AN EFFICIENT CPN MODELLING APPROACH TO TACKLE THE PALLET PACKING PROBLEM

Miquel Angel Piera [1], Miguel Mújica[1], Toni Guasch [2]

[1]Autonomous University of Barcelona, Faculty of Telecommunication and Systems Engineering,
Barcelona, Spain
[2]Politechnique University of Catalonia, Enginyeria de Sistemes, Automàtica e Informàtica Ind.
Barcelona, Spain

miquelangel.piera@uab.es (Miquel Piera)

Abstract

There is a lack of commercial decision support tools that could help to deal with the best configuration of decision variables to optimize or quasi-optimize the performance of a system with a stochastic, dynamic and synchronous behavior. Simulation models have proved to be useful for examining the performance of different system configurations and/or alternative operating procedures for complex systems. It is widely acknowledged that simulation is a powerful computer-based tool that enables decision-makers in business and industry to improve operational and organizational efficiency. However, when applying simulation techniques to increase the performance of those systems, several limitations arise due to its inability to evaluate more than a fraction of the immense range of options available. In this paper a new approach to integrate evaluation (simulation) methods with search methods (optimization) based on not only simulation results, but using also information from the simulation model will be presented.

Keywords: Coloured Petri Nets, Constraints, Coverability Tree, Simulation..

Presenting Author's biography

Miquel Àngel Piera received his MSc (Control Engineering) from the University of Manchester Institute of Technology in 1990 and his PhD degree from the Autonomous University of Barcelona (Spain) in 1994. He participates in industrial research projects in the logistics and manufacturing field and at present he is Co-director of LogiSim, a Modelling and Simulation Institution sponsored and founded by the local government of Catalonia. Recently, he has published a modelling and simulation book that is being used for teaching in many Spanish universities.

ISBN 978-3-901608-32-2

1 Introduction

World-wide market competition, short cycle product time, together with random demands instead of steady demands, are some key-factors which have forced industry to change traditional rigid and/or non-automated production architectures (such as Flow Shop, Job Shop) towards Flexible Manufacturing Systems (FMS).

Despite flexibility to react to market fluctuations can easily be achieved by reprogramming production units (CNC machines), and transport resources, efficient flexibility can only be achieved by a correct coordination of all the entities (material and resources), that takes part on the production and transport process. A key factor in this highly competitive market is the ability to respond rapidly to changes in the demand while minimizing costs.

Most research and development efforts have been focussed in the process of planning the efficient, cost effective flow and storage of raw materials, in-process inventory, finished goods and related information from point of origin to point of consumption for the purpose of meeting customer requirements is considered now a day a complex problem to be solved. However, the picking of the final deliverable products into pallets is a non automated task, mainly due to NP-Hard character of the allocation of different goods into a pallet.

The complexity in transport, the changing mentality in logistics and the constant need to improve competitiveness provoke the necessity of developing new tools that could tackle the problem from a global point of view, considering operational, strategic and tactic decisions.

There are different methodologies that have been used traditionally to give response to planning problems. Modelling and Simulation techniques have proved very useful in strategic and tactic design. However, several limitations appear when trying to find a feasible solution to a NP-complete problem. A limited number of scenarios can be evaluated in an acceptable time interval.

A solution for these operational decision problems would be finding a modelling tool that permits the treatment of NP hard problems by integrating different approaches: search methods (AI, OR) with evaluation methods (Simulation).

2 Lack of Mathematically Tractable Simulation Model Formalisms

Present simulation software packages offer powerful modelling tools to describe at the desired abstraction level all the relationships between operations, processes, resources, conditions and time or state events.

Among them, flowcharts probably are the most widely used, may be because they are very user-friendly tools that can help modellers to codify and describe the nature and flow of the steps in a process by means of graphic symbols. There are several factors that justify the broad extension of flowchart tools in most commercial simulation environments:

- Promotes understanding of a process (by explaining the steps pictorially).
- Provides a tool for training (by explaining the steps).
- Helps engineers to identify opportunities for process improvement.
- Used extensively during design stages.

However, although simulation constructors have proved to offer accuracy enough to represent any system behaviour, simulation models lack a mathematical tractable structure. Most simulation toolboxes offer a black box modelling approach (i.e. flowcharting) that avoids understanding how events interact between them. So then, optimisation methodologies based on the evaluation of alternatives cannot benefit from the modeller knowledge about the system behaviour, which is essential in the particular field of logistic and manufacturing.

In the absence of tractable mathematical structures, methodologies [3] focus is on outputs from stochastic discrete-event simulation models, they don’t take any information from the model, neither from historical information about the system state evolution. In the authors’ opinion, the use of knowledge about the system behaviour is essential to sort out the most promising configuration alternatives to be evaluated.

3 Modelling Formalism Requirements

A modelling methodology that could fulfil the information and data required by heuristic, optimization and simulation methods could help to deal with an optimal response to operational decisions problems.

Modelling requirements in terms of relationship event specifications, demands a technique of knowledge representation that considers the dynamic and synchronous nature of picking and pallet configuration, and allows representing so much the structure as the different ways in which the pallet can be organized. A proper representation, analysis and evaluation of all the event-relationships that determine the layout distribution are essential to suit modelling demands on a methodology to improve system performance.

Traditional optimisation procedures have been designed to search for optimal solutions when the system is properly modelled in terms of a set of *decision variables*, a set of *constraints* and an *objective function*.

To apply heuristic search, the modelling formalism should supply also:

- To specify all the *system states* which could be reached from a certain initial system state.
- The *event sequence* to be fired to drive the system from a certain initial state to a desired end-state.

Simulation techniques requires the specification of all possible events that can provoke a state change, for each event the model should supply all the *event preconditions*, the time consumed when firing the event together with the system *state changes* that will appears as a result of firing the event.

Both three techniques require the specification of the initial state and the goal state.

4 The Coloured Petri Net Formalism

Coloured Petri Nets (CPN) 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 [1,6,7]. Furthermore, CPN are very suitable to model and visualize patterns of behaviour comprising concurrency, synchronization and resource sharing, which are key factors when trying to optimize logistic or manufacturing systems performance.

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.

From the AI point of view, the coverability tree [1,5] of a CPN model allows to determine:

- All the events that could appear according to a particular system state (figure 1).
- 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 M0.
- The transition sequence to be fired to drive the system from a certain initial state to a desired end-state.

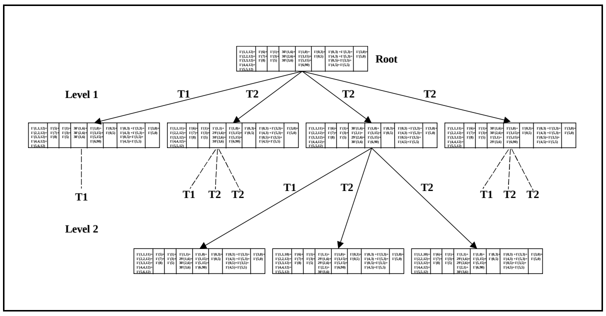

Fig. 1 First 2 levels of a coverability tree

The goal of the coverability tree is to find all the markings, which can be reached from a certain initial system state, representing in each tree node a new system state and representing in each arc a transition firing.

The main disadvantage of Coloured Petri Nets as a formalism to determine possible schedules lies in the size of the marked graphs (coverability tree) produced by modelling very complex discrete event systems, such as the picking and pallet organization system. The scheduling goal will consist of finding the sequence of operations that will allow driving the system from its original state to the final state (Mf). Since it will not always be possible to build the overall coverability tree, search control routines to determine the best options to be evaluated are required.

5 A Discrete Event Model for Pallet Maker

The palletizing problem can be seen as a DES if we use an abstraction level in which events represents the placement of a certain box in the pallet surface, as it is represented if figure-2.

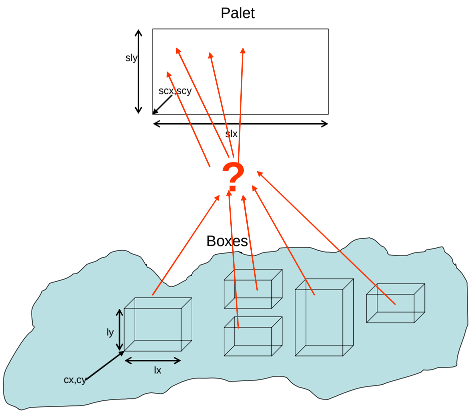

Fig. 2 Placing boxes into a pallet

Colours should 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.

As a consequence of placing a box in the pallet, fragmented surfaces will appear. Figure 3 illustrate this situation, where 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 of different dimensions.

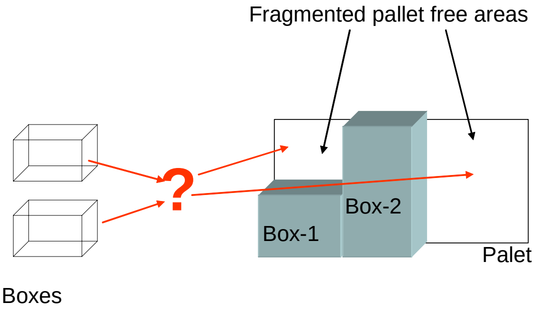

Fig. 3 Fragmented areas in a pallet

Table-1 summarizes the colours 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 Colour specification

Colour	Definition	Meaning
idc	integer	Box identifier
cr	integer	0: original orientation 1: rotated 90º 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 lenght in coordinate X
Ly	Real	Box lenght in coordinate y
Lz	Real	Box lenght in coordinate z
Scx	Real	Coordinate X where the surface is located
Scy	Real	Coordinate Y where the surface is located
Slx	Real	Surface lenght in coordinate X.
Sly	Real	Surface lenght in coordinate Y.

Ge	Integer	0: A Box can be placed in the pallet 1: Box to be assigned 2: Looking for a surface 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

Figure 4 illustrate the event that formalizes fitting a box into a free pallet area when the length of the box is the same as the length of the pallet. It should be noted that under these circumstances, the fragmented new space in the pallet is incremented in one new free area. Figure 5 describe a similar situation, but this time the length x and y of the box are shorter than the length x and y of the free area in the pallet where the box will be fitted. Under this new conditions, the fragmented new space in the pallet is incremented in two new squares than can be used as free pallet areas to place future boxes.

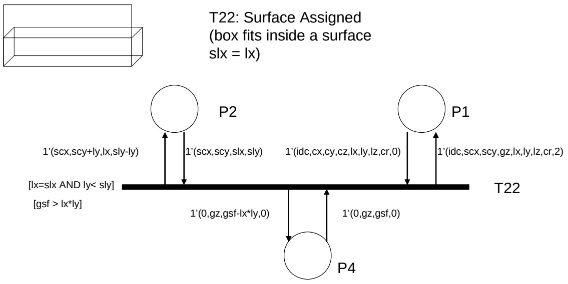

Fig. 4 Fitting a box with lx = slx

Node place P1 represents the tokens associated to boxes, and node place P2 represents the tokens describing the free areas in the pallet. Thus, in transition T22 (figure 4) only one token is generated to describe the new free area, but in transition T21 (figure 5) two new tokens are generated to describe the two new squares generated due to space fragmentation.

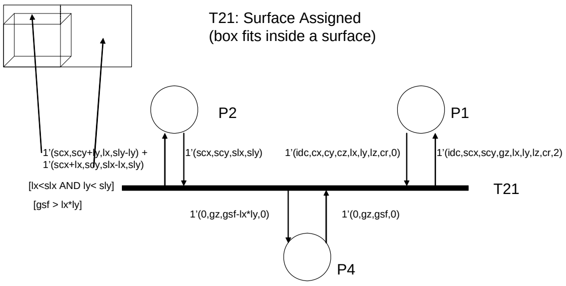

Fig. 5 Fitting a box with lx < slx and ly < sly

Additionally to all the events that specify how to place a box into a free pallet area, there is particular event that allows to change 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 that can only be fired only once per each box. Figure 6 illustrate the arc expressions that describe this event

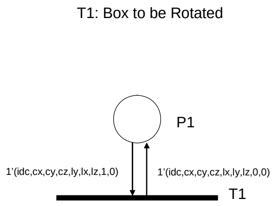

Fig. 6 Rotation of a box to be fitted in the pallet

As it can be easily noted, the new box orientation can be specified just by crossing the values of length x and y respectively.

In a similar way, the floor level in which the simulation is fitting the boxes can be easily updated just by increasing the colour **cz** in one unit each time the free area in the pallet is smaller than the area of the smaller box to be placed in the pallet. The arc expression of the output arc describing this transition initializes the free area in the pallet to pallet surface each time the trasition is fired.

By using a CPN simulator that can support the evaluation of the coverability tree of the system described under different work loads (different boxes specified in place P1) 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=[*'(*,*,*,*,*,*,*,2),*,*,*]

Which is interpreted that 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.

Because to explore the whole coverability tree is quite expensive in terms of computer memory requirements and computational time, some heuristics has been designed to avoid the evaluation of certain sequence of events that will not lead a good solution. In [2] 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 packing problem using a reduced number of different type of boxes by means of formalizing specific knowledge in terms of heuristics.

One of this heuristics that provides quite good results consist to choose those tokens from node place P1 with higher value lx*ly. In some sense, this rule try to place first those boxes that requires bigger surface areas that smaller boxes, which probably could be fitted later in the fragmented pallet surface space.

A second heuristic that provided quite good results as far as the computational time required to reach a feasible configuration is to fit the boxes that minimizes the fragmented areas in the pallet. This heuristic is easily implemented by giving more priority to transition T22 instead of T21.

6 Conclusions

The high number of decision variables in present logistic systems, usually can lead to a huge coverability 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 academicals 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.

7 Acknowledgments

The Department of Universities, Research and Information Society of the Catalonia Autonomous Government have supported this work.

It also has been possible thanks to the Dirección General de Investigación, Subdirección General de Proyectos de Investigación through the project DPI2004-08056-C03-01.

8 References

[1] K. Jensen. 1997. “Coloured Petri Nets: Basics Concepts, Analysis Methods and Practical Use”, vol. 1,2,3, Springer-Verlag.

[2] M. Narciso, Miquel Àngel Piera, T. Guasch. 2003. “A Knowledge representation approach suitable to support heuristic search methods”. Frontiers in Artificial Intelligence and Applications (IOS Press 100).

[3] M.A. Piera, M. Narciso, T. Guasch and D. Riera. 2004. “Optimization of Logistic and Manufacturing Systems through Simulation: A Colored Petri Net-Based Methodology”. SIMULATION: Transactions of The Society for Modeling and Simulation International. V80 (3). pp 121 – 130.

[4] M.A. Piera, T. Guasch, L. Vilalta, M. Mujica. 2006. “Several Hints for a Faster Search using Coloured Petri Net Simulators”. SCSC’06 Proceedings.

[5] J.M. Proth and X. Xie. 1996. “Petri Nets. A Tool for Design and Management of Manufacturing Systems”. Wiley.

[6] M. Silva and R. Valette, “Petri Nets and Flexible Manufacturing”, Lecture Notes in Computer Science, vol. 424, Advances in Petri Nets 1989, pp.374-417.

[7] A. Zimmermann, K. Dalkowski and G. Hommel, “A Case Study In Modelling And Performance Evaluation Of Manufacturing Systems Using Colored Petri Nets”, in Proceedings of the 8th European Simulation Symposium (ESS '96), pp. 282-286.