IEEE SMC Systems, Man, and Cybernetics Society Indonesia Chapter

IEEE Universitas Sriwijaya IEEE Student Branch

ICECOS 2024

ICECOS

The 4th International Conference on Electrical Engineering and Computer Science

September 25 –26, 2024

PROCEEDING

ISBN : 979 - 8 - 3503 - 6825 - 3

CoCIS Control and Computational Intelligent System · EMREC · comnets Communication Network and Information Security Research Group · Electrical Energy and Safety Research Group

Table of Content

INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING AND COMPUTER SCIENCE (ICECOS) 2024

Streering Committee

Yanuarsyah Haroen, Institut Teknologi Bandung
Zainuddin Nawawi, Universitas Sriwijaya
Suwarno, Institut Teknologi Bandung
Hussein Ahmad, Universiti Teknologi Malaysia (UTM), Malaysia
Anton Satria Prabuwono, King Abdulaziz University
Muhammad Abu Bakar Sidik, Universitas Sriwijaya

General Chair

Herlina, Universitas Sriwijaya, Indonesia

General Co-chairs

Deris Stiawan, Universitas Sriwijaya, Indonesia
Zolkafle Buntat, Universiti Teknologi Malaysia, Malaysia
Deris Stiawan, Universitas Sriwijaya, Indonesia
Silfani Sandra, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)

Secretary

Syarifa Fitria, Universitas Sriwijaya, Indonesia
M. Fikri Aulian, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)

Publication Chairs

Deris Stiawan, Universitas Sriwijaya, Indonesia
Badar Alpian, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)

Finance Chairs & Treasurer

Suci Dwijayanti, Universitas Sriwijaya, Indonesia
Shalsabila Puteri, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)

Public Relation Chairs

Huda Ubaya, Universitas Sriwijaya, Indonesia
Nurul Afifah, Universitas Sriwijaya, Indonesia

Technical Program Chairs

Bhakti Yudho Suprapto, Universitas Sriwijaya, Indonesia
Mohd. Riduan Ahmad, Universiti Teknikal Malaysia Melaka, Malaysia
Krutthika Hirebasur Krishnappa, Southern University and A&M College, USA
Aditya Afriansyah, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)

International Scientific Committee

Local Chairs

Ahmad Heryanto, Universitas Sriwijaya, Indonesia
M. Qurhanul Rizkie, Universitas Sriwijaya, Indonesia
Adi Hermansyah, Universitas Sriwijaya, Indonesia
Kemahyamto Exsaudi, Universitas Sriwijaya, Indonesia
Ahmad Zarkasi, Universitas Sriwijaya, Indonesia
Pacu Putra, Universitas Sriwijaya, Indonesia
Iman Saladin Azhar, Universitas Sriwijaya, Indonesia
Abdurahman, Universitas Sriwijaya, Indonesia
Muhammad Karisma, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)
Muhammad Akbar, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)
Syanti Prasetyani, Universitas Sriwjaya, Indonesia (IEEE SB Universitas Sriwijaya)
M. Apriyadi Tri Putra, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)
Muhammad Ari Anggara, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)
Muhammad Rizal Mutaqin, Universitas Sriwijaya, Indonesia (IEEE SB Universitas Sriwijaya)

2024 International Conference on Electrical Engineering and Computer Science (ICECOS) took place in Palembang, Indonesia (Hybrid), on September 25-26, 2024.
ISBN: 979-8-3503-6825-3

Greedy Approach for Optimizing Image View Layout on Various Sizes of 2D UI Container

1st Lely Hiryanto
Department of Informatics Engineering
Tarumanagara University
Jakarta, Indonesia
lelyh@fti.untar.ac.id

2nd Andhika Putra Wirawan
Department of Informatics Engineering
Tarumanagara University
Jakarta, Indonesia
andhika.535210010@stu.untar.ac.id

3rd Tony
Department of Informatics Engineering
Tarumanagara University
Jakarta, Indonesia
tony@fti.untar.ac.id

***Abstract*— One important feature of every augmentative and alternative communication (AAC) application is the image view layout (IVL) in a user interface (UI) container, whose size is bound by the device's screen size. The IVL problem deals with arranging a set of images in a two-dimensional UI container with a definitive size. The study seeks to maximize the use of the container space to fit in as many images as possible in such a way that the number of display sets for the overall images is minimum. We consider each image a picture card containing an image or a photograph and a text label with a fixed height but varying width. The IVL problem is a special case of the 2D Bin Packing Problem, which is NP Hard. To provide approximate solutions, we propose two greedy-based heuristic algorithms, which are Order-IVL and Best-Fit-IVL. For the experimental dataset, we use 148 picture cards obtained from an AAC application called VICARA. For 30% of the experimental scenarios, our evaluations show that Order-IVL produces one extra display set than Best-Fit-IVL. Furthermore, Best-Fit-IVL results in the same number of image display sets as the optimal solutions generated by solving the integer linear programming of the IVL problem. In addition, the Best-Fit-IVL consistently produces a tidy and neater arrangement of picture cards than Order-IVL for all experiment scenarios.**

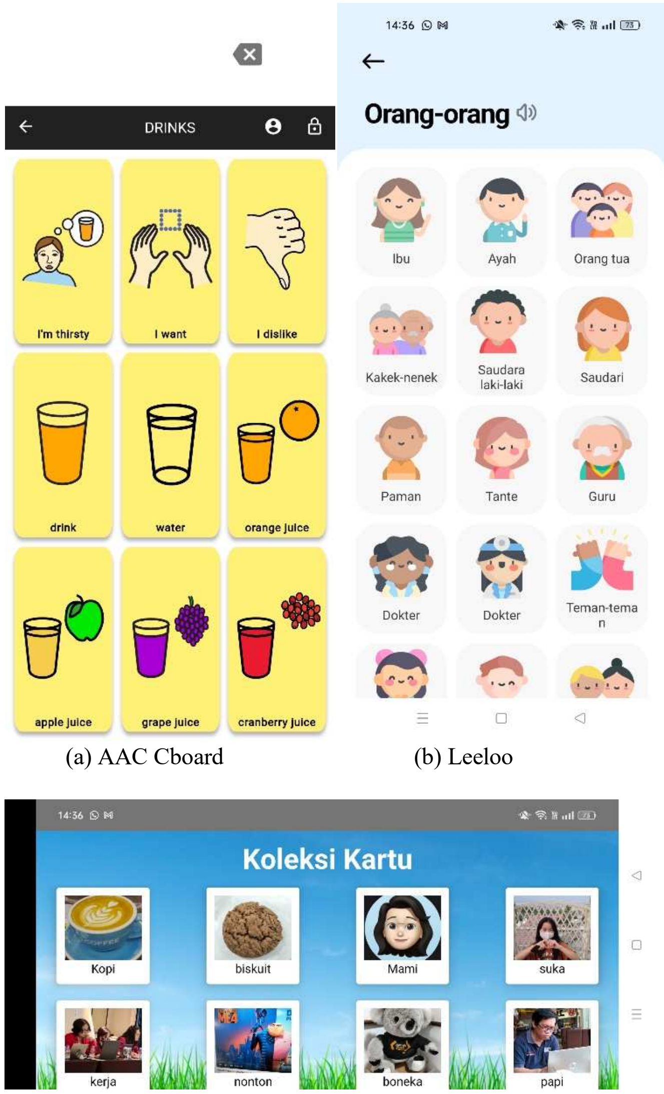
Keywords*— *Image View Layout, Picture Cards, 2D Bin Packing Problem, Integer Linear Programming, Greedy Approach.

I. Introduction

Bad experiences due to poor UI layout can cause a distressful feeling for users, and they should be avoided when dealingwith cognitively disabled users, such as people with autism spectrum disorder (ASD) [1], [2]. This paper studies a case of UI design for an augmented and alternative communication (AAC) application for users with ASD. In particular, the imagecollection feature, which is a display of images in an UI container with predefined size.

Each image in AAC is contained in a frame with a fixed width and height. Furthermore, the frame includes a short description of the image called a text label. We call a frame containing an image or a photograph and its text label as a picture card. Fig. 1 shows examples of picture card layout in three different mobile-based AAC applications: (a) AAC Cboard [3], (b) Leeloo [4], and (c) VICARA [5],[6]. Note that any tapping action on each picture card will generate speech according to the text label on the card. In this way, autistic users with speech difficulties can communicate their feelingsand needs to other people.

As can be seen in Fig. 1, the space to display the picture cards is bounded by the device screen. Therefore, the available picture cards cannot be displayed at once. Notice that each card frame has the same size, and the image size will be scaled to fit the frame. In this case, any image with a wider



(a) AAC Cboard (b) Leeloo

(c) VICARA

Fig. 1. Image View Layout in AAC Cboard, Leeloo andVICARA Mobile Applications

size is going to look smaller than the other images. This inconsistency may distract autistic people because, in general, they can be easily attracted to detail rather than the whole object [7].

In this paper, we propose an optimization problem that optimally arranges picture cards within a container with a given maximum width and height. All images of picture cards are various in width. Therefore, the picture cards are also vary in their width. We consider the width of images is scaled by fixing their height to a given size. Moreover, between every consecutive two picture cards must have a white space or padding, horizontally and vertically. Padding helps users organize visual information better. Furthermore, fixing the height size keeps the tidiness and simplicity ofthe picture cards display and prevents wider images frombecoming smaller.

In accordance with AAC applications, the problem aims to maximize the use of the container space by laying down as many picture cards as possible so that the overall number of picture card sets for display is minimum. Thus, the number of user actions to display the next set of picture cards can also be minimized. Consequently, the search for the required picture cards is more prompt. We call the optimization problem *imageview layout* (IVL), which is a special case of a well-known two dimensional bin packing problem, or 2DBPP [8].

2DBPP [8]–[10] is a combinatorial problem that has been proven to be NP-hard. It optimizes the allocation of a given collection of small rectangles (items) to as many large identicalrectangles (bins) as possible without having them overlap. The problem considers each item has different width and height.

Various exact methods for 2DBPP have been shown in [9] to solve only up to 100 rectangular items and sometimes unable to obtain solutions for a few as 20 items. On the other hand, approximate methods such as greedy algorithms [11], [12], meta-heuristics [13]–[15], reinforcement learning [16], and deep learning [17] can provide a close-to-optimal solution for a large number of items with significantly faster runtime than the exact techniques.

Chen *et al.* [14] addresses an user interface layout problem that maximizes the usage space of a large rectangle UI container by allocating as many as possible a set of small rectangular display components that vary in their width and height. They regard their problem as 2DBPP. To approach the optimal solution, they use the Firefly algorithm. Another study in [17] addresses an optimization of mobile UI layout, whichis the arrangement of UI components in a mobile device in such way that it minimize the task completion time and error rate when users interact with the components. They use adeep learning approach called gradient descent to find its near-optimal solution.

The IVL problem is a novel application of the 2D Bin Packing Problem. Chen *et al.* [14] work is the closest related work to ours. Their work deals with the arrangement of UI elements with various sizes in an UI container with predefinedwidth and height. On the other hand, the IVL problem arranges a set of images in a frame with a fixed height but its width is adjusted according to the width of its image.

Our main contributions are in three methods to solve the IVL problem. First, we formulate IVL as an integer linear programming and solve it using an optimizer to get its optimalsolution as a benchmark for a small number of rectangular items, i.e., picture cards, and UI containers. Second, two greedy-based heuristic algorithms are designed to provide a near-optimal solution for a larger number of the items and containers with a low computational cost.

The remainder of this paper is organized as follows. Section II formulates the IVL optimization problem. We present two heuristic algorithms in Section III and evaluate their performance in Section IV. Section V concludes the paper.

II. PROBLEM FORMULATION

We follow the six general steps of operations research methodology as described in [18] to perform the optimization study. They are (i) problem description, (ii) parameters and variables definition, (iii) mathematical modeling, (iv) optimal solution, (v) interpretation, and (vi) validation. This section discusses the first three steps, while the last three steps are described in Section IV.

A. Problem Description

The goal of IVL is to display as many picture cards as possible into a 2D UI container with a fixed width and height in order to minimize the number of such containers. IVL isa special case of 2DBPP, where the item has a fixed height. Furthermore, IVL guarantees that each picture card must be packed into exactly one container, in other words, it has to be displayed only once. Finally, IVL ensures that every card lies inside the UI container.

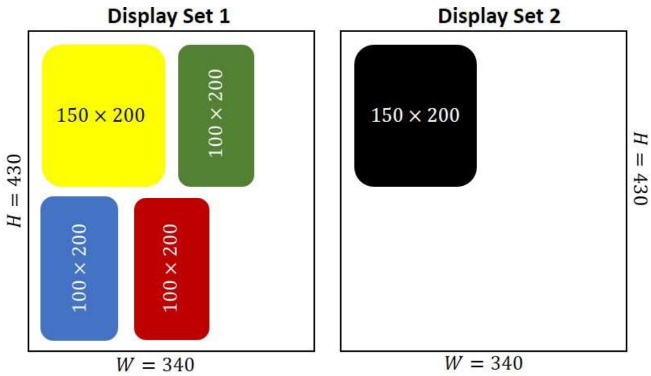
To illustrate IVL problem, consider (1) an 340 x 430 UI container depicted in Fig. 2, i.e., the container has a width size of 340 and height size of 430 units, e.g., in pixel, (2) five picture cards and their respective sizes of 150 x 200 (yellow), 100 x 200 (green), 100 x 200 (blue), 100 x 200 (red), and 150 x 200 (black), with a padding size of 10 units. Note that each display set is the same size of UI container containing a set of non-overlapping picture cards laying inside the container. Each container has a number of line sections with the same width and height. For example, display set 1in Fig. 2a consists of two line sections. The first or top line section contains yellow and green picture cards, while the second or bottom line section comprises of blue and red picture cards. Fig. 2a illustrates the non-optimum arrangement of the picture cards according to their order in point (2). The first display, i.e., display set 1, contains only the first four picture cards, which are the yellow, green, blue and red cards. The last picture card, i.e., the black card, is in the second display (display set 2). However, if we optimize the arrangement, as shown in Fig. 2b, all five picture cards can be laid in one display.

B. Parameters and Variables Definition

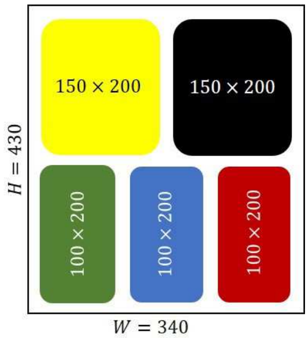
Let n rectangular picture cards with the same height are allocated into a limited number of m bins or display sets. Each bin $\forall i \in \{1,2, \ldots, m\}$has the same capacity, which is the same maximum weight W and height H. Every rectangular item i has its own width w_i and height h_i, for $\forall i \in \{1,2, \ldots, n\}$.

Following Liu *et al.* [10], 2DBPP consists of six decision variables:

- $z_k$is a binary variable set to 1 if the bin contains at least one item.
- $s_{ik}$is a binary variable set to 1 if item i is inside bin k.



(a) Non-optimum Display Set



(b) Optimum, One Display Set

Fig. 2. Image View Layout Illustration

- l_{ij} and u_{ij} are binary variables set to 1 if item i is located to the left or under item j, respectively, for $i \neq j$.
- x_i, y_i are positive integer variables including 0, denoted by $Z_{(i\geq 0)}$; they indicate the location of item i, its bottom and left corner, respectively.

C. Mathematical Modeling

An Integer Linear Programming in Eq. (1) presents the mathematical model of 2DBPP. Eq. (1a) is the objective of 2DBPP which is a minimum number of used bins. The objective is subject to constraints (1b) to (1o).

Constraint (1b) makes sure that each bin is set as used if it contains items, and constraint (1c) ensure that each item is put into exactly one bin. Constraints (1d) and (1e) guarantee that every item lays inside its bin. Next, constraints (1f) and (1g) ensure that all rectangular objects do not overlap. According to constraint (1h), this non-overlapping checking applies for the items that are in the same bin. Constraints (1i) and (1j) aid in lessening the symmetry issue, which is all used bins have their index value in order. While the later guarantees that every item is placed in a bin with an index no higher than the item's index, the former guarantees that a bin can only be used if all of the lower index bins are used. For instance, item 1 has to go in bin 1, and item 2 needs to go in either bin 1 or 2. Lastly, the lower bound o, which is the minimum number of used bins,is applied by constraint (1k). Finally, constraints (1l) to (1o) define the data types of every decision variable.

Constraints (1b), (1i), and (1l) are repeated for each bin k, where $\forall k \in \{1,2, \ldots, m\}$, while constraints (1c), (1d) to (1f), (1j) and (1o) is for $\forall i \in \{1,2, \ldots, n\}$. Next, constraints (1f), (1g), and (1n) apply to two different items $i \neq j$, where $\forall i, j \in \{1,2, \ldots, n\}$. Similarly, constraint (1h) is for two different items $i < j$ but it is for each bin k, where $\forall i, j \in \{1,2, \ldots, n\}$ and $\forall k \in \{1,2, \ldots, m\}$. Finally, constraint (1c) exists for each item i in each bin k.

min

$$\sum_{k=1}^{m} z_k \quad 1(a)$$

s.t.

$$\sum_{i=1}^{n} s_{ik} \leq n z_k, \quad 1(b)$$

$$\sum_{k=1}^{m} s_{ik} = 1, \quad 1(c)$$

$$x_i + w_i \leq W, \quad 1(d)$$

$$y_i + h_i \leq H, \quad 1(e)$$

$$x_i + w_i \leq x_j + W(1 - l_{ij}) \quad i \neq j, \quad 1(f)$$

$$y_i + h_i \leq y_j + H(1 - u_{ij}) \quad i \neq j, \quad 1(g)$$

$$s_{ik} + s_{jk} - 1 \leq l_{ij} + l_{ji} + u_{ij} + u_{ji} \quad i < j, \quad 1(h)$$

$$z_k \leq z_{k-1} \quad k > 1, \quad 1(i)$$

$$\sum_{k=1}^{i} s_{ik} = 1 \quad i < m, \quad 1(j)$$

$$\sum_{k=1}^{m} z_k \geq o, \quad 1(k)$$

$$z_k \in \{0,1\}, \quad 1(l)$$

$$s_{ik} \in \{0,1\}, \quad 1(m)$$

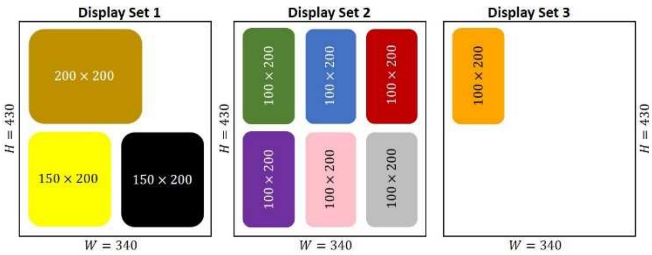
$$l_{ij}, u_{ij} \in \{0,1\} \quad i < j, \quad 1(n)$$

$$x_i, y_i \in Z_{(i\geq 0)}. \quad 1(o)$$

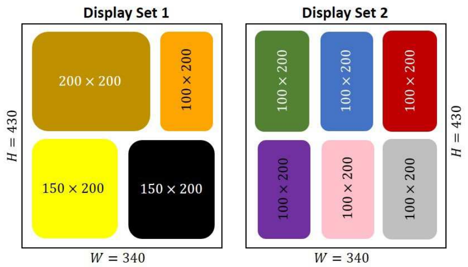
We solve the ILP and use its optimal solutions as benchmarks to evaluate the performance of the following two proposed greedy algorithms. The performance measurement involves the number of bins, which are the display sets, resulted from both ILP and the two algorithms.

III. GREEDY-BASED HEURISTIC ALGORITHMS

As stated in [10], [17] and [14], 2DBPP is an NP-Hard problem and its running time to obtain an optimal solution increases exponentially with larger number of picture cards and the maximum number of display sets. Although IVL fixes the height size of all items, their width is still vary. Therefore, IVL can be categorized as a combinatorial problem. Accordingly, IVL is also an NP-hard problem. Solving a large number of instances using an exact method is time-consuming and does not provide certainty of a sufficiently good convergence to a global optimum [9]. Therefore, we propose two greedy-based heuristic algorithms to provide approximate solutionsfor larger number of pictures cards and display sets.



(a) Solution using descending Order-IVL algorithm



(b) Solution using Best-Fit-IVL algorithm

Fig. 3. Solutions using descending Order-IVL and Best-Fit-IVL algorithms

A. Order-IVL

The first greedy algorithm simply allocates picture cards in ascending or descending order of their width, starting from the top line section of the container. We call the first algorithm as Order-IVL with its time complexity of $O(n \log n)$. We assume that an $O(n \log n)$ sorting algorithm such as merge sort is used. Therefore, this algorithm's time complexity increases *linearly* with the number of picture cards. Fig. 2b shows a solution generated by Order-IVL by sorting the five picture cards as described in Section II-A in decreasing order.

Let consider adding another five picture cards with the following sizes: 200 x 200 (brown), 100 x 200 (purple), 100 x 200 (pink), 100 x 200 (gray), and 100 x 200 (orange). Fig. 3a shows the results of using descending Order-IVL algorithm. Note that using ascending Order-IVL also produces three display sets. The orange picture card in the third display set can be fit into the first display set so that two display sets should be sufficient.

B. Best-Fit-IVL

We propose another greedy algorithm called **Best-Fit-IVL**. It considers the following two cases and takes different actions accordingly when a current picture card has a width larger than the remaining width space for the current line section:

1) **case** a: the remaining width space is smaller than an unassigned picture card which has not been assigned to any display set and has the smallest width.
   - action: if it is fit, put the current picture card to the next line section, otherwise put it in the first line section of the next display set.
2) **case** b: the remaining width space is equal or larger than a remaining picture card with the smallest width.
   - action: find an unassigned picture card that best fit for the remaining width space, i.e., the card which takes the most remaining space.

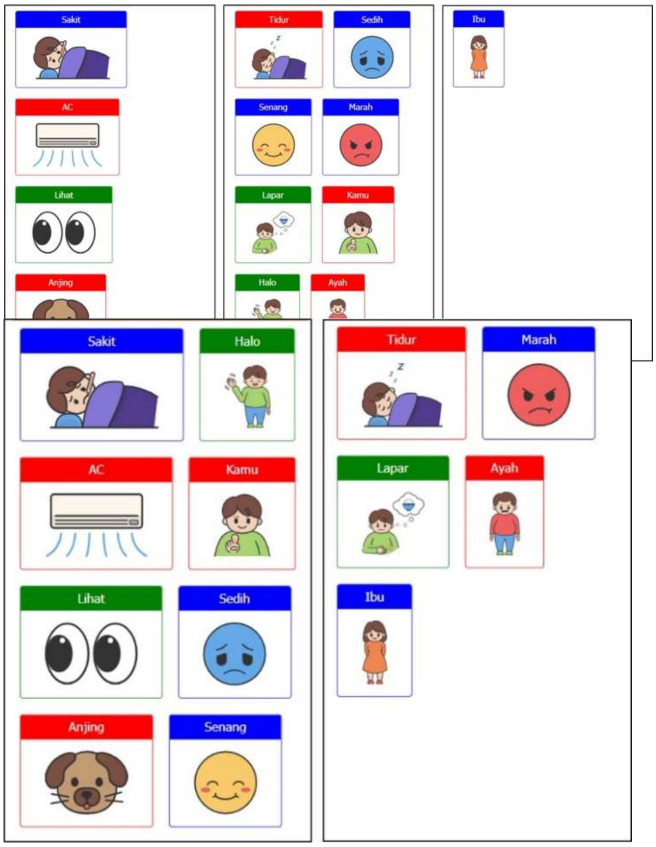
In the worst case, Best-Fit-IVL performs action for case b for every picture card allocation. Therefore, its overall time complexity can be accounted as a time to sort the picture cards in $O(n \log n)$ find the best-fit picture card among those that have not yet laid in the container in $O(n^2)$. Therefore, the total time complexity for Best-Fit-IVL is$O(n \log n + n^2) = O(n^2)$. Fig. 3b shows the results of using Best-Fit-IVL. As shown in the figure, 10 picture cards can now be arranged into only two display sets. The time complexity of Best-Fit-IVL increases quadratically with the number of picture cards. With this type of time complexity, the algorithm will be impractical for a very large number of picture cards.

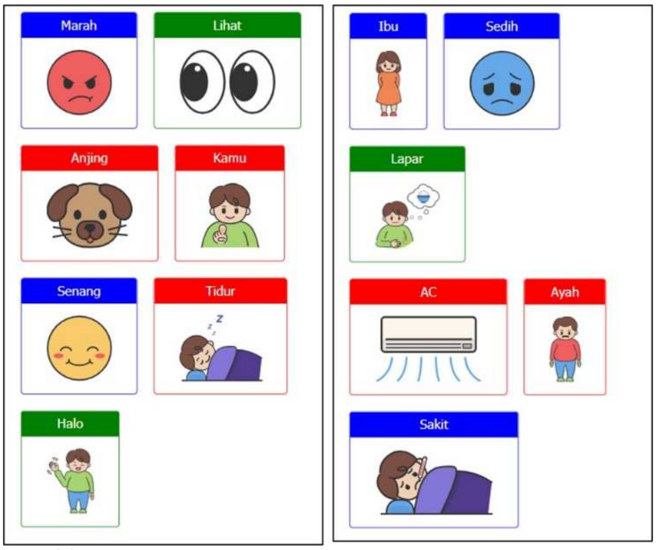
IV. RESULT AND DISCUSSION

We have implemented Order-IVL and Best-Fit-IVL in Python and used Gurobi for Python [19] to solve ILP in Eq. (1). Furthermore, a simple website using Django framework is developed to provide a visualization of IVL solution from the two greedy algorithms and ILP. All of our experiments are conducted on a 64-bit Windows machine with an 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz and 16 GB of memory.

(a) Descending Order-IVL

(b) Best-Fit-IVL





(c) ILP

Fig. 4. Solution samples using descending Order-IVL, Best-Fit-IVL and ILP

We use up to 148 picture cards of VICARA application [20] to evaluate the two greedy algorithms against the optimal(benchmark) solution obtained by solving ILP (1). Each picture card has a fixed height of 145 pixels and a white space for each side of the card set to 23 pixels. Furthermore, we use device mode in Chrome browser to simulate different sizes ofthe UI container in four mobile devices, iPhone 12 Pro, Pixel7, iPad Mini and Samsung Galaxy A51/71. Note that the sizeof the container is made less than the device screen resolution.

Fig. 4 shows a sample of IVL solution produced by the three methods in iPhone 12 Pro. The three solutions are visualized as a Django website. Among the three solutions, Order-IVLgenerates three displays sets, while the other two methods require only two sets for 13 picture cards. Notice that Best-Fit-IVL provides better arrangement, i.e., tidier and neater, than the other two methods. It is reasonable because Best-Fit-IVL tries to use the container space as much as possible and atearlier display sets as possible.

The number of display sets is mainly used to measure the performance measurement of Order-IVL, Best-Fit-IVL, and ILP. Table I shows the number of display sets resulted from Order-IVL, Best-Fit-IVL and ILP in four different mobile devices. We define three scenarios based on the amount of picture cards, which are 13, 30 and 148 cards. After running ILP for one day, we fail to get the optimal solutions in anyof the four mobile devices for the larger number of picture cards, i.e., 30 and 148 cards. We exclude the running timeof the three methods from Table I because from their time complexity described in Section III, Order-IVL and ILP have respectively the fastest and the slowest runtime than theircounterparts. These trends consistently apply to all scenarios. As an example, Order-IVL, Best-Fit-IVL and ILP produce two display sets for 13 picture cards in 1.39, 2.01 and 146.96 seconds, respectively, in Samsung Galaxy A51/71.

TABLE I. EXPERIMENTAL RESULTS OF RUNNING ORDER-IVL, BEST-FIT-IVL AND ILP

Number of Picture Cards	Methods	Number of Display Sets in Four Mobile Devices			
		iPhone 12 Pro (370 × 645)	Pixel 7 (410 × 716)	iPad Mini (748 × 825)	Samsung Galaxy A51/71 (392 × 715)
13	**Order-IVL**	3	2	1	2
	Best-Fit-IVL	2	2	1	2
	ILP	2	2	1	2
30	**Order-IVL**	5	4	2	4
	Best-Fit-IVL	4	4	2	4
	ILP	-	-	-	-
148	**Order-IVL**	9	7	4	8
	Best-Fit-IVL	9	7	3	7
	ILP	-	-	-	-

As shown in Table I, ILP and Best-Fit-IVL arrange the first 13 picture cards in only 2 display sets for iPhone 12 Pro, while Order-IVL needs one extra display set. However, for the other three devices, all methods lay the 13 cards into the same number of display sets. For the larger number of picture cards, Order-IVL needs one more display set to lay 30 cardsin iPhone 12 Pro and 148 cards in iPad Mini and Samsung Galaxy A51/71 than Best-Fit-IVL.

Overall, Best-Fit-IVL outperforms Order-IVL in minimizing the number of display sets in order to maximize the use of container space at the earlier display sets. Further, our research findings find that consistently, Best-Fit-AVL provides a better,i.e., a tidy and neater, arrangement of the picture cards in eachdisplay set for all card sets in all four devices. Moreover, our experimental results show that the runtime of Order-IVL is only slightly faster than Best-Fit-IVL. Notice also that Best- Fit-AVL and ILP always produce the same optimal number ofdisplay sets for the 13 picture cards. Here, we can conclude that Best-Fit-IVL is the best alternative solution for IVL whengetting the optimal solution from ILP is not feasible due to its time-consuming solver.

V. CONCLUSION

This paper introduces the optimization problem of image view layout (IVL) focusing on placing a collection of images in a given number of 2D user interface (UI) containers withthe same size. Here, we view each image as a picture card witha text label and an image or photograph on it. Furthermore, a container for our case is called a display set containing a set ofnon-overlapping images that can be laid within one container. In order to put a maximum number of picture cards into one container while using the fewest available containers, the studyaims to maximize container space. We regard IVL as a specialcase of 2DBPP, where the items are picture cards with the same height.

In order to model IVL problem, the study uses integer linear programming (ILP). Since IVL is an NP-hard problem, we design two greedy-based heuristic algorithms, Order-IVL and Best-Fit-IVL, to generate close-to-optimal solutions. Based onthe research results, it appears that Best-Fit-IVL consistently arranges picture cards neatly in minimum number of display sets than Order-IVL. Moreover, Best-Fit-IVL can produce the same minimum number of display sets as ILP for a small number of photo cards.

For future work, we plan to (1) use more picture cards, (2) approach the near-optimal solution using meta-heuristic methods such as swarm optimization algorithms and machine learning such as Gradient Descent, and (3) consider other factors influencing IVL, particularly in relation to AAC application for users with ASD and speech difficulty.

ACKNOWLEDGMENT

The authors would like to thank Oxana Agatha Guijaya, S.Sn and Ruby Chrissandy, S.Sn., M.Ds for their picture cards design used as the experimental dataset in this paper.

REFERENCES

[1] M. Rezae, N. Chen, D. McMeekin, T. Tan, A. Krishna, and H. Lee, “Theevaluation of a mobile user interface for people on the autism spectrum: An eye movement study,” *International Journal of Human-Computer Studies*, vol. 142, p. 102462, 2020.

[2] M. Alzahrani, A. L. Uitdenbogerd, and M. Spichkova, “Human-computer interaction: Influences on autistic users,” *Procedia Computer Science*, vol. 192, pp. 4691–4700, 2021.

[3] luxi, “Cboard AAC - Communication for Everyone — cboard.io,” https://www.cboard.io/en/, [Accessed 06-08-2024].

[4] “Leeloo AAC - Assistive Cards — assistivecards.com,” https://assistivecards.com/leeloo/, [Accessed 06-08-2024].

[5] H. Hersinta, C. Bangun, and O. Hutagaol, “Developing vicara 2.0: Exploring the potential use of augmentative and alternative communication (aac) apps for the parents and teachers of autistic students,” in *AIP Conference Proceedings*, vol. 2680, no. 1. AIP Publishing, 2023.

[6] "VICARA - Apps on Google Play - play.google.com," https://play.google.com/store/apps/details?id=com.vicara.vicara2, [Accessed 30-08-2024].

[7] “Sensory differences - a guide for all audiences — autism.org.uk,” https://www.autism.org.uk/advice-and-guidance/topics/sensory-differences/sensory-differences/all-audiences, [Accessed 30-08-2024].

[8] A. Lodi, S. Martello, and D. Vigo, “Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems,” *INFORMS journal on computing*, vol. 11, no. 4, pp. 345–357, 1999.

[9] S. Polyakovskiy and R. M’Hallah, “Just-in-time two-dimensional bin packing,” *Omega*, vol. 102, p. 102311, 2021.

[10] C. Liu, K. Smith-Miles, T. Wauters, and A. M. Costa, “Instance space analysis for 2d bin packing mathematical models,” *European Journal of Operational Research*, vol. 315, no. 2, pp. 484–498, 2024.

[11] A. M. Chwatal and S. Pirkwieser, “Solving the two-dimensional bin-packing problem with variable bin sizes by greedy randomized adaptive search procedures and variable neighborhood search,” in *International Conference on Computer Aided Systems Theory*. Springer, 2011, pp. 456–463.

[12] H. Zhang, Q. Liu, L. Wei, J. Zeng, J. Leng, and D. Yan, “An iteratively doubling local search for the two-dimensional irregular bin packing problem with limited rotations,” *Computers & Operations Research*, vol.137, p. 105550, 2022.

[13] S. Kosari, M. Hosseini Shirvani, N. Khaledian, and D. Javaheri, “A hybrid discrete grey wolf optimization algorithm imbalance-ness aware for solving two-dimensional bin-packing problems,” *Journal of Grid Computing*, vol. 22, no. 2, pp. 1–36, 2024.

[14] W. Chen, H. Yu, X. Li, L. Qu, and Z. Mi, “Layout design with afirefly algorithm for user interfaces in vehicle system,” in *2020 IEEE 3rdInternational Conference on Information Systems and Computer Aided Education (ICISCAE)*. IEEE, 2020, pp. 110–113.

[15] X. Zhang, M. Shan, and J. Zeng, “Parallel batch processing machine scheduling under two-dimensional bin-packing constraints,” *IEEE Trans-actions on Reliability*, vol. 72, no. 3, pp. 1265–1275, 2022.

[16] K. Zhu, N. Ji, and X. D. Li, “Hybrid heuristic algorithm based on improved rules & reinforcement learning for 2d strip packing problem,” *IEEE Access*, vol. 8, pp. 226 784–226 796, 2020.

[17] P. Duan, C. Wierzynski, and L. Nachman, “Optimizing user interface layouts via gradient descent,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[18] Z. Fan, B. Ghaddar, X. Wang, L. Xing, Y. Zhang, and Z. Zhou, "Artificial intelligence for operations research: Revolutionizing the operations research process,” *arXiv preprint arXiv:2401.03244*, 2024.

[19] “Python API Overview - Gurobi Optimization — gurobi.com,” https://www.gurobi.com/documentation/current/refman/py python api overview.html, [Accessed 09-08-2024].

[20] "VICARA - Aplikasi di Google Play - play.google.com," https://play.google.com/store/apps/details?id=com.bithouse.vicara& hl=id, [Accessed 09-08-2024].

ICECOS

The 4th International Conference on Electrical Engineering and Computer Science

September 25 - 26, 2024

https : //icecos.unsri.ac.id/2024

IEEE
IEEE SMC Systems, Man, and Cybernetics Society
Indonesia Chapter
IEEE Universitas Sriwijaya IEEE Student Branch
ICECOS 2024 INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

CERTIFICATE
OF APPRECIATION

No. 0835/UN9.FT/TU.KT/2024

THIS CERTIFICATE IS PRESENTED TO

Lely Hiryanto

In recognition and appreciation of your contribution as

PRESENTER

ICECOS 2024 INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Dr. Bhakti Yudho Suprapto
Dean of Faculty Engineering Universitas Sriwijaya

Dr. Herlina
General Chair

CoCIS EMREC comnets Electrical

CERTIFICATE

OF APPRECIATION

No. 0835/UN9.FT/TU.KT/2024

THIS CERTIFICATE IS PRESENTED TO

Lely Hiryanto; Andhika Putra Wirawan; Tony Tony

In recognition and appreciation of your contribution as **Author(s)** at the 4th International Conference on Electrical Engineering and Computer Science (ICECOS) 2024.

Dr. Bhakti Yudho Suprapto
Dean of Faculty Engineering Universitas Sriwijaya

Dr. Herlina
General Chair