On Analysis of Puzzle-Based Warehouse Systems using Modular Petri Nets

1st Kasuni Vimasha Weerasinghe
Norwegian University of Science and Technology
Trondheim, Norway
kasuni.weerasinghe@ntnu.no

2nd Andrei Lobov
Norwegian University of Science and Technology
Trondheim, Norway
andrei.lobov@ntnu.no

3rd Fabio Sgarbossa
Norwegian University of Science and Technology
Trondheim, Norway
fabio.sgarbossa@ntnu.no

4th Lars Tingelstad
Norwegian University of Science and Technology
Trondheim, Norway
lars.tingelstad@ntnu.no

Abstract—**Puzzle-based storage (PBS) systems are intended to increase storage density using optimally available storage space. Having dynamic puzzle-based storage systems requires finding and validating tools to support the optimal operation of such systems. The problem is that as the parameters of PBS change, the optimal way for operating that new setup may need to be recalculated. The changing parameters can be related to the change of the physical layout of the PBS (e.g., dimensions, number of carriers) or mode (e.g., the number of nodes that can be simultaneously moved, amount of orders simultaneously processed). Each parameter may require reevaluation for optimal approaches to control changed PBS. This article proposes using a modular Petri Nets-based formalism to analyze PBS setup for finding optimal control approaches. We architect an adaptive PBS system that dynamically adjusts to changing configuration and/or control parameters.**

I. Introduction

Due to the rapid growth of the e-commerce industry, warehouses confront new challenges. Companies must manage enormous and varying daily order volumes while storing millions of unique items. Distribution centers have greater requirement of implementing robotized material handling to fulfill the demand.A flexible, automated, and optimized system with increased throughput and higher productivity is needed to handle the rising inventory turnover rates, shorter storage periods, and many orders in small quantities with faster deliveries. Warehouse Automation has advanced rapidly during the last decade with the acceleration of autonomous vehicle–based storage and retrieval and shuttle-based storage and retrieval systems.

Robotized storage systems are especially suited for e-commerce operations as they can function in a small space and are flexible enough to meet changing demand. When there is a strong demand for warehouse space, building up a warehouse design using the area designated for traditional aisles is not practical. In these situations, "puzzle-based" solutions provide a possible warehousing option. Moving desired things to an input/output site at escort places, unit loads in puzzle-based systems (PBS) can be continually moved in one of four directions.

PBS was not considered able to fulfill the demanding delivery deadlines in an e-commerce context. It is contended that system's mobility is not adequate to meet the expectations of a retrieval systems due to low throughput. Therefore, PBS systems with movable storage racks that can move autonomously have been investigated in this study. This will lessen the restrictions placed on movement in the PBS system. Loads might theoretically move in any direction with autonomous wheels. According to the latest studies on PBS, diagonal cross aisles in conventional aisle-based warehouses can shorten travel distances. To reduce travel distance, we tried to build a configuration that would allow loads to move diagonally within a PBS system. Further, studies on operational planning in PBS systems suggest that retrieval time can be shortened by loosening restrictions imposed by the initial configuration. It is possible to establish virtual aisles with more escorts, allowing specified loads to flow without interruption.

The main aim of this paper is to model puzzle-based systems using Petri Nets. A graphical and mathematical modeling tool called a Petri Net describes and evaluates systems with concurrent processes and interactions. They can be used to describe and examine a system's numerous components. Modeling PBS systems using petri nets can be highly beneficial for several reasons. The structure and behavior of the puzzle-based system can be represented visually using Petri Nets. When it comes to complicated, automated systems, in particular, this visual paradigm makes it simpler to comprehend and convey how the system works.

The paper is organized as follows. The next section goes into a literature review on puzzle-based systems and the modeling of those with Petri Nets (PN). The section ends with the highlighted contributions of the paper. The third section presents the approach we used to model PBS. The modeling and analysis based on the presented approach comes in section four. Discussion and lessons learned come in section five. Finally, there are conclusions and future research and development work.

II. Literature Review

A. *Puzzle-based storage systems*

Today, the need for more storage space or a compact storage policy is driven by fast deliveries and short storage periods of goods. Since it is not economical to expand warehouse space with the space scarcity challenge, using compact storage policy would be an ideal solution. Under that, a puzzle-based system is a more practical way to enhance the storage density.

The PBS helps to use storage space more efficiently by, for example, removing the aisles [1]. PBS systems was initially introduced by [1]. The basic structure is similar to the 15 puzzle game as in Fig. 1. This increases overall storage density which is important for being able to handle more products in such a system. The article [1] analyses single-escort and multi-escort setups of PBS, where escort refers to an open location. Expected retrieval times for such situations are calculated depending on the capacity of the PBS.

In [2], different storage strategies were analysed and compared with respect to retrieval times and the number of escorts. The examination of a novel management strategy for puzzle-based storage systems based on AGVs and AMRs rather than self-propelled shelves is suggested in previous studies [4]. Density and retrieval time have been examined as key design variables in this study for PBS systems.

Even though various objectives and configurations have been investigated in previous studies, the common goal is to improve the retrieval performance while maintaining high density. In [5], methods to determine the optimal retrievals are developed for the cases where multiple escorts are randomly placed within a high-density PBS system.

Improvements to the state of the art are made in [6], focusing on optimal algorithms to retrieve goods using the minimum number of moves when escorts are arbitrarily positioned in the system. As a result of energy efficiency, warehouse operations become more sustainable and simultaneously lower operational expenses.

Among many other complex scheduling issues that have been studied, minimizing of the total number of moves when retrieving a single item with several escorts is a crucial component of the PBS system's operation. Therefore, [8] further addressed that issue proposing heuristics with different computational complexity setting which can improve the accuracy while maintaining stronger robustness.

According to PBS research, complexity rises as the number of escorts rises. Furthermore, simultaneous movements and block movements add more complexity since the possibility of conflicting moves can be introduced when loosening the movement constraints. However, retrieval of multiple loads simultaneously will increase the throughput capacity. [7] has already proposed algorithms that enable multiple load retrieval considering how and in which sequence loads should be retrieved to reduce overall travel time.

In the work done by [9], a multi-item retrieval problem in the PBS system with general settings, where numerous sought items, escorts, and I/O locations are randomly distributed has been solved using deep reinforcement learning algorithm. The developed framework can be used to deal with simultaneous movements and large-scale instances which expands the applicability of PBS systems.

Increasing the knowledge related to PBS system, a new paradigm of work was carried out by [10] where loads are moved by a limited number of autonomous mobile robots (AMRs). Being able to shift all AMRs simultaneously with or without loads makes these PBS-AMR systems challenging to study. The suggested model through their work is capable of handling both single-load and block motions, several I/O points, and different restrictions on concurrent vehicle movements.

Overall, looking at the research described so far, one can see a constant pursuit of better control for complexity in PBS. That starts from simpler setups with single versus multiple escorts up to synchronized moves for the blocks of escorts in PBS. In order to further develop better control solutions for PBS, different modeling and analysis approaches have to be considered. Thus, the next subsection is dedicated to a family of formalisms used to represent and analyse concurrent systems – Petri Nets, which look like a promising paradigm to study PBS.

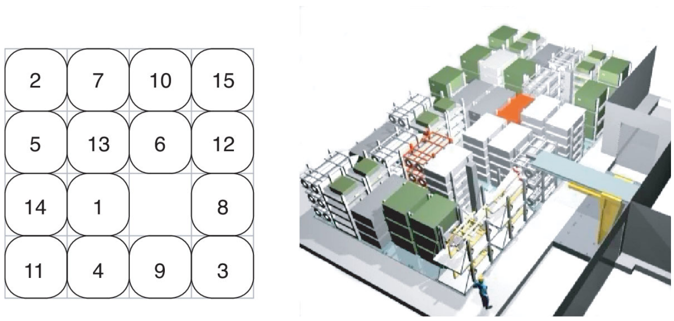

Fig. 1. PBS system example [1]

B. *Modelling of concurrent systems with Petri Nets*

The puzzle-based storage systems present concurrent features with interactivity-based processes. In order to fully understand the capabilities and complexity of these kinds of multi-agent systems, a formal analysis of system properties is required. In addition to simulation, analytical procedures are required for the development of modeling and verifying formal system attributes. Therefore, Petri Nets are a suitable modeling tool due to their extensive range of application areas.

Petri Net modeling has several benefits. Petri Nets offer a graphical representation that makes it simple to visualize a complex system. State space analysis for the purpose of structural property discovery is made possible by Petri Nets. It is possible to model corresponding systems at different levels of abstraction or in depth. Timed Petri nets can be used to analyze the system's performance. Moreover, it can be used to develop, model, and analyze complex software systems with minimal time and effort. The book by [11] provides a comprehensive overview of practical applications of Petri Nets which presents the fundamentals of discrete, continuous, and hybrid net formalism in an integrated manner.

Due to the enormous state space, model checking of Petri nets is difficult. Also, Petri Nets are difficult to be used for real-time applications due to the slowness in simulation. Thus, as mentioned by [12] Petri Nets can be performed in a modular way to overcome these issues. In modular Petri Nets, modules are designed, developed, and operated independently of each other.

In [13], an overview of the historical evolution of Petri Nets (PNs) from the standpoint of systems theory and automatic control is provided. It demonstrates how Petri nets have been applied to tackle a variety of traditional issues with control systems, such as analysis, control, diagnostics, state estimation, and observability.

Intelligent production control systems based on agents can quickly react to and adjust to environmental changes. Learning techniques that boost agent intelligence can be used to solve manufacturing system adaptation and evolution. The study by [14] presented a factory scheduling system based on Reinforcement Learning (RL) and Colored Timed Petri Nets (CTPNs). CTPNs implement the scheduling and model the manufacturing system. A scheduling agent employs RL, specifically the Q-learning algorithm, to find the best solution. They have demonstrated the technique by providing a case study of order-picking scheduling in a warehouse.

The previous literature shows applicability in PN when analysing the intricate interactions arising in various logistics system components. This kind of investigation was performed by [15] using PNs as a discrete event modeling and simulation framework to serve the functional requirements of warehouse systems, highlighting both their advantages and disadvantages. They have claimed that PNs have been shown to be adequate to describe the main features of logistic processes in which both labor and equipment are engaged.

Petri Nets have been introduced as a technique that can be used in warehouse research. Even though it is not a widely used methodology in the warehouse context, [18] claimed that using PNs to model and analyse the system and improve control strategies will offer fresh, innovative investigation into stochastic approaches used in warehouse research.

The usage of Colored Timed Petri Nets (CTPN) as a tool for automated warehouse modeling has proven to be successful. To concisely and effectively represent the dynamics of an AS/RS (automated storage and retrieval system) supplied by rail-guided vehicles, a CTPN model was presented in [16]. Based on the analysis of the behaviors of the active resources in the system, a modular and computerized model is presented using a colored timed Petri Net approach. To simplify the model and characterize the control flow of the resources, places are multicolored and token colors are defined as the routes of storage/retrieval operations.

Another hierarchical framework for AS/RS operation modeling presented in [17] is based on Petri Nets. System simulations are frequently used to choose the optimum control rules for an AS/RS operation. By extending the PN model to the colored-timed PN model, the analysis and evaluation of the performance of an AS/RS will be accomplished successfully. The system status can occasionally change for an AS/RS. As a result, the PN structure needs to have temporal capabilities to simulate the dynamic features of AS/RS's.

When considering compact storage systems such as puzzle-based-storage systems, [19] illustrated a simple PBS system with Petri Net diagram. Also, [20] has used a Petri Net model to investigate the structural properties of a PBS system. He applied the model to validate deadlock conditions in a semi-automated grid pick system.

In the literature, Petri Nets have been used to simulate both decentralized control models and material handling systems, although decentralized control models for material handling systems and puzzle based storage systems have not been well addressed.

C. Contributions of the Study

Based on the literature, it is seen that an analysis of PBS systems using Petri Nets is at a beginner level. Through this study, we explain the significance of replicating the PBS system through a PN model.

We explain how Petri Nets are useful for describing the concurrency of a system because these help to depict how various system components interact and synchronize based on a given system setup.

The state space of the puzzle-based system can be examined with the use of Petri Nets. To check certain features, including reachability, deadlock-freeness, and liveness, which are essential for assuring the system's proper operation. Through this study, we have experimented with different combinations of states and transitions to derive a proof-of-concept for PN application on PBS. PN-based modeling enables formal verification methods. Further, we have mathematically proven some specific system properties and how the model gets complicated with the added characteristics.

III. Approach

In this work, we use Timed Net Condition/Event systems (TNCES) for representing and analyzing the PBS. TNCES is a modular formalism that allows the application of separation of concerns when modeling the PBS. A formalism for the composition of input-output PNs to novel systems is indicated by TNCES. This was first presented by Rausch and Hanisch in [21]. This turned out to be an effective method of representing discrete control systems with discrete states. While modeling modular systems, Conditions/Event systems help to study their generic properties [24].

The classical examples of PBS with Petri Nets can be modeled with Timed Net Condition/Event systems (TNCES) and analyzed using the MOVIDA tools framework[1]. TNCES is a modular typed formalism that was inspired by Petri Nets. TNCES consists of transitional places, transitions, and flow arcs. The number of locations is limited and not infinite. The number of transitions is also limited and not zero. An arc connects a place to a transition or a transition to a place and it is directed. It is a modular typed formalism, where a module can be created and reused. An example of the TNCES module can be seen in Fig. 1. According to the definitions provided by [22], TNCES is a place-transition net that is formally represented by a tuple;

$$PN = (P, T, F, M, EI, EO, CI, CO, EA, CA). \quad (1)$$

The places, p1 and p2, which are elements of P, represent the status of the modeled phenomenon (e.g. a switch). The labels next to the places may reflect semantics within the context of a problem domain - a switch can be either ON or OFF. Transitions t1 and t2 – the elements of T, represent the possibility of change of the PN marking or the state. Firing transition t2 leads to the switch changing its state from ON to OFF. The firing of t1 represents a switch going into the ON state. As can be seen in Fig. 2, the places and transitions are joined by flow arcs, F. The flow arcs can have time intervals assigned of the form [a, b], where a - is a lower bound for enabling corresponding post-transition and b is the upper bound for firing the transition. If the time of firing is not reached (a) or expired (b), the transition becomes disabled.

The places have initial marking $M=(0,1)$, meaning that the first place has no tokens or marks and the second place, p2, has a token. A TCNES module can have event and condition inputs outputs defined, correspondingly *EI, EO, CI, and CO*. These allow the passing of information about firing a transition – event arcs *EA* going from a transition via event output(s) of one module to the event input(s) of another and then to the targeted transition. The information about the marking of a place can be passed between modules with the condition arcs, *CA*, interconnecting a place via module condition output(s) to condition input(s) of another module and then to its transitions. Thus, events propagate from transition to transition, and conditions propagate from place to transition.

When considering an application of PN-based modelling to the PBS system, 1 location is depicted by two places mark which indicates the busy or free status of the location. The evolution of the marking is a representation of the dynamics of a system as described by a PN. A transition represents an operation of a loading between two locations. It also represents the loading direction. As loading of one location means unloading another location, the same (one) transition will represent loading of one location and unloading of another. As the movement of a load is possible in two directions,twice amount of transitions can be applied. Each transition should have two pre-places and two post-places.

[1]MOVIDA Tools Framework: http://www.lobov.biz/academia/index.html#tools

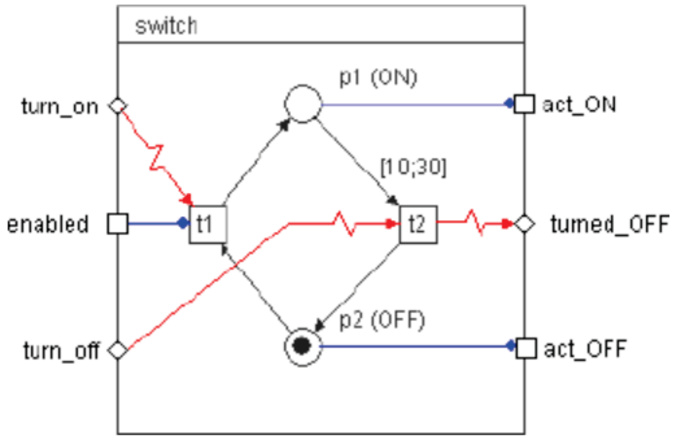

Fig. 2. TNCES module example [23]

Using the TNCES, one can model a single location in the PBS. The storage location has to represent its status (<empty, occupied>, or <free, busy>) and, if busy, being able to move the stored item(s) either to left, right, up, down (or diagonal) to neighboring locations.

IV. Modelling and Analysis of PBS with PN

A. 2-by-2 PBS example

For the illustration of the approach, as the initial step, we created a PNS model for a 2-by-2 PBS system. The total number of locations in this system is 4 with 1 escort, as illustrated in Fig. 3. The first grid location is considered as the I/O location. As a result of escort movement, a desired load can reach the I/O location.

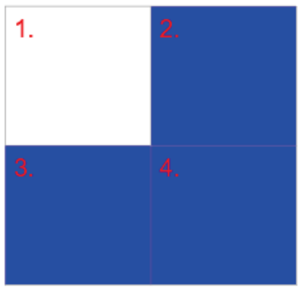

Fig. 3. 2x2 PBS system with 1 escort

The developed TNCES module is illustrated in Fig. 4. In a 2-by-2 PBS system, each load in every location can move in 2 directions. Therefore, the number of transitions should be 8. The marking (the token inside places) specifies the status of the PN or, to be more precise, the state of the system described by PN. Thus, while the marking evolves, the state also does, and the marking's evolution results from transitions being fired. A transition can only be fired if a token is presented in each of its pre-places. Then, it is said that the transition is fireable or enabled. It is most likely a design issue with the system described when some transitions are no longer fireable and when certain parts of the PN no longer "function.". This can be recognized as a deadlock or sink state. Therefore, for an initial marking, PN is considered to be deadlock-free.

In this study, reachability graphs were created to analyze the behavior and properties of PNs. It is also known as the state transition graph as it can graphically represent all the reachable states (State space). When there is only one free place (escort), it can generate 4 states. (state1: locations 1 is free, state2: location 2 is free, state3: locations 3 is free, state4: Location 4 is free).The generated state space in Fig. 5 show 4 states for 2x2 PBS. There are 8 paths in total.

B. 3-by-3 PBS example

When considering the 3-by-3 PBS system, there are 9 locations as in Fig. 6. The top left corner and top right corner loads can move in two directions while the top middle load can move in three directions. The very middle load can move in four directions. Likewise, the total number of transitions can be identified as 24. The TNCES model with all the locations and transitions is depicted in Fig. 7.

The generated state space in Fig. 8 shows 9 states for 3-by-3 PBS. Through the diagram, we can identify what are the states pertaining to corner places which can move only for 2 directions. State 3, 4, 8 and 9 are the states where corner locations are free. State 2 represents the state in which the middle location is free as it can be loaded from any of the 4 directions. The system's potential states are depicted on the graph, along with the transitions that can be fired to change between them.

The number of states for each scenario can be mathematically derived from the combinatorial formula:

$$_nC_r = \left(\begin{array}{c} n \\ r \end{array} \right) = \frac{n!}{r!(n-r)!} \tag{2}$$

In (2), n is a number of locations and r is a number of escorts. Thus, we have got 4 states for the 2-by-2 instance ($n = 4$, $r = 1$) and 9 states for the 3-by-3 instance ($n = 9$, $r = 1$).

V. Discussion and lessons learned

PNs assist in formal verification to validate system characteristics and ensure accuracy. It can help to spot any mistakes or inconsistencies in the system design by being used to check features including reachability, boundedness, and liveness. However, real-world physical system modeling and analysis is a challenging topic, particularly for compact complex systems like PBS. It is not required to manually draw and understand all the interactions and components of a PBS system since it can be replicated through a PN model. However, still there are limitations of the PN modelling. For simple models (2-by-2 PBS, as in Figure 3, and 3x3 PBS with one free location, as shown in Figure 6) which we illustrated in this paper, PN models can be developed to provide a better understanding. Increasing the number of free places (escorts) for the same PBS system will generate significantly complex reachability graphs. For a 3x3 PBS with more than 1 free location, 84 states were generated as can be seen in Fig. 9. For the given case there are 3 escorts or free locations. Thus, by equation (2) the result match as

$$_9C_3 = \left(\begin{array}{c} 9 \\ 3 \end{array} \right) = \frac{9!}{3!(9-3)!} = 84. \tag{3}$$

As the size and complexity of the system increase, the PN model can also become complicated and challenging to manage. Petri Nets' graphical representation can get crowded and difficult to understand, especially in complex systems with many of components and interactions. Therefore, there is a need for more tools that would help analysis, and navigation of state spaces, and also to check whether there are any deadlocks. When considering retrieval of goods using PBS system, time is a significant performance matrix. Instead of explicitly describing time, traditional Petri Nets concentrate on modeling the occurrence of events and transitions. This can be another drawback if the modeled system depends on accurate time and temporal correlations between events. Despite these drawbacks, Petri Nets remain valuable for modeling PBs systems. However, there is a need for advanced modeling tools that integrate with simulation and automation. Inventory management software in a warehouse is frequently connected with puzzle-based storage systems. As a result, it is possible to track and monitor inventory levels in real time and to integrate with other warehouse management systems seamlessly. Automatic recording and updating of each storage movement ensures accurate and current inventory location data. The PN model can be further developed to accurately represent the system's real-time behavior.

VI. Conclusions and future work

In comparison to conventional shelving or racking systems, the PBS system maximizes the use of the warehouse's available storage space and permits higher storage density. Since PBS is designed to retrieve items efficiently in compact configurations, Petri Nets model can be used to identify the system behaviour more accurately. Concurrent processes and interactions can be visually represented and analyzed by PN model. It can demonstrate the different states that the system might be in, how it changes states depending on the occurrence of events or the availability of resources, and overall activities in the system. Therefore this study will highlight the importance of modeling and analysing the behavior of concurrent systems and making them applicable in different domains such as logistics and manufacturing. Further, through this study, principles behind locations and modeling of locations were identified and validated. This can be easily extended by implementing tools that could automatically generate those models.

In future work, an automated generation of the PNs of arbitrary PBS size and movement possibilities between the locations will be implemented. Thus allowing fast analysis of PBS systems. For both models we explained in this study,

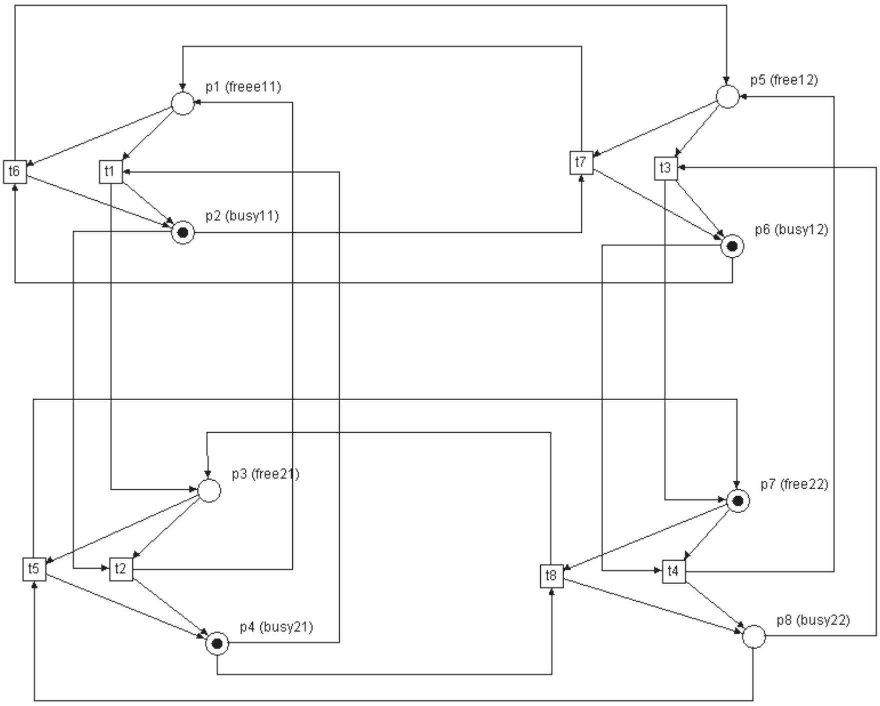

Fig. 4. TNCES module for 2x2 PBS system

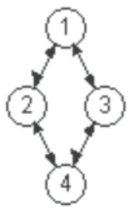

Fig. 5. State space for 2x2 PBS system with 1 escort

Fig. 6. 3x3 PBS system

we considered rectilinear movements. However, it can be further developed for the PN models which represent diagonal movements of the loads. Considering diagonal movement in PBS system will enhance the system's performance as it enables the load to reach to input/output point more easily. Therefore, Petri Nets can be integrated to model PBS with diagonal movement and evolve it as a promising research area.

References

[1] Gue, K.R. and Kim, B.S. (2007), Puzzle-based storage systems. Naval Research Logistics, 54: 556-567. https://doi.org/10.1002/nav.20230

[2] Taylor, G Don; Gue, Kevin R., The Effects of Empty Storage Locations in Puzzle-Based Storage Systems, IIE Annual Conference. Proceedings; Norcross (2008): 519-523.

[3] K. R. Gue, K. Furmans, Z. Seibold and O. Uludağ, "GridStore: A Puzzle-Based Storage System With Decentralized Control," in IEEE Transactions on Automation Science and Engineering, vol. 11, no. 2, pp. 429-438, April 2014, doi: 10.1109/TASE.2013.2278252.

[4] Alfieri, A., Cantamessa, M., Monchiero, A. et al. Heuristics for puzzle-based storage systems driven by a limited set of automated guided vehicles. J Intell Manuf 23, 1695–1705 (2012). https://doi.org/10.1007/s10845-010-0471-7.

[5] Kota, V.R., Taylor, D. and Gue, K.R. (2015), "Retrieval time performance in puzzle-based storage systems", Journal of Manufacturing Technology Management, Vol. 26 No. 4, pp. 582-602. https://doi.org/10.1108/JMTM-08-2013-0109

[6] Altan Yalcin, Achim Koberstein and Kai-Oliver Schocke (2019) An optimal and a heuristic algorithm for the single-item retrieval problem in puzzle-based storage systems with multiple escorts, International Journal of Production Research, 57:1, 143-165, DOI: 10.1080/00207543.2018.1461952

[7] Masoud Mirzaei, René B.M. De Koster and Nima Zaerpour (2017) Modelling load retrievals in puzzle-based storage systems, International Journal of Production Research, 55:21, 6423-6435, DOI:

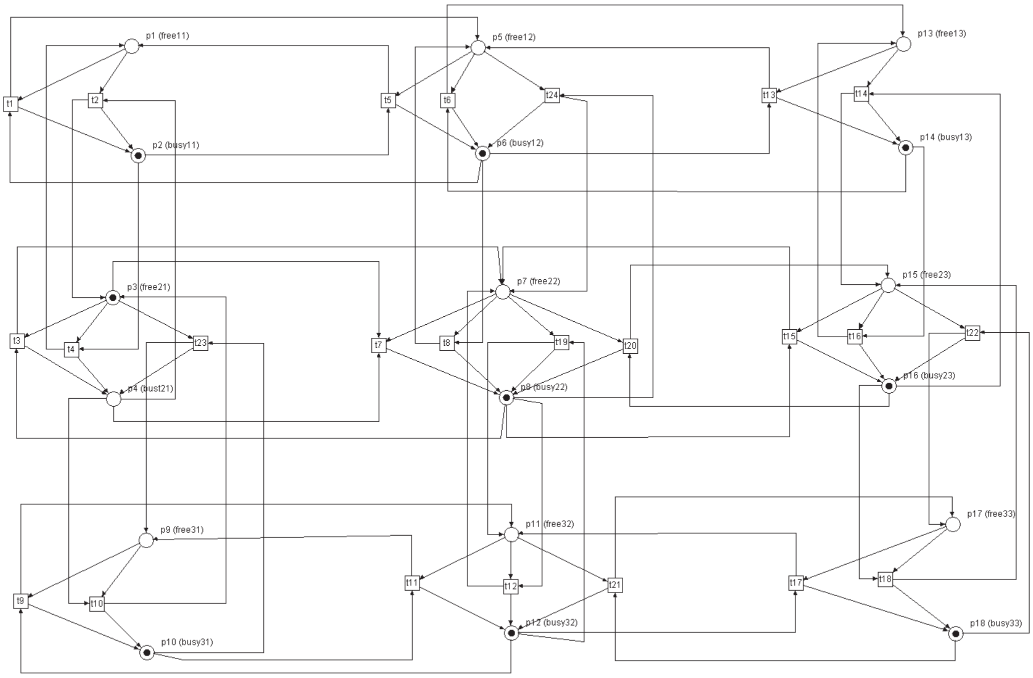

Fig. 7. TNCES model for 3x3 PBS with one free location.

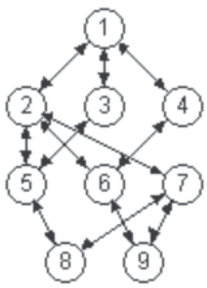

Fig. 8. State space for 3x3 PBS system with 1 escort

10.1080/00207543.2017.1304660

[8] Yunfeng MA, Haoxun CHEN, Yugang YU, An efficient heuristic for minimizing the number of moves for the retrieval of a single item in a puzzle-based storage system with multiple escorts, European Journal of Operational Research, Volume 301, Issue 1, 2022, Pages 51-66, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2021.09.032.

[9] Jing He, Xinglu Liu, Qiyao Duan, Wai Kin (Victor) Chan, Mingyao Qi, Reinforcement learning for multi-item retrieval in the puzzle-based storage system, European Journal of Operational Research, Volume 305, Issue 2, 2023, Pages 820-837, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2022.03.042.

[10] Tal Raviv, Yossi Bukchin, René de Koster (2023) Optimal Retrieval in Puzzle-Based Storage Systems Using Automated Mobile Robots. Transportation Science. https://doi.org/10.1287/trsc.2022.1169

[11] René David, Hassane Alla (2010)Discrete, Continuous, and Hybrid Petri Nets. Springer Berlin, Heidelberg. ISBN 978-3-642-10669-9, https://doi.org/10.1007/978-3-642-10669-9

[12] Davidrajuh, R. (2021). Design of a New Modular Petri Nets. In: Petri Nets for Modeling of Large Discrete Systems. Asset Analytics. Springer, Singapore. https://doi.org/10.1007/978-981-16-5203-5-7

[13] Alessandro Giua, Manuel Silva (2018) Petri nets and Automatic Control: A historical perspective, Annual Reviews in Control,Volume 45, Pages 223-239,ISSN 1367-5788,https://doi.org/10.1016/j.arcontrol.2018.04.006.

[14] Drakaki, Maria, and Panagiotis Tzionas. 2017. "Manufacturing Scheduling Using Colored Petri Nets and Reinforcement Learning" Applied Sciences 7, no. 2: 136. https://doi.org/10.3390/app7020136

[15] Gerini, C., Sciomachen, A. Evaluation of the flow of goods at a warehouse logistic department by Petri Nets. Flex Serv Manuf J 31, 354–380 (2019). https://doi.org/10.1007/s10696-018-9312-3

[16] Cheng, F., He, S. (2010). How to Construct an Automated Warehouse Based on Colored Timed Petri Nets. In: Sobh, T., Elleithy, K. (eds) Innovations in Computing Sciences and Software Engineering. Springer, Dordrecht. https://doi.org/10.1007/978-90-481-9112-3-38

[17] S. Hsieh (1998) A Petri-net-based structure for AS/RS operation modelling, International Journal of Production Research, 36:12, 3323-3346, DOI: 10.1080/002075498192085

[18] Gong, Y., de Koster, R.B.M. A review on stochastic models and analysis of warehouse operations. Logist. Res. 3, 191–205 (2011). https://doi.org/10.1007/s12159-011-0057-6

[19] Chionglo, John Frederick (2018) A Petri Net Model for the 8-Puzzle Dynamic and Interactive Process Model plus Two Prototypes as PDF Form Applications. https://doi.org/10.13140/RG.2.2.12694.65607

[20] Uludağ, Onur. 2014 "GridPick: A high density puzzle based order picking system with decentralized control." PhD dissertation., Auburn University,

[21] Rausch,M., Hanisch, M (1995) Net condition/event systems with multiple condition outputs. https://doi.org/10.1109/ETFA.1995.496811. Vol.1, 592-600.

[22] Hanisch, M., Luder, A., Thieme, J., (1998) A Modular Plant Modeling Technique and Related Controller Synthesis Problems. In SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218) (Vol. 1, pp. 686-691). IEEE.

[23] A. Lobov, C. Popescu and J. L. M. Lastra, "An Algorithm for Siemens STL representation in TNCES," 2006 IEEE Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic, 2006, pp. 641-647, doi: 10.1109/ETFA.2006.355453.

[24] M. Xavier, S. Patil, V. Dubinin and V. Vyatkin, "Formal Modelling, Analysis, and Synthesis of Modular Industrial Systems Inspired by Net Condition/Event Systems", Lecture Notes in Computer Science book series, 2023.

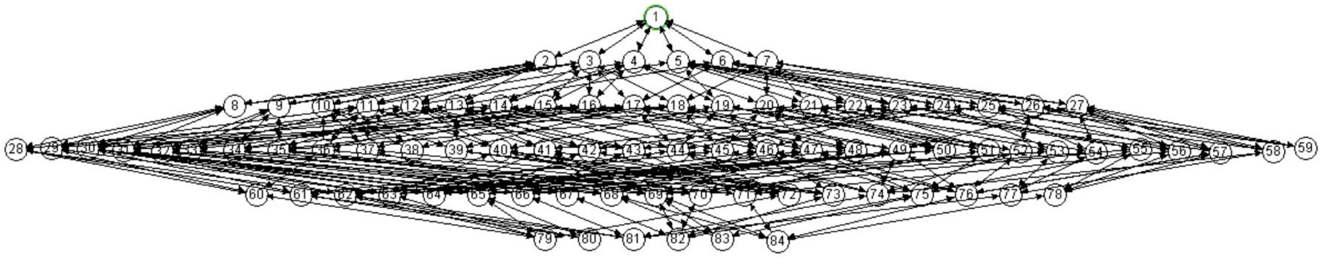

Fig. 9. State space for 3-by-3 PBS with more free locations, i.e., three free locations. See calculations in the expression (3).