A Tabu Search Algorithm for the Optimization of the Long Term Parking of Aircraft

O'Callaghan De Jonghe, V.E.; Roling, P.C.

DOI
10.2514/6.2022-3219

Publication date
2022

Document Version
Final published version

Published in
AIAA AVIATION 2022 Forum

Citation (APA)
O'Callaghan De Jonghe, V. E., & Roling, P. C. (2022). A Tabu Search Algorithm for the Optimization of the Long Term Parking of Aircraft. In *AIAA AVIATION 2022 Forum: June 27-July 1, 2022, Chicago, IL & Virtual* Article AIAA 2022-3219 (AIAA AVIATION 2022 Forum). American Institute of Aeronautics and Astronautics Inc. (AIAA). https://doi.org/10.2514/6.2022-3219

AIAA AVIATION Forum
June 27-July 1, 2022, Chicago, IL & Virtual
AIAA AVIATION 2022 Forum

10.2514/6.2022-3219

A Tabu Search Algorithm for the Optimization of the Long Term Parking of Aircraft

Victor O'Callaghan[1] and Paul Roling[2]
Delft University of Technology, Delft, 2629 HS, The Netherlands

The 2020 coronavirus pandemic lead to a virtual standstill of air passenger traffic in the spring of that same year. While some travel restrictions have since been lifted, passenger air travel is not expected to return to pre-coronavirus levels for several years. Then the question arises of how to park the large amounts of grounded aircraft efficiently, minimizing valuable airport space used. While aircraft parking for this purpose is a largely unexplored area in academic literature, the problem shows similarities with cutting and packing problems which have been researched for many years. Hence, the proposed model in the paper is modelled similar to that of the irregular strip packing model, where a fixed width is used and the length of the parking layout is to be minimized. Aircraft are represented as non-convex polygons and are allowed to rotate in discrete intervals. The concept of the no-fit polygon (NFP) is used in order to prevent overlap between aircraft. A tabu search algorithm with an adaptive tabu list is proposed in order to optimize the sequence and orientations in which the aircraft are placed onto the placement area using a bottom-left (BL) placement strategy. In order to evaluate the effectiveness of the proposed algorithm, several instances are created and tested using computational experiments.

I. Introduction

During the 2020 global crisis due to the outbreak of the COVID-19 coronavirus, governments throughout the world imposed lock downs or stay at home orders, banning all non-essential travel. As a result, air travel demand decreased rapidly and airlines had to significantly reduce capacity, with some airlines even grounding their entire fleet for several weeks. In April 2020 the amount of traffic (in terms of revenue passenger kilometers) decreased by 94% and capacity (in terms of available seat kilometers) decreased by 87% compared to the same period in the previous year[1]. In order to accommodate all the grounded aircraft, airlines and airports creatively parked the aircraft on runways, taxiways, and/or unused gates and stands. Although travel restrictions are gradually being lifted, it is expected that air travel demand will not return to pre-coronavirus levels for several years[2].Consequently, aircraft will need to remain parked for quite some time and the challenge for airlines and airports then becomes to how to park these aircraft efficiently.

During normal airport operations aircraft are parked at fixed airport infrastructure such as gates and stands, where large margins are necessary to allow for aircraft servicing (such as the passenger boarding bridge, cargo/luggage loading, galley service trucks, etc.). However, for the purpose of long term parking, such margins are not needed and aircraft can be parked anywhere on a given surface at any arbitrary orientation and close to each other. Aircraft could even be parked in nonconventional ways, with their wings overlapping.

Although the gate/stand allocation problem has been researched extensively[3], it is not relevant for the reasons mentioned earlier. Limited research has been carried out recently on the somewhat related problem of optimizing the parking of aircraft within a maintenance hangar[4,5], however the problem size is naturally limited due to limited space available within a hangar. Additionally, due to maneuverability constraints within a hangar, aircraft are usually parked tail or nose first into the hangar and are thus limited to a maximum of two orientations. Due to these limitations, such problems do not relate well to the long term parking of aircraft. Another similar but more general problem of generating tight layouts is found in the field of cutting and packing problems, where e.g. shapes must be cut from a sheet of metal or items must be packed close together in order to minimize wasted material or space. When irregular shapes such as aircraft are involved, this problem is also referred to as the nesting problem. While many papers have been published on cutting and packing of regular shapes without or only a limited amount of possible orientations, the

---

[1] MSc student.

[2] Lecturer / researcher. AIAA member

nesting problem that allows many or even continuous rotations in combination with large problem instances has not been researched extensively. The nesting problem is known to be NP-hard[6] and therefore solutions approaches are mainly based on 5 heuristic methods in order to obtain good solutions in a timely manner.

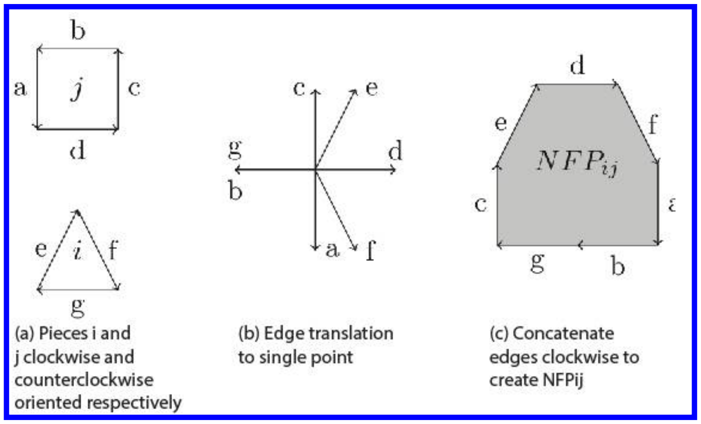
As airports usually have long rectangular paved areas such as runways, taxiways and remote stands where aircraft could be parked, the objective in this paper is to minimize the total length of such a rectangular area with fixed width for a given set of aircraft, which is similar to the irregular strip packing problem. A tabu search algorithm for optimizing the long term parking layout of aircraft is proposed, using a bottom-left placement strategy based on the no-fit polygon (NFP). Aircraft are allowed to rotate in discrete intervals, although more orientations are considered than only the 2 or 4 orientations commonly used for the nesting problem in literature. While this paper focuses on the long term parking of aircraft (on outdoor surfaces), the results could possibly also be beneficial for the purpose of maintenance hangar space optimization or in the broader scope for nesting problems in general.

II. Methodology

A common strategy for solving similar problems, such as cutting and packing problems, is to search over the sequence (and rotations) of items to be placed. This sequence with the corresponding orientations is then decoded through a placement algorithm to obtain an actual layout, and new sequences with corresponding (new) orientations are found through a heuristic algorithm. The advantage of such an approach is that layout is guaranteed to be feasible because the placement rules of the placement algorithm ensure items are placed at a feasible location, i.e. within the placement area and with no overlap. In contrast, solution approaches searching over the actual physical layout allow for items to overlap during the search and hence the algorithm could potentially get stuck in resolving overlap, resulting in an infeasible solution[7]. For that reason, the algorithm proposed in this paper follows the former approach as the general solution strategy.

A. No-fit Polygon (NFP)

The no-fit polygon (NFP) is a geometric tool commonly used for similar problems such as the cutting and packing problems, and has also been applied to aircraft parking optimization within a maintenance hangar. The boundary of the NFP represents the relative position of the reference points of the two items where the two items touch each other. If the reference point of the item placed within the NFP, the two items overlap; if it is placed outside the NFP no overlap is present (and the two items do not touch). The NFP can be understood as sliding an item (i.e. the orbiting piece) around the other item (i.e. the fixed piece) and tracing its reference point, where the resulting line is the NFP.



(a) Pieces i and j clockwise and counterclockwise oriented respectively

(b) Edge translation to single point

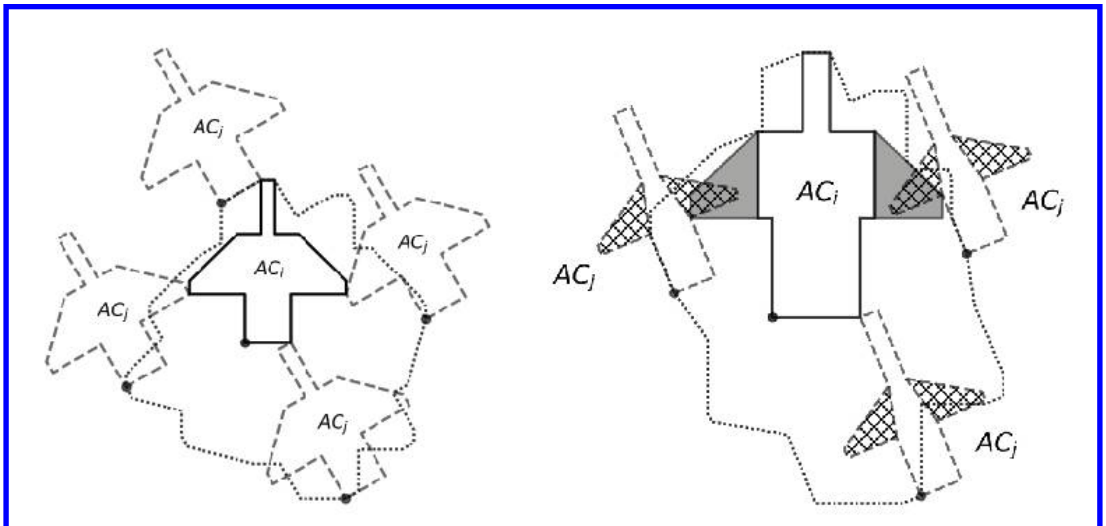
(c) Concatenate edges clockwise to create NFPij

Figure 1: No-fit polygon[8,9]

The method used to construct the NFP between two items used in this paper is based on the approach originally proposed by Cuninghame-Green[8] for the case of two convex polygons and works as follows. First, the edges of polygons i and j are oriented in a clockwise and counterclockwise manner respectively, where i is the fixed polygon and j is orbiting polygon (Figure 1a). Next, the edges are translated to start at a single point (Figure 1b). In the final step the translated edges from Figure 1b are linked together in a clockwise manner, resulting in the NFPij (Figure 1c).

Although the aircraft representation introduced in section 3 is not a convex polygon, it can easily be decomposed into several smaller, convex sections (namely a nose section, mid/wing section, and tail section). The NFP is then generated for each of the sections individually, and by recombining the separate section NFPs, the full NFP for two aircraft can be obtained. An example of the NFP between two aircraft is shown in Figure 2a, where AC_i is the fixed polygon and AC_j is the orbiting polygon.

In instances where the wing height difference between aircraft types allows for overlap, the NFP is modified. The wing of the higher wing aircraft can extend over the lower winged aircraft, but cannot protrude its fuselage. In addition, the position of the engines of the higher wing aircraft should be considered such that the lower wing aircraft's wing is not protruding the engines. Therefore, in order to take into account those two conditions, two NFPs are generated: one for the full lower wing aircraft and the higher wing aircraft where the outer wing is cut-off after the (outer) engines, and a second for the lower wing aircraft's fuselage (and including its tail) and the full higher wing aircraft. The two resulting NFPs are combined to obtain the complete modified NFP and an example is shown in Figure 2b. The hatched area of the wing of AC_j can be placed underneath the part of the wing of AC_i extending beyond its engines, marked by the grey area.



(a) Example of the NFP (dotted line) for aircraft AC_i and aircraft AC_j. Aircraft AC_j is plotted multiple times at various arbitrary locations to show its relative position with respect to AC_i.

(b) Example of the modified NFP (dotted line) for aircraft AC_i and aircraft AC_j, where the wing of AC_j can be placed underneath the wing of AC_i.

Figure 2: No-fit polygon examples for aircraft

In reality, aircraft are not placed touching each other and therefore a safety margin is desired between aircraft. A safety margin can be imposed by extending the boundary of the aircraft with the desired margin (Figure 3a). By definition of the NFP, when the reference point of the orbiting aircraft ACj is placed on the boundary of the NFP, the two aircraft touch. Hence moving the edges of the NFP outwards is equivalent to adding a safety margin around the aircraft (blue region in Figure 3b). Therefore, in order to add a safety margin sf between aircraft, the edges of the NFP are moved outwards by distance sf resulting in the revised NFP (Figure 3c).

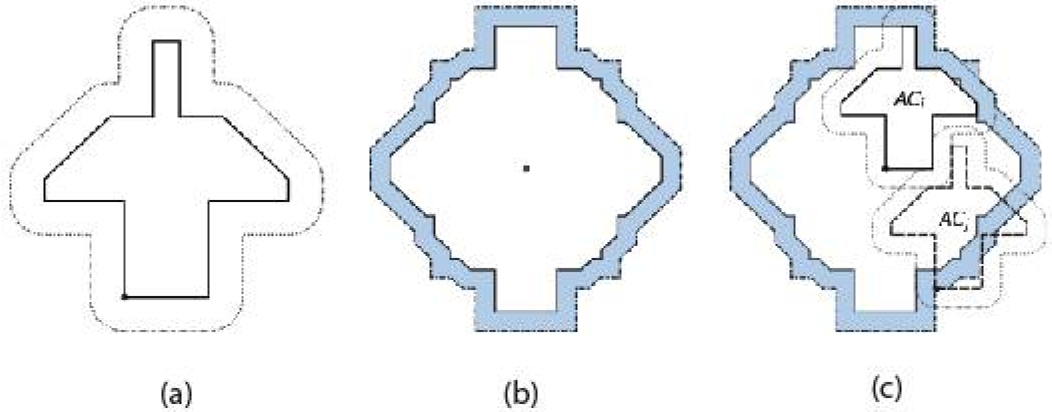


Figure 3: Safety margin around aircraft (a) and revised NFP (b,c)

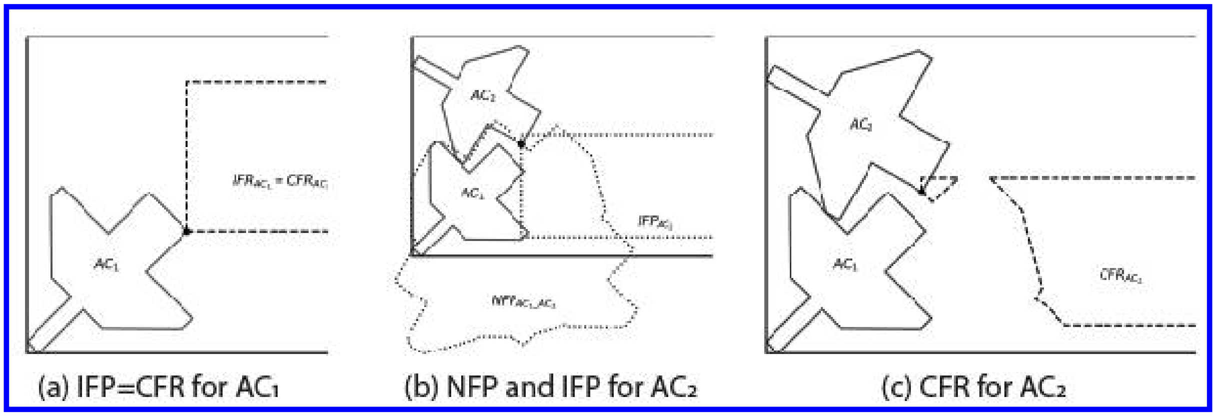
Using concept similar to the NFP, the inner-fit polygon, or IFP, can be created. Instead of sliding around a fixed item as is done with the NFP, the item slides within the other. The NFP is used to verify an item remains within the placement area. Since the wing is allowed to extend over non-load bearing surfaces, one IFP is generated for the landing gear and the load bearing placement area, and a second IFP is generated for the entire aircraft and the overhang area. The final IFP is then the limiting side of each individual IFP, such that the landing gear and the entire aircraft remain within their respective boundaries.

The NFPs and IFPs are generated in a preprocessing phase, where for each aircraft type and for each orientation combination a NFP (IFP) is generated and stored in the computer's memory. The placement algorithm is then able to retrieve the NFPs (IFPs) during the placement process in the optimization phase.

B. Placement Algorithm

In order to place a sequence of aircraft with their corresponding orientations onto the placement area, the single pass bottom-left (BL) placement algorithm is used. While originally proposed by Art[10], it has become a widely used placement strategy and is still relevant today[11,12]. In the context of the current problem of parking aircraft, illustrated in Figure 1, the BL algorithm is equivalent to placing an aircraft as far to the left as possible (i.e. lowest feasible x-coordinate) and in case there are multiple such positions possible, the aircraft is placed at the bottom most position (i.e. of the positions with lowest x-coordinate, the position with the lowest y-coordinate is chosen).

As the placement area is a continuous space, the NFPs and inner-fit polygons (IFP) are used to obtain a feasible placement area for an aircraft, also known as the collision free region (CFR), from which a set of placement points can be derived and the bottom-left position is then easily selected from those points. While the CFR still is a continuous region, when looking to place an item at its BL position on the placement area, only the vertices of the boundary of the CFR can be considered (Gomes and Oliveira, 2002). The CFR for the current aircraft to be placed is obtained by subtracting the NFPs for each aircraft already placed from the IFP of the current aircraft.



(a) IFP=CFR for AC_1 (b) NFP and IFP for AC_2 (c) CFR for AC_2

Figure 4: Placement process

The process of placing an aircraft sequence and their corresponding orientations is illustrated in Figure 4. For the first piece, only the corresponding IFP is retrieved, since no other aircraft have been placed onto the layout yet. The CFR is then simply equal to the IFP and hence the aircraft is placed at the BL coordinate of the IFP (Figure 4a). For the second aircraft to be placed, in addition to its IFP, the NFP between the aircraft already placed and the aircraft to be placed is retrieved (Figure 4b). By subtracting the NFP from the IFP, the CFR is obtained from which the BL coordinate of its vertices is selected to place the aircraft (Figure 4c). This process is repeated until all aircraft for a given sequence are placed onto the placement area. Note that this approach can place smaller aircraft into empty spaces in the partial layout when placing a sequence, as can be seen in Figure 4c, which otherwise would not be filled using e.g. a sliding or translate approach.

Tabu Search

In order to search over the sequence and aircraft orientations, the tabu search (TS) metaheuristic is used. A tabu search includes a local search procedure which aims to improve the current solution. A key feature of the tabu search is that when no improving moves can be found (i.e. a local optimum is reached), the search is allowed to continue by accepting non-improving moves and is therefore able to escape local optima. To avoid immediately circling back to the same local optimum, a tabu list temporarily stores recent moves. Moves on the tabu list are not allowed, unless that move would result in a better solution than the best (global) solution found in any of the previous iterations (also known as the aspiration criterion).

For the tabu search the aircraft sequence and orientations are encoded as a binary string. The first part of the bit string represents the sequence, and the second part represents the orientation for each aircraft. The sequence part of the bit string, bits are used to point to each aircraft in their initial sorting, which is the sequence in which air-craft are sorted by non-increasing area and all bits are set to zero. For the orientations part of the bit string the possible orientations for each aircraft is given by the variable orient bit size resulting in steps of 180 degrees divided by the number of bits.

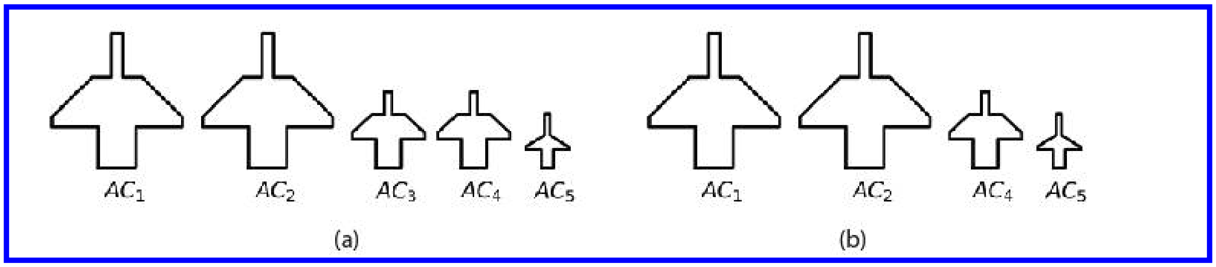


Figure 5: Aircraft sequence

The concept is best illustrated using an example. Consider the case for five aircraft, which are sorted by non-increasing area as described above and are given the following numbering: AC_1, AC_2, AC_3, AC_4, AC_5 (Figure 5a). Consider the sequence bitstring 010 11 10 1; in order to point to any of the 5 aircraft in the initial list, 3 bits are needed ($\log_2(5)$, rounded up). Hence, in this case the first three bits in the bitstring (010) are used to select the first aircraft to be placed. The binary number 010 represents the number 2 ($0*2^2+1*2^1+0*2^0$), add 1 to account for bits numbering starting at zero, and thus points to the third aircraft in the initial list, AC_3. Aircraft AC_3 is therefore selected to be placed first, and the remaining aircraft are AC_1,AC_2,AC_4,AC_5 (Figure 5b). In order to point to any of the 4 aircraft in the remaining list of aircraft (Figure 5b), now only two ($\log_2(4)$) bits are needed. Therefore, the next two bits in the bitstring (11) represent the number 3, add 1 to account for bits numbering starting at zero, and thus point to the fourth aircraft in the remaining aircraft list, AC_5. The process is repeated until all aircraft are placed. The full sequence bitstring 010 11 10 1 for example therefore represents the sequence of aircraft to be placed in the order AC_3,AC_5,AC_4,AC_2,AC_1. Note that in order to select the first aircraft, 3 bits were needed. However, 3 bits gives $2^3 = 8$ possibilities, while there are only 5 aircraft. Bits pointing to an aircraft number greater than the remaining amount of aircraft (5 in this case) would not result in a meaningful aircraft selection (e.g. if the first 3 bits were 111, it would refer to the (nonexistent) eighth aircraft in the initial list). In order to obtain a meaningful result in such cases, the total amount of remaining aircraft is subtracted from the binary number. Hence, 111 would point to the eighth aircraft, subtracting the total remaining aircraft (i.e. 5 in this case), results in the third aircraft AC_3 being selected for placement.

The orientation for each aircraft is described by a certain amount of bits (orient bit size). The amount of bits dictate the amount of discrete orientations possible and their increments. E.g. a 5 bit orientation representation results in $2^5 = 32$ possible orientations in 360/32 = 11.25 degree increments. The total amount of bits needed to encode the orientations of all aircraft is equal to the amount of aircraft times number of bits per orientation.

Using the binary sequence and orientation encoding, the local search method used for the tabu search is done by flipping bits. A move is defined as flipping one bit in the current bitstring, and each iteration each bit in the current bitstring is flipped to obtain new candidate bitstrings. The neighborhood size is therefore automatically dependent on the length of the bitstring, which in turn is dependent on the amount of aircraft to be placed and the bitsize used for representing the orientations. The bitstring corresponding to the layout with the lowest length where none of the bits are tabu is chosen as the new current bitstring. If two bitstrings result in the same objective value, a bitstring is chosen by evaluating the bitstring's binary numerical value and the binary string with the lowest numerical value is selected as the current bitstring.

The tabu list size in an ordinary tabu search is set at a fixed valuebut this might not always yield good results. Therefore a reactive tabu search (RTS) is proposed, based on the method described by Battiti and Tecchiolli[11], which can automatically adapt the tabu list size to the problem and the progression of the search. While a simple adapting tabu list could avoid cycling of the search, it can still become trapped in a region of the solution. Therefore, a diversifying escape mechanism is included when the former mechanism is inadequate.

III.Results

A. Tabu list size

A key element of a tabu search is of course the tabu list. The length of the tabu list (also known as tabu tenure) is usually determined early on through experimental analysis. In order to find an appropriate tabu list length for the aircraft parking problem, several instances with different problem sizes and varying tabu list lengths are tested (while all other parameters remain the same). 4 different instances, with 4, 8, 12, and 16 Airbus A350-900s, are tested for this purpose. A 5 bit representation was used for the orientations, the width of the placement area was W = 60, overhang values of $\delta_{upper} = 20$, $\delta_{lower} = 20$, $\delta_{side} = 10$ were used, and no diversification strategies were applied during the search at this stage. The results are shown in Figure 6, and are expressed as relative percentages in order to compare each problem instance.

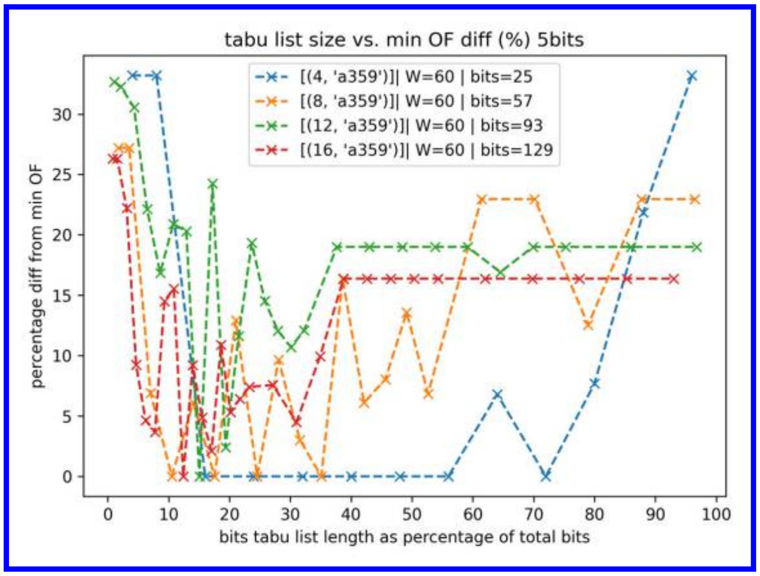


Figure 6: Effect of Tabu list size

As is expected, at the extreme ends of the different tabu list lengths tested the worst results are observed. If the tabu list is too short, the search will simply cycle back and forth between the same solutions. When the list is too long, it is likely the search does not reach an area with good solutions. However, no clear optimum is observed in the middle tabu list length values and there is a high degree of variability between adjacent data points. Most notably, in the instance of 12 A359 aircraft, the length of the final layout increases 25% from its minimum value when the tabu list sizes increases from 16 to 18 (Figure 6). This indicates a hilly solution space which is difficult to navigate and the algorithm is sensitive to its parameter settings. Hence, using a fixed tabu list may result sub optimal results for specific problem instances, while other instances may yield good results. Therefore, the reactive tabu search with a variable tabu list size is used in the remainder of this paper.

Using this strategy, the best solutions found for each problem instance with fixed tabu list length from Figure 6 were matched in initial experimental analysis. As this removes the need to define a fixed tabu list length, it is concluded an adaptive tabu list is likely the better strategy to be adopted for the aircraft parking problem.

B. Orientation bits

The orientation bits define the discrete set of orientations an aircraft is allowed to rotate. More orientations could lead to better solutions, however, more orientations could also increase the complexity of the solution space and lead to longer run times (when not using a fixed time limit) or to search can become trapped. In this subsection, the effect of varying the orientation bit representation is investigated. All other parameters equal, the orientation bits are varied from two bits (i.e. 22 = 4 orientations) up to six bits (i.e. 26 = 64 orientations). Two instances were tested, one consists of 12 Airbus A350-900 aircraft (Figure 12b), the second problem includes a mix of aircraft types namely 3 Airbus 350-900, 4 Airbus A320neo, and 6 Bombardier CRJ200 (Figure 7a).

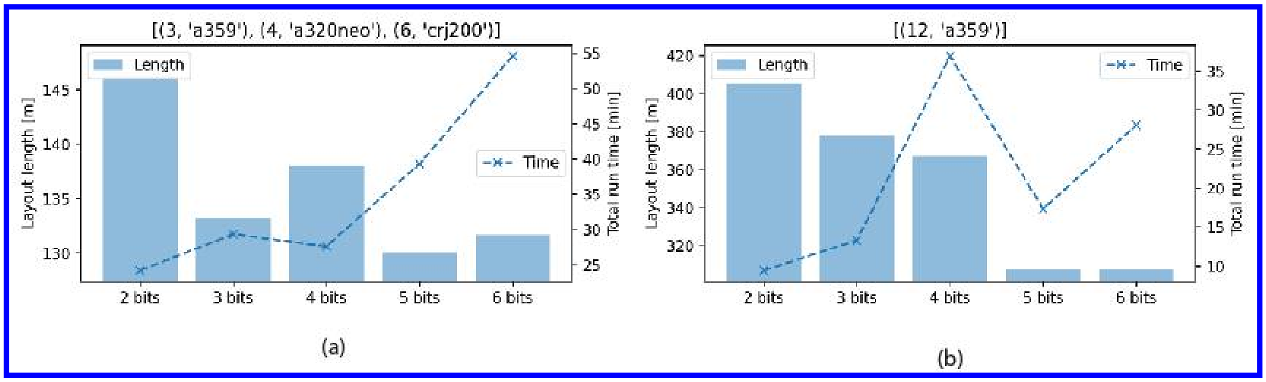


Figure 7: Results of variation in orientation bits

From Figure 12 the general trend is that the layout length decreases as the orientation bits resolution increases, while computational time increases. There are some exceptions though, the run with 4bits for the instance with multiple aircraft types in Figure 7a results in a longer layout compared to 3bit run. Since the computational time is also longer, it appears that the search did not reach a favorable region of the search space when the stopping criterion was met. The 12 A359s instances in Figure 7b show a peak in run time at 4bits. This indicates the search only reached favorable solutions later in the search. The corresponding length however is in line with expectations (i.e. lower than 3 bits, higher than 5 bits).

The layouts resulting from the test runs in Figure 7b are plotted in Figure 8 in order to visually understand how the orientation bit resolution affects the layout. The benefits of using more orientations than typically found in literature (i.e. none or two orientations) is clear, as the worst results are consistently found at lower bits resolutions. It is up to the user to determine the tradeoff between orientation bits resolution and computational complexity.

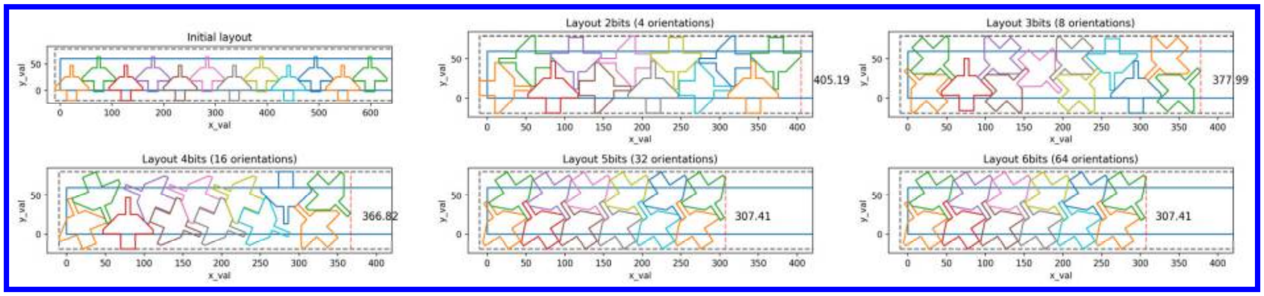


Figure 8: Resulting layout of variation in orientation bits

C. Safety Margin

The safety margin is used in order to maintain a safe distance between aircraft and gives some margin when maneuvering aircraft into their position. In this section it is investigated what the effect on the final layout is when the safety margin is increased. In this instance 2 CRJ200s, 2 A320neos, 2 A359s are considered with safety margins of 0, 1, 3, 5, 10 meters. The resulting layouts are shown in Figure9. As is expected, the length of the resulting layouts increase as the safety margin is increased.

D. Initial Solutions

A tabu search can be sensitive to the initial solution used for starting the search. In the previous experiments the zero bitstring was used for all problems, in this subsection the effect of using other starting solutions is investigated. The results from selected instances from the previous sections are compared to the results when started from an initial solution generated by the greedy best fit heuristic.

It is not immediately clear which initial solution is better. Although the results are within roughly < 1% of each other, the notable exception here is the [12 A359] instance, where the greedy initial solution results in a 10% longer layout. One possible explanation for this behavior when starting from a seemingly promising initial solution is that the search becomes trapped in that region of the solution space (and the diversification strategies are not powerful enough to escape). These results show that the model is indeed sensitive to its input solution.

E. Case study

In this subsection the model is applied to a real-world scenario. Here, 8 Airbus A380-800 (among others) are parked at the Southern California Logistics Airport, near the beautifully named city Victorville, USA (Figure 15). The aircraft can be enclosed by a rectangle measuring approximately 345m x 160m. For the purpose of this analysis, the shorter edge 160m is assumed as the fixed width of the placement area, no overhang is allowed and a safety margin of 5m is chosen in order to resemble similar conditions.

The result from the RTS algorithm can be seen in Figure 11. The length of the layout is 239.97m, which is a reduction of roughly 30% compared to Figure 10.

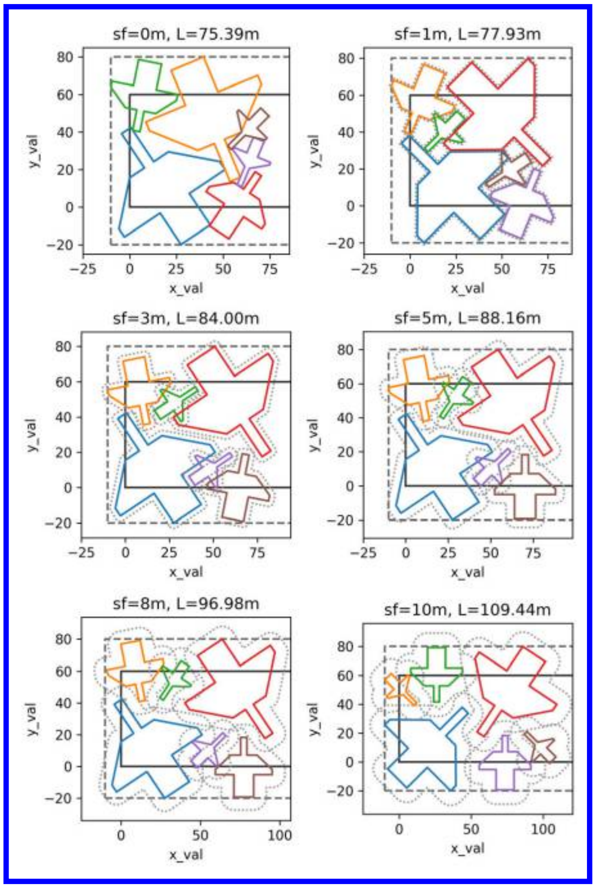


Figure 9: Results for increasing safety margins

Figure 10: A380s parked at the Southern California Logistics Airport (Google, 2021)

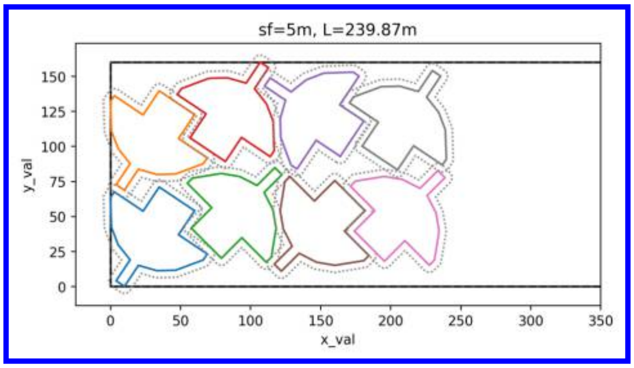


Figure 11: Resulting layout for A380 parking

IV.Conclusions

In this paper a tabu search algorithm was proposed, where the search was over the sequence and orientations of aircraft to be placed using a bottom-left placement strategy. The no-fit polygon was used *as a geometric tool to ensure aircraft do not overlap.* An adaptive tabu list was adopted in order to automatically adjust the tabu size to the problem and progression of the search. The results of computational experiments showed that the algorithm is highly sensitive to input variables and algorithm settings, indicating a hilly solution landscape. Nevertheless, the layouts generated show a tight placement of aircraft and are guaranteed to be feasible. The results show an improvement over greedy single-pass placement heuristics. Furthermore, a case study showed a decrease in layout length compared to the real-world scenario.

As the long term aircraft parking problem is an unexplored research area (expect for the tangential but different problem of aircraft hangar parking optimization), there are many avenues for future research. In the proposed tabu search the search was over the sequence and orientations of aircraft to be placed, using a bottom-left placement heuristic. However, different local search procedures, metaheuristics or matheuristics, and placement strategies could be explored or developed for the purpose of long term aircraft parking. Since the model has been shown to be sensitive to its input and settings, more robust procedures should be investigated. Furthermore, the proposed algorithm assumes a placement area with a fixed width and infinite length. The algorithm could be extended by considering a placement area with fixed dimensions, and selecting which aircraft of a given set should be parked and which aircraft remain unparked to obtain the highest space utilization. Or even multiple different areas with fixed dimensions can be considered (similar to bin packing problems).

From a practical perspective, some layouts generated by the proposed algorithm might not be possible in practice due to aircraft tugs unable to move aircraft into the required position. In addition, maintenance requirements may require aircraft to be moved to avoid tire deformation, or engines to be run periodically such that aircraft cannot be placed in the neighborhood of the exhaust area (and which could require stairs access to the aircraft). Such requirements are not taken into account in the presented algorithm. Within the broader scope of cutting and packing problems, the irregular shape cutting/packing (also known as nesting) problem with many (or continuous) rotations remains a challenging and under-researched topic.

References

[1] IATA (2020). Covid-19 air travel reaching a turning point. https://www.iata.org/en/iatarepository/publications/economic-reports/air-travel-reaching-a-turning-point/. Retrieved: 25 June 2020.

[2] REUTERS (2020). Airline industry braces for lengthy recovery from coronavirus crisis. https://www.reuters.com/article/ushealth-coronavirus-airlines/airlineindustry-braces-for-prolonged-recovery-fromcoronavirus-crisis-idUSKBN21K3KL. Retrieved: 25 June 2020.

[3] Guepeta, J., Acuna-Agost, R., Briant, O., and J.P.Gayon (2015). Exact and heuristic approaches to the airport stand allocation problem. European Journal of Operational Research, 246(2):597–608.

[4] Qin, Y., Chan, F. T. S., Chung, S. H., Qu, T., and Niu, B. (2018). Aircraft parking stand allocation problem with safety consideration for independent hangar maintenance service providers. *Computer and Operations Research*, 91:225–236.

[5] Li, X., Wang, Z. X., Chan, F. T. S., and Chung, S. H. (2019). A genetic algorithm for optimizing space utilization in aircraft hangar shop. *International Transactions in Operations Research*, 26(5):1655–1675.

[6] Fowler, R. J., Paterson, M. S., and Tanimoto, S. L. (1981). Optimal packing and covering in the plane are np-complete. *Information Processing Letters*, 12:133–137.

[7] Ramakrishnan, K., Bennell, J. A., and Omar, M. K. (2008). Solving two dimensional layout optimization problems with irregular shapes by using metaheuristic. In *2008 IEEE International Conference on Industrial Engineering and Engineering Management*, Singapore, pages 178–182.

[8] Cuninghame-Green, R. (1989). Geometry, shoemaking and the milk tray problem. *New Scientist* 12 August 1989 No 1677, pages 50–53.

[9] Cherri, L. H., Mundim, L. R., Andretta, M., Toledo, F. M. B., Oliveira, J. F., and Carravilla, M. A. (2016). Robust mixed-integer linear programming models for the irregular strip packing problem. *European Journal of Operational Research*, 253(3):570–583.

[10] Art, R. C. (1966). An approach to the two dimensional irregular cutting stock problem. (Bachelor's Thesis, Massachusetts Institute of Technology, Cambridge (Massachusetts), United States). Retrieved from https://dspace.mit.edu/bitstream/handle/1721.1/97782/25841973-MIT.pdf.

[11] Battiti, R. and Tecchiolli, G. (1994). The reactive tabu search. ORSA Journal on Computing, 6(2):107–220.