ISSN 0005–1144
ATKAAF 50(1–2), 29–38 (2009)

Miquel Angel Piera, Catya Zuñiga, Miguel Mújica

A Pallet Packing CPN Optimization Approach for Distribution Center

UDK 658.788.4
IFAC 5.6.0; 2.8.1

Original scientific paper

The pallet packing process is a critical supply chain component for many distribution centers. Most pallet loading optimization tools provide very good solutions when boxes' dimensions are practically quite uniform, which is the case of most production industrial systems, since they are used to work with a reduced number of master boxes. However, there are other systems such as distribution centers characterized by a high diversity of boxes to be fitted altogether in the same pallet. In this paper a Coloured Petri Net model that formalizes the Pallet Loading problem and its optimization by integrating evaluation methods with search methods will be presented. The model developed provides very good results by using certain heuristics that avoids the analysis of the whole search space, simulating only the best scenarios. The proposed approach can be used to integrate warehouse layout configuration, storage policy, distribution policy together with picking policy. As a result of the proposed strategy, considerable savings on transport resource costs can be achieved.

Key words: Coloured Petri Nets, Constraints, Coverability Tree, Simulation

1 INTRODUCTION

Due to market competitivity, continuous improvements in the design and operation of distribution networks are needed, affecting somehow to new challenges (i.e. lean management philosophies) into warehouse quality factors performance. The well known pallet loading problem (PLP), is considered a critical operation that sometimes must be performed under time and space constraints, and can lead to poor truck/belly occupancy factors. Therefore, it is an important piece in the Supply Chain Management (SCM) which involves different activities beginning with the supply, acquisition and conversion (among others) of unprocessed raw materials and ending with the delivery of the final products to the end users.

The Supply Chain Management approach has shown the necessity to improve the collaboration between production, supplier and the distribution centres. The material flow part of this collaboration requires sometimes palletizing the product, which can force under JIT policies to propagate upstream some requirements in the logistics flow. When the palletizing process is performed corectly, quickly and taking into consideration all the characteristics of the system, costs such as inventory and warehousing among others can be reduced because it is an important link in all those areas of logistics companies, which must prepare orders from customers, placing the packages on pallets in an optimal way for the issue.

Keeping low warehouses has a reduction of the capital associated with them, in this regard; the palletizing plays an important role since a proper configuration of the pallet brings benefits in these strategies. However, in order to obtain benefits in this area an integral vision of the system is needed.

Furthermore, a solver to the palletizing problem that could consider the PLP as a part of a SCM, can create business opportunities in areas such as kitting and sequencing. In kitting, the partner would take parts or materials from bulk supply inventories and put together the necessary combination to be delivered to the production plant for the technician to assemble into a specific product. This is attractive to businesses because it reduces labor and storage requirements on the production line while often improving ergonomics and increasing production speed.

There are several modelling approaches to tackle PLP, some of them are based in container loading ideas which try to construct vertical »walls« across the container [9]. George and Robinson [4] were the first to formulate a heuristic algorithm called the wall-building method for the packing of up to 20 different box types into one container.

Gehring, Menschner and Meyer [3], proposed a method to pack rectangular boxes of different size into a shipping container of known dimensions. This problem specification does not allow for orientation constraints but supports several ranking rules such as box position ranking rule. Bischoff and Dowsland [1] developed a method based on filling the container by building layers across its width. Han, Knotta and Egbelu [5] provided a heuristic based on the dynamic programming method and wall-building method. In his study, the objective was to maximize the volume occupancy without no orientation constrains. The algorithm calls for the boxes to be packed along the base and one vertical wall of the container. After a L-shaped packing is complete, a »new« container can be formed, which become the focus of the successive packing.

Xue and Lai [3], provided another algorithm based on the wall-building method in which both the cargo and container must be rectangular. This algorithm integrated three heuristics:

- The ordering heuristic sorts the cartons according to the depth, quantity and surface area. Higher priority was assigned to cartons with larger values of the above characteristics.
- The placement heuristic determined the depth of the new layer.
- Layer-building heuristic.

Ngoi, Tay and Chua [10], designed a heuristic algorithm based on »spatial representation« techniques to solve the problem of packing rectangular carton boxes into a single rectangular container, which includes finding the best placement position and orientation for each box. By increasing the number of boxes to be fitted in the container, the matrix expands to accommodate the additional information. This packing algorithm is independent of the ordering of the boxes which usually exists in other algorithms. However, the strategy divides the original container into many small empty spaces which are not suitable for holding big boxes in successive packing. Chua, Narayanan and Loh [2] described an improved method based on Ngoi, Tay and Chua [10], which allows the user to suggest the locations of certain boxes which exists in real-life packing problems.

Ivancic, Mathur and Mohanty [6] proposed an integer programming based heuristic approach to the three dimensional packing problem in which a container is packed considering to maximize a linear function of the boxes packed.

Most of these packing strategies were based on the wall building approach with some adaptations. The method itself is a greedy heuristics which can lead to weak final solutions. Other methods include using spatial representation, reducing the problem to maximum clique problem or an integer programming problem. For methods based on the spatial representation, empty volume search routines are used to generate small spaces, whereas the successful utilization of these small spaces is difficult. For methods based on graph theory and integer programming, some disadvantages appear due to model simplification aspects and the inherent difficulty of the combinatorial nature of these problems. On the whole, published works generally give successful implementations and provided some interesting insights into the various views on how successful packing can be best achieved [9].

2 A DISCRETE EVENT MODEL APPROACH

The PLP considered in this paper deals with n different types of boxes (rectangular shaped) with known dimensions cxi, cyi, czi (with $i = 1 \ldots\ldots n$) which have to be placed onto a surface pallet of size scx * scy. The objective is assumed to be the maximisation of the number of boxes to be fitted inside the pallet (i.e. the maximisation of pallet utilisation) preserving several constraints such as the maximum weight supported in each pallet, the maximum height and floors recommended, etc.

Boxes may be rotated 90° so long as they are placed with edges parallel to the pallet's edges, i.e., the placement must be orthogonal, and they can be rotated only once. It is assumed, without loss of generality, that cxi, cyi, czi, scx, scy, gz, are positive integers, in which scx, scy defines the pallet position in which the left hand corner of the box-i has been placed, and gz corresponds the pallet floor where box-i has been placed. It is also assumed that each box has a »this side up« restriction

This palletizing problem can be seen as a DES by using a proper abstraction level in which events represent the placement of a certain box in the pallet surface, as it is represented in figure 1.

Attributes can be defined to describe the pallet configuration: the coordinates of each box placed in the pallet surface together with the coordinates of the fragmented space as the results of placing a box in the pallet.

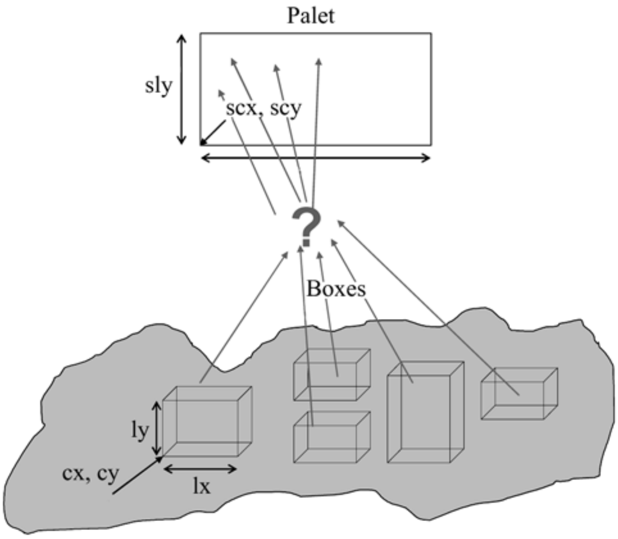

Fig. 1 Placing boxes into a pallet

As a consequence of placing a box in the pallet, fragmented surfaces will appear. Figure 2 illustrate this situation, in which it is easy to identify two different surfaces areas in the pallet that should be evaluated to fit the next box in the pallet. Thus, events describing different possibilities for placing a box in a pallet should compute the new layout configuration of the pallet once the box has been placed: position and orientation of each box in the pallet, together with the computation of the dimensions of each new free fragmented space generated as a consequence of placing boxes.

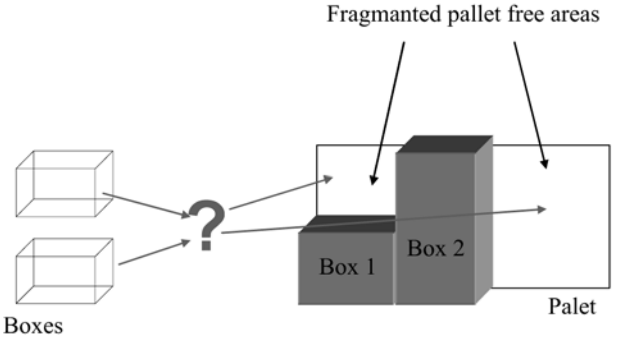

Fig. 2 Fragmented areas in a pallet

Thus, by considering the pallet state as the information that describes the boxes placed and the fragmented areas, it is easy to note that system state remains constant between events (a new box placed in the pallet) which open the opportunity to use DES modelling formalisms.

3 COLOURED PETRI NET OPTIMAZITION APPROACH

Coloured Petri Nets (CPN) [7] have proved to be successful tools for modelling complex systems due to several advantages such as the conciseness of embodying both the static structure and the dynamics, the availability of the mathematical analysis techniques, and its graphical nature [11].

The main CPN components that fulfil the modelling requirements are:

- Places: They are very useful to specify both queues and logical conditions. Graphically represented by circles.
- Transitions: They represent the events of the system. Graphically represented by rectangles.
- Input Arc Expressions and Guards: Are used to indicate which type of tokens can be used to fire a transition.
- Output Arc Expressions: Are used to indicate the system state change that appears as a result of firing a transition.
- Colour Sets: Determines the types, operations and functions that can be used by the elements of the CPN model. Token colours can be seen as entity attributes of commercial simulation software packages
- State Vector: The smallest information needed to predict the events that can appear. The state vector represents the number of tokens in each place, and the colours of each token.

The Colour sets will allow the modeller to specify the entity attributes. The output arc expressions will allow specifying which actions should be coded in the event routines associated with each event (transition). The input arc expressions will allow specifying the event pre-conditions. The state vector will allow the modeller to understand why an event can appears, and consequently to introduce new pre-conditions (or remove them) in the model, or change some variable or attribute values in the event routines to disable active events.

From the OR point of view, the CPN model can provide with the following mathematical structures:

- Variables: A variable can be identified for each colour specified in every place node.
- Domains: The domains of the variables can be easily determined by enumerating all the tokens specified in the initial state.
- Constraints: Can be obtained by straightforward from the arc and guard expressions. Arc expressions can contain constant values, colour variables or mathematical expressions.

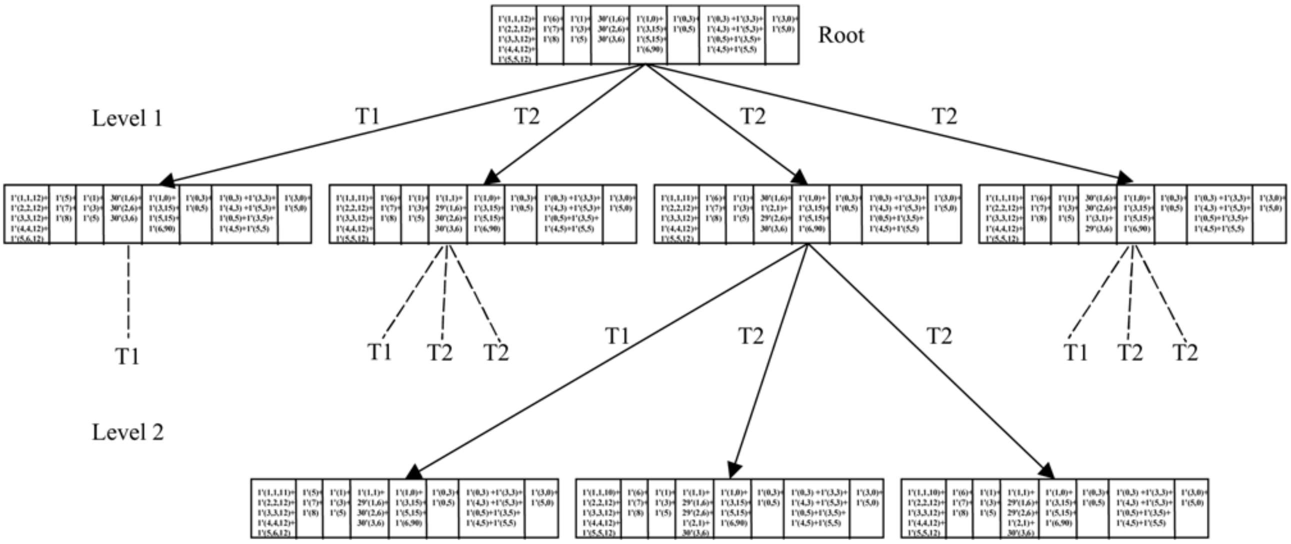

Fig. 3 First two levels of a coverability tree

From the AI point of view, the coverability tree of a CPN model allows to determine:

- All the events that could appear according to a particular system state (figure 3).
- All the events that can set off the firing of a particular event.
- All the system states (markings) that can be reached starting from a certain initial system operating conditions M_0.
- The transition sequence to be fired to drive the system from a certain initial state to a desired end-state.

4 COLOURED PETRI NET SPECIFICATION OF THE PALLET PACKING PROBLEM

The Pallet loading model has been specified in the CPN formalism and it can be decomposed into 5 different set of transitions:

- Transitions T_{21}…T_{24}: These transitions represent fitting a box into a free pallet area where the length *x* and/or *y* of the box are equal or shorter than the length *x* and *y* of the free pallet surface
- Transitions T_{31}… T_{35}: These transitions represent fitting a box into a free pallet area where the length *x* and/or *y* of the box are longer than the length *x* and *y* of the free pallet surface
- Transitions T_{41}… T_{48}: These transitions represent fitting a virtual box into a free pallet area where the length *x* and/or *y* of the box are equal or shorter than the length *x* and *y* of the free pallet surface
- Transitions T_{51}… T_{58}: These transitions represent fitting a virtual box into a free pallet area where the length *x* and/or *y* of the box are longer than the length *x* and *y* of the free pallet surface

Colour specification

Table 1 and 2 summarizes the colours and places used to describe all the information required to fit boxes in a pallet using the abstraction level of the pallet maker process introduced in this section.

Table 1 **Place specification**

Place	Colour	Meaning
P1	1'(idc, cx, cy, cz, lx, ly, lz, cr, ce)	Boxes
P2	1'(scx, scy, slx, sly)	Free Pallet Surfaces
P3	1'(idc, cx, cy, cz, lx, ly, lz, ce)	Virtual Boxes
P4	1'(ge, gz, gsf, gncv, gnc)	Global Variables

Event examples

Figure 5 illustrates the event that formalizes fitting a box into a free pallet area when the length *x* and *y* of the box are shorter than the length *x* and *y* of the free fragmented pallet surface. It should be noted that under these circumstances, the fragmented new spaces in the pallet is incremented in two new free areas that can be used as free pallet areas to place future boxes.

Node P1 holds the tokens associated to boxes while P2 holds the tokens representing the free areas in the pallet. Thus, in transition T_{22} (figure

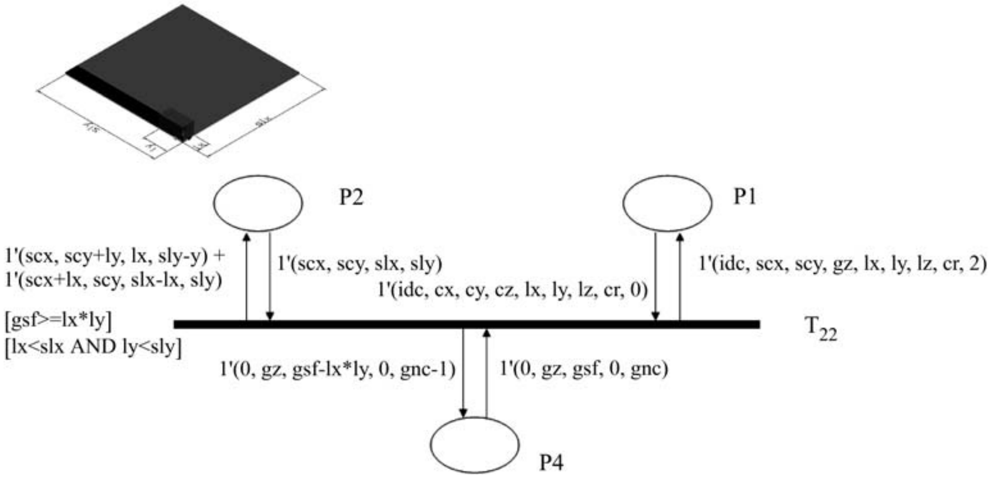

Fig. 4 T_{22}. Fitting a box with lx< slx and ly< sly

Table 2 **Colour specification**

Colour	Definition	Meaning
idc	Integer	Box identifier
Cr	Integer	0: original orientation 1: rotated 90° wrt Z
Ce	Integer	0: not assigned 1: working 2: placed in the pallet
Cx	Real	Coordinate X where the box is located
Cy	Real	Coordinate Y where the box is located
Cz	Real	Coordinate Z where the box is located
Lx	Real	Box length in coordinate X
Ly	Real	Box length in coordinate Y
Lz	Real	Box length in coordinate Z
Scx	Real	Coordinate X where the surface is located
Scy	Real	Coordinate Y where the surface is located
Slx	Real	Surface length in coordinate X
Sly	Real	Surface length in coordinate Y
ge	Integer	0: A Box can be placed in the pallet 1: Box to be assigned 2: Looking for a surfaces 3: Evaluating the new fractioned surfaces
gz	Integer	Indicates the pallet floor
gsf	Real	Available surface in the pallet
Gncv	Integer	Number of virtual boxes
Gnc	Integer	Total number of boxes

4) two tokens are generated to describe the two new free squares generated due to the space fragmentation

When a box does not fit in a free area, as shown in Figure 5, but there is a way to place it by compacting different free areas that can be attached altogether, the box is virtually decomposed as a sequence of smallest boxes (virtual box concept). A virtual box is the division of a box into small boxes

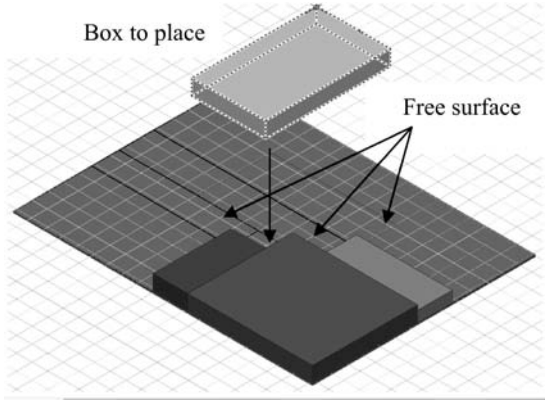

Fig. 5 T_{31}. Fitting a box with lx>slx and ly>sly

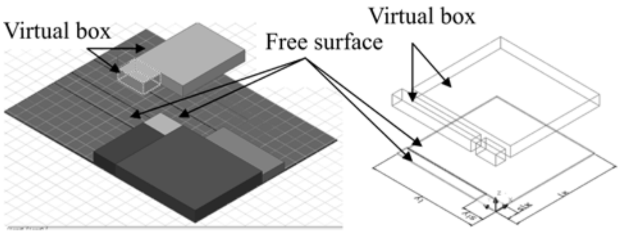

Fig. 6 T_{31}. Virtual boxes

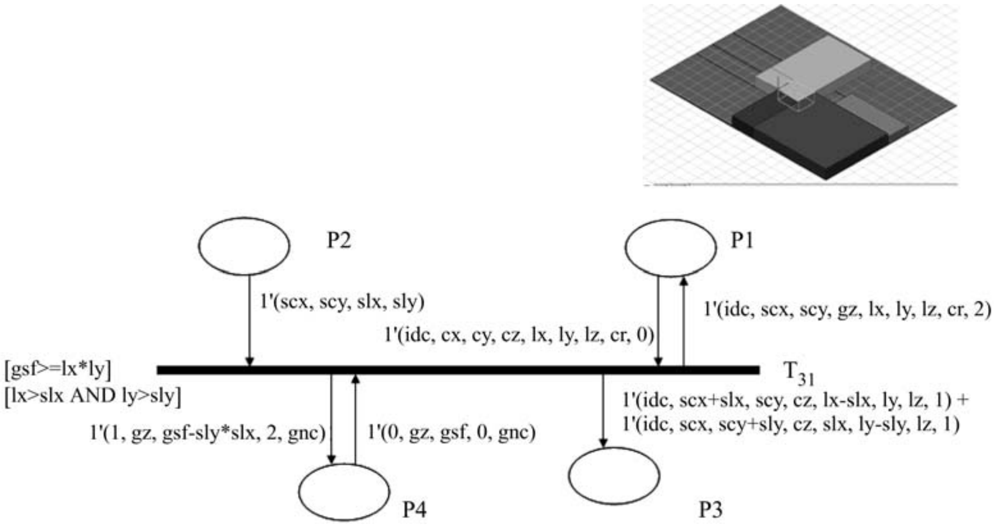

Fig. 7 T_{31}. Fitting a box with lx>slx and ly>sly

which must be placed in contiguous free areas (Figure 6).

Figure 7 illustrates the event that formalizes fitting a box into a free pallet area when the box's lengths *x* and *y* are longer than the lengths *x* and *y* of a free pallet surface. Therefore, two new virtual boxes are generated and added in the place P3 to describe the dimensions. P4 is used to control and coordinate the sequence of events to solve a virtual box placement before any other box could be chosen to be placed in the pallet. Colour *Ge* changes form *0* to *1* in the output arc to mark that the original box has been divided into one or more virtual boxes; the global free surface also is decremented in *sly***slx* which are the dimensions of the free surface used and the number of virtual boxes *gncv* is incremented in two units, which are the two virtual boxes generated due to the faring of this transition.

To place a virtual box, two different set of events have been considered: the first set represents the placement of a box that do not overcome a free surface, while the second set of events places a box which does overcome the surface and generate more virtual boxes.

Figure 5 which was used to introduce the concept of virtual box is used again to illustrate the idea of placing virtual boxes. It is assumed that part of the original box has been placed and it remains only one virtual box to be placed as shown in Figure 8.

In this situation, transition T_{58} would be fired (see Figure 9), in which a virtual box is placed generating one virtual box, which is indicated by the output arc of place P3. Also another free surface is generated, output arc of place P2. Finally as explained in the transition T_{31}, global free area is decremented and the number of virtual boxes keeps steady because one virtual box is placed but another is created.

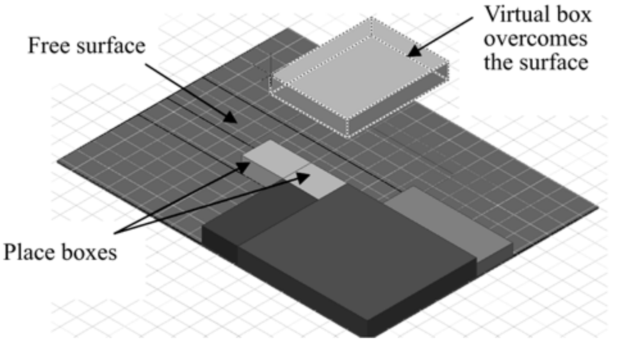

Fig. 8 Fitting a box with lx>slx, ly<sly, scx=cx and scy=cy

Additionally to all these events that specify how to place a box into a free pallet area, there is a particular event that allows changing the orientation of the box in order to fit better in a free pallet area. This new event can be fired at any time but can only be fired only once per each box. Figure 10 illustrates the arc expressions that describe this event.

5 HEURISTICS

The coverability tree relies on the computation of all reachable states and state changes of the sys-

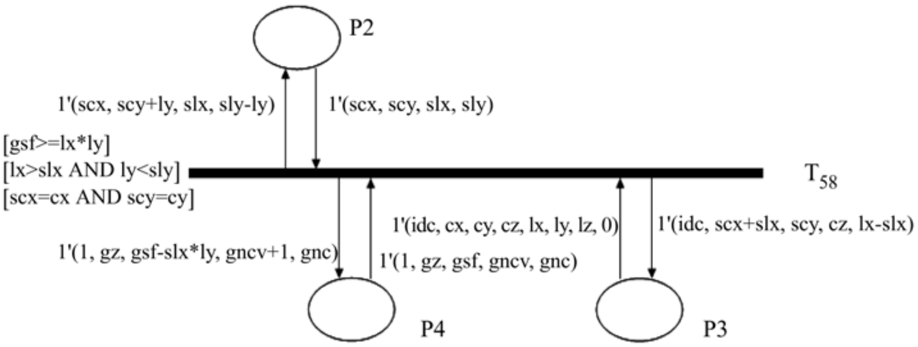

Fig. 9 T_{58}. Fitting a box with lx>slx, ly<sly, scx=cx and scy=cy

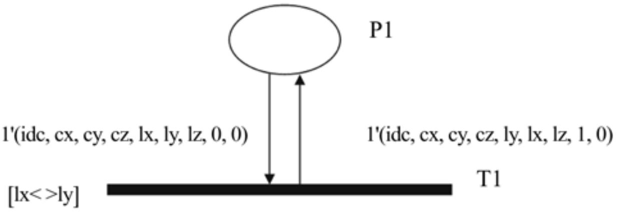

Fig. 10 Rotation of a box to be fitted in the pallet

tem, and it is based on an explicit state enumeration. Thus, by means of the state space exploration of the CPN model developed, it is possible to check the different combination in which boxes can be fitted in the pallet, and choose the one that minimize the number of levels of boxes in the pallet.

The specification of the final state consists to force all the tokens in node P1 to set the colour ce with value 2, mathematically represented by the vector:

$$Mf = \left[*,\left(*,*,*,*,*,*,*,2\right)*,*,*,\right] \qquad (1)$$

Thus, any state with all the tokens in place P1 with colour *ce*=2 can be considered the goal state since all the boxes has been fitted inside the pallet area.

Since the exploration of the whole coverability tree is quite expensive in terms of computer memory requirements and computational time, some heuristics have been designed to avoid the evaluation of certain sequence of events that will not lead a good solution. In Piera [12] the main aspects of the CPN tool used to support heuristics and knowledge representation to improve the analysis of the coverability tree is presented. This tool has been used to get feasible results solving the pallet loading problem using a reduced number of different types of boxes by means of formalizing specific knowledge in terms of heuristics.

An heuristic has been developed and implemented in the model so boxes are placed by groups of the same type. Thus, if a box of certain type is chosen, the model is forced to place all boxes of this type before any other type of box could be chosen.

This heuristic is specified by the addition of two transition nodes to the model (see Figure 11), two place nodes (see Table 3) and two colours (see Table 4). These colours are *tp*, type box which can take values of *0* if the type of box has never been picked and *1* if it has been picked; the other colour added to the model is *nc*, the number of boxes of the same type.

Place P1 is modified to consider the colours *tp* and *nc* instead colour *ce*, thus transition T_0 can only pick a type of box that has not been picked before. This transition introduces the boxes to the system allowing a box type to be used by the system. This is done by changing tp from 0 to 1 so the box can be available to the place P1.

The other transition introduced to the model is T_C. When all the boxes of the same type have been

Table 3 **Changes in place specification**

Place	Colour	Meaning
P1	1'(idc, cx, cy, cz, lx, ly, lz, cr, ,tp, nc)	Boxes
P	1'(tp)	Type Box
CO	1'(idc, cx, cy, cz, lx, ly, lz, cr)	Placed Boxes

Table 4 **Changes in colour specification**

Colour	Definition	Meaning
tp	Integer	Type Box
nc	Integer	Number of boxes of a type

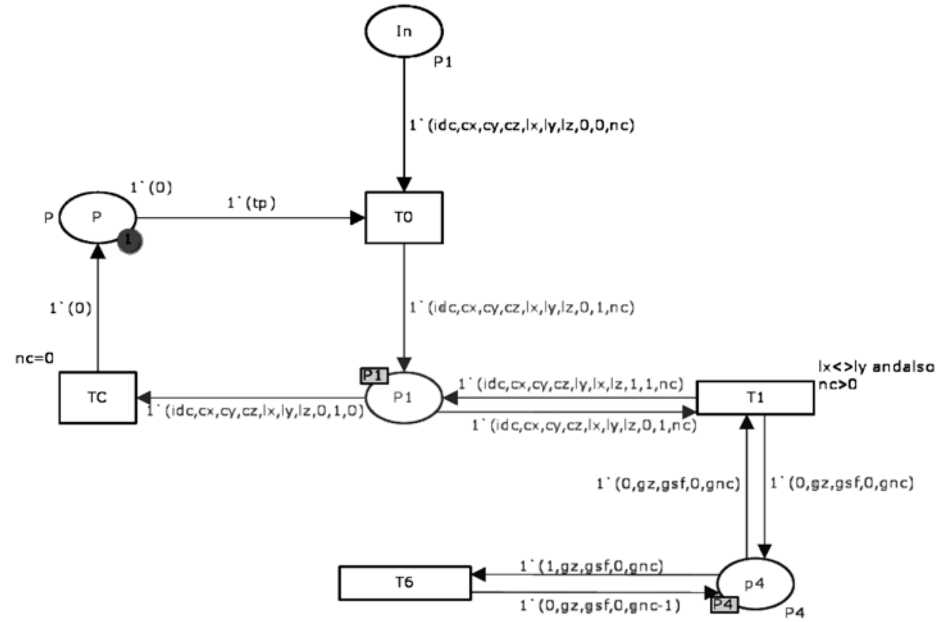

Fig. 11 Model with and heurestic

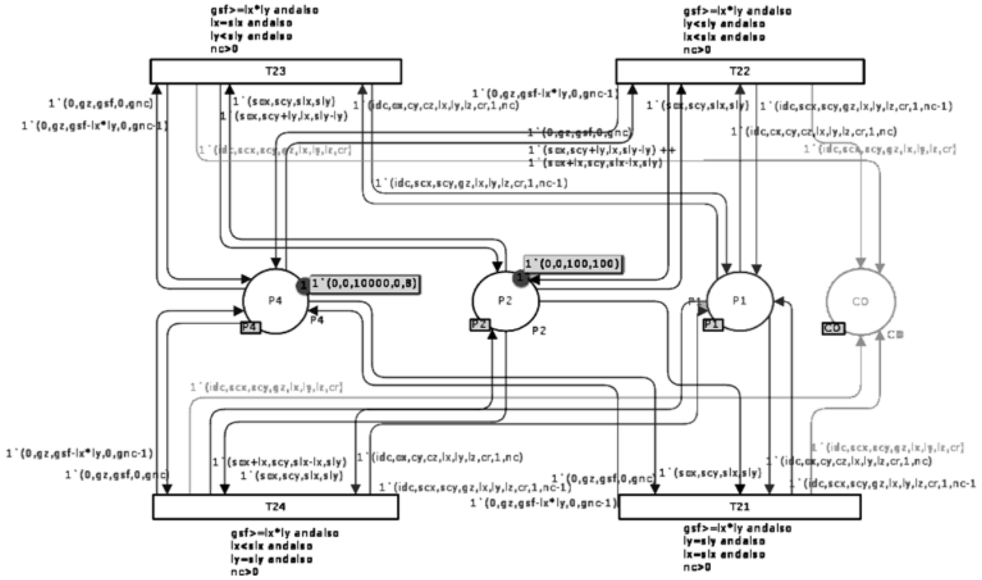

Fig. 12 Model with and heurestic

placed, $nc=0$, transition T_C is ready to be fired changing the colour *tp* from 0 to 1 and keeping this information in place P. Finally a new place CO was introduced to support placed box information (see Figure 11 and 12).

6 RESULTS

Several tests exploring the coverability tree have been performed. To illustrate the benefits of the CPN model, let's consider eight boxes of three different types. The initial markings of the box place is:

$1'(1,0,0,0,60,40,10,0,0,2) ++1'(2,0,0,0,40,35,10,0,0,2) ++ 1'(3,0,0,0,30,20,10,0,0,4)$.

These boxes must be accommodated in a surface of 100×100 units. Two feasible solutions were reached, and both fired the same number of transitions to reach the final state, the information is shown in table 5 (figure 13).

Table 5 **Optimal solutions**

Solution No. 1
1'(1,40,0,0,60,40,10,0)+1'(1,40,40,0,60,40,10,0)+ 1'(2,0,0,0,40,35,10,0)+1'(2,0,35,0,40,35,10,0)+ 1'(3,0,70,0,20,30,10,1)+1'(3,20,70,0,20,30,10,1)+ 1'(3,40,80,0,30,20,10,0)+1'(3,70,80,0,30,20,10,0)
No empty space
Solution No. 2
1'(1,0,0,0,60,40,10,0)+1'(1,0,40,0,60,40,10,0)+ 1'(2,60,0,0,40,35,10,0)+1'(2,60,35,0,40,35,10,0)+ 1'(3,0,80,0,30,20,10,0)+1'(3,30,80,0,30,20,10,0)+ 1'(3,60,70,0,20,30,10,1)+1'(3,80,70,0,20,30,10,1)
No empty space

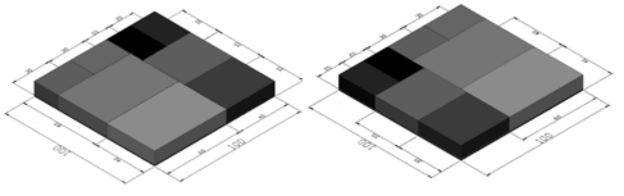

Fig. 13 Two feasible solutions

Using this information it can noticed that the 8 boxes are placed successfully inside the pallet and there is no free space unused. This means that a 100 % pallet utilisation can be obtained with all boxes properly placed.

7 CONCLUSION

The PLP has been formalized by means of a discrete event model together with a heuristic to place heterogeneous boxes in an optimal way, considering not only the space constraint but also the upstream and downstream requirements. One of the most important features of this model is the flexibility that it provides due of its modelling method. The procedure presents a good basis for incorporating different kinds of particular considerations which is a feature currently lacking in many pallet loading optimization tools

The high number of decision variables in present logistic systems, usually can lead to a huge decision tree, which make practically impossible its computational handling. Some concepts from the field of Constraint Logic Programming have been implemented in a CPN simulator to avoid the firing of infertile events that would drive the system to unfeasible states.

Despite the model has been used to solve academic pallet packing problems, the use of constraints and heuristics can help considerably to use the proposed approach to deal with feasible solutions when applied to real industrial pallet packing problems.

A very demanding system in which the developed model could contribute to improve drastically its performance is the air cargo transport system. It should be noted, that most bellies in narrow body aircrafts are usually empty because of tight time constraints in the turn-round operation. A proper solution considering both time and volume constraints would contribute to a more sustainable air transport systems, since the belly transport is mainly paid by passengers.

8 ACKNOWLEDGEMENTS

This work is partly funded by the Science and Innovation Ministry of the Spanish Government »Discrete Event Simulation Platform to improve the flexible coordination of land/air side operations in the Terminal Manoeuvring Area (TMA) at a commercial airport« CICYT Spanish program TRA2008-05266/TAIR.

9 REFERENCES

[1] E. Bischoff, W. B. Dowsland, **An application of the microcomputer to product design and distribution**, Operational Research Society Journal, 1982.

[2] C. K. Chua, V. Narayanan, J. Loh, **Constraint-based spatial representation technique for the container packing problem**, Integrated Manufacturing Systems, 1998.

[3] H. Gehring, K. Menschner, M. Meyer, **A computer-based heuristic for packing pooled shipment containers**, European Journal of Operational Research, 1990.

[4] J. A. George, B. F. Robinson, **A heuristic for packing boxes into a container**, Computer and Operational Research, 7, pp. 147–156, 1980.

[5] C. P. Han, K. Knott, J. P. Egbelu, **A heuristic approach to the three-dimensional cargo-loading problem**, Int. J. Prod. Res., 27 (5), pp. 757–774, 1989.

[6] N. Ivancic, K. Mathur, B. B. Mohanty, **An integer programming based heuristic approach to the three dimensional packing problem**, Journal of Manufacturing and Operations Management, 1989.

[7] K. Jensen, **Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, 2**, Springer-Verlag. Berlin, 1997.

[8] M. L. Kristensen, S. Christensen, K. Jensen, **The practitioner's guide to coloured Petri Nets**, Springer-Verlag, pp. 98–132, 1998.

[9] A. Lim, B. Rodrigues, Y. Yang, **3-D Container Packing Heuristics Applied Intelligence**, Springer Science + Business Media, l22, pp. 125–134, 2005.

[10] B. K. A. Ngoi, M. L. Tay, E. S. Chua, **Applying spatial representation techniques to the container packing problem**, Int. J. Prod. Res, 1994.

[11] M. A. Piera, et al., **Optimization of Logistic and Manufacturing Systems through Simulation: A Colored Petri Net-Based Methodology, SIMULATION**, Transactions of The Society for Modeling and Simulation International, 8, pp. 121–129, 2004.

[12] M. A. Piera, M. A. Mujica Guasch, **An efficient CPN modeling approach to tackle the pallet packing problem**, in Proceedings of the 6th EUROSIM Congress on Modeling and Simulation, September 9–13, Ljubljana (Slovenia), 2007.

[13] J. Xue, K. K. Lai, **Effective methods for a container packing operation**, Mathl. Comput. Modelling, 1997.

Optimiranje pakiranja paleta u distribucijskom centru pomoću obojenih Petrijevih mreža. Proces pakiranja paleta kritična je faza opskrbnog lanca u distribucijskim centrima. Većina alata za optimiranje pakiranja paleta daje vrlo dobre rezultate kada su dimenzije kutija ujednačene, što je često slučaj u industrijskim distribucijskim centrima, ali ne daju dobre rezultate u distribucijskim centrima s velikom neujednačenošću dimenzija kutija koje treba upakirati u jednu paletu. U radu su primijenjene obojene Petrijeve mreže, koje formaliziraju proces pakiranja palete i njegovu optimizaciju integracijom evaluacijskih metoda i metoda pretraživanja. Razvijeni model daje vrlo dobre rezultate uz primjenu određene heuristike kojom se pretraživanje svodi samo na najbolje scenarije umjesto pretraživanja čitavog prostora. Predloženi pristup može se koristiti za integraciju organizacije skladišta, strategije spremanja, strategije distribucije i strategije pakiranja, čime se mogu postići značajne uštede u transportnim troškovima.

Ključne riječi: obojene Petrijeve mreže, ograničenja, stablo pokrivenosti, simulacije

AUTHORS' ADDRESSES:
Miquel Angel Piera, Catya Zuñiga, Miguel Mújica
Autonomous University of Barcelona, Faculty of Telecommunication and Systems Engineering
Barcelona, Spain
e-mail:
miguelangel.piera@uab.es; catyaatziry.zuniga@uab.cat; miguelantonio.mujica@uab.es;

Received: 2008-07-05
Accepted: 2009-02-16