Balancing the workload in logistics platforms by joint optimization of inbound and outbound flows

Marc-Antoine Coindreau[1], Olivier Gallay[2], Nicolas Zufferey[1] , Gilbert Laporte[3]

[1] GSEM - University of Geneva, 1205 Geneva 4, Switzerland,
{marc-antoine.coindreau,n.zufferey}@unige.ch

[2] Université de Lausanne, HEC, 1015 Lausanne, Switzerland, olivier.gallay@unil.ch

[3] HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal H3C 3J7, Canada,
gilbert.laporte@hec.ca

Abstract : *Over a weekly planning horizon, we consider the daily cross-docking operations at intermodal logistics platforms. Each day, products are received by truck from inland suppliers. The same day or later in the week, the products have then to be loaded into containers which are then shipped by boat to offshore production plants at the end of the week. The full content of a container must be available at the cross-docking platform to enable its loading operations to start. The objective is to smooth the workload over the week. Results have been obtained for real instances provided by a large European car manufacturer.*

Keywords : *Cross-docking, Matheuristic, Real Instances.*

1 Introduction

In this study, we model and solve a problem encountered by a European Car Manufacturer (this manufacturer cannot be named for confidentiality reasons and will be designated here by ECM). ECM consolidates the product flows of its European suppliers to its offshore plants (in America, Africa, or Asia) by routing these products through cross-docking platforms (CDPs). Over one week (from Monday to Friday), products are delivered by trucks to the CDPs. These products are then sorted and reconditioned (to satisfy the constraints of sea transport, in particular) and loaded into containers which are finally shipped by boat to the production plants.

ECM seeks to smooth the workload over the week. The workload of a day is proportional to the volume of products handled (i.e., the sum of the volumes of products unloaded from trucks and loaded into containers). To do this, ECM has to determine the content of the trucks, the content of the containers, and the loading day of the containers. We denote this problem as the ECM problem.

For each CDP, ECM solves several complex optimization problems. First of all, ECM defines in advance the truck routes to collect the products from the suppliers (e.g., [1]). Truck routes cannot be modified (i.e., the truck arrival day and the list of visited suppliers), but ECM can nevertheless choose the products that are collected from each supplier in each route (as long as the truck loading constraints are satisfied). Next, ECM minimizes the number of containers needed to ship all the products (this problem was proposed for the ROADEF 2015 ESICUP competition). Finally, during the loading phase, ECM must wait until all the products composing a container are present in the CDP before starting to load the container (the number of loading doors being limited). From an operational point of view, the loading day of each container and the number of containers loaded per day are not constrained.

Currently, ECM independently solves the optimization problems related to the content of trucks and containers. Once these problems have been solved, ECM decides on the loading day of each container. As the entire content of the containers must be available in the CDP at the time of loading, it appears that ECM must wait until the end of the week to carry out most of the loading operations. This creates unbalanced workloads within the CDPs over the week (the workload is 2-3 times larger in the last days of the week than in the first days).

In this work, we propose to jointly optimize the content of trucks and containers to smooth the workload over the week. Changing the content of trucks and containers involves taking into account complex loading constraints (size, weight and position of the transported products). A detailed description of these constraints can be found in [3]. The full consideration of loading constraints would lead to an untractable optimization problem. However, these loading problems can be simplified as the products are loaded into boxes, and the loading constraints only apply to the boxes regardless of the products they contain. In the data provided by ECM, more than 70% of the boxes can transport different products with weight variations of less than 10kg. Thus, for a given loading of boxes into containers (resp. trucks), we can evaluate a large number of box-to-product assignments without violating the loading constraints.

The contributions of this work are the following. We model the ECM problem with a mixed-integer linear program (MILP) and we propose a matheuristic (called fix-and-optimize) to solve the ECM instances (that involves large MILPs). This matheuristic consists in iteratively solving sub-problems generated by fixing the value of certain variables in the MILP. From a managerial point of view, we quantify the gain brought by the joint optimization of truck and container contents in the CDPs by comparing our results with current industrial practice on real instances provided by ECM.

2 Model

In a preliminary work [2], we considered a simplified version of the ECM problem detailed above, where we did not consider the truck contents. The reader is referred to that work for a description of the associated literature review and for the justification of the $\mathcal{NP}$-hardness of the problem considered here.

2.1 Sets, parameters, and variables

For the needs of the matheuristic, we detail here a MILP in which the contents of some of the trucks and some of the containers are fixed. The exponents $^{(in)}$ (resp. $^{(out)}$) refer to the parameters related to trucks (resp. containers).

T is the time horizon (i.e., days). The following sets are considered. C is the set of CDP clients (i.e., production plants), P is the set of product types, B is the set of box types, S is the set of suppliers, I is the set of trucks (I_t is the set of trucks arriving on day $t \in T$), and O is the set of containers (O_c is the set of containers that are sent to the client $c \in C$). The sets for which the variables are fixed (resp. not fixed) have as exponent $^{(f)}$ (resp. $^{(nf)}$). Therefore, $I^{(nf)}$ refers to all trucks for which the content can be modified.

The following parameters are given.

$g_p \in \mathbb{N}$: number of units of product type $p \in P$ already available in the inventory at the beginning of the week

M_{op} : largest amount of product of type $p \in P$ that can be transported in container $o \in O$

$d_{cp} \in \mathbb{N}$: demand of client $c \in C$ for product type $p \in P$ (in units)

$l_{pb} \in \mathbb{R}^+$: weight of a box of type $b \in B$ when filled with product type $p \in P$ (in kg)

$q_{pb} \in \mathbb{N}$: number of units of product type $p \in P$ that can be transported in box type $b \in B$

$n_{ib}^{(in)} \in \mathbb{N}$: maximum number of units of boxes of type $b \in B$ transported in truck $i \in I$

$n_{ob}^{(out)} \in \mathbb{N}$: maximum number of units of boxes of type $b \in B$ transported in container $o \in O$

$q_{ip}^{(in)} \in \mathbb{N}$: number of products of type $p \in P$ delivered by truck $i \in I^{(f)}$

$q_{op}^{(out)} \in \mathbb{N}$: number of products of type $p \in P$ sent by container $o \in O^{(f)}$

$l^{(in)} \in \mathbb{R}^+$: maximum allowed weight that can be transported by a truck (in kg)

$l^{(out)} \in \mathbb{R}^+$: maximum allowed weight that can be transported by a container (in kg)

$h_p \in \mathbb{R}^+$: volume of a product of type $p \in P$ (in m^3)

π_{pi} = 1 if truck $i \in I$ visits the supplier that can provide product type $p \in P$ ($\pi_{pi} = 0$ otherwise)

The following variables have to be determined.

$u_{pt} \in \mathbb{N}$: number of units of product type $p \in P$ in stock on day $t \in T$ before loading the containers

$v_{pt} \in \mathbb{N}$: number of units of product type $p \in P$ in stock on day $t \in T$ after loading the containers

$r_{pt} \in \mathbb{N}$: number of units of product type $p \in P$ received on day $t \in T$

$s_{pt} \in \mathbb{N}$: number of units of product type $p \in P$ sent on day $t \in T$

$z_{ibp} \in \mathbb{N}$: number of boxes of type $b \in B$ assigned to product type $p \in P$ in truck $i \in I^{(nf)}$

$x_{obp} \in \mathbb{N}$: number of boxes of type $b \in B$ assigned to product type $p \in P$ in container $o \in O^{(nf)}$

$w_{opt} \in \mathbb{N}$: number of units of product type $p \in P$ sent by container $o \in O^{(nf)}$ on day $t \in T$

$m_t \in \mathbb{R}$: workload performed on day $t \in T$ (in m^3)

$f \in \mathbb{R}$: workload difference between the most loaded day and the least loaded one

y_{ot} = 1 if container $o \in O$ is loaded on day $t \in T$ ($y_{ot} = 0$ otherwise)

2.2 MILP

We denote by $Q(I^{(nf)}, O^{(nf)})$ the MILP associated with the ECM problem. Only the contents of the $I^{(nf)}$ trucks and the $O^{(nf)}$ containers can be modified in this MILP. The MILP is described as follows :

$$\text{Minimize} \quad f \tag{1}$$

Subject to

$$f \geq m_{t_1} - m_{t_2} \qquad t_1, t_2 \in T \tag{2}$$

$$m_t = \sum_{p \in P} h_p \cdot r_{pt} + \sum_{o \in O} \sum_{p \in P} h_p \cdot w_{opt} \qquad t \in T \tag{3}$$

$$v_{pt} = u_{pt} - s_{pt} \qquad p \in P,\, t \in T \tag{4}$$

$$u_{pt} = v_{p,t-1} + r_{pt} \qquad p \in P,\, t \in T \tag{5}$$

$$v_{p0} = g_p \qquad p \in P \tag{6}$$

$$r_{pt} = \sum_{b \in B} \sum_{(i \in I_t^{(nf)} | \pi_{pi} > 0)} q_{pb} \cdot z_{ibp} + \sum_{i \in I_t^{(f)}} q_{ip}^{(in)} \qquad p \in P,\, t \in T \tag{7}$$

$$s_{pt} = \sum_{o \in O^{(nf)}} w_{opt} + \sum_{o \in O^{(f)}} q_{op}^{(out)} \cdot y_{ot} \qquad p \in P,\, t \in T \tag{8}$$

$$\sum_{t \in T} y_{ot} = 1 \qquad o \in O \tag{9}$$

$$w_{opt} \leq M_{op} \cdot y_{ot} \qquad t \in T,\, o \in O,\, p \in P \tag{10}$$

$$w_{opt} \leq \sum_{b \in B} q_{pb} \cdot x_{obp} \qquad t \in T,\, o \in O,\, p \in P \tag{11}$$

$$\sum_{o \in O_c^{(nf)}} \sum_{t \in T} w_{opt} = d_{cp} - \sum_{o \in O_c^{(f)}} q_{op}^{(out)} \qquad c \in C,\, p \in P \tag{12}$$

$$\sum_{b \in B} \sum_{p \in P} l_{pb} \cdot x_{obp} \leq l^{(out)} \qquad o \in O^{(nf)} \tag{13}$$

$$\sum_{p \in P} x_{obp} \leq n_{ob}^{(out)} \qquad o \in O^{(nf)},\, b \in B \tag{14}$$

$$\sum_{b \in B} \sum_{p \in P} l_{pb} \cdot z_{ibp} \leq l^{(in)} \qquad i \in I^{(nf)}. \tag{15}$$

$$\sum_{p \in P | \pi_{pi} > 0} z_{ibp} \leq n_{ib}^{(in)} \qquad i \in I^{(nf)},\, b \in B \tag{16}$$

Constraints (2) sets the difference between the most loaded working day and least one. Constraints (3) calculate for each day the value of the workload. Constraints (4) (resp. (5)) compute the available inventory in the CDP at the end (resp. at the beginning) of the day. Constraints (6) fix the initial inventory in the CDP at the beginning of the planning horizon (i.e., the products not obtained in the planning horizon already belong to the inventory from the first day of the week). For each day, constraints (7) determine the amount of products obtained at the CDP, whereas constraints (8) compute the number of units sent for each product type. Constraints (9) ensure that no container is loaded more than once. Constraints (10) restrict each product to be sent on the loading day of a container. For each containes, constraints (11) bound the amount of products sent. Constraints (12) satisfy the demand of each client. The loading constraints of the containers (resp. trucks) are in constraints (13) and (14) (resp. (15) and (16)). More precisely, constraints (13) (resp. (15)) limit the weight of the transported products to the container (resp. the truck) capacity, and constraints (14) (resp. (16)) ensure that the number of boxes transported in a container (resp. in a truck) satisfy the allowed upper bound.

2.3 Specific configurations

We consider the configurations presented below.

$$
\begin{array}{rcl}
(Q) & : & \text{the truck and container contents are optimized} \\
 & & \text{(i.e., } O^{(nf)} = O \text{ and } I^{(nf)} = I) \\
(Q_z) & : & \text{the container contents are optimized} \\
 & & \text{(i.e., the } z_{ibp} \text{ variables are fixed : } O^{(nf)} = O \text{ and } I^{(nf)} = \emptyset) \\
(Q_x) & : & \text{the truck contents are optimized} \\
 & & \text{(i.e., the } x_{obp} \text{ variables are fixed : } O^{(nf)} = \emptyset \text{ and } I^{(nf)} = I) \\
(Q_{x,z}) & : & \text{the loading day of the containers is optimized} \\
 & & \text{(i.e., } O^{(nf)} = \emptyset \text{ and } I^{(nf)} = \emptyset)
\end{array}
$$

$(Q_{x,z})$ captures the current practice at ECM, where the decision maker builds "by hand" the loading day of each container (in a step-by-step fashion). In such a context, the truck and container contents are built at an earlier stage with two different optimization systems.

3 Fix-and-optimize matheuristic

As CPLEX is unable to find any feasible solution for the larger instances provided by ECM, we introduce a fix-and-optimize matheuristic (FOM) in Algorithm 1. This algorithm optimizes the truck content, the container content, and the container loading day, by iteratively optimizing a subset of trucks and containers. It takes as an input an initial feasible solution (i.e., satisfying constraints (2) – (16)). For example, we can initialize FOM with the solution currently used by ECM. At each iteration of the algorithm, a subset $I^{(nf)}$ of truck and $O^{(nf)}$ of containers are selected. The resulting $Q(I^{(nf)}, O^{(nf)})$ problem is then solved with CPLEX. We set CPLEX to stop when it finds a solution at σ% of the lower bound or after t_{MILP} minutes of execution time. The new solution is then updated. As CPLEX is launched with an initial feasible solution (the one obtained at the previous iteration), $Q(I^{(nf)}, O^{(nf)})$ never returns a deteriorating solution. FOM stops if the execution time is greater than t_{max} or after η_{max} iterations without improvements. Preliminary experiments show that efficient FOM parameters can be defined as follows : $\rho = 10\%$, $\sigma = 2\%$, $t_{MILP} = 5$ minutes, $\eta_{max} = 50$, $t_{max} = 10$ hours. Despite ECM allows a runtime of 10 hours (i.e., FOM can be run overnight), our solutions have been usually obtained within one hour.

Algorithm 1 Fix-and-optimize matheuristic

Input : s_0 : initial solution, ρ, (σ, t_{MILP}), (η_{max}, t_{max})

Set : $l = 1$.

While no stopping is met **do :**

1. Choose randomly the trucks $I^{(nf)}$ and the containers $O^{(nf)}$ to optimize (i.e., set $|I^{(nf)}| = \rho \cdot |I|$ and $O^{(nf)} = \rho \cdot |O|$).
2. Solve $Q(O^{(nf)}, I^{(nf)})$. CPLEX stops when the gap between the best known solution and the lower bound is lower than σ% or after an execution time greater than t_{MILP}. Let s_l denote the resulting solution.
3. Stop FOM if the solution has not been changed for η_{max} iterations or if the execution time is larger than t_{max}. Otherwise set $l = l + 1$.

Return : s_l

4 Results

We use C++ for the algorithms and CPLEX 12.4 to solve the MILPs. The employed computer has the following configuration : 2.2 GHz Intel Core i7 with 16 Go 1600 MHz DDR3 of RAM memory.

4.1 Instances

ECM provided us with data of three different CDPs : V, G, and M. Several thousand of different product types transit through the CDPs (up to 8,649 product types). On average, the smallest instances involve a few dozen trucks and containers whereas the largest instances involve several hundred trucks and containers (up to 1,104 trucks and 829 containers). Our experiments have shown that for the smaller instances (i.e., instances V and G), CPLEX can solve the (Q) formulation within an hour. In contrast, for the M instances, CPLEX fails to solve the (Q_z) and (Q) formulations after 10 hours of execution time.

4.2 Performance of FOM

For the formulation (Q_z), Table 1 compares the results of CPLEX and FOM for the M instances (which cannot be solved with CPLEX). The execution time at disposal is 1 hour. We do not allocate the entire execution time at disposal as we solve (Q_z), which is a simplified formulation of the ECM problem in which the content of the trucks is not optimized. We compare the results of FOM with those of CPLEX for different sizes of $O^{(nf)}$. We display the percentage gain, which is computed as follows : $\frac{f_{Q(I^{(nf)},O^{(nf)})}}{f_{Q_{x,z}}}$, where $f_{Q(I^{(nf)},O^{(nf)})}$ is the solution of $Q(I^{(nf)},O^{(nf)})$ and $f_{Q_{x,z}}$ denotes the workload gap observed with the current ECM solution. It highlights how ECM current solutions can be improved by considering different subsets $O^{(nf)}$ in $Q(I^{(nf)},O^{(nf)})$ (we recall that, here, $I^{(nf)} = \emptyset$). For column FOM, we display $\frac{f_{FOM}}{f_{Q_{x,z}}}$, where f_{FOM} is the solution returned by FOM. Hence, For instance M1, Table 1 indicates that the workload gap returned by FOM is 60% of the workload gap of the ECM current solution. We can first observe that CPLEX is more efficient when some variable are fixed. Indeed, the solution returned by CPLEX for (Q_z) is never better than the solution of $Q(\emptyset, O^{(nf)})$ where $|O^{(nf)}| < |O|$. Second, with one hour of execution time, FOM outperforms CPLEX, whatever the size of $O^{(nf)}$ used in $Q(\emptyset, O^{(nf)})$. Solving iteratively small MILPs leads to better solutions than directly solving one large MILP. Using CPLEX, the best solutions are achieved when solving $Q(\emptyset, O^{(nf)} = 20\% \cdot |O|)$. In that case, the workload gap found is 83.9% of the workload gap observed in ECM current solutions. Considering FOM allows to find a workload gap of 67.1% of the one observed in the ECM current solutions (average over the seven M instances).

4.3 Gain achieved by joint optimization of truck and container contents

In Table 2, the following solutions are compared : (1) the Q solutions (i.e., the truck and contained contents are optimized) ; (2) the Q_x solutions (i.e., the container contents are fixed) ; (3) the Q_z solutions (i.e., the truck contents are fixed) ; (4) the $Q_{x,z}$ solutions (i.e., the truck and contained contents are fixed). The optimal solution value is given in column "Obj.", whereas column "Time" indicates the time needed for CPLEX to find optimality (in minutes). Columns "% (ECM)", "% Q_z" and "%(Q_x)" give the improvement percentage with respect to ECM, Q_z and Q_x, respectively. For instance, the improvement of configuration Q over Q_z is given in column "% Q_z" and is computed as $\frac{f(Q)-f(Q_z)}{f(Q_z)}$, where $f(Q_z)$ (resp. $f(Q)$) is the obtained workload gap when considering configuration Q_z (resp. Q).

TAB. 1 – CPLEX versus FOM results ($I^{(nf)} = \emptyset$)

Inst.	$\lvert O^{(nf)}\rvert = 10\% \cdot \lvert O\rvert$		$\lvert O^{(nf)}\rvert = 20\% \cdot \lvert O\rvert$		$\lvert O^{(nf)}\rvert = 30\% \cdot \lvert O\rvert$		$\lvert O^{(nf)}\rvert = 100\% \cdot \lvert O\rvert$		FOM
	Obj.	LB	Obj.	LB	Obj.	LB	Obj.	LB	Obj.
M1	96.1%	96.1%	86.1%	83.1%	99.9%	61.1%	100.0%	41.2%	60.0%
M2	99.7%	99.7%	93.8%	93.8%	85.3%	81.1%	100.0%	63.6%	81.9%
M3	90.5%	90.5%	80.6%	79.9%	74.0%	73.2%	100.0%	60.0%	69.3%
M4	83.5%	83.5%	71.5%	66.6%	84.7%	60.8%	99.8%	46.0%	67.9%
M5	86.7%	86.7%	78.4%	78.4%	74.4%	74.4%	100.0%	57.6%	70.1%
M6	87.2%	87.2%	77.2%	77.2%	99.9%	53.7%	99.7%	38.2%	60.8%
M7	89.6%	89.6%	100.0%	72.4%	100.0%	46.5%	100.0%	33.6%	59.9%
Avg.	90.5%	90.5%	83.9%	78.8%	88.3%	64.4%	99.9%	48.6%	67.1%

As already highlighted in [2], reworking the container contents leads to significant improvements with respect to the current practice (on average, an improvement of 6.2% for the V instances, and of 30% for the G instances). Similar improvements are obtained here when reworking the truck contents (on average, an improvement of 13% for the V instances and of 19% for the G instances). The main achievement is obtained when reworking the truck and container contents in an integrated manner (together with the loading day of the containers). Indeed, when compared to [2], the additional average improvement obtained by solving Q instead of Q_z increases to 11.8% (resp. 30%) for the V (resp. G) instances. When compared to the current practice at ECM, the average improvement is of 19% for the V instances, and 70% for the G instances. Similar performances are obtained by FOM for the M instances.

TAB. 2 – Results of the different formulations for the V and G instances.

Inst.	$(Q_{x,z})$		(Q_z)			(Q_x)			(Q)				
	Obj.	Time	Obj.	Time	%(ECM)	Obj.	Time	%(ECM)	Obj.	Time	%(ECM)	%(Q_z)	%(Q_x)
V1	1,037	< 1	1,037	< 1	0.0%	780	< 1	-24.8%	778	< 1	-25.0%	-25.0%	-0.3%
V2	1,863	< 1	1,757	< 1	-5.7%	1,728	< 1	-7.2%	1,626	2	-12.7%	-7.5%	-5.9%
V3	1,946	< 1	1,841	< 1	-5.4%	1,781	< 1	-8.5%	1,778	< 1	-8.6%	-3.4%	-0.2%
V4	2,528	< 1	2,304	< 1	-8.9%	2,237	< 1	-11.5%	2,024	< 1	-19.9%	-12.2%	-9.5%
G1	3,964	< 1	3,317	< 1	-16.3%	3,701	< 1	-6.6%	2,774	< 1	-30.0%	-16.4%	-25.0%
G2	2,492	< 1	1,593	8	-36.1%	1,818	< 1	-27.0%	881	50	-64.6%	-44.7%	-51.5%
G3	5,014	< 1	3,977	< 1	-20.7%	4,661	< 1	-7.0%	3,700	< 1	-26.2%	-7.0%	-20.6%
G4	3,565	< 1	2,633	5	-26.1%	3,173	< 1	-11.0%	2,352	10	-34.0%	-10.7%	-25.9%
G5	4,859	< 1	3,703	31	-23.8%	3,328	< 1	-31.5%	2,022	> 60	-58.4%	-45.4%	-39.2%

5 Conclusions

Considering various operations in cross-docking platforms, we have proposed models and solution techniques for a real problem using real data. The workload has to be smoothed over the planning horizon (a week), which is obtained through the minimization of the gap between the most loaded working day and the least loaded one. We compare our results with current practice at ECM, that acts as a non-integrated solution method where truck and container contents are optimized independently. Computational experiments showed that allowing product reassignment from one container to another and from a truck to another leads to improvements up to 70%.

From a methodological point of view, we have developed a fix-and-optimize matheuristic (FOM). For large MILPs, FOM is able to find competitive solutions while CPLEX can barely improve the initial one within the same amount of time. In an industrial context, we think that FOM should be more widely used. Indeed, FOM allows managers to find good solutions for complex MILPs. Compared to other matheuristics or metaheuristics, FOM does not require cumbersome developments and is easy to maintain. For instance, changes in the problem formulation can easily be taken into account by just modifying the associated MILP, while it could involve costly computational effort when considering metaheuristics. For these reasons, we expect that FOM could have a great future in other industrial contexts.

References

[1] F. Boctor, G. Laporte, and J. Renaud. Heuristics for the traveling purchaser problem. *Computers & Operations Research*, 30(4) :491–504, 2003.

[2] M.-A. Coindreau, O. Gallay, N. Zufferey, and G. Laporte. Integrating workload smoothing and inventory reduction in three intermodal cross-docking platforms of a European car manufacturer. *Computers & Operations Research*, 112 :104762, 2019.

[3] T Toffolo, E. Esprit, T. Wauters, and G. Vanden Berghe. A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem. *European Journal of Operational Research*, 257(2) :526–538, 2017.