Contribution Title

Anna Vitali[1][0000−1111−2222−3333], Second Author[2,3][1111−2222−3333−4444], and Third Author[1][2222−−3333−4444−5555]

Alma Mater Studiorum Università di Bologna, Bolgona, Italy
{anna.vitali7}@unibo.it

Abstract. In the manufacturing industry, maintaining the stability of workpieces during operations such as cutting, milling, and contouring is essential to ensuring the quality of the final product. This stability is achieved using fasteners that are strategically placed to minimize vibrations and oscillations. However, determining the optimal positioning of these fasteners, ensuring their safety during cutting operations, and keeping the workpiece securely in place is a complex task, especially when dealing with composite materials that consist of multiple components requiring simultaneous support.
Although modern machining systems offer assistance in determining fastener positions, these suggestions must closely align with the expertise of experienced operators. This paper presents a solution to the fastener positioning problem using Constraint Programming (CP). By capturing and formalizing the knowledge of skilled operators, we demonstrate how CP can be applied to improve upon existing system recommendations. The results show that this approach can reduce errors and enhance the overall stability of the component during manufacturing processes, leading to improved quality and precision in the final product.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

In manufacturing industry, finding the correct positioning of the parts fasteners during the milling and cutting processes, in order to reduce vibrations and keep the element as firm as possible, is a fundamental problem to be solved to ensure a certain quality of the finished product [5].

Several studies have been carried out on the forms and materials of fasteners that can be used to minimize oscillations and vibrations [12, 3], among these one of the most adopted strategies in the manufacturing field involves the use of adsorption methods based on the adoption of vacuum suction cups [16, 4].

In this study, we will use as a reference a machine capable of performing operations on a wooden panel, such as milling, contouring, and drilling, supported by a system of movable bars on which suction cups can be vertically adjusted for secure holding.

Several factors must be taken into account for effective positioning of suction cups, including the type of machining operation, the specific areas of the workpiece that will be affected, and the geometry of the part [11]. In particular, when dealing with cutting operations, it is essential to avoid placing fixtures beneath areas where through-cuts will occur to prevent potential damage to the fasteners.

It is also important to consider that the available resources are limited. Typically, the machine has a fixed capacity, allowing for only a certain number of bars and a finite number of various types of fixings. Within these constraints, we must decide how to allocate the resources efficiently, ensuring that we do not exceed the available quantity while providing adequate support for the workpiece. The objective, however, is not to maximize the use of resources but to strategically cover the most critical areas of the workpiece to achieve optimal support.

To address this problem, we propose a constraint optimization model to decide where to place the bars and on them where the suction cups can be inserted and we compered the results with those generated by the current system. The comparison showed that the solution derived from CP was more straightforward to implement, lowering the risk of human error. In the current system, the machine operator is required to manually determine the number of suction cups to use, which heightens the possibility of mistakes, particularly when an insufficient number is chosen.

Moreover, the CP-based solution demonstrated an improvement in the support of the workpiece. Specifically, the solution ensures that if components are cut from the original piece, each sub-part large enough to accommodate suction cups is adequately supported, thereby preventing its unintended detachment. This is achieved by ensuring that the center of gravity of each sub-part remains within the perimeter formed by the fixtures inside it. In contrast, examples show that some solutions generated by the current system fail to meet this requirement, whereas the CP solution consistently satisfies this condition.

The remainder of this article is organized as follows. Section 2 presents an overview of similar problems that have been addressed using Constraint Programming. In Section 3, we explain the method used to identify the optimal points for support placement and how these points are represented. Section 3 details the constraint model developed to determine the positioning of the elements. Finally, Section 4 presents the results of our approach, comparing them with those of the previous system.

2 Related works

In this section, we analyze research on the optimal placement of fasteners and suction cups in manufacturing processes, with a particular emphasis on constraint optimization, vibration reduction, and support during machining operations. Additionally, we present studies addressing positioning problems that have been solved through Constraint Programming (CP), alongside strategies closely related to those implemented in our work.

In the study by Avigal et al. [1], the authors investigate suction cup grasping in industrial applications, focusing on the balance between suction forces and motion to prevent detachment. They propose a hybrid optimization approach that integrates deep learning techniques, demonstrating that the learned constraint can allow the solver to speed-up motions. Lee et al. [8] introduce a grasp failure model applicable to various suction cup configurations, highlighting constraints that ensure fast and reliable object manipulation. These constraints have been incorporated into trajectory optimization algorithms and used to address time-optimal trajectory planning. In another study [10], the authors explore optimization techniques for fastener placement under constrained resources in airplane assembly process, proposing and comparing several algorithms.

The problem of determining the optimal positioning of elements with specific characteristics within a defined area has been extensively studied in the literature, particularly within the field of telecommunications [15]. Several studies have addressed the optimal positioning of devices within environments, such as beacons or workstations, to ensure adequate coverage of a specific area, considering the characteristics of devices. These studies often employ Constraint Programming to determine the most effective solution [9, 6], and a commonly strategy for modeling such problems involves generating a map of the area of interest, wherein regions are identified as more or less suitable for device placement [9, 6, 13]. This approach is highly applicable to our problem and has been adopted in the present study to guide the positioning of fasteners.

3 Identification of Optimal Fastener Placement Points

The geometry and surface area of the part must be considered when identifying the optimal fixture points. It is important to recognize that different types of elements may be produced: individual components that do not require further machining to generate additional parts, and composite components, where material is removed from a single element to create the final shape or where multiple components are derived from a single element through machining.

Expert knowledge regarding the proper positioning of suction cups suggests that, for small parts capable of supporting at least one fixture, the central area should be prioritized. For medium-sized parts, it is preferable to focus on the perimeter, while for larger parts, both the perimeter and the center must be adopted for support. To categorize the part size, the relationship between the surface area of the element and the available support area is examined:

$$ratio = \frac{element\_area}{support\_area}$$

The proposed solution involves representing the part and the machining process in a matrix, where precision is measured in millimeters to match the machine's accuracy. Each cell in the matrix is assigned a score that reflects the suitability of that specific point for securing a fixture (fig. 2). In the case of composite parts, higher priority is given to the innermost components, resulting in higher scores compared to the outermost elements, thereby favoring their priority. For positioning purposes, the top-left corner of the fixture is used as a reference point, as illustrated in fig. 1, meaning that the solution aims to determine the coordinates of this point while accounting for the spatial dimensions of the part.

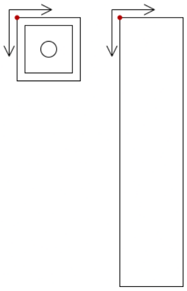


Fig. 1: Reference point for bar and suction cups positioning

The score matrix functions as a map to be utilized by the model in guiding the placement of fixtures. This mapping process allows for the identification of optimal fixture locations, ensuring that constraints related to the geometry of the part are respected. For instance, in the case of a medium-sized circle, the matrix scores are likely to be higher near the center and lower towards the perimeter.

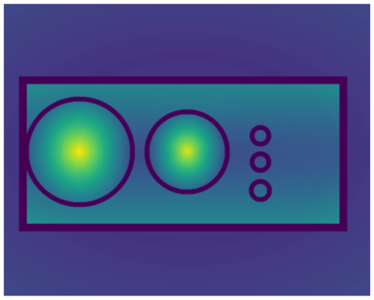


Fig. 2: Score matrix for a dashboard

The fig. 2 shows a composite piece (a dashboard) consisting of a medium-sized inner rectangle, so the perimeter is highlighted and five circles for which three of them are too small to be supported and the other two, instead, are considered small, and for them the center is highlighted.

4 Model

In this section, we delineate the fundamental components of our model, including the input parameters, variables, constraints, and objective function. The model is characterized as *parametric* because the definitions of the variables, constraints, and objective function are contingent upon the input parameters.

The model developed to establish the positioning of the devices employs the strategy of the Knapsack algorithm [7]. Specifically, this problem can be interpreted as a variation of the Geometric Knapsack problem, in which only one dimension of the objects is considered: the width for the bars and the height for the suction cups.

The model is executed iteratively. Initially, it determines the optimal positioning of the bars, and subsequently, based on this, it identifies the placement of the suction cups for each bar.

4.1 Parameters

The input parameters are those *constant* values shaping a particular *instance* of the model [14]. We formalize them as follows.

Capacity, Resources and Position We define $\mathsf{capacity}$ as the total space available along the reference axis for the placement of bar or suction cup devices. Additionally $\mathsf{max\_resources}$ represents the maximum number of bar or suction cups that can be utilized by the machine, while $\mathsf{max\_position}$ indicates the maximum number of locations available for device insertion.

We assume that all these values are integers, determined based on the geometry of the workpiece and the resources allocated to the machine.

The placement of objects must be arranged in such a manner that they do not overlap with one another and do not exceed the available capacity.

Object and Positions Let $\mathsf{OBJECT} = \{1, \ldots, \mathsf{max\_position}\}$ be the set of objects, and $\mathsf{POSITION} = \{1, \ldots, \mathsf{max\_positions}\}$ be the set of available positions. The size of each object $i \in \mathsf{OBJECT}$ is given by the function $\mathsf{object\_size} \colon \mathsf{OBJECT} \to \mathbb{N}$, where $\mathsf{object\_size}[i]$ denotes the size of object i.

The profit of each position $j \in \mathsf{POSITION}$ is defined by the function $\mathsf{position\_profit} \colon \mathsf{Position} \to \mathbb{N}$ where $\mathsf{position\_profit}[j]$ indicates the profit associated with the position j.

The goal of the model is to maximize the total profit resulting form locating objects in the available positions.

4.2 Variables

We define the model's primary variable, $\mathsf{object\_position}[i]$, which indicates the placement of object i on the axis of interest, where $\mathsf{object\_position}[i] \in \mathsf{POSITION}$. Each object is either selected or not selected for placement. The selection of object i is represented by a binary variable $\mathsf{object\_selected}[i] \in \{0, 1\}$, defined as:

$$\mathsf{object\_selected}[i] = \begin{cases} 1, & \text{if object } i \text{ is selected,} \\ 0, & \text{if object } i \text{ is not selected.} \end{cases}$$

This approach allows us to know the number of resources that will be used and for each of them what will be its placement in the workplane, leaving out the resources that will not be allocated.

4.3 Constraint and Objective

The idea of Constraint Programming is to solve problems by stating constraints about the problem, and consequently, finding solution satisfying all the constraints. A *constraint* is simply a logical relation among variables, restricting the possible values which they can take [2]. For our problem we defined the following constraints.

Capacity Constraint First of all we need to ensure that the total space occupied by the selected objects does not exceed the available $\mathsf{capacity}$ which is the width for the bars and the height for the suction cups of the workpiece. This is formalized by the constraint

$$\sum_{i \in \mathsf{OBJECT}} \mathsf{object\_size}[i] \cdot \mathsf{selected}[i] \leq \mathsf{capacity}$$

Non-overlapping Constraint To avoid overlapping objects, we ensures that the space occupied by the $i-th$ object, which is specified by $\mathsf{object\_position}[i]$, if selected does not occupy the position of the next object indicated by $\mathsf{object\_position}[x+1]$. This is written as:

$$(\mathsf{object\_position}[i]+\mathsf{object\_size}[i]) \cdot \mathsf{object\_selected}[i] < \mathsf{object\_position}[i+1] \quad \forall i \in \mathsf{OBJECT}$$

Increasing Order Constraint The increasing order constraint ensures that the positions of the selected objects are in strictly increasing order. This constraint is of interest to obtain the coordinates of elements in ascending order, for correctly represent them later in post-processing, and is captured by the built-in *increasing* constraint:

$$increasing(\mathsf{object\_position})$$

4.4 Objective Function

The objective function is designed to maximize the profit generated from the strategic positioning of devices.

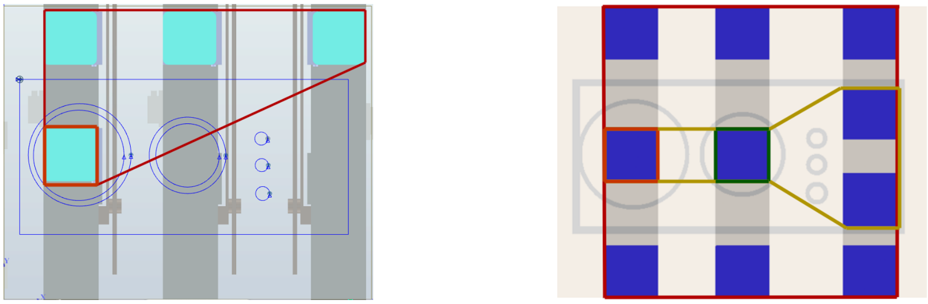
The selected object ($\mathsf{object\_selected}$) are positioned in specific locations ($\mathsf{object\_position}$), each associated with a known profit ($\mathsf{position\_profit}$), as a result the objective function becomes:

$$\max \sum_{i \in \mathsf{OBJECT}} \mathsf{position\_profit}[\mathsf{object\_position}[i]] \cdot \mathsf{object\_selected}[i]$$

This function ensures that only the selected objects contribute to the overall profit, with the profit being contingent upon their respective placement positions.

5 Analysis of results

Fig. 3: Comparing the solution provided by the system with the one obtained through Constraint Programming



(a) Current System Dashboard Solution

(b) Constraint Programming Dashboard Solution

Comparing the results obtained through the solution implemented through Constraint Programming with that of the current system, is noted that the support provided to the workpiece is better, as it is able to comply with the rules stated above, succeeding in case of composite parts to support the elements that must be exported.

The following figure illustrates the solution provided by the system for the dashboard illustrated above. It is evident that the perimeter defined by the suction cups does not sufficiently support either the smaller circle or the inner rectangle. Additionally, the support for the outermost rectangle is critically compromised, as the lower right section lacks fasteners.

In contrast, the solution derived from Constraint Programming demonstrates the successful integration of suction cups that supports both circles, as well as the inner rectangle. The inner rectangle is particularly challenging to support due to its medium size, necessitating a focus on its perimeter and the machining process required to create the circles restricts the available space for fasteners along this. However, the strategic placement of suction cups at the center of the circles provides adequate support for the rectangle, which will be processed first. For further examples, please refer to the appendix.

Finally, the solution obtained through constraint programming can automatically manage the available resources, eliminating the need for the operator to manually specify the number of suction cups required. The necessary quantity is autonomously determined.

References

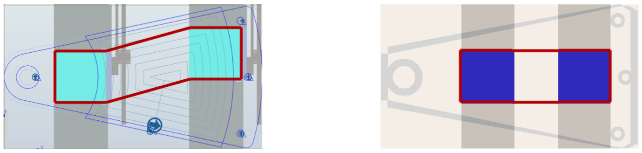
1. Avigal, Y., Ichnowski, J., Cao, M.Y., Goldberg, K.: Gomp-st: Grasp optimized motion planning for suction transport. In: International Workshop on the Algorithmic Foundations of Robotics. pp. 488–505. Springer (2022)
2. Barták, R.: Constraint programming: In pursuit of the holy grail. In: Proceedings of the Week of Doctoral Students (WDS99). vol. 4, pp. 555–564. MatFyzPress Prague (1999)
3. Chung, D.: Materials for vibration damping. Journal of materials science **36**, 5733–5737 (2001)
4. Del Sol, I., Rivero, A., López de Lacalle, L.N., Gamez, A.J.: Thin-wall machining of light alloys: A review of models and industrial approaches. Materials **12**(12), 2012 (2019)
5. Fei, J., Xu, F., Lin, B., Huang, T.: State of the art in milling process of the flexible workpiece. The International Journal of Advanced Manufacturing Technology **109**, 1695–1725 (2020)
6. Frühwirth, T., Brisset, P.: Optimal placement of base stations in wireless indoor telecommunication. In: International Conference on Principles and Practice of Constraint Programming. pp. 476–480. Springer (1998)
7. Kellerer, H., Pferschy, U., Pisinger, D., Kellerer, H., Pferschy, U., Pisinger, D.: Multidimensional knapsack problems. Springer (2004)
8. Lee, J.e., Sun, R., Bylard, A., Sentis, L.: Grasp failure constraints for fast and reliable pick-and-place using multi-suction-cup grippers. arXiv preprint arXiv:2408.03498 (2024)
9. Löffler, S., Kroll, F., Becker, I., Hofstedt, P.: Optimal beacon placement for indoor positioning using constraint programming. In: 2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA). pp. 1–8. IEEE (2022)
10. Miriyev, T., Huber, L., Frieden, M., Amer, Z., Chen, Y.: Fastener installation pattern optimization in airplane assembly process. arXiv preprint arXiv:2010.07256 (2020)
11. RACHIERU, N., NIȚĂ, R.M.: Methodology for some orientation and clamping devices integration on cnc machine tools
12. Sivasubramanian, S.: Optimization and simulation of formed suction cup to position aerospace cfrp parts during milling. (2019)
13. Vlasenko, I., Nikolaidis, I., Stroulia, E.: The smart-condo: Optimizing sensor placement for indoor localization. IEEE Transactions on Systems, Man, and Cybernetics: Systems **45**(3), 436–453 (2014)
14. Wallace, M.: Building decision support systems: using MiniZinc. Springer (2020)
15. Younis, M., Akkaya, K.: Strategies and techniques for node placement in wireless sensor networks: A survey. Ad Hoc Networks **6**(4), 621–655 (2008)
16. Zhu, T., Liu, R., Wang, X.D., Wang, K.: Principle and application of vibrating suction method. In: 2006 IEEE International Conference on Robotics and Biomimetics. pp. 491–495. IEEE (2006)

A Appendix

Below we will show some examples of results, comparing the solutions obtained by the current system with those obtained through Constraint Programming.

In this first example, the element to be realized is a stair step. The solution provided by the current system is not wrong but not optimal, as the right suction cup is too high. In the solution obtained through Constraint Programming, on the other hand, the element is considered of medium size and although the number of fixings used is the same as that of the current system, the solution allows the right suction cup to be positioned more toward the center of the height of the workpiece.

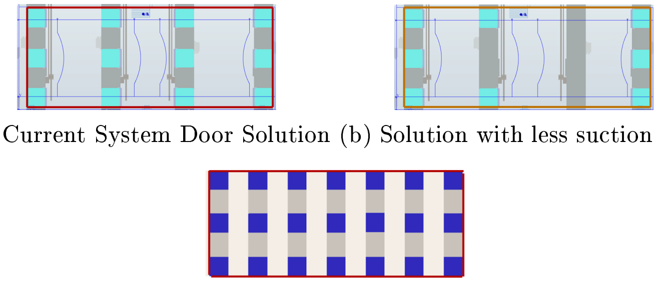
Fig. 4: Comparing the solution provided by the system with the one obtained through Constraint Programming



(a) Current System Dashboard Solution (b) Constraint Programming Dashboard Solution

In this example, the creation of a door is required. The solution provided by the system is deemed satisfactory, assuming that the operator accurately inputs the number of suction cups. However, if the operator makes an error in this input, the resulting solution is likely to be incorrect, as evidenced by the lack of support in the center section of the component.

Fig. 5: Comparing the solution provided by the system with the one obtained through Constraint Programming



(a) Current System Door Solution (b) Solution with less suction cups

(c) Constraint Programming Door Solution

By employing a constraint-based programming approach, the necessity for manual input of the number of fasteners is eliminated. Consequently, the provided solution will consistently reflect the optimal configuration, thereby mitigating the risk of errors.