Operational Integration of Supply Chain Activities with Earliness and Tardiness Considerations

Ece Yağmur(✉) and Saadettin Erhan Kesen

Konya Technical University, Konya, Turkey
{ecyagmur,sekesen}@ktun.edu.tr

Abstract. The operational integration of the production and distribution activities in a supply chain requires concurrent consideration of two major problems known as production scheduling and vehicle routing. While the production environment of the integrated problem under study consists of identical parallel machines, the distribution environment has a limited number of homogeneous vehicles that deliver completed orders to customers. The objective function is to minimize the sum of total earliness and tardiness according to the time windows specified by each customer. In this paper, we first describe the integrated problem as a Mixed Integer Programming (MIP) model and subsequently present an Iterated Local Search (ILS) algorithm to find optimal or near-optimal solutions in a reasonable time. The performance of the ILS is evaluated against the MIP-based solutions obtained by applying a standard solver like CPLEX. According to the computational findings, the suggested ILS can find optimal or near optimal solutions for randomly generated test instances in just a few seconds.

Keywords: parallel machine scheduling · vehicle routing · iterated local search · earliness · tardiness

1 Introduction

Today, the management of supply chain operations is a trending topic for many businesses where competition is of utmost importance. Thus, it is inevitable to provide effective coordination between the production and distribution phases of the supply chain. Fulfilling customer demands with the required quantity on time becomes an extremely difficult task, especially for companies which apply the zero-inventory model and make-to-order businesses. The integrated approach is also beneficial for time management of time-sensitive or perishable products. Considering all these aspects, operational integration for the two main stages is carried out by making decisions simultaneously for production scheduling and vehicle routing. According to the integrated approach, as the output of either does not affect the other, the decisions on production and distribution stages will not limit each other.

Recently, there has been a remarkable growth in the online shopping sector [2]. Due to the abundance of competitors in the market and the ease of communication between

J. R. Daduna et al. (Eds.): ICCL 2023, LNCS 14239, pp. 437–449, 2023.
https://doi.org/10.1007/978-3-031-43612-3_27

customers and companies, firms are under increasing pressure to ensure customer satisfaction. For this reason, they have turned to customer-oriented performance measures rather than classical measures such as cost and profit. According to the current applications, customers know when they can receive the orders and even specify their own preferences regarding delivery time, which is often handled by time windows in the vehicle routing literature.

A current survey, which covers the enlarged literature on integrated production scheduling and outbound vehicle routing problems, can be found in [4]. In the survey, the literature is classified in terms of the number of operations, the number of vehicles, and the number of trips. According to this classification, the problem under study can be represented in a section of parallel machines, several vehicles, and single trips, as in the studies of [3, 5, 6, 11, 12, 18].

Although there are several criteria (i.e., objectives) identified by researchers, in general, objectives can be divided into two main groups: cost and service. In addition, sometimes multiple criteria can be handled simultaneously as a summation of objectives in a single form [5, 9, 21, 22] or multi-objective structure [7, 14, 23]. While many criteria that are related to the problem under study have been defined in the literature [4], the objective of this study can be expressed as minimizing the sum of total earliness and tardiness. Ullrich [18] considers a variant of the integrated problem with the objective of minimizing the total tardiness and proposes a genetic algorithm to produce optimal or near optimal solutions in a reasonable amount of time. Wu et al. [20] formulate a mathematical model for a production scheduling-based routing problem with time windows and setup times for minimizing total tardiness. Hou et al. [10] address an integrated problem where the production stage has multiple factories, and each factory has multiple machines based on the flow shop environment. For the objective of minimizing total weighted earliness and tardiness, they propose a swarm intelligence approach which is called brainstorm optimization algorithm.

In the integrated problem under investigation, the distribution phase has a homogeneous fleet that can only be utilized once, while the production environment consists of identical parallel machines and each job has a single operation that needs to be performed on any machines. It is very important for customer satisfaction to deliver orders according to the predefined time windows. In addition, vehicles that arrive earlier must wait idle until the lower time bound of any customer. So, the objective is to minimize the sum of total earliness and tardiness. After formulating the problem, we propose a metaheuristic approach, Iterated Local Search (ILS), to obtain good quality solutions in a reasonable time.

The contribution of the paper can be summarized as follows: (*i*) The sum of the total earliness and tardiness is studied simultaneously for the first time for the integrated problem at hand. (*ii*) A new mixed integer programming formulation is developed. (*iii*) A metaheuristic algorithm, Iterated Local Search (ILS), is presented, and a perturbation mechanism is proposed for the only production environment adapted for the integrated problem.

The remainder of the study is organized as follows: In Sect. 2, we define the MIP formulation for the integrated problem with an illustrative example. While in Sect. 3, the proposed ILS structure is given, in Sect. 4, computational experiments are discussed based on the different problem parameters. Finally, in Sect. 5, conclusions and future directions are discussed.

2 Problem Definition

This section is dedicated to two subsections including the MIP model and an illustrative example for visual presentation.

2.1 MIP Model

The assumptions of the problem can be summarized as follows: There is a single production facility in which a set of identical parallel machines is used to produce customer orders. After the production of a particular batch is completed, a limited number of vehicles is delivering the consolidated orders to associated customers by considering the time window of each customer. Vehicles can be used only once (i.e., multiple use is not allowed). Vehicle capacities are homogeneous, and the capacity of a vehicle is enough to serve all customers. As a result, there is no requirement to use all vehicles in the system. The sequence of production and distribution does not have to be identical. The objective is to minimize the sum of total earliness and tardiness penalties. The notations used in the MIP formulation are shown in Table 1.

Based on the formal definition and notations, we formulate an MIP model for the problem as follows:

$$minimize \sum_{j \in N_C} \left(E_j + T_j\right) \tag{1}$$

Subject to;

$$\sum_{j \in N_C} W_{jm} \leq 1 \qquad \forall m \in M \tag{2}$$

$$\sum_{i \in N_C} Z_{ji} + L_j = 1 \qquad \forall j \in N_C \tag{3}$$

$$\sum_{m \in M} W_{jm} + \sum_{i \in N_C} Z_{ij} = 1 \qquad \forall j \in N_C \tag{4}$$

$$C_j \geq p_j - H\left(1 - W_{jm}\right) \qquad \forall j \in N_C; m \in M \tag{5}$$

$$C_j \leq p_j + H\left(1 - W_{jm}\right) \qquad \forall j \in N_C; m \in M \tag{6}$$

$$C_i - C_j + HZ_{ij} + \left(H - p_i - p_j\right)Z_{ji} \leq H - p_j \qquad \forall i, j \in N_C; i \neq j \tag{7}$$

$$\sum_{j \in N_C} X_{0jk} \leq 1 \qquad \forall k \in K \tag{8}$$

Table 1. Notations used in the MIP model.

Notation	**Definition**
Indices:	
i, j, u	: Job (order, customer)
m	: Machine
k	: Vehicle
Sets:	
N_C	: Set of jobs (orders, customers)
N	: Set of nodes (including depot node)
M	: Set of machines
K	: Set of vehicles
Parameters:	
d_j	: Demand of customer j
p_j	: Processing time of order j
s_j	: Service time at customer j
t_{ij}	: Travel time between customers i and j
Q	: Capacity of any vehicle
H	: Sufficiently large number
a_j	: Lower bound of time windows for customer j
b_j	: Upper bound of time windows for customer j
Variables:	
Z_{ij}	: 1 if job i is processed just before job j, 0 otherwise.
X_{ijk}	: 1 if vehicle k goes directly from node i to node j, 0 otherwise.
W_{jm}	: 1 if job j is the first to be processed on machine m, 0 otherwise.
L_j	: 1 if job j is the last to be processed on any machine, 0 otherwise.
F_{ij}	: Total amount of load when a vehicle goes directly from node i to node j.
C_j	: Production completion time of order j.
V_k	: Service starting time of vehicle k from the depot.
Y_j	: Arrival time at customer j.
Y_j^*	: Service starting time at customer j.
E_j	: Earliness for customer j.
T_j	: Tardiness for customer j.
G_{ij}	: 1 if job i is processed before job j, 0 otherwise.
O_j	: 1 if earliness occurs for job j, 0 otherwise.
R_{jk}	: 1 if job j is served by the vehicle k, 0 otherwise.
$AX1_{ijk}$	$= R_{ik}R_{jk}$
$AX2_{ij}$	$= G_{ij}\sum_{k\in V} AX1_{ijk}$
$AX3_{ijk}$	$= R_{ik}AX2_{ij}$
$AX4_j$	$= Y_jO_j$

$$\sum_{k\in K}\sum_{i\in N} X_{ijk} = 1 \qquad \forall j \in N_C \tag{9}$$

$$\sum_{u \in N} X_{ujk} = \sum_{i \in N} X_{jik} \qquad \forall j \in N; \forall k \in K \tag{10}$$

$$\sum_{u \in N} F_{uj} - \sum_{i \in N} F_{ji} = d_j \qquad \forall j \in N_C \tag{11}$$

$$d_j \sum_{k \in K} X_{ijk} \leq F_{ij} \qquad \forall i, j \in N; i \neq j \tag{12}$$

$$Q - d_i + \sum_{k \in K} X_{ijk} \geq F_{ij} \qquad \forall i, j \in N; i \neq j \tag{13}$$

$$R_{jk} = \left(\sum_{i \in N_C} X_{jik} \right) + X_{j0k} \qquad \forall j \in N_C; \forall k \in K \tag{14}$$

$$C_i - C_j \leq H\left(1 - G_{ij}\right) \qquad \forall i, j \in N_C; i \neq j \tag{15}$$

$$C_j - C_i \leq HG_{ij} \qquad \forall i, j \in N_C; i \neq j \tag{16}$$

$$V_k \geq C_i - H\left(1 - X_{ijk}\right) \qquad \forall i, j \in N; i \neq j; \forall k \in K \tag{17}$$

$$\begin{aligned} V_k \leq C_j + H(1 - (R_{jk} - \sum_{i \in N_C} AX3_{jik} \ \forall j \in N_C; \forall k \in K \\ + \sum_{i \in N_C} \left(G_{ij} + G_{ji} - 1\right))) \end{aligned} \tag{18}$$

$$V_k - Y_j + s_0 + t_{0j} \leq H\left(1 - X_{0jk}\right) \qquad \forall j \in N_C; \forall k \in K \tag{19}$$

$$Y_j - V_k - s_0 - t_{0j} \leq H\left(1 - X_{0jk}\right) \qquad \forall j \in N_C; \forall k \in K \tag{20}$$

$$Y_i^* - Y_j + s_i + t_{ij} \leq H\left(1 - \sum_{k \in K} X_{ijk}\right) \qquad \forall i, j \in N_C; i \neq j \tag{21}$$

$$Y_j - Y_i^* - s_i - t_{ij} \leq H\left(1 - \sum_{k \in K} X_{ijk}\right) \qquad \forall i, j \in N_C; i \neq j \tag{22}$$

$$T_j \geq Y_j^* - b_j \qquad \forall j \in N_C \tag{23}$$

$$E_j \geq a_j - Y_j \qquad \forall j \in N_C \tag{24}$$

$$Y_j - a_j \leq H\left(1 - O_j\right) \qquad \forall j \in N_C \tag{25}$$

$$a_j - Y_j \leq HO_j \qquad \forall j \in N_C \tag{26}$$

$$Y_j^* = Y_j - AX4_j + a_j O_j \qquad \forall j \in N_C \tag{27}$$

The objective function minimizing the sum of total earliness and tardiness is given in (1). It is determined whether a job is the successor of any job on the same machine or the first/last job to be processed on any machine by Cons. (2)–(4). While Cons. (5) and (6) are used to determine the production completion time of the first jobs on each machine, Cons. (7) defines the production completion time for predecessor and successor jobs on each machine. Cons. (8) identifies the first customer on any tour. Cons. (9) and (10) guarantee that each customer must be visited exactly once, and the number of entering and leaving arcs must be the same for any node, respectively. Cons. (11)–(13) are based on a single-commodity flow formulation described by [8]. This set of constraints guarantee a connected tour. So, we define F_{ij} as a continuous variable indicating the amount of load flowing between nodes i and j. In order to assure subtour elimination in the formulation, we impose tight bounds on the flow variables F_{ij} in addition to a set of flow conservation constraints. Cons. (11) and (12) indicate that the total amount of load on any vehicle decreases as it delivers the order to the related customer. According to Cons (13), the total amount of load on any vehicle should not exceed its capacity. While Cons. (14) defines the variable of R_{jk}, , Cons. (15) and (16) define the variable of G_{ij}. . According to Cons. (17) and (18), each vehicle must wait in production facility until the production process of the batch assigned completes. Cons. (19) and (20) specify the arrival time to the first customer on any tour. Cons. (21) and (22) state the arrival time to the two consecutive customers on any tour. Cons. (23) and (24) calculate tardiness and earliness, respectively, if either of them occurs. Cons. (25) and (26) define the variable of O_j. . According to Cons. (27), the service starting time of any customer is equal to the maximum value of the lower bound of the time windows and arrival time for that customer.

As can be seen, the developed model is not linear because it consists of the product of two variables. So, while we use the Cons. (28)–(30) for $AX1_{ijk}$, $AX2_{ij}$ and $AX3_{ijk}$ where both x and y are binary variables, Cons. (31)–(33) are used for the linearization of $AX4_j$ where x is a binary variable and y is a non-negative continuous variable, and M is a sufficiently large number.

$$z \leq x \tag{28}$$

$$z \leq y \tag{29}$$

$$z \geq x + y - 1 \tag{30}$$

$$z \leq xM \tag{31}$$

$$z \leq y \tag{32}$$

$$z \geq y - (1 - x)M \tag{33}$$

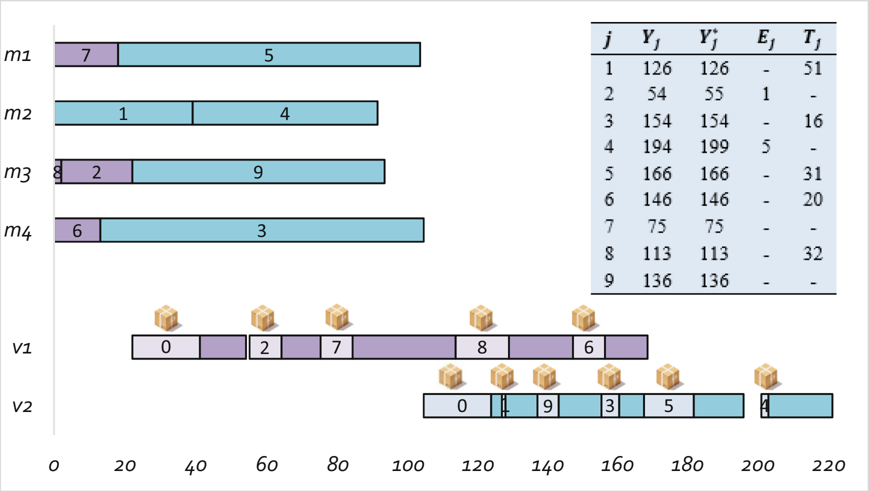
2.2 Illustrative Example

Based on the formal definition given in the previous section, we present an illustrative problem for visualizing the integrated approach. In the example, whose parameters are listed in Table 2, there are nine customers, four identical parallel machines, and two homogenous vehicles with enough capacity to serve all customers. Table 1 contains the descriptions of parameters listed in the first row of Table 2.

Table 2. Parameters of an illustrative example

j	p_j	s_j	a_j	b_j	t_{ij}	*0*	*1*	2	*3*	*4*	*5*	*6*	*7*	*8*	9
0	–	19	–	–	*0*	0	3	13	5	18	10	12	22	18	10
1	39	1	69	75	*1*	3	0	13	7	19	12	10	23	16	9
2	20	9	55	61	*2*	13	13	0	14	30	19	22	11	19	5
3	91	5	128	138	*3*	5	7	14	0	18	7	15	21	23	12
4	52	2	199	201	*4*	18	19	30	18	0	14	13	38	31	27
5	85	14	114	135	*5*	10	12	19	7	14	0	16	26	27	18
6	13	9	108	126	*6*	12	10	22	15	13	16	0	33	18	18
7	18	9	61	106	*7*	22	23	11	21	38	26	33	0	29	16
8	2	15	54	81	*8*	18	16	19	23	31	27	18	29	0	15
9	71	6	132	139	*9*	10	9	5	12	27	18	18	16	15	0

A feasible solution for the given example is illustrated in Fig. 1. According to Fig. 1, jobs 7 and 5 are processed on the first machine. Jobs assigned to other machines are depicted similarly. While customers 2, 7, 8, and 6 are served by the first vehicle, the second vehicle delivers orders 1, 9, 3, 5, and 4, respectively. Shipping boxes show the service time spent on each customer. Vehicle one (v1) departs from the depot when the production process of the first batch completes at time 22, then goes directly from the depot to customer 2 at time 54. Based on the time window of customer 2, the vehicle must wait 1 min until the lower bound of the time window (i.e., 55). This process proceeds in the same way for other customers. Based on the following feasible solution, earliness occurs for customers 2 and 4, while tardiness occurs for customers 1, 3, 5, 6, and 8, as seen from Fig. 1. So, the objective value of the problem is equal to the 156 which is a summation of 6 earliness and 150 tardiness.



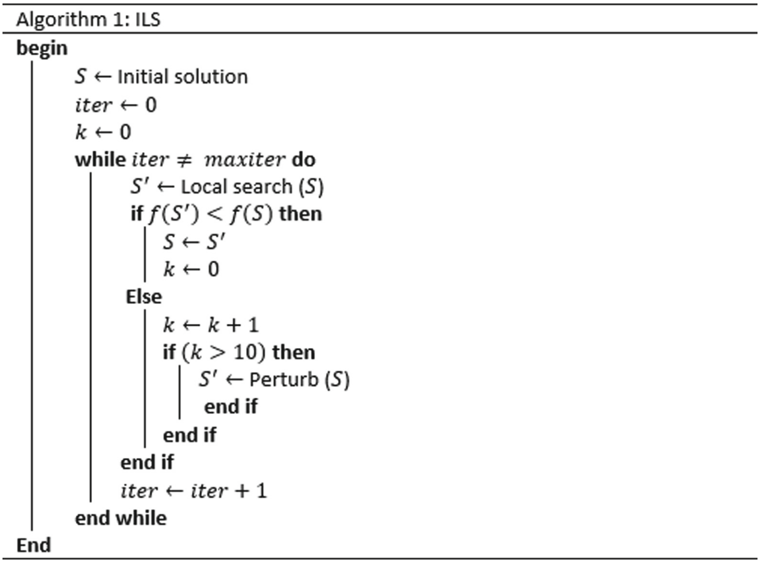
j	Y_j	Y_j^*	E_j	T_j
1	126	126	-	51
2	54	55	1	-
3	154	154	-	16
4	194	199	5	-
5	166	166	-	31
6	146	146	-	20
7	75	75	-	-
8	113	113	-	32
9	136	136	-	-

Fig. 1. A Gantt chart of a feasible solution (also optimal solution) for the illustrated example.

3 Iterated Local Search

The idea of Iterated Local Search (ILS) as an extension of the local search procedure was first introduced in [13]. Basically, ILS is formed by incorporating a perturbation mechanism into the local search process to avoid getting stuck in local optima. ILS has been used for many combinatorial optimization problems in the literature [15—17, 19]. The pseudo code of the proposed ILS is given in Fig. 2.

In the perturbation process, we use the regeneration operator proposed by [1]. In the selection process, parents are selected as the current local optimal solution and the neighbor solution with the best objective. The encoding scheme developed by Afralizad and Rezaeian [1] is also adapted to the integrated problem. According to the adapted chromosome, while the first and third rows represent the permutation of the jobs and customers, respectively, the second and fourth rows demonstrate the assignments of machines and vehicles for each job and customer in the corresponding position. The encoding scheme for the illustrated problem is depicted in Fig. 3. In the local search process, we use swap, insert, and 2-opt operators for both production and distribution phases so as to consider inter and intra machines and vehicles.



```
Algorithm 1: ILS
begin
    $S \leftarrow$ Initial solution
    $iter \leftarrow 0$
    $k \leftarrow 0$
    while $iter \neq maxiter$ do
        $S' \leftarrow$ Local search ($S$)
        if $f(S') < f(S)$ then
            $S \leftarrow S'$
            $k \leftarrow 0$
        Else
            $k \leftarrow k + 1$
            if $(k > 10)$ then
                $S' \leftarrow$ Perturb ($S$)
            end if
            end if
        end if
        $iter \leftarrow iter + 1$
    end while
End
```

Fig. 2. The pseudocode of the proposed ILS.

Job order:	7	5	1	4	8	2	9	6	3
Machine assignment:	1	1	2	2	3	3	3	4	4
Customer order:	2	7	8	6	1	9	3	5	4
Vehicle assignment:	1	1	1	1	2	2	2	2	2

Fig. 3. The encoding scheme

4 Computational Results

For computational experiments, we generate a total of 486 instances randomly based on the number of customers, machines, and vehicles. While the number of customers includes six levels from 5 to 10, machines and vehicles are set to three levels covering 2, 3, and 4. In addition, for each factor level of the parameters, nine different random instances are generated.

The CPLEX solver embedded in GAMS is used as the MIP-solver. The maximum computational time for the CPLEX solver is set to 10800 s, and the stopping criterion for ILS is run 1000 × N iterations, where N represents the number of customers.

Table 3 reports the comparative results of CPLEX and ILS based on the problem parameters. The column *Parameters* describes the level of problem parameters based on the number of customers, machines, and vehicles, respectively. In the *CPLEX* column, while *Obj.* Indicates the average objective value (i.e., upper bound), *LB* represents the lower bound found by CPLEX in a given time limit of 3 h. The number of optimal solutions found by CPLEX and ILS are given in column *#Opt*. Finally, the solution times in seconds for CPLEX and ILS are indicated by the *CPU* column. The deviation of objective values from each other (*DEV%*) is calculated by $\frac{Obj_{CPLEX}-Obj_{ILS}}{Obj_{CPLEX}} \times 100$ if the solution found by CPLEX is worse than ILS; otherwise, it is calculated by $\frac{Obj_{ILS}-Obj_{CPLEX}}{Obj_{ILS}} \times 100$.

We summarize the following remarks from Table 3: CPLEX can find all optimal solutions for all instances of six and seven customers. As the number of customers increases, the ability of CPLEX to find optimal solutions declines dramatically. For ten customers, CPLEX can only find seven optimal solutions out of 81 instances. Like the first remark, the CPU time also increases drastically when the number of customers increases from 5 to 10 and reaches to about 10,000 s for the level of 10 customers. In addition, when the effect of the number of machines and vehicles are examined, we reach the following results: As the number of machines increases, CPLEX's ability to find optimal solutions increases. This is a result of the reduction of the tardiness component in the objective function due to the simultaneous use of resources in the production phase. Similarly, as the number of vehicles increases, the average objective value decreases. However, the number of vehicles has not as much an effect as the number of machines in terms of the number of optimal solutions found and CPU times. As can be seen from Table 3, CPLEX is able to find the optimal solution in 331 instances out of 486 instances within 3 h. When the same instances are solved with ILS, it is seen to provide optimal solutions for 329 instances out of 331 instances whose optimality are proven by CPLEX. Especially in instances where the number of customers is set to 9 and 10, the solutions obtained by ILS are of higher quality, and the percent deviation of CPLEX from ILS (*DEV%)* reaches to 77%. As a result, ILS has shown superior performance in terms of both solution quality and solution time.

Table 3. The comparison of CPLEX and ILS based on the number of customers, machines, and vehicles.

	CPLEX					*ILS*			
N	***Obj.***	***LB***	***DEV%***	***#Opt.***	***CPU***	***Obj.***	***DEV%***	***#Opt.***	***CPU***
5	22.98	22.98	0	81/81	1.84	22.98	0	81/81	3.29
6	35.88	35.88	0	81/81	58.25	35.88	0	81/81	3.89
7	72.35	64.65	0	76/81	1594.47	72.35	0	76/76	4.57
8	117.43	39.76	18.45	50/81	4811.80	111.43	0	50/50	5.31
9	158.74	26.37	38.08	36/81	7047.14	140.07	0.10	35/36	6.11
10	293.09	9.66	77.34	7/81	9990.05	227.17	0.15	6/7	7.93
M	***Obj.***	***LB***	***DEV%***	***#Opt.***	***CPU***	***Obj.***	***DEV%***	***#Opt.***	***CPU***
2	230.29	46.94	35.09	81/162	5810.84	197.57	0.00	81/81	4.89
3	77.67	28.85	20.23	117/162	3594.04	67.88	0.05	116/117	5.14
4	42.28	23.86	11.62	133/162	2346.89	39.50	0.07	132/133	5.52
K	***Obj.***	***LB***	***DEV%***	***#Opt.***	***CPU***	***Obj.***	***DEV%***	***#Opt.***	***CPU***
2	153.83	51.50	20.91	113/162	3857.39	137.50	0.00	113/113	5.02
3	104.69	28.70	22.71	109/162	3916.62	88.55	0.07	108/109	5.14
4	91.71	19.45	23.31	109/162	3977.76	78.89	0.05	108/109	5.39
Avg.	***116.75***	***33.22***	***22.31***	***331/486***	***3917.26***	***101.65***	***0.04***	***329/331***	***5.18***

5 Final Remarks and Future Directions

In this study, we develop a new MIP model for a variant of the integrated production and distribution problem that consists of parallel machine scheduling and homogeneous vehicle routing problems with time windows. The objective of the model is formed by minimizing the sum of total earliness and tardiness. The performance of the exact method is found to be poor, even for only seven customers. We also examine the performance of CPLEX based on the number of customers, machines, and vehicles. According to the results, the increase in the number of customers reduces the performance of CPLEX dramatically. Especially for nine and ten customers, the exact method can no longer be considered a reasonable solution method for an operational-level problem. On the contrary, the increase in the number of machines positively affects the performance of CPLEX. The reason for this is that the departure times of the vehicles are moved earlier due to the simultaneous use of identical parallel machines in the production environment. Thus, the total tardiness in the objective function tends to decrease. When the number of vehicles increases, we see that the objective function value decreases, but there is no significant difference when CPU time and the number of optimal solutions found is examined. For a customer number of 10, CPLEX can only find optimality in 7 out of 81 instances in a given time limit of 10800 s, and the deviation between the upper and lower bounds found by CPLEX is calculated at 96.7%. Then, we propose the ILS algorithm for solving the integrated problem efficiently. According to the comparative results, the proposed ILS algorithm performs quite well in terms of both solution quality and time. The CPU time of ILS for a level of ten customers is about 8 s, even for 10000 iterations. In addition, ILS is found to produce 77% better solutions in terms of objective function value when compared to CPLEX for ten customers.

Future work may focus on larger instances, and the performance of ILS can be compared to other metaheuristics such as genetic algorithms, particle swarm optimization, etc. By considering specific situations of problem assumptions, such as heterogeneous vehicles, unrelated parallel machines, and sequence dependent set-up times, a new variant of an integrated problem can be handled. In addition, sustainable objectives such as energy-efficient strategies at both the production and distribution levels can be considered.

References

1. Afzalirad, M., Rezaeian, J.: Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. Comput. Ind. Eng. **98**, 40–52 (2016). https://doi.org/10.1016/j.cie.2016.05.020
2. Alzoubi, H., Alshurideh, M., Kurdi, B., Alhyasat, K., Ghazal, T.: The effect of e-payment and online shopping on sales growth: evidence from banking industry. Int. J. Data Netw. Sci. **6**(4), 1369–1380 (2022). https://doi.org/10.5267/j.ijdns.2022.5.014
3. Belo-Filho, M., Amorim, P., Almada-Lobo, B.: An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products. Int. J. Prod. Res. **53**(20), 6040–6058 (2015). https://doi.org/10.1080/00207543.2015.1010744
4. Berghman, L., Kergosien, Y., Billaut, J.-C.: A review on integrated scheduling and outbound vehicle routing problems. Eur. J. Oper. Res. (2023). https://doi.org/10.1016/j.ejor.2022.12.036

5. Chang, Y.-C., Li, V.C., Chiang, C.-J.: An ant colony optimization heuristic for an integrated production and distribution scheduling problem. Eng. Optim. **46**(4), 503–520 (2014). https://doi.org/10.1080/0305215X.2013.786062
6. Farahani, P., Grunow, M., Günther, H.-O.: Integrated production and distribution planning for perishable food products. Flex. Serv. Manuf. J. **24**, 28–51 (2012). https://doi.org/10.1007/s10696-011-9125-0
7. Ganji, M., Kazemipoor, H., Molana, S.M.H., Sajadi, S.M.: A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. J. Clean. Prod. **259**, 120824 (2020). https://doi.org/10.1016/j.jclepro.2020.120824
8. Gavish, B., Graves, S.C.: The travelling salesman problem and related problems (1978). http://hdl.handle.net/1721.1/5363
9. He, P., Li, K., Kumar, P.R.: An enhanced branch-and-price algorithm for the integrated production and transportation scheduling problem. Int. J. Prod. Res. **60**(6), 1874–1889 (2022). https://doi.org/10.1080/00207543.2021.1876941
10. Hou, Y., Fu, Y., Gao, K., Zhang, H., Sadollah, A.: Modelling and optimization of integrated distributed flow shop scheduling and distribution problems with time windows. Expert Syst. Appl. **187**, 115827 (2022). https://doi.org/10.1016/j.eswa.2021.115827
11. Kesen, S.E., Bektaş, T.: Integrated production scheduling and distribution planning with time windows. Lean Green Supply Chain Manage: Optim. Models Algorithms, 231–252 (2019).https://doi.org/10.1007/978-3-319-97511-5_8
12. Liu, H., Guo, Z., Zhang, Z.: A hybrid multi-level optimisation framework for integrated production scheduling and vehicle routing with flexible departure time. Int. J. Prod. Res. **59**(21), 6615–6632 (2021). https://doi.org/10.1080/00207543.2020.1821927
13. Lourenço, H.R., Martin, O.C., Stützle, T., Iterated local search. In: Handbook of Metaheuristics, pp. 320–353. Springer, Cham (2003). https://doi.org/10.1007/0-306-48056-5_11
14. Mohammadi, S., Al-e-Hashem, S.M., Rekik, Y.: An integrated production scheduling and delivery route planning with multi-purpose machines: a case study from a furniture manufacturing company. Int. J. Prod. Econ. **219**, 347–359 (2020). https://doi.org/10.1016/j.ijpe.2019.05.017
15. Penna, P.H.V., Subramanian, A., Ochi, L.S.: An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. J. Heuristics **19**(2), 201–232 (2013). https://doi.org/10.1007/s10732-011-9186-y
16. Stützle, T.: Applying iterated local search to the permutation flow shop problem. Technical Report AIDA-98–04, FG Intellektik, TU Darmstadt (1998)
17. Stützle, T.: Iterated local search for the quadratic assignment problem. Eur. J. Oper. Res. **174**(3), 1519–1539 (2006). https://doi.org/10.1016/j.ejor.2005.01.066
18. Ullrich, C.A.: Integrated machine scheduling and vehicle routing with time windows. Eur. J. Oper. Res. **227**(1), 152–165 (2013). https://doi.org/10.1016/j.ejor.2012.11.049
19. Vansteenwegen, P., Souffriau, W., Berghe, G.V., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. Comput. Oper. Res. **36**(12), 3281–3290 (2009). https://doi.org/10.1016/j.cor.2009.03.008
20. Wu, G.-H., Cheng, C.-Y., Pourhejazy, P., Fang, B.-L.: Variable neighborhood-based cuckoo search for production routing with time window and setup times. Appl. Soft. Comput. **125**, 109191 (2022). https://doi.org/10.1016/j.asoc.2022.109191
21. Yağmur, E., Kesen, S.E.: A memetic algorithm for joint production and distribution scheduling with due dates. Comput. Ind. Eng. **142**, 106342 (2020). https://doi.org/10.1016/j.cie.2020.106342

22. Yağmur, E., Kesen, S.E.: Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: memetic algorithm and simulated annealing approaches. Comput. Ind. Eng. **161**, 107649 (2021). https://doi.org/10.1016/j.cie.2021.107649
23. Yağmur, E., Kesen, S.E.: Bi-objective coordinated production and transportation scheduling problem with sustainability: formulation and solution approaches. Int. J. Prod. Res. **61**(3), 774–795 (2023). https://doi.org/10.1080/00207543.2021.2017054