n-Cyclic Hoist Scheduling Problem to manufacture more than 2 products

Manuel Mateo, Ramon Companys

Departamento de Organización de Empresas

Universidad Politécnica de Cataluña

Avda Diagonal, 647, 7th. E-08028 Barcelona (Spain)

manel.mateo@upc.edu*, ramon.companys@upc.edu

ABSTRACT: *Jobs of more than two products (n>2) must be manufactured in a production line of tanks. A hoist transports the jobs between tanks. If the size of both batches is equal and the number of jobs is high, a cyclic scheduling can be proposed. The problem consists in the scheduling of hoist movements, which is known as CHSP (Cyclic Hoist Scheduling Problem). The objective is double: first, determine a sequence of the n products in a job cycle and then minimise the cycle time for the given sequence of products (n-cycle). We propose the use of a branch-and-bound procedure previously developed, which was developed for the 2-cycle. As there are different sequences of products, the experience demonstrates that the use of upper bounds as a result of previous steps in the algorithms generally leads to the optimum solution in lower times.*

KEYWORDS: *Scheduling, Hoist Scheduling Problem, branch-and-bound, n-cycle.*

1. Introduction

This work is included in the frame of the well-known *Hoist Scheduling Problem* (HSP). Sometimes, companies need to manufacture several different products in a single line, which is composed of a sequence of tanks.

Some of the main features for the production are:

- The manufacturing system is composed of several tanks, in which the jobs are submerged to receive a certain treatment. The production is composed of a sequence of operations, done in a set of tanks (1, 2, ..., m).

* Corresponding author: Manuel Mateo. Address: Avda Diagonal, 647, 7th. E-08028 Barcelona (Spain). Tel.: (34) 934011702; fax: (34)934016054. E-mail: manel.mateo@upc.edu.

- Each job has a processing time windows, associated to an operation through the line. Windows are defined as the difference between the maximum times $b_{i,j}$ and the minimum times $a_{i,j}$, for the operation *i* to be received by product *j*.
- Transportation of jobs among tanks is carried out by a hoist moving in the upper part of the installation. It picks up a job from a tank at any feasible time and transfers the job to the next processing tank. This is done from the loading station to the unloading station, moving sequentially the jobs through the *m* tanks. The jobs must always remain inside the baths within the predefined time limits.

The first model for the HSP was developed by Phillips and Unger (1976). Manier and Bloch (2003) grouped the different works on this problem and classified them taking into account the variants considered in each one of the papers. According to their notation, our case can be classified as: CHSP | mt/ /diss | /2,mt+2 | *Tmin*.

Generally, the problem to determine the scheduling for operations done by a hoist with the objective to optimize the productivity appears as a NP-complete problem (Lei and Wang, 1989), even for the simplest variant of CHSP (CHSP | mt/ /∅ | /1,mt | *Tmin*.). Considering these conditions, the CHSP is solved for a single hoist, tanks with capacity for only one job and no problems in the carriers' circulation.

2. Formulation of the problem n-cycle for n products

A large number of jobs must be treated on a production line, in which we suppose each treatment will take place in a tank (i=1,2,...,m). The loading station is noted as i=0 and the unloading station as i=m+1. We consider the jobs of different products j=1,...,n. For any tank, there is a particular time window delimited by two values: a minimum $a_{i,j}$ and a maximum $b_{i,j}$. Hoists require time to load a product from a tank *i*, and move it to the tank *i+1*, for the next operation. This full-hoist time will be noted as f_i (i=0,1,..,m). An empty-hoist travel, between two full-hoist movements, requires a time $e_{i,i'}$ (i=0,...,m+1; i'=0,...,m+1) from a tank *i* to another tank *i'*.

The concept of **cyclic sequence** is introduced to study the case of identical jobs (all the jobs of the same product), as Phillips and Unger (1976) and Shapiro and Nuttle (1988) described. The hoist repeats cyclic movements to treat the input of jobs. The sequence is large enough to be considered infinite. The objective, according to that approach, is to minimise the cycle time, which is defined as the time consumed by the hoist to carry out a complete sequence of movements (Shapiro and Nuttle, 1988).

If a single job is introduced into the production system and another leaves the system during each cycle, it is referred to as **1-cycle** schedule or simple cyclic schedule. Otherwise, if more than one job is introduced and is removed from the system during each cycle, the optimum result is known as **n-cycle** schedule.

The 1-cycle, in which one job but not necessarily the same, is introduced into and removed from a tank during each cycle, is the most studied case in literature. Besides the above mentioned Phillips and Unger (1976) who used Integer Linear Programming, Constraint Logic Programming is used by Baptiste et al (1992) and Baptiste et al (1993) or graphs are used by Lei (1993), Armstrong et al (1994), Chen et al (1998), among others.

Authors like Shapiro and Nuttle (1988) propose that their algorithms could run on the n-cycle as well as on the 1-cycle, but they give no solutions. Lei and Wang (1989) affirm that the n-cycle scheduling problem is also NP-complete. If the variety of products is not high and a lot of jobs must be produced, a cycle schedule may be convenient. For these problems, the objective is again to obtain the minimum cycle time to have any kind of job in the same positions at the beginning and at the end of that cycle. Lei and Liu (2001) develop a branch and bound to obtain the optimal cycle scheduling in a line with two products, i.e. searching the best sequence for the 2-product 2-cycle. Che et al (2011) want to find the cyclic sequence of hoist moves for the n-cycle, but considering a single product.

For the n-cycle model with different products, we defined two concepts (Mateo and Companys, 2006): **tank or bath** (associated to the real layout) and **stage or operation** (for the model).

A **tank or bath** is the physical element located in the production line, where all the operations for each product, relative to the same treatment, are done. We consider there are as many tanks as treatments.

A **stage** is each one of the treatments received by any product in any tank of the line.

Considering n products in a cycle, a line is composed of a set of stages, n of which are associated to a tank. Let $k, ..., k+n-1$ be n consecutive stages in the line for the n-cycle model with n different products. We only consider the values k such that $k=n\cdot i$, being i the index for the tanks (i=0,...,m+1).

Considering stages $s_{0,...,}\ s_{n-1}$ for the loading station and stages $s_{n\cdot(m+1),...,}s_{\,[n\cdot(m+1)]+(n-1)}$ associated to the unloading station, a vector **S** includes the set of stages: **S**=(s_0, s_1, ..., s_k, ..., $s_{[n\cdot(m+1)]+(n-1)}$).

Product	1	...	n	...	j	...	1	.	n
Stage	0	...	n-1	...	k=n·i+(j-1)	...	(m+1)·n	.	[n·(m+1)] +(n-1)
Tank/station	0 (loading)			i			m+1 (unloading)		

Table 1. Notation of stages and tanks in the model of n products.

The model to schedule an n-cycle sequence for n different products considers the following hypothesis:

1. In an n-cycle, only one job of each product enters the line and only one a job of each product leaves the line.
2. After a job of product 1 is taken from stage s_0, it is processed in the stages s_n, s_{2n}, .., $s_{m\cdot n}$ and leaves the line at $s_{(m+1)\cdot n}$; and so on for products 2 to n starting from the stages s_1 to s_{n-1}, respectively.
3. Some usual conditions are considered: one treatment and single capacity for a tank.
4. A single hoist is scheduled to handle the movements of jobs between stages. The time required to move a job between stages s_k and s_{k+n} (k=0, ..., n·m+(n-1)) is f_k. The time required by the hoist to travel empty from the stage k to another stage k' is $e_{k,k'}$. The time to carry out both movements does not depend on the product. The hoist is not allowed to stop loaded, except for the stabilization pause above the bath.
5. The processing time for job j at the tank i, equivalently converted to time at the stage k, must be between the values $a_k=a_{i,j}$ and $b_k=b_{i,j}$, being $k=n\cdot i+(j-1)$ and i and j the corresponding tank and job.

6. The stages s_0,..., s_{n-1} are prepared with jobs when the hoist arrives to pick them up. Any stage s_k with k=n, ..., [n·(m+1)]+(n-1) must be empty if a hoist arrives to leave a job inside.

We will work only the steady state and omit the transitory state.

3. Model for the n-cycle sequence for n products

3.1. Definition of the vectors for movements and times

A cyclic sequence is defined from two vectors (**H,T**).

- Let **H**=($h_0,h_1,...,h_{n\cdot m+(n-1)}$) be a circular permutation of the n·(m+1) scheduled full-hoist movements, where value $h_{[l]}=h_k$ (l,k=0,...,n·m+(n-1)) means that the stage *k* will be visited in the *k*-th position of the sequence. Therefore, $h_{[l]}=h_k$ is the movement with a job from stage s_k to s_{k+n}. We assume, without loss of generality, $h_{[0]}=h_0$.
- There is also a vector **T** associated to each vector **H**, with the respective instants t_k, in which the hoist picks up a job from a stage *k*. Therefore, $t_{[l]}$ is the starting time of the l-th movement, considering without loss of generality $t_{[0]}= t_0=0$.

Given a sequence of the *n* products **J**=(j_1,..., j_n), the time required by the hoist to complete a whole cycle of movements is known as cycle time, C(**J**,**H**,**T**)=$t_{[n\cdot(m+1)]}$. The objective in this problem, the n-product n-cycle sequence, is to determine the sequence of products, each one with individual window constraints, and the hoist sequence of movements with minimum cycle time C(**J**,**H**,**T**).

While vector **H** points out the order of the *n·(m+1)* hoist movements with transport of jobs in a cycle, the cycle schedule will be defined by **T**=($t_{[0]}$, $t_{[1]}$,..., $t_{[n\cdot(m+1)]}$) being ($t_{[0]}$=0) <$t_{[1]}$<...< $t_{[n\cdot(m+1)]}$.

A cyclic sequence (**J**,**H**,**T**) will be feasible if and only if:

- The hoist movement sequence with load can be done, i.e. if there is a job in the tank to be taken and the destination tank for the job is empty.

- The processing times into the tanks have values within the established time windows.

3.2. Model associated to the n-cycle for n products

Given that the data for time windows are initially related to product *j* and tank *i*, they are converted into the defined stages *k*: $[a_k,b_k]$ for k=n,...,n·m+(n-1), where k=n·i+(j-1).

On the other hand, let be f_k (k=0,...,n·m+(n-1)) the time for loaded hoist travelling between consecutive tanks and $e_{k,k'}$ the time for unloaded hoist travelling between the stages *k* and *k'* (k,k'=0,...,(n+1)·m+(n-1)).

Given a product sequence **J**, a sequence **H** or any subsequence **H°** with less than *n·(m+1)* elements can be evaluated. A set of *p* variables arises from the values of the associated vector **T** or **T°** (remember that t_0 is fixed by hypothesis). There is an additional variable for the cycle time *C* (below we will omit vectors **H** and **T** in C(**H**,**T**) in order to simplify the notation).

3.2.1. Constraints and objective function

The processing times in tanks, or stages, are limited. Let $h_{[l]}=h_{k-n}$ and $h_{[l']}=h_k$ be two consecutive movements on the same job, in the first one of which the job is put into the tank of stage *k* and in the second it is removed after a feasible time period:

$$a_k \le t_{[l']} - (t_{[l]} + f_{k-n}) \le b_k \qquad k=n, ..., n\cdot m+(n-1) \qquad \text{if } l'>l \qquad (1)$$

$$a_k \le t_{[l']} + C - (t_{[l]} + f_{k-n}) \le b_k \qquad k=n, ..., n\cdot m+(n-1) \qquad \text{if } l'<l \qquad (2)$$

On the other hand, let $h_{[k]}$ and $h_{[k+1]}$ be two consecutive movements in the vector **H**. The constraints related to hoist times are defined as:

$$t_{[k+1]} \ge t_{[k]} + f_{[k]} + e_{[k]+n,[k+1]} \qquad k=0,1, ..., n\cdot m+(n-1) \qquad (3)$$

If one of these constraints is not accomplished with equality, the hoist must wait above the tank associated to hoist movement *k+1* until $t_{[k+1]}$. The last movement in the cycle includes C to connect with the next cycle:

$$t_0 + C \geq t_{[n\cdot(m+1)+(n-1)]} + f_{[n\cdot(m+1)+(n-1)]} + e_{[n\cdot(m+1)+(n-1)]+n,0} \tag{4}$$

The total number of constraints (1) and (2) is $2\cdot n\cdot m$. Additionally, the number of constraints (3) and (4) is $n\cdot(m+1)$.

The value to be optimized is the length of the cycle:

$$[\text{MIN}]\ C \qquad \text{if } t_{[0]} = 0 \tag{5}$$

3.2.2. Coherent subsequences

As it was presented in Mateo and Amorós (2002), if we consider two consecutive tanks (*i* and *i+1*) and the hoist movements of jobs taken from the associated stages (from *k* to *k+2n-1*), the order of movements is prefixed due to the general hypothesis of the model (the sequence of products is the same for any cycle).

The relation of hoist movements between these four stages depends on the occupation of tank *i* by a job at the beginning of the cycle, i.e., it is necessary to define a vector **D** for the initial job distribution, as in Chen, Chu and Proth (1998): $\mathbf{D}_{[m]}=(d_{[0]}, d_{[1]},..., d_{[m]})$. If we suppose the hoist only visits the loading station and tank 1, for *n* products the only feasible sequence of movements is $\mathbf{H}=(h_0, h_n, h_1,..., h_{n-1}, h_{2n-1})$.

For the addition of a new tank to the model, the *n* following stages are added to the previous permutation of movements (with one less tank). Therefore, the last $2\cdot n$ stages in the considered part of the line must accomplish some suitable conditions. That relation follows the same condition as the one considering only tanks 0 and 1. We could develop the following four cases between 2 consecutive jobs $j\in\mathbf{J}$ and $j+1\in\mathbf{J}$ (it can be also applied for product 1, or also *n+1*, as $j+1\in\mathbf{J}$ and product *n* as $j\in\mathbf{J}$):

Case 1: Stage *k+n*, in the additional tank *i+1*, with a job of product *j*.

Case 2: Stage *k+n+1*, in the additional tank *i+1*, with a job of product *j+1*.

Case 3: Tank *i+1* with no job and the first job to arrive is product *j*.

Case 4: Tank *i+1* with no job and the first job to arrive is product *j+1*.

3.3. A graph to solve a sequence for the multiple-product cycle

For the resolution of a model through a graph (Mateo et al, 2002), the constraints (1)-(4) can be rewritten grouping relevant information in a single form using four vectors, called respectively *Gfrom*, *Gto*, *Gtime* and *Gcycle*. Being Gfrom(q), Gto(q)$\in$\{0, ..., p+1\}; Gtime(q) $\in$ Z (positive or negative) and Gcycle(q) $\in$ \{-1,0,1\}, similar to what Chen, Chu and Proth (1998) proposed:

$$t_{Gto(q)} - t_{Gfrom(q)} \geq G_{time(q)} + G_{cycle(q)} \cdot C \qquad q=1, ..., [3\cdot n\cdot(m+1)]-2 \qquad (5)$$

4. Procedure for an n-cycle given n products

The objective in cyclic sequences with *n* products is to determine the sequence with a minimal cycle time in which jobs of each product are regularly mixed along the line. We remember that all the jobs visit all the tanks in the given sequence of operations, but there is an individual time window for each product.

In a cycle, a job of each one of the *n* products is introduced and finished during a cycle in the n-cycle sequence. Without loss of generality, the sequence will start with product 1. Therefore, for n=2 the whole entry sequence is automatically determined **J**=(j1,j2). But for n>2, the sequence of products **J** must be first determined.

Then, the procedure is composed of a main structure to determine the sequence of products **J**, one of the *(n-1)!*, and once given a sequence of products, the best sequence of hoist movements **H** is searched through a branch-and-bound algorithm.

4.1. Branch and bound given a sequence of products J

One of the methods to obtain the optimum solution for a problem is *branch and bound*. In this case, the procedure is based on the physical exploration of the production line, and consists in the addition of the stages of a tank to the vector **H** that describes a node of the tree. This implies in the child vertex an enlargement of the line with a new tank with respect to the vertex in the immediately previous level. The solutions in the tree are constructed according to Mateo (2001), inspired in Shapiro and Nuttle (1988) and Armstrong et al (1994).

Vertex definition. The number of levels in the tree is equal to the number of tanks along the system, *m*. Given that the sequence begins with the movement at stage 0, the rest of movements on stages related to tanks 0 and 1 are automatically determined: $\mathbf{H}=(h_0, h_n, h_1,.., h_{2n-1})$. This is related to the coherent subsequences explained in Section 3.2.2.

Given a vertex be at level *r* (r=2,…,m), it is defined by a permutation of the movements assigned to the first $n\cdot(r+1)$ stages, considering also stage 0, in the sequence **H**. The number of total sequences contained in this vertex and with the same subsequence for the initial components is $(r+1)\cdot\ldots\cdot(r+n)$.

A vertex at level *r* is defined by $\mathbf{H}_{[v][r]}=(h_{[0]},...,h_j,...,h_{[n\cdot(r+1)-1]})$, which corresponds to a permutation such that $\{h_{[0]}=h_0, h_j \mid j=1,...,n\cdot(r+1)-1\}$ and any value h_j respects the constraints on coherent subsequences. The subindex *v* shows the appearance order of the vertex in the algorithm. At each vertex, a graph is developed considering constraints (5) and is solved with a minimum-path algorithm (Mateo et al, 2002).

Branching procedure. The vector **D** is necessary for the branching. If *v* is a non-leaf vertex at level *r*, it has an associated vector $\boldsymbol{D}_{[v][r]}$. Therefore, the descendant vertices will have vectors $\boldsymbol{D}_{[v'][r+1]}$ with *n* new stages corresponding to the next tank. If two consecutive products are considered, two related stages from the added tank and two others from the previous one must form a coherent subsequence, as was shown in Section 3.2.2. Then, the rules to accomplish these coherent subsequences lead to accept or reject the descendant vertices.

Bounding procedure. Let a vertex *v* at level r (with *r* tanks) be defined by $\mathbf{H}_{[v][r]}$. A bound for the value C of all the included permutations can be determined, considering the movements included and, if necessary, an additional time for the movements that are still not fixed (Mateo, 2001).

If the vertex is a leaf, the value C corresponds to a complete sequence. If it is lower than the cycle time of the best known solution, $\boldsymbol{H}_{[v][m]}$ (and the cycle time *C*) will be the new best known solution. If the bound of any vertex is higher than C, this vertex is pruned.

Search strategies. The algorithm gives priority to the unexplored vertex with a lower bound for C. For tie-breaks, a vertex with greater value *r* has a higher priority.

4.2 Determination of the sequence of products

If *n* products (with n>2) are going to be manufactured, the first decision is the evaluation of the each possible sequence. Let us start with two examples, with the simplest situations.

If there are 3 products, and we call them as A, B and C, there are only two sequences to be analysed: J=(A,B,C) and J=(A,C,B). Equivalently, we also use respectively ABC and ACB. We recall that we are working with circular permutations; therefore, ABC is equivalent to BCA and CAB and similarly with ACB respect to CBA and BAC.

If there are 4 products, using the same notation, they can be called as A, B, C and D. The number of sequences **J** to be analysed are now six: ABCD, ABDC, ACBD, ACDB, ADBC, ADCB. For each one of them, the corresponding branch-and-bound procedure is developed.

4.3 General procedure

Let *sp* an index for the different sequences of products in a cycle and $\boldsymbol{J}_{sp}$ the sequence of products in the cycle. For instance, $\mathbf{J}_1$=(1, 2,..., n) takes the products in the initial order introduced in the algorithm.

The first proposed algorithm is composed of the following steps:

Algorithm 1

Step 1. Obtain data (number of products: n; time windows: a_k, b_k; hoist times: f_k, $e_{k,k'}$)

Step 2. sp=1; $\mathbf{J_1}$=(1, 2,..., n)

Solve a branch and bound for the sequence of products $\mathbf{J_1}$

→ Optimal sequence $\mathbf{H^*_1}$ and optimal cycle time C^*_1

$\mathbf{H^*} = \mathbf{H^*_1}$; $C^* = C^*_1$

Step 3. While not all the permutations are evaluated (product 1 is always the first one)

sp=sp+1; $\mathbf{J_{sp}}$=(1, [2],...,[n])

Solve a branch and bound for another sequence of products $\mathbf{J_{sp}}$

→ Optimal sequence $\mathbf{H^*_{sp}}$ and optimal cycle time C^*_{sp}

If $C^*_{sp} < C^*$

$\mathbf{H^*} = \mathbf{H^*_{sp}}$; $C^* = C^*_{sp}$

In Algorithm 2, a new step is added and it will be inserted between Step 1 and Step 2 in Algorithm 1. It defines a virtual product *vp*, which considers the tightest lower and upper limits in the time windows $[a_i,b_i]$ for each tank i = 1,...,m:

$$a_i = \max_j \{a_{i,j}\} \qquad i = 1,...,m \qquad (6)$$

$$b_i = \min_j \{b_{i,j}\} \qquad i = 1,...,m \qquad (7)$$

Algorithm 2

Step 1. Obtain data (number of products: n; time windows: a_k, b_k; hoist times: f_k, $e_{k,k'}$)

Step 2. Define a single virtual product *vp* with time windows:

$[a_i,b_i] = [\max_j \{a_{i,j}\}, \min_j \{b_{i,j}\}]$:

Solve a branch and bound for the this single virtual product

→ Optimal sequence $\mathbf{H^*_0}$ and optimal cycle time C^*_0

Step 3. sp=1; $\mathbf{J_1}$=(1, 2,..., n)

Given an upper bound $UB_1 = C^*_0$, solve a branch and bound for the sequence of products $\mathbf{J_1}$

→ Optimal sequence $\mathbf{H^*_1}$ and optimal cycle time C^*_1

$\mathbf{H^*} = \mathbf{H^*_1}$; $C^* = C^*_1$

Step 4. While not all the permutations are evaluated (product 1 is always the first one)

sp=sp+1; $\mathbf{J_{sp}}$=(1, [2],...,[n])

Solve a branch and bound for another sequence of products $\mathbf{J_{sp}}$

→ Optimal sequence $\mathbf{H^*_{sp}}$ and optimal cycle time C^*_{sp}

If $C^*_{sp} < C^*$

$\mathbf{H^*} = \mathbf{H^*_{sp}}$; $C^*=C^*_{sp}$

Nevertheless, the resolution of the second sequence of products may also start considering the optimal cycle time for the first analyzed sequence as an upper bound, and so on for the rest of possible sequences of products. Thus, the Algorithm 1 becomes the Algorithm 3 taking advantage of the upper bound from Step 2 in Algorithm 1.

Algorithm 3

Step 1. Obtain data (number of products: n; time windows: a_k, b_k; hoist times: f_k, $e_{k,k'}$)

Step 2. sp=1; $\mathbf{J_1}$=(1, 2,..., n)

Solve a branch and bound for the sequence of products $\mathbf{J_1}$

→ Optimal sequence $\mathbf{H^*_1}$ and optimal cycle time C^*_1

$\mathbf{H^*} = \mathbf{H^*_1}$; $C^*=C^*_1$

Step 3. While not all the permutations are evaluated (product 1 is always the first one)

sp=sp+1; $\mathbf{J_{sp}}$=(1, [2],...,[n])

Given an upper bound $UB_{sp}= C^*$, solve a branch and bound for another sequence of products $\mathbf{J_{sp}}$

→ Optimal sequence $\mathbf{H^*_{sp}}$ and optimal cycle time C^*_{sp}

If $C^*_{sp} < C^*$

$\mathbf{H^*} = \mathbf{H^*_{sp}}$; $C^*=C^*_{sp}$

Similarly, the Algorithm 2 becomes the Algorithm 4 considering the upper bound at Step 3 of Algorithm 3.

Algorithm 4

Step 1. Obtain data (number of products: n; time windows: a_k, b_k; hoist times: f_k, $e_{k,k'}$)

Step 2. Define a single virtual product *vp* with time windows:

$$[a_i,b_i] = [\max_j \{a_{i,j}\}, \min_j \{b_{i,j}\}]:$$

Solve a branch and bound for the this single virtual product

→ Optimal sequence $\mathbf{H^*_0}$ and optimal cycle time C^*_0

Step 3. sp=1; $\mathbf{J_1}$=(1, 2,..., n)

Given UB_1= C^*_0, solve a branch and bound for the sequence of products $\mathbf{J_1}$

→ Optimal sequence $\mathbf{H^*_1}$ and optimal cycle time C^*_1

$\mathbf{H^*} = \mathbf{H^*_1}$; $C^*=C^*_1$

Step 4. While not all the permutations are evaluated (product 1 is always the first one)

sp=sp+1; $\mathbf{J_{sp}}$=(1, [2],...,[n])

Given UB_{sp}= C*, solve a branch and bound for another sequence of products $\mathbf{J_{sp}}$

→ Optimal sequence $\mathbf{H^*_{sp}}$ and optimal cycle time C^*_{sp}

If $C^*_{sp} < C^*$

$\mathbf{H^*} = \mathbf{H^*_{sp}}$; $C^*=C^*_{sp}$

Example

Four products must be manufactured in a line with 3 tanks. Each product must remain in a tank for a time between a minimum and a maximum value, shown in Table 2.

Prod j		**P1**			**P2**			**P3**			**P4**	
Tank i	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
$a_{i,j}$	40	30	30	35	25	35	50	60	80	50	70	55
$b_{i,j}$	160	250	110	105	80	100	115	145	190	125	170	150

Table 2. Time windows of the example (4 products, 3 tanks).

The time required to transport a product between tanks of successive treatments is 15. Without load, the hoist needs 10 units of time.

Next, there are the results applying Algorithm 2 (Algorithm 1 would omit Step 2). Later we will comment the differences with those from Algorithm 4.

Step 2. 1-product 1-cycle (virtual product)
J_0=(A, B, C, D)

OPTIMUM:
H = 0 3 1 2
T = 0 35 80 165
C = 225

Solved nodes / created nodes: 8/9

Step 3. 4-product 4-cycle
J_1=(A, B, C, D)
OPTIMUM:
H = 0,14,11,4,15,8,1,12,5,2,9,6,3,13,10,7
T = 0,35,70,105,140,175,220,255,300,335,360,400,435,470,505,540
C = 575
Solved nodes / created nodes: 74/802

J_2=(A, C, B, D)
OPTIMUM:
H = 0,14,11,4,15,8,1,12,5,2,9,6,3,13,10,7
T = 0,35,70,105,140,175,220,255,300,335,360,400,435,470,505,540
C = 575
Solved nodes / created nodes: 71/802

J_3=(A, C, D, B)
OPTIMUM:
H = 0,11,4,15,8,1,12,5,2,9,6,3,13,10,7,14
T = 0,25,60,95,130,175,210,255,290,330,365,400,435,470,505,530
C = 585
Solved nodes / created nodes: 117/802

J_4=(A, D, C, B)
OPTIMUM:
H = 0,14,11,4,15,8,1,12,5,2,9,6,3,13,10,7
T = 0,35,70,105,140,180,225,275,320,355,380,415,450,485,520,555
C = 590
Solved nodes / created nodes: 164/874

J_5=(A, D, B, C)
OPTIMUM:
H = 0,11,4,1,15,8,5,2,12,9,6,13,10,3,14,7
T = 0,30,65,100,135,170,205,240,275,310,345,370,405,450,485,530
C = 585
Solved nodes / created nodes: 138/802

J_6=(A, B, D, C)
OPTIMUM:
H = 0,11,4,1,15,8,5,12,9,2,13,6,3,10,7,14
T = 0,25,60,95,130,165,200,225,275,320,370,415,450,475,515,540
C = 595
Solved nodes / created nodes: 177/802

Table 3 shows a comparison between the results applying Algorithm 2 and Algorithm 4.

	Algorithm 2				Algorithm 4			
Sequence of products *sp*	UB_{sp}	C^*_{sp}	Solved nodes	Created nodes	UB_{sp}	C^*_{sp}	Solved nodes	Created nodes
$\mathbf{J_0}$=(A, B, C, D)	300[1]	225[1]	8	9	300[1]	225[1]	8	9
$\mathbf{J_1}$=(A, B, C, D)	900	575	74	802	900	575	74	802
$\mathbf{J_2}$=(A, C, B, D)	900	575	71	802	575	575	59	802
$\mathbf{J_3}$=(A, C, D, B)	900	585	117	802	575	575	58	802
$\mathbf{J_4}$=(A, D, C, B)	900	590	164	874	575	575	59	802
$\mathbf{J_5}$=(A, D, B, C)	900	585	138	802	575	575	59	802
$\mathbf{J_6}$=(A, B, D, C)	900	595	177	802	575	575	59	802

Table 3. Comparison of upper bound, optimum cycle time, solved and created nodes (for Algorithms 2 and 4). [1]values for the 1-product 1-cycle problem.

Finally, we can compare both computing times. The Algorithm 2 requires 5.17 ms, while the time has been reduced to 4.03 ms using the Algorithm 4.

6. Computational results

6.1 Description of instances

The computational experience is based on a set of instances corresponding to different time windows and speed in the hoist movements.

Time windows can be classified according to the difference between the maximum and minimum times (b_k-a_k) for any stage. The minimum processing time for operations is randomly generated, as in Ng(1996), according to a uniform distribution: a_k=U[20;80]. Following Ng(1996), the time for unloaded hoist movements between two stages *k* and *k+2* is $e_{k,k+2}$=U[5;10]. Ng(1996) adds a constant value to obtain the loaded times f_k from the unloaded $e_{k,k+2}$. We have several kinds of instances according to relation between the loaded and unloaded hoist movements: f_k =K $e_{k,k+2}$.

Some of the instances have been generated considering the products from the instances in Mateo (2001). Later, data from different instances have been mixed to obtain the time windows for the *n* products. A summary of the number of instances in this experiment is shown in Table 4. Annex 1 shows the tested instances.

Products	Tanks	Set of Instances
3	5	30
3	6	15
4	4	15
4	5	15

Table 4. Number of instances according to the number of products and tanks.

This computational experience is composed of instances of three or four different products. The objective is to analyse and compare the results and computing time of instances for the 3-product 3-cycle and the 4-product 4-cycle sequences. In these cases, additionally to the branch-and-bound resolution, the input sequence for jobs in both situations must be determined and repeat as many times as necessary the branch-and-bound. It has been developed in two parts:

- Block I: 3-product 3-cycle problem.
- Block II: 4-product 4-cycle problem.

The algorithms were written in Visual C++ and run in a Pentimum 1.73 GHz and 1GB RAM.

6.2. Analysis of the results

6.3.1. Analysis of 3-product 3-cycle

There are 30 instances of 3 products in 5 tanks. Table 5 indicates the number of instances in which the Algorithm 1, 2, 3 or 4, with the two different input sequences of products (ABC or ACB), is faster than the previously executed one(s). For example, the value between row 3-ACB and column 1-ABC indicates that 19 of the 30 instances were solved in less time with Algorithm 3 and sequence of products ACB than with Algorithm 1 and sequence of products ABC. The two last columns give the number of instances with the shortest time for each of the 8 algorithms (4 algorithms for the 2 sequences ABC and ACB) and the mean computing time per instance using each running of the 8 cases. Annex 2.1 and Annex 2.2 show more detailed results.

	1-ABC	1-ACB	2-ABC	2-ACB	3-ABC	3-ACB	4-ABC	4-ACB	Instances shortest time	Mean computing time (ms)
1-ABC									5	316.62
1-ACB	**21**								1	269.22
2-ABC	16	11							2	298.40
2-ACB	21	23	**21**						6	255.49
3-ABC	3	9	4	9					0	325.06
3-ACB	19	15	18	7	**20**				2	271.29
4-ABC	18	11	24	8	26	12			2	298.16
4-ACB	20	23	21	18	20	25	**21**		12	255.51

Table 5. Number of instances solved faster applying an algorithm and an input sequence of products compared with another algorithm and/or sequence (3 products, 5 tanks).

The conclusions on the values of Table 5 are:

- Nearly half of the instances (14 over 30) require less time with Algorithm 4, considering any of the two sequences of products.
- The shortest mean time to solve an instance is obtained with Algorithm 2 or Algorithm 4 (there is technically a tie for sequence ABC, 298.40 ms and 298.16 ms, and for sequence ACB, 255.49 ms and 255.51 ms).
- Looking at more detailed results (see Annex 2), some instances are easier to be solved with an initial sequence (first ABC and later ACB) than the other one (first ACB followed by ABC). These values are 21 (1-ACB vs. 1-ABC), 21 (2-ACB vs. 2-ABC), 20 (3-ACB vs. 3-ABC), 21 (4-ACB vs. 4-ABC).

Table 6 is similar to Table 5, but with the 15 instances of 3 products in 6 tanks. The two last columns also give the number of instances with the shortest time after the 8 executions and the mean computing time per instance, respectively. Annex 2.3 and Annex 2.4 contain the results of cycle times, computing times...

	1-ABC	1-ACB	2-ABC	2-ACB	3-ABC	3-ACB	4-ABC	4-ACB	Instances shortest time	Mean computing time (s)
1-ABC									1	9.85
1-ACB	**9**								1	12.05
2-ABC	10	6							0	6.95
2-ACB	10	8	**11**						4	11.65
3-ABC	9	6	15	4					1	7.82
3-ACB	11	7	11	5	**10**				1	11.71
4-ABC	11	6	9	4	13	5			2	6.93
4-ACB	12	13	12	10	11	12	**11**		5	9.52

Table 6. Number of instances solved faster with an algorithm and an input sequence of products compared with another algorithm and/or sequence (3 products, 6 tanks).

The conclusions on the values of Table 6 are:

- Nearly half of the instances (7 over 15) require less time with Algorithm 4, using any of the two sequences of products.
- The shortest mean time to solve an instance is obtained, for both initial sequences, with Algorithm 4 (6.93 s for the initial sequence ABC, which is close to time with Algorithm 2, and 9.52 s for the initial sequence ACB).
- Once again, more detailed results (see Annex 2) show that an instance is easier to be solved with an initial sequence than with the other one: bold values are 9 (1-ACB vs. 1-ABC), 11 (2-ACB vs. 2-ABC), 10 (3-ACB vs. 3-ABC), 11 (4-ACB vs. 4-ABC).

6.3.2. Analysis of 4-product 4-cycle

There are 15 instances of 4 products in a line of 4 tanks and 15 more instances in a line of 5 tanks. As the time differences between starting with the sequence ABC or the ACB in the case of 3 products were no relevant and now with 4 products the sequences to be checked are 3!=6, in block II the algorithms have been only applied considering ABCD as the initial sequence.

Table 7, similar to Table 5, indicates that the number of instances in which the Algorithm 1, 2, 3 or 4 is faster than the others. The two last columns give the number of instances with the shortest time for each of the 4 algorithms and the mean computing time per instance in seconds. The more complete results are given in Annex 2.5.

	1-ABCD	2-ABCD	3-ABCD	4-ABCD	Instances shortest time	Mean computing time (ms)
1-ABCD					1	74.52
2-ABCD	0				0	146.46
3-ABCD	6	15			2	75.48
4-ABCD	12	15	15		12	64.11

Table 7. Number of instances solved faster with an algorithm compared with another one (4 products, 4 tanks).

The conclusions on the values of Table 7 are:

- 80% of the instances (12 over 15) require less time with Algorithm 4.
- The shortest mean time to solve an instance is obtained with Algorithm 4, clearly lower than Algorithm 2, which was technically tied for the instances of 3 products.

Table 8 is similar to Table 7, but with the 15 instances of 4 products in 5 tanks. The more complete results are given in Annex 2.6.

	1-ABCD	2-ABCD	3-ABCD	4-ABCD	Instances shortest time	Mean computing time (s)
1-ABCD					3	10.57
2-ABCD	9				7	10.37
3-ABCD	3	4			0	10.78
4-ABCD	8	6	13		5	10.29

Table 8. Number of instances solved faster with an algorithm compared with another one (4 products, 5 tanks).

The conclusions on the values of Table 8 are:

- Nearly half of the instances (7 over 15) require less time with Algorithm 2, followed by Algorithm 4 (only 5 instances).
- The shortest mean time to solve an instance is obtained with Algorithm 4 10.29 s, close to time with Algorithm 2, 10.37 s.

7. Conclusions

The proposed model is built on the difference between the concepts of tanks and stages, as stage is known as each one of the operations (one for each kind of product) which happen in a tank. The idea of assigning multiple stages to each tank brings us closer to the usual models and reduces the complexity of solving the problem.

The model for the 2-product 2-cycle has been the base for studying more complex cases. Here we provide a model to solve the n-product n-cycle with jobs of multiple products. The computational experience for 3 or 4 products shows the importance of the correct use of bounds to guaranty the optimal sequence for any of the n! feasible sequences of products. After the algorithms 1, 2, 3 and 4 have been presented, it is demonstrated the last one has a higher performance.

References

Armstrong R., Lei L., Gu S. A bounding scheme for deriving the Minimal Cycle Time of a single transporter N-stage process with time window constraints. *European Journal of Operational Research*, 1994; 130-140.

Baptiste P., Legeard B., Varnier C. Hoist Scheduling Problem: an Approach Based on Constraints Logic Programming, *IEEE International Conference on Robotics and Automation*, 1992, 2, 1139-1144.

Baptiste P., Legeard B., Manier M. A., Varnier, C. Optimization with constraint logic programming: the hoist scheduling problem solved with various solvers. *Proceedings of the 8th International Conference on Applications of Artificial Intelligence in Engineering*, 1993, 599-614.

Che, A.; Zhou, Z.; Chu, C.; Chen, H. Multi-degree cyclic hoist scheduling with time window constraints. International Journal of Production Research, 2011, 49, n. 19, 5679-5693.

Chen, H., Chu C., Proth J. M. A constraint-posting framework for scheduling under complex constraints, *IEEE*, 1995, 0-7803-2535-4/95, 269-280.

Chen, H., Chu C., Proth J. M. Cyclic Scheduling of a Hoist with Time Window Constraints, *IEEE Transactions on Robotics and Automation*, 1998; 14; n. 1; 144-152.

Lei, L. Determining the Optimal Starting Times in a Cyclic Schedule with a Given Route, *Computers and Operations Research*, 1993, 20, n. 8, 807-816.

Lei, L., Liu, Q., 2001. Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints. INFOR, 39, 2, 185-199.

Lei, L., Wang, T.J. *A proof: the cyclic hoist scheduling problem is NP-complete*, Working paper #89-0016, Rutgers University, 1989.

Manier, M.A., Bloch, C. A classification for Hoist Scheduling Problems, *International Journal of Flexible Manufacturing Systems*, 2003, 15, n. 1, 37-55.

Mateo, M. *Procedimientos de secuenciación y programación en un sistema productivo de estaciones en serie con transportadores asíncronos de material*. Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2001. http://hdl.handle.net/10803/80622

Mateo, M., Amorós, A. *Estudio del caso 2-cíclico para el Hoist Scheduling Problem*. Document Intern de Treball 2002/14. Departamento de Organización de Empresas. Universitat Politècnica de Catalunya, Barcelona, Spain, 2002.

Mateo, M., Companys, R., Bautista, J. (2002). Resolution of graphs with Bounded Cycle Time for the Cyclic Hoist Scheduling Problem. *8th International Workshop on Project Management and Scheduling*, 257-260. Valencia, Spain, 2002.

Ng, W.C. A branch and bound algorithm for hoist scheduling of a circuit board production line. *The International Journal of Flexible Manufacturing Systems*, 1996, 8, 45-65.

Phillips, L.W., Unger, P. S. Mathematical Programming Solution of a Hoist Scheduling Program, *AIIE Transactions*, 1976, 8, n. 2, 219-225.

Shapiro, G.W., Nuttle, H. W. Hoist Scheduling For A PCB Electroplating Facility, *IEE Transactions*, 1988, 20, n. 2, 157-167.

ANNEX 1

ANNEX 1.1. Data for 3 products and 5 tanks

1 104407704
5
50 114   80 184   60 145   80 191   28 67
49 126   71 178   54 152   53 159   56 166
77 184   57 142   67 160   45 105   27 65
12
8
2 105409102
5
55 123   57 138   43 104   67 151   62 143
45 123   20 52   68 204   73 207   25 73
46 107   51 115   79 194   24 52   35 87
10
7
3 109405706
5
20 47   49 121   22 54   68 167   67 148
49 131   78 206   27 78   30 88   30 81
71 417   43 412   44 372   62 320   27 215
15
10
4 102406702
5
46 107   51 115   79 194   24 52   35 87
43 125   22 55   40 103   49 131   64 169
43 279   73 762   24 168   56 274   76 693
10
7
5 107402704
5
77 184   57 142   67 160   45 105   27 65
75 209   30 90   34 102   26 77   48 130
33 361   33 249   43 286   28 112   23 208
12
8
6 110403706
5
66 162 41 98   74 171 35 85   23 52
71 186 67 183   37 94 71 190 42 119
71 417   43 412   44 372   62 320   27 215
15
10
7 102409702
5
46 107   51 115   79 194   24 52   35 87
45 123   20 52   68 204   73 207   25 73
43 279   73 762   24 168   56 274   76 693
10
7
8 102409707
5
46 107   51 115   79 194   24 52   35 87
45 123   20 52   68 204   73 207   25 73
76 299   26 268   72 673   72 484   74 726
10
7
9 110405710
5
66 162 41 98   74 171 35 85   23 52
49 131 78 206   27 78   30 88   30 81
48 388 46 241   45 386 62 358   72 618
15
10
10 109403705
5
20 47   49 121   22 54   68 167   67 148
71 186   67 183   37 94   71 190   42 119
75 505   45 413   37 240   57 193   21 202
15
10
11 205506806
5
71 167   63 146   49 109   45 109   58 141
40 109   45 130   40 109   59 162   34 99
27 153   30 183   22 207   38 308   55 371
16
8
12 203509807
5
43 98   61 143   75 167   35 79   68 160
57 168   72 196   44 130   75 193   69 177
59 372   71 632   30 235   40 255   23 137
10
5
13 208507803
5
68 149   23 54   29 65   74 183   66 160
48 139   26 75   77 224   72 190   20 58
44 257   32 339   54 179   71 290   69 259
14
7
14 209507803
5
68 167   39 95   69 165   51 122   45 110
48 139   26 75   77 224   72 190   20 58
44 257   32 339   54 179   71 290   69 259
14
7
15 204508809
5
32 72   70 168   31 75   20 50   28 67
53 135   62 177   47 121   50 142   58 159
24 134   37 135   69 572   58 246   46 332
12
6
16 204501801
5
32 72   70 168   31 75   20 50   28 67
68 173   53 135   64 170   24 64   37 104
33 119   67 629   52 330   31 234   54 417
12
6

17 202506802
5
65 158 42 101 31 69 70 160 56 127
40 109 45 130 40 109 59 162 34 99
31 231 78 423 21 128 59 178 78 663
16
8
18 203504808
5
43 98 61 143 75 167 35 79 68 160
68 179 30 79 59 168 31 86 28 78
45 439 53 208 40 147 22 111 37 220
10
5
19 201503804
28 67 40 95 26 59 29 66 43 106
44 119 59 164 64 173 31 91 43 107
28 164 61 378 26 225 32 203 57 486
20
10
20 210510804
68 152 54 120 38 84 64 159 70 159
69 176 37 107 60 166 53 156 21 61

28 164 61 378 26 225 32 203 57 486
20
10
21 308602902
5
40 89 63 155 34 84 73 171 33 78
44 129 27 75 80 238 55 148 27 70
46 403 63 370 78 386 24 227 45 411
18
6
22 301604904
5
63 157 70 161 80 199 73 174 74 185
55 158 41 121 30 86 50 138 45 134
75 727 72 249 56 492 51 316 74 289
27
9
23 304604904
5
54 128 42 94 47 113 32 77 70 166
55 158 41 121 30 86 50 138 45 134
75 727 72 249 56 492 51 316 74 289
27
9

24 303609909
5
35 81 80 187 38 94 58 130 45 111
32 92 48 123 77 215 67 194 33 83
59 260 74 662 43 248 50 464 71 457
15
5
25 309608907
5
58 128 76 185 64 156 48 118 72 172
43 110 21 60 63 179 26 72 23 60
32 352 43 210 40 135 23 90 43 158
24
8
26 309610910
5
58 128 76 185 64 156 48 118 72 172
36 94 64 168 75 224 63 187 35 105
32 317 47 217 71 698 76 609 33 120
24
8
27 302602902
5
40 92 62 148 47 116 66 159 67 157
44 129 27 75 80 238 55 148 27 70
46 403 63 370 78 386 24 227 45 411
18
6
28 305605905
5
47 105 66 161 73 174 26 61 32 80
49 133 48 136 51 131 30 78 30 83
68 221 29 141 70 737 52 560 31 219
21
7
29 307606906
5
78 186 65 152 68 156 33 79 58 141
75 202 54 145 46 129 22 57 73 184
61 319 39 421 28 290 60 322 34 299
21
7
30 310605906
5
77 180 21 47 58 129 42 102 34 78
49 133 48 136 51 131 30 78 30 83
61 319 39 421 28 290 60 322 34 299
21
7

ANNEX 1.2. Data for 3 products and 6 tanks

1 111416420
6
60 134 63 147 80 199 71 161 62 145 72 161
38 110 63 188 45 117 34 86 37 95 78 213
75 210 72 189 22 62 66 188 52 150 70 191
12
8

2 116418417
6
30 70 48 119 70 155 45 105 24 57 72 170
73 212 55 146 69 189 44 122 34 86 48 128
73 212 38 113 69 183 23 59 64 179 49 131
9
6

3 117417713
6
30 70 59 144 44 108 45 109 24 57 32 76
73 212 38 113 69 183 23 59 64 179 49 131
40 381 40 403 27 116 50 159 24 160 44 197
9
6
4 116713119
6
30 70 48 119 70 155 45 105 24 57 72 170
40 381 40 403 27 116 50 159 24 160 44 197
31 74 29 65 56 137 71 160 49 117 72 167
9
6
5 117717119
6
30 70 59 144 44 108 45 109 24 57 32 76
51 477 53 366 35 304 68 741 35 225 60 462
31 74 29 65 56 137 71 160 49 117 72 167
9
6
6 213511814
6
33 77 32 74 44 98 23 57 53 122 56 134
77 228 31 82 23 63 36 93 40 105 44 125
55 397 28 245 62 532 39 217 60 537 56 378
14
7
7 217518215
6
73 172 65 152 50 115 79 186 37 83 74 173
59 175 76 192 30 84 51 148 72 182 57 147
47 114 42 95 47 106 72 168 74 181 57 140
18
9
8 211514518
6
39 93 65 155 78 171 69 152 24 59 35 79
26 70 32 85 54 143 22 63 70 182 50 126
59 175 76 192 30 84 51 148 72 182 57 147
18
9
9 217816811
6
73 172 65 152 50 115 79 186 37 83 74 173
42 229 70 399 20 88 26 123 56 601 36 200
49 437 56 435 56 392 37 383 66 648 75 322
18
9
10 215216816
6
47 114 42 95 47 106 72 168 74 181 57 140
26 57 65 146 66 147 64 143 37 85 60 136
42 229 70 399 20 88 26 123 56 601 36 200
18
9
11 311618913
6
68 159 47 116 68 149 35 81 26 61 23 55
57 159 42 120 48 143 52 141 20 57 53 154
25 232 71 218 61 502 47 447 58 615 41 124
24
8
12 313619915
6
36 85 68 167 30 68 52 130 70 157 40 91
27 68 54 147 26 68 63 186 45 126 29 82
40 199 63 206 54 375 39 255 77 691 55 534
21
7
13 315618918
6
21 46 47 105 45 107 77 183 37 83 21 48
57 159 42 120 48 143 52 141 20 57 53 154
58 566 53 196 48 361 79 375 77 644 31 181
24
8
14 316619919
6
21 50 20 46 38 84 21 49 59 130 28 63
27 68 54 147 26 68 63 186 45 126 29 82
23 95 57 335 40 333 26 85 34 247 29 118
21
7
15 318611920
6
24 52 56 128 68 165 59 135 37 88 36 90
48 129 77 227 61 162 52 156 22 62 71 201
66 703 22 71 66 247 45 406 34 121 60 546
18
6

ANNEX 1.3. Data for 4 products and 4 tanks

1
4
44 257 32 339 54 179 71 290
68 149 23 54 29 65 74 183
48 139 26 75 77 224 72 190
68 167 39 95 69 165 51 122
14
7

2
4
63 144 63 146 37 88 30 70
71 167 63 146 49 109 45 109
40 109 45 130 40 109 59 162
27 153 30 183 22 207 38 308
16
8

3
4
68 149 23 54 29 65 74 183
48 139 26 75 77 224 72 190
44 257 32 339 54 179 71 290
68 167 39 95 69 165 51 122
14
7
4
4
32 72 70 168 31 75 20 50
53 135 62 177 47 121 50 142
24 134 37 135 69 572 58 246
33 119 67 629 52 330 31 234
12
6
5
4
68 152 54 120 38 84 64 159
69 176 37 107 60 166 53 156
28 164 61 378 26 225 32 203
44 119 59 164 64 173 31 91
20
10
6
4
54 128 42 94 47 113 32 77
55 158 41 121 30 86 50 138
75 727 72 249 56 492 51 316
63 157 70 161 80 199 73 174
27
9
7
4
77 180 21 47 58 129 42 102
49 133 48 136 51 131 30 78
61 319 39 421 28 290 60 322
78 186 65 152 68 156 33 79
21
7
8
4
58 128 76 185 64 156 48 118
43 110 21 60 63 179 26 72
32 352 43 210 40 135 23 90
32 317 47 217 71 698 76 609
24
8
9
4
40 92 62 148 47 116 66 159
44 129 27 75 80 238 55 148
46 403 63 370 78 386 24 227
40 89 63 155 34 84 73 171
18
6
10
4
47 105 66 161 73 174 26 61
49 133 48 136 51 131 30 78
68 221 29 141 70 737 52 560
75 202 54 145 46 129 22 57
21
7
11
4
30 70 48 119 70 155 45 105
30 70 59 144 44 108 45 109
73 212 55 146 69 189 44 122
73 212 38 113 69 183 23 59
9
6
12
4
31 74 29 65 56 137 71 160
30 70 59 144 44 108 45 109
73 212 38 113 69 183 23 59
40 381 40 403 27 116 50 159
9
6
13
4
51 477 53 366 35 304 68 741
30 70 48 119 70 155 45 105
40 381 40 403 27 116 50 159
31 74 29 65 56 137 71 160
9
6

14
4
73 212 38 113 69 183 23 59
30 70 48 119 70 155 45 105
51 477 53 366 35 304 68 741
40 381 40 403 27 116 50 159
9
6
15
4
40 381 40 403 27 116 50 159
30 70 59 144 44 108 45 109
51 477 53 366 35 304 68 741
31 74 29 65 56 137 71 160
9
6

ANNEX 1.4. Data for 4 products and 5 tanks

1
5
33 361 33 249 43 286 28 112 23 208
50 114 80 184 60 145 80 191 28 67
49 126 71 178 54 152 53 159 56 166
77 184 57 142 67 160 45 105 27 65
12
8

2
5
43 125 22 55 40 103 49 131 64 169
55 123 57 138 43 104 67 151 62 143
45 123 20 52 68 204 73 207 25 73
43 279 73 762 24 168 56 274 76 693
10
7
3
5
55 123 57 138 43 104 67 151 62 143
46 107 51 115 79 194 24 52 35 87
43 125 22 55 40 103 49 131 64 169
43 279 73 762 24 168 56 274 76 693
10
7
4
5
49 126 71 178 54 152 53 159 56 166
77 184 57 142 67 160 45 105 27 65
33 361 33 249 43 286 28 112 23 208
25 55 34 78 79 182 48 111 25 55
12
8
5
5
76 299 26 268 72 673 72 484 74 726
46 107 51 115 79 194 24 52 35 87
43 279 73 762 24 168 56 274 76 693
45 123 20 52 68 204 73 207 25 73
10
7
6
5
44 257 32 339 54 179 71 290 69 259
68 149 23 54 29 65 74 183 66 160
48 139 26 75 77 224 72 190 20 58
68 167 39 95 69 165 51 122 45 110
14
7
7
5
63 144 63 146 37 88 30 70 53 124
71 167 63 146 49 109 45 109 58 141
40 109 45 130 40 109 59 162 34 99
27 153 30 183 22 207 38 308 55 371
16
8
8
5
68 149 23 54 29 65 74 183 66 160
48 139 26 75 77 224 72 190 20 58
44 257 32 339 54 179 71 290 69 259
68 167 39 95 69 165 51 122 45 110
14
7
9
5
32 72 70 168 31 75 20 50 28 67
53 135 62 177 47 121 50 142 58 159
24 134 37 135 69 572 58 246 46 332
33 119 67 629 52 330 31 234 54 417
12
6
10
68 152 54 120 38 84 64 159 70 159
69 176 37 107 60 166 53 156 21 61
28 164 61 378 26 225 32 203 57 486
44 119 59 164 64 173 31 91 43 107
20
10
11
5
54 128 42 94 47 113 32 77 70 166
55 158 41 121 30 86 50 138 45 134
75 727 72 249 56 492 51 316 74 289
63 157 70 161 80 199 73 174 74 185
27
9
12
5
77 180 21 47 58 129 42 102 34 78
49 133 48 136 51 131 30 78 30 83
61 319 39 421 28 290 60 322 34 299
78 186 65 152 68 156 33 79 58 141
21
7
13
5
58 128 76 185 64 156 48 118 72 172
43 110 21 60 63 179 26 72 23 60
32 352 43 210 40 135 23 90 43 158
32 317 47 217 71 698 76 609 33 120
24
8
14
5
40 92 62 148 47 116 66 159 67 157
44 129 27 75 80 238 55 148 27 70
46 403 63 370 78 386 24 227 45 411
40 89 63 155 34 84 73 171 33 78
18
6
15
5
47 105 66 161 73 174 26 61 32 80
49 133 48 136 51 131 30 78 30 83
68 221 29 141 70 737 52 560 31 219
75 202 54 145 46 129 22 57 73 184
21
7

ANNEX 2

Notation:

#: number of instance

C^0: cycle time for the "virtual" product

Ev^0: explored vertices in the branch and bound for the "virtual" product

Ov^0: opened vertices in the branch and bound for the "virtual" product

C^1, Ev^1, Ov^1: idem for the sequence 1 of products (ABC)

C^2, Ev^2, Ov^2: idem for the sequence 2 of products (ACB)

ANNEX 2.1. Results for 3 products and 5 tanks, initial sequence of products ABC

#	C^0	Ev^0	Ov^0	C^1	Ev^1	Ov^1	C^2	Ev^2	Ov^2	m	s	ms
1				589	24050	94565	593	27912	96729	0	0	609,689
2				519	22062	46511	533	11023	24564	0	0	377,005
3				674	2676	13319	742	3432	12116	0	0	86,219
4				529	10628	33669	462	3603	23932	0	0	288,522
5				527	5586	36639	575	27943	110170	0	0	223,514
6				699	13307	84630	659	5583	44235	0	0	472,200
7				541	18815	42224	460	2741	23101	0	0	351,219
8				560	18917	29851	480	2683	17153	0	0	376,444
9				704	8334	65358	773	17392	55479	0	0	345,657
10				689	4014	24027	694	4616	25692	0	0	153,335
11				640	15536	142947	618	9017	123590	0	0	632,135
12				390	1576	15945	400	2886	25932	0	0	56,952
13				559	5902	21166	566	8287	23443	0	0	176,140
14				566	22551	99701	541	8261	74280	0	0	672,639
15				492	18135	63489	491	12793	54384	0	0	417,220
16				502	13182	48981	481	9055	39355	0	0	297,958
17				615	5052	48022	642	8498	65726	0	0	253,631
18				419	23864	60244	390	3803	32433	0	0	378,615
19				794	1932	5505	826	1883	5147	0	0	41,658
20				780	9274	68818	802	9919	67323	0	0	386,356
21				603	16062	101346	589	10700	86077	0	0	415,333
22				904	8031	36103	893	6772	32816	0	0	275,844
23				932	3545	13033	937	5124	16287	0	0	94,911
24				507	21476	53482	498	7686	55498	0	0	379,378
25				813	5228	21760	809	4478	16563	0	0	133,445
26				800	7409	53898	768	4811	68585	0	0	311,049
27				603	18144	107420	572	5906	58920	0	0	481,033
28				716	10169	38196	696	6639	34925	0	0	294,846
29				658	5664	43585	668	7692	51314	0	0	253,156
30				686	6203	46603	744	14920	42588	0	0	262,376
1	480	55	55	589	24050	94565	593	27912	96729	0	0	633.185

2	428	3	3	519	22062	46511	533	11023	24564	0	0	380.013
3	478	3	3	674	2676	13319	742	3432	12116	0	0	86.1788
4	432	3	3	529	10628	33669	462	3603	23932	0	0	289.914
5	340	54	54	527	5586	36639	575	27943	110170	0	0	227.572
6	393	44	44	699	13307	84630	659	5583	44235	0	0	477.149
7	449	3	3	541	18815	42224	460	2741	23101	0	0	358.634
8	455	21	21	560	18917	29851	480	2683	17153	0	0	376.428
9	502	36	36	704	8334	65358	773	17392	55479	0	0	351.774
10	467	3	3	689	4014	24027	694	4616	25692	0	0	155.146
11	289	30	31	640	15536	142947	618	9017	123590	0	0	614.419
12	149	25	29	390	1527	15945	400	2845	25932	0	0	45.1968
13	446	9	9	559	5902	21166	566	8287	23443	0	0	178.219
14	451	73	73	566	22551	99701	541	8261	74280	0	0	575.372
15	416	25	25	492	18135	63489	491	12793	54384	0	0	401.293
16	236	31	31	502	13182	48981	481	9055	39355	0	0	262.65
17	286	40	45	615	5052	48022	642	8498	65726	0	0	229.699
18	155	29	29	419	21787	60244	390	3549	32433	0	0	292.508
19	438	9	9	794	1932	5505	826	1883	5147	0	0	40.3646
20	504	45	45	780	9274	68818	802	9919	67323	0	0	395.419
21	285	44	45	603	16062	101346	589	10700	86077	0	0	407.186
22	404	30	31	904	8031	36103	893	6772	32816	0	0	260.657
23	467	26	27	932	3545	13033	937	5124	16287	0	0	95.8239
24	177	22	24	507	17114	53482	498	6925	55498	0	0	239.368
25	510	3	3	813	5228	21760	809	4478	16563	0	0	134.73
26	308	26	26	800	7398	53898	768	4810	68585	0	0	267.464
27	253	36	40	603	18143	107420	572	5905	58920	0	0	433.764
28	273	26	26	716	10168	38196	696	6638	34925	0	0	224.432
29	512	29	29	658	5664	43585	668	7692	51314	0	0	253.434
30	445	3	3	686	6203	46603	744	14920	42588	0	0	264.122
1				589	24050	94565	589	21256	92932	0	0	615.721
2				519	22062	46511	519	5257	18608	0	0	404.734
3				674	2676	13319	674	1326	6432	0	0	88.853
4				529	10628	33669	462	3465	23932	0	0	307.731
5				527	5586	36639	527	3895	47843	0	0	232.447
6				699	13307	84630	659	4652	44235	0	0	484.283
7				541	18815	42224	460	2690	23101	0	0	365.751
8				560	18917	29851	480	2606	17153	0	0	388.977
9				704	8334	65358	704	5692	34582	0	0	350.204
10				689	4014	24027	689	2477	24877	0	0	154.564
11				640	15536	142947	618	7204	123590	0	0	626.700
12				390	1576	15945	390	1269	17112	0	0	57.017
13				559	5902	21166	559	2618	21596	0	0	190.557
14				566	22551	99701	541	6050	74280	0	0	582.469
15				492	18135	63489	491	8180	54384	0	0	476.764
16				502	13182	48981	481	7223	39355	0	0	298.378
17				615	5052	48022	615	3445	46435	0	0	272.547

18				419	23864	60244	390	2795	32433	0	0	385.269
19				794	1932	5505	794	1363	4882	0	0	40.006
20				780	9274	68818	780	5076	59945	0	0	403.549
21				603	16062	101346	589	8034	86077	0	0	439.086
22				904	8031	36103	893	5646	32816	0	0	277.263
23				932	3545	13033	932	4621	16102	0	0	98.131
24				507	21476	53482	498	6472	55498	0	0	397.211
25				813	5228	21760	809	4364	16563	0	0	134.986
26				800	7409	53898	768	4344	68585	0	0	327.043
27				603	18144	107420	572	3767	58920	0	0	508.870
28				716	10169	38196	696	6295	34925	0	0	307.623
29				658	5664	43585	658	4123	41322	0	0	262.269
30				686	6203	46603	686	3693	28209	0	0	272.690
1	480	55	55	589	24050	94565	589	21256	92932	0	0	618.642
2	428	3	3	519	22062	46511	519	5257	18608	0	0	403.795
3	478	3	3	674	2676	13319	674	1326	6432	0	0	88.4314
4	432	3	3	529	10628	33669	462	3465	23932	0	0	287.208
5	340	54	54	527	5586	36639	527	3895	47843	0	0	225.111
6	393	44	44	699	13307	84630	659	4652	44235	0	0	465.896
7	449	3	3	541	18815	42224	460	2690	23101	0	0	353.602
8	455	21	21	560	18917	29851	480	2606	17153	0	0	374.875
9	502	36	36	704	8334	65358	704	5692	34582	0	0	353.745
10	467	3	3	689	4014	24027	689	2477	24877	0	0	153.697
11	289	30	31	640	15536	142947	618	7204	123590	0	0	612.037
12	149	25	29	390	1527	15945	390	1269	17112	0	0	45.2734
13	446	9	9	559	5902	21166	559	2618	21596	0	0	177.372
14	451	73	73	566	22551	99701	541	6050	74280	0	0	549.064
15	416	25	25	492	18135	63489	491	8180	54384	0	0	399.991
16	236	31	31	502	13182	48981	481	7223	39355	0	0	260.131
17	286	40	45	615	5052	48022	615	3445	46435	0	0	297.562
18	155	29	29	419	21787	60244	390	2795	32433	0	0	291.925
19	438	9	9	794	1932	5505	794	1363	4882	0	0	40.0235
20	504	45	45	780	9274	68818	780	5076	59945	0	0	389.763
21	285	44	45	603	16062	101346	589	8034	86077	0	0	404.961
22	404	30	31	904	8031	36103	893	5646	32816	0	0	259.292
23	467	26	27	932	3545	13033	932	4621	16102	0	0	95.637
24	177	22	24	507	17114	53482	498	6472	55498	0	0	240.363
25	510	3	3	813	5228	21760	809	4364	16563	0	0	133.696
26	308	26	26	800	7398	53898	768	4344	68585	0	0	258.852
27	253	36	40	603	18143	107420	572	3767	58920	0	0	430.726
28	273	26	26	716	10168	38196	696	6295	34925	0	0	221.702
29	512	29	29	658	5664	43585	658	4123	41322	0	0	251.442
30	445	3	3	686	6203	46603	686	3693	28209	0	0	259.983

ANNEX 2.2. Results for 3 products and 5 tanks, initial sequence of products ACB

#	C^0	Ev^0	Ov^0	C^1	Ev^1	Ov^1	C^2	Ev^2	Ov^2	m	s	ms
1				589	24050	94565	593	27912	96729	0	0	609.689
2				519	22062	46511	533	11023	24564	0	0	377.005
3				674	2676	13319	742	3432	12116	0	0	86.219
4				529	10628	33669	462	3603	23932	0	0	288.522
5				527	5586	36639	575	27943	110170	0	0	223.514
6				699	13307	84630	659	5583	44235	0	0	472.200
7				541	18815	42224	460	2741	23101	0	0	351.219
8				560	18917	29851	480	2683	17153	0	0	376.444
9				704	8334	65358	773	17392	55479	0	0	345.657
10				689	4014	24027	694	4616	25692	0	0	153.335
11				640	15536	142947	618	9017	123590	0	0	632.135
12				390	1576	15945	400	2886	25932	0	0	56.952
13				559	5902	21166	566	8287	23443	0	0	176.140
14				566	22551	99701	541	8261	74280	0	0	672.639
15				492	18135	63489	491	12793	54384	0	0	417.220
16				502	13182	48981	481	9055	39355	0	0	297.958
17				615	5052	48022	642	8498	65726	0	0	253.631
18				419	23864	60244	390	3803	32433	0	0	378.615
19				794	1932	5505	826	1883	5147	0	0	41.658
20				780	9274	68818	802	9919	67323	0	0	386.356
21				603	16062	101346	589	10700	86077	0	0	415.333
22				904	8031	36103	893	6772	32816	0	0	275.844
23				932	3545	13033	937	5124	16287	0	0	94.911
24				507	21476	53482	498	7686	55498	0	0	379.378
25				813	5228	21760	809	4478	16563	0	0	133.445
26				800	7409	53898	768	4811	68585	0	0	311.049
27				603	18144	107420	572	5906	58920	0	0	481.033
28				716	10169	38196	696	6639	34925	0	0	294.846
29				658	5664	43585	668	7692	51314	0	0	253.156
30				686	6203	46603	744	14920	42588	0	0	262.376
1	480	55	55	589	24050	94565	593	27912	96729	0	0	633.185
2	428	3	3	519	22062	46511	533	11023	24564	0	0	380.013
3	478	3	3	674	2676	13319	742	3432	12116	0	0	86.1788
4	432	3	3	529	10628	33669	462	3603	23932	0	0	289.914
5	340	54	54	527	5586	36639	575	27943	110170	0	0	227.572
6	393	44	44	699	13307	84630	659	5583	44235	0	0	477.149
7	449	3	3	541	18815	42224	460	2741	23101	0	0	358.634
8	455	21	21	560	18917	29851	480	2683	17153	0	0	376.428
9	502	36	36	704	8334	65358	773	17392	55479	0	0	351.774
10	467	3	3	689	4014	24027	694	4616	25692	0	0	155.146
11	289	30	31	640	15536	142947	618	9017	123590	0	0	614.419
12	149	25	29	390	1527	15945	400	2845	25932	0	0	45.1968
13	446	9	9	559	5902	21166	566	8287	23443	0	0	178.219
14	451	73	73	566	22551	99701	541	8261	74280	0	0	575.372

15	416	25	25	492	18135	63489	491	12793	54384	0	0	401.293
16	236	31	31	502	13182	48981	481	9055	39355	0	0	262.65
17	286	40	45	615	5052	48022	642	8498	65726	0	0	229.699
18	155	29	29	419	21787	60244	390	3549	32433	0	0	292.508
19	438	9	9	794	1932	5505	826	1883	5147	0	0	40.3646
20	504	45	45	780	9274	68818	802	9919	67323	0	0	395.419
21	285	44	45	603	16062	101346	589	10700	86077	0	0	407.186
22	404	30	31	904	8031	36103	893	6772	32816	0	0	260.657
23	467	26	27	932	3545	13033	937	5124	16287	0	0	95.8239
24	177	22	24	507	17114	53482	498	6925	55498	0	0	239.368
25	510	3	3	813	5228	21760	809	4478	16563	0	0	134.73
26	308	26	26	800	7398	53898	768	4810	68585	0	0	267.464
27	253	36	40	603	18143	107420	572	5905	58920	0	0	433.764
28	273	26	26	716	10168	38196	696	6638	34925	0	0	224.432
29	512	29	29	658	5664	43585	668	7692	51314	0	0	253.434
30	445	3	3	686	6203	46603	744	14920	42588	0	0	264.122
1				589	24050	94565	589	21256	92932	0	0	615.721
2				519	22062	46511	519	5257	18608	0	0	404.734
3				674	2676	13319	674	1326	6432	0	0	88.853
4				529	10628	33669	462	3465	23932	0	0	307.731
5				527	5586	36639	527	3895	47843	0	0	232.447
6				699	13307	84630	659	4652	44235	0	0	484.283
7				541	18815	42224	460	2690	23101	0	0	365.751
8				560	18917	29851	480	2606	17153	0	0	388.977
9				704	8334	65358	704	5692	34582	0	0	350.204
10				689	4014	24027	689	2477	24877	0	0	154.564
11				640	15536	142947	618	7204	123590	0	0	626.700
12				390	1576	15945	390	1269	17112	0	0	57.017
13				559	5902	21166	559	2618	21596	0	0	190.557
14				566	22551	99701	541	6050	74280	0	0	582.469
15				492	18135	63489	491	8180	54384	0	0	476.764
16				502	13182	48981	481	7223	39355	0	0	298.378
17				615	5052	48022	615	3445	46435	0	0	272.547
18				419	23864	60244	390	2795	32433	0	0	385.269
19				794	1932	5505	794	1363	4882	0	0	40.006
20				780	9274	68818	780	5076	59945	0	0	403.549
21				603	16062	101346	589	8034	86077	0	0	439.086
22				904	8031	36103	893	5646	32816	0	0	277.263
23				932	3545	13033	932	4621	16102	0	0	98.131
24				507	21476	53482	498	6472	55498	0	0	397.211
25				813	5228	21760	809	4364	16563	0	0	134.986
26				800	7409	53898	768	4344	68585	0	0	327.043
27				603	18144	107420	572	3767	58920	0	0	508.870
28				716	10169	38196	696	6295	34925	0	0	307.623
29				658	5664	43585	658	4123	41322	0	0	262.269
30				686	6203	46603	686	3693	28209	0	0	272.690

1	480	55	55	589	24050	94565	589	21256	92932	0	0	618.642
2	428	3	3	519	22062	46511	519	5257	18608	0	0	403.795
3	478	3	3	674	2676	13319	674	1326	6432	0	0	88.4314
4	432	3	3	529	10628	33669	462	3465	23932	0	0	287.208
5	340	54	54	527	5586	36639	527	3895	47843	0	0	225.111
6	393	44	44	699	13307	84630	659	4652	44235	0	0	465.896
7	449	3	3	541	18815	42224	460	2690	23101	0	0	353.602
8	455	21	21	560	18917	29851	480	2606	17153	0	0	374.875
9	502	36	36	704	8334	65358	704	5692	34582	0	0	353.745
10	467	3	3	689	4014	24027	689	2477	24877	0	0	153.697
11	289	30	31	640	15536	142947	618	7204	123590	0	0	612.037
12	149	25	29	390	1527	15945	390	1269	17112	0	0	45.2734
13	446	9	9	559	5902	21166	559	2618	21596	0	0	177.372
14	451	73	73	566	22551	99701	541	6050	74280	0	0	549.064
15	416	25	25	492	18135	63489	491	8180	54384	0	0	399.991
16	236	31	31	502	13182	48981	481	7223	39355	0	0	260.131
17	286	40	45	615	5052	48022	615	3445	46435	0	0	297.562
18	155	29	29	419	21787	60244	390	2795	32433	0	0	291.925
19	438	9	9	794	1932	5505	794	1363	4882	0	0	40.0235
20	504	45	45	780	9274	68818	780	5076	59945	0	0	389.763
21	285	44	45	603	16062	101346	589	8034	86077	0	0	404.961
22	404	30	31	904	8031	36103	893	5646	32816	0	0	259.292
23	467	26	27	932	3545	13033	932	4621	16102	0	0	95.637
24	177	22	24	507	17114	53482	498	6472	55498	0	0	240.363
25	510	3	3	813	5228	21760	809	4364	16563	0	0	133.696
26	308	26	26	800	7398	53898	768	4344	68585	0	0	258.852
27	253	36	40	603	18143	107420	572	3767	58920	0	0	430.726
28	273	26	26	716	10168	38196	696	6295	34925	0	0	221.702
29	512	29	29	658	5664	43585	658	4123	41322	0	0	251.442
30	445	3	3	686	6203	46603	686	3693	28209	0	0	259.983

ANNEX 2.3. Results for 3 products and 6 tanks, initial sequence of products ABC

#	C^0	Ev^0	Ov^0	C^1	Ev^1	Ov^1	C^2	Ev^2	Ov^2	m	s	ms
1				704	121663	503113	698	98327	348988	0	7	573.43
2				513	121629	610686	505	70925	565008	0	5	722.68
3				491	131968	1340958	525	518646	2696209	0	44	133.32
4				535	371280	1489665	542	323498	1370504	0	25	117.93
5				529	335099	1142517	545	370899	1139101	0	19	818.93
6				691	54454	170866	680	24687	89273	0	2	801.58
7				1002	110861	240626	963	81371	140401	0	3	949.27
8				1013	39709	66963	896	19006	41679	0	1	479.15
9				856	253203	1495647	830	154472	834523	0	24	54.463
10				844	24899	111432	841	20528	52425	0	1	287.26
11				1007	117680	172975	1003	139258	236090	0	3	589.61
12				830	39145	180175	853	42219	217465	0	2	464.92
13				990	38044	91099	1006	26463	60373	0	1	261.7
14				866	6975	25869	859	7400	20493	0	0	197.78
15				689	104052	428874	715	107270	333218	0	4	309.24
1	578	9	9	704	121663	503113	698	98327	348988	0	7	422.33
2	484	3	3	513	121629	610686	505	70925	565008	0	5	680.13
3	469	3	3	491	131968	1340958	525	518646	2696209	0	17	6.4519
4	271	88	88	535	371280	1489665	542	323498	1370504	0	21	649.85
5	254	62	63	529	335099	1142517	545	370899	1139101	0	15	445
6	316	26	28	691	54454	170866	680	24687	89273	0	2	68.228
7	351	36	37	1002	110202	240626	963	79958	140401	0	2	182.79
8	600	17	17	1013	39709	66963	896	19006	41679	0	1	461.22
9	331	51	57	856	244711	1495647	830	154466	834523	0	18	47.873
10	578	24	24	844	24899	111432	841	20528	52425	0	1	239.99
11	594	54	54	1007	117680	172975	1003	139258	236090	0	3	651.25
12	553	48	48	830	39145	180175	853	42219	217465	0	2	492.95
13	592	3	3	990	38044	91099	1006	26463	60373	0	1	302.98
14	471	3	3	866	6975	25869	859	7400	20493	0	0	199.41
15	546	3	3	689	104052	428874	715	107270	333218	0	4	415.98
1				704	121663	503113	698	78004	348988	0	7	380.86
2				513	121629	610686	505	68981	565008	0	6	38.016
3				491	131968	1340958	491	120434	1257158	0	16	312.37
4				535	371280	1489665	535	242608	1251815	0	24	48.6
5				529	335099	1142517	529	168065	895488	0	19	244.01
6				691	54454	170866	680	18897	89273	0	2	779.54
7				1002	110861	240626	963	73108	140401	0	3	906.91
8				1013	39709	66963	896	18988	41679	0	1	509.73
9				856	253203	1495647	830	117568	834523	0	22	855.39
10				844	24899	111432	841	14251	52425	0	1	251.8
11				1007	117680	172975	1003	120432	236090	0	3	612.96
12				830	39145	180175	830	23095	159902	0	2	328.06
13				990	38044	91099	990	18822	57049	0	1	426.5
14				866	6975	25869	859	6178	20493	0	0	199.57

15				689	104052	428874	689	61601	228771	0	4	370.61
1	578	9	9	704	121663	503113	698	78004	348988	0	7	309.53
2	484	3	3	513	121629	610686	505	68981	565008	0	5	802.55
3	469	3	3	491	131968	1340958	491	120434	1257158	0	16	336.25
4	271	88	88	535	371280	1489665	535	242608	1251815	0	21	908.4
5	254	62	63	529	335099	1142517	529	168065	895488	0	15	749.72
6	316	26	28	691	54454	170866	680	18897	89273	0	2	54.65
7	351	36	37	1002	110202	240626	963	73108	140401	0	2	208.99
8	600	17	17	1013	39709	66963	896	18988	41679	0	1	448.27
9	331	51	57	856	244711	1495647	830	117568	834523	0	18	55.027
10	578	24	24	844	24899	111432	841	14251	52425	0	1	217.96
11	594	54	54	1007	117680	172975	1003	120432	236090	0	3	578.36
12	553	48	48	830	39145	180175	830	23095	159902	0	2	299.12
13	592	3	3	990	38044	91099	990	18822	57049	0	1	424.34
14	471	3	3	866	6975	25869	859	6178	20493	0	0	199.21
15	546	3	3	689	104052	428874	689	61601	228771	0	4	384.63

ANNEX 2.4. Results for 3 products and 6 tanks, initial sequence of products ACB

#	C^0	Ev^0	Ov^0	C^1	Ev^1	Ov^1	C^2	Ev^2	Ov^2	m	s	ms
1				698	98327	348988	704	121663	503113	0	5	576.05
2				505	70925	565008	513	121629	610686	0	4	187.93
3				525	518646	2696209	491	131968	1340958	1	47	342.55
4				542	323498	1370504	535	371280	1489665	0	18	385.23
5				545	370899	1139101	529	335099	1142517	0	15	390.03
6				680	24687	89273	691	54454	170866	0	1	137.10
7				963	81371	140401	1002	110861	240626	0	4	145.70
8				896	19006	41679	1013	39709	66963	0	0	687.72
9				830	154472	834523	856	253203	1495647	0	10	205.94
10				841	20528	52425	844	24899	111432	0	0	987.29
11				1003	139258	236090	1007	117680	172975	0	4	623.13
12				853	42219	217465	830	39145	180175	0	2	581.50
13				1006	26463	60373	990	38044	91099	0	0	823.48
14				859	7400	20493	866	6975	25869	0	0	203.14
15				715	107270	333218	689	104052	428874	0	4	474.20
1	578	9	9	698	98327	348988	704	121663	503.113	0	5	600.37
2	484	3	3	505	70925	565008	513	121629	610.686	0	4	152.05
3	469	3	3	525	518646	2696209	491	131968	1.340.958	1	48	657.64
4	271	88	88	542	323498	1370504	535	371280	1.489.665	0	17	272.83
5	254	62	63	545	370899	1139101	529	335099	1.142.517	0	13	775.03
6	316	26	28	680	24687	89273	691	54454	170.866	0	0	983.53
7	351	36	37	963	79958	140401	1002	110202	240.626	0	1	703.98
8	600	17	17	896	19006	41679	1013	39709	66.963	0	0	687.95
9	331	51	57	830	154466	834523	856	244711	1.495.647	0	8	164.64
10	578	24	24	841	20528	52425	844	24899	111432	0	0	988.48

11	594	54	54	1003	139258	236090	1007	117680	172975	0	4	640.19
12	553	48	48	853	42219	217465	830	39145	180175	0	2	578.77
13	592	3	3	1006	26463	60373	990	38044	91099	0	0	826.46
14	471	3	3	859	7400	20493	866	6975	25869	0	0	198.70
15	546	3	3	715	107270	333218	689	104052	428874	0	4	509.79
1				698	98327	348988	698	105527	466210	0	5	719.84
2				505	70925	565008	505	95838	485468	0	4	312.78
3				525	518646	2696209	491	126224	1340958	1	42	91.55
4				542	323498	1370504	535	314504	1489665	0	18	412.75
5				545	370899	1139101	529	250726	1142517	0	15	390.28
6				680	24687	89273	680	29280	147022	0	1	131.07
7				963	81371	140401	963	69132	205028	0	4	106.76
8				896	19006	41679	896	17606	39938	0	0	697.48
9				830	154472	834523	830	82540	934750	0	10	237.43
10				841	20528	52425	841	16166	106998	0	0	998.38
11				1003	139258	236090	1003	106342	171169	0	4	550.99
12				853	42219	217465	830	32689	180175	0	2	421.03
13				1006	26463	60373	990	35552	91099	0	0	810.90
14				859	7400	20493	859	4856	25285	0	0	195.85
15				715	107270	333218	689	101935	428874	0	4	543.83
1	578	9	9	698	98327	348988	698	105527	466210	0	5	549.31
2	484	3	3	505	70925	565008	505	95838	485468	0	4	360.55
3	469	3	3	525	518646	2696209	491	126224	1340958	1	17	48.99
4	271	88	88	542	323498	1370504	535	314504	1489665	0	17	113.84
5	254	62	63	545	370899	1139101	529	250726	1142517	0	13	331.84
6	316	26	28	680	24687	89273	680	29280	147022	0	0	989.61
7	351	36	37	963	79958	140401	963	69132	205028	0	1	820.86
8	600	17	17	896	19006	41679	896	17606	39938	0	0	689.98
9	331	51	57	830	154466	834523	830	82540	934750	0	8	540.10
10	578	24	24	841	20528	52425	841	16166	106998	0	0	982.08
11	594	54	54	1003	139258	236090	1003	106342	171169	0	4	585.78
12	553	48	48	853	42219	217465	830	32689	180175	0	2	420.42
13	592	3	3	1006	26463	60373	990	35552	91099	0	0	810.52
14	471	3	3	859	7400	20493	859	4856	25285	0	0	196.02
15	546	3	3	715	107270	333218	689	101935	428874	0	4	433.76

ANNEX 2.5. Results for 4 products and 4 tanks, sequence of products ABCD and others

#	C^0	Ev^0	Ov^0	C^1	Ev^1	Ov^1	C^2	Ev^2	Ov^2	C^3	Ev^3	Ov^3	C^4	Ev^4	Ov^4	C^5	Ev^5	Ov^5	C^6	Ev^6	Ov^6	m	s	ms
1				583	4355	16043	581	2186	13942	576	1481	10061	575	3020	7945	583	1729	10827	578	1280	7264	0	0	293.94
2				661	2347	61215	665	2335	58079	694	6887	67578	672	2465	58345	654	2788	60755	665	1958	60177	0	0	192.629
3				576	1481	10061	575	3020	7945	583	1729	10827	578	1280	7264	583	4355	16043	581	2186	13942	0	0	84.7812
4				516	4301	27257	518	5802	31120	534	10187	36634	508	1787	17605	518	6554	27897	518	3896	27592	0	0	87.517
5				844	6312	102113	831	2343	78661	840	2292	88055	850	5033	102060	820	1771	71860	834	1849	69283	0	0	357.853
6				970	3276	35104	999	8894	54936	944	3675	26575	982	6393	61472	978	4022	33490	944	4428	27873	0	0	161.058
7				742	4982	43838	747	4857	28399	714	1386	17322	764	6339	39387	728	1524	31139	766	6976	39458	0	0	236.027
8				870	9082	64588	832	2368	48365	851	6622	55840	864	7210	62702	863	6293	61280	852	7119	63334	0	0	255.506
9				629	4123	27314	618	2994	25175	618	1584	25284	630	3571	25526	633	2912	32395	633	2480	25588	0	0	144.585
10				728	1854	38934	737	4456	52628	763	13639	63334	755	9116	51411	753	9689	48831	742	5314	44309	0	0	128.103
11				490	6766	14973	486	6278	15524	466	1981	6658	482	5054	10056	464	2538	6035	445	743	3189	0	0	79.5408
12				450	3509	19856	477	9234	34215	449	3335	15821	461	8115	24403	462	4436	21783	451	3383	20754	0	0	75.1573
13				450	5018	29216	448	2058	34503	444	921	20983	459	3976	37053	469	5706	32786	468	7678	42084	0	0	174.001
14				443	512	8323	444	606	12048	458	3393	16338	459	5514	14770	467	8598	17174	460	6327	15330	0	0	23.184
15				444	1621	30476	444	1026	25773	449	2248	39899	448	3403	37514	462	6872	43942	461	5107	37122	0	0	127.442
1	363	9	9	583	4355	16043	581	2186	13942	576	1481	10061	575	3020	7945	583	1729	10827	578	1280	7264	0	0	379.234
2	206	20	21	661	2347	61215	665	2335	58079	694	6887	67578	672	2465	58345	654	2788	60755	665	1958	60177	0	0	178.252
3	363	9	9	576	1481	10061	575	3020	7945	583	1729	10827	578	1280	7264	583	4355	16043	581	2186	13942	0	0	85.3743
4	340	25	25	516	4301	27257	518	5802	31120	534	10187	36634	508	1787	17605	518	6554	27897	518	3896	27592	0	0	88.9778
5	230	16	17	844	5498	102113	831	2255	78661	840	2212	88055	850	4830	102060	820	1697	71860	834	1776	69283	0	0	293.329
6	480	17	17	970	3276	35104	999	8894	54936	944	3675	26575	982	6393	61472	978	4022	33490	944	4428	27873	0	0	152.259
7	411	3	3	742	4982	43838	747	4857	28399	714	1386	17322	764	6339	39387	728	1524	31139	766	6976	39458	0	0	234.435
8	441	3	3	870	9082	64588	832	2368	48365	851	6622	55840	864	7210	62702	863	6293	61280	852	7119	63334	0	0	258.863
9	164	12	14	629	1936	27314	618	1398	25175	618	743	25284	630	2194	25526	633	2125	32395	633	1833	25588	0	0	77.4196
10	225	20	21	728	1842	38934	737	4446	52628	763	13624	63334	755	9103	51411	753	9673	48831	742	5303	44309	0	0	118.876
11	322	3	3	490	6766	14973	486	6278	15524	466	1981	6658	482	5054	10056	464	2538	6035	445	743	3189	0	0	81.6797
12	347	3	3	450	3509	19856	477	9234	34215	449	3335	15821	461	8115	24403	462	4436	21783	451	3383	20754	0	0	77.4199
13	128	13	14	450	3739	29216	448	1999	34503	444	864	20983	459	3887	37053	469	5303	32786	468	6653	42084	0	0	91.2381
14	339	3	3	443	512	8323	444	606	12048	458	3393	16338	459	5514	14770	467	8598	17174	460	6327	15330	0	0	23.8902
15	128	13	14	444	1252	30476	444	981	25773	449	2202	39899	448	3168	37514	462	5494	43942	461	4878	37122	0	0	82.9432

1				583	4355	16043	581	610	13942	576	553	10061	575	507	7945	575	567	8164	575	519	6544	0	0	383.23
2				661	2347	61215	661	1298	56755	661	1179	49123	661	1317	53232	654	1301	60755	654	788	53994	0	0	196.925
3				576	1481	10061	575	507	7945	575	567	8164	575	519	6544	575	593	12435	575	568	10957	0	0	85.0728
4				516	4301	27257	516	2253	30271	516	2303	28189	508	1448	17605	508	1480	21502	508	1315	21895	0	0	88.0134
5				844	6312	102113	831	1732	78661	831	1386	85133	831	1527	90375	820	1048	71860	820	867	63234	0	0	368.464
6				970	3276	35104	970	3540	48330	944	1395	26575	944	1779	49818	944	1166	28277	944	1208	27873	0	0	158.772
7				742	4982	43838	742	1732	26275	714	679	17322	714	552	17629	714	586	23842	714	560	20090	0	0	236.454
8				870	9082	64588	832	2099	48365	832	984	48218	832	930	48445	832	1002	48433	832	891	54983	0	0	262.931
9				629	4123	27314	618	893	25175	618	567	25284	618	509	22198	618	582	25805	618	501	21565	0	0	148.969
10				728	1854	38934	728	1184	47304	728	1169	46116	728	951	38887	728	947	39332	728	961	38053	0	0	133.178
11				490	6766	14973	486	6079	15524	466	1603	6658	466	1448	6620	464	1219	6035	445	389	3189	0	0	80.601
12				450	3509	19856	450	1161	15223	449	1020	15821	449	1110	15538	449	1051	13662	449	1325	19408	0	0	75.5294
13				450	5018	29216	448	1709	34503	444	562	20983	444	545	25800	444	485	17132	444	528	22148	0	0	180.82
14				443	512	8323	443	357	11424	443	405	9906	443	337	9807	443	359	8167	443	320	8585	0	0	23.2046
15				444	1621	30476	444	627	25773	444	755	36252	444	695	33613	444	655	31667	444	619	25769	0	0	132.09
1	363	9	9	583	4355	16043	581	610	13942	576	553	10061	575	507	7945	575	567	8164	575	519	6544	0	0	336.346
2	206	20	21	661	2347	61215	661	1298	56755	661	1179	49123	661	1317	53232	654	1301	60755	654	788	53994	0	0	170.616
3	363	9	9	576	1481	10061	575	507	7945	575	567	8164	575	519	6544	575	593	12435	575	568	10957	0	0	86.6554
4	340	25	25	516	4301	27257	516	2253	30271	516	2303	28189	508	1448	17605	508	1480	21502	508	1315	21895	0	0	88.9633
5	230	16	17	844	5498	102113	831	1732	78661	831	1386	85133	831	1527	90375	820	1048	71860	820	867	63234	0	0	298.625
6	480	17	17	970	3276	35104	970	3540	48330	944	1395	26575	944	1779	49818	944	1166	28277	944	1208	27873	0	0	153.811
7	411	3	3	742	4982	43838	742	1732	26275	714	679	17322	714	552	17629	714	586	23842	714	560	20090	0	0	235.135
8	441	3	3	870	9082	64588	832	2099	48365	832	984	48218	832	930	48445	832	1002	48433	832	891	54983	0	0	251.996
9	164	12	14	629	1936	27314	618	893	25175	618	567	25284	618	509	22198	618	582	25805	618	501	21565	0	0	74.1175
10	225	20	21	728	1842	38934	728	1184	47304	728	1169	46116	728	951	38887	728	947	39332	728	961	38053	0	0	112.985
11	322	3	3	490	6766	14973	486	6079	15524	466	1603	6658	466	1448	6620	464	1219	6035	445	389	3189	0	0	79.8948
12	347	3	3	450	3509	19856	450	1161	15223	449	1020	15821	449	1110	15538	449	1051	13662	449	1325	19408	0	0	76.1641
13	128	13	14	450	3739	29216	448	1709	34503	444	562	20983	444	545	25800	444	485	17132	444	528	22148	0	0	87.9467
14	339	3	3	443	512	8323	443	357	11424	443	405	9906	443	337	9807	443	359	8167	443	320	8585	0	0	23.1574
15	128	13	14	444	1252	30476	444	627	25773	444	755	36252	444	695	33613	444	655	31667	444	619	25769	0	0	81.1273

ANNEX 2.6. Results for 4 products and 5 tanks, sequence of products ABCD and others

#	C^0	Ev^0	Ov^0	C^1	Ev^1	Ov^1	C^2	Ev^2	Ov^2	C^3	Ev^3	Ov^3	C^4	Ev^4	Ov^4	C^5	Ev^5	Ov^5	C^6	Ev^6	Ov^6	m	s	ms
1				752	211728	1000052	765	328831	1000078	744	105293	1000025	736	88239	1000101	769	496902	1000082	762	296976	1000274	0	15	620.932
2				635	75000	1000498	650	194851	1000238	636	77363	1000327	647	75309	1000481	658	123181	1000494	627	58015	974884	0	8	911.264
3				689	254048	1000451	653	135254	1000344	694	422910	1000127	654	105046	1000204	618	50497	876066	632	70554	983743	0	15	389.897
4				738	95107	1000412	758	155949	1000004	754	294396	1000350	755	169735	1000224	756	130449	1000020	763	290464	1000261	0	8	805.773
5				659	102788	1000047	664	148713	1000209	618	43035	893859	655	97494	1000312	686	179305	1000069	659	90972	615514	0	7	996.929
6				723	78662	751352	737	74527	931398	745	102422	1000533	741	71697	841977	741	83642	756143	743	145114	822252	0	7	841.096
7				847	226633	1000097	864	394409	1000237	837	108717	1000386	821	89381	1000173	821	130290	1000485	837	147006	1000431	0	13	827.354
8				745	102422	1000533	741	71697	841977	741	83642	756143	743	145114	822252	723	78662	751352	737	74527	931398	0	8	497.646
9				645	285986	1000058	690	748769	1000280	647	124033	1000109	637	99057	1000528	656	249090	1000287	670	299705	1000279	0	12	702.285
10				1053	170046	1000162	1085	170855	1000053	1077	144717	1000190	1089	183997	1000154	1077	173812	1000287	1079	150896	1000019	0	12	394.067
11				1211	67443	537663	1231	135342	600766	1170	45642	238624	1213	93444	575948	1220	59523	483332	1152	49909	172102	0	4	513.095
12				942	156162	1000436	959	165433	1000173	926	107721	946007	994	390665	1000246	956	366876	1000299	951	149232	1000189	0	10	763.777
13				1052	96812	949673	1031	85899	1000044	1032	81194	1000349	1057	98567	890678	1036	90224	1000118	1053	93751	930897	0	8	431.308
14				794	197749	1000261	778	93667	1000168	782	130450	1000514	792	229801	1000047	810	426291	1000256	806	281601	1000183	0	11	334.074
15				963	227407	1000454	954	212039	1000034	927	150734	1000499	952	198967	1000170	951	214073	1000538	916	93455	989835	0	11	457.165
1	480	55	55	752	211728	1000052	765	328831	1000078	744	105293	1000025	736	88239	1000101	769	496902	1000082	762	296976	1000274	0	16	527.156
2	447	3	3	635	75000	1000498	650	194851	1000238	636	77363	1000327	647	75309	1000481	658	123181	1000494	627	58015	974884	0	7	833.791
3	452	3	3	689	254048	1000451	653	135254	1000344	694	422910	1000127	654	105046	1000204	618	50497	876066	632	70554	983743	0	14	665.954
4	456	3	3	738	95107	1000412	758	155949	1000004	754	294396	1000350	755	169735	1000224	756	130449	1000020	763	290464	1000261	0	8	708.572
5	479	3	3	659	102788	1000047	664	148713	1000209	618	43035	893859	655	97494	1000312	686	179305	1000069	659	90972	615514	0	8	237.094
6	453	9	9	723	78662	751352	737	74527	931398	745	102422	1000533	741	71697	841977	741	83642	756143	743	145114	822252	0	7	709.401
7	379	34	35	847	226633	1000097	864	394409	1000237	837	108717	1000386	821	89381	1000173	821	130290	1000485	837	147006	1000431	0	13	588.190
8	453	9	9	745	102422	1000533	741	71697	841977	741	83642	756143	743	145114	822252	723	78662	751352	737	74527	931398	0	8	424.125
9	416	25	25	645	285986	1000058	690	748769	1000280	647	124033	1000109	637	99057	1000528	656	249090	1000287	670	299705	1000279	0	12	740.701
10	508	36	36	1053	170046	1000162	1085	170855	1000053	1077	144717	1000190	1089	183997	1000154	1077	173812	1000287	1079	150896	1000019	0	11	990.266
11	513	26	27	1211	67443	537663	1231	135342	600766	1170	45642	238624	1213	93444	575948	1220	59523	483332	1152	49909	172102	0	4	443.212
12	497	3	3	942	156162	1000436	959	165433	1000173	926	107721	946007	994	390665	1000246	956	366876	1000299	951	149232	1000189	0	11	391.741
13	545	3	3	1052	96812	949673	1031	85899	1000044	1032	81194	1000349	1057	98567	890678	1036	90224	1000118	1053	93751	930897	0	8	432.530
14	285	44	45	794	197749	1000261	778	93667	1000168	782	130450	1000514	792	229801	1000047	810	426291	1000256	806	281601	1000183	0	9	246.442
15	507	35	35	963	227407	1000454	954	212039	1000034	927	150734	1000499	952	198967	1000170	951	214073	1000538	916	93455	989835	0	11	668.549

1				752	211728	1000052	752	125268	1000078	744	98585	1000025	736	79233	1000101	736	83725	1000082	736	81724	1000274	0	16	366.125
2				635	75000	1000498	635	46442	1000238	635	43116	1000327	635	41879	999881	635	42597	1000494	627	36457	974884	0	8	205.012
3				689	254048	1000451	653	121589	1000344	653	51938	607054	653	80845	1000204	618	44400	876066	618	39710	709158	0	14	880.099
4				738	95107	1000412	738	85497	1000004	738	84647	1000350	738	81717	1000224	738	71690	1000020	738	70882	1000261	0	9	665.316
5				659	102788	1000047	659	93960	1000209	618	36727	893859	618	28557	497949	618	20482	211363	618	19519	147858	0	8	191.419
6				723	78662	751352	723	30683	711670	723	32952	756926	723	29552	587592	723	35123	523102	723	30253	545202	0	7	971.245
7				847	226633	1000097	847	104717	1000237	837	87508	1000386	821	76487	1000173	821	90480	1000485	821	82929	1000431	0	14	722.961
8				745	102422	1000533	741	45933	841977	741	51131	756143	741	46705	792991	723	43370	751352	723	30683	711670	0	8	722.472
9				645	285986	1000058	645	109475	1000280	645	110411	1000109	637	94726	1000528	637	90419	1000287	637	97087	1000279	0	13	405.429
10				1053	170046	1000162	1053	105274	1000053	1053	108447	1000190	1053	116422	1000154	1053	105300	1000287	1053	92753	1000019	0	12	796.959
11				1211	67443	537663	1211	66615	535953	1170	40056	238624	1170	61359	440131	1170	35856	356175	1152	37012	172102	0	4	554.119
12				942	156162	1000436	942	97811	994452	926	80956	946007	926	82605	937223	926	110482	1000299	926	80959	882592	0	10	581.021
13				1052	96812	949673	1031	74450	1000044	1031	67227	1000349	1031	65961	707235	1031	65127	1000118	1031	68632	783946	0	8	699.222
14				794	197749	1000261	778	74223	1000168	778	57980	1000514	778	49479	1000047	778	66527	1000256	778	48663	1000183	0	11	429.067
15				963	227407	1000454	954	210173	1000034	927	141447	1000499	927	106651	1000170	927	94262	1000538	916	87746	989835	0	11	546.766
1	480	55	55	752	211728	1000052	752	125268	1000078	744	98585	1000025	736	79233	1000101	736	83725	1000082	736	81724	1000274	0	16	140.545
2	447	3	3	635	75000	1000498	635	46442	1000238	635	43116	1000327	635	41879	999881	635	42597	1000494	627	36457	974884	0	7	962.123
3	452	3	3	689	254048	1000451	653	121589	1000344	653	51938	607054	653	80845	1000204	618	44400	876066	618	39710	709158	0	14	934.197
4	456	3	3	738	95107	1000412	738	85497	1000004	738	84647	1000350	738	81717	1000224	738	71690	1000020	738	70882	1000261	0	8	833.751
5	479	3	3	659	102788	1000047	659	93960	1000209	618	36727	893859	618	28557	497949	618	20482	211363	618	19519	147858	0	7	969.463
6	453	9	9	723	78662	751352	723	30683	711670	723	32952	756926	723	29552	587592	723	35123	523102	723	30253	545202	0	7	876.812
7	379	34	35	847	226633	1000097	847	104717	1000237	837	87508	1000386	821	76487	1000173	821	90480	1000485	821	82929	1000431	0	13	842.998
8	453	9	9	745	102422	1000533	741	45933	841977	741	51131	756143	741	46705	792991	723	43370	751352	723	30683	711670	0	8	505.512
9	416	25	25	645	285986	1000058	645	109475	1000280	645	110411	1000109	637	94726	1000528	637	90419	1000287	637	97087	1000279	0	12	848.227
10	508	36	36	1053	170046	1000162	1053	105274	1000053	1053	108447	1000190	1053	116422	1000154	1053	105300	1000287	1053	92753	1000019	0	11	512.288
11	513	26	27	1211	67443	537663	1211	66615	535953	1170	40056	238624	1170	61359	440131	1170	35856	356175	1152	37012	172102	0	4	499.735
12	497	3	3	942	156162	1000436	942	97811	994452	926	80956	946007	926	82605	937223	926	110482	1000299	926	80959	882592	0	10	452.919
13	545	3	3	1052	96812	949673	1031	74450	1000044	1031	67227	1000349	1031	65961	707235	1031	65127	1000118	1031	68632	783946	0	8	577.578
14	285	44	45	794	197749	1000261	778	74223	1000168	778	57980	1000514	778	49479	1000047	778	66527	1000256	778	48663	1000183	0	9	201.104
15	507	35	35	963	227407	1000454	954	210173	1000034	927	141447	1000499	927	106651	1000170	927	94262	1000538	916	87746	989835	0	11	153.756