Pesquisa Operacional (2018) 38(1): 135-152
© 2018 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
doi: 10.1590/0101-7438.2018.038.01.0135

MINIMIZING THE PREPARATION TIME OF A TUBES MACHINE: EXACT SOLUTION AND HEURISTICS

Robinson S.V. Hoto[1], Glaucia M. Bressan[2*] and Marcos O. Rodrigues[3]

Received September 3, 2017 / Accepted March 8, 2018

ABSTRACT. In this paper we optimize the preparation time of a tubes machine. Tubes are hard tubes made by gluing strips of paper that are packed in paper reels, and some of them may be reused between the production of one and another tube. We present a mathematical model for the minimization of changing reels and movements and also implementations for the heuristics Nearest Neighbor, an improvement of a nearest neighbor (Best Nearest Neighbor), refinements of the Best Nearest Neighbor heuristic and a heuristic of permutation called Best Configuration using the IDE (integrated development environment) WxDev C++. The results obtained by simulations improve the one used by the company.

Keywords: Combinatorial optimization, Generalized traveling salesman problem, Tubes machine, Heuristic, Mixed integer programming.

1 INTRODUCTION

In recent decades, studies have been conducted to develop techniques to find satisfactory solutions for many different companies, which have realized the importance of planning an effective and intelligent design of their time and costs of production.

In this context, industries are trying to find ways to optimize their processes considering mathematical models and computational tools, in order to obtain the greatest possible results in their activities, like the quality of service, the increase in productivity, cost reduction and consequently increased profits [3]. In these situations, the problem is to decide the order of production, what product must be manufactured and its quantity and the relationship between costs and benefits for each product. Thus, the production planning influences significantly the quality of service and the cost for the whole process.

*Corresponding author.

[1] Universidade Estadual de Londrina, Departamento de Matemática, Paraná, Brazil. E-mail: hoto@uel.br

[2] Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio, Departamento Acadêmico de Matemática, Paraná, Brazil. E-mail: glauciabressan@utfpr.edu.br; galbressan@gmail.com

[3] Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Paulo, Brazil. E-mail: marokamura@gmail.com

The tubes machine problem, which is considered in this paper, can be mathematically interpreted as a Traveling Salesman Problem. A literature review that supports this approach is described as follows. Dantzig et al. [6] presented a mathematical formulation to the Traveling Salesman Problem. Particularly, it is shown that a certain tour of 49 cities, one in each of the 48 states and Washington, DC, has the shortest road distance. In Laporte [16], some of the main known algorithms for the Traveling Salesman Problem are surveyed. Lin & Kernighan [19] discuss a highly effective heuristic procedure for generating optimum and near-optimum solutions to the symmetric Traveling Salesman Problem, based on a general approach to heuristics that is believed to have wide applicability in combinatorial optimization problems.

Some new heuristics were developed in Golden et al. [9] to solve the Traveling Salesman Problem. They compared these techniques on the basis of efficiency and accuracy. Balas & Christofides [2] describe an algorithm for the asymmetric Traveling Salesman Problem using a new restricted Lagrangean relaxation based on the assignment problem. In this sense, Adranbinski & Suslom [1] present some computational experiments using several heuristic algorithms for solving the Traveling Salesman Problem.

Recently, heuristics and meta-heuristics have been performed to help the Traveling Salesman Problem resolution, since exact models are not well applied for big data.

The Generalized Traveling Salesman Problem was introduced by Henry-Labordere [11], which presented an application in relation to the optimal sequencing of computer files. An attempt to derive a recurrence relation to solve a Generalized Traveling Salesman Problem when a job can be completed at multiple places is presented by Srivastava et al. [22].

We can find applications of the Generalized Traveling Salesman Problem in programming processes of machines in industries, postal routing and layout of networks. Noon & Bean [21] employed a Lagrangian relaxation to compute a lower bound on the total cost of an optimal solution. Many studies about the Generalized Traveling Salesman Problem and their applications were done to obtain a reduction of this problem to a Traveling Salesman Problem, since there are many heuristics already performed for this problem. The first reduction method of a Generalized Traveling Salesman Problem to a Traveling Salesman Problem was presented by Lien et al. [18].

More recent, Mestria [20] proposes a hybrid heuristic algorithm to solve the Clustered Traveling Salesman Problem (CTSP). Hybrid Heuristic algorithm uses several variable neighborhood structures combining the intensification (using local search operators) and diversification (constructive heuristic and perturbation routine). Experimental results show that the proposed hybrid heuristic obtains competitive results within reasonable computational time.

Hernández-Pérez et al. [12] propose a hybrid heuristic approach for a pickup and delivery routing problem, which is a routing problem for a capacitated vehicle that has to serve a set of customers that provide or require certain amounts of different products. It is the first heuristic approach designed for this problem. Carrabs et al. [5] address a variant of the Euclidean traveling salesman problem in which the traveler visits a node if it passes through the neighborhood set of that node. This problem is named the close-enough traveling salesman problem. By reducing the discretization error, the new scheme allows to compute tighter upper and lower bounds for the problem.

According to Laporte et al. [17], a wide variety of combinatorial optimization problems can be modeled as Generalized Travelling Salesman Problem. These problems include location-routing problems, material flow system design, post-box collection, stochastic vehicle routing and arc routing. Considering this issue, this paper describes a case of study with the aim of optimizing the preparation time of a tubes machine. A Tube machine consists of 3 parts: Yard (Fig. 1), Tank of glue (Fig. 2) and Superpositioner (Fig. 3).

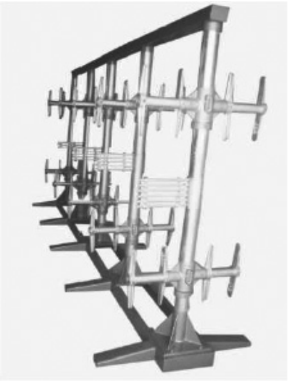

Figure 1 – Yard.

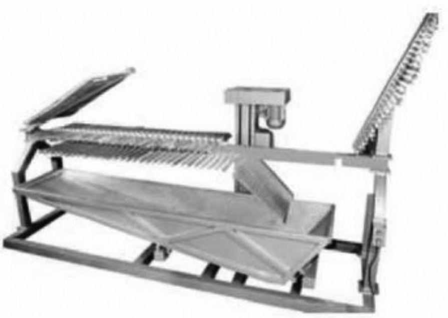

Figure 2 – Tank of glue.

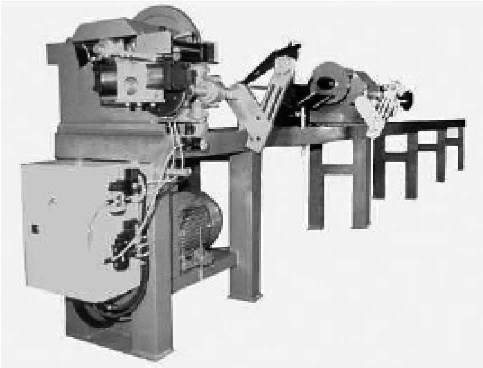

Figure 3 – Superpositioner.

The function of a tube machine is to make hard tubes which are called tubes (Fig. 4) by orderly superposition of multiple layers of recycled paper wrapped in strips of paper that are packed in paper reels (Fig. 5), glued on top of each other with a technically determined pressure.

The thickness and stiffness of the tubes depend mainly on the amount of layers and the type of paper used in its manufacture. Tubes are used as packaging for protection and conservation for products of other companies. They have a cylindrical shape, as can be seen in some examples illustrated in Figure 6; for this reason, tubes can be used as support to roll plastic films and

Figure 4 – Tube.

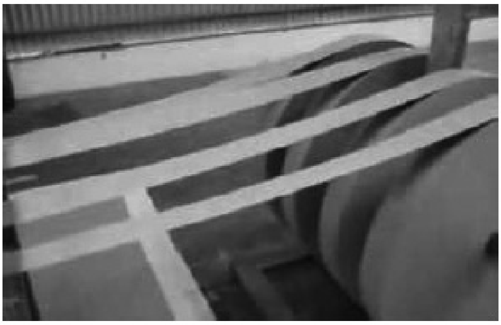

Figure 5 – Paper reels.

Figure 6 – Tubes packaging and Tubes supports.

tapes. The tubes are differentiated by the resistance to pressure, wall thickness, inner diameter and length, which are determined by customer needs.

The preparation of a tubes machine consumes too much time, since it involves the movement (removal and installation) of paper reels in the Yard. In order to reduce the preparation time of tubes machine, the problem is conceived as a Traveling Salesman and later as a Generalized Traveling Salesman, enabling the minimization of paper reels exchanges between the production of two arbitrary tubes. Therefore, the tubes to be produced must be efficiently sequenced.

Simulations using IBM ILOG Cplex 12.6 were run and the Best Nearest Neighbor, Bipartition and Tripartition heuristics were applied. Finally, a heuristic was developed specifically for the case study, named Best Configuration.

This paper is organized as follows. Section 2 describes the mathematical model for the studied problem, which is treated as a Generalized Traveling Salesman Problem. Section 3 describes the algorithms of the heuristics methods and their adaptations to solve the problem. Obtained results using real data from a company are presented and discussed in Section 4. Finally, Section 5 describes the final considerations about this paper.

2 MATHEMATICAL MODEL

The mathematical language is one way of describing a real situation abstractly. The importance of this description is to enable the simulation of situations that, in practice, could be damaging if executed, considering the costs involved in the process. In addition, not all real problems (actually most of them) have appreciable solutions, capable not only of being quickly tested but found.

It is difficult to find the solution to the problem discussed in this paper, since 34 paper reels can be stored in the Yard of the tubes machine, from which place will be unroll to tank of glue. In the company where this problem arises, about 10 to 20 different tubes, on average, are made in one week. Therefore, it is difficult for humans to decide which sequence will be most appropriate for reducing the number of paper reel exchanges, since many tubes have common paper reels that can be used in other processes.

The strips of paper are unrolled from the Yard to the tank of glue, which has a reservoir in its lower part, where the glue is deposited, then raised and poured onto strips of paper. Later, the strips of paper pass through pallets that remove the excess. The Superpositioner receives the strips of paper and overlaps them, on top of each other with an appropriate pressure, producing a tube whose length is “infinite”, from which the tubes are cut using a circular saw coupled to the superpositioner.

The production of a tube must obey, rigorously, the defined sequence of paper reel, in order to provide ideal physical conditions to the application. In addition, the tubes differ in relation to its internal diameter, which is defined by the mandrel of superpositioner.

In order to provide a better comprehension of the process, consider the production of 4 tubes, called A, B, C and D, in which paper reels coded 100, 101, 102 and 103 will make the nucleus of these tubes and the paper reels coded 200 and 220 will make both internal and external complement. Table 1 illustrates the configuration of the tubes.

Table 1 – Configurations of the paper reels.

Tubes			
A	B	C	D
200	200	200	200
100	100	101	101
102	101	102	102
103	103	220	103
220	220		220

Consider the production sequence (D, B, C, A) of these tubes, organized in Table 2.

Table 2 – A production sequence of the Table 1 tubes.

Tubes						
D		B		C		A
200	keep	200	keep	200	keep	200
101	change	100	change	101	change	100
102	change	101	change	102	keep	102
103	keep	103	change	220	change	103
220	keep	220	remove		insert	220

Between the production of one and another tube, some paper reels can be reused and others need to be removed or inserted to the proper functioning of the tube machine. In this study, the reposition time of the paper reels is not considered. In this described example, 16 movements (Insertion and Removal) were required, since “change” consists of 2 movements (1 insertion and 1 removal),“insert” and “remove” consist of 1 movement and “keep”, none. If, on the other hand, a different production sequence is selected, as (B, A, D, C), only 7 movements are required, as described in Table 3.

Table 3 – Another production sequence of the Table 1 tubes.

Tubes						
B		A		D		C
200	keep	200	keep	200	keep	200
100	keep	100	change	101	keep	101
101	change	102	keep	102	keep	102
103	keep	103	keep	103	change	220
220	keep	220	keep	220	remove	

One way to model these production sequences is to idealize each tube as a vertex of a complete graph [10], in which a Hamiltonian cycle of minimum cost is required; the number of movements between tubes is the cost per use on an edge. Therefore, an artificial vertex O has to be inserted; it will be the origin and the end of each sequence. Thus, a Traveling Salesman Model can be formulated for this problem.

The Yard is composed of 4 positions, where the paper reels are installed. They are called internals (I) or externals (E), as illustrated in Figure 8. For example, to change the internal paper reel number 2, four movements are required: two movements to remove the paper reels numbers 3 and 4, one movement to insert a new paper reel in the internal position and one more movement to re-insert the external paper reel number 4. Hoto et al. [13] associate only one movement with this action, but here, four movements are required since the situation is more realistic.

Considering paper reels movements in function of their positions in the yard, in order to promote changes, external paper reels require two movements, and the internal, four or two (if there is

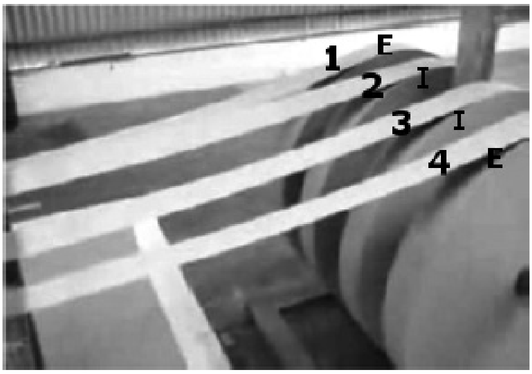

Figure 7 – Paper reels position.

no an external paper reel at its side). For technical reasons, a single empty position between paper reels is allowed in the Yard, if this is advantageous. In fact it is, since this increases the combinatorial of the problem. In the example given in Table 1, the configurations are those of Table 4.

Table 4 – Paper reels configurations.

Tubes									
A					B				
200	200	200	200	200	200	200	200	200	200
100	∅	100	100	100	100	∅	100	100	100
102	100	∅	102	102	101	100	∅	101	101
103	102	102	∅	103	103	101	101	∅	103
220	103	103	103	∅	220	103	103	103	∅
	220	220	220	220		220	220	220	220
C					D				
	200	200	200	200	200	200	200	200	200
	101	∅	101	101	101	∅	101	101	101
	102	101	∅	102	102	101	∅	102	102
	220	102	102	∅	103	102	102	∅	103
		220	220	220	220	103	103	103	∅
						220	220	220	220

This new approach leads to considering the case of the tube machine from the perspective of a Generalized Traveling Salesman Problem (GTSP), which is an extension of the classic Traveling Salesman Problem (TSP), in which the vertices are divided into groups called clusters, and aims to visit at least one vertex from each cluster, by means of a Hamiltonian cycle of minimal cost. According to Laporte et al. [17], the aim of the Generalized Travelling Salesman Problem is to determine a least cost Hamiltonian circuit or cycle through several clusters of vertices.

The Generalized Traveling Salesman Problem is defined as a valued graph $G = (V; E)$ where $V = 1, 2, \dots, n$ is the set of nodes and

$$E = (i, j) : i, j \in V, i \neq j$$

is the set of edges, and let c_{ij} be the travel distance (or cost or time) from node i to node j. Let $V_1, V_2, \dots, V_k$ be disjoint subsets of V such that union of these subsets equals to V. These subsets are called clusters.

The GTSP is to find the route that (i) starts from a node and visits exactly one node from each cluster and turns back to the starting node (ii) never visits a node more than once and (iii) has the minimum total route length.

In the formulation of Kara et al. [14], u_p is the auxiliary variable defined as the visit number of cluster p. In addition, we define one more auxiliary variable as:

$$w_{pq} = \begin{cases} 1, & \text{if traveler goes from cluster } p \text{ to cluster } q \\ 0, & \text{otherwise} \end{cases}$$

Each cluster $C_i; i = 1, \dots, n$ is made by possible configurations of the paper reels that compose a tube, as Table 4 illustrates. The set E is composed by edges that connect the vertex of the graph. A cost c_{ij} is associated with the edges; this cost represents the number of paper reels movements required to the tube j production immediately after tube i. For this approach, the intraclusters edges are not considered, since this fact would cause the production of the same tube two or more times. The goal is to define a production sequence of tubes that has the fewest number of paper reels movements, prioritizing the successive construction of tubes with the same mandrel, since the change of a mandrel spends much time in relation to the movement of a paper reel, so the tubes that have the same mandrel must be made in sequence. A tube is thus defined not only by the paper reels but by the mandrel used for its production as well. Consider:

x_{ij}: binary decision variable that indicates if the edge formed by the vertices i and j belongs to the solution, representing the production of tube j after tube i;

c_{ij}: cost to travel the edge from vertex i to j, represented by the exchange tube i for j. Note the cost is symmetric ($c_{ij} = c_{ji}$) and it is not defined for tubes of the same cluster;

m_{ij}: necessary cost to exchange the mandrel M_i for the M_j. Note the cost is symmetric ($m_{ij} = m_{ji}$).

In the mathematical model, n tubes are considered and the binary decision variable x_{ij}, where $i, j = 1, 2, \dots, n; i \neq j$ represent the tubes, such as:

$$x_{ij} = \begin{cases} 1, & \text{if tube } j \text{ is made imediately after tube } i \\ 0, & \text{otherwise} \end{cases}$$

Consider cost c_{ij}, which will be associated with edge $(i; j)$, representing the number of paper reels movements between tubes i and j production. The costs related to the artificial vertex satisfy

$c_{k0} = c_{0k} = 0$. Parameter m_{ij} is defined as one that is associated with the mandrel exchange cost:

$$m_{ij} = \begin{cases} 2B_{\max} + 3, & \text{if } M_i \neq M_j \\ 0, & \text{otherwise} \end{cases}$$

$B_{\max}$ is the maximum number of paper reels necessary to product a tube.

The mathematical model that represents the proposed problem, considering the paper reels movements, is based on *F1 model* [15], as follows.

$$\min \; z' = \sum_{(i,j)\in E} x_{ij}(c_{ij} + m_{ij}) \tag{1}$$

subject to:

$$\sum_{i\in V_p} \sum_{j\in V\setminus V_p} x_{ij} = 1, \quad \text{for } p = 1, 2, ..., k \tag{2}$$

$$\sum_{i\in V\setminus V_p} \sum_{j\in V_p} x_{ij} = 1, \quad \text{for } p = 1, 2, ..., k \tag{3}$$

$$\sum_{j\in V\setminus\{i\}} x_{ji} - \sum_{j\in V\setminus\{i\}} x_{ij} = 0, \quad \text{for all } i \in V \tag{4}$$

$$w_{pq} = \sum_{i\in V_p} \sum_{j\in V_q} x_{ij}; \qquad w_{pq} \in \{0, 1\}; \; p, q = 1, 2, ..., k \tag{5}$$

$$u_p - u_q + (k-1)w_{pq} + (k-3)w_{qp} \leq k-2, \tag{6}$$

$$p \neq q; \; p, q = 2, ..., k$$

$$u_p - \sum_{q;q\neq p} w_{qp} \geq 1, \quad p, q = 2, ..., k \tag{7}$$

$$u_p + (k-2)w_{1p} \leq k-1, \qquad p = 2, ..., k \tag{8}$$

$$x_{ij} \in \{0, 1\}, \qquad \text{for all } i, j \in E, i \neq j \tag{9}$$

In the objective function (1), the costs related to mandrel exchanges are summed together with the costs related to the paper reels movements, so that the tubes with the same mandrel can be produced in sequence. By applying the objective function (1), the cycle (route) of minimal cost to visit only one vertex from each cluster, with the lowest mandrel exchange, is obtained.

Constraints (2) and (3) are degree constraints of each cluster, whereas, constraints (4) balance the inflow and outflow of each node. These constraints were proposed by Noon & Bean (1991). Constraints (2) and (3) are imposed in order to guarantee that only one edge reaches each cluster and that only one edge out of each cluster.

Constraints (5) are expressions of the new variables by the previous decision variables. Constraints (6) are the subroute elimination constraints. Constraints (7) and (8) form the lower and upper bounds of the auxiliary variables. The validity of these constraints is shown in the propositions given below, which are proofed in Kara et al. (2012). Constraints (9) represent the integrality of variable x_{ij}.

Proposition 1: The equalities given in (6) guarantee that w_{pq} equals to 1 if the traveler goes from cluster p to q, and 0 otherwise.

Proposition 2: Inequalities given in (7), (8) and (9) are valid for the GTSP.

In order to obtain only the number of paper reels movements required to the production between two tubes i and j, it is sufficient the follow equation:

$$z = z' - \sum_{(i,j)\in E} x_{ij} m_{ij} = z' - (M-1)(2B_{\max} + 3),$$

Since the number of different mandrels is M, there will have $(M-1)$ mandrel exchanges and the cost related to each mandrel exchange is $(2B_{\max} + 3)$.

In the next section, the heuristics are presented, which aims to minimize the number of paper reels movements between tubes production.

3 HEURISTICS

In order to determine a viable solution to the tubes machine problem without the use of a solver, heuristics are applied, as described in this section, which are refinements of the Best Nearest Neighbor. In addition, the Best Configuration heuristic is performed, which permutes the tubes configurations after a viable solution is found. These heuristics were combined, so that the Bipartitioned Best Nearest Neighbor heuristic, Tripartitioned Best Nearest Neighbor heuristic, Best Nearest Neighbor with the Best Configuration heuristic, Bipartitioned Best Nearest Neighbor with the Best Configuration heuristic and Tripartitioned Best Nearest Neighbor with the Best Configuration heuristic were obtained.

The heuristics were implemented using C++ language, considering the empty positions in the tubes configurations. Moreover, a graphic interface was developed, using the integrated development environment (IDE) WxDev C++, which allows the user to view the available options to the portfolios to be analyzed.

The Nearest Neighbor heuristic (NN) is a Greedy constructive heuristic, originally presented in [4]. The basic idea is to choose randomly an initial vertex and then select the closest vertex that has not been visited, constructing the edge that will belong to the solution. The process terminates when all vertices are visited.

Unless the graph is complete, this procedure may have difficulties to find a route, even if there is one, since it may have vertices that are not connected by any edge. Even in complete graphs, this heuristic can provide a bad solution, since it is forced to choose very distant edges in the last steps. Another problem of this heuristic method is the fact that the first vertex is chosen arbitrarily, which can lead to a solution far from optimal.

The heuristic Best Nearest Neighbor (BNN) analyzes all of the possible routes constructed by NN heuristic. Considering each tube as initial from artificial vertex O, and choose the one with the lowest cost or the solution that return the lowest objective function value. The BNN heuristic run a NN considering each vertex i as the begin of the solution route.

Algorithm 1 shows the called Bipartitioned Best Nearest Neighbor Heuristic (BNN-2-part). In this algorithm, “sol” and “cost” are, respectively, the solution generator tree and the costs matrix; m and n are, respectively, the number of tubes and the number of configurations in each cluster. A cluster consists of different configurations of a tube and it is considered as a hiper-node.

Algorithm 1: Bipartitioned Best Nearest Neighbor Heuristic – BNN-2-part.

```
inputs: sol, cost, $m$, $n$
sum $\leftarrow \infty$
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        sum* $\leftarrow$ BNN(sol*, cost, $m$, $n$)
        while sum** $\leftarrow$ H2part(sol*, sum*, $n$) $<$ sum* then do
            sum* $\leftarrow$ sum**
        if sum* $<$ sum then do
            sum $\leftarrow$ sum*
            sol $\leftarrow$ sol*
return sum
```

In Algorithm 1, BNN is the classic Best Nearest Neighbor heuristic, applied to all the vertices $i = (p, c)$, considering each cluster p and each configuration c as the beginning of the solution route. The solution is updated, if necessary, in each iteration of the algorithm. H2part heuristic removes one edge of the solution route, creating 2 partitions, and it generates a new edge reconnecting 2 extremes vertices of the solution, without sub-routes. The procedure is performed in all positions where it is possible to partition the solution. The solution is updated only at the end of the algorithm with the best route found during the partitioning. Algorithm 2 shows the H2part heuristic.

Algorithm 2: H2part Heuristic.

```
inputs: sol, sum, $n$
sum** $\leftarrow$ sum
for $i \leftarrow 1$ to $n - 1$ do
    partition the solution removing the edge between $i$ and $i + 1$ vertices
    sol* $\leftarrow$ best solution among possible reconnections of the created partitions
    if sum* $<$ sum then do
        sum $\leftarrow$ sum*
        sol** $\leftarrow$ sol*
sum $\leftarrow$ sum**
sol $\leftarrow$ sol**
return sum
```

Similarly, Algorithm 3 shows the called Tripartitioned Best Nearest Neighbor Heuristic (BNN-3-part), a refinement to the solution obtained from the BNN heuristic. If there is a improvement, the solution is then updated.

Algorithm 3: Tripartitioned Best Nearest Neighbor Heuristic – BNN-3-part.

```
inputs: sol, cost, $m$, $n$
sum $\leftarrow \infty$
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        sum* $\leftarrow$ BNN(sol*, cost, $m$, $n$)
        while sum** $\leftarrow$ H3part(sol*, sum*, $n$) $<$ sum* then do
            sum* $\leftarrow$ sum**
        if sum* $<$ sum then do
            sum $\leftarrow$ sum*
            sol $\leftarrow$ sol*
return sum
```

The BNN-3-part uses the H3part heuristic in the sequence initially determined. This heuristic removes 2 edges of the solution route, creating 3 partitions, and it generates 2 edges reconnecting the extremes vertices of each partition, without sub-routes. Algorithm 4 shows the H3part heuristic.

In order to consider the intrinsic characteristics of the studied problem, the Best Configuration (BC) heuristic is performed. This heuristic analyzes if it is necessary to do exchanges in the configurations of the tubes, without the obtained sequence is altered.

Algorithm 4: H3part Heuristic.

```
inputs: sol, sum, $n$
sum** $\leftarrow$ sum
for $i \leftarrow 1$ to $n-2$ do
    for $j \leftarrow i+1$ to $n-1$ do
        Partition the solution removing the edge between $i$ and $i+1$ vertices and the edge between $j$ and $j+1$ vertices
        $sol*$ $\leftarrow$ best solution among possible reconnections of the created partitions
        if sum* $<$ sum then do
            sum $\leftarrow$ sum*
            sol** $\leftarrow$ sol*
sum $\leftarrow$ sum**
sol $\leftarrow$ sol**
return sum
```

The Best Configuration (BC) heuristic is exclusive to the problem that considers empty positions. For each vertex i of the solution, the heuristic analyzes if the configuration exchange c of the cluster optimizes the solution. If so, it updates the solution with the new configuration c'. Algorithm 5 describes this procedure.

Algorithm 5: Best Configuration Heuristic – BC.

```
inputs: sol, sum, $m$, $n$
sum $\leftarrow \infty$
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        create sol* replacing the node of the cluster $i$ of the solution by other node of the same cluster, but with configuration $j$
            if sum* $<$ sum then do
                sum $\leftarrow$ sum*
                sol $\leftarrow$ sol*
return sum
```

The Best Nearest Neighbor with the Best Configuration heuristic (BNNBC) is a refinement of the BNN heuristic. In each algorithm iteration, the BNN and BC heuristics are respectively applied, updating the solution if necessary, as described by Algorithm 6.

The Bipartitioned/Tripartitioned Best Nearest Neighbor with the Best Configuration heuristics (BNNBC-2-part/ BNNBC-3-part) are also a refinement of BNN heuristic. Such as the BNN-2-part/ BNN-3-part, after run the BNN heuristic for each vertex $i = (p, c)$ (in each configuration c of each cluster S_p), the H2part/H3part heuristics are applied to the found solution while there is cost reduction. Besides, the BC heuristic is applied in the same solution. The final solution is uploaded in each iteration that minimize the cost. These heuristics are described, respectively, in Algorithm 7 and Algorithm 8.

Algorithm 6: Best Nearest Neighbor with the Best Configuration Heuristic – BNNBC.

```
inputs: sol,cost,$m$, $n$
sum $\leftarrow \infty$
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        sum* $\leftarrow$ BNN (sol*,cost,$m$, $n$, $i$, $j$)
        sum* $\leftarrow$ BC (sol*,sum*,$m$, $n$)
        if sum* $<$ sum then do
            sum $\leftarrow$ sum*
            sol $\leftarrow$ sol*
return sum
```

Algorithm 7: Bipartitioned Best Nearest Neighbor with the Best Configuration Heuristic – BNNBC-2-part.

```
inputs: sol,cost,$m$, $n$
sum $\leftarrow \infty$
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        sum* $\leftarrow$ BNN (sol*,cost,$m$, $n$, $i$, $j$)
        while sum** $\leftarrow$ 2-part (sol*,sum*,$m$)$<$ sum* do
            sum* $\leftarrow$ sum**
        sum* $\leftarrow$ BC (sol*, sum*, $m$, $n$)
        if sum* $<$ sum then do
            sum $\leftarrow$ sum*
            sol $\leftarrow$ sol*
return sum
```

4 COMPUTATIONAL RESULTS

This section presents the results of the proposed model and all proposed heuristics described in Section 3. The experiments were run on a PC with Intel R Core TM i7-5500 @2.4 GHz processor and 16 GB RAM under Ubuntu 16.04 LTS 64 bits system. We solve the model using the commercial MIP solver IBM ILOG CPLEX 12.6 with default parameters with a time limit of 3.600 seconds.

Algorithm 8: Tripartitioned Best Nearest Neighbor with the Best Configuration Heuristic – BNNBC-3-part.

```
inputs: sol,cost,$m$, $n$
sum $\leftarrow \infty$
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        sum* $\leftarrow$ BNN (sol*,cost,$m$, $n$, $i$, $j$)
        while sum** $\leftarrow$ 3-part (sol*,sum*,$m$)$<$ sum* do
            sum* $\leftarrow$ sum**
        sum* $\leftarrow$ BC (sol*, sum*, $m$, $n$)
        if sum* $<$ sum then do
            sum $\leftarrow$ sum*
            sol $\leftarrow$ sol*
return sum
```

The data used to produce the results presented below are real data provided by the company. The portfolios consist of tubes of real data relating to the planning horizon of one week. Thus, the portfolios are constituted by different numbers of tubes, which also have different configurations for its production. In order to illustrate the Mathematical Model, Section 2 presents data relating to only one portfolio, which contains 4 tubes (A, B, C and D). The maximum number of reels per tube of this portfolio is 5, the number of mandrels is 1 and the number of vértices of the graph (considering one empty position) is 19. In this current section, Table 5 presents general characteristics about 7 portfolios – or 63 tubes. The "tube" column describes the total number of tubes contained in each portfolio and the "reels" column presents the maximum number of reels per tube (no tube has less than 80% of the maximum number of paper reels). Last column describes the number of mandrels. Thus, the biggest portfolio has 31 paper reels and the smallest portfolio has 10. We do not present details of these portfolios, as shown in Section 2, due to the large amount of data.

Table 5 – Portfolios characteristics.

Portfolio	Tubes	Reels	Mandrels
1	10	10	1
2	7	23	5
3	6	27	5
4	8	25	5
5	6	23	5
6	10	31	8
7	16	26	9

Table 6 presents, for each portfolio, the results obtained from the run heuristics, described in Section 3. These results provide the number of paper reels movements required to produce the portfolio. The last column presentes the solution (number of paper reels movements) used by the company. The best results is in bold and BNNBC-3-part heuristic provides the best performance for all portfolios.

Table 6 – The number of paper reels movements obtained from the heuristics.

Portfolio	NN	BNN	BNNBC	BNN 2-part	BNN 3-part	BNNBC 2-part	BNNBC 3-part	Company
1	91	88	84	86	86	84	**83**	116
2	116	**104**	**104**	**104**	**104**	**104**	**104**	122
3	148	**124**	**124**	**124**	**124**	**124**	**124**	166
4	170	151	151	**146**	**146**	**146**	**146**	170
5	164	164	160	158	**154**	**154**	**154**	178
6	238	234	234	232	232	232	**231**	313
7	311	302	292	302	300	292	**290**	370

The computational results are shown in Table 7. For each portfolio, Table 7 reports, respectively, the number of vertices of the graph, the objective value obtained by BNNBC-3-part heuristic (that has presented the best performance), the optimum value (obtained by Mixed Integer Programming model), the value obtained by the company and the reduction index (in percentage) of the company solution in comparison to the optimum solution.

Table 7 – Comparison between BNNBC-3-part heuristic and company performances.

Portfolio	Vertices	BNNBC-3-part	Optimum	Company	Reduction (%)
1	80	83	79	116	31.9
2	123	104	104	122	14.8
3	155	124	124	166	25.3
4	133	146	146	170	14.1
5	105	154	154	178	13.5
6	206	231	230	313	26.5
7	248	290	290	370	21.6

BNNBC-3-part heuristic obtained the optimal value for almost all the evaluated instances, except for instances A and F, in less than one second of execution time. On the other hand, the model obtained the optimal solution for all portfolios in less than six minutes. Furthermore, the proposed model can reduce from 13.5% up to 31.9% the movement cost of the company.

5 CONCLUSION

In this paper, the preparation time of a tubes machine was optimized. A mathematical model for the minimization of changing reels and movements was executed and numerical results, obtained from heuristics simulations, have improved the company practices.

The company is a manufacturer, which has only human material, without automation of production processes. Therefore, the results obtained in this study provided a considerable impact on the production system of the company, presenting a substantial gain in production processes, making it more efficient and competitive.

The company was pleased with the obtained results. For best employment and use of these results, an application was developed to be incorporated into the production of the company, significantly reducing the movement cost. Thus, the study of optimization models for the control and production planning becomes an essential tool for industrial advancement.

ACKNOWLEDGMENTS

This work received financial support from CNPq (National Council for Scientific and Technological Development) and FICO optimization (company).

REFERENCES

[1] ADRANBINSKI A & SUSLOM MM. 1983. Computational Experiments with Some Approximation Algorithms for the Traveling Salesman Problem. *Applicationes Mathematicae*, **18**(1): 91–95.

[2] BALAS E & CHRISTOFIDES NA. 1981. A Restricted Lagrangean Approach to the Traveling Salesman Problem. *Mathematical Programming*, **21**(1): 19–46.

[3] BAZARAA MS, JARVIS JJ & SHERALI HD. 2010. *Linear Programming and Network Flows*. Wiley, 4th Edition.

[4] BELLMORE M & NEMHAUSER G. 1958. The Traveling Salesman Problem: A survey. *Operations Research*, **16**(3): 538–558.

[5] CARRABS F, CERRONE C, CERULLI R & GAUDIOSO M. 2017. A novel discretization scheme for the close enough traveling salesman problem. *Computers and Operations Research*, **78**: 163–171.

[6] DANTZIG G, FULKERSON R & JOHNSON S. 1954. Solution of a large-scale traveling salesman problem. *Journal of the operations research society of America*, **2**(4): 393–410.

[7] DASH OPTIMIZATION. 2000. Applications Of Optimization With XpressMP, Translation to English from Programmation Linéaire de C. Guéret, C. Prins E M. Sevaux, Dash Optimization Ltda.

[8] FENATO AJ. 2008. *A model of generalized traveling salesman to minimize the preparation time of a tube machine.* Electrical Engineering Master, State University of Londrina, Brazil.

[9] GOLDEN BL, BODIN L, DOYLE T & STEWART JR W. 1980. Approximate Traveling Salesman Algorithms. *Operations Research*, **28**: 694–711.

[10] GONDRAN M & MINOUX M. 1984. *Graphs and Algorithms*. Wiley-Interscience.

[11] HENRY-LABORDERE AL. 1969. The record balancing problem: a dynamic programming solution of a generalized traveling salesman problem. *Revue Francaise d'Informatique de Recherche Operationnelle*, **3**(NB2): 43–49.

[12] HERNÁNDEZ-PÉREZ H, RODRIGUEZ-MARTIN I & SALAZAR-GONZÁLEZ JJ. 2016. A hybrid heuristic approach for the multi-commodity pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, **251**(1): 44–52.

[13] HOTO RSV, BORSSOI AH, MACULAN N & FENATO A. 2010. Reducing the Setup of a Tubes Machine. *IEEE Latin America*, **8**(1): 101–106.

[14] KARA I, DERYA T, DEMIR E & BEKTAS T. 2005. An Integer Programming Formulation for Generalized Traveling Salesman Problem (In Turkish), YA/EM 2005, Istanbul.

[15] KARA I, GUDEN H & KOC ON. 2012. New Formulations for the Generalized Traveling Salesman Problem. In Proceedings of the *6th International Conference on Applied Mathematics, Simulation, Modelling, ASM'12*, pages 60-65, Stevens Point, Wisconsin, USA, 2012. World Scientific and Engineering Academy and Society (WSEAS). http://dl.acm.org/citation. cfm?id=2209505.2209517.

[16] LAPORTE G. 1992. The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms. *European Journal of Operational Research*, **59**: 231–247.

[17] LAPORTE G, ASEF-VAZIRI A & SRISKANDARAJAH C. 1996. Some Applications of the Generalized Traveling Salesman Problem. *Journal of the Operational Research Society*, **47**: 1461–1467.

[18] LIEN Y, MA E & WAH BW. 1993. Transformation of the Generalized Traveling Salesman Problem into the Standard Traveling Salesman Problem. *Information Sciences*, **74**(1-2): 177–189.

[19] LIN S & KERNIGHAN BW. 1973. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, **21**(2): 498–516.

[20] MESTRIA M. 2016. A Hybrid Heuristic Algorithm for the Clustered Traveling Salesman Problem. *Pesquisa Operacional*, **36**(1): 113–132.

[21] NOON CE, BEAN JCA. 1991. Lagrangean Based Approach for the Asymmetric Generalized Traveling Salesman Problem. *Operations Research*, **39**(4): 623–632.

[22] SRIVASTAVA SS, KUMAR S, GARG RC & SEN P. 1969. Generalized traveling salesman problem through *n* sets of nodes. *CORS Journal*, **7**(2): 97.