Journal of Mathematical Modeling
Vol. 10, No. 4, 2022, pp. 481-494. Research Article

JMM

Two efficient heuristic algorithms for the integrated production planning and warehouse layout problem

Mohammad Pourmohammadi Fallah, Maziar Salahi*

Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
Email(s): mohammadfallah.math@gmail.com, salahim@guilan.ac.ir

Abstract. In (Zhang et al. An integrated strategy for a production planning and warehouse layout problem: modeling and solution approaches, Omega 68 (2017) 85–94) the authors have proposed a mixed-integer linear programming model for the integrated production planning and warehouse layout problem. To solve the model, they proposed a Lagrangian relax-and-fix heuristic that takes significant amount of time to stop with gaps above 5% for large-scale instances. Here, we present two heuristic algorithms to solve the problem. In the first one, we use a greedy approach by allocating warehouse locations with less reservation costs, and also less transportation costs from the production area to locations and from locations to the output point to items with higher demands. Then a smaller model is solved. In the second heuristic, first we sort items in descending order according to the fraction of sum of the demands for that item in the time horizon plus the maximum demand for that item in the time horizon and sum of all its demands in the time horizon. Then we categorize the sorted items into groups of 3, 4, or 5, and solve a small-scale optimization problem for each group, hoping to improve the solution of the first heuristic. Our preliminary numerical results show the effectiveness of the proposed heuristics.

Keywords: Capacitated lot-sizing, warehouse layout, mixed-integer linear programming, heuristics algorithm.
AMS Subject Classification 2020: 90C10, 90B05.

1 Introduction

The Capacitated Lot Sizing-Problem (CLSP) is the planning of lot sizes of multiple items over a planning horizon with the objective of minimizing setup and inventory holding costs while satisfying the demands over the planning horizon, where the production in each period is limited by a given capacity [9]. The first mathematical model for the CLSP with bounded inventory is proposed in [11]. Later the authors in [12] studied the CLSP for multiple products with bounded warehouse space. Gutierrez et al. [6] presented a

*Corresponding author.

Received: 8 July 2022 / Revised: 30 July 2022 / Accepted: 31 July 2022
DOI: 10.22124/JMM.2022.22637.2002

dynamic CLSP with warehouse capacity and bounded inventory levels based on the model of Love [11]. Hwang et al. discussed an economic lot-sizing problem with lost sales and bounded inventory [8]. In [1], the authors studied single-item and the multi-item CLSP with inventory bounds. Also, Brahimi et al. in [2] proposed models for a two-level lot-sizing problem with bounded inventory, and used Lagrangian heuristics to solve the proposed model.

Warehouse layout or capacity can be a critical issue for the CLSP. Zhang et al. [15] studied a capacitated dynamic lot-sizing problem with setup, production, freight, and inventory holding costs. In [13] a heuristic algorithm is proposed for the multi-level CLSP with inventory constraints, where storage capacities on shop floors are strictly limited. The authors in [7] also developed a model to optimize lot sizing in an integrated multi-level supply chain under stochastic constraints and limited warehouse space.

In all the abovementioned research, the CLSP is considered with inventory or warehouse capacity bound. However, lot-sizing is strongly related to the layout type and organization structure [3]. In the CLSP, the challenge that companies may face is the difficulty in finding available spaces during periods when the products are transferred from production area to the warehouse. Occasionally, items might be misplaced due to employee errors or lack of visibility, and thus the operators may not be able to find the items to be picked up although they are available, leading to extra production. The CLSP with warehouse layout decision in production planning has been studied in [16] for the first time. The authors combined the production planning with the storage location assignment and developed an integrated strategy that combines a dedicated storage location assignment with the CLSP into a single mixed-integer linear programming (MILP) model. It minimizes the total cost of travel, reserved storage space, handling, production, inventory holding, and setup costs. In a recent study also, the authors in [17] proposed an integrated production planning and warehouse storage assignment problem, where the visibility and traceability of items provided by IoT-enabled tracking systems in order to increase space utilization

Here, we present two heuristic algorithms to solve the Integrated Production Planning and Warehouse Layout problem (IPPWLP). In the first one, which is called the One-step heuristic, we use a greedy approach to allocate warehouse locations with less reservation costs, and also less transportation costs from the production area to locations and from locations to the output point of items with higher demands. Then a smaller problem is solved. In the second heuristic, which is called the Improved One-step heuristic, first we sort indices of items in descending order according to the fraction of sum of the demands for that item in the time horizon plus the maximum demand for that item in the time horizon and sum of its demands in the time horizon. Then we categorize the sorted items into groups of 3, 4, or 5 in a row that may also overlap with each other, and solve a small-scale optimization problem for each group, hoping to improve the solution of the first step. For the sake of comparison, we also have implemented Particle Swarm Optimization (PSO) algorithm for IPPWLP [10]. The rest of the paper is organized as follows. In Section 2, we review the MILP formulation of [16]. The One-step heuristic and its preliminary experimental results are presented in Section 3. Section 4 presents the Improved One-step heuristic and related numerical experiments and finally Section 5 concludes the paper.

2 The MILP formulation

To present the formulation of integrated production planning and warehouse layout problem developed in [16], we need the following notaions and variables:

- **Indices**:
  i : The index of an item.
  l : The index of a storage location within the warehouse.
  t : The index of a period within the planning horizon.
  The parameters in the model are:
  O^l : Unit cost of moving a column of any item from storage location l to the output point.
  P^l : Unit cost of moving a column of any item from the production area to storage location l.
  R^l : Unit cost of reserving storage location l.
  h_t^i : Unit inventory cost of holding item i at period t.
  c_t^i : Variable unit production cost of item i at period t.
  u_t^i : Unit setup cost of item i at period t.
  d_t^i : The amount of required demand of item i in period t. This parameter denotes the demand of each item, which varies from product to product and period to period.
  v_t^i : The variable capacity of item i at period t.
  f_t : A key resource constraint on production, for example, budget, production, labor, etc.
  M : A big number, used for logical constraints.

- **Variables**:
  x_t^i : The quantity of item i produced during period t.
  s_t^i : The inventory level for item i at the end of period t.
  y_t^i : 1, if item i is produced during period t; 0, otherwise.
  w_t^{il} : 1, if item i is moved from the production area and placed in storage location l during period t; 0, otherwise.
  q_t^{il} : 1, if item i is requested (by demand) from location l during period t; 0, otherwise.
  n_t^{il} : 1, if item i is inventoried in location l during period t; 0, otherwise.
  z^{il} : 1, if location l is reserved for item i for the planning horizon; 0, otherwise.
  Note: Each location can only hold one unit at a time.
  The objective function of the IPPWLP model consists of four parts, that are:

  1. Cost of reserving locations for items under a specific storage strategy:
  $$\sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il}.$$
  2. Cost of items transportation from the production area to assigned storage locations:
  $$\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T} P^l w_t^{il}.$$
  3. Cost of items transportation from the storage locations to the output point of warehouse:
  $$\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T} O^l q_t^{il}.$$
  4. Production, setup and holding inventory costs:
  $$\sum_{i=1}^{I}\sum_{t=1}^{T}(c_t^i x_t^i + u_t^i y_t^i + h_t^i s_t^i).$$

Thus the total objective function of the model can be written as follows:

$$Z=\sum_{i=1}^{I}\sum_{l=1}^{L}R^{l}z^{il}+\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T}P^{l}w_t^{il}+\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T}O^{l}q_t^{il}+\sum_{i=1}^{I}\sum_{t=1}^{T}(c_t^i x_t^i+u_t^i y_t^i+h_t^i s_t^i). \tag{1}$$

The goal is to find an optimal plan for the production of products and to allocate warehouses to the products produced, so that the total cost is minimized under the following constraints:

$$\sum_{i=1}^{I} z^{il}\le 1, \quad \forall l. \tag{2}$$

Constraints (2) guarantee that a location can only be reserved by one item.

$$\sum_{i=1}^{I} q_t^{il}\le 1, \quad \forall l,\forall t. \tag{3}$$

Constraints (3) guarantee that at most one item can be requested from each storage location in each period.

$$\sum_{i=1}^{I} w_t^{il}\le 1, \quad \forall l,\forall t. \tag{4}$$

Constraints (4) state that at most one item can be moved from production area to a specific storage location l in period t.

$$\sum_{i=1}^{I} n_t^{il}\le 1, \quad \forall l,\forall t. \tag{5}$$

Constraints (5) state that at most one item can be inventoried at each storage location in each period.

$$\sum_{l=1}^{L} q_t^{il}= d_t^i, \quad \forall i,\forall t. \tag{6}$$

Constraints (6) state that the number of products requested from storage locations during period t is equal to the demand of that product during period t.

$$\sum_{l=1}^{L} w_t^{il}= x_t^i, \quad \forall i,\forall t. \tag{7}$$

Constraints (7) ensure that the number of products moved from production area to the warehouse in a period is the same as what produced in the period.

$$\sum_{l=1}^{L} n_t^{il}= s_t^i, \quad \forall i,\forall t. \tag{8}$$

Constraints (8) state that all the locations that items remain in at the end of time period t are consistent with the inventory at the end of time period t.

$$q_t^{il}\le w_t^{il}+n_{t-1}^{il}, \quad \forall i,\forall l,\forall t. \tag{9}$$

Constraints (9) state that items could only be retrieved from locations which they have previously been moved in from production area or inventoried.

$$w_t^{il} + n_{t-1}^{il} \leq z^{il}, \quad \forall i, \forall l, \forall t. \tag{10}$$

Constraints (10) guarantee that when moving an item from production area to the warehouse, it can only be placed at a storage location that is empty and there is no item in that location.

$$n_t^{il} = w_t^{il} - q_t^{il} + n_{t-1}^{il}, \quad \forall i, \forall l, \forall t. \tag{11}$$

Constraints (11) are the flow balance of an item from the production area to a storage location, and from the storage location to the output point.

$$q_t^{il} \leq z^{il}, \quad \forall i, \forall l, \forall t. \tag{12}$$

Constraints (12) ensure that items are only retrieved from locations that had been reserved for them.

$$w_t^{il} \leq z^{il}, \quad \forall i, \forall l, \forall t. \tag{13}$$

Constraints (13) ensure that items are only placed in locations that had been reserved for them.

$$n_t^{il} \leq z^{il}, \quad \forall i, \forall l, \forall t. \tag{14}$$

Constraints (14) ensure that items are only inventoried in locations that have been reserved for them.

$$\sum_{i=1}^{I} \sum_{l=1}^{L} v_t^i \cdot w_t^{il} \leq f_t, \quad \forall t. \tag{15}$$

Constraints (15) ensure that the number of each item produced multiplied by the variable capacity of each item does not exceed any key resource during the planning horizon.

$$\sum_{l=1}^{L} w_t^{il} \leq M y_t^i, \quad \forall i, \forall t. \tag{16}$$

Constraints (16) are setup constraints; the constraints put a limit on production during each period, where M is an upper bound for both production capacity and item demands.

In short, the model of IPPWLP is as follows:

$$\begin{aligned} \min\ Z = & \sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T} P^l w_t^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T} O^l q_t^{il} + \sum_{i=1}^{I}\sum_{t=1}^{T} (c_t^i x_t^i + u_t^i y_t^i + h_t^i s_t^i) \\ \text{s.t. } & (2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15),(16), \\ & z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\} \quad \forall i, \forall l, \forall t. \end{aligned} \tag{17}$$

A simpler model can be derived by some substitution that are explained in the following:

- If we rewrite constraints (11) as $q_t^{il} = w_t^{il} + n_{t-1}^{il} - n_t^{il}$, then according to $n_t^{il} \geq 0\, \forall i$, $\forall l$, $\forall t$ constraints (9) will be a redundant constraints. Therefore it can be deleted. Also constraints (6) can be rewritten as $\sum_{l=1}^{L} (w_t^{il} + n_{t-1}^{il} - n_t^{il}) = d_t^i$.

- Constraints (3) can be deduced from constraints (2) and (12), therefore they can be removed.
- Constraints (12) can be deduced from constraints (9) and (10), Therefore they can be removed.
- Since $z^{il}, n_t^{il}, w_t^{il} \in \{0,1\}$, constraints (10) guarantee that constraints (13) and (14) are satisfied. Thus constraints (13) and (14) can be removed.
- In the objective function, we put $\sum_{l=1}^{L} w_t^{il}$ instead of x_t^i, so constraints (7) can be removed from the problem.
- In the objective function, we put $\sum_{l=1}^{L} n_t^{il}$ instead of s_t^i, so constraints (8) can be removed from the problem.

Therefore, the final simplified form of IPPWLP is as follows:

$$\begin{aligned} \min\ & Z = \sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T}\Big((P^l + c_t^i)w_t^{il} + O^l(w_t^{il} + n_{t-1}^{il} - n_t^{il}) + h_t^i n_t^{il}\Big) + \sum_{i=1}^{I}\sum_{t=1}^{T}(u_t^i y_t^i) \\ \text{s.t.}\ & (2),(4),(5),(6),(10),(11),(15),(16) \\ & z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\} \quad \forall i, \forall l, \forall t. \end{aligned} \tag{18}$$

A widely used technique for solving such MILPs are variants of Lagrangian relaxation, see for example [4, 5, 13, 14]. In [16], the authors proposed a Lagrangian relax-and-fix heuristic algorithm to solve (18) that needed significant amount of time for large-scale instances. In the next two sections, we will present two heuristic algorithms to solve it.

3 One-step heuristic

For simplicity, we drop constraints (15) from model (18). By doing so, model (18) can be considered as separate subproblems. First, we construct the matrix z as follows. According to the objective function of (18), index l appears only in the following terms:

$$\sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T} P^l w_t^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T} O^l q_t^{il}. \tag{19}$$

Therefore, we propose problems (20), (21) and (22) to obtain z by assigning warehouses that are less expensive to items with higher demands:

$$\begin{aligned} \min\quad & Z = \sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}(P^l + O^l) z^{il} \\ \text{s.t.}\quad & \sum_{i=1}^{I} z^{il} \le 1, \quad \forall l, \\ & \sum_{l=1}^{L} z^{il} = \max\{d_1^i, d_2^i, \dots, d_T^i\}, \quad \forall i, \\ & z^{il} \in \{0,1\} \quad \forall i, \forall l, \end{aligned} \tag{20}$$

$$
\begin{aligned}
\min \quad & Z=\sum_{i=1}^{I}\Big(\sum_{l=1}^{L}(R^l+P^l+O^l)z^{il}\Big)DRT_i \\
\text{s.t.} \quad & \sum_{i=1}^{I} z^{il} \le 1, \quad \forall l, \\
& \sum_{l=1}^{L} z^{il} = \max\{d_1^i, d_2^i, \ldots, d_T^i\}, \quad \forall i, \\
& z^{il} \in \{0,1\} \quad \forall i, \forall l,
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
\min \quad & Z=\sum_{i=1}^{I}\Big(\sum_{l=1}^{L}(R^l+P^l+O^l)z^{il}\Big)\Big(DRT_i+\frac{1}{COI_i}\Big) \\
\text{s.t.} \quad & \sum_{i=1}^{I} z^{il} \le 1, \quad \forall l, \\
& \sum_{l=1}^{L} z^{il} = \max\{d_1^i, d_2^i, \ldots, d_T^i\}, \quad \forall i, \\
& z^{il} \in \{0,1\} \quad \forall i, \forall l,
\end{aligned} \tag{22}
$$

where

$$DRT_i = \frac{\sum_{t=1}^{T} d_t^i}{\sum_{j=1}^{I}\sum_{t=1}^{T} d_t^j}, \quad \forall i,$$

$$COI_i = \frac{\max\{d_1^i, d_2^i, \ldots, d_T^i\}}{\sum_{j=1}^{I}\sum_{t=1}^{T} d_t^j}, \quad \forall i.$$

Then, we solve the following problem:

$$
\begin{aligned}
\min \ & Z=\sum_{i=1}^{I}\sum_{l=1}^{L} R^l z^{il} + \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{t=1}^{T}\Big((P^l+c_t^i)w_t^{il} + O^l(w_t^{il}+n_{t-1}^{il}-n_t^{il}) + h_t^i n_t^{il}\Big) + \sum_{i=1}^{I}\sum_{t=1}^{T}(u_t^i y_t^i) \\
\text{s.t.} \ & (2),(4),(5),(6),(10),(11),(16), \\
& w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\} \quad \forall i, \forall l, \forall t.
\end{aligned} \tag{23}
$$

To solve (23), we reduce it to several separable subproblems as follows. Based on the z obtained, some entries of w, q and n can be set to zero as follows:

- From constraints (12) if $z^{il}=0$ then $q^{il}=0$.
- From constraints (13) if $z^{il}=0$ then $w^{il}=0$.
- From constraints (14) if $z^{il}=0$ then $n^{il}=0$.

This leads to a significant reduction in the number of variables in the model. For example, suppose $I=153, L=813, T=12$ and also suppose

$$\max\{d_1^i, d_2^i, \ldots, d_T^i\} = 5, \quad \forall i.$$

Then number of variables n, w and q, which is 4478004 in total, reduces to 36720.
Now to solve problem (23), for each $1 \leq i \leq I$, let the index set of all locations with $z^{il} = 1$ be L_i, and solve the following problem for each i:

$$\begin{aligned}
\min\ Z_i = & \sum_{l \in L_i} R^l z^{il} + \sum_{l \in L_i} \sum_{t=1}^{T} c_t^i w_t^{il} + \sum_{l \in L_i} \sum_{t=1}^{T} P^l w_t^{il} + \sum_{l \in L_i} \sum_{t=1}^{T} (O^l (w_t^{il} + n_{t-1}^{il} - n_t^{il}) + h_t^i n_t^{il}) + \sum_{t=1}^{T} (u_t^i y_t^i) \\
\text{s.t.} \ & \sum_{l \in L_i} q_t^{il} = d_t^i, \quad \forall t, \\
& w_t^{il} + n_{t-1}^{il} \leq z^{il}, \quad \forall t, \forall l \in L_i, \\
& n_t^{il} = w_t^{il} - q_t^{il} + n_{t-1}^{il}, \quad \forall t, \forall l \in L_i, \\
& \sum_{l \in L_i} w_t^{il} \leq M y_t^i, \quad \forall t, \\
& w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\} \quad \forall i, \forall l, \forall t.
\end{aligned} \tag{24}$$

Solving problem (24) for each i determines the optimal production planning and warehouse layout for item i based on the z matrix obtained from the previous step. This is summarized in the Algorithm 1.

Algorithm 1 : One-step heuristic

1: Solve (22) to get the z.
2: **for** $i \leftarrow 1$ to I **do**
3: $\quad L_i = \emptyset$
4: $\quad$ **for** l **do**
5: $\quad\quad$ **if** $z^{il} = 0$ **then**
6: $\quad\quad\quad q^{il} = 0$, $w^{il} = 0$, $n^{il} = 0$
7: $\quad\quad$ **else**
8: $\quad\quad\quad L_i \leftarrow L_i \cup l$.
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad$ Solve (24) to get Z_i.
12: **end for**

3.1 Numerical results

Here, we apply the One-step heuristic for solving randomly generated instances taken from [16]. Table 1 gives instances size and Table 2 gives parameters values.

Table 1: Test instances

Case	Items No.	Storages No.	Period No.
Small_1	5	10	5
Small_2	10	25	6
Medium_1	25	65	8
Medium_2	50	233	12
Large	70	300	12

Table 2: Random data information

R^l	:	4.24	u_t^i	:	$4.50 \sim 15.0$
O^l	:	$3.51 \sim 5.51$	d_t^i	:	$1 \sim 2$
P^l	:	$8.14 \sim 9.06$	c_t^i	:	$5.82 \sim 6.73$
h_t^i	:	2.94	M	:	5
f_t	:	10	v_t	:	1

Parameters d_t^i, M and f_t may acquire other values according to the size of the problem.

Table 3 shows computational results, where we applied Gurobi in CVX-MATLAB environment to solve (18) directly. In this table, LB and UB denote lower and upper bounds of model (18), respectively, without constraints (15). The gap here is computed as follows:

$$\frac{UB - LB}{LB} \times 100.$$

As one can see, for the last three instances it takes significant amount of time to stop with gaps 0.74%, 2.77% and 4.52%, respectively.

Table 3: Performance of Gurobi.

Instance	LB	UB	GAP[%]	Time[s]
Small_1	952.1691	952.1691	0	0.78
Small_2	2373.1	2373.1	0	8.80
Medium_1	7443.4	7499.1	0.74	4000.00
Medium_2	33882.1578	34847.325427	2.77	10184.22
Large	46773.5414	48988.0636	4.52	30563.59

Table 4 reports the results of the One-step heuristic, where we use Gurobi to solve models (22) and (24). The gap here is computed as follows:

$$\frac{\mathrm{Obj}_{\mathrm{Onestep}} - \mathrm{LB}}{LB} \times 100,$$

where $\mathrm{Obj}_{\mathrm{Onestep}}$ is the objective function value obtained by the One-step heuristic. As can be seen, except for the first small instance it is significantly faster, especially for the last three instances with slightly higher gaps. For the comparison, we have also implemented PSO for IPPWLP and results are summarized in Table 4. As we see, in terms of gaps, PSO is better just for the two small instances and in terms of time the One-step heuristic is significantly faster. In the next section, we further improve the One-step heuristic.

Table 4: Performance of One-step heuristic.

Instances	LB	$\text{Obj}_{\text{Onestep}}$	GAP[%]	$\text{Time}_{\text{Onestep}}$[s]	Obj_{PSO}	Time_{PSO}[s]	GAP_{PSO}[%]
Small_1	952.1691	955.2202	0.32	1.25	952.1691	5.26	0
Small_2	2373.1	2412.1	1.64	2.30	2397.78	30.28	1.04
Medium_1	7443.4	7628.4	2.48	6.10	7690.52	35.90	3.32
Medium_2	33882.1578	35037.3731	3.4	14.20	35711.7943	221.00	5.4
Large	46773.5414	49279.5212	5.35	31.80	50510.7473	514.00	7.99

4 Improved One-step heuristic

Here, first we sort items in descending order in terms of $COI_i + DRT_i$ introduced in the previous section. Then we categorize sorted items successively into groups of 3, 4, or 5 in a row that may also overlap with each other. We denote the set of indices of the items in that group with I', and according to the solution obtained from the One-step heuristic, we denote the set of indices of the storage locations reserved for each group of items and empty storage locations with L'. Then we solve the following problem for each group:

$$
\begin{aligned}
\min\ Z = &\sum_{i\in I'}\sum_{l\in L'} R^l z^{il} + \sum_{i\in I'}\sum_{l\in L'}\sum_{t=1}^{T}\Big((P^l + c_t^i)w_t^{il} + O^l(w_t^{il} + n_{t-1}^{il} - n_t^{il}) + h_t^i n_t^{il}\Big) + \sum_{i\in I'}\sum_{t=1}^{T}(u_t^i y_t^i) \\
\text{s.t.}\ &\sum_{i\in I'} z^{il} \le 1, \quad \forall l \in L', \\
&\sum_{i\in I'} w_t^{il} \le 1, \quad \forall t, \forall l \in L', \\
&\sum_{i\in I'} n_t^{il} \le 1, \quad \forall t, \forall l \in L', \\
&\sum_{i\in I'}\sum_{t=1}^{T} q_t^{il} = d_t^i, \quad \forall t, \forall l \in L', \\
&w_t^{il} + n_{t-1}^{il} \le z^{il}, \quad \forall t, \forall i \in I', \forall l \in L', \\
&n_t^{il} = w_t^{il} - q_t^{il} + n_{t-1}^{il}, \quad \forall t, \forall i \in I', \forall l \in L', \\
&\sum_{l\in L'} w_t^{il} \le M y_t^i, \quad \forall t, \forall i \in I', \\
&z^{il}, w_t^{il}, q_t^{il}, n_t^{il}, y_t^i \in \{0,1\}, \quad \forall i, \forall l, \forall t.
\end{aligned}
\tag{25}
$$

Two strategies can be used to implement the above method. The first strategy is to solve problem (24) after each time we solve problem (25) for a group of items that are likely to cause a change in the structure of the z related to the reserved locations for the items in that group. Then if the solution of (24) is better than the last available solution, the new structure obtained will be saved for the location of the items of the current group. Otherwise, no changes will be applied, and we move to the next group of items. This is summarized in Algorithm 2. The computational results of this strategy are reported in Table 5. The gap here is computed as follows:

$$\frac{\text{Obj}_{s1} - LB}{LB} \times 100,$$

where Obj_{s1} denotes the objective function value of the first strategy and LB lower and upper bounds of model (18).

Table 5: Performance of Improved One-step heuristic with the first strategy.

Case	LB	Obj_{s1}	GAP[%]	Time[s]
Small_1	952.1691	954.0566	0.1965	4.01
Small_2	2373.1	2387.8	0.6	8.80
Medium_1	7443.4	7544	1.31	61.54
Medium_2	33882.1578	35001.391	3.3	303.31
Large	46773.5414	49200.5078	5.18	1050.02

As can be seen, the gaps of this approach are less than those of the One-step heuristic, where in the first three instances it is more effective than the last two instances. Although the solution time has increased, but still much better compared to the time in Table 3. Compared to PSO, the gaps here are always better except for the first small instance, and times are worst for the medium and large instances.

Algorithm 2 : Improved One-step heuristic with the first strategy

Input: S (number of groups)

1: Obtain matrices z, q, w and n by the One-step heuristic and set $Bestopt$ equal to the objective value of the One-step heuristic.
2: $Item \leftarrow$ arrange $COI_i + DRT_i$ for items in descending order.
3: **for** $j \leftarrow 1$ to S **do**
4: Solve model (25) for jth group of items to get a new structure for z .
5: Solve model (24) to get new values for n, q, w and y and set $newopt$ equal to the objective value of (24).
6: **if** $newopt < Bestopt$ **then**
7: Save the values z, q, w, n and y obtained from Steps 4 and 5 as the new values of these variables.
8: $Bestopt \leftarrow newopt$.
9: **end if**
10: **end for**

In the second strategy, which is appropriate for larger instances, solving problem (25) is not done by changing the structure for each group after each time. We solve problem (24) for groups of 3, 4 or 5 items successively, and at the end with the structure obtained for the matrix z, we solve (25). This method is random in nature and therefore may sometimes lead to good improvements to the One-step heuristic, but it is suitable for large-scale problems. This strategy is summarized in Algorithm 3. In Table 6, we present the results of this strategy, where Obj_{s2} denotes the objective function value of the second strategy

Table 6: Performance of Improved One-step heuristic with the second strategy

Instances	LB[-]	Obj_{s2}[-]	GAP[%]	Time_{s2}[s]
Medium_2	33882.1578	34989.884	3.26	230.27
Large	46773.5414	49161.3676	5.10	368.17

As one can see in Table 6, we have slightly better gaps compared to Table 5 and much better solution time. Compared to PSO also, the solution time of the large instance is better, while gaps of both instances are better for the proposed heuristic.

Algorithm 3 : Improved One-step heuristic with the second strategy

Input: S (number of groups), Iteration (maximum number of iterations)

Obtain matrices z, q, w and n by the One-step heuristic and set $Bestopt$ equal to the objective value of the One-step heuristic.
for $iter \leftarrow 1$ to $Iteration$ **do**
  $Item \leftarrow$ arrange $COI_i + DRT_i$ for items in descending order.
  for $j \leftarrow 1$ to S **do**
    Solve model (25) for jth group of items to get a new structure for z.
  end for
  Solve model (24) to get new values for n, q, w and y and set $newopt$ equal to the objective value of (24).
  if $newopt < Bestopt$ **then**
    Save the values z, q, w, n and y obtained from Steps 5 and 7 as the new values of these variables.
    $Bestopt \leftarrow newopt$.
  end if
end for

5 Conclusions

We proposed two heuristic algorithms for solving MILP formulation of IPPWLP. The first one is designed based on allocating warehouse locations with less reservation costs, and also less transportation costs from the production area to locations and from locations to the output point to items with higher demands. In the second approach, first items are sorted in descending order according to the fraction of sum of the demands for that item in the time horizon plus the maximum demand for that item in the time horizon and sum of its demands in the time horizon. Then the sorted items are categorized into groups of 3, 4, or 5 in a row that may also overlap with each other, and a small-scale optimization problem for each group is solved, hoping to improve the solution of the first step. Our preliminary computational experiments showed that the two heuristics are significantly faster compared to Gurobi with slightly higher gaps. Compared to PSO, the One-step heuristic is always faster and has better gaps except for the first two small instances. Other improvements of the One-step heuristic are proposed further reducing the gaps and solution time. Incorporating other heuristics to further reduce the gaps and solution time may be considered as future research direction.

Acknowledgements

The authors would like to thank the reviewer for useful comments and suggestions.

References

[1] A. Akbalik, B. Penz, C. Rapine, *Capacitated lot sizing problems with inventory bounds*, Ann. Oper. Res. **229** (2015) 1–18.

[2] N. Brahimi, N. Absi, S. Dauzre-Prs, S. Kedad-Sidhoum, *Models and Lagrangian heuristics for a two-level lot-sizing problem with bounded inventory*, OR Spectr. **37** (2015) 983–1006.

[3] G.R. Buschkuhl, F. Sahling, S. Helber, H. Tempelmeier, *Dynamic capacitated lotsizing problems: a classification and review of solution approaches*, OR Spectr. **32** (2010) 231–261.

[4] D.M. Carvalho, M.C.V. Nascimento, *Lagrangian heuristics for the capacitated multi-plant lot sizing problem with multiple periods and items*, Comput. Oper. Res. **71** (2016) 137–148.

[5] A. Diabat, J.P. Richard, C.W. Codrington, *A Lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design*, Ann. Oper. Res. **203** (2013) 55–80.

[6] J. Gutierrez, A. Sedeno-Noda, M. Colebrook, J. Sicilia, *A new characterization for the dynamic lot size problem with bounded inventory*, Comput. Oper. Res. **35** (2002) 2162–2174.

[7] S.A. Hoseini-Shekarabi, A. Gharaei, M. Karimi, *Modelling and optimal lot-sizing of integrated multi-level multi-wholesaler supply chains under the shortage and limited warehouse space: generalised outer approximation*, Int. J. Syst. Sci. Oper. Logis. **6** (2019) 237–257.

[8] H.C. Hwang, W. Van den Heuvel, A.P. Wagelmans, *The economic lot-sizing problem with lost sales and bounded inventory*, IIE Trans. **45** (2013) 912–924.

[9] B. Karimi, S.M.T. Fatemi Ghomi, J.M. Wilson, *The capacitated lot-sizing problem: a review of models and algorithms*, Omega **31** (2003) 365–378.

[10] J. Kennedy, R. Eberhart, *Particle swarm optimization*, Proc. IEEE Int. Conf. Neu. Net. Vol IV. (1995) 1942–1948. doi:10.1109/ICNN.1995.488968.

[11] S.F. Love, *A facilities in series inventory model with nested shedules*, Manag. Sci. **18.** (1972) 327–338.

[12] E. Page, R.J. Paul, *Multi-product inventory situations with one restriction*, J. Oper. Res. Soc. **27** (1976) 815–834.

[13] A. Witt, *A heuristic for the multi-level capacitated lot sizing problem with inventory constraints*, Int. J. Manag. Sci. Eng. Manag. **14** (4). (2019) 240252.

[14] Z. Zhang, H. Jiang, X. Pan, *A Lagrangian relaxation based approach for the capacitated lot sizing problem in closed-loop supply chain*, Int. J. Prod. Econ. **140** (2012) 249–255.

[15] Z. Zhang, X. Yan, S. Li, *Analysis of a dynamic lot-sizing problem with production capacity constraint*, Int. Trans. Oper. Res. **23**. (2016) 813–833.

[16] G. Zhang, T. Nishi, S.D.O. Turner, K. Oga, X. Li, *An integrated strategy for a production planning and warehouse layout problem: modeling and solution approaches*, Omega **68** (2017) 85–94.

[17] G. Zhang, X. Shang, F. Alawneh, Y. Yang, T. Nishi, *Integrated production planning and warehouse storage assignment problem: An IoT assisted case*, Int. J. Prod. Econ. **234** (2021) 108058.