Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-19 (2024) 854–857

Minimizing the number of mail sorting sessions as a variant of vector bin-packing: A case study at La Poste

Emmanuelle Amann *,** **Evgeny Gurevsky** **
Arnaud Laurent ** **Nasser Mebarki** **

* *La Poste, Nantes, France (e-mail: emmanuelle.amann@laposte.fr)*
** *LS2N, Université de Nantes, France*

Abstract: This article deals with an optimization problem arising in industrial mail platforms. To increase the profitability of sorting machines, it is necessary to minimize the number of mail sorting sessions processed on these machines. This problem can be seen as a variant of the vector bin-packing problem, with a particular structure of the vector items. In this paper, we propose a 0-1 linear programming model to deal with small- and medium-size instances of this problem. It demonstrated good performance for certain instance categories. Computational results are reported as well.

Keywords: Mail organization, vector bin-packing, 0-1 LP formulation.

1. INTRODUCTION

In recent years, the French postal distribution sector has seen a significant reduction in its mail activity, with a drop from 9 billion mail items delivered in 2018 to 5 billion forecast for 2025. Faced with these changes, postal companies need to evolve their current mail sorting organization.

In France, at La Poste, mail is delivered by factors' routes, *i.e.* a given order of addresses to be delivered. To facilitate these routes, the mail assigned to a route is sorted, prior to distribution, in the order in which the addresses appear geographically. Every day, several hundred routes are prepared on sorting machines located in one of the twenty industrial mail platforms. The outputs of these sorting machines are limited in number and maximum capacity. As the mail arrives in a disorderly manner, it is prepared in two stages. In the first stage, mail assigned to the same route is separated into different outputs according to its rank on the route, and stored in containers. In the second stage, the containers are fed back into the machine in a precise order, so that at the end they are filled with the ordered mail from the same route. In order to prepare for these stages in the best possible way, routes are grouped into batches, also called as *sessions*, which are then sorted in two stages. At present, La Poste wrongly assumes that each address on a route receives mail every day. The same batches are therefore made up daily. As a result, the maximum capacity of the containers during sorting is not optimally used.

The problem of forming these route batches can be seen, among other things, as a vector variant of the well-known problem of arranging objects in a minimum number of boxes, more often called as *vector bin-packing* (hereafter VBP) (see, *e.g.*, Johnson, 2016). Thus, in the context of La Poste, routes and batches, seen as vectors with non-negative components, respectively play the role of objects and boxes, which have the same dimension equal to the number of outputs of the sorting machine. Moreover, the same capacity (that of the sorting machine's outputs) not to be exceeded is set for each batch dimension. The challenge is therefore to find a way of arranging all routes into a minimum number of batches, while respecting the capacity constraints for each dimension within these batches.

The presented problem differs from the classical VBP (see, *e.g.*, Gabay and Zaourar, 2015) by the particular structure of the vector objects to be arranged, which represent routes. Since the amount of mail to be delivered can vary from day to day, many vectors can contain null components. Thus, each null component corresponds to the absence of mail for a given address on a given route on that day. Furthermore, it is important to note that any assignment of non-zero size mail to the sorting machine's outputs within the same route that satisfies the address precedence constraints between these mails does not affect the two-stage sorting approach. It is this last property, called as *spacing*, which adds flexibility to the allocation of routes by batch, and which can therefore be very beneficial when it comes to minimizing their number. To the best of our knowledge, this variant of VBP has never been considered before in the literature.

In order to help La Poste adapt its mail sorting organization to actual postal traffic and render the constitution of its sorting sessions more efficient in everyday life, we address in this paper a new variant of VBP described above. The rest of the paper is organized as follows. The principal motivation and some illustrative examples are given in Section 2. A 0-1 linear programming (LP) formulation, incorporating the spacing property, is presented in Section 3. Section 4 is dedicated to reporting the numerical results of the 0-1 LP model on instances close to industrial

10.1016/j.ifacol.2024.09.182

ones. Finally, conclusion and perspectives are provided in Section 5.

2. MOTIVATION AND EXAMPLES

To simplify the visualization of mail distribution in the two sorting stages, we use the table format. Thus, in Table 1, we present a dummy example of a possible arrangement of eight routes into three sorting sessions for the five-output sorting machine, where all the mail in the respective routes is initially aligned to the left. Here, each line of the session (with the exception of the last) represents a route with an expected order of mail at the end of sorting. The last line of each session, meanwhile, indicates the actual total volume of mail to be processed by each of the five outputs, whose maximum capacity is set at five units. Addresses receiving no mail are indicated by an empty cell. In turn, Table 2 shows a better possible arrangement of the eight routes into just two sessions, taking advantage of the spacing property of these routes, mentioned above.

	Route	Out. 1	Out. 2	Out. 3	Out. 4	Out. 5
Ses. 1	1	1	1	1	1	1
	2	2	1	2		
	3	2	2	1		
	$\sum$	**5**	**4**	**4**	**1**	**1**
Ses. 2	4	1	2	2		
	5	1	3	1		
	$\sum$	**2**	**5**	**3**	**0**	**0**
Ses. 3	6	1	1	2		
	7	2	1			
	8	1	1	1	1	1
	$\sum$	**4**	**3**	**3**	**1**	**1**

Table 1. An arrangement of 8 routes into 3 sessions

	Route	Out. 1	Out. 2	Out. 3	Out. 4	Out. 5
Ses. 1	1	1	1	1	1	1
	2	2	1	2		
	3	2	2	1		
	6		1	1	2	
	7				2	1
	$\sum$	**5**	**5**	**5**	**5**	**2**
Ses. 2	4	1	2	2		
	5	1			3	1
	8	1	1	1	1	1
	$\sum$	**3**	**3**	**3**	**4**	**2**

Table 2. An arrangement of 8 routes into 2 sessions

3. 0–1 LP FORMULATION

This section presents a 0-1 LP formulation for the studied problem. We introduce some useful notations, describe the binary decision variables, and explain the necessary statements for the objective function and any constraint.

Notations:

- $R = \{1, \ldots, |R|\}$ is the set of all routes;
- $M^{(r)} = \{1, \ldots, |M^{(r)}|\}$ is the set of mails for the route $r \in R$;
- $v_j^{(r)}$ is the size of the j-th mail in the route $r \in R$. Here, $j \in M^{(r)}$;
- $O = \{1, 2, \ldots, |O|\}$ is the set of available machine outputs;
- $O_j^{(r)} = \left\{j, j+1, \ldots, |O| - |M^{(r)}| + j\right\}$ is the interval of potential outputs for the j-th mail in the route $r \in R$;
- $U_k^{(r)} = \{j \in M^{(r)} : k \in O_j^{(r)}\}$ is the set of mails of round r, which can be potentially assigned to the output $k \in O$;
- $L_{\max}$ is the maximal capacity per output for any sorting session;
- $S = \{1 \ldots, N\}$ is the set of sorting sessions, where N is an upper bound on its number.

Binary variables:

- $x_{jk}^{(r)}$ is equal to 1 if the j-th mail of round r is assigned to the output $k \in O_j^{(r)}$, 0 otherwise;
- $y_s^{(r)}$ is equal to 1 if route r is located to session s, 0 otherwise;
- z_s is equal to 1 if at least one route is located to session s, 0 otherwise;
- $\theta_{jks}^{(r)}$ is equal to 1 if route r is allocated to session s and the j-th mail of round r is assigned to the output $k \in O_j^{(r)}$, 0 otherwise.

Model:

$$\min \quad f := \sum_{s=1}^{N} z_s \tag{1}$$

subject to:

$$\sum_{s=1}^{N} y_s^{(r)} = 1, \quad \forall r \in R \tag{2}$$

$$\sum_{r \in R} y_s^{(r)} \leq |O| \cdot z_s, \quad \forall s \in S \tag{3}$$

$$\sum_{k \in O_j^{(r)}} x_{jk}^{(r)} = 1, \quad \forall j \in M^{(r)}, \quad \forall r \in R \tag{4}$$

$$\sum_{j \in U_k^{(r)}} x_{jk}^{(r)} \leq 1, \quad \forall k \in O, \quad \forall r \in R \tag{5}$$

$$x_{jk}^{(r)} + y_s^{(r)} \leq \theta_{jks}^{(r)} + 1, \quad \forall k \in O_j^{(r)}, \quad \forall j \in M^{(r)}, \quad \forall r \in R, \quad \forall s \in S \tag{6}$$

$$1 + \sum_{k \in O_j^{(r)}} k \cdot x_{jk}^{(r)} \leq \sum_{k \in O_{j+1}^{(r)}} k \cdot x_{j+1,k}^{(r)}, \quad \forall j \in M^{(r)} \setminus |M^{(r)}|, \quad \forall r \in R \tag{7}$$

$$\sum_{r \in R} \sum_{j \in U_k^{(r)}} v_j^{(r)} \cdot \theta_{jks}^{(r)} \leq L_{\max} \cdot z_s, \quad \forall k \in O, \quad \forall s \in S \tag{8}$$

$$z_{s+1} \leq z_s, \quad \forall s \in S \setminus \{N\} \tag{9}$$

$$z_s \in \{0, 1\}, \quad \forall s \in S$$

$$y_s^{(r)} \in \{0, 1\}, \quad \forall r \in R, \quad \forall s \in S$$

$$x_{jk}^{(r)} \in \{0, 1\}, \quad \forall k \in O_j^{(r)}, \quad \forall j \in M^{(r)}, \quad \forall r \in R$$

$$\theta^{(r)}_{jks} \in \{0,1\}, \quad \forall j \in M^{(r)}, \quad \forall k \in O^{(r)}_j, \quad \forall r \in R, \quad \forall s \in S$$

The principal goal of the objective function (1) is to find a solution minimizing the number of used sessions. Equalities (2) force each route to be allocated to exactly one session. The fact that the number of allocated routes per session can not be greater than the number of machine outputs is expressed by (3). Constraints (4) require that each mail j of route r be assigned to exactly one output. Inequalities (5) show that, for a given route, there can be at most one mail per output. Constraints (6) provide a link between the variables x, y and θ. The precedence mail constraints for each route r are modelled by (7). Constraints (8) express the total volume for each output of each session, which can not be greater than $L_{\max}$ (resp. 0), if the session is considered as not empty (resp. empty). Expressions (9) are used to avoid symmetry in this problem by preventing the next session from opening when the previous one is still empty. Note that a trivial lower bound on the number of used sessions can be computed as $\left\lceil \frac{|R|}{|O|} \right\rceil$.

4. NUMERICAL EXPERIMENTS

This section first describes the instances used and the input data for the numerical experiments. The computational results for the 0-1 LP model are then presented and analyzed.

The 0-1 LP model was implemented using the solver Gurobi 10.0.3. The experiments were conducted on a computer having a 2.80 GHz Intel® Core™ i7-1165G7 processor with 16 GB RAM.

A time limit of 150 seconds per instance was fixed for solving the 0-1 LP model by the Gurobi solver. The parameter $L_{\max}$ was set to $\max\{v_{\max}, \lambda \cdot r_{\max}\}$, where $v_{\max}$ is the size of the biggest mail among all the routes, $r_{\max} = \max_{r\in R} \sum_{j\in M^{(r)}} v^{(r)}_j$ is the size of the biggest route and $\lambda \in [0, 1]$ is a real-valued control parameter.

The parameter N was fixed heuristically by adapting the first-fit strategy (see, *e.g.*, Dósa and Sgall, 2013) to our problem, without seeking the best allocation of routes in each session. This also enabled us to obtain an initial heuristic solution for each instance, which was then used as a warm-start one for the respective 0-1 LP model.

The 0-1 LP model was examined on 450 randomly generated instances, close to realistic postal traffic and organized into three categories regarding to the number of sorting machine outputs, *i.e.* $|O| \in \{20, 40, 80\}$. Each of these categories is then divided into five series according to the total number of routes per instance, *i.e.* $|R| \in \{40, 80, 120, 160, 180\}$. Finally, each series consists of three sets of 10 instances each, with respect to $\lambda \in \{0.25, 0.50, 0.75\}$.

The computational results for the randomly generated instances are presented in Table 3. Each row of this table corresponds to a particular set of instances. The first three columns indicate the values for $|O|$, $|R|$ and λ, respectively. The forth column reports the number of instances solved to optimality by the 0-1 LP model for each set. The fifth and sixth columns show the average number of sessions per set, found by the first-fit strategy and the 0-1 LP model, respectively. The seventh (resp. eighth) column specifies the average `GAP` (resp. `CPU` time), provided by the 0-1 LP model for each set of instances.

First of all, it is interesting to note that practically all warm-start solutions were improved in average by the 0-1 LP model. However, the number of instances solved to optimality per category decreases rapidly as the number of machine outputs increases: 92% for the first category, 37.33% for the second category and only 26.67% for the last one. Another observation is that smaller values of λ lead to instances that are more difficult to solve. Thus, 25.33% of instances were solved to optimality for $\lambda = 0.25$, 51.33% of instances were solved to optimality for $\lambda = 0.5$, and finally 79.33% of instances were solved to optimality for $\lambda = 0.75$.

5. CONCLUSION AND PERSPECTIVES

In this paper, a new industrial optimization problem was introduced. This latter is related to one of the important steps of the mail sorting process at the main French mail delivery provider La Poste. The problem consists in organizing a given set of factor routes into a minimal number of mail sorting sessions. It was first shown that such a problem can be considered as a particular variant (not yet treated in the literature) of the vector bin-packing problem with a specific structure of vector items.

A 0-1 LP model was proposed for handling this problem. Three categories of randomly generated instances were used to test the model. Only small-size instances have been successfully treated by the model.

In order to be able to tackle large-size instances, we intend to develop approximate methods which could be based on smoothing techniques, among others. The latter have already proved their worth in our previous work on the problem of load balancing on the sorting machine outputs to facilitate the work of operators at the end of the first sorting stage (see Amann et al., 2023).

REFERENCES

Amann, E., Gurevsky, E., Laurent, A., and Mebarki, N. (2023). Mail sorting optimization : a case study of the French postal company La Poste. In *Proceedings of the 22nd IFAC World Congress*.

Dósa, G. and Sgall, J. (2013). First fit bin packing: A tight analysis. In N. Portier and T. Wilke (eds.), *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics*, 538–549.

Gabay, M. and Zaourar, S. (2015). Vector bin packing with heterogeneous bins: application to the machine reassignment problem. *Annals of Operations Research*, 242(1), 161–194.

Johnson, D.S. (2016). *Vector Bin Packing*, 2319–2323. Springer New York.

Table 3. Computational results for the 0-1 LP model on randomly generated instances

$\|O\|$	$\|R\|$	λ	#OPTI	Avg. FF	Avg. 0-1 LP	Avg. GAP, (%)	Avg. CPU, (s.)
20 ouputs on machine	40	0.25	9	7.4	2.2	3.33	30.01
		0.5	10	5.5	2.0	⊕	1.56
		0.75	10	3.8	2.0	⊕	0.42
	80	0.25	5	13.0	6.5	26.26	57.41
		0.5	10	8.5	4.0	⊕	10.50
		0.75	10	6.3	4.0	⊕	2.90
	120	0.25	7	13.6	8.6	17.63	30.38
		0.5	10	10.8	6.0	⊕	21.60
		0.75	10	7.9	6.0	⊕	6.18
	160	0.25	8	16.4	11.5	13.72	72.64
		0.5	10	13.7	8.0	⊕	49.76
		0.75	10	9.8	8.0	⊕	10.76
	180	0.25	9	17.3	11.0	6.90	72.29
		0.5	10	14.9	9.0	⊕	61.78
		0.75	10	10.6	9.0	⊕	8.95
40 ouputs on machine	40	0.25	0	6.3	6.3	68.71	⊖
		0.5	10	3.2	1.0	⊕	22.51
		0.75	10	2.2	1.0	⊕	1.87
	80	0.25	0	10.6	10.6	80.98	⊖
		0.5	7	5.0	2.8	17.0	99.60
		0.75	10	3.3	2.0	⊕	6.41
	120	0.25	0	14.5	14.5	79.08	⊖
		0.5	0	7.3	7.3	58.75	⊖
		0.75	10	4.7	3.0	⊕	17.37
	160	0.25	0	18.2	18.2	85.42	⊖
		0.5	0	9.3	9.3	56.89	⊖
		0.75	6	6.0	4.8	13.33	92.86
	180	0.25	0	20.5	20.5	91.72	⊖
		0.5	0	10.8	10.8	56.29	⊖
		0.75	3	7.0	6.4	20.00	129.19
80 ouputs on machine	40	0.25	0	4.0	4.0	74.00	⊖
		0.5	10	2.0	1.0	⊕	5.53
		0.75	10	1.4	1.0	⊕	1.99
	80	0.25	0	6.2	6.2	83.46	⊖
		0.5	0	3.1	3.1	67.50	⊖
		0.75	9	2.1	1.2	6.67	13.45
	120	0.25	0	8.6	8.6	82.09	⊖
		0.5	0	4.1	4.1	51.00	⊖
		0.75	9	3.0	2.1	3.33	53.48
	160	0.25	0	10.8	10.8	90.88	⊖
		0.5	0	5.1	5.1	60.74	⊖
		0.75	0	3.4	3.4	40.00	⊖
	180	0.25	0	12.3	12.3	92.03	⊖
		0.5	0	5.9	5.9	59.81	⊖
		0.75	2	4.0	3.8	20.00	114.91

(⊕) All optimal solutions were found within the time limit of 150 seconds.
(⊖) No optimal solution was found within the time limit of 150 seconds.