



QUBO formulation for aircraft load optimization

Laura Gatti^{1,2} · Rafael Sotelo^{1,2} · Juan Orihuela² · Diego Gibert² · Renzo D'ambrosio¹ · Federico Fuidio¹

Received: 30 December 2023 / Accepted: 2 October 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

In this article, we tackle the aircraft load optimization problem using classical optimization algorithms and optimization algorithms with QUBO (quadratic unconstrained binary optimization) formulation to run on quantum annealers. The problem is realistic based on plans of a certain aircraft model, the Airbus A330 200F, and can be adapted to other models from other manufacturers. We maximize a characteristic of the combination of containers (unit load device, ULD) to be transported, be it weight, volume, profit, or another, while complying with necessary parameters related to the flight such as the balance of the center of gravity as well as stress in the structure. Finally, examples of the results of different runs on QUBO in the D-Wave simulator are presented.

Keywords Aircraft load optimization · Quantum annealing · QUBO · Quantum computing

Laura Gatti, Rafael Sotelo, Juan Orihuela, Diego Gibert, Renzo D'ambrosio and Federico Fuidio have contributed equally to this work.

✉ Laura Gatti
lgatti@correo.um.edu.uy

Rafael Sotelo
rsotelo@um.edu.uy

Juan Orihuela
jorihuela@quantum-south.com

Diego Gibert
dgibert@quantum-south.com

Renzo D'ambrosio
rdambrosio@correo.um.edu.uy

Federico Fuidio
ffuidio@correo.um.edu.uy

¹ ICT Department, Universidad de Montevideo, Av. Dra. María Luisa Saldún de Rodríguez 2097, 11500 Montevideo, Uruguay

² RD Department, Quantum-South, Ponce 1307, 11000 Montevideo, Uruguay

1 Introduction

The air cargo industry plays a crucial role in global trade. During the pandemic, cargo became fundamental to airline revenues. Before the pandemic, cargo contributed to 15% of airline revenue, growing to 40% by 2021 [1]. Its turnover in 2022 was expected to exceed USD 200 billion [2], though it was projected to decrease to USD 149.9 billion in 2023. Air cargo moves about 60 million tons annually.

However, the airline industry demonstrates differing efficiencies in transporting passengers versus cargo. While the passenger load factor is approximately 79% [3], the cargo load factor is below 43% [4], indicating that the industry's efficiency in cargo is nearly half that of passenger transport. This disparity can be attributed to various factors, including operational and sales complexities and differing levels of process digitalization. Therefore, studying how new optimization techniques can enhance efficiencies in the air cargo industry is worthwhile. Given the annual revenue of the industry, even small percentage increases in efficiency can result in significant additional income or savings.

The importance of this problem is highlighted by Airbus's proposal of the Airbus Quantum Computing Challenge in 2019 [5], seeking to test and assess new computing capabilities to solve some of its most complex problems.

In this article, we explore optimization applications using quantum computing, specifically the QUBO technique, for cargo aircraft. To make the study realistic, we use the Airbus A330 200F model, a widely used full-freighter model, as our implementation example.

Quantum computing is a new and powerful technology expected to disrupt various industries [6]. Its versatility can significantly impact logistics, solving complex problems in parcel processes, routing, and priority selection that were previously not scalable. "Quantum annealing" is one approach in quantum computing [7], focusing on solving optimization problems using quantum tunneling, guided by the adiabatic theorem [8]. This differs from traditional methods like simulated annealing, which rely on thermal fluctuations [9]. Quantum annealing is limited to specific problems expressed as quadratic unconstrained binary optimization (QUBO) [9].

Recent studies have explored quantum computing in the aerospace industry [10, 11], highlighting discoveries and advantages in flight trajectory optimization, quantum sensing, and logistics challenges [12, 13]. The Airbus Quantum Computing Challenge provided insights into aircraft load optimization [5]. Airbus simplified the real problem by standardizing unit load devices (ULDs) to sizes 1, 2, and 1/2 relative to a reference length, aiming to maximize load while ensuring flight stability [14].

In our study, we introduce a QUBO model tailored to address aircraft load optimization with real-world relevance. We utilize data from actual aircraft, featuring multiple cargo bays with standardized ULDs commonly used in the industry. Unlike the simplified version, the real problem considers additional constraints imposed by actual geometry.

To achieve valid loading configurations, we leverage the problem's symmetries to avoid a direct assignment problem [14]. Our approach involves a hybrid solution, where quantum subroutines [15–17] tackle combinatorial challenges known for their high computational demands on classical processors [18]. These quantum routines are

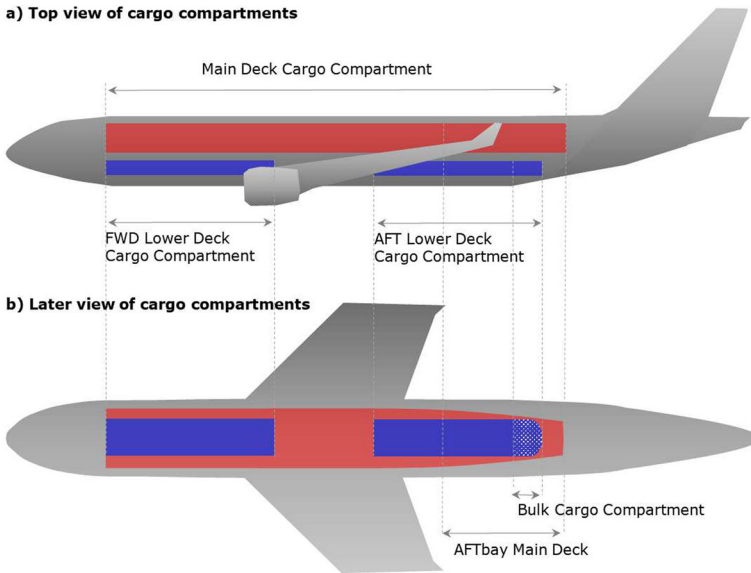


Fig. 1 Lateral (a) and top (b) views of the main deck, AFT lower deck, FWD lower deck

combined with classical subroutines responsible for finalizing the load configuration and conducting check functions.

This strategy significantly reduces the number of binary variables and, consequently, the number of qubits needed, enabling the implementation of our complete model on various backends, including real quantum annealers like D-Wave [19] and well-established classical solvers such as Gurobi [20].

The main findings of this paper are that our hybrid approach utilizing both quantum and classical subroutines demonstrates that quantum annealers can effectively handle real-world cargo optimization problems. The results obtained from quantum annealers are comparable to those from classical solvers, showcasing the potential of quantum computing in logistics optimization. This study highlights the importance of considering problem symmetries and leveraging advanced techniques to reduce the complexity and number of variables in QUBO formulations.

In Sect. 2, we outline the problem specifications, along with the assumptions and constraints considered in developing the model, as well as the use of QUBO modeling to solve this class of problems. Section 3 introduces the overall structure of the solution, detailing the model and the classical subroutines supporting the QUBO problems. The obtained results using various solvers are presented in Sect. 4, and the solution’s pros and cons, along with potential enhancements, are discussed in Sect. 5.

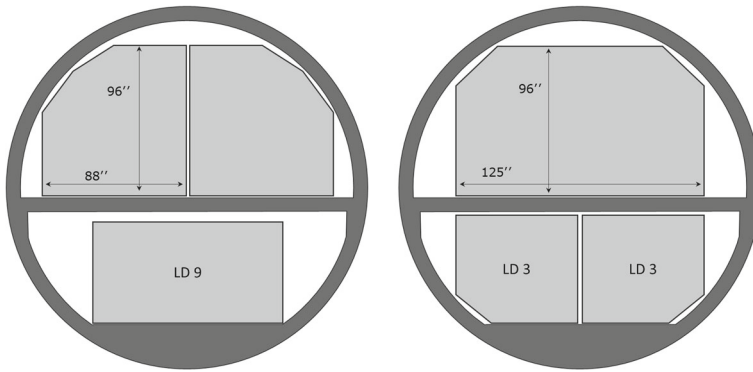


Fig. 2 Cross section of the cargo bays

2 Problem description

2.1 Cargo aircraft specifications

Aircraft specifically designed for cargo operations distributes their cargo capacity among various storage bays. A typical example is the Airbus A330-200F, which comprises a main deck, two lower decks, and a cargo loading space, referred as "bulk." See Fig. 1.

Except for bulk cargo, which allows the transportation of loose packages, all items in the main bays must be organized into containers with predefined shapes or onto pallets with standardized dimensions. Containers and pallets will be referred to as ULDs (unit load device). The configuration of both containers and assembled pallets follows the circular shape of the aircraft fuselage. Refer to Fig. 2 for a schematic representation of the circular cross section of the cargo bays.

In air cargo, there are two main formats of ULDs: containers and pallets. While containers have rigid structures, pallets only have a base where packages are stacked and covered with a mesh that secures the packages to the base. Despite the standardization of container dimensions and pallet base size, there exists a considerable diversity of ULDs compared to sea cargo containers due to the varying fuselage radii across different models of cargo aircraft. The A330-200F model accommodates nearly all existing types of ULDs, making it compatible with a wide range due to its modern design. In Appendix A, Tables 2 and 3 enumerate all the chosen types of containers and pallets for this model, selected based on their widespread usage.

Several important specifications for modeling include:

- ULDs suitable for the main deck are incompatible with those suitable for the lower decks, resulting in disjoint sets for selection. Therefore of the total available N_T ULDs, N denotes those available for the main deck, and \hat{N} denotes those available for the lower deck.
- On the main deck there are ULDs that can be placed in *side-by-side* configurations or *single row* configurations occupying the entire length of the main deck by its own. Figure 3 may help to visualize this.

- ULDs suitable for side-by-side placement on the main deck can also be placed in a single-row configuration by rotating them 90 degrees.

2.2 Resources and constraints of the problem

The main problem can be stated as follows: Let there be N_T ULDs to be transported. Each unit has a number of attributes:

- A weight associated w_i .
- Width and length dimensions d_i, l_i , respectively.
- An associated profit p_i .
- A unique identifier of the package type Id_i

We add an attribute to each unit a new attribute called *type* that indicates whether the container can go side by side or can only go in a single row.

- Type associated t_i which takes the value $t_i = 1$ if the ULD can be loaded side by side, or the value $t_i = 2$ if this is not possible.

The objective is to identify the optimal selection of the ULDs to be transported that maximizes the profit transported in each aircraft, to a set of restrictions that we list:

1. *Weight restriction* Each deck supports a maximum weight that must be respected.
2. *Volume restriction* There is a maximum volume of ULDs that can be transported on each deck.
3. *AFTbay restriction* At the tail of the aircraft, on the main deck, there is an area which, due to the width reduction of the fuselage, it is impossible to fit side by side ULDs.
4. *Center of gravity* The center of gravity of the load carried on the aircraft has to be within a specific range with respect to a target point on the length of the aircraft, in order to make the flight safe and to minimize fuel consumption.
5. *Shear force restriction* In the same way, the load must be distributed in such a way that it respects the maximum shear force curve given by the manufacturer, see Fig. 4. Shear force is caused by the weight of the ULDs. To calculate the shear force the plane is modeled as two cantilevered beams supported at the center of the main deck. The lower decks are supported by these same beams; this means that the shear force from the lower decks can be simply summed. A detailed explanation of how shear is calculated for a given A330-200f aircraft configuration is given in Appendix B.

2.3 QUBO modelling

QUBO (quadratic unconstrained binary optimization) modeling is particularly useful in quantum annealing, as quantum annealers are naturally suited for quadratic optimization problems. It can be used to solve combinatorial problems. Unlike traditional linear formulations, QUBOs focus on finding an optimal solution by minimizing a quadratic objective function without explicit constraints. The goal is to find a binary

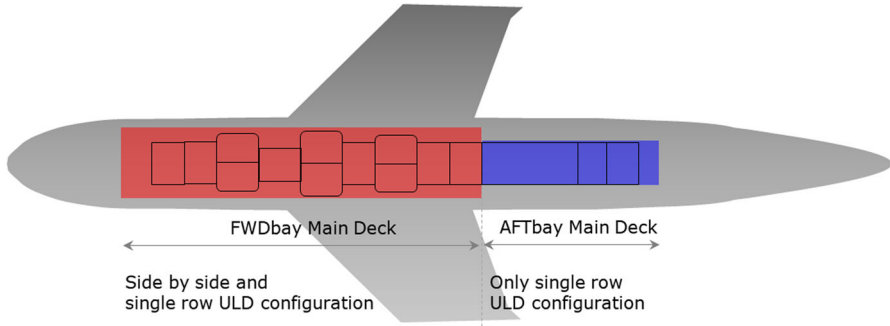


Fig. 3 *Aft bay of the main deck* Due to fuselage width reduction, side-by-side ULDs are not feasible in the blue area. In the remaining red section, a potential loading configuration with side-by-side configurations is exemplified

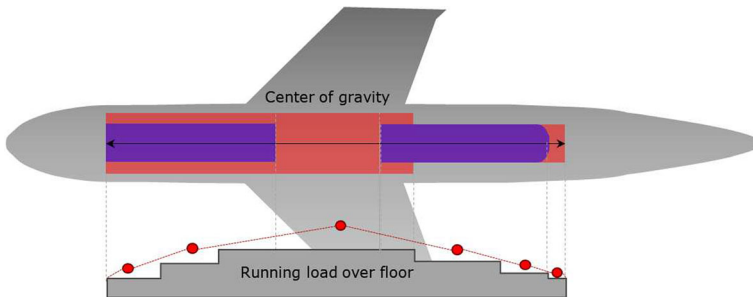


Fig. 4 *Shear representation of the shear force caused by the weight of the ULDs in both bays*

vector x that minimizes $x^T Qx$, where Q is a symmetric matrix representing the interactions between binary variables.

To convert a linear problem to a QUBO, both the objective function and constraints are transformed into quadratic terms. The matrix Q is decomposed into $Q = -Q_{\text{objective}} + \alpha Q_{\text{constraint}}$, where $Q_{\text{objective}}$ maximizes the objective function and $Q_{\text{constraint}}$ represents the constraints, with α adjusting the penalty.

Constraints are handled by minimizing the error of unmet constraints. Instead of ensuring exact compliance, penalty terms are added, turning linear constraints into quadratic ones. For example, a linear constraint $w^T x = W$ can be rewritten as $(w^T x - W)^2$, which is a quadratic form. This quadratic term is zero when the constraint is satisfied but adds positive terms to the objective function when the equality $w^T x = W$ is violated. This penalizes solutions that do not meet the constraint.

In the case of working with inequality constraints of the type $w^T x \leq W$, it is necessary to use slack variables to ensure that $w^T x$ is less than or equal to W . For example, if we assume that w_i are integers, and W is a power of two minus one, e.g., $W = 15$, $W = 63$, we can rewrite this constraint as a quadratic form by imposing the penalty

$$\left(w^T x + \sum_{i=0}^{\log_2(W)} 2^i q_i - W \right)^2$$

where q_i are auxiliary variables named slack variables. The term $\sum_{i=0}^{\log_2(W)} 2^i q_i$ represents the binary expansion of any number between 0 and W . In the worst-case scenario, where the binary vector x is null (a poor solution that still respects the constraint), all slack variables q_i are activated, summing to W and nullifying the total sum. In the best-case scenario, where $w^T x = W$, all slack variables remain inactive, resulting in a total sum of zero. If $w^T x > W$, the slack variables cannot compensate, adding positive terms to the objective function and penalizing this solution. If W is not a power of two, the quasi-binary¹ expansion can be used.

This technique allows the QUBO formulation to capture the essential characteristics of the original problem. The final matrix Q integrates both the objective function and quadratic constraints, ensuring the optimal solution to $x^T Q x$ matches the original problem's solution.

3 Solution

The architecture of the solution aims to reduce the usage of variables by decoupling the main and lower bays, as well as the selection process of unit load devices (ULDs) during their positioning into each bay. Despite the decoupling, the ULD selection considers physical bay constraints, ensuring a feasible ULD positioning. This step is represented by a QUBO model adaptable to different backends.

Within each bay, ULD positioning encompasses a two-step process involving a QUBO formulation and a classical routine. The first part (QUBO) makes a selection of the containers to be transported from all the available ones, and it assigns them to groups for placement in different loading bay sections to meet shear and center of gravity constraints. Following this, the classical algorithm selects the optimal combination, minimizing the center of gravity distance to the desired target, considering various main and lower bay arrangements.

If the desired tolerance is not achieved, the algorithm can be rerun with a different selection of suboptimal containers to meet the parameters. The solution architecture is summarized in Table 1.

¹ The quasi-binary representation of a number c is obtained by taking M Boolean variables y_k with $M = \lfloor \log_2(c) \rfloor$ for its binary expansion in which the coefficient of the most significant term y_M is changed from 2^M to $(c + 1 - 2^M)$. Therefore, the range of values that can be represented with the Boolean variables goes from 0 to c . In this representation, values smaller than c can have more than one valid configuration.

Table 1 Solution architecture diagram

Main deck	Lower deck
ULDs selection main	ULDs selection lower
↓	↓
ULDs positioning main	ULDs positioning lower
↓	↓
Global load configuration	
Restriction check	

3.1 ULDs selection

3.1.1 Main deck

The problem is formulated as a binary optimization problem. To each ULD in the warehouse, the binary variable $x_i \in \{0, 1\}$ is introduced for $1 \leq i \leq N$. Each ULD has specific attributes, including profit p_i , weight w_i , size l_i , unique identification number Id_i , and type t_i . When $x_i = 1$, it signifies that the ULD i is selected for loading.

To allow for the placement of ULDs side by side in a single central row, ULDs of type 1 are represented twice. For each container i with $t_i = 1$, a new container j is created. This new container has a width $d_j = l_i$, length $l_j = d_i$, type $t_j = 2$, and retains the same identifier and profit as ULD i ($Id_j = Id_i, p_j = p_i$).

Taking into account the attributes of each ULD and the physical loading constraints described in 2.2, along with the variables introduced in this section, the problem of selecting ULDs for the main deck can be formulated as follows:

$$\max \sum_{i=1}^N p_i x_i \tag{1.a}$$

subject to

$$\sum_{i=1}^N w_i x_i \leq C_{MD} \tag{1.b}$$

$$\sum_{i=1}^N t_i l_i x_i \leq 2l_{MDmax} \tag{1.c}$$

$$\sum_{i=1}^N (2 - t_i) x_i = 2k \text{ with } k \in \mathbb{N} \tag{1.d}$$

$$\sum_{i=1}^N \sum_{j<i}^n \delta_{Id_i Id_j} x_i x_j = 0 \tag{1.e}$$

$$\sum_{i=1}^N (t_i - 1)l_i x_i \geq AB \tag{1.f}$$

Equation (1.a) represents the objective function, which aims to maximize the profit of the total selected cargo. On the other hand, Eqs. (1.b) to (1.e) impose constraints on the selection of ULDs to respect the physical capacities of the aircraft, and each of them will be explained separately.

The constraint (1.b) ensures that the total weight of the selected ULDs does not exceed the maximum weight supported by the main deck, represented by the parameter C_{MD} .

Similarly, constraint (1.c) ensures that the selected ULDs do not exceed the aircraft’s volume capacity. To achieve this objective and with the motivation to reduce variables, the volume constraint was translated into a constraint on the longitudinal occupation of the main deck. This was achieved by considering that, on the main bay, ULDs of the single-row type ($t_i = 2$) occupy their entire corresponding longitudinal space l_i , whereas those of the side-by-side type ($t_i = 1$) occupy half of their longitudinal space, so that two ULDs placed side by side occupy the longitudinal space of one of them.

Therefore, when a ULD is placed in the single-row configuration ($t_i = 2$), its length l_i is multiplied by two, whereas when two ULDs are placed side by side ($t_i = 1$), their length l_i is multiplied by one, and by adding the length of each ULD placed side by side twice, we get $2l_i$. Given the configurations of the selected ULDs, the sum of these must be less than twice the maximum length of the main deck, represented by the parameter l_{MDmax} .

The decision to impose the volume constraint in the aforementioned manner implies that a new constraint must be added (1.d). This constraint ensures that the number of items placed side by side is even, preventing the occurrence of an empty space on only one side. This arrangement is unfavorable for lateral center of gravity and loading efficiency. When a container is of the single-row type, the term $2 - t_i$ is 0, while for side-by-side containers, it is 1. Therefore, the sum over i of these terms counts the number of side-by-side containers and ensures that it is even.

Since ULDs of the side-by-side type can be placed as single row, and to achieve this, as explained, a duplication of this container was made using another variable with the same identifier, constraint (1.e) prohibits the same ULD from being selected twice (it is either selected as single row or as side by side). Here δ is the Kronecker delta function and $\delta_{I_d_i I_d_j} = 1 \iff I_d_i = I_d_j$, and zero in any other case.

Finally, constraint (1.f) ensures the occupation of the AFTbay area. As explained in Sect. 2.2, due to the narrowing of the tubular section of the aircraft near the tail, there is a part of the main bay where side-by-side containers cannot be placed. Therefore, it must be ensured that there are a sufficient number of single-row containers to cover this space, and this is what this constraint does, as it only sums the length of the selected single-row containers. AB is the length of the AFTbay area.

QUBO formulation As explained in Sect. 2.3, the initial formulation of the problem (1.a) must be translated into a QUBO model in order to solve this optimization problem on a quantum annealing. The objective function (1.a) can be represented as a diagonal matrix in the variables x_i and therefore as a symmetric matrix ($Q_{\text{objective}}$).

The constraints 1.b, 1.c, and 1.f are inequality constraints that can be formulated as quadratic forms using the approach proposed in Sect. 2.3. The only difference is that since C_{MD} , $l_{MD_{max}}$, and AB are not generally powers of two, their quasi-binary representation must be used. The quadratic forms associated with the constraints 1.b, 1.c, and 1.f are therefore 2, 3, and 4, respectively:

$$H_{w_M} = A \left(\sum_{i=1}^N w_i x_i + \left(\sum_{j=0}^{M-1} 2^j y_j + (C_{MD} + 1 - 2^M) y_M \right) - C_{MD} \right)^2 \tag{2}$$

$$H_{v_M} = B \left(\sum_{i=1}^N t_i l_i x_i + \left(\sum_{j=0}^{R-1} 2^j z_j + (2l_{MD_{max}} + 1 - 2^R) z_R \right) - 2l_{MD_{max}} \right)^2 \tag{3}$$

$$H_{AB} = C \left(\sum_{i=1}^N (t_i - 1) l_i x_i - \left(\sum_{j=0}^{K-1} 2^j s_j + (AB + 1 - 2^K) s_K \right) - AB \right)^2 \tag{4}$$

where $M = \lfloor \log_2(C_{MD}) \rfloor$, $R = \lfloor \log_2(2l_{MD_{max}}) \rfloor$ and $K = \lfloor \log_2(AB) \rfloor$ denotes the required number of qubits for the quasi-binary representation of C_{MD} , $2l_{MD_{max}}$ and AB , respectively.

Constraint (1.e) is already a quadratic form by construction, making it straightforward to obtain the corresponding symmetric matrix.

$$H_{ev_M} = D \left(\sum_{i=1}^N (2 - t_i) x_i - \sum_{j=1}^Q 2^j p_j \right)^2 \tag{5}$$

$$H_{rep} = E \sum_{i=1}^N \sum_{j=1}^n \delta_{I d_i I d_j} x_i x_j \tag{6}$$

where $R = \lfloor \log_2(2l_{MD_{max}}) \rfloor$, $K = \lfloor \log_2(AB) \rfloor$, and Q is the smallest number to represent the maximum ULDs in side-by-side position entering the aircraft. The summation in Eq. (5) starts from 1, allowing only even numbers. The constants B, C, D, and E (like A) are positive real values that need adjustment to assign different weights to components.

Therefore, the problem (1.a) can be written as a quadratic form:

$$H_M = - \sum_{i=1}^n p_i x_i + H_{w_M} + H_{v_M} + H_{AB} + H_{ev_M} + H_{rep} \tag{7}$$

The objective function (1.a) yields negative values, while the penalties from the constraints are always non-negative. When a solution satisfies all constraints, the penalties are 0. Hence, solving (1.a) translates to minimizing the quadratic form (7).

The total number of qubits used in this formulation is $n_{single\ row} + 2n_{side\ by\ side} + (M + 1) + (R + 1) + (K + 1) + Q$.

3.1.2 Lower deck

The lower deck comprises two cargo compartments (refer to Fig. 1), front and back. Apart from this distinction, the problem mirrors the main deck cargo. Constraints on maximum weight and volume between the two bays persist in the ULD selection, with values c_{LD} and l_{LDmax} , respectively. The absence of an AFTbay and a limited set of ULD types, denoted by \hat{N} , simplifies certain algorithmic design choices. Specifically, no ULDs will be rotated, and some side-by-side ULDs will be allowed to travel individually.

To maintain the volume restriction, controlled solely by the ULD length as they cannot be rotated, an additional cargo unit $x_{\hat{N}+1}$ is introduced. This unit has width $d_{\hat{N}+1} = l_i$, length $l_{\hat{N}+1} = d_i$, where l_i and d_i are the dimensions of ULDs that can be placed side by side. It is designated as type $t_{\hat{N}+1} = 1$, with weight $w_{\hat{N}+1} = 0$, profit $p_{\hat{N}+1} = 0$, and a new identifier.

$$\max \sum_{i=1}^{\hat{N}+1} p_i x_i \tag{8.a}$$

subject to

$$\sum_{i=1}^{\hat{N}+1} w_i x_i \leq c_{LD} \tag{8.b}$$

$$\sum_{i=1}^{\hat{N}+1} t_i l_i x_i \leq 2l_{LDmax} \tag{8.c}$$

$$\sum_{i=1}^{\hat{N}+1} (2 - t_i) x_i = 2k \text{ with } k \in \mathbb{N} \tag{8.d}$$

Therefore the quadratic form is:

$$H_{w_L} = \hat{A} \left(\sum_{i=1}^{\hat{N}+1} w_i x_i + \left(\sum_{j=0}^{\hat{M}-1} 2^j y_j + (c_{LD} + 1 - 2^{\hat{M}}) y_{\hat{M}} \right) - c_{LD} \right)^2 \tag{9}$$

$$H_{v_L} = \hat{B} \left(\sum_{i=1}^{\hat{N}+1} t_i l_i x_i + \left(\sum_{j=0}^{\hat{R}-1} 2^j z_j + (2l_{LDmax} + 1 - 2^{\hat{R}}) z_{\hat{R}} \right) - 2l_{LDmax} \right)^2 \tag{10}$$

$$H_{ev_L} = \hat{D} \left(\sum_{i=1}^{\hat{N}+1} (2 - t_i) x_i - \sum_{j=1}^{\hat{Q}} 2^j p_j \right)^2 \tag{11}$$

where $\hat{M} = \lfloor \log_2(C_{LD}) \rfloor$ $\hat{R} = \lfloor \log_2(2l_{MLmax}) \rfloor$ and Q is the same as above.

$$H_L = - \sum_{i=1}^{\hat{N}+1} p_i x_i + H_{wL} + H_{vL} + H_{evL} \tag{12}$$

3.2 ULDs positioning

Following the previous step, we now have the set of ULDs for both the main deck and the lower deck, adhering to weight and volume restrictions while maximizing profit. The positioning process aims to achieve an optimal load distribution, considering the physical characteristics of the aircraft. The primary goal is to align the center of gravity as closely as possible to the target, while respecting the total shear of the aircraft.

The positioning system unfolds in the following steps:

1. *Main deck partitioning* The set of main deck ULDs is divided into front and back sections, ensuring that the weight of each section allows for the placement of the center of gravity at the desired position. The partitioning aims to distribute volume proportionally, taking into account the AFTbay part in the back section.
2. *Heuristic positioning* A positioning heuristic is applied, placing denser ULDs closer to the center of gravity, while progressively positioning lighter ones toward the extremes.
3. *Center of gravity calculation* The center of gravity and shear curve of the main deck are calculated. If the center of gravity deviates significantly from the target, iterations are made to refine the arrangement.

3.2.1 Main deck

Main deck partition The quantum aspect of the optimization algorithm is responsible for dividing the selected ULDs into two groups, each associated with a Boolean variable x_i . This variable functions as a label: if $x_i = 0$, ULD i is placed in the front group, and if $x_i = 1$, it goes to the back group. The grouping is designed to accommodate the aircraft’s physical constraints and fulfill design criteria, including ensuring an even number of type 1 ULDs in each subgroup.

The equations that govern this partition are as follows:

$$\sum_{i=1}^N w_i x_i = P_c * C_{RMD} \tag{13.a}$$

$$\sum_{i=1}^N t_i l_i x_i \leq 2l_{MDback} \tag{13.b}$$

$$\sum_{i=1}^N t_i l_i (1 - x_i) \leq 2l_{MDfront} \tag{13.c}$$

$$\sum_{i=1}^N (t_i - 1)l_i x_i \geq AB \tag{13.d}$$

$$\sum_{i=1}^N (2 - t_i)x_i = 2k \text{ with } k \in \mathbb{N} \tag{13.e}$$

N is the total number of ULDs selected. In equation (13.aa), Pc is the percentage of the total weight to be put on the back of the aircraft over the real weight carried C_{RMD} . This equation, unlike the rest, is an equality. As it is already known that this group of ULDs fits in the airplane, the partition by volume (both back and front, equations (13.b) and (13.c), respectively) seeks that the volume put in both groups is balanced without being an exact partition in volume; hence, they are inequalities. It should be noted that in equation (13.c) a change of variable $1 - x_i$ is made in order to take into account the ULDs of the front group.

The equation (13.d) imposes that in the back part enough ULDs of type 1 are placed to complete the AFTbay (this condition can be fulfilled, since it was taken into account in the selection of ULDs). Finally, equation (13.e) requires that the number of type 1 ULDs in the back group is even. As the total number of type 1 ULDs in the selection is even, this condition imposes in addition that there is an even number of type 1 ULDs in the front group.

Therefore the QUBO formulation of this problem can be obtained as:

$$I_{wM} = A \left(\sum_{i=1}^N w_i x_i - P_c C_{RMD} \right)^2 \tag{14}$$

$$I_{vMB} = B \left(\sum_{i=1}^N t_i l_i x_i + \left(\sum_{j=0}^{R-1} 2^j z_j + (2l_{MDback} + 1 - 2^R)z_R \right) - 2l_{MDback} \right)^2 \tag{15}$$

$$I_{vMF} = C \left(\sum_{i=1}^N t_i l_i (1 - x_i) + \left(\sum_{j=0}^{P-1} 2^j z_j + (2l_{MDfront} + 1 - 2^P)z_R \right) - 2l_{MDfront} \right)^2 \tag{16}$$

$$I_{AB} = D \left(\sum_{i=1}^N (t_i - 1)l_i x_i - \left(\sum_{j=0}^{K-1} 2^j s_j + (AB + 1 - 2^K)s_K \right) - AB \right)^2 \tag{17}$$

$$I_{ev} = E \left(\sum_{i=1}^N (2 - t_i)x_i - \sum_{j=1}^N 2^j p_j \right)^2 \tag{18}$$

where $R = \lfloor \log_2(2l_{MDback}) \rfloor$, $P = \lfloor \log_2(2l_{MDfront}) \rfloor$, $K = \lfloor \log_2(AB) \rfloor$ and Q is the smallest number to represent the maximum number of ULDs in side-by-side position entering in $R = \lfloor \log_2(2l_{MDback}) \rfloor + 1$. $A, B, C, D,$ and E are constants that balance the weight of each of the components of the partition. Finally the complete quadratic form is:

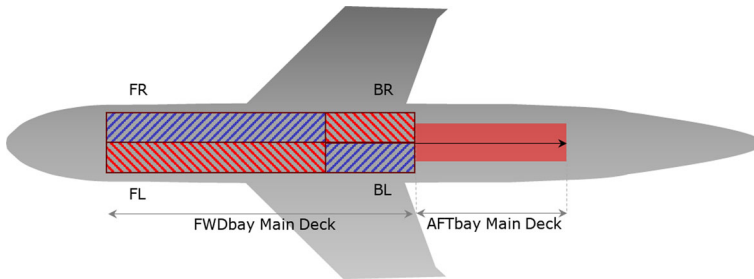


Fig. 5 Positioning main deck Partitioning the selection of ULDs for the main deck into two groups, FWDbay and AFTbay, this partition takes into account the constraint specified for AFTbay in Sect. 2.2

$$I_M = I_{w_M} + I_{v_{MB}} + I_{v_{MB}} + I_{AB} + I_{ev} \quad (19)$$

Main deck positioning Once the ULDs are partitioned in two sets, a classical algorithm is used to determine the exact positions of each ULD in the plane. The algorithm starts by defining 4 arrays of ULDs inside the plane: front right and left and back right and left. Both the main and the lower deck are divided in this way.

In the initial step, the tail of the plane (AFTbay) is filled by selecting type 2 ULDs with lower density, covering the bay from back to center based on density.

The subsequent step involves positioning the remaining ULDs separately for the front and back (see Fig. 5). An iterative process selects the denser unordered ULD for positioning inside the plane. For type 1 ULDs, the best position is determined by examining left and right arrays separately. Type 2 ULDs are positioned simultaneously on both sides, seeking adjacent and free positions. Positions are searched from the center to the tail or front, prioritizing the center for heavier ULDs. Before confirming a position, it is ensured that the next ULD to order will fit; otherwise, the current ULD is returned for reordering.

Following the initial positioning, an iterative process adjusts the center of gravity (CoG) in the x-axis. ULDs are swapped in pairs from the front or tail, moving the denser one further away. Changes are validated by recalculating the CoG. The procedure is repeated for the remaining ULDs further from the center. Type 1 ULDs at the array ends are simultaneously switched to the center.

Once the ULD order is established, arrays are repositioned to utilize unused space at the back. This iterative process adjusts the unused space either to the front or between front and back partitions, optimizing the CoG with a learning rate of 1 for each correct decision.

The pseudocode for the aforementioned algorithms can be found in Algorithms 1 and 2.

3.2.2 Lower deck

The lower deck positioning strategy closely mirrors that of the main deck. Initially, the total cargo is divided into two groups utilizing a quantum algorithm. It is crucial in this step to observe the maximum volume for each compartment due to physical

Algorithm 1 First positioning within half a deck

Require: $N \geq 0$ ▷ Number of ULDs to be loaded
 Complete AftBay with ULDs of type 2 and lower density.
 $N \leftarrow N - n'$
while $N \neq 0$ **do**
 Select the highest density ULD
 if ULD type = 1 **then** ▷ Either top or low
 Place in the first free position near the center in only one row
 else
 Place in the first free contiguous position near the center in both rows
 end if
 if next ULD will fit **then**
 $N \leftarrow N - 1$
 else
 Unplace the ULD and send it to the queue to be ordered again
 end if
end while

Algorithm 2 CoG improvement

Require: K defined positions on half deck
 $old|CoG| \leftarrow$ Calculate the CoG of the entire deck
 $k \leftarrow K$
while $k - 1 \geq 0$ **do**
 Swapping of ULDs in position k and k-1 ▷ Two type 1 ULDs occupying the main and lower rows are taken as a block.
 $new|CoG| \leftarrow$ Calculate the CoG of the entire deck
 if $old|CoG| \leq new|CoG|$ **then**
 Break
 end if
 $k \leftarrow k - 1$
end while

separation. Subsequently, the unit loads are positioned following a logic similar to the previous stage.

Lower deck partition The problem is very analogous to the one presented in Eq. (13.a) except that the unit type restriction in the AFTbay is eliminated (Eq. 13.d). Here the weight is distributed between both compartments taking into account that in the back compartment the total weight carried in the bulk compartment is taken into account.

$$\sum_{i=1}^N w_i x_i = P_c * C_{RLD} \tag{20.a}$$

$$\sum_{i=1}^N t_i l_i x_i \leq 2l_{LDback} \tag{20.b}$$

$$\sum_{i=1}^N t_i l_i (1 - x_i) \leq 2l_{LDfront} \tag{20.c}$$

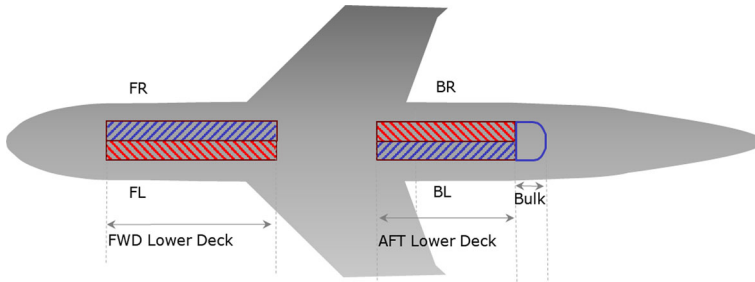


Fig. 6 Positioning lower deck Partitioning the selection of ULDs for the lower deck into two groups, ensuring that both subgroups fit within the physical separation of the lower bay

$$\sum_{i=1}^N (2 - l_i)x_i = 2k \text{ with } k \in \mathbb{N} \tag{20.d}$$

Again, let N be the total number of selected ULDs, and P_c represents the percentage of the total weight designated for the back of the aircraft relative to the actual weight carried (C_{RLD}). The choice of P_c is determined by the ratio of compartment lengths ($11.82(FC)/10.19(BC) = 1.16$, see 1) and the estimated total weight in the bulk compartment, set at 1000 kg.

Equations (20.b) and (20.c) ensure that the volume allocated in both compartments fits within the length of each compartment. Similar to the previous formulation (13.a), equation (20.d) mandates an even number of type 1 ULDs in the back group.

Lower deck positioning The positioning strategy is identical to the main deck, following the same heuristics (see Fig. 6). Denser ULDs aim to be placed close to the center of the plane, followed by a successive first-order rearrangement (moving consecutive positions) to enhance the center of gravity of this section.

3.3 Global condition check of both decks

For every main deck and lower deck selection, we create 10 distinct sorting arrangements through 10 potential partitions. Due to potential conflicts in meeting various constraints during partitioning, one arrangement might excel in weight distribution but struggle to fully comply with volume constraints, especially critical in the lower deck. To address this, we iterate the algorithm 10 times for each deck.

From the 10 sortings for each deck, we choose the combination that optimizes the center of gravity while adhering to shear constraints. The selection criterion focuses on achieving the best possible center of gravity. To determine this, we compute all conceivable combinations of main deck and lower deck arrangements classically, a swift process since there are at most 100 combinations (10 for the upper deck times 10 for the lower deck).

The shear force restriction ensures the cargo configuration stays within the aircraft’s structural capacity, preventing excessive stress. We represent the airframe as two cantilevered beams, supported at a point between the wings. While initially placed at the

center of the main deck, this point's position is adjustable in our formulation. This model is based on the shear restriction formulation from the original Airbus challenge. For calculation purposes, we assume ULDs have a uniform weight distribution on their support surface. Refer to Appendix B for the complete calculation formulation.

4 Results

The results below show that the presented hybrid architecture works on various backends and successfully achieves the proposed objectives.

The proposed solution underwent testing through the generation of random instances simulating various loading scenarios. In most cases, 40 unit load devices (ULDs) were selected for both the main and lower decks. Each ULD generated was randomly assigned one of the possible formats available for the main deck and lower deck which can be seen in Tables 2 and 3 under the column "Name." Additionally, each ULD received a profit randomly distributed between 500 and 5000, and a weight distributed between 500–4000 kg for the main deck and 300–2000 kg for the lower deck. These chosen values align with real-world ULD data, ensuring representative test cases.

For each instance of the problem, the expected outcome of the obtained solution is a load configuration for both the main deck and the lower deck. This includes the ULDs that were loaded, along with their positions within the aircraft, the selected profit, total weight, as well as the distance to the target center of gravity and the shear curve. The center of gravity was chosen as the beam position from which shear is calculated, but this can be easily modified. In the absence of better data, a deviation of 10% from the total length of the aircraft fuselage was deemed acceptable, as proposed in the Airbus challenge [5].

The selected backends were:

- Classical solvers
 - *Classic solver* The classical solver employed was Gurobi Optimizer [20] version 10.0.3 build v10.0.3rc0 (win64), running on a PC equipped with a 12th Gen Intel(R) Core(TM) i7-1255U CPU. The system features 10 physical cores, 12 logical processors, and utilizes up to 12 threads for computation.
 - *Classic solver for annealing simulation* The SimulatedAnnealingSampler [21] provided by D-Wave was used to simulate the implementation of a thermal annealing process. This simulator runs locally and was used on the same personal laptop as for Gurobi.
- Solvers that utilize a QPU (quantum processing unit)
 - *Quantum-classical hybrid solver* D-Wave provides a solver that uses both classical and quantum resources (CPU and QPU). For the hybrid solver configuration, we utilized the example code provided in [22] on page 3, making slight modifications to specify the number of shots and the topology type.
 - *Quantum annealing* For the QPU-based solution, we used the D-Wave advantage_system6.3 quantum computer, with 5614 qubits. We manually configured

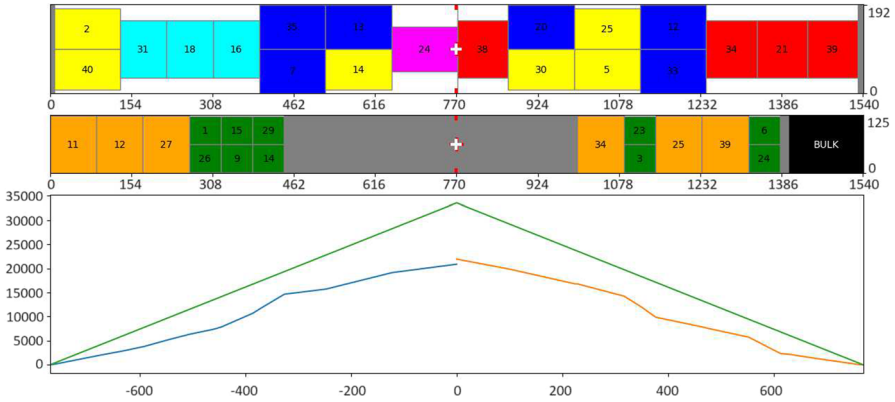


Fig. 7 Images **a** and **b** represent the final load configuration of the main deck and lower deck, respectively. The units of the aircraft are in inches. The two red lines indicate the desired center of gravity, while the white crosses represent the center of gravity of each deck separately, and the red cross represents the center of gravity of the entire aircraft. Image **c**, the total shear curve of the configuration is depicted. The horizontal axis represents inches along the aircraft, considering 0 as the beam position. The vertical axis represents the shear force and is measured in kg/Inch

the following parameters: the number of shots and ‘reduce_intersample_correlation’ introducing a delay between reads to mitigate sample-to-sample correlation. Explanations for all parameters QPU and hybrid solvers parameters can be found in the D-Wave documentation [23].

We accessed both the hybrid and the QPU solvers using the sapi interface [21].

In our evaluation of classical solvers, we tested 100 instances of the problem on the two classical backends. In all cases, valid results adhering to established constraints were achieved. Notably, Gurobi consistently produced the best solutions, optimizing both profit and proximity to the desired center of gravity. This will be shown below in this section. In Fig. 7, the best result obtained is presented.

Figure 8 compares the profit obtained with all the instances for each solver. Note that maximizing profit is the objective of the problem. The profit values obtained are in the range of 30,000 to 140,000. As can be seen, the Gurobi solver (in red) achieves better results. The second-best results were obtained with the Hybrid solver (in blue). Results from the simulated annealer (in green) are in third place. The worst results were obtained using the D-Wave advantage system QPU solver (in orange). We shall highlight that for every instance, the best profit was obtained using the Gurobi solver.

In comparing both classical solvers, Gurobi significantly outperforms the thermal annealing simulator provided by D-Wave. The comparison of the 100 aggregated runs is shown in Fig. 9 using relevant indicators. The figure includes the results for profit, center of gravity (CoG), volume loaded, weight loaded, and shear indicator for all the classical and quantum solvers tested. Superior solutions were consistently obtained with the Gurobi solver. The classical solvers obtained valid results for the 100 instances, while the quantum solvers failed in the majority of the cases. The hybrid solver obtained 29 valid results and the QPU 34 valid solutions. It is worth noting

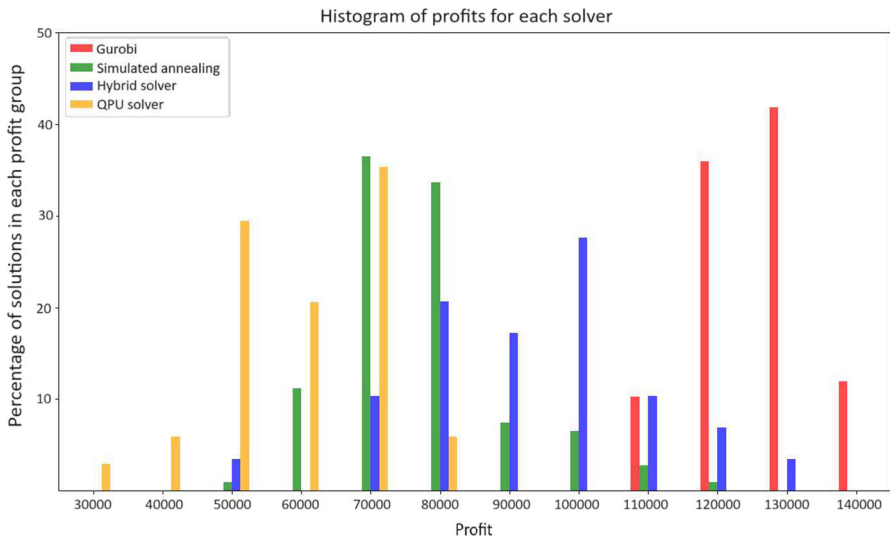


Fig. 8 Profit comparison between solutions obtained using Gurobi (red), D-Wave simulated annealing (green), D-Wave hybrid solver (blue), D-Wave advantage system (QPU) (orange). The histogram shows the distribution of the profit values obtained for all the instances using the four solvers

that a better selection of ULDs, optimizing space availability and aircraft weight, will always bring the shear closer to its limit.

The results for the solvers using QPU are less promising. In both cases, fewer than 10% of the instances resulted in valid solutions. These solutions are notably suboptimal as they underutilize load capacity, leading to suboptimal profit. However, it is worth mentioning that despite this suboptimal choice, the algorithm still achieves a good center of gravity. Figure 10 illustrates the best solution obtained using QPU.

5 Discussion

The proposed approach offers several advantages, contributing to a more efficient problem-solving process.

Working in series, separating the computation into different steps, has a profound impact by significantly reducing the complexity of variables involved in solving the problem. This streamlined approach allows to focus on the variables specific to each task, enhancing the overall efficiency of the solution process.

Furthermore, the approach demonstrates scalability, making it suitable for addressing larger problems. Whether dealing with a greater number of ULDs or aircraft with larger capacities, the series-based methodology accommodates a broader range of scenarios, expanding its practical applicability.

However, the approach comes with certain drawbacks. One notable limitation is the decoupled nature of the center of gravity. As it is not a global objective of the problem,

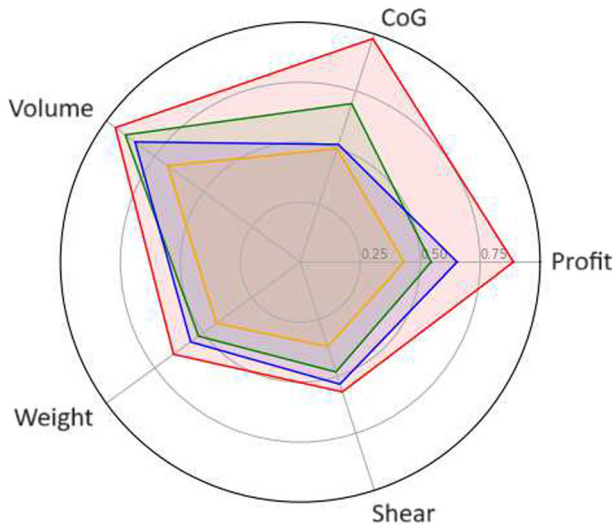


Fig. 9 Comparison between solutions obtained using Gurobi (red), D-Wave simulated annealing (green), D-Wave hybrid solver (blue), D-Wave advantage system (QPU) (orange). *Profit* Represents the profit of the solution, expressed as a fraction of an arbitrary number, in this case, 150,000. *Center of gravity (CoG)* A value of 1 on the diagram indicates a deviation of 0 inches from the objective CoG. Conversely, a value of 0 implies a deviation of 150 inches or more (150 inches is the 10% of the total length of the aircraft fuselage). *Volume* Denotes the loaded volume as a fraction of the total volume allowed. *Weight* Represents the loaded weight as a fraction of the total weight allowed. *Shear* This parameter measures the mechanical stress on the airplane, with a higher numerical value indicating higher mechanical stress. The better solutions are those in which Profit is higher. In every other dimension of this chart, a higher value—not exceeding the maximum allowed—is a desirable feature

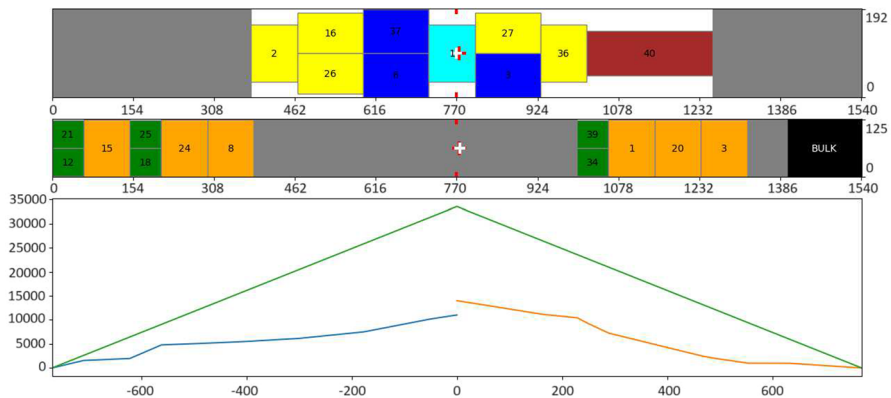


Fig. 10 The three images correspond to the same explanation provided in Fig. 7. The units of the aircraft are in inches and the shear force and is measured in kg/inch

its optimization is contingent on the prior choice of ULDs, potentially limiting its overall effectiveness.

Additionally, the possibility of certain ULD configurations not adhering to the center of gravity or shear constraints introduces a need for iterative solutions. In such cases, where a different selection is necessary, running the sorting part of the problem more than once may be required. This introduces an added layer of complexity and potential iterations to achieve a compliant solution.

Regarding the results obtained in different backends, we can conclude that the model works very well when the backends are classical. In the case of those using QPU, it is clear that they present more complexities to consider. The container selection problem is highly demanding for the real QPU. The embedding necessary to accommodate logical qubits within the real topology of the QPU—in our case, the advantage system 6.3 with the Pegasus graph [23] topology—requires each variable representing a ULD to be connected to at least all other variables representing the remaining ULDs and the corresponding slack variables. Each variable can be viewed as a node in this graph. This implies that every node of the graph shall have a degree at least of the number of containers. This results in very long chains of interconnected qubits and this tends to break, resulting in invalid solutions.

Furthermore, the selection of constants for each constraint plays a crucial role in the pursuit of good solutions. In the case of classical solvers, as long as the constants are chosen in a way that the penalty for violating a constraint exceeds the profit gain obtained from violating that constraint, the model will perform well. When these models are run on a QPU, these constants also impact the energy difference between different solution levels, making their influence on QPU performance significant. Simply imposing high penalties does not resolve the problem, as these penalties compete against additional constraints imposed by the embedding process into the Pegasus graph in which weights are placed on the chains constructed.

A future task is to find a way to optimize these constants so that the comparison against classical solvers is genuinely fair.

6 Conclusions and future work

Conclusions

In conclusion, this study presents an advancement in the field of aircraft load optimization using quantum computing methods. Our research focused on developing a sophisticated model that employs quantum unconstrained binary optimization (QUBO) for use with quantum annealers, offering a novel perspective in tackling the complexities of the problem of aircraft load optimization. We successfully ran this new technique in different backends, both quantum and classical.

Our results indicate that the classical solver Gurobi currently provides superior solutions (although comparable) to quantum ones like D-Wave hybrid solver and D-Wave advantage. Although this might seem counterintuitive, it is important to consider that the topology of the D-Wave annealer used during the tests (Pegasus) allows each

variable to connect with at most 15 others. This necessitates using multiple physical qubits to simulate larger connections, significantly limiting the maximum number of variables encoded in the annealer and resulting in long chains of qubits that are highly fragile. If these chains break, they can lead to invalid solutions. Quantum annealers are a recent technology and are continuously improving in terms of noise resilience (chain breaks), better topologies, and improved embedding methods. The embedding method used during these tests was provided by D-Wave.

Another interesting finding is that the D-Wave hybrid solver (quantum backends) outperformed simulated annealing in many cases. These classical backends provided by D-Wave are specifically designed to solve problems modeled by QUBO. In terms of profit optimization and adherence to constraints, quantum computing methods, though currently less efficient, hold significant potential for future advancements.

This paper not only demonstrates the feasibility of this innovative approach but also sheds light on the current limitations and future potential of quantum computing in real-world applications. It is important to note that as quantum annealers continue to evolve, they may provide better solutions to larger problems or in shorter times. Running a real-world problem on these devices allows us to observe that, at the very least, they match current technologies with great potential for growth.

The future directions of this research involve a deeper dive into refining the constants used in the penalization terms in the formulation for the quantum annealers. We plan to continue testing new quantum annealers as they become available, with a higher number of qubits and higher qubit interconnectivity. Additionally, extending the application of our model to a broader range of aircraft types and operational scenarios could provide more comprehensive insights. Finally, the integration of real-time data and the development of automated systems for decision making would also be important in bringing these theoretical models into practical, real-world applications.

Appendix A ULDs compatible with the model

Given the diverse fuselage radii observed in different cargo aircraft models, despite the standardization of container dimensions and pallet base size, a notable variety of ULDs exists, setting them apart from sea cargo containers. The A330-200F model distinguishes itself by accommodating nearly all existing ULD types due to its modern design. In Appendix A, Tables 2 and 3 enumerate the selected containers and pallets for this model, chosen based on their widespread usage.

Table 2 Selection of ULDs used for main deck

Kind	Name	With	Height ^a	Length	Position
Pallet	M2	238.5	96	96	Single row
Pallet	PMC/PAG	125	96	96	Side by side
Pallet	PMC/PAG	125	96	88	Side by side
Pallet	PMC/PAG	96	96	125	Single row
Pallet	P9A	88	96	125	Single row
Container	AMV	125	96	96	Side by side
Container	AMA	125	96	125	Single row

Container names may vary from airline to airline due to the use of different standardizations. All measurements are in inches

^aThe maximum height measurement must be adaptable to the side-by-side position, with one side’s height lower to fit in the aircraft fuselage

Table 3 Selection of ULDs used for lower deck

Kind	Name	With	Height	Length	Position
Pallet	LD7	125	64	88	Single row
Pallet	LD7Wings	125/160 ^a	64	88	Single row
Container	LD3	61.5/79	64	60	Side by side
Container	LD6	125/160	64	60	Single row
Container	LD9	125	64	88	Single row

Container names may vary from airline to airline due to the use of different standardizations. All measurements are in inches

^aCorrespond to the lower and main width, respectively

Appendix B Shear force calculations

Shear force is caused by the weight of the ULDs and is calculated assuming the aircraft is static and level. The lower decks are modeled to be continuously supported by the same beams, which means that the load distribution is simply summed to the main deck. ULDs are modeled as uniformly distributed loads along their length in the direction of the beam, as shown in Fig. 4.

The discrete model calculates the following equations:

For the front of the wings:

$$Q(x) = \int_{a_0}^{\text{supp}} q(x) dx$$

where a_0 is the point on the main deck closest to the cabin, x goes from a_0 to the point of support, and $q(x)$ is the linear load density.

For the cargo loaded to the back of the wings:

$$Q(x') = \int_{a_1}^{\text{supp}} q(x') dx'$$

where a_1 is the point on the main deck furthest from the cabin, x' goes from a_1 to the point of support, and $q(x')$ is the linear load density.

The integrals are calculated as a Riemann sum with partitions of one inch over the length of the beams:

$$Q(x_i) = \sum_{i=0}^{i_{\text{supp}}} q(x_i) \cdot 1 \text{ in}$$

for the front beam, and

$$Q(x'_i) = \sum_{i=0}^{i_{\text{supp}}} q(x'_i) \cdot 1 \text{ in}$$

for the rear beam.

When ULDs are uniformly distributed loads, the sum gives exact results and will continue to do so if the load distribution is any step function with steps in whole inches.

Author Contributions All the authors contributed equally to this work.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. IATA-McKinsey study shows imbalanced aviation value chain. <https://www.iata.org/en/pressroom/2022-releases/2022-12-06-02/>. Accessed 05 June 2023
2. Airlines cut losses in 2022; return to profit in 2023. <https://www.iata.org/en/pressroom/2022-releases/2022-12-06-01/>. Accessed 05 June 2023
3. Passenger demand recovery continued in December 2022 & for the full year. <https://www.iata.org/en/pressroom/2023-releases/2023-02-06-02/>. Accessed 05 June 2023
4. Air cargo demand decline slows in April. <https://www.iata.org/en/pressroom/2023-releases/2023-05-31-01/>. Accessed 05 June 2023
5. Airbus: airbus quantum computing challenge—problem statement n°5 - aircraft loading optimisation. Airbus quantum computing challenge problem statement n°5 aircraft loading optimisation. Accessed 22 July 2021
6. Preskill, J.: Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
7. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. *Phys. Rev. E* **58**(5), 5355 (1998)
8. Kato, T.: On the adiabatic theorem of quantum mechanics. *J. Phys. Soc. Jpn.* **5**(6), 435–439 (1950)
9. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum computation by adiabatic evolution. arXiv preprint [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106) (2000)
10. Bushnell, D.M.: Survey of quantum technologies in aerospace (2023). IEEE
11. Rosch-Grace, D., Straub, J.: Analysis of the potential benefits from using quantum computing for aerospace applications. In: 2022 IEEE Aerospace Conference (AERO), pp. 1–6 (2022). IEEE

12. Stollenwerk, T., O’Gorman, B., Venturelli, D., Mandra, S., Rodionova, O., Ng, H., Sridhar, B., Riefel, E.G., Biswas, R.: Quantum annealing applied to de-conflicting optimal trajectories for air traffic management. *IEEE Trans. Intell. Transp. Syst.* **21**(1), 285–297 (2019)
13. Makhanov, H., Setia, K., Liu, J., Gomez-Gonzalez, V., Jenaro-Rabadan, G.: Quantum computing applications for flight trajectory optimization. arXiv preprint [arXiv:2304.14445](https://arxiv.org/abs/2304.14445) (2023)
14. Pilon, G., Gugole, N., Massarenti, N.: Aircraft loading optimization—QUBO models under multiple constraints. arXiv preprint [arXiv:2102.09621](https://arxiv.org/abs/2102.09621) (2021)
15. Çela, E., Punnen, A.P.: Complexity and polynomially solvable special cases of QUBO. In: *The Quadratic Unconstrained Binary Optimization Problem: Theory, Algorithms, and Applications*, pp. 57–95. Springer, Cham (2022)
16. Lucas, A.: Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014)
17. Coffey, M.W.: Adiabatic quantum computing solution of the Knapsack problem. arXiv preprint [arXiv:1701.05584](https://arxiv.org/abs/1701.05584) (2017)
18. Karp, R.M.: *Reducibility among combinatorial problems*. Springer, Cham (2010)
19. D-Wave systems: welcome to D-Wave system documentation. https://docs.dwavesys.com/docs/latest/c_gs_1.html. Accessed 13 Oct 2023
20. Gurobi optimization, LLC: Gurobi optimizer reference manual (2023). <https://www.gurobi.com>
21. D-Wave systems Inc.: D-Wave hybrid release 0.6.10. https://docs.ocean.dwavesys.com/_downloads/hybrid/en/latest/pdf/ (2023)
22. Configuring access to Leap’s solvers—ocean documentation 6.7.1. Accessed 7 Dec 2023
23. D-Wave systems Inc.: QPU-specific physical properties: advantage-system6.3 user manual. (2023)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.