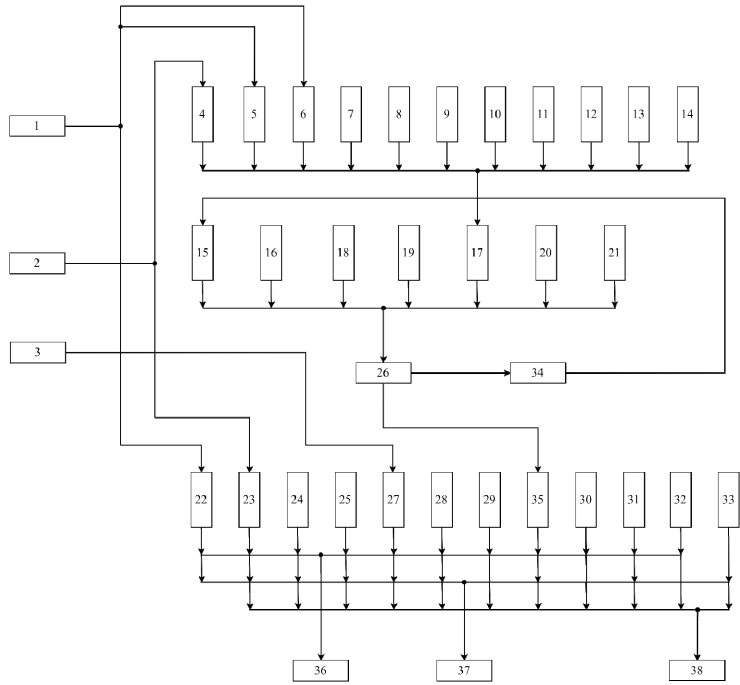


Figure 10 The product structure for $|N| = 45$

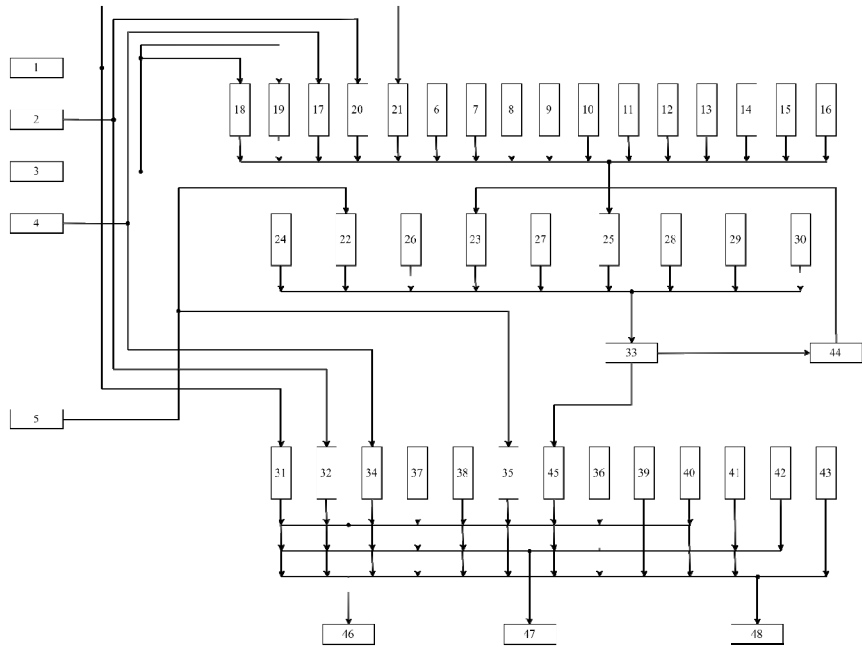


Figure 11 The product structure for $|N| = 55$

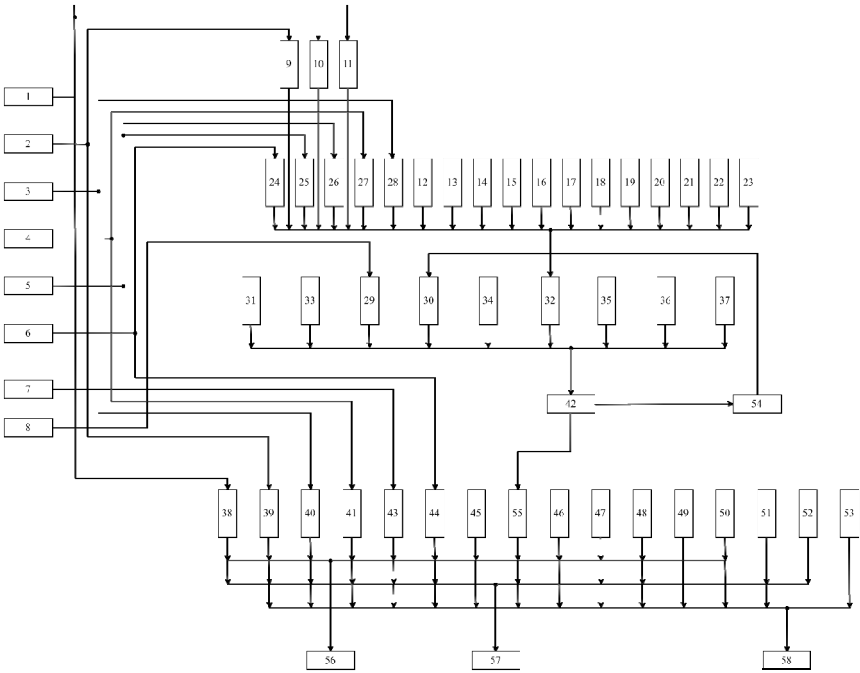


Figure 12 The product structure for $|N| = 65$

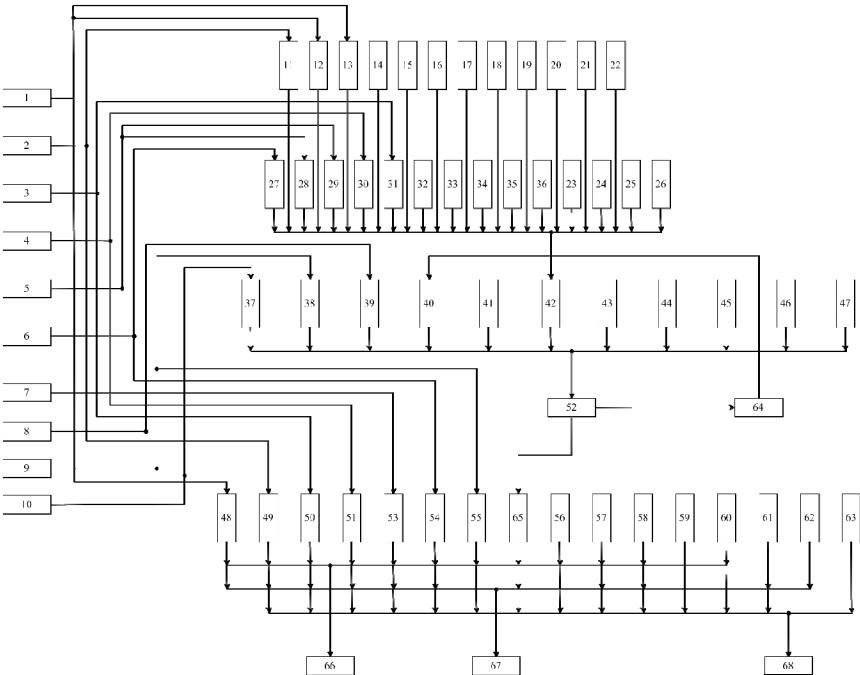


Table 1 Average duality gaps (%) of algorithm $\mathbf{LR}_P$

<i>Problem no.</i>	<i>Problem size</i> $ N \times T $	<i>Average duality gap (%)</i>				
		R_1	R_2	R_3	R_4	R_5
1	15×6	0.202432	1.900283	5.068862	6.979276	7.903041
2	25×6	0.235261	1.942138	4.862419	7.277542	9.720938
3	35×6	0.266746	2.743013	6.891615	8.083761	14.935796
4	45×6	0.221956	1.840793	5.659024	7.898050	11.376373
5	55×6	0.204960	2.336833	5.772908	8.332396	11.269117
6	65×6	0.193610	2.544253	6.787778	7.852786	12.177459
7	75×6	0.291707	2.553905	7.510904	8.321829	13.346272
8	15×12	0.194167	1.904848	4.571696	7.372943	9.467672
9	25×12	0.138393	1.347915	3.423497	6.121500	8.562933
10	35×12	0.237446	1.800828	5.995831	7.000875	11.404983
11	45×12	0.184779	1.878629	4.775632	7.580312	10.275945
12	55×12	0.221300	2.314160	5.394546	8.572676	10.876347
13	65×12	0.241116	2.278456	5.944181	8.932985	11.443790
14	75×12	0.138685	2.349057	5.965682	8.272121	11.972971
15	15×18	0.250522	2.072056	5.191539	8.396502	9.867016
16	25×18	0.175642	1.382180	3.680568	5.645686	7.771206
17	35×18	0.232610	2.360596	5.412890	6.561391	11.064344
18	45×18	0.155718	1.841446	4.143707	7.320655	9.111388
19	55×18	0.161539	1.881606	4.295036	6.951141	9.770800
20	65×18	0.252792	2.244636	5.554235	8.791584	10.963592
21	75×18	0.154529	2.030509	5.683016	6.984902	12.559484
22	15×24	0.214914	1.962991	4.953130	7.954776	9.286811
23	25×24	0.155734	1.478099	3.827843	5.831005	7.708885
24	35×24	0.236688	2.190656	6.094842	7.920382	9.532598
25	45×24	0.175096	1.751782	4.661826	6.416722	9.313707
26	55×24	0.193941	1.711516	4.853919	7.205386	10.966694
27	65×24	0.149720	2.234314	5.571039	7.308793	10.334833
28	75×24	0.183024	1.926492	5.908557	8.137918	11.250872
29	15×30	0.201083	1.670074	4.321663	6.217415	8.176248
30	25×30	0.134371	1.397664	4.057719	5.470979	7.364100
31	35×30	0.209980	1.865593	4.979091	7.705791	10.132463
32	45×30	0.147178	1.357357	3.765254	6.511878	8.562166
33	55×30	0.187919	1.966883	4.311455	7.319003	9.165274
34	65×30	0.229924	1.911086	4.878589	7.684026	9.849881
35	75×30	0.175163	1.942591	4.957115	7.973464	9.744161
Average	4.981326	0.198590	1.969007	5.135075	7.397384	10.20658

Table 2 Average running times (s) of algorithm **LRP**

Problem no.	Problem size $ N \times T $	Average running time (s)				
		R1	R2	R3	R4	R5
1	15×6	1.380000	1.588000	1.470000	1.459000	1.409000
2	25×6	5.225000	8.559000	6.219000	6.516000	6.679000
3	35×6	9.812000	6.118000	6.994000	6.340001	5.923000
4	45×6	6.700999	24.518001	21.979001	17.052002	15.481999
5	55×6	15.112001	18.103001	27.685001	18.762001	19.359999
6	65×6	14.076001	18.164000	32.847000	33.789005	20.841000
7	75×6	23.239000	21.876001	34.770001	32.771002	27.942999
8	15×12	4.289000	4.584999	4.832000	4.579000	4.808000
9	25×12	6.924999	10.240001	10.089000	10.080001	11.238000
10	35×12	16.556999	19.830000	25.052000	22.058998	22.072000
11	45×12	19.976999	26.641998	34.332001	35.303998	34.930002
12	55×12	38.872000	41.729001	56.537000	51.429004	53.740002
13	65×12	52.217999	58.764008	71.688000	72.252997	72.632001
14	75×12	55.154999	72.720996	89.216003	92.596002	89.714001
15	15×18	8.822000	9.389999	9.755000	9.545999	10.062000
16	25×18	18.934000	23.779002	22.628000	23.139998	24.259000
17	35×18	33.524002	46.914001	51.086002	48.235004	48.534998
18	45×18	42.014999	55.179004	74.216992	77.349994	75.810999
19	55×18	70.464001	89.425000	115.514001	110.760999	113.114001
20	65×18	115.368994	127.671008	152.614001	154.782007	155.157007
21	75×18	129.013989	176.932007	212.553003	207.684009	206.108008
22	15×24	14.732001	18.148001	16.578999	18.139999	17.664000
23	25×24	32.887000	38.410001	38.887997	41.954001	41.220999
24	35×24	66.048993	78.900995	89.127997	83.222003	82.757001
25	45×24	91.694006	101.178003	130.889001	125.881995	123.895996
26	55×24	157.068994	166.323987	205.870996	195.019995	194.783997
27	65×24	177.169995	231.805005	275.066016	279.362012	275.260010
28	75×24	264.804004	317.621997	370.847974	369.495996	363.122998
29	15×30	20.165001	25.209000	25.557999	25.590999	25.103000
30	25×30	46.544000	55.872998	65.517004	63.481000	64.272998
31	35×30	94.073999	127.539001	139.984009	129.829004	129.244995
32	45×30	119.879004	168.303992	210.379004	199.465991	206.473999
33	55×30	259.500977	256.701001	318.209009	301.734033	303.345972
34	65×30	346.276001	362.456006	422.402002	416.231006	416.112988
35	75×30	346.161963	472.652979	545.128027	545.039014	541.995947
Average	100.3489	77.8481700	93.8242900	111.900700	109.455300	108.716300

5 Variable splitting

Variable splitting is another type of relaxation suggested and named by Jörnsten et al. (1985). The idea behind the technique is to use a different ‘copy’ of original variables to reformulate one problem in an equivalent form. The resulting problem has twice as many variables and then is usually relaxed by traditional LR algorithms. This technique can be exploited to solve the production-inventory problem (CP) as follows. A new set of variables $y_{il}^t = y_{il}^t$ is introduced. Then, problem (CP) can be rewritten as:

$$\begin{aligned}
 (\text{SCP}) \text{ Minimise } C_S, \text{ with } C_S \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\
 & + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\
 & + \sum_{i \in N} \sum_{t \in T} \frac{1}{2} \cdot h_i \cdot (I_i^{t-1} + x_i^t + I_i^t)
 \end{aligned} \tag{20}$$

Subject to:

$$I_i^{t-1} + x_i^t - I_i^t - \sum_{j \in S_i^t} \sum_{l \in H_{ij}^t} r_{ijl}^t \cdot z_{jl}^t - \sum_{j \in LI_i^t} \rho_{ij} \cdot w_{ij}^{t+ld_{ij}} = 0, \text{ for } i \in N, t \in T \tag{21}$$

$$\sum_{l \in pr_i^t} z_{il}^t + \sum_{j \in bk_i} w_{ji}^t = x_i^t, \text{ for } i \in N, t \in T \tag{22}$$

$$z_{j,pr_j^t}^t = z_{j,L_f}^t = d_j^t, \text{ for } j \in N_f, t \in T \tag{23}$$

$$z_{jl}^t = r_{ijl}^t \cdot z_{il}^t, \text{ for } l \in L_1, t \in T, i \in F_l^t, j \in B_{il}^t \tag{24}$$

$$z_{il}^t \leq M \cdot y_{il}^t, \text{ for } i \in N, t \in T, l \in pr_i^t \tag{25}$$

$$\sum_{i \in F_l^t} p_{il}^t \cdot z_{il}^t \leq cap_l^t, \text{ for } l \in L_1, t \in T \tag{26}$$

$$\sum_{i \in F_l^t} p_{il}^t \cdot z_{il}^t \leq cap_l^t, \text{ for } l \in L_1, t \in T \tag{27}$$

$$\frac{1}{2} \cdot \sum_{i \in U_k} (I_i^{t-1} + x_i^t + I_i^t) \leq V_k, \text{ for } k = 1, \dots, K, t \in T \tag{28}$$

$$y_{il}^t = 1, \text{ for } z_{il}^t > 0, pr_j^t = \{l\}, E_j^t = \{i\}, bk_i = \Phi, j \in N \cup N_f, t \in T \tag{29}$$

$$y_{il}^t = y_{il}^t, \text{ for } i \in N, t \in T, l \in pr_i^t \tag{30}$$

$$0 \leq y_{il}^t \leq 1, \text{ for } i \in N, t \in T, l \in pr_i^t \tag{31}$$

$$z_{il}^t \geq 0, \text{ for } i \in N \cup N_f, t \in T, l \in pr_i^t \tag{32}$$

$$w_{ij}^t \geq 0, \text{ for } i \in N, j \in rs_i, t \in T \tag{33}$$

$$x_i^t, I_i^t \geq 0, \text{ for } i \in N, t \in T \quad (34)$$

$$y_{il}^t \in \{0, 1\}, \text{ for } i \in N, t \in T, l \in pr_i^t \quad (35)$$

5.1 Lagrangian relaxation scheme

Notice that there are two types of decision variables in the problem, i.e., binary variables and continuous variables, and that the equality constraints (30) are the only constraints involving both types of variables. Without the equality constraints, the problem can be decomposed into two separate subproblems, one in binary variables and the other in continuous variables. This observation motivates us to relax these constraints. Denoting μ_{il}^t as the dual variable for the equality constraint (30) corresponding to item i , operation l and period t in problem (SCP), we obtain the following problem (SLR):

$$\begin{aligned} \text{(SLR) Minimise } C_{SLR}, \text{ with } C_{SLR} \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\ & + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\ & + \sum_{i \in N} \sum_{t \in T} \frac{1}{2} \cdot h_i \cdot (I_i^{t-1} + x_i^t + I_i^t) \\ & + \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_i^t} \mu_{il}^t \cdot (y_{il}^t - y_{il}^t) \end{aligned} \quad (36)$$

Subject to constraints (21)–(29) and (31)–(35).

The relaxed problem (SLR) can be decomposed into two separate subproblems, **SLR₁** and **SLR₂**.

$$\begin{aligned} \text{(SLR}_1\text{) Minimise } C_{SLR_1}, \text{ with } C_{SLR_1} \equiv & \sum_{t \in T} \sum_{l \in L_1} \sum_{i \in F_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\ & + \sum_{t \in T} \sum_{l \in L_2} \sum_{i \in J_l^t} (sc_{il}^t \cdot y_{il}^t + pc_{il}^t \cdot z_{il}^t) \\ & + \sum_{i \in N} \sum_{t \in T} \frac{1}{2} \cdot h_i \cdot (I_i^{t-1} + x_i^t + I_i^t) \\ & - \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_{it}} \mu_{il}^t \cdot y_{il}^t \end{aligned} \quad (37)$$

Subject to constraints (21)–(29) and (31)–(34).

$$\text{(SLR}_2\text{) Minimise } C_{SLR_2}, \text{ with } C_{SLR_2} \equiv \sum_{i \in N} \sum_{t \in T} \sum_{l \in pr_{it}} \mu_{il}^t \cdot y_{il}^t \quad (38)$$

Subject to constraints (35).

Attempting to find the set of multipliers for (SLR) that can attain the tightest lower bound corresponds to solving the Lagrangian dual problem (SLD) expressed as:

$$\text{(SLD) Maximise } C_{SD}(\mu_{il}^t), \text{ with } C_{SD}(\mu_{il}^t) \equiv \min C_{SLR} \quad (39)$$

Subject to constraints (22)–(30), and (32)–(36).

Note that problems (SLR₁) and (SLR₂) still can be directly solved by OSL and the polynomial algorithm in Section 4.1, respectively. In what follows, a new heuristic algorithm is developed to further decrease the upper bound by changing the feasible production quantities.

5.2 Heuristic strategy II

This approach begins with a feasible solution $(I_i^t, x_i^t, z_{il}^t, y_{il}^t, w_{ij}^t)$ and then attempts to improve that solution by adjusting the values of z_{il}^t , for $i \in N \cup N_f$, $t \in T$, $l \in pr_i^t$. The steps of the heuristic are followed as:

- Step 1 Initialisation: set $flag = 1$, $visit_mark[i][l][t] = 0$, for $i \in N \cup N_f$, $t \in T$, $l \in pr_i^t$.
- Step 2 In this feasible solution, search for variables z_{il}^t and $z_{il}^{t_1}$ that satisfy the conditions: $z_{il}^t > 0$, $z_{il}^{t_1} > 0$, $t_1 \geq t$, and $visit_mark[i][l][t_1] = 0$. If there exist no such variables, set $flag = 0$, go to Step 7.
- Step 3 If $(pc_{il}^t - pc_{il}^{t_1}) \cdot z_{il}^t + h_i \cdot z_{il}^{t_1} \cdot (t_1 - t) - sc_{il}^{t_1} < 0$, let $z_{il}^{t_1} = 0$, $visit_mark[i][l][t_1] = 1$; otherwise return to Step 2.
- Step 4 Introduce the constraint, $z_{il}^{t_1} = 0$, into the problem (SLR₁) leading to a tighter problem (SLR₃).
- Step 5 Use standard software package OSL to solve problem (SLR₃). If the problem has no feasible solution, set $flag = 0$, go to Step 7.
- Step 6 Construct a feasible solution based on the solution obtained from Step 5.
- Step 7 Stop. If $flag = 1$, a new feasible solution $(I_i^{t(1)}, x_i^{t(1)}, z_{il}^{t(1)}, y_{il}^{t(1)}, w_{ij}^{t(1)})$ is obtained. If the objective value is less than the upper bound, update the current optimal solution and the upper bound. If $flag = 0$, the heuristic is not effective for this problem.

Subsequently, an algorithm for solving the Lagrangian dual problem (SLD) with respect to (39) is suggested. The algorithmic details are described formally as follows.

5.3 Subgradient algorithm (SAS) for solving problem (SLD)

The subgradient algorithm for problem (SLD) is implemented as follows.

- Step 1 Initialise
 - 1 Denote the upper bound and lower bound of function C_s as C^{SU} and C^{SL} , respectively.
 - 2 Set $\mu_{il}^{t(0)} = -M$, $C^{SU} = +\infty$, $C^{SL} = 0$, $flag = 0$, $flag_v = 0$.
 - 3 Refer to Section 4.4 for the initialisation of other parameters.

- Step 2 Solve Lagrangian relaxed problem (SLR).
- Step 3 Use the heuristics in 4.2 to construct a feasible solution based on the solution of Step 2.
- Step 4 If $flag \geq 10$ and $flag_v = 0$:
- 1 Solve the linear relaxed problem of problem (SLR) and set $flag_v = 1$.
 - 2 Let SC_1 represent the optimal objective value.
 - 3 Construct a feasible solution to the original problem based on this LP relaxed solution. Denote the corresponding objective value as SC_2 .
 - 4 If $C^{SL} < SC_1$, take SC_1 as the lower bound.
 - 5 If $C^{SU} > SC_2$, take SC_2 as the upper bound, set $flag = 0$; otherwise let $flag = flag + 1$.
- Step 5 If $flag < 10$, perform heuristic strategy II to update the upper bound.
- Step 6 Convergence check.
- Step 7 Update the Lagrangian multipliers and return to Step 2.

5.4 Experiment B

To assess the performance of algorithm SAS, the same instances used in experiment A have been tested in experiment B. The computational results are listed in Tables 3 and 4. By comparing the average duality gap and average running time of algorithms LRP and SAS, we can draw the following conclusions:

- 1 The average duality gap of algorithm SAS is 2.916039. Compared with algorithm LRP, the percent reduction in the average duality gap is 41.46%, which proves the effectiveness of algorithm SAS.
- 2 For algorithm SAS, the average growth rate of the duality gaps for different ranges of setup costs = $((1.202020 - 0.138244) + (3.031516 - 1.202020) + (4.263587 - 3.031516) + (5.944825 - 4.263587))/4 = (5.944825 - 0.138244)/4 = 1.45164525$. It is reduced to 0.58 times of that obtained in algorithm LRP. The fact means that algorithm SAS is relatively stable and significantly less affected by problem size.
- 3 The average running time of algorithm SAS is 252.3441 seconds. Although it is 2.51 times that of algorithm LRP, it is still acceptable.
- 4 Without consideration of running times, algorithm SAS outperforms algorithm LRP in all cases.
- 5 In summary, algorithm SAS is the best in cost performance. Since the production-inventory problem is an offline optimisation problem, the major concern is the quality of solutions. Thus, algorithm SAS is recommended.

Table 3 Average duality gaps (%) of algorithm SAS

<i>Problem no.</i>	<i>Problem size $N \times T$</i>	<i>Average duality gap (%)</i>				
		R_1	R_2	R_3	R_4	R_5
1	15×6	0.064865	0.456061	1.304912	1.469335	1.746652
2	25×6	0.150931	1.111181	2.730329	4.081159	5.650718
3	35×6	0.174242	1.588785	3.911268	4.294080	8.496389
4	45×6	0.143380	1.290602	3.545244	4.970428	7.040873
5	55×6	0.134433	1.303222	3.056768	4.496362	6.079207
6	65×6	0.175899	1.630547	4.064969	4.365901	7.100291
7	75×6	0.203588	1.790333	4.935644	5.438166	8.708100
8	15×12	0.069780	0.706980	1.484203	2.617586	2.877434
9	25×12	0.088965	0.777596	1.884457	3.490922	4.531084
10	35×12	0.145876	1.449949	3.296745	3.707165	6.113657
11	45×12	0.137481	1.409096	3.321066	4.879991	6.506573
12	55×12	0.158144	1.561836	3.440524	5.773742	7.123040
13	65×12	0.178322	1.499527	3.872741	5.717322	7.366020
14	75×12	0.190550	1.567115	3.982262	5.346023	8.024812
15	15×18	0.102674	0.759299	1.935952	2.885630	3.215219
16	25×18	0.103292	0.790694	2.093205	3.180161	4.438794
17	35×18	0.147471	1.245012	2.924624	3.390488	5.865893
18	45×18	0.130138	1.296231	2.791872	4.881190	6.014440
19	55×18	0.140919	1.228115	2.789012	4.349262	6.087984
20	65×18	0.183059	1.596342	3.882353	6.043990	7.475513
21	75×18	0.209296	1.499191	4.095093	4.998631	8.671709
22	15×24	0.086202	0.682937	1.890197	2.849832	3.252705
23	25×24	0.093295	0.930869	2.369252	3.389241	4.511287
24	35×24	0.140355	1.185591	3.289926	4.223836	4.951801
25	45×24	0.133116	1.167126	3.074189	4.256707	6.209682
26	55×24	0.128699	1.112749	3.298273	4.768717	7.311802
27	65×24	0.155280	1.520361	3.933182	4.904429	7.042474
28	75×24	0.184844	1.547975	4.224388	5.683824	7.858074
29	15×30	0.084558	0.588553	1.505115	2.015250	2.754336
30	25×30	0.088631	0.852067	2.445224	3.110983	4.393134
31	35×30	0.141344	1.038854	2.652212	4.073314	5.390221
32	45×30	0.115180	0.899754	2.468449	4.179338	5.389807
33	55×30	0.121981	1.253320	2.692568	4.433237	5.608032
34	65×30	0.155586	1.298256	3.161209	5.220101	6.790602
35	75×30	0.176179	1.434588	3.755640	5.739195	7.470512
Average	2.916039	0.138244	1.202020	3.031516	4.263587	5.944825

Table 4 Average running times (s) of algorithm SAS

Problem no.	Problem size $ N \times T $	Average running time (s)				
		R_1	R_2	R_3	R_4	R_5
1	15×6	1.246000	3.511000	3.633000	3.733000	3.607000
2	25×6	8.742999	10.117001	11.556999	12.190001	19.104999
3	35×6	16.793999	18.843999	20.650000	19.723001	28.066998
4	45×6	22.617999	26.318002	33.172000	25.500999	27.638000
5	55×6	28.560999	33.968997	35.212000	50.195007	34.524002
6	65×6	44.788995	46.370001	51.838000	52.567004	49.775000
7	75×6	65.898999	65.142999	76.933997	70.859003	74.548993
8	15×12	4.143000	11.998000	12.064000	12.607000	12.813000
9	25×12	11.784000	29.622998	29.586996	30.300998	30.264001
10	35×12	51.594995	50.885001	53.330005	56.306995	56.507001
11	45×12	75.584003	83.541003	87.556995	92.221997	87.469995
12	55×12	115.482007	125.983008	126.108008	132.119006	144.375000
13	65×12	156.420020	163.969006	169.741003	176.805994	173.721021
14	75×12	190.225000	207.237012	233.440039	217.884009	236.481006
15	15×18	18.253000	25.106000	26.220999	26.569998	27.542999
16	25×18	35.973999	58.313000	59.955005	61.128003	62.332996
17	35×18	103.547998	104.254004	114.516003	113.427002	116.531982
18	45×18	157.837988	175.252014	190.593994	200.494006	184.201990
19	55×18	246.346997	259.380005	271.416016	263.429004	304.564990
20	65×18	357.079980	371.953979	386.061011	375.836011	367.097021
21	75×18	466.145996	455.095996	494.154980	497.675000	555.277979
22	15×24	23.512000	41.435004	40.542999	42.944995	44.481000
23	25×24	51.917999	94.071997	102.468994	108.848999	111.216992
24	35×24	180.005005	185.865991	195.110986	191.504993	216.271997
25	45×24	260.750000	278.659033	300.118994	295.692969	312.493994
26	55×24	443.389990	455.626953	469.715039	511.786035	456.522021
27	65×24	580.720996	590.330957	627.256006	654.933008	659.873926
28	75×24	845.725000	817.669043	852.925977	833.633984	934.331934
29	15×30	35.378003	61.387006	62.858008	61.751001	69.697998
30	25×30	85.506006	153.938989	157.773999	162.532996	168.594995
31	35×30	250.495996	285.945996	292.209985	300.181006	309.368994
32	45×30	375.395972	438.747021	440.702979	457.987012	461.537988
33	55×30	539.077051	686.635010	697.687988	722.612988	702.074023
34	65×30	922.194043	950.878027	934.989941	991.917969	997.152930
35	75×30	1244.356055	1221.601953	1291.958105	1387.550977	1343.492969
Average	252.3441	229.0712000	245.4187000	255.8303000	263.2986000	268.1016000

6 Conclusions

With increasing concerns about sustainable production and environmental protection, production planning with reverse logistics has received growing attention. In this paper, we discuss the production-inventory problem with reverse logistics derived from the ironmaking production system in Baosteel and formulate it as a mixed integer programming model in order to determine the production and inventory quantities of each material in each time period under complicated material-balance constraints and capacity constraints. The objective of the model is to minimise the total setup, production/purchasing and inventory holding cost over the planning horizon. A series of Lagrangian relaxation algorithms and heuristic strategies are applied to solve the problem. Property 1 and variable splitting are induced to reduce the duality gaps. The computational results for test problems generated based on the production data of Baosteel indicate that the proposed algorithms can usually find high quality solutions in a timely fashion, and the Lagrangian relaxation approach based on variable splitting performs best.

In this paper, we have effectively solved the multi-level capacitated lot-sizing problem in a special system with combined divergent, convergent and cyclic features. It should be noted that our models and algorithms are only effective for accurate production data. However, in some cases, it is difficult to obtain such data. For example, customer demand is uncertain, and the supply lead times of raw material suppliers are uncertain, and so on. How to optimise the production-inventory management problem with complex product structure under various uncertain environments is a future direction worthy of further investigation. In addition, the collaborative optimisation of energy, transportation and production-inventory planning is also a direction worth exploring.

Acknowledgements

The authors are grateful to the editor and the two anonymous reviewers for their valuable comments, which led to significant improvements of this paper. This research is partly supported by the National Natural Science Foundation of China (Grant No. 71301066) and the Social Science Planning Fund of Liaoning Province (Grant No. L21CXW008).

References

- Afra, A.P. and Behnamian, J. (2021) 'Lagrangian heuristic algorithm for green multi-product production routing problem with reverse logistics and remanufacturing', *J. Manuf. Syst.*, Vol. 58, pp.33–43, <https://doi.org/10.1016/j.jmsy.2020.11.013>.
- Barzinpour, F. and Taki, P. (2018) 'A dual-channel network design model in a green supply chain considering pricing and transportation mode choice', *J. Intell. Manuf.*, Vol. 29, pp.1465–1483, <https://doi.org/10.1007/s10845-015-1190-x>.
- Bazan, E., Jaber, M.Y. and Zanoni, S. (2016) 'A review of mathematical inventory models for reverse logistics and the future of its modeling: an environmental perspective', *Appl. Math. Model.*, Vol. 40, pp.4151–4178, <https://doi.org/10.1016/j.apm.2015.11.027>.
- Bing, X., Bloemhof-Ruwaard, J.M. and van der Vorst, J.G.A.J. (2014) 'Sustainable reverse logistics network design for household plastic waste', *Flex. Serv. Manuf. J.*, Vol. 26, pp.119–142, <https://doi.org/10.1007/s10696-012-9149-0>.

- Bouras, A. (2015) 'Optimal control for advertised production planning in a three-level stock system with deteriorating items: case of a continuous-review policy', *Arab. J. Sci. Eng.*, Vol. 40, pp.2829–2840, <https://doi.org/10.1007/s13369-015-1761-3>.
- Bouras, A., Hedjar, R. and Tadj, L. (2016) 'Production planning in a three-stock reverse-logistics system with deteriorating items under a periodic review policy', *J. Ind. Manag. Optim.*, Vol. 12, pp.1075–1089, <https://doi.org/10.3934/jimo.2016.12.1075>.
- Chan, C.K., Man, N., Fang, F. and Campbell, J.F. (2020) 'Supply chain coordination with reverse logistics: A vendor/recycler-buyer synchronized cycles model', *Omega*, Vol. 95, p.102090, <https://doi.org/10.1016/j.omega.2019.07.006>.
- Chen, S-P. and Lin, Y-M. (2022) 'Inventory control for the closed-loop supply chain with technological progress and planned shortages', *Eur. J. Ind. Eng.*, Vol. 16, No. 2, pp.143–168, <https://doi.org/10.1504/EJIE.2022.121189>.
- Chen, W.H. and Thizy, J.M. (1990) 'Analysis of relaxations for the multi-item capacitated lot-sizing problem', *Ann. Oper. Res.*, Vol. 26, pp.29–72, <https://doi.org/10.1007/BF02248584>.
- Chung, S.H., Ma, H.L. and Chan, H.K. (2018) 'Maximizing recyclability and reuse of tertiary packaging in production and distribution network', *Resour. Conserv. Recy.*, Vol. 128, pp.259–266, <https://doi.org/10.1016/j.resconrec.2016.06.025>.
- Coelho, E.K.F. and Mateus, G.R. (2017) 'A capacitated plant location model for reverse logistics activities', *J. Clean. Prod.*, Vol. 167, pp.1165–1176, <https://doi.org/10.1016/j.jclepro.2017.07.238>.
- Cunha, J.O. and Melo, R.A. (2016) 'A computational comparison of formulations for the economic lot-sizing with remanufacturing', *Comput. Ind. Eng.*, Vol. 92, pp.72–81, <https://doi.org/10.1016/j.cie.2015.11.024>.
- Cunha, J.O., Konstantaras, I., Melo, R.A. and Sifaleras, A. (2017) 'On the multi-item economic lot-sizing with remanufacturing and uncapacitated production', *Appl. Math. Model.*, Vol. 50, pp.772–780, <https://doi.org/10.1016/j.apm.2016.10.037>.
- Das, K. and Chowdhury, A.H. (2012) 'Designing a reverse logistics network for optimal collection, recovery and quality-based product-mix planning', *Int. J. Prod. Econ.*, Vol. 135, pp.209–221, <https://doi.org/10.1016/j.ijpe.2011.07.010>.
- Demirel, E., Demirel, N. and Gökçen, H. (2016) 'A mixed integer linear programming model to optimize reverse logistics activities of end-of-life vehicles in Turkey', *J. Clean. Prod.*, Vol. 112, pp.2101–2113, <https://doi.org/10.1016/j.jclepro.2014.10.079>.
- Demirel, N.Ö. and Gökçen, H. (2008) 'A mixed integer programming model for remanufacturing in reverse logistics environment', *Int. J. Adv. Manuf. Tech.*, Vol. 39, pp.1197–1206, <https://doi.org/10.1007/s00170-007-1290-7>.
- Desport, P., Liret, A., C.D. Cairano-Gilfedder, Owusu, G., Lesaint, D. and Lardeux, F. (2017) 'A combinatorial optimisation approach for closed-loop supply chains inventory planning with deterministic demands', *Eur. J. Ind. Eng.*, Vol. 11, No. 3, pp.303–327, <https://doi.org/10.1504/EJIE.2017.084878>.
- Diabat, A., Battaia, O. and Nazzal, D. (2015) 'An improved Lagrangian relaxation-based heuristic for a joint location-inventory problem', *Comput. Oper. Res.*, Vol. 61, pp.170–178, <https://doi.org/10.1016/j.cor.2014.03.006>.
- Diabat, A., Richard, J-P. and Codrington, C.W. (2013) 'A Lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design', *Ann. Oper. Res.*, Vol. 203, pp.55–80, <https://doi.org/10.1007/s10479-011-0915-2>.
- Dobos, I. (2003) 'Optimal production-inventory strategies for a HMMS-type reverse logistics system', *Int. J. Prod. Econ.*, Vols. 81–82, pp.351–360, [https://doi.org/10.1016/S0925-5273\(02\)00277-3](https://doi.org/10.1016/S0925-5273(02)00277-3).
- Dumez, D., Tilk, C., Irnich, S., Lehuédé, F., Olkis, K. and Péton, O. (2023) 'A matheuristic for a 2-echelon vehicle routing problem with capacitated satellites and reverse flows', *Eur. J. Oper. Res.*, Vol. 305, pp.64–84, <https://doi.org/10.1016/j.ejor.2022.05.022>.

- Fang, C., Liu, X., Pardalos, P.M. and Pei, J. (2016) 'Optimization for a three-stage production system in the internet of things: procurement, production and product recovery, and acquisition', *Int. J. Adv. Manuf. Tech.*, Vol. 83, pp.689–710, <https://doi.org/10.1007/s00170-015-7593-1>.
- Garcia-Sabater, J.P., Maheut, J. and Marin-Garcia, J.A. (2013) 'A new formulation technique to model materials and operations planning: the generic materials and operations planning (GMOP) problem', *Eur. J. Ind. Eng.*, Vol. 7, No. 2, pp.119–147, <https://doi.org/10.1504/EJIE.2013.052572>.
- Ghanbarzadeh-Shams, M., Yaghin, R.G. and Sadeghi, A.H. (2022) 'A hybrid fuzzy multi-objective model for carpet production planning with reverse logistics under uncertainty', *Socio-Econ. Plan. Sci.*, Vol. 83, p.101344, <https://doi.org/10.1016/j.seps.2022.101344>.
- Govindan, K., Jha, P.C. and Garg, K. (2016) 'Product recovery optimization in closed-loop supply chain to improve sustainability in manufacturing', *Int. J. Prod. Res.*, Vol. 54, pp.1463–1486, <https://doi.org/10.1080/00207543.2015.1083625>.
- Guo, J., Wang, X., Fan, S. and Gen, M. (2017) 'Forward and reverse logistics network and route planning under the environment of low-carbon emissions: a case study of Shanghai fresh food E-commerce enterprises', *Comput. Ind. Eng.*, Vol. 106, pp.351–360, <https://doi.org/10.1016/j.cie.2017.02.002>.
- Hamdan, B. and Diabat, A. (2020) 'Robust design of blood supply chains under risk of disruptions using Lagrangian relaxation', *Transport. Res. E*, Vol. 134, p.101764, <https://doi.org/10.1016/j.tre.2019.08.005>.
- Hong, X., Zhang, H., Zhong, Q. and Liu, L. (2016) 'Optimal decisions of a hybrid manufacturing-remanufacturing system within a closed-loop supply chain', *Eur. J. Ind. Eng.*, Vol. 10, No. 1, pp.21–50, <https://doi.org/10.1504/EJIE.2016.075097>.
- Hu, T.L., Sheu, J.B. and Huang, K.H. (2002) 'A reverse logistics cost minimization model for the treatment of hazardous wastes', *Transport. Res. E-Log.*, Vol. 38, pp.457–473, [https://doi.org/10.1016/S1366-5545\(02\)00020-0](https://doi.org/10.1016/S1366-5545(02)00020-0).
- Jauhar, S.K. and Pant, M. (2016) 'Genetic algorithms in supply chain management: a critical analysis of the literature', *Sādhanā*, Vol. 41, pp.993–1017, <https://doi.org/10.1007/s12046-016-0538-z>.
- Jiang, Z., Zhou, T., Zhang, H., Wang, Y., Cao, H. and Tian, G. (2016) 'Reliability and cost optimization for remanufacturing process planning', *J. Clean. Prod.*, Vol. 135, pp.1602–1610, <https://doi.org/10.1016/j.jclepro.2015.11.037>.
- Jing, Y., Li, W., Wang, X. and Deng, L. (2016) 'Production planning with remanufacturing and back-ordering in a cooperative multi-factory environment', *Int. J. Comput. Integ. M.*, Vol. 29, pp.692–708, <https://doi.org/10.1080/0951192X.2015.1068450>.
- Johnson, P.F. (1998) 'Managing value in reverse logistics systems', *Transport. Res. E-Log.*, Vol. 34, pp.217–227, [https://doi.org/10.1016/S1366-5545\(98\)00016-7](https://doi.org/10.1016/S1366-5545(98)00016-7).
- Jörnsten, K., Näsberg, M. and Smeds, P. (1985) *Variable Splitting: A New Lagrangean Relaxation Approach to Some Mathematical Programming Models*, Working Paper LITH-MAT-R-85-04, Linköping University.
- Jung, K.S., Dawande, M., Geismar, H.N., Guide Jr., V.D.R., Sriskandarajah, C. (2016) 'Supply planning models for a remanufacturer under just-in-time manufacturing environment with reverse logistics', *Ann. Oper. Res.*, Vol. 240, pp.533–581, <https://doi.org/10.1007/s10479-014-1569-7>.
- Kannan, D., Garg, K., Jha, P.C. and Diabat, A. (2017) 'Integrating disassembly line balancing in the planning of a reverse logistics network from the perspective of a third party provider', *Ann. Oper. Res.*, Vol. 253, pp.353–376, <https://doi.org/10.1007/s10479-016-2272-7>.
- Khorshidian, H., Shirazi, M.A. and Ghomi, S.M.T.F. (2019) 'An intelligent truck scheduling and transportation planning optimization model for product portfolio in a cross-dock', *J. Intell. Manuf.*, Vol. 30, pp.163–184, <https://doi.org/10.1007/s10845-016-1229-7>.

- Kilic, H.S., Cebeci, U. and Ayhan, M.B. (2015) 'Reverse logistics system design for the waste of electrical and electronic equipment (WEEE) in Turkey', *Resour. Conserv. Recy.*, Vol. 95, pp.120–132, <https://doi.org/10.1016/j.resconrec.2014.12.010>.
- Kim, K., Song, I., Kim, J. and Jeong, B. (2006) 'Supply planning model for remanufacturing system in reverse logistics environment', *Comput. Ind. Eng.*, Vol. 51, pp.279–287, <https://doi.org/10.1016/j.cie.2006.02.008>.
- Kumar, V.N.S.A., Kumar, V., Brady, M., Garza-Reyes, J.A. and Simpson, M. (2017) 'Resolving forward-reverse logistics multi-period model using evolutionary algorithms', *Int. J. Prod. Econ.*, Vol. 183, pp.458–469, <https://doi.org/10.1016/j.ijpe.2016.04.026>.
- Lee, H., Zhang, T., Boile, M., Theofanis, S. and Choo, S. (2013) 'Designing an integrated logistics network in a supply chain system', *KSCE J. Civ. Eng.*, Vol. 17, pp.806–814, <https://doi.org/10.1007/s12205-013-0087-5>.
- Lee, J.E., Chung, K.Y., Lee, K.D. and Gen, M. (2015) 'A multi-objective hybrid genetic algorithm to minimize the total cost and delivery tardiness in a reverse logistics', *Multimed. Tools Appl.*, Vol. 74, pp.9067–9085, <https://doi.org/10.1007/s11042-013-1594-6>.
- Lei, J., Che, A. and van Woensel, T. (2024) 'Collection-disassembly-delivery problem of disassembly centers in a reverse logistics network', *Eur. J. Oper. Res.*, Vol. 313, No. 2, pp.478–493, <https://doi.org/10.1016/j.ejor.2023.07.008>.
- Mar-Ortiz, J., González-Velarde, J.L. and Adenso-Díaz, B. (2014) 'A VNS algorithm for a disassembly cell formation problem with demand variability', *Eur. J. Ind. Eng.*, Vol. 8, No. 1, pp.22–49, <https://doi.org/10.1504/EJIE.2014.059347>.
- Mutha, A. and Pokharel, S. (2009) 'Strategic network design for reverse logistics and remanufacturing using new and old product modules', *Comput. Ind. Eng.*, Vol. 56, pp.334–346, <https://doi.org/10.1016/j.cie.2008.06.006>.
- Panagiotidou, S., Nenes, G., Zikopoulos, C. and Tagaras, G. (2017) 'Joint optimization of manufacturing/remanufacturing lot sizes under imperfect information on returns quality', *Eur. J. Oper. Res.*, Vol. 258, No. 2, pp.537–551, <https://doi.org/10.1016/j.ejor.2016.08.044>.
- Pishvaei, M.S., Kianfar, K. and Karimi, B. (2010) 'Reverse logistics network design using simulated annealing', *Int. J. Adv. Manuf. Tech.*, Vol. 47, pp.269–281, <https://doi.org/10.1007/s00170-009-2194-5>.
- Qiu, Y., Ni, M., Wang, L., Li, Q., Fang, X. and Pardalos, P.M. (2018) 'Production routing problems with reverse logistics and remanufacturing', *Transport. Res. E.*, Vol. 111, pp.87–100, <https://doi.org/10.1016/j.tre.2018.01.009>.
- Reddy, K.N., Kumar, A., Choudhary, A. and Cheng, T.C.E. (2022) 'Multi-period green reverse logistics network design: an improved Benders-decomposition-based heuristic approach', *Eur. J. Oper. Res.*, Vol. 303, pp.735–752, <https://doi.org/10.1016/j.ejor.2022.03.014>.
- Sahling, F. (2016) 'Integration of vendor selection into production and remanufacturing planning subject to emission constraints', *Int. J. Prod. Res.*, Vol. 54, pp.3822–3836, <https://doi.org/10.1080/00207543.2016.1148276>.
- Sasikumar, P., Kannan, G. and Haq, A.N. (2010) 'A multi-echelon reverse logistics network design for product recovery-a case of truck tire remanufacturing', *Int. J. Adv. Manuf. Tech.*, Vol. 49, pp.1223–1234, <https://doi.org/10.1007/s00170-009-2470-4>.
- Shih, L.H. (2001) 'Reverse logistics system planning for recycling electrical appliances and computers in Taiwan', *Resour. Conserv. Recy.*, Vol. 32, pp.55–72, [https://doi.org/10.1016/S0921-3449\(00\)00098-7](https://doi.org/10.1016/S0921-3449(00)00098-7).
- Subulan, K. and Tasan, A.S. (2013) 'Taguchi method for analyzing the tactical planning model in a closed-loop supply chain considering remanufacturing option', *Int. J. Adv. Manuf. Tech.*, Vol. 66, pp.251–269, <https://doi.org/10.1007/s00170-012-4322-x>.
- Suzanne, E., Absi, N. and Borodin, V. (2020) 'Towards circular economy in production planning: challenges and opportunities', *Eur. J. Oper. Res.*, Vol. 287, pp.168–190, <https://doi.org/10.1016/j.ejor.2020.04.043>.

- Taskhiri, M.S., Garbs, M. and Geldermann, J. (2016) 'Sustainable logistics network for wood flow considering cascade utilisation', *J. Clean. Prod.*, Vol. 110, pp.25–39, <https://doi.org/10.1016/j.jclepro.2015.09.098>.
- Tasoglu, G. and Ilgin, M.A. (2024) 'A simulation-based genetic algorithm approach for the simultaneous consideration of reverse logistics network design and disassembly line balancing with sequencing', *Comput. Ind. Eng.*, Vol. 187, p.109794, <https://doi.org/10.1016/j.cie.2023.109794>.
- Wang, Z., Soleimani, H., Kannan, D. and Xu, L. (2016) 'Advanced cross-entropy in closed-loop supply chain planning', *J. Clean. Prod.*, Vol. 135, pp.201–213, <https://doi.org/10.1016/j.jclepro.2016.04.006>.
- Yu, H. and Solvang, W.D. (2016) 'A general reverse logistics network design model for product reuse and recycling with environmental considerations', *Int. J. Adv. Manuf. Tech.*, Vol. 87, pp.2693–2711, <https://doi.org/10.1007/s00170-016-8612-6>.
- Zhou, J., Yang, S., Feng, H. and An, Z. (2023) 'Multi-echelon sustainable reverse logistics network design with incentive mechanism for eco-packages', *J. Clean. Prod.*, Vol. 430, p.139500, <https://doi.org/10.1016/j.jclepro.2023.139500>.