

HEURISTIC METHOD FOR THE TWIN-ROBOTS SCHEDULING PROBLEM IN PICKUP AND DELIVERY SYSTEMS

David Lourenzo de Oliveira Kailer

Federal University of Santa Catarina – UFSC

Rua Dona Francisca, 8300, Bloco U, Zona Industrial Norte, CEP: 89.219-600, Joinville, SC, Brasil

davidkailer09@gmail.com

Silvia Lopes de Sena Taglialenha

Federal University of Santa Catarina – UFSC

Rua Dona Francisca, 8300, Bloco U, Zona Industrial Norte, CEP: 89.219-600, Joinville, SC, Brasil

s.taglialenha@ufsc.br

ABSTRACT

This research considers the Twin-Robot Pallet Assignment and Scheduling Problem - TRPASP, in which two robots are positioned on opposite sides of a rail to perform pickup/delivery operations. The problem consists of defining which products should be assigned to each robot, indicating the pickup point and the delivery point, as well as the sequence in which each robot will perform the operations, aiming to minimize the total time to perform the tasks, and addressing the safety constraints to avoid collisions between the robots. A PLIM model that determines the global optimal solution for small instances is presented, considering that the problem is NP-hard. For larger instances, a heuristic approach capable of determining feasible solutions in very satisfactory computational time is presented. The heuristic is validated for instances where the optimal solution is known. For larger instances the results analyzed have been satisfactory.

KEYWORDS. Twin-Robots Scheduling, Assignment task location, Heuristic methods.

1. Introduction

Planning is key for efficiency and increased competitiveness in various segments, being an important factor for quick and effective decision-making. In the field of operational planning, it is essential to understand the optimization methods to reduce the resources required to perform the operations and services. In automated or hybrid industrial environments, the use of autonomous robots has been increasingly present, but its use without planning can cause losses, decreased service levels, and even production failures.

In the scope of planning the use of robots in manufacturing environments, the collection or delivery of materials is the Assignment Problem, which consists of optimization models to perform the assignment of tasks to robots, with the aim to maximize the use of robots without having to worry about the order in which the assigned tasks are to be performed. The Scheduling Problem, which is knowing the assigned tasks, determines the sequence as to when these tasks are to be carried out by each operator, with the objective of minimizing, for example, the total time to complete the tasks (makespan). Some models are capable of jointly determining both the assignment and the sequencing [Fernandes and Tagliavilla 2021], but with a more complex model.

[Thomasson et al. 2022] presents an application in the context of the Twin-Robot Pallet Assignment and Scheduling Problem - TRPASP, in which two robots working together, initially positioned on opposite sides of a rail, must be programmed (Assignment + Scheduling) to perform a set of operations to collect and deliver products positioned at various points along the rail, and following a safe distance between the robots, with the goal of minimizing the total time of performing the operations. The authors propose an integer linear programming model to determine the assignment of tasks (jobs) to the robots, the sequence in which the tasks will be performed by the robots (scheduling), as well as the choice of pickup and delivery locations for the products on pallets. To handle large instances, the authors propose a meta-heuristic algorithm based on GRASP.

The objective of this work is to put forward an adaptation in the PLIM model considered in [Thomasson et al. 2022] and present a new solution method that uses a constructive heuristic with random components to obtain the scheduling of two twin robots in a picking environment. To validate the feasibility of the obtained solutions, an algorithm that constructs a geometric representation of the solution in a Cartesian plane is employed, indicating the position of the robots and task completion as a function of time. Specialized Constructive Heuristics for TRPASP (SCH-TRPASP).

2. Problem Definition

TRPASP consists in programming two robots, which are initially positioned on opposite sides of the rail, must perform the picking of the purchase order (orders) containing a set of different products (jobs) requested by customers, in which the quantity and types of products can vary according to the customer's order, and then must return to their initial positions. Each product type has pickup locations defined along one side of the rail, and delivery locations defined on the other side of the rail, as illustrated in Figure 1.

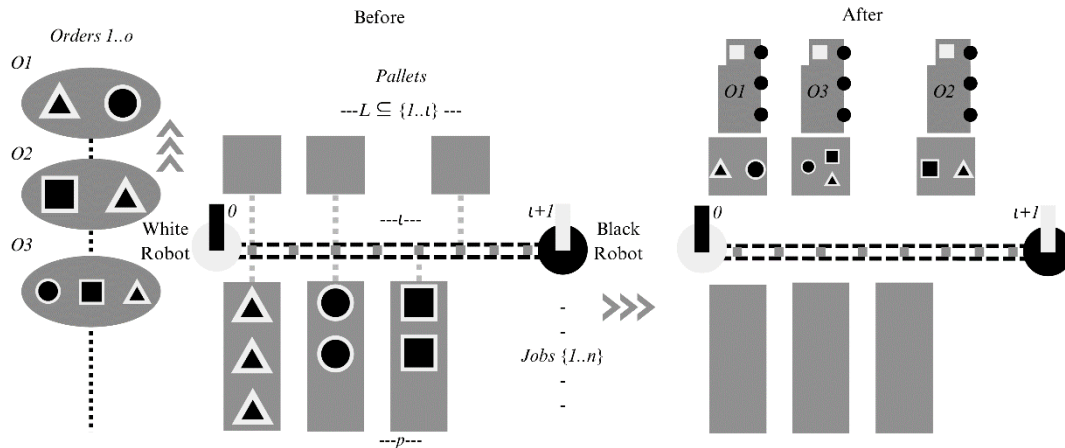
Thus, given a set of orders $o_1, \dots, o_m$, where each order contains a set of jobs $o_i = \{j_{1i}, j_{2i}, \dots, j_{ki}\}$ and given a trail of l locations, robot 1 (White) is positioned at location 0 and robot 2 (Black) at location $l + 1$. The solution to the problem is going to indicate which jobs are to be picked up by each robot, and in which delivery locations they are to be taken, in order to form the customers' orders, and indicate the best sequence of tasks to minimize the total time required to carry out all the orders (makespan), i.e., the performance of all activities (picking and delivery).

To ensure that there is no collision between the robots while performing the tasks, a safety distance parameter was considered, that is, the minimum distance required between the robots, given by positions of locations. Solutions that do not comply with this safety distance restriction are considered infeasible solutions.

To calculate the makespan, the travel time and the handling time are considered. The former represents the amount of time needed for the robot to travel from one location to another,

and the latter, represents the time needed to perform any activities, be it the pickup of a job from its pickup location or the delivery of a job to its delivery location.

Figure 1 - TRPASP Representation.



Source: Adapted from [Thomasson et al. 2022].

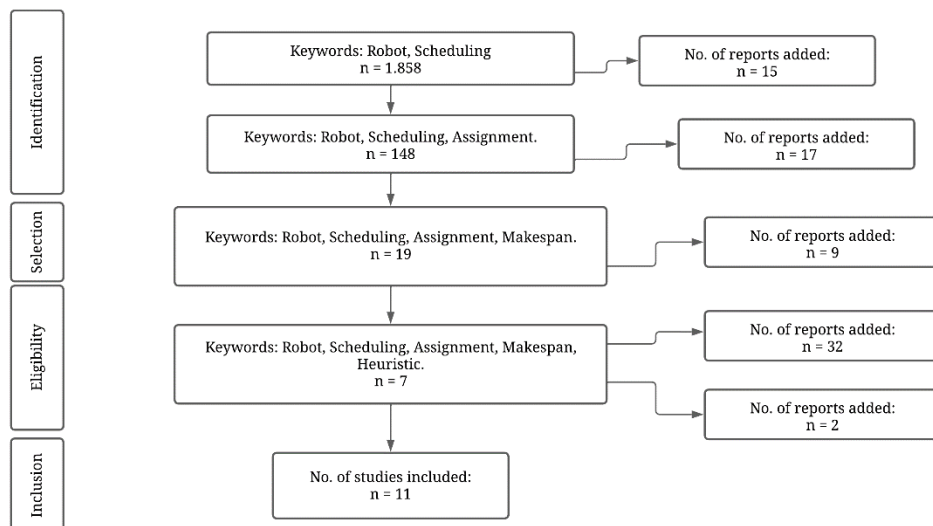
3. Materials and Methods

The PRISMA methodology [Page et al. 2021] was used for the bibliographic review of TRPASP, as summarized below.

3.1 PRISMA Systematic Review

The systematic review is an essential phase of the research methodology and seeks to obtain and critically evaluate all relevant literature on the topic under study.

Figure 2 - PRISMA Flowchart.



Source: Authors [2023].

To conduct the systematic review presented here, the literature review went through four sequential phases to examine recent studies involving the use of heuristic optimization methods in

the Scheduling Problem, focusing on the application of robot scheduling in manufacturing environments. With the adoption of TRPASP as the focal point of the research, the study also focused on the solution methods for the Assignment Problem.

The database chosen for the research was the CAPES periodical portal, from 2012 to 2022, using the keywords Robot, Scheduling, Assignment, Makespan and Heuristics. Figure 2 illustrates the summary of the results obtained from the research according to the PRISMA methodology.

The PRISMA methodology [Page et al. 2021] is a valuable tool for performing high-quality systematic collections. By following its guidelines, it is possible to ensure a complete and reliable revision of the Scheduling problems and the main solution methods currently employed, allowing a comprehensive and well-founded analysis of the subject.

PRISMA takes on a predetermined protocol that performs a systematic search for relevant articles, performs careful selection of suitable studies, extracts data in a standardized manner, and critically evaluates the quality of the included studies. Therefore, the articles added to this systematic review meet the inclusion and exclusion criteria pre-established in the search protocol.

3.1.1. Selected Surveys

As already mentioned, TRPASP was first introduced by [Thomasson et al. 2022]. The authors presented a PLIM model and Variable Neighbourhood Search (VNS) algorithms, Genetic Algorithm, Hungarian Algorithm and Hybrid Algorithm for pallet allocation which, based on the results, the last method showed excellent outcomes.

Another important basis for this research were the works presented in [Thomasson et al. 2018] and [Erdogan, Battarra and Laporte. 2014]. In the paper of 2018, the problem considered was the Twin-Robot Palletising Problem (TRPP), in which the significant difference identified when compared to TRPASP, is the fact that their orders are pre-allocated to the pallets. In the aforementioned paper, the authors presented two mathematical formulations and multiple heuristic methods to solve TRPP and having considered the data, the authors conclude that the hybrid genetic algorithm proved to be the most efficient to find an acceptable solution in a reasonable time.

TRPP is a more comprehensive version of the Twin-Robot Scheduling Problem (TRSP), previously presented by [Erdogan et al. 2014]. In this scheduling system, the pickup location of the jobs need not be considered since all items are introduced into the system at the warehouse and the robots handling them are predetermined. To solve the problem, two Mixed Integer Linear Programming models and two heuristics built specifically for this purpose were utilized.

The application of heuristic solutions using autonomous robots has expanded to several sectors, with path planning being one of the most studied. Authors such as Pattnaik, Mishra and Panda [2022], Kanagaraj, Sheik Masthan and Yu [2022], and Sahu, Das and Kabat [2022], used heuristics in a hybrid and comparative manner to find the best paths in their respective situations.

Wang, Liu, and Gombolay [2022] present a scheduleNet approach for solving a 5-robot scheduling with multi-task testing using a heterogeneous task completion. The results demonstrated computational efficiency and a more generic model for such cases.

Dang, Nguyen and Rudová [2019] put forward a further developed solution for scheduling in flexible manufacturing, based on a hybrid metaheuristic between Genetic Algorithm and Tabu Search. This research is an evolution of the previous work by the same author in [Dang et al. 2014].

In [Fernandes and Taglialenha 2021] a new approach to the previous work by Dang et al. [2019] was presented, with an improvement in the mathematical model used. As a result, it was possible to obtain a 20.00% reduction in the total travel time of the robot in the same example considered. It is noted that this addition was made from Portuguese keywords.

[Tereshchuk et al. 2021] brings a proposal for scheduling in aircraft assembly tasks involving multiple manipulator robots. The approach is based on allocating tasks to multiple robots with precedence constraints, thus bringing in Assignment elements. The proposed method is based

on machine learning along with a scheduling heuristic to automatically select the appropriate precedence relations.

3.1.2. Complexity

Bases on the analysis regarding complexity done by [Thomasson et al. 2022], very similar to the one carried out by [Erdogan et al. 2014] for TRSP, it was verified that TRPASP, can be reduced to a Partition problem, dividing it into smaller sets. By itself the Partition problem is known as an NP-Complete, but from the analyses built in the research, there is proof that TRPASP is an NP-Hard and that medium and large instances cannot be solved in an efficient time.

3.1.3 Solution methods

The solution methods used in TRPASP were studied and analyzed through the 11 selected articles with the application of PRISMA. As quantified in Table 1, a wide range of heuristic techniques were considered in the selected articles.

Table 1 - Number of Papers by Solution Method.

Heuristic Solution Method	Included Articles
Genetic Algorithm	40.00%
Taboo Search	10.00%
Hybrid algorithms	60.00%
Local search	20.00%
Branch-and-Bound	10.00%
PairMatch	10.00%
First Fit Decreasing	10.00%
Bat Algorithm	10.00%
Gravitational Search	10.00%
Particle Swarm	30.00%
Chemical Reaction	10.00%
Water drop	10.00%

Source: Authors [2023].

As seen in Table 1, the hybrid heuristic algorithms, resulting to 60.00% of the time, were the most utilized. These algorithms are results of using concepts from other existing methods, or simply modifying them to work in different contexts from the ones they were originally designed for. The approach may include combining different heuristic techniques or integrating other optimization techniques into the models. The data from [Thomasson et al. 2022], [Thomasson et al. 2018], and research in Path Planning, [Pattnaik et al. 2022], [Kanagaraj et al. 2022], and [Sahu et al. 2022], indicated promising results of improved computational performance by considering hybrid methods.

Next is the Genetic Algorithm, with 40.00%, coming from its wide range of possible problems to be solved with the model and its efficiency in working with multiple constraints and solutions together, considering that in TRPASP the Scheduling and Assignment issues need to be represented simultaneously within the solution, this algorithm shows itself to be a very favorable tool to be used.

4. Mathematical formulation

The following presents the nomenclatures, parameters and variables based on the model proposed by [Thomasson et al. 2022]:

- $J = \{1, \dots, n\}$ represents the set of jobs;

- $O = \{1, \dots, o\}$ represents the set of orders;
- $L \subseteq \{1, \dots, l\}$ represents the set of pallet locations (where orders can be assigned);
- o_i represents the order containing the job $i \in J$;
- p_i represents the pickup location of the job $i \in J$;
- η represents the handling time, the time required for one of the robots to pick up or drop off one of the products;
- σ represents the safe distance, the distance that must be kept between the robots for accidents not to occur;
- τ represents the travel time it takes for the robot to move from one location to the neighboring location;
- $\alpha_{ii'kk'}$ is the matrix with the minimum time values between the difference at the beginning of the job i and i' the safety distance σ to be maintained. The details are as follows: job i is carried out by Black going to location k and job i' is carried out by White going to location k' , with job i being done prior to i' ;
- $\beta_{ii'kk'}$ is the matrix with the minimum time values between the difference at the beginning of the job i and i' for the safety distance σ to be maintained. The details are as follows: job i is performed by Black going to location k and job i' is being run by White going to location k' , with the job i being performed after i' .

In this work, $\eta = \sigma = \tau = 1$ is considered. The matrices α and β were obtained from a specific algorithm so as not to affect the computational complexity, as stated in [Thomasson et al. 2022].

Variables:

- $x_{ij} = 1$, if job i is the j -th job to be done by White, and 0 otherwise;
- $y_{ij} = 1$, if job i is the j -th job to be done by Black, and 0 otherwise;
- $z_{ok} = 1$, if order o is assigned to the pallet k and 0 otherwise;
- t_j is the start time of the job organized to start at the j -th time;
- e_j is the end time of the job arranged to be started at j -th time;
- W is the makespan.

The TRPASP model is then given by Equations (1)-(19):

$$\text{Minimize } W \quad (1)$$

Subject to:

$$\sum_{o \in O} z_{ok} \leq 1 \quad k \in L \quad (2)$$

$$\sum_{k \in L} z_{ok} = 1 \quad o \in O \quad (3)$$

$$\sum_{i \in J} (x_{ij} + y_{ji}) = 1 \quad j \in J \quad (4)$$

$$\sum_{j \in J} (x_{ij} + y_{ij}) = 1 \quad i \in J \quad (5)$$

$$t_1 \geq \sum_{i \in J} \tau p_i x_{i1} + \sum_{i \in J} \tau (l + 1 - p_i) y_{i1} \quad (6)$$

$$e_j \geq t_j + (2\eta + \tau |p_i - k|)(x_{ij} + y_{ij} + z_{o_{ik}} - 1) \quad i, j \in J; k \in L \quad (7)$$

$$t_j \geq e_{j'} + \tau |p_i - k| z_{o_{i'k}} - M(2 - x_{ij} + x_{i'j'}) \quad i, i', j \in J; 1 \leq j' < j; k \in L \quad (8)$$

$$t_j \geq e_{j'} + \tau |p_i - k| z_{o_{i'k}} - M(2 - y_{ij} + y_{i'j'}) \quad i, i', j \in J; 1 \leq j' < j; k \in L \quad (9)$$

$$t_{j'} \geq t_j + \beta_{ii'kk'} + M(x_{ij} + y_{i'j'} + z_{o_{ik}} + z_{o_{i'k'}} - 4) \quad i, i', j \in J; 1 \leq j < j'; k, k' \in L \quad (10)$$

$$\begin{aligned}
 t_j &\geq t_{j'} + \alpha_{ii'kk'} + M(x_{ij} + y_{i'j'} + z_{o_{ik}} + z_{o_{i'k'}} - 4) & i, i', j \in J; 1 \leq j' < j; k, k' \in L & (11) \\
 w &\geq e_j + \tau \cdot k \cdot z_{o_{ik}} - M(1 - x_{ij}) & i, j \in J; k \in L & (12) \\
 w &\geq e_j + \tau(l + 1 - k)z_{o_{ik}} - M(1 - y_{ij}) & i, j \in J; k \in L & (13) \\
 x_{i,j} &\in \{0,1\} & i, j \in J & (14) \\
 y_{i,j} &\in \{0,1\} & i, j \in J & (15) \\
 z_{o,k} &\in \{0,1\} & k \in L; o \in O & (16) \\
 t_j &\geq 0 & j \in J & (17) \\
 e_j &\geq 0 & j \in J & (18) \\
 w &\geq 0 & & (19)
 \end{aligned}$$

Equation (1) defines the Objective Function for minimizing the makespan. The makespan is calculated in constraints (12) and (13). Constraint (2) guarantees that every location that has a pallet must have at most one order. Constraint (3) represents that every order must be linked to only one pallet location. Constraints (4) and (5) guarantee that each job must be performed solely by one of the robots and occupies only one position in the schedule.

Constraint (6) defines the start time of the first job, again emphasizing that the robots always start at the edges of the rail, so White is located at position 0 while Black starts at position $(l + 1)$. Constraint (7) defines the finishing times of all jobs, considering the starting time, pickup time, and delivery time of each job, and the travel time between locations.

Constraints (8) and (9) ensure that job i is assigned to one of the two robots, (8) for White and (9) for Black, and guarantees that the starting time of the respective robot is not less than the finishing time added to the travel time p_i of the respective robot.

Constraint (10) uses matrix β to ensure ample delay in the starting time of White's job to maintain the safe distance from Black. Constraint (11) uses matrix α to ensure ample delay in the starting time of Black's job to keep the safe distance from White.

Constraints (12) and (13) ensure that the makespan is not less than the completion time of any job plus the return times to the depots, where both robots must arrive at their depots. Constraints (14) to (19) explain the domain of the variables used.

AMPLIDE software had been used with the GUROBI solver to solve the model implemented in AMPL for small instances (up to $n = 4$ e $O = 1$). For slightly larger instances, the public NEOS-SERVER server was used, but with high processing time (up to $n = 7$ e $O = 3$).

4.1 Proposed Model

The model proposed in this work differs from the model proposed by [Thomasson et al. 2022]. In the original model, the authors defined in Constraint (6) only the starting time of the first job performed, that is, only the first job of one of the robots will be considered, thus bringing infeasibility to the problem. Taking into consideration that from the moment the first job is scheduled for one of the robots, any subsequent job n can be assigned as the other robot's first job. Consequently, any job time must satisfy the starting time constraint, a factor brought in by a new Constraint (20) which is used in the modeling of this research, thus replacing Constraint (6):

$$t_j \geq \sum_{i \in J} \tau p_i x_{ij} + \sum_{i \in J} \tau (l + 1 - p_i) y_{ij} \quad j \in J \quad (20)$$

5. Proposed Heuristic Method

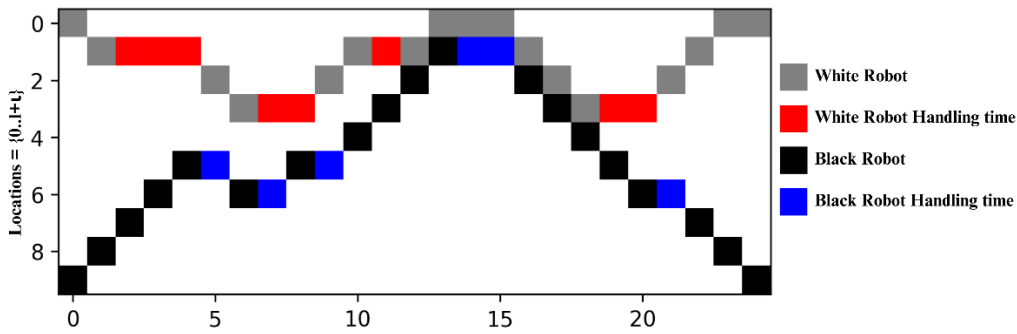
Since the mathematical model only solves small instances, a heuristic algorithm that determines feasible solutions in a low computational time has been suggested in this paper. This algorithm uses a geometric and vector-based solution creation approach to find the best solution for the problem at hand. The geometric approach allows a better visualization of a macro view of

the problem, while the vector approach gives way to understanding the Scheduling and Assignment procedures and in a micro view of the problem.

The SCH-TRPASP approach used in this algorithm allows it to generate solutions using random components divided into 3 vectors. S_i represents the schedule of the ordered jobs, thus the Scheduling factor of the problem. A_i is the Boolean vector that presents which of the robots is to carry out each job where 0 represents White and 1 Black. D_i is the destination vector of each job, where the destinations are the points wherein the pallet responsible for a given order is located. A_i and D_i will be the representation of the Assignment within the solution. All vectors are to have the same size n , where n is the number of jobs, making it easier to work with both in terms of operator visualization and computational performance.

Once a solution is obtained $[S_i, A_i, D_i]$ to visualize whether this solution is feasible, an algorithm is developed to determine a geometric representation of the solution, as illustrated in Figure 3, for the example presented in view of Figure 1, considering $n = 7$; $l = 8$; $L = (1, 3, 6)$; $o = 3$; $o_i = (1, 1, 2, 2, 3, 3, 3)$; $p_i = (1, 3, 5, 1, 3, 5, 1)$.

Figure 3 – Instance Geometric Representation



Source: Authors [2023].

The geometric representation has two axes: the x -axis represents the elapsed time and the y -axis represents the number of locations (l). In addition, the handling time visualization is represented by the colors red and blue, for robots White and Black consecutively. During the period in which the robot is performing the action of picking up or leaving a job, the corresponding color is displayed, making the representation more intuitive.

5.1 Algorithm Development

After defining the form in which the solution is to be represented, it is necessary to obtain the objective function (OF), the makespan. The model from random components, brings solution i where the value of i is varied between 1 to I , where I is the number of solutions generated for an instance. Created the solution, Algorithm 1 finds the OF and the development of the robots along the entire path, thus bringing the Geometric representation. As opposed to the solution methods proposed by [Thomasson et al. 2022], SCH-TRPASP does not require the use of the Alpha and Beta matrices, thus eliminating the need to process multiple different algorithms to obtain the solution for an instance.

Algorithm 1, to calculate objective value w , is executed until all jobs are completed, incrementing τ units of time (travel, handle and wait time).

To better represent the robot performances, a series of attributes to both White and Black robots were defined, as follow:

- $R = \text{Black, White}$;
- $Job_R = \{l_{R1}, l_{R2}, \dots, l_{Rk}\}$: Where l_{Ri} is the i -th job (chronological order) of robot R and k is the number of jobs assigned to robot R ;

Algorithm 1 Objective Function Calculation

```

01: Receive: White, Black,  $\sigma, \eta, l, \tau$ 
02:  $W = 0$ 
03: If * White.job/Black.job is empty:
04:   White.status/Black.status = -1
05: While there is 0 in White.complete or 0 in Black.complete:
06:    $W = W + \tau$ 
07:   If White.wait/Black.wait > 0:
08:     White.wantedlocation/Black.wantedlocation  $\leftarrow$  White.location/Black.location
09:     White.wait/Black.wait  $\leftarrow$  White.wait/Black.wait -  $\tau$ 
10:   If White.status/Black.status == 1:
11:     White.goal/Black.goal  $\leftarrow$  White.pallet[Next]/Black.pallet[Next]
12:   If White.status/Black.status == 0:
13:     White.goal/Black.goal  $\leftarrow$  White.joblocation[Next]/Black.joblocation[Next]
14:   If White.status/Black.status  $\neq$  -1 and White.wait/Black.wait  $\leq$  0:
15:     If White.location/Black.location < White.goal/Black.goal
16:       White.wantedlocation/Black.wantedlocation  $\leftarrow$  White.location/Black.location + 1
17:     Else:
18:       If White.location/Black.location > White.goal/Black.goal
19:         White.wantedlocation/Black.wantedlocation  $\leftarrow$  White.location/Black.location - 1
20:       Else If White.location/Black.location == White.goal/Black.goal:
21:         If White.status/Black.status == 1 and White.wait/Black.wait  $\leq$  0:
22:           White.status/Black.status  $\leftarrow$  0
23:           White.wait/Black.wait  $\leftarrow$   $\eta$ 
24:           White.complete[current]/Black.complete[current]  $\leftarrow$  1
25:         Else If White.status/Black.status == 0 and White.wait/Black.wait  $\leq$  0:
26:           White.status/Black.status  $\leftarrow$  1
27:           White.wait/Black.wait  $\leftarrow$   $\eta$ 
28:       If White.job/Black.job not have 0:
29:         White.status/Black.status  $\leftarrow$  -1
30:       If White.status/Black.status == -1 and White.wait/Black.wait  $\leq$  0:
31:         If White.location  $\neq$  0/Black.location  $\neq$   $l + 1$ :
32:           White.location/Black.location (Return)
33:           White.wantedlocation/Black.wantedlocation  $\leftarrow$  White.location - 1/Black.location + 1
34:       If White.wait  $\leq$  0 and Black.wait  $\leq$  0:
35:         If there is White.job and there is Black.job:
36:           If Black.wantedlocation - White.wantedlocation <  $\sigma$ :
37:             Prior  $\leftarrow$  Priority
38:             If Prior == 0:
39:               White.location  $\leftarrow$  White.wantedlocation
40:               If Black.location - White.location <  $\sigma$ :
41:                 Black.location(Return)
42:             If Prior == 1:
43:               Black.location  $\leftarrow$  Black.wantedlocation
44:               If Black.location - White.location <  $\sigma$ :
45:                 White.location(Return)
46:           Else:
47:             If White.status/Black.status  $\neq$  -1
48:               White.location/Black.location  $\leftarrow$  White.wantedlocation/Black.wantedlocation
49:           Else:
50:             If White.status/Black.status  $\neq$  -1
51:               White.location/Black.location  $\leftarrow$  White.wantedlocation/Black.wantedlocation
52:           Else:
53:             If White.status/Black.status  $\neq$  -1 and White.wait/Black.wait  $\leq$  0
54:               White.location/Black.location  $\leftarrow$  White.wantedlocation/Black.wantedlocation
55:             If Black.location - White.location <  $\sigma$ :
56:               If Black.wait/White.wait  $\leq$  0:
57:                 Black.location/White.location(Return)
58:             If White.location < 0 or Black.location >  $l + 1$ : Return:  $\infty$ 
59:       While White.location > 0 or Black.location <  $l + 1$ :
60:          $W = W + \tau$ 
61:         If White.location > 0:
62:           White.location(Return)
63:         If Black.location <  $l + 1$ :
64:           Black.location(Return)
65:       Return:  $W$ 

```

*: The notation "If A/B do C/D" means that we got both logical tests If A do C and if B do D.

- $Pallet_R = \{d_{R1}, d_{R2}, \dots, d_{Rk}\}$: List of the delivery locations d_{Ri} that have the pallet containing the order equivalent of job i of the robot R ;
- $Joblocation_R = \{jl_{R1}, jl_{R2}, \dots, jl_{Rk}\}$: List of the pickup locations jl_{Ri} that have job i of the robot R ;
- $Location_R$: Position that robot R is located (Input for Geometric solution);
- $Wantedlocation_R$: Possible location to the next movement of robot R ;
- $Goal_R$: Final location of R robot after traveling all wanted locations, either a pickup location or a delivery location;
- $Status_R$: 0 if R robot is empty, 1 if it is loaded, or -1 if there is no job assigned to be done;
- $Wait_R$: Being the waiting time for actions that demand the robot R to stop at a location;
- $Complete_R = \{c_{R1}, c_{R2}, \dots, c_{Rk}\}$: Boolean list c_{Ri} , that indicates if job i has been or not completed (pickup and delivery) by the R robot: 1 if yes; 0 otherwise;
- Next: Returning the next element in the list;
- Current: Returning the current element in the list;
- Return: Movement made by the robot to return a location, in case of White, $location \leftarrow location - 1$, in case of Black, $location \leftarrow location + 1$.

5.1.1 Function Priority

The function Priority, in line 37, is considered to deal with situations in that one of the robots wants to access a certain location that would violate the safety distance rule σ . In this function, it is checked which robot is performing the highest priority job, using the S_i list. So, the function returns 0 if White is developing a job with a lower index in S_i , or returns 1 if Black is developing a job with a lower index in S_i .

6. Results

Two main tests were implemented: the first with smaller instances (instances I1 to I3 Table 2), in which it had been possible to find the global optimum through the mathematical model, allowing the comparison of the effectiveness of the heuristic model in reaching the prime result.

Table 2 - Results For 1000 Replications Per Instance.

Instance Size				Computational Time (s)	Best OF	nBest	AvgDev
Instance	O	n	L				
I1	1	4	8	0.27	23	27	5.68
I2	2	5	8	0.32	20	10	5.4
I3	3	7	8	0.49	24	1	6.2
I4	5	20	10	1.37	102	3	16.65
I5	5	50	10	4.24	201	1	45.09
I6	5	100	10	11.36	422	1	71.05
I7	10	20	20	2.08	155	1	36.89
I8	10	50	20	6.42	384	1	65.08
I9	10	100	20	19.68	805	1	108.52

Source: Authors [2023].

The second test was carried out with larger instances (instances I4 to I9 Table 2), wherein the best OF value obtained is compared with the results derived from each iteration (average GAP). 1000 repetitions were also considered for each of the instances.

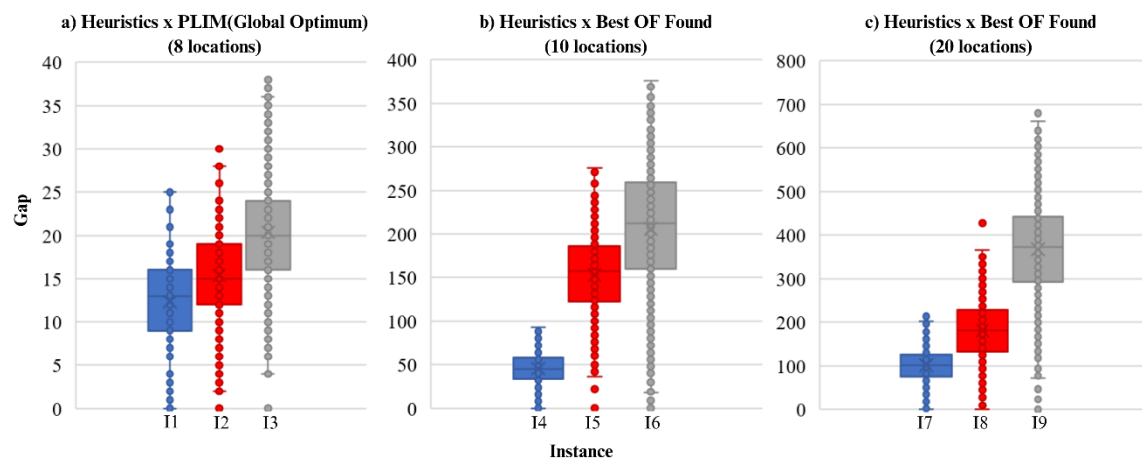
Table 2 shows the computational times, in seconds, for each instance, considering specifications: Intel(R) Core (TM) i5-10300H CPU @ 2.50GHz and 24.0 GB RAM, best OF found,

number of times the heuristic model arrived at the best known OF (nBest) and the standard deviation of all instances relative to the known OF (AvdDev).

Figure 4a. shows the results for the instances in which the mathematical model found the global optimum (8 locations), where for the first two, the heuristic was able to determine the optimal solution a considerable number of times. However, for the third one, the optimum was found only once. In relation to AvgDev, the three instances remained in a similar pattern, evidencing that, even with the optimal OF being more difficult to attain, the values obtained have a satisfactory proximity.

In Figure 4b., for instances with 5 orders and 10 locations each, the number of jobs used in [Thomasson et al. 2022] was considered to analyze the efficiency of the heuristics. For these instances, a marked increase in AvgDev is noticeable as the complexity of the problem increases with a larger number of jobs considered. The computational time presented a noticeable increase in relation to the smaller models, considering that the scale is measured in seconds, it is still assessed as an acceptable time for instances with a high number of jobs.

Figure 4 - Statistical plot of average error for different instances.



Source: Authors [2023].

For the instances illustrated in Figure 4c, despite the fact that the number of locations increased, the computational time did not present such a significant rise. It is improbable to make a direct comparison with the performance of the algorithms presented in [Thomasson et al. 2022], since the algorithm used is not characterized as a meta-heuristic.

7. Conclusions

This paper presents the results of the study of the Twin-Robot Pallet Assignment and Scheduling Problem - TRPASP, in which two robots are positioned on opposite sides of a rail to perform pickup/delivery operations. The problem consists in defining which products must be assigned to each robot, indicating the pickup point and the delivery point, as well as the sequence in which each robot will execute the operations, with the objective of minimizing the total time to perform the tasks, and complying with safety constraints to avoid collisions between the robots, and is considered an NP-hard combinatorial problem.

A PLIM model that determines the global optimal solution for small instances, and a heuristic approach capable of determining feasible solutions in satisfactory computational time for larger instances, have been put forward.

It is understood that the main contributions of this work attributed in presenting a way to verify the feasibility of the solutions obtained. The heuristic is validated for instances where the optimal solution is known. For the instances in which the optimal solution is unknown, the results gathered were measured as satisfactory.

For future works, it is recommended to use the heuristic proposed within this study to obtain initial solutions for re-optimization using meta-heuristics.

8. Acknowledgment

This article was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

- Dang, Q. V., Nielsen, I., Steger-Jensen, K., Madsen, O. (2014). Scheduling a single mobile robot for part-feeding tasks of production lines. *Journal of Intelligent Manufacturing*, v. 25, p. 1271–1287.
- Dang, Q.V., Nguyen, C T., Rudová, H. (2018). Scheduling of mobile robots for transportation and manufacturing tasks. *Journal of Heuristics*, v. 25, p. 175–213.
- Erdogan, G., M. Battarra, G. Laporte. (2014). Scheduling Twin Robots on a Line. In *Naval Research Logistics*, n. 61, p. 119-130.
- Fernandes, H. A., Tagliailenha, S. L. S. (2021). Sequenciamento de tarefas e agendamento de robôs móveis autônomos. *ENEGEP 2021, Encontro Nacional de Engenharia de Produção*, v.1, Foz do Iguaçu. ENEGEP.
- Kanagaraj, G., Sheik Masthan, S. e Yu, V. F. (2022). Meta-heuristics based inverse kinematics of robot manipulator's path tracking capability under joint limits. *Mendel*, v. 28, n. 1, p. 41-54.
- Page, M. J. et al. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BioMed Central*, v. 10, n. 1, p. 1–11.
- Pattnaik, S. K., Mishra, D., Panda, S. A. (2022). Comparative study of meta-heuristics for local path planning of a mobile robot. *Engineering Optimization*, v. 54, n. 1, p. 134-152.
- Sahu, B., Das, P. K., Kabat, M. R. (2022). Multi-robot co-operation for stick carrying application using hybridization of meta-heuristic algorithm. *Mathematics and Computers in Simulation*, v. 195, p. 197-226.
- Tereshchuk, V. et al. (2021). A Scheduling method for multi-robot assembly of aircraft structures with soft task precedence constraints. *Robotics and Computer-Integrated Manufacturing*, v. 71, ed. 102154, 10 p.
- Thomasson, O., Battarra, M., Erdoğan, G., Laporte, G. (2018). Scheduling twin robots in a palletising problem. *International Journal of Production Research*, v. 56(1-2), p. 518-542.
- Thomasson, O., Battarra, M., Erdoğan, G., Laporte, G. (2022). Pallet location and job Scheduling in a Twin-Robot system. *Computers & Operations Research*, v. 147, ed. 105956, 12 p.
- Wang, Z., Liu, C., Gombolay, M. (2022). Heterogeneous graph attention networks for scalable multi-robot Scheduling with temporospatial constraints. *Autonomous Robots*, v. 46, p. 249–268.