

A Hybrid Grouping Genetic Algorithm with Heterogeneous Boxes and Pallets for Pallet and Vehicle Loading Optimization: An Application in TP Electric

**Müfide Defne Akbay, Duru Atılgan, Eren Efe Bulut, Ayça Kılıçarslan, Gökçe Başak Salt
and Bartu Şıracı**

Industrial Engineering Department
Yaşar University
Bornova, 35100, Türkiye

Ayhan Özgür Toy

Professor
Department of Industrial Engineering
Yaşar University
Bornova, 35100, Türkiye

Abstract

This study presents an integrated optimization framework for the hierarchical pallet and vehicle loading problem arising in real export logistics operations. The problem involves assigning heterogeneous boxes to compatible pallet types under strict type-compatibility constraints, packing each pallet in three dimensions, arranging the loaded pallets within a single vehicle, and selecting the most economical vehicle from an available fleet. A Grouping Genetic Algorithm (GGA) combined with an Extreme Point (EP) decode heuristic is proposed to address all decision levels within a single framework. Each chromosome encodes a variable-length sequence of pallet groups rather than individual item assignments, preserving well-formed pallet structures across generations and naturally enforcing type-compatibility constraints. Four fitness-guided mutation operators targeting volumetric fill ratio imbalances restructure weak pallet groups during the search. The EP heuristic decodes each chromosome into a geometrically feasible loading plan at both the pallet and vehicle levels, with vehicle type selection resolved automatically at the end of each evaluation. The framework was implemented in C++ with OpenMP parallelism and evaluated on eight real shipment instances obtained from TP Electric Electrical Materials Industry and Trade Inc., a manufacturer based in İzmir, Türkiye. The proposed algorithm matched or improved the company's manual loading plan in seven out of eight cases, achieving an average pallet count reduction of 12.5% and an average lower bound gap of 22.8%, compared to 47.3% for the company baseline. These results confirm the practical applicability of the proposed approach for real-world heterogeneous pallet and vehicle loading decisions.

Keywords

Grouping Genetic Algorithm, Three-Dimensional Bin Packing, Extreme Point Heuristic, Pallet Loading, Vehicle Loading Optimization

1. Introduction

Efficient logistics planning plays a crucial role in modern manufacturing and distribution systems, directly affecting delivery performance, operational costs, and resource utilization. Among these decisions, palletization and truck loading are particularly important, yet in many real-world settings they are still made manually, leading to inconsistent

results and inefficient use of space. As customer orders become more diverse, these limitations become more pronounced, making structured and data-driven approaches increasingly necessary.

The challenge becomes considerably more difficult when boxes are heterogeneous in size and type, pallets belong to multiple families with different floor dimensions and load height limits, and the fleet consists of vehicles of varying capacities. The loading decision is then a hierarchical two-level optimization problem: boxes must first be assigned to compatible pallet types and packed in three dimensions, and the resulting loaded pallets must be arranged within a selected vehicle. Each level interacts with the other, so a poor box-to-pallet assignment can render the pallet-to-vehicle placement infeasible regardless of how well either level is solved independently.

Exact optimization methods become computationally intractable for real-world problem sizes in this domain, as the underlying structure belongs to the class of NP-hard combinatorial problems (Martello et al. 2000). Industrial logistics operations therefore require fast and effective heuristic or metaheuristic approaches that can produce high-quality solutions within practical time limits. Genetic algorithms have been widely explored for packing and loading problems, but standard item-permutation representations perform poorly in bin packing contexts because crossover operations tend to disrupt well-formed groupings. The Grouping Genetic Algorithm (GGA), originally introduced by Falkenauer (1996), addresses this fundamental limitation by operating directly on groups rather than individual items, preserving high-quality pallet assignments across generations.

Despite the practical importance of integrated pallet and vehicle loading, the existing literature largely treats these two levels separately. Few works simultaneously optimize box-to-pallet assignment, three-dimensional geometric feasibility within pallets, and vehicle selection within a single framework. This gap motivates the present study, which proposes a Grouping Genetic Algorithm (GGA) combined with an Extreme Point heuristic to address both loading levels in an integrated manner. The originality of this work lies in the group-level solution representation that naturally encodes pallet assignments, the fitness-guided mutation operators targeting weak pallet groups, and the group-level local search applied in parallel to top-performing individuals.

1.1 Objectives

This study aims to develop a GGA-based optimization framework that simultaneously handles heterogeneous box-to-pallet assignment and pallet-to-vehicle loading with full three-dimensional geometric feasibility, implement an Extreme Point decode procedure for efficient placement at both loading levels, and demonstrate that fitness-guided mutation operators targeting fill ratio imbalances improve solution quality on real-world logistics instances.

2. Literature Review

The three-dimensional bin packing problem (3D-BPP) involves packing rectangular items into the minimum number of bins and is strongly NP-hard, making exact methods impractical for large instances (Martello, Pisinger, and Vigo, 2000). Within the established typology of Wäscher, Haußner, and Schumann (2007), this study falls under the category of three-dimensional multiple bin size bin packing. Prior research shows that hybrid approaches combining constructive heuristics with metaheuristic search outperform single-method solutions, particularly in mixed-case palletization settings (Lodi, Martello, and Vigo, 2002; Elhedhli, Gzara, and Yildiz, 2019).

The placement strategy plays a key role in solution quality. The Extreme Point (EP) heuristic generates candidate positions by projecting along orthogonal axes, enabling efficient and compact packings (Crainic, Perboli, and Tadei, 2008). EP-based heuristics such as EP-FFD and EP-BFD have demonstrated competitive performance, while comparisons with Maximal Empty Space (EMS) methods indicate similar solution quality but faster convergence for EP (Zhao et al., 2021). Moreover, EP has been successfully integrated into evolutionary frameworks, confirming its effectiveness as a flexible decoding mechanism (Zheng, Chien, and Gen, 2015).

For metaheuristic search, encoding choice is critical. Falkenauer and Delchambre (1992) demonstrated that standard genetic algorithms perform poorly on grouping problems because item-level crossover routinely destroys the bin groupings that define solution structure. Falkenauer (1996) introduced the Grouping Genetic Algorithm (GGA), in which chromosomes encode entire bins and crossover operates at the group level. The GGA fitness function, defined as the average squared bin fill ratio, rewards configurations with some nearly full bins over uniformly partially filled ones, a design principle that directly promotes bin count minimization. Rohlfshagen and Bullinaria (2007) confirmed through systematic benchmarking that group-preserving crossover is structurally necessary: standard crossover applied to grouping encodings converges to significantly worse solutions on hard instances. Gonçalves and Resende

(2013) proposed a Biased Random-Key Genetic Algorithm (BRKGA) tested on 858 instances, demonstrating that combining population-based search with structured placement heuristics produces competitive results on large instances, with hybrid approaches typically achieving 5–10% improvements in bin utilization over purely constructive methods (Lodi et al., 2002).

The simultaneous optimization of box-to-pallet assignment and vehicle selection has received limited attention. Bortfeldt and Wäscher (2013) identified the joint consideration of cargo assignment and container selection as underexplored, as sequential treatment produces substantially suboptimal outcomes (Côté, Guastaroba, and Speranza, 2017). Fuellerer et al. (2010) addressed vehicle routing with 3D loading constraints but did not model heterogeneous pallet types. Ruan et al. (2013) applied a genetic algorithm to combined pallet building and truck loading but assumed homogeneous pallet types, limiting applicability to the heterogeneous setting studied here. More recent work, including Yang et al. (2024) and Alonso et al. (2022), still treats vehicle type as a fixed parameter and separates pallet-building from vehicle assignment into sequential stages.

Across the reviewed literature, a clear structural limitation emerges. Existing approaches do not simultaneously address heterogeneous box-to-pallet assignment, geometrically feasible three-dimensional packing within pallets, and dynamic vehicle type selection within a unified optimization framework. Most studies either assume homogeneous pallet types, treat vehicle selection as a fixed input, or decompose the problem into sequential stages, thereby eliminating any feedback between palletization and vehicle loading decisions.

To overcome these limitations, this study proposes an integrated optimization framework based on a GGA combined with an Extreme Point decoding mechanism. The proposed approach preserves pallet-level structure through group-based encoding, explicitly models heterogeneous pallet types, and treats vehicle selection as a decision variable rather than a predefined parameter, addressing a critical gap in the literature.

3. Methods

3.1 Problem Definition

The problem addressed in this study involves three interrelated sets. Let $B = \{b_1, b_2, \dots, b_n\}$ denote the set of n boxes to be shipped. Let $P = (p_1, p_2, \dots, p_m)$ denote the set of available pallet instances, each belonging to one of three pallet types $T = (P80\text{-}BOX, P80\text{-}CARTON, P100\text{-}CARTON)$. Let $V = (v_1, v_2, \dots, v_k)$ denote the fleet of available vehicle types, ordered by increasing cargo volume.

Each box $b_i \in B$ is characterized by its external dimensions (l_i, w_i, h_i) and belongs to one of three packaging classes: Box, P80-Carton, or P100-Carton. The packaging class determines a strict type compatibility mapping $\tau: B \rightarrow T$, where Box items map exclusively to P80-BOX pallets, P80-Carton items to P80-CARTON pallets, and P100-Carton items to P100-CARTON pallets. No mixing of item types across pallet types is permitted. Each pallet type $t \in T$ defines a floor footprint (L_t, W_t) , a fixed base height H^{base}_t , and a maximum permissible load height H^{max}_t including the H^{base}_t . Each vehicle type $v_j \in V$ defines an internal cargo envelope (L_j, W_j, H_j) .

The loading problem is hierarchical. At the first level, every box must be assigned to exactly one pallet instance whose type is compatible with $\tau(b_i)$. Each assigned box must then be placed at a feasible three-dimensional position within the pallet boundaries: no two boxes may overlap, and every box must remain within the pallet footprint and below H^{max}_t . At the second level, the resulting loaded pallets must be arranged within a single selected vehicle. A pallet may be placed on the vehicle floor or stacked on top of another pallet, provided its footprint falls entirely within the supporting pallet's footprint. Every placed pallet must remain within the vehicle's cargo envelope (L_j, W_j, H_j) .

It is important to note that the objective of this study is not to place every box in the shipment order without exception. The input data reflects real demand records, and the total order volume may exceed the capacity of a single vehicle. In line with the company's operational policy, each planning instance is restricted to the use of only one vehicle, even in cases where the total demand cannot be accommodated by the largest available vehicle in the fleet. Therefore, boxes that cannot be loaded within the selected vehicle are deferred to the next shipment rather than forcing the model to allocate additional vehicles.

Furthermore, since the company's product portfolio mainly consists of lightweight plastic-based industrial electrical components, weight-related limitations do not constitute a binding constraint in pallet or vehicle loading operations.

As a result, box weights are disregarded in this problem formulation, and the optimization process focuses on geometric feasibility, pallet utilization, volumetric efficiency, vehicle selection, and type-compatibility constraints.

Accordingly, the loading objective is to minimize the number of pallets used, maximize the average volumetric fill rate ρ across all loaded pallets, and select the smallest feasible vehicle from the available fleet, subject to the geometric and type-compatibility constraints described above. The fitness function $f(C)$ that encodes these goals is formally defined in Section 3.5.

3.2 Proposed Algorithm

The proposed solution approach addresses the two-level pallet and vehicle loading problem through a Grouping Genetic Algorithm (GGA) combined with an Extreme Point (EP) decode heuristic. The algorithm maintains a population of candidate loading plans, each encoded as a group-based chromosome in which every gene represents a complete pallet assignment rather than an individual box. This encoding is essential for packing problems, as it preserves well-formed pallet groups across generations and prevents the structural disruption that standard permutation-based representations cause during crossover. At each generation, the population evolves through selection, crossover, mutation, and a group-level local search applied in parallel to the top-performing individuals.

The algorithm operates in three main stages. In the first stage, an initial population of *pop_size* chromosomes is constructed using three complementary strategies: a First Fit Decreasing (FFD) heuristic, a balanced volume-based distribution, and random initialization applied in proportions of 40%, 30%, and 30% respectively. This mix ensures both solution quality and diversity from the outset. In the second stage, the population evolves over *max_gen* generations. At each generation, offspring are produced through tournament selection and the Bin Packing Crossover with Replacement (BPRX) operator, followed by fitness-guided mutation operators that target structurally weak pallet groups. The entire offspring batch is evaluated in parallel using OpenMP threads. In the third stage, a group-level local search is applied in parallel to the top-*n* individuals of the sorted new population at each generation to refine solution quality before the population is updated (Figure 1).

Each candidate solution is evaluated through a decode procedure that translates the chromosome into a concrete loading plan. The decode procedure first applies a repair step to enforce feasibility constraints, then packs each pallet group independently using the Extreme Point heuristic, and finally arranges the loaded pallets within the selected vehicle. Vehicle selection is performed automatically at the end of decoding: the smallest vehicle from the available fleet whose cargo envelope accommodates the spatial footprint of all placed pallets is chosen. The fitness value assigned to each chromosome is computed from the decoded solution as described in Section 3.4.

Algorithm 1 Grouping Genetic Algorithm for Pallet and Vehicle Loading	
1:	Input: $B, P, T, V, \text{pop_size}, \text{max_gen}, \text{crossover_rate}, \text{mutation_rate}, \text{elite_count}, \text{ls_top_n}$
2:	Output: Best chromosome C^* , decoded solution S^*
3:	for $i = 1$ to pop_size do
4:	$\text{pop}[i] \leftarrow \text{RandomChromosome}$ $\triangleright$ FFD 40% / balanced 30% / random 30%
5:	end for
6:	$[\text{pop}, S_{\text{init}}] \leftarrow \text{Parallel Evaluate}(\text{pop})$ $\triangleright$ Repair + decode + fitness (OpenMP)
7:	$C^* \leftarrow \text{argmin } f(C) \text{ over pop}$
8:	$S^* \leftarrow \text{decoded solution of } C^*$
9:	for $\text{gen} = 1$ to max_gen do
10:	Sort pop by $f(C)$ ascending
11:	$\text{new_pop} \leftarrow \text{pop}[1 .. \text{elite_count}]$ $\triangleright$ elitism
12:	$\text{offspring} \leftarrow \emptyset$
13:	while $ \text{new_pop} + \text{offspring} < \text{pop_size}$ do
14:	$P_1, P_2 \leftarrow \text{Tournament Selection}(\text{pop})$
15:	if $\text{rand}() < \text{crossover_rate}$ then $\text{child} \leftarrow \text{bprx}(P_1, P_2)$
16:	else $\text{child} \leftarrow \text{copy of } P_1$
17:	if $\text{rand}() < \text{mutation_rate}$ then $\text{child} \leftarrow \text{Mutate}(\text{child})$
18:	$\text{offspring} \leftarrow \text{offspring} \cup \{\text{child}\}$
19:	end while
20:	$[\text{offspring}, S_{\text{off}}] \leftarrow \text{Parallel Evaluate}(\text{offspring})$ $\triangleright$ Repair + decode inside
21:	$\text{new_pop} \leftarrow \text{new_pop} \cup \text{offspring}$
22:	Sort new_pop by $f(C)$ ascending
23:	for $i = 1$ to ls_top_n do $\triangleright$ parallel local search
24:	$\text{new_pop}[i], S_{\text{li}} \leftarrow \text{Local Search}(\text{new_pop}[i])$
25:	if $f(\text{new_pop}[i]) < f(C^*)$ then
26:	$C^* \leftarrow \text{new_pop}[i]$ $S^* \leftarrow S_{\text{li}}$
27:	end if
28:	end for
29:	$\text{pop} \leftarrow \text{new_pop}$
30:	end for
31:	return C^*, S^*

Figure 1. Main loop of the proposed Grouping Genetic Algorithm.

The key components referenced in Algorithm 1 are detailed in the following subsections. The chromosome representation and the feasibility repair procedure are described in Section 3.3. The Extreme Point decode procedure is presented in Section 3.4, the fitness function in Section 3.5, and the crossover, mutation, and local search operators in Section 3.6.

3.3 Solution Representation

In standard permutation-based genetic algorithms for packing problems, each gene encodes the index of a single item, and the chromosome defines a sequence over all items. While straightforward to implement, this representation does not align naturally with the grouping structure of the problem: a crossover operation that exchanges segments of two item sequences will typically scatter items across incompatible pallets, producing infeasible offspring that require extensive repair. The proposed algorithm instead adopts a group-based chromosome representation following the Grouping Genetic Algorithm (GGA) framework introduced by Falkenauer (1996). In this representation, each gene corresponds to an entire pallet group rather than an individual item. This encoding preserves well-formed pallet assignments across generations, since genetic operators act at the group level and the item-to-pallet assignment structure is never inadvertently disrupted.

A chromosome C consists of a variable-length sequence of K pallet groups ($g_1, g_2, \dots, g_k$) together with a vehicle index $vehicle_idx$ that identifies the selected vehicle type. Each pallet group g_k is defined by three fields. The field $pallet_instance_idx$ identifies which physical pallet instance from P is used for this group.

The field $orientation \in \{0, 1\}$ specifies whether the pallet is placed in its default footprint ($L \times W$) or rotated 90° , yielding a footprint of ($W \times L$) within the vehicle. The field $item_indices$ is the list of box indices assigned to this group, where every box $b_i \in B$ appears in exactly one group across the entire chromosome. The number of groups K is not fixed: genetic operators may add or remove groups during the search, allowing the chromosome to explore solutions with varying numbers of pallets. Figure 2 illustrates the general chromosome structure, where the final group g_k emphasizes the variable-length nature of the representation.

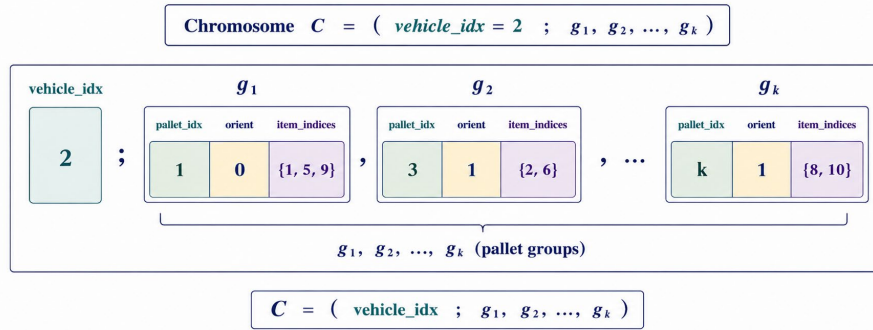


Figure 2. Illustration of Group-Based Chromosome Structure with Pallet-Level Assignments

A chromosome C is an abstract assignment structure and does not by itself constitute a loading plan. To evaluate a candidate solution, the chromosome must be translated into a concrete arrangement of boxes on pallets and pallets within a vehicle. This translation is performed by the decode procedure described in Section 3.4, which applies the Extreme Point heuristic at both the pallet and vehicle levels and automatically selects the smallest feasible vehicle from V . The fitness value of C is then computed from the resulting decoded solution as defined in Section 3.5.

3.4 Decode Procedure

The decode procedure translates a chromosome C into a concrete loading plan by operating in two sequential levels. Before any packing begins, a repair step is applied to enforce the hard constraints described in Section 3.1, ensuring that every box appears in exactly one group and that no pallet instance is used more than once. The first level then processes each pallet group independently, packing the assigned boxes into the corresponding pallet using the Extreme Point heuristic. The second level takes the resulting loaded pallets and arranges them within the cargo envelope of a selected vehicle. The fitness value of the chromosome is computed from the decoded solution upon completion of both levels, as defined in Section 3.5.

At the first level, boxes assigned to a group are packed into their designated pallet instance using the Extreme Point (EP) heuristic. Following Crainic, Perboli, and Tadei (2008), an extreme point is a candidate placement position derived from the surfaces of already-placed boxes: specifically, for each placed box, three new candidate positions are generated by projecting along the x-, y-, and z-axes to the nearest obstacle or container wall. This process yields a compact set of positions that tend to produce tight, space-efficient packings without requiring an exhaustive search over the continuous placement space. Before packing, boxes are sorted in descending order of volume so that larger items claim space first, reducing fragmentation. For each box, all six axis-aligned orientations are evaluated in sequence, and the box is placed at the first extreme point that admits a feasible, non-overlapping placement within the pallet footprint ($L_i \times W_i$) and below the maximum load height $H^{\max}$. The EP list is updated incrementally after each placement: positions that fall inside the newly placed box are removed, and up to three new candidates are appended. Boxes that cannot be accommodated in any orientation are recorded as unplaced and penalized in the fitness function. At the second level, the loaded pallets produced by the first level are arranged within the vehicle's cargo envelope ($L_j \times W_j \times H_j$). Pallets are sorted in descending order of floor area ($dx \times dy$), with ties broken by pallet height, so that large-footprint pallets are placed first. For each pallet, a set of candidate positions is assembled by combining floor-level candidates points at $Z = 0$ generated incrementally as previously placed pallets create new corners with stacked candidates at $Z > 0$ derived from the top surfaces of already-placed pallets. A stacked placement is accepted only if the incoming pallet's footprint falls entirely within the footprint of its supporting pallet, ensuring structural stability. The first feasible, non-overlapping candidate within the vehicle envelope is selected. Pallets that cannot be placed are recorded as unplaced. Upon completing the vehicle packing, the smallest vehicle type from V whose cargo envelope fully contains all placed pallets is selected as the final vehicle for the solution.

3.5 Fitness Function

Each candidate solution is evaluated through a composite fitness function that captures both packing efficiency and feasibility. The objective is to minimize the total cost associated with a loading configuration while maximizing space utilization.

The objective function is defined as:

$$F = w_1 N_{\text{pallet}} - w_2 U + w_3 N_{\text{unplaced}} + w_4 R_{\text{vehicle}}$$

where N_{pallet} denotes the total number of pallets allocated (both placed and unplaced), U represents the mean squared pallet fill ratio, N_{unplaced} is the number of items that remain unassigned, and $R_{\text{vehicle}} = (V_{\text{vehicle}} - V_{\text{items}})/V_{\text{vehicle}} \in [0,1]$ is the fractional wasted volume of the selected vehicle. The coefficients are weighting parameters that regulate the trade-off between compactness and feasibility. They were set as $w_3 > w_4 > w_1 > w_2$, prioritizing feasibility by heavily penalizing unplaced items, penalizing vehicle underutilization as a secondary objective, minimizing total pallet count as the tertiary efficiency objective, and treating pallet fill maximization as a marginal improvement criterion. For a given pallet $i \in \{1, \dots, k\}$, the utilization ratio is defined as:

$$u_i = \frac{V_i^{\text{assigned}}}{V_i^{\text{capacity}}}$$

where V_i^{assigned} is the total volume of items assigned to pallet i , and V_i^{capacity} denotes its usable volume capacity.

To promote balanced and well-filled pallet configurations, the average utilization is calculated using a quadratic aggregation:

$$U = \frac{1}{k} \sum_{i=1}^k u_i^2$$

This formulation penalizes uneven distributions and rewards solutions with consistently high utilization across all pallets. To account for constraint violations such as exceeding height limits or infeasible placements, a penalty term λ is incorporated into the final fitness function:

$$f(C) = F + \lambda$$

3.6 Genetic Operators

3.6.1 Crossover

The crossover operator follows the Bin Packing Crossover with Replacement (BPRX) scheme introduced by Falkenauer (1996), adapted here to operate on the group-based chromosome representation. BPRX is designed to preserve well-formed pallet groups across generations by transferring entire groups from one parent rather than individual item assignments. This prevents the structural disruption that segment-based crossovers cause when applied to packing problems, where the value of a grouping lies in the collective arrangement of items within a pallet rather than in the presence of any single item.

Given two parent chromosomes p_1 and p_2 , the operator constructs an offspring as follows. A contiguous segment of groups $[a, b]$ is selected at random from p_1 . These groups are injected into the offspring, which is otherwise initialised as a copy of p_2 . All boxes that appear in the injected groups are then removed from the remaining groups inherited from p_2 , eliminating any duplicate assignments. Because this removal may leave some groups empty or cause certain boxes to become unassigned, a repair procedure is applied before evaluation to restore feasibility. The vehicle index of the offspring is inherited from either parent with equal probability.

3.6.2 Mutation

The mutation operator acts at the group level and is guided by the fill scores of the current groups, targeting structurally weak groups rather than modifying the chromosome at random. The fill score of a group is defined as the ratio of total item volume to pallet usable volume; groups with low fill scores represent inefficient pallet utilization and are the primary candidates for restructuring. Four mutation strategies are implemented and applied stochastically.

The first strategy, *weak group deletion*, identifies the one or two groups with the lowest fill scores and removes them entirely from the chromosome. The items displaced by this operation are redistributed to remaining groups or used to open new groups via the repair procedure. This operator directly reduces the number of pallets when under-utilized groups can be consolidated elsewhere.

The second strategy, *imbalance rebalancing*, targets the pair of groups with the greatest fill-score difference within the same pallet type. The items of both groups are merged and redistributed proportionally: items are sorted by volume in descending order and assigned to the two groups such that the denser group receives the larger-volume items up to a capacity threshold, while the remainder populates the lighter group. This reduces the variance in fill rates across groups of the same type.

The third strategy, *consolidation relocation*, selects the weakest group as a source and the strongest compatible group as a destination. A single item is transferred from the source to the destination, provided the destination has sufficient remaining capacity. If the source group becomes empty after the transfer, it is removed, reducing the total pallet count by one.

The fourth strategy, *overflow rebuild*, identifies the group whose total item volume most exceeds the pallet capacity and dissolves it. The displaced items are re-inserted by the repair procedure in descending order of volume, ensuring that large items claim space first in newly opened or existing groups. This operator is triggered only when the overflow ratio exceeds a threshold, preventing unnecessary restructuring of groups that are only marginally over capacity.

3.6.3 Local Search

A group-level local search is applied at each generation to the top- n individuals of the sorted population, running in parallel across threads. The local search performs a fixed number of iterations, in each of which a neighbourhood move is sampled and evaluated. If the move improves fitness it is accepted immediately; a slightly worse solution is accepted with a small probability to allow limited exploration, while solutions that are significantly worse are always rejected.

Five neighbourhood move types are defined. The *item relocation* move selects a random box from one group and transfers it to a compatible group of the same pallet type, reducing fragmentation between groups. The *group split* move divides the most populated group into two halves, opening a new pallet group using a previously unused pallet instance; this is applicable only when an unused compatible instance is available. The *group merge* move combines two randomly selected groups of the same pallet type into one, reducing the total pallet count at the cost of potentially increasing the overflow of the merged group. The *group rebuild* move dissolves the smallest group and redistributes its items via the repair procedure, targeting groups that are likely causing unnecessary pallet usage. Finally, the *item swap* move exchanges a randomly selected box between two groups of the same pallet type, allowing fine-grained rebalancing without altering group count or structure.

3.7 Parameter Configuration

Algorithm parameters were determined through a preliminary tuning study conducted on a representative shipment instance prior to the main computational experiments. The tuning procedure evaluated combinations of two key parameters: the crossover rate $p_c \in \{0.70, 0.75, 0.80, 0.85, 0.90\}$ and the mutation rate $p_m \in \{0.05, 0.10, 0.15, 0.20\}$, yielding 20 distinct combinations. Each combination was executed under three independent random seeds to reduce the influence of stochastic variation, resulting in 60 total runs. All remaining parameters were held fixed throughout the tuning phase: population size was set to 120, number of generations to 300, elite count to 4, tournament size to 4, local search iterations to 40, and local search application to the top 4 individuals per generation.

Each run was evaluated on the basis of the decoded solution fitness value and the number of pallets placed. To obtain a seed-robust performance estimate, results were aggregated by computing the average fitness across the three seeds for each parameter combination. Combinations producing anomalous outlier fitness values under one or more seeds were identified and excluded from the ranking to avoid distortion of the average.

The aggregated results revealed a clear separation among the top-performing combinations. The three combinations achieving the lowest average fitness were $p_c = 0.75$, $p_m = 0.05$, $p_c = 0.80$, $p_m = 0.05$, and $p_c = 0.70$, $p_m = 0.10$. All three achieved an average pallet count of 46.33 and produced the target pallet count of 46 in two out of three seeds, outperforming all remaining combinations. Based on these results, $p_c = 0.75$ and $p_m = 0.05$ were selected for all subsequent experiments. This combination produced the lowest average fitness across all tuning runs and is theoretically consistent with the group-based structure of the GGA: Falkenauer (1996) recommended keeping mutation rates low in grouping representations, as aggressive mutation disrupts well-formed pallet groups and undermines the structural advantages of group-level encoding. The moderate crossover rate ensures sufficient

recombination across generations while preserving high-quality pallet assignments inherited from parent chromosomes.

4. Data Collection

The dataset used in this study was obtained from TP Electric Electrical Materials Industry and Trade Inc., a manufacturer of low-voltage electrical equipment located in İzmir, Türkiye. The company regularly fulfills domestic and international orders involving heterogeneous product mixes shipped via road and sea freight, and the loading scenarios examined in this paper reflect real operational planning decisions made by the company's logistics department. Data was extracted directly from the company's warehouse management and enterprise resource planning systems and encompasses product dimensions, packaging classifications, and shipment quantities as recorded in actual outbound logistics operations. The dataset comprises 25 distinct box types and 18 distinct carton types. All products are pre-assigned to one of three operational pallet categories based on packaging type and dimensional compatibility. Box products are loaded exclusively onto P80_BOX pallets (80 × 120 cm floor, 120 cm max height for both road and sea). Carton products are distributed between P80_CARTON pallets (80 × 120 cm floor, 262 cm road / 250 cm sea) and P100_CARTON pallets (100 × 120 cm floor, 262 cm road / 250 cm sea), with the assignment determined by individual product dimensions and compatibility rules enforced by the company's warehouse management system. These pallet categories differ in allowable stacking heights due to structural and handling requirements. A fundamental operational constraint enforced throughout the loading process is that box and carton products must never be co-loaded onto the same pallet, as each pallet category reflects distinct handling, stacking, and stability requirements. Item rotation is permitted along all three axes during packing, and weight constraints are treated as non-binding given that volumetric capacity is the operative limiting factor in the company's logistics context. The available vehicle fleet consists of six configurations covering both sea and road transportation. For sea freight, two container types are used: a 20-ft DC Container (590 × 235 × 269 cm, 37.3 m³) and a 40-ft HC Container (1,198 × 235 × 269 cm, 75.7 m³). For road freight, four options are available: a standard Truck (720 × 245 × 270 cm, 47.7 m³), a 40-ft Truck (820 × 245 × 270 cm, 54.3 m³), an Optima Tilt semi-trailer (1,360 × 250 × 270 cm, 91.8 m³), and a Mega Tilt semi-trailer (1,360 × 245 × 300 cm, 100.0 m³). Each shipment order is assigned to exactly one vehicle type, and all loaded pallets must fit entirely within the selected vehicle's internal dimensional envelope.

5. Results and Discussion

All computational experiments were conducted on a machine equipped with an Intel 12th Gen Core I5-12500H processor and 16 GB of RAM. The optimization solver is implemented in C++ with OpenMP parallelism for population evaluation and local search. The algorithm parameters were held constant across all test instances. The population size was set to 120 and the algorithm was run for 300 generations. A crossover rate of 0.75 and a mutation rate of 0.05 were used, with an elite count of 4 and a tournament size of 4. The local search was configured with 40 iterations per call, applied to the top 4 individuals in the population at each generation.

5.1 Numerical Results

The proposed GGA was evaluated on 8 real shipment orders sourced from the company's operational data, spanning a range of order sizes from 174 to 3,797 items and covering destinations across four countries. For each instance, the algorithm output is compared against the company's manual loading plan (TP) and, where available, a volumetric theoretical lower bound (LB) computed as the ceiling of the total item volume divided by the usable pallet capacity (Table 1).

Table 1. Summary of computational results on real shipment instances

Case	Items	LB	TP	GGA	Pallet Savings	Time (s)
Holland (JACS)	1,332	5	8	7	1	178.3
Spain (APARELEC)	189	2	2	2	0	10
Sweden (BIRGMA)	590	11	29	16	13	6.5
Sweden (MALM. 2)	896	7	9	8	1	24.5

Sweden (MALM. 1)	1,172	16	21	19	2	14.9
Egypt	3,797	14	18	16	2	375.1
Ukraine	571	6	7	7	0	18.5
India	174	6	9	8	1	3.5

The GGA matched or improved the company's solution in all cases, achieving an average pallet count reduction of 12.5%. The largest improvement was observed in the Sweden (BIRGMA) instance, where the pallet count decreased from 29 to 16, corresponding to a 44.8% reduction. Holland (JACS) also showed a strong improvement, decreasing from 8 to 7 pallets. The remaining improved cases produced smaller but consistent reductions, ranging from 5.6% to 11.1%. The Spain (APARELEC) instance resulted in a tie, with both TP and GGA using 2 pallets and matching the theoretical lower bound. This indicates that for small and relatively homogeneous orders, the company's manual loading plan may already be optimal or near-optimal.

5.2 Graphical Results

Figure 3 presents the normalized gap closure performance of the proposed GGA relative to the company baseline and the theoretical lower bound. This metric is computed as $(TP - GGA) / (TP - LB)$, where TP is the company's manual pallet count, GGA is the algorithm's pallet count, and LB is the theoretical lower bound; a score of 1.0 indicates that the GGA reaches the lower bound exactly, while a score of zero indicates no improvement over the company baseline. The results show that the GGA closes a substantial part of the available gap in the Spain and Sweden BIRGMA instances, while achieving moderate improvement in the Holland, MALMBERGS, Egypt, and India cases. The Ukraine instance yields a score of zero, indicating that the GGA matches the company baseline without improvement or deterioration.

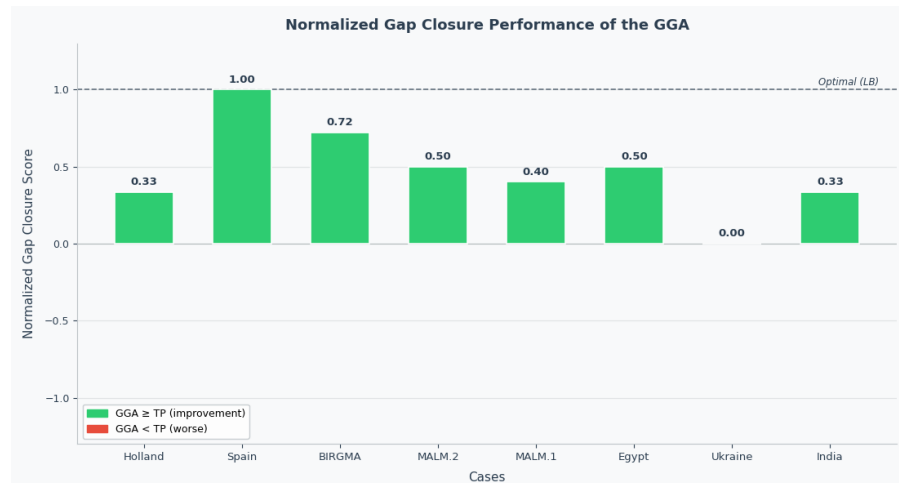


Figure 3. Normalized gap closure of the GGA relative to the company baseline and theoretical lower bound

Figure 4 illustrates the three-dimensional loading plan generated by the proposed GGA for the Sweden (MALM.1) instance, which comprises 1,172 items loaded onto 19 pallets within a 40-ft Truck. Each color represents a distinct pallet group, and the orange borders indicate pallet boundaries within the vehicle cargo envelope. The visualization confirms that the decoded solution produces a geometrically feasible and compact arrangement with no overlapping pallets and full containment within the vehicle dimensions.

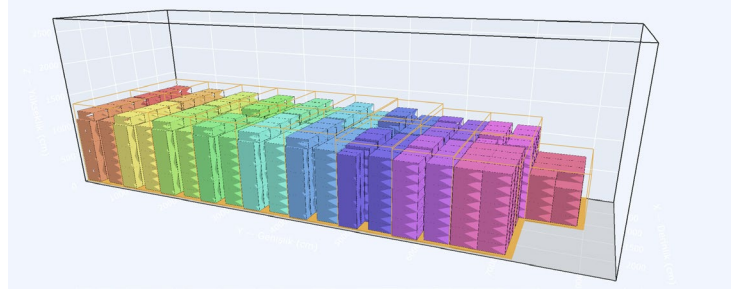


Figure 4. GGA Loading Plan — Sweden (MALM.1): 19 Pallets in 40-ft Truck

5.3 Proposed Improvements

Several directions for future improvement can be identified based on the computational results and the structural characteristics of the proposed framework.

The first direction concerns the placement heuristic used within the decode procedure. The current implementation relies exclusively on the Extreme Point (EP) heuristic for both pallet-level and vehicle-level packing. While EP has demonstrated competitive performance in the literature, alternative placement strategies such as the Maximal Empty Space (MES) heuristic or guillotine-cut based methods may produce more compact arrangements for certain item configurations. A systematic comparison of placement heuristics within the same GGA framework would clarify whether the EP heuristic represents the most effective decoding mechanism for the heterogeneous pallet loading setting studied here, or whether a hybrid placement strategy could yield further reductions in pallet count and improvements in volumetric fill rate.

The second direction concerns parameter configuration. The tuning study conducted in this work evaluated crossover and mutation rates across three random seeds on a single representative instance. A more comprehensive tuning procedure covering a wider parameter space, including population size, elite count, tournament size, local search depth, and fitness function weights, applied across multiple instances simultaneously, would provide stronger statistical guarantees for the selected configuration and potentially identify parameter settings that generalize better across diverse shipment structures.

The third direction concerns runtime scalability for large instances. The Egypt case (3,797 items) required 529 seconds, nearly three times longer than any other instance. An adaptive convergence criterion that reduces local search depth or population size once the improvement rate falls below a threshold could reduce computation time for large instances without meaningful loss in solution quality.

5.4 Validation

To assess the quality of the GGA solutions relative to optimality, the gap between the algorithm's output and the theoretical lower bound is computed for all eight instances. The lower bound gap is defined as:

$$\Delta\% = (GGA - LB) / LB \times 100$$

Across all eight instances, the GGA achieves an average gap of 22.8% above the lower bound, compared to 47.3% for the company plan. The GGA matches the lower bound exactly on the Spain instance ($\Delta\% = 0$) and achieves gaps below 20% on the MALM.2, MALM.1, and Egypt instances. The Holland instance shows a gap of 40.0%, and the India instance 33.3%, reflecting the difficulty of small mixed-type orders where a single partially filled pallet has a disproportionate effect on the gap metric. The BIRGMA instance, despite the largest absolute pallet reduction in the study (29 to 16), still shows a gap of 45.5% above the lower bound, consistent with the inherent consolidation difficulty of its mixed pallet-type structure.

These results confirm that the proposed GGA consistently produces solutions closer to the theoretical optimum than the company's current manual approach, while operating under the same real-world constraints of strict type compatibility, three-dimensional geometric feasibility, and single-vehicle capacity limits.

6. Conclusion

This study developed an integrated optimization framework for the hierarchical pallet and vehicle loading problem observed in real export logistics operations. The proposed approach combines a Grouping Genetic Algorithm (GGA) with an Extreme Point (EP) decoding heuristic to assign heterogeneous boxes to compatible pallet types, pack boxes in three dimensions, arrange loaded pallets within a selected vehicle, and determine the smallest feasible vehicle within a single solution framework.

The research objectives stated in Section 1.1 were achieved. First, a GGA-based framework was developed to simultaneously handle heterogeneous box-to-pallet assignment and pallet-to-vehicle loading with three-dimensional geometric feasibility. Second, the EP decoding procedure was implemented at both the pallet and vehicle levels to generate feasible loading plans efficiently. Third, the computational results confirmed that the fitness-guided mutation operators supported competitive solution quality by improving weak pallet groups and increasing packing efficiency.

The framework was tested on eight real shipment instances obtained from TP Electric. The proposed GGA matched or improved the company's manual loading plan in seven out of eight cases, achieving an average pallet count reduction of 12.5%. The largest improvement was obtained in the Sweden (BIRGMA) instance, where the pallet count decreased from 29 to 16, corresponding to a 44.8% reduction. Across all instances, the GGA achieved an average lower bound gap of 22.8%, compared to 47.3% for the company baseline, demonstrating that the proposed algorithm consistently produces solutions closer to the theoretical optimum than the company's current manual approach.

Despite these positive results, two limitations remain. First, the runtime for large-scale instances, such as the Egypt case with 3,797 items requiring 375.1 seconds, may be high for time-sensitive operational use. Second, mixed pallet-type orders remain challenging under strict compatibility constraints, as observed in the Ukraine instance. Future research may focus on systematic comparison of alternative placement heuristics within the GGA framework, more comprehensive parameter tuning across multiple instances, adaptive stopping criteria, improved consolidation procedures for under-filled pallets, multi-vehicle planning, and the inclusion of weight balance constraints for road transportation.

Acknowledgment

The authors would like to express their sincere gratitude to Prof. Dr. Ayhan Özgür Toy for his insightful comments, constructive feedback, and valuable guidance throughout this research. The authors also acknowledge TP Electric Electrical Materials Industry and Trade Inc. for providing the real-world data and operational insights that significantly contributed to this study.

References

- Alonso, M. T., Martinez-Sykora, A., Alvarez-Valdes, R., and Parreño, F., The pallet-loading vehicle routing problem with stability constraints, *European Journal of Operational Research*, vol. 302, no. 3, pp. 860–873, 2022.
- Bortfeldt, A. and Wäscher, G., Constraints in container loading – a state-of-the-art review, *European Journal of Operational Research*, vol. 229, no. 1, pp. 1–20, 2013.
- Côté, J. F., Guastaroba, G., and Speranza, M. G., The value of integrating loading and routing, *European Journal of Operational Research*, vol. 257, no. 1, pp. 89–105, 2017.
- Crainic, T. G., Perboli, G., and Tadei, R., Extreme point-based heuristics for three-dimensional bin packing, *INFORMS Journal on Computing*, vol. 20, no. 3, pp. 368–384, 2008.
- Elhedhli, S., Gzara, F., and Yildiz, B., Three-dimensional bin packing and mixed-case palletization, *INFORMS Journal on Optimization*, vol. 1, no. 4, pp. 323–352, 2019.
- Falkenauer, E., A hybrid grouping genetic algorithm for bin packing, *Journal of Heuristics*, vol. 2, no. 1, pp. 5–28, 1996.
- Falkenauer, E. and Delchambre, A., A genetic algorithm for bin packing and line balancing, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, pp. 1186–1192, 1992.
- Fuellerer, G., Doerner, K. F., Hartl, R. F., and Iori, M., Metaheuristics for vehicle routing problems with three-dimensional loading constraints, *European Journal of Operational Research*, vol. 201, no. 3, pp. 751–759, 2010.
- Gonçalves, J. F. and Resende, M. G. C., A biased random-key genetic algorithm for the container loading problem, *Proceedings of MIC 2013 – 10th Metaheuristics International Conference*, Singapore, 2013.

- Lodi, A., Martello, S., and Vigo, D., Heuristic algorithms for the three-dimensional bin packing problem, *European Journal of Operational Research*, vol. 141, no. 2, pp. 410–420, 2002.
- Martello, S., Pisinger, D., and Vigo, D., The three-dimensional bin packing problem, *Operations Research*, vol. 48, no. 2, pp. 256–267, 2000.
- Rohlfshagen, P. and Bullinaria, J. A., A grouping genetic algorithm with exon shuffling crossover for bin packing, *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, London, UK, pp. 1529–1536, 2007.
- Ruan, Q., Zhang, Z., Miao, L., and Lim, A., A hybrid approach for the mixed load vehicle routing problems with time windows, *Flexible Services and Manufacturing Journal*, vol. 25, no. 3, pp. 374–402, 2013.
- Wäscher, G., Haußner, H., and Schumann, H., An improved typology of cutting and packing problems, *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.
- Yang, X., Ruan, J., Chen, Z., Liu, K., and Tang, X., Optimization of truck–cargo matching for the LTL logistics hub based on three-dimensional pallet loading, *Mathematics*, vol. 12, no. 21, pp. 3336, 2024.
- Zhao, X., Liu, W., Yamamoto, H., and Zheng, X., Empty-space-based heuristic algorithms for the three-dimensional strip packing problem with three open dimensions, *Applied Sciences*, vol. 11, no. 21, pp. 10484, 2021.
- Zheng, Y., Chien, C. F., and Gen, M., Multi-objective genetic algorithm with EP-based local search for 3D pallet loading, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2132–2137, 2015.

Biographies

Müfide Defne Akbay is a senior undergraduate student of Industrial Engineering at Yaşar University, İzmir, Türkiye. Her academic work focuses on operational research, optimization and logistics systems. Her research interests include combinatorial optimization and metaheuristic algorithms, particularly their applications in palletization, vehicle loading, and supply chain optimization.

Duru Atılğan is a senior undergraduate student of Industrial Engineering at Yaşar University, İzmir, Türkiye. Her academic interests include operations research, optimization, production planning, and supply chain management. Her studies focus on developing analytical and data-driven solutions for real-world industrial engineering problems.

Eren Efe Bulut is an undergraduate student in the Department of Industrial Engineering at Yaşar University. His areas of interests are OR, network optimization and scheduling. He has also completed short-term internships in the fields of production and quality; gaining practical experience in ERP-based inventory tracking, production planning and quality control approaches.

Ayça Kılıçarslan is a senior undergraduate student of Industrial Engineering at Yaşar University, İzmir, Türkiye. Her research interests include combinatorial optimization and metaheuristic algorithms, particularly their applications in palletization, vehicle loading, and supply chain optimization.

Gökçe Başak Salt is a senior undergraduate student of Industrial Engineering at Yaşar University, İzmir, Türkiye. She is interested in operations research, production planning, and logistics optimization. Her studies mainly involve mathematical modeling and manufacturing systems.

Bartu Şıracı is a senior undergraduate student of Industrial Engineering at Yaşar University, İzmir, Türkiye. His academic work focuses on optimization and operations research. His research interests include combinatorial optimization, metaheuristic algorithms, and their applications in logistics and production systems.