İSTANBUL UNIVERSITY PRESS

Journal of Transportation and Logistics
JTL 2023, 8 (2), 180–190
DOI: 10.26650/JTL.2023.1338878

RESEARCH ARTICLE

Mathematical Models for Disassembly Line Balancing and Pickup - Delivery Vehicle Routing Problem

Zeynel Abidin Çil[1], Damla Kızılay[2], Hande Öztop[3]

[1](Assoc. Prof. Dr.), Izmir Democracy University, Department of Industrial Engineering, Izmir, Turkiye
[2](Assoc. Prof. Dr.), Izmir Democracy University, Department of Industrial Engineering, Izmir, Turkiye
[3](Dr.), Izmir Democracy University, Department of Industrial Engineering, Izmir, Turkiye

ABSTRACT
This study addresses the integrated disassembly line balancing and pickup-delivery vehicle routing problem of companies with multiple disassembly centers. In a supply chain with distributed disassembly centers, the products to be disassembled must be collected from the factories where they are supplied and brought to the disassembly centers. Then, these collected products must be disassembled in the disassembly centers and these disassembled components must be distributed to the factories that demand the disassembled parts. Since there are distributed disassembly centers, factories that request components and factories that supply products should be assigned to the disassembly centers. This study aims to provide an integrated plan for the assignment, disassembly line balancing and collection-distribution processes. In this study, there are distributed disassembly centers with limited product supplies, and distribution and collection operations are considered together in the vehicle routing problem. The problem differs from the studies in the literature with these features. The simultaneous collection and distribution operations aim to save time and reduce transportation costs of vehicles. A mixed-integer nonlinear programming model, a mixed-integer linear programming model and a constraint programming model are presented to solve the integrated problem. The performance of the mixed-integer linear programming and constraint programming models has been evaluated using small-sized instances, and the computational findings indicate that both models can provide effective solutions for the problem.

Keywords: *Disassembly line balancing, pickup and delivery vehicle routing problem, mixed-integer programming, constraint programming*

1. Introduction

With the developing technology, companies aim to find solution methods that will save time and cost in production and distribution processes. In addition, regarding the environmental regulations and high environmental pollution, companies need to consider environmental factors as well as the production and distribution costs. The importance of the recycling and disassembly processes is increasing due to the increasing consumption rate of natural resources. Consequently, it is expected that the integration of vehicle routing problem (VRP) and disassembly line balancing (DLB) problem can result in more profitable solutions in terms of cost minimization and help to enhance the sustainability of companies.

Besides the disassembly line balancing, this study considers the collection of the products to be disassembled from the factories and the distribution of the components that emerge after the disassembly process to the factories. The focus of this study is the integrated problem of balancing the disassembly lines in a system where there is more than one disassembly center, and the collection of the products to be disassembled for the next planning period from the factories, while the disassembled components that emerge after the disassembly process are distributed to the factories. Therefore, there are three subproblems. The first subproblem is balancing the disassembly lines in the disassembly centers, the second subproblem is the collection of the products to be disassembled for the next planning period with minimum cost, and the third subproblem is the distribution of the components resulting from the disassembly process to the factories with minimum cost. Consequently, this study integrates the disassembly line balancing, collection of products and distribution of components processes across multiple disassembly centers.

Corresponding Author: Zeynel Abidin Çil **E-mail:** zabidin.cil@idu.edu.tr

Submitted: 07.08.2023 • **Revision Requested:** 24.09.2023 • **Last Revision Received:** 13.10.2023 • **Accepted:** 18.10.2023

DLB and VRP have been widely studied separately in the literature (Özceylan et al., 2019; Braekers et al., 2016). There are relatively few studies on integrated DLB and VRP in the literature. Habibi et al. (2014) studied an integrated disassembly problem with the collection of products to be recycled, where the distribution of recycled components was not considered. Diri Kenger et al. (2020a) considered the disassembly line balancing with the distribution of disassembled components, and they proposed mathematical models for the case where there is a single disassembly center. Later, as extensions of that problem, Diri Kenger et al. (2020b) considered a production system with 3D printers to support the disassembly line and Diri Kenger et al. (2021b) considered a distribution system with a fleet of trucks equipped with 3D printer technology. In these studies, the authors made scenario analyses by presenting mathematical models. As far as we know, there is a study in the literature that considers the collection-distribution problem as integrated with disassembly line balancing (Diri Kenger et al., 2021a). Diri Kenger et al. (2021a) proposed a mathematical model for the problem and solved small-sized instances. However, only one disassembly center was considered in that study. Recently Çil et al. (2023) have presented mixed-integer programming models, a constraint programming model and a multi-start simulated annealing algorithm for the integrated distributed disassembly line balancing and vehicle routing problem, where there are multiple disassembly centers. However, in the study by Çil et al. (2023), only the distribution of disassembled components was considered, and the collection of products to be recycled was not considered in the vehicle routing phase. Also, it is assumed that there are no product supply limits for the disassembly centers in the study by Çil et al. (2023). The studied problem in our study differs from the integrated DLB and vehicle routing problems in the literature, since in the studied problem, there are multiple disassembly centers with limited product supplies, and distribution and collection operations are considered together as a pickup-delivery vehicle routing problem.

In real-life problems, there can be more than one disassembly center, and for these disassembly centers, performing the distribution of components and collection of products simultaneously can provide advantages in terms of cost and time. Therefore, different from the related studies in literature, this study integrates the distributed disassembly line balancing problem with the pickup and delivery vehicle routing problem. In the studied integrated problem, both the optimization of the disassembly lines in the disassembly centers and the optimization of the vehicle routes for the collection of products and distribution of components are provided simultaneously. Today, enterprises can establish more than one disassembly center to increase capacity of their systems and to adapt the changes in demand easily. Therefore, in the studied problem, multiple disassembly centers with limited product supplies are considered. In addition, in the studied problem, the collection and distribution processes are planned simultaneously using the same vehicle fleet. To the best of our knowledge, the integrated distributed disassembly line balancing and pickup-delivery vehicle routing problem (DDLB-PDVRP) has not been studied in the literature before. Consequently, this study aims to resolve this research gap existing for practical applications by studying the integrated disassembly line balancing and pickup-delivery vehicle routing problem for the distributed disassembly centers, which has not been studied before in the literature.

In reverse supply chain systems, effective disassembly line balancing and collection-distribution processes are crucial to achieve an effective resource management, reduced disassembly and transportation costs, and a high sustainability. In a dynamic supply chain environment, effective planning for these disassembly and transportation processes is essential for adapting to changing customer demands and product supplies. Integrating disassembly line balancing and pickup-delivery vehicle routing problem for distributed disassembly centers can help to reduce disassembly and transportation costs and provide a better resource management by optimizing disassembly line balancing, product collection, and component distribution operations of all disassembly centers with an integrated approach. Also, in terms of environmental factors, integrating these problems can help to align with sustainability goals. The optimization of the integrated problem can help to reduce waste, enhance resource efficiency, reduce environmental pollution, and support recycling practices, by providing an effective planning for disassembly and transportation processes in a supply chain.

In this study, a mixed-integer nonlinear programming model, a mixed-integer linear programming model and a constraint programming model are presented for the DDLB-PDVRP. Also, the mathematical models are solved on small-sized instances. The rest of the paper is organized as follows. A problem definition is provided in Section 2. The mixed-integer programming models and constraint programming model are presented in Section 3. Section 4 presents the computational results. Finally, conclusions are provided in Section 5.

2. Problem Definition

In the studied problem, products are collected from factories and brought to the disassembly centers with vehicles belonging to these centers, while distributing the disassembled components to the factories that demand the disassembled parts. While the disassembled components of the products are distributed to the factories from the disassembly centers, the vehicles are allowed to have both the collected products and the distributed components in the same vehicle, regarding their capacities. Thus, by considering collection and distribution together, the optimization of the pickup and delivery vehicle routes is ensured. The collected products represent the product supply for the next planning period, and the distributed components represent the component demands for the current planning period. While the vehicles distribute the components to the factories for the current planning period, they also collect the products to be disassembled from the factories for the next planning period. An illustrative representation for the studied problem is provided in Figure 1.

The cycle time of each disassembly center is calculated according to the component demand of the factories assigned to that disassembly center in the current planning period. However, the number of products to be disassembled within a planning period should not exceed the number of existing products that arrived at that disassembly center in the previous planning period. Therefore, there is a capacity constraint for each disassembly center. Each disassembly center can perform product disassembly regarding its limited product supply from the previous planning period.

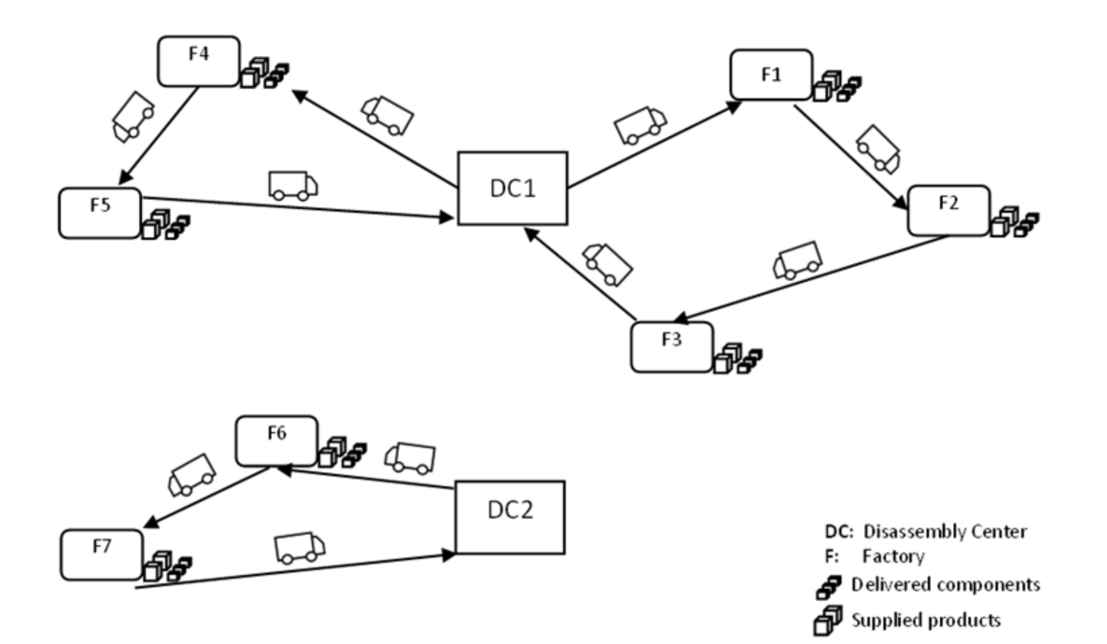

Figure 1. Illustrative representation of the problem

In the DDLB-PDVRP, the disassembly line of each disassembly center should be balanced by assigning tasks to the stations according to the AND/OR precedence graph. While the AND precedence denotes that a task can be began after all its AND predecessors are completed, the OR precedence denotes that a task can be began after at least one of its OR predecessors is completed. When a product is disassembled, one unit of each component type is obtained. The DLB part of the problem aims to minimize the total number of stations regarding the cycle time of each disassembly center. The cycle time of each disassembly center is determined based on the maximum demand of each component type to be sent from that center in the current planning period, i.e., the number of products planned to be disassembled in the current planning period. However, the number of products to be disassembled based on the component demands in the current planning period should not exceed the number of existing products that arrived at that disassembly center in the previous planning period.

In addition to deciding which disassembly center (DC) should collect products from which factory and distribute components to which factory, it is also important to decide on the routes of the vehicles to reduce transportation costs. Therefore, in the DDLB-PDVRP, pickup and delivery routes are also determined for the vehicle fleet of each disassembly center, resulting in a multi-depot pickup and delivery vehicle routing problem. It is assumed that a vehicle can only transport component/product from a single disassembly center and return to its own center after its route, and there are identical vehicles. The component demand and product supply of a factory cannot be divided between disassembly centers.

The DDLB-PDVRP aims to optimize simultaneously the assignment of the factories to disassembly centers, disassembly line balancing of each disassembly center, and the pickup-delivery vehicle routes for each disassembly center, by minimizing the total station and transportation costs. The developed mixed-integer nonlinear programming (MINLP) model, the mixed-integer linear programming (MILP) model, and the constraint programming (CP) model are presented in the following section of the paper, respectively. The sets and parameters used in these models are explained in Table 1.

Table 1. Sets and parameters

Sets
$\boldsymbol{A}$ = Set of tasks
$\boldsymbol{B}$ = Set of stations
$\boldsymbol{DC}$ = Set of disassembly centers {1,2…, *s*}
$\boldsymbol{E}$ = Set of customer/supplier factories {1, 2,…, *n*}
$\boldsymbol{H}$ = Set of nodes {1, 2,…, *n, n+1, …, n+s*}, where {*n+1, …, n+s*} are starting/ending DCs
$\boldsymbol{H'}$ = Set of nodes {1, 2,…, *n, n+1, …, n+s,…, n+2s*}, where {*n+1, …, n+s*} are starting DCs and {*n+s+1, …, n+2s*} are ending DCs
$\boldsymbol{F_d}$ = Set of vehicles for DC *d*
$\boldsymbol{G}$ = Set of components
Parameters
$\boldsymbol{wtime}$ = total working time
$\boldsymbol{ptime_i}$ = processing time of task *i*
$\boldsymbol{demand_{jg}}$ = demand of node *j* for component *g* (demands of DCs are zero)
$\boldsymbol{prod_j}$ = number of products supplied from node *j* for the next planning period (supply of DCs are zero)
$\boldsymbol{scap_d}$ = number of products that have been supplied to DC *d* in the previous planning period
$\boldsymbol{capacity}$ = capacity of a vehicle
$\boldsymbol{travel_{jl}}$ = travelling distance from node *j* to node *l*
$\boldsymbol{cost_{jl}}$ = travelling cost from node *j* to node *l*
$\boldsymbol{scost}$ = cost of a station
$\boldsymbol{AND_i}$ = set of AND predecessors of task *i*
$\boldsymbol{OR_i}$ = set of OR predecessors of task *i*

3. Mathematical Models

Since the studied integrated problem has not been addressed before in the literature, mixed-integer programming and constraint programming models are developed for the problem. There are three different models, and each has some advantages and disadvantages in terms of ease of modeling, solution time, and solution quality within a limited time. All these features of the models are discussed after explaining each model.

3.1. Mixed-Integer Nonlinear Programming Model

In this section, the MINLP model of the problem is presented. The decision variables used in this model are described in Table 2. The objective function and the constraints of the MINLP model are explained as follows.

Table 2. Decision variables for MINLP

$\boldsymbol{U_j}$ = visiting rank of node *j*
$\boldsymbol{v_{ibd}}$ =1, if task *i* is assigned to station *b* in DC *d*; 0 otherwise
$\boldsymbol{x_{jlmd}}$ =1, if vehicle *m* of DC *d* travels from node *j* to node *l*; 0 otherwise
$\boldsymbol{y_{bd}}$ =1, if station *b* is opened in DC *d*; 0 otherwise
$\boldsymbol{np_d}$ = number of products (maximum number of components) that are demanded from DC *d*
$\boldsymbol{cycle_d}$ = cycle time of DC *d*
$\boldsymbol{ld_j}$ = load of vehicle after having serviced node *j*
$\boldsymbol{lb_{md}}$ = load of vehicle *m* of DC *d* when leaving the DC

$$MinimizeZ = \sum_{b\in B}\sum_{d\in DC} scost * y_{bd} + \sum_{d\in DC}\sum_{j,l\in H}\sum_{m\in F_d} cost_{jl} * x_{jlmd} \tag{1}$$

$$\sum_{b\in B} v_{ibd} = 1 \qquad \forall i \in A, d \in DC \tag{2}$$

$$\sum_{i\in A} ptime_i v_{ibd} \le cycle_d y_{bd} \qquad \forall b \in B, d \in DC \tag{3}$$

$$v_{ibd} \le \sum_{c\in B: c\le b} v_{kcd} \qquad \forall i \in A, k \in AND_i, b \in B, d \in DC \tag{4}$$

$$v_{ibd} \leq \Sigma_{k\in OR_i}\Sigma_{c\in B:c\leq b}v_{kcd} \qquad \forall i \in A : OR_i \neq \emptyset, b \in B, d \in DC \tag{5}$$

$$y_{b+1,d} \leq y_{bd} \qquad \forall b, b+1 \in B, d \in DC \tag{6}$$

$$cycle_d = \frac{wtime}{np_d} \qquad \forall d \in DC \tag{7}$$

$$cycle_d \geq ptime_i \qquad \forall d \in DC, i \in A \tag{8}$$

$$\Sigma_{j\in H:j\neq l}\Sigma_{d\in DC}\Sigma_{m\in F_d}x_{jlmd} = 1 \qquad \forall l \in E \tag{9}$$

$$\Sigma_{j\in H:j\neq l}x_{jlmd} - \Sigma_{j\in H:j\neq l}x_{ljmd} = 0 \qquad \forall d \in DC, m \in F_d, l \in E \tag{10}$$

$$U_j - U_l + n * x_{jlmd} \leq n-1 \qquad \forall d \in DC, m \in F_d, j, l \in E : j \neq l \tag{11}$$

$$1 \leq U_j \leq n \qquad \forall j \in E \tag{12}$$

$$x_{jlmd} = 0 \qquad \forall d \in DC, j \in H : j \neq n+d \& j > n, l \in E, m \in F_d \tag{13}$$

$$x_{ljmd} = 0 \qquad \forall d \in DC, j \in H : j \neq n+d \& j > n, l \in E, m \in F_d \tag{14}$$

$$\Sigma_{l\in E}x_{n+d,lmd} \leq 1 \qquad \forall d \in DC, m \in F_d \tag{15}$$

$$\Sigma_{j\in H}\Sigma_{l\in E:j\neq l}\Sigma_{m\in F_d}demand_{lg}x_{jlmd} \leq np_d \qquad \forall g \in G, d \in DC \tag{16}$$

$$\Sigma_{l\in E}\Sigma_{m\in F_d}x_{n+d,lmd} \geq 1 \qquad \forall d \in DC \tag{17}$$

$$lb_{md} = \Sigma_{j\in H}\Sigma_{l\in E:j\neq l}\Sigma_{g\in G}demand_{lg}x_{jlmd} \qquad \forall d \in DC, m \in F_d \tag{18}$$

$$ld_j \geq lb_{md} - \Sigma_{g\in G}demand_{jg} + prod_j - M(1 - x_{n+d,jmd}) \qquad \forall j \in E, d \in DC, m \in F_d \tag{19}$$

$$ld_l \geq ld_j - \Sigma_{g\in G}demand_{lg} + prod_l - M(1 - \Sigma_{d\in DC}\Sigma_{m\in F_d}x_{jlmd}) \qquad \forall j, l \in E : j \neq l \tag{20}$$

$$lb_{md} \leq capacity \qquad \forall d \in DC, m \in F_d \tag{21}$$

$$ld_l \leq capacity \qquad \forall l \in E \tag{22}$$

$$np_d \leq scap_d \qquad \forall d \in DC \tag{23}$$

The objective (1) minimizes the total station cost and total travelling cost. Constraint (2) guarantees that each task in each DC is assigned to a station. Constraint (3) specifies that the total processing time at each station does not exceed the cycle time of that DC. Constraints (4) and (5) regulate the AND and OR precedence relations between the tasks, respectively. Constraint (6) guarantees the sequential opening of stations. Constraint (7) computes the cycle time for each DC. Constraint (8) states that the cycle time of each DC should be greater than or equal to the processing time of any task. Constraint (9) guarantees that each customer/supplier node is visited once. Constraint (10) provides the flow balance between nodes. Constraints (11) and (12) prevent sub-tours. Constraints (13)-(14) prohibit vehicles from visiting another disassembly center apart from their own center. Constraint (15) ensures that each vehicle is employed at most once. Constraint (16) calculates the number of products that are demanded from each DC. Constraint (17) states that each DC must serve at least one customer/supplier node. Constraint (18) computes the initial load of each vehicle when leaving its own DC. Constraints (19) and (20) compute the load of a vehicle after having serviced each customer/supplier node regarding the pickup and delivery operations in that node. Constraints (21) and (22) state that load of a vehicle cannot exceed the capacity. Constraint (23) states that the number of products that are demanded from each DC cannot exceed the total supply of that DC from the previous planning period.

The MINLP model provides an advantage in terms of ease of modeling. The model is established using fewer decision variables and constraints compared to the MILP model. Such a MINLP model provides a more understandable expression of the structure of the problem and ease of modeling. However, as shown in the study by Çil et. al. (2023) for the integrated distributed disassembly line balancing and vehicle routing problem, the MINLP model can provide a very poor performance in terms of both solution time and solution quality. One of the disadvantages of the MINLP model is that it can have a worse performance compared to the MILP and CP models. For this reason, even though the MINLP model is presented, it is not used in solution generation within the scope of this study.

3.2. Mixed-Integer Linear Programming Model

In this section, a MILP model is presented for the problem by linearizing the nonlinear constraints of MINLP model. The constraint linearization method proposed by Feng & Che (2023) is used to linearize the nonlinear constraints. The MILP model uses the same decision variables as the MINLP model. However, $cycle_d$ decision variable is not used in the MILP model and an additional decision variable is defined for the MILP model, which is described below.

rv_{ibd}= auxiliary decision variable for task i, station b, and DC d

The objective function and the constraints of the MILP model are provided below.

Objective Function (1) from MINLP
Constraint (2) from MINLP

$$\Sigma_{i \in A} ptime_i rv_{ibd} \leq wtime \quad \forall b \in B, d \in DC \tag{24}$$

$$rv_{ibd} \leq Np^U v_{ibd} \quad \forall i \in A, b \in B, d \in DC \tag{25}$$

$$rv_{ibd} \leq np_d + Np^L (v_{ibd} - 1) \quad \forall i \in A, b \in B, d \in DC \tag{26}$$

$$rv_{ibd} \geq np_d + Np^U (v_{ibd} - 1) \quad \forall i \in A, b \in B, d \in DC \tag{27}$$

$$\Sigma_{i \in A} v_{ibd} \leq |A| * y_{bd} \quad \forall b \in B, d \in DC \tag{28}$$

Constraints (4)-(6) from MINLP

$$wtime \geq ptime_i np_d \quad \forall d \in DC, i \in A \tag{29}$$

Constraints (9)-(23) from MINLP

The objective function of the MILP model remains the same as the objective function of the MINLP model. Since constraint (3) in the MINLP model is nonlinear, constraints (24)-(28) are added instead of this constraint. Constraints (24) – (27) states that total operation time of each station cannot exceed the cycle time of that DC, where Np^U and Np^L parameters are upper and lower bound values for the variable np_d, respectively. Constraint (28) indicates that if a task is assigned to a station, the station is open. To complete the transformation of the model into a linear model, constraint (29) is added instead of constraints (7) and (8) in the MINLP model. Constraint (29) states that the cycle time of each DC should be greater than or equal to the processing time of any task. The remaining constraints are the same as constraints (2), (4)-(6) and (9)-(23) in the MINLP model.

The MILP model has a more complex structure because it uses more decision variables and constraints than the MINLP model. However, making the constraints linear can improve the solution quality and performance of the model significantly, as shown in the study of Çil et al. (2023). For this reason, it is aimed to benefit from the solution quality advantage of a linear model by establishing the MILP model.

3.3. Constraint Programming Model

In this section, a CP model is presented for the problem. The decision variables and functions used in this model are described in Table 3. Subsequently, the objective function and constraints of the CP model are explained.

Table 3. Decision variables and functions of the CP Model

Decision Variables for CP
$\boldsymbol{np_d}$ = number of products (maximum number of components) that are demanded from DC *d*.
$\boldsymbol{nd_j}$ = interval variable for node *j*.
$\boldsymbol{y_{j,m}}$ = an optional interval variable exists if node *j* is served by vehicle *m*.
$\boldsymbol{Seq_m}$ = a sequence variable defined over interval variables $y_{j,m}$ and represents the route of each vehicle with $\boldsymbol{travelDistance}$ between nodes, respecting $\boldsymbol{travel_{jl}}$.
$\boldsymbol{st_{id}}$ = indicates the station number of task *i* in DC *d*.
Expressions of CP
$\boldsymbol{sta_{id} = min(k \in OR_i)st_{kd}}$ calculates the minimum of assigned station number of tasks *k*'s which are OR predecessors of task *i*.
$\boldsymbol{stationCost} = \sum_{d \in DC} \max_{i \in A} \boldsymbol{st_{id}} * \boldsymbol{scost}$ calculates the total station cost of DCs.
$\boldsymbol{travelCost} = \sum_{j \in H', d \in DC, m \in F_d} \boldsymbol{cost_{j,typeOfNext(Seq_m, y_{jm}, j, j)}}$ calculates the total travel cost of all vehicles.
Cumulative Function of CP
$\boldsymbol{cf_m} = \sum_{j \in E} \boldsymbol{stepAtStart(y_{jm}, prod_j)} - \sum_{j \in E, g \in G} \boldsymbol{stepAtStart(y_{jm}, demand_{jg})}$ cumulative function calculates the load of each vehicle *m* after visiting each node by considering loaded supplies of next period and the unloaded demands of the current period.

$$MinimizeZ = travelCost + stationCost \tag{30}$$

$$st_{kd} \leq st_{id} \qquad \forall i \in A, k \in AND_i, d \in DC \tag{31}$$

$$st_{id} \geq sta_{id} \qquad \forall i \in A, d \in DC \tag{32}$$

$$\Sigma_{i \in A}(st_{id} = b) ptime_i np_d \leq wtime, \qquad \forall b \in B, d \in DC \tag{33}$$

$$ptime_i np_d \leq wtime, \qquad \forall i \in A, d \in DC \tag{34}$$

$$alternative(nd_j, all(d \in DC, m \in F_d) y_{jm}) \qquad \forall j \in E \tag{35}$$

$$\Sigma_{j \in E, m \in F_d} presenceOf(y_{jm}) \geq 1, \qquad d \in DC \tag{36}$$

$$presenceOf(y_{n+d,m}) = 1, \qquad \forall d \in DC, m \in F_d \tag{37}$$

$$presenceOf(y_{n+s+d,m}) = 1, \qquad \forall d \in DC, m \in F_d \tag{38}$$

$$presenceOf(y_{n+d,m}) = 0, \qquad \forall d \in DC, m \notin F_d \tag{39}$$

$$presenceOf(y_{n+s+d,m}) = 0, \qquad \forall d \in DC, m \notin F_d \tag{40}$$

$$\Sigma_{j \in H', g \in G} stepAtStart(y_{jm}, demand_{jg}) + cf_m \leq capacity, \qquad \forall d \in DC, m \in F_d \tag{41}$$

$$\Sigma_{j \in H', g \in G} stepAtStart(y_{jm}, demand_{jg}) \leq capacity, \qquad \forall d \in DC, m \in F_d \tag{42}$$

$$first(Seq_m, y_{n+d,m}), \qquad \forall d \in DC, m \in F_d \tag{43}$$

$$last(Seq_m, y_{n+s+d,m}), \quad \forall d \in DC, m \in F_d \tag{44}$$

$$noOverlap(Seq_m, travelDistance, 1), \quad \forall d \in DC, m \in F_d \tag{45}$$

$$np_d \geq \Sigma_{j \in H', m \in F_d} stepAtStart(y_{jm}, demand_{jg}), \quad \forall g \in G, d \in DC \tag{46}$$

$$np_d \leq scap_d, \quad \forall d \in DC \tag{47}$$

The objective (30) minimizes the total station and total travel costs. Constraint (31) indicates that if task k is an AND predecessor of task i, task k should be assigned earlier or at the same station as task i. In constraint (32), the decision variable sta represents the minimum station number that OR predecessors of task i have been assigned, so the station of task i must be at least this minimum station or one of the subsequent stations. Constraint (33) calculates the cycle time of each DC, considering the total working time and the maximum number of components that are required from that DC. This constraint also states that total operation time of each station cannot exceed the cycle time of that DC. Constraint (34) specifies that the cycle time must be greater than or equal to the longest processing time of the tasks. Constraint (35) guarantees that each factory is visited by exactly one vehicle. Also, constraint (36) provides that at least one factory should be served from a DC. Constraint sets (37-40) state that all vehicles depart and return to their corresponding disassembly centers, and vehicles cannot depart or return to the other DCs to which they do not belong. Constraint (41) calculates the load of each vehicle during their tours using the cumulative function. In this constraint, a vehicle starts its tour by loading all the demands of the factories included in its tour, and it delivers these demands when visiting those factories, hence these demands are unloaded from the vehicle. Additionally, during their tours, vehicles collect product supplies for the next period, so the quantities of these product supplies are loaded onto the vehicle from the visited factories. Thus, Constraint (41) ensures that the vehicle capacity is not exceeded during the tour, and Constraint (42) provides the same constraints at the beginning of the tour, considering all the demands to be distributed. Constraints (43-44) specify that their respective DCs should be the first and last stops of the vehicles. Constraint (45) prevents overlapping tours of vehicles and provides the sequence of the visits. Constraint (46) calculates the number of demanded products from each DC based on the demands of the visited factories from that disassembly center. Similarly, constraint (47) states that the number of demanded products from each DC cannot exceed the total supply of that DC from the previous planning period.

The CP model differs from the MINLP and MILP models in terms of the decision variables used, modeling technique and solution search mechanism. The CP model is preferred for the integrated problem in this study, because of its good performance in scheduling and vehicle routing problems and it can narrow the solution space and the domain of decision variables by using constraints. Within the scope of this study, since the integrated problem is a complex problem with many constraints, it is aimed to benefit from using the constraints of the CP model effectively.

4. Computational Results

In the computational experiment, small-sized instances are generated to evaluate the performance of the MILP and CP models, following a similar instance generation procedure in Çil et al. (2023). The DLB instances presented in the study of Çil (2021) contain both AND precedence (P) and AND/OR precedence (POR) relationships for tasks. Among these instances, P9, P11 and POR10 instances are used for the DLB part of the studied problem. P9 and P11 instances consist of 9 and 11 tasks, respectively, and there is only AND precedence relationship between tasks in these instances. In POR10 instance, there are 10 tasks and there is an AND/OR precedence relationship between the tasks. For the VRP part of the studied problem, instances for the multi-depot VRP presented by Cordeau et al. (1997) have been modified according to the studied problem. The first 6, 8 and 10 customer coordinates are selected from the clusters to represent the factories in the studied problem. In addition, two and three depots are randomly selected from the clusters to represent the disassembly centers. The vehicle capacities have been increased by multiplying the number of components. The demand value for each customer is assumed as the demand for the first component, and the demands for the rest of the components are randomly generated between the minimum and maximum of the demand values in the instances. The product supply quantities of the factories to be collected are randomly generated between [1, 200]. The supplied products from the previous planning period for the disassembly centers are randomly generated between [200, 500]. It is assumed that there are four and six components in the instances. It is assumed that there is enough number of vehicles. Consequently, a total of 36 instances are generated considering the instances of P9, P11, POR10, the number of factories as 6, 8, 10, number of disassembly centers as 2, 3, number of components as 4, 6, where there is one instance for each parameter combination.

While the MILP model is solved on IBM ILOG CPLEX 12.10 platform, CP Optimizer of IBM ILOG CPLEX 12.10 is used to solve the CP model. All results of the models are obtained on a laptop with Intel Core i7 with 8GB of RAM. The results are obtained by solving the 36 instances by MILP and CP models. Both models are solved under a time limit of 1800 seconds for each instance. Table 4 presents the results of the MILP and CP models for these small-sized instances. The first column of Table

4 includes the instance number, and the next column includes the number of tasks. The next two columns represent the number of DC and number of factories for each instance. The relative percentage deviation (RPD%) values and solution time (in seconds) values of the MILP and CP models are also included in this table. The RPD computes the percentage deviation of each result found by the models from the best result for each instance, which is calculated as in Eq. (48). In the RPD calculation, result indicates the result of each model, while best result indicates the best result found among two models.

$$RPD\% = \frac{(result - best\,result)}{best\,result} * 100 \tag{48}$$

In Table 4, the number of tasks refers to the number of disassembly tasks in disassembly centers, and as this number increases, more workloads occur at the workstations. DCs are places where the products are disassembled and turned into components. There is a disassembly line in each DC. Any facility can have more than a single DC, so different numbers of DCs are considered to show their effect on the solution quality. Factories are also places where the products to be disassembled are supplied and the disassembled components are delivered. Production is carried out by reusing the components in these factories. In this study, factories represent both suppliers (where products are collected) and customers (where components are distributed) of DCs. Therefore, different number of factories are considered to evaluate the sensitivity of the solution quality as the number of suppliers and customers changes.

Table 4. Computational results for the models

Instance	*Number of Tasks*	*Number of DC*	*Number of Factories*	*MILP*		*CP*	
				RPD (%)	*Computational Time(s)*	*RPD (%)*	*Computational Time(s)*
1	9	2	6	0.00	28	1.39	1800
2	9	2	6	0.00	61	0.00	1800
3	9	3	6	0.00	102	0.00	1800
4	9	3	6	0.00	419	0.00	1800
5	9	2	8	0.00	1800	0.00	1800
6	9	2	8	0.00	1800	0.00	1800
7	9	3	8	0.00	1800	0.00	1800
8	9	3	8	0.00	1800	0.00	1800
9	9	2	10	0.00	1800	0.41	1800
10	9	2	10	0.00	1800	0.00	1800
11	9	3	10	0.00	1800	3.04	1800
12	9	3	10	1.41	1800	0.00	1800
13	11	2	6	0.00	11	0.09	1800
14	11	2	6	0.00	5	0.00	1800
15	11	3	6	0.00	112	0.00	1800
16	11	3	6	0.00	196	0.00	1800
17	11	2	8	0.00	131	1.02	1800
18	11	2	8	0.00	1800	0.00	1800
19	11	3	8	0.00	1800	0.23	1800
20	11	3	8	0.00	1800	0.00	1800
21	11	2	10	0.00	1800	0.32	1800
22	11	2	10	0.76	1800	0.00	1800
23	11	3	10	0.11	1800	0.00	1800
24	11	3	10	0.00	1800	0.00	1800
25	10	2	6	0.00	113	0.00	1800
26	10	2	6	0.00	334	0.00	1800
27	10	3	6	0.00	1253	0.00	1800
28	10	3	6	0.00	633	0.00	1800
29	10	2	8	0.12	1800	0.00	1800
30	10	2	8	0.00	1800	0.00	1800
31	10	3	8	0.00	1800	0.87	1800
32	10	3	8	1.69	1800	0.00	1800
33	10	2	10	0.00	1800	0.06	1800
34	10	2	10	0.00	1800	0.00	1800
35	10	3	10	0.17	1800	0.00	1800
36	10	3	10	0.30	1800	0.00	1800
	Average			**0.13**	**1244.39**	**0.21**	**1800.00**

When Table 4 is analyzed, there is no significant difference between MILP and CP models in solving small-sized problems. The MILP model has an average RPD value of 0.13%, which is slightly better than the average RPD value of the CP model (0.21%). While the MILP model finds better results than the CP model for 9 out of 36 instances, the CP model finds better results for 7 out of 36 instances. For the remaining 20 instances, the CP model and the MILP model find the same result. On the other hand, the MILP model obtains optimal solutions for 13 of these 36 instances. The MILP model can find the optimal solutions for only some of the instances which have 6 factories. As the number of factories increases, the MILP model cannot find optimal results within the limited time given. These cases show the effect of the number of factories on the complexity of the problem and the solution time. On the other hand, the CP model cannot find any proven optimal solution. Although the CP model finds the same result for 10 of these 13 optimal solutions found by the MILP model, it cannot prove the optimality of the solutions. If we analyze the results in terms of solution time, the CP model uses the entire time limit for all instances, while the MILP model finds the optimal solutions for 13 instances in less computational time. Although the CP model can find the optimal results found by MILP in a short time, since it cannot prove optimality, it uses the entire time limit, and the CPU time seems to be high. The reason why the CP model cannot prove optimality is that it has difficulty finding a lower bound. The CP model narrows the domain of decision variables by using constraints. At each stage, it propagates the constraints in the solution space, cuts the domains of the decision variables and infers the values of the variables to find a solution. Since the standard branch and bound structure applied in MILP models is not available in the CP model, it can be difficult to find a lower bound. Therefore, even if the CP finds good solutions, it cannot prove optimality. When the computational results were analyzed in detail, in general, MILP or CP models do not have significant superiority over each other for solving the problem. It can be said that both models are able to provide good quality solutions for small-sized instances.

5. Conclusions

This study addressed the integrated disassembly line balancing and pickup-delivery vehicle routing problem with distributed disassembly centers. Products collected from the factories are disassembled in the disassembly centers, and then the disassembled components are distributed to the factories. The studied integrated problem optimizes both the line balancing of the disassembly lines in disassembly centers and the vehicle routes for the collection of products and distribution of components, simultaneously. The studied problem differs from the related studies on integrated DLB and vehicle routing problem in the literature, as there are multiple disassembly centers with limited product supplies and distribution and collection operations are planned together within the scope of this study. In terms of practical implications of the study, the integration of pickup-delivery vehicle routing and disassembly line balancing problems for distributed disassembly centers can help to minimize total costs, achieve a more effective resource management, and enhance the sustainability and adaptability of the system to dynamic changing demands. In this study, mixed-integer nonlinear programming, mixed-integer linear programming and constraint programming models were presented for this complex integrated problem. The presented mathematical models can be used by the managers for solving small-sized problems in the planning of disassembly line balancing and collection-distribution processes for multiple disassembly centers. Also, the performance of the MILP and CP models was tested on the small-sized instances. The computational results show that both MILP and CP models can provide effective solutions.

In terms of limitations of the study, presented mathematical models can provide good-quality solutions for only small-sized instances due to the complexity of the problem, as shown in Section 4. It is expected that, as the problem size increases the solution time of the mathematical models will also increase. Hence, in future studies, heuristic or metaheuristic algorithms can be developed for the problem to solve large-sized instances. Since effective heuristic/metaheuristic algorithms can provide good-quality solutions in short computational times, it can be practical to develop a heuristic/metaheuristic algorithm for solving the large-sized instances of the problem. Also, in a planning period, factories can be available for only certain time windows for the collection of products or delivery of components. In such case, the visiting time for these factories should comply with their time windows. In future studies, time window constraints for the factories can also be considered for the problem.

Peer Review: Externally peer-reviewed.

Conflict of Interest: Author declared no conflict of interest.

Financial Disclosure: This study was supported by Scientific Research Fund of the Izmir Demokrasi University. Project Number: HIZDEP-MHF/2201.

Author Contributions: Conception/Design of Study- Z.A.Ç., D.K., H.Ö.; Data Acquisition- Z.A.Ç., D.K., H.Ö.; Data Analysis/ Interpretation- Z.A.Ç., D.K., H.Ö.; Drafting Manuscript- Z.A.Ç., D.K., H.Ö.; Critical Revision of Manuscript- Z.A.Ç., D.K., H.Ö.; Final Approval and Accountability- Z.A.Ç., D.K., H.Ö.

ORCID IDs of the authors

Zeynel Abidin Çil 0000-0002-7270-9321
Damla Kızılay 0000-0002-6561-8819
Hande Öztop 0000-0002-6503-7299

REFERENCES

Braekers K., Ramaekers K., and Van Nieuwenhuyse I. (2016), "The vehicle routing problem: State of the art classification and review", Computers & Industrial Engineering, vol. 99, pp. 300-313.

Cordeau J.-F., Gendreau M., and Laporte G. (1997), "A tabu search heuristic for periodic and multi-depot vehicle routing problems", Networks, vol. 30, no. 2, pp. 105–119.

Çil Z. A. (2021), "An exact solution method for multi-manned disassembly line design with AND/OR precedence relations", Applied Mathematical Modelling, vol. 99, pp. 785–803.

Çil, Z. A., Öztop, H., Kenger, Z. D., and Kizilay, D. (2023), "Integrating distributed disassembly line balancing and vehicle routing problem in supply chain: Integer programming, constraint programming, and heuristic algorithms", International Journal of Production Economics, 265, 109014.

Diri Kenger Z., Koç Ç., and Özceylan E. (2020a), "Integrated disassembly line balancing and routing problem", International Journal of Production Research, vol. 58, no. 23, pp. 7250–7268.

Diri Kenger Z., Koç Ç., and Özceylan E. (2020b), "Smart Factories: Integrated Disassembly Line Balancing and Routing Problem with 3D Printers", in Logistics 4.0, CRC Press.

Diri Kenger Z., Koç Ç., and Özceylan E. (2021a), "Integrated Collection-Disassembly-Distribution Problem", in Sustainable Production and Logistics, CRC Press.

Diri Kenger Z., Koç Ç., and Özceylan E. (2021b), "Integrated disassembly line balancing and routing problem with mobile additive manufacturing", International Journal of Production Economics, vol. 235, p. 108088.

Feng J. and Che A. (2023), "A note on integrated disassembly line balancing and routing problem", International Journal of Production Research, vol. 61, no. 9, pp. 3145–3151.

Habibi M. K. K., Battaïa O., Cung V.-D., and Dolgui A. (2014), "Integrated Procurement–Disassembly Problem", in Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World, pp. 482–490.

Özceylan E., Kalayci C. B., Güngör A., and Gupta S. M. (2019), "Disassembly line balancing problem: a review of the state of the art and future directions", International Journal of Production Research, vol. 57(15-16), pp. 4805-4827.

How cite this article

Çil, Z.A., Kızılay, D., & Öztop, H. (2023). Mathematical models for disassembly line balancing and pickup - delivery vehicle routing problem. *Journal of Transportation and Logistics, 8*(2), 180-190. https://doi.org/10.26650/JTL.2023.1338878