



Journal of Applied and Computational Mechanics



Research Article

Reducing retrieval time in Automated Storage and Retrieval System with a gravitational conveyor based on Multi-Agent Systems

Imén Kouloughli¹, Pierre Castagna², Zaki Sari¹

¹Manufacturing Engineering Laboratory of Tlemcen, Department of Electrical Engineering and Electronics, University of Tlemcen
13000, Algeria, kouloughli_imen@hotmail.com,

²LUNAM University, QLIO Department, University of Nantes
44000, France, pierre.castagna@univ-nantes.fr

Received June 15 2017; Revised July 09 2017; Accepted for publication July 10 2017.

Corresponding author: Imén Kouloughli, kouloughli_imen@hotmail.com

Copyright © 2018 Shahid Chamran University of Ahvaz. All rights reserved.

Abstract. The main objective of this study is to reduce the retrieval time of a list of products by choosing the best combination of storage and retrieval rules at any time. This is why we start by implementing some storage rules in an Automated Storage/Retrieval System (Automated Storage and Retrieval System: AS/RS) fitted with a gravity conveyor while some of these rules are dedicated to storage and others to retrieval. The system is seen as a Multi-Agent System (MAS) where the produced agents are reactive agents that can interact to achieve a behavior (organizing the store). Our MAS is characterized by a decentralized control, which means that there is no preset plan. The produced agents exchange information such as their color, their distance from the output station, etc. Each product merely applies a set of behavioral rules. The aim is to choose the best product to be retrieved in the shortest possible time. The product-type agents have no cognitive ability, but still perform complex tasks.

Keywords: AS/RS fitted with gravity conveyor, Storage / retrieval, Combination of rules, MAS, Retrieval time.

1. Introduction

The automated storage and retrieval systems (AS / RS) are used to handle, store, and retrieve materials with accuracy, precision, and speed under a certain degree of automation [1]. These systems consist of a series of storage racks served by a storage/retrieval (S/R) machine that moves along aisles separating the racks, and one or more input/output (I/O) stations known as pick-up/delivery (P/D) stations. The AS / RS help improve the productivity in manufacturing and distribution facilities, reduce the storage costs, meliorate the material flow and resource management. Moreover, they have great interface flexibility with other components of flexible manufacturing systems (FMS) such as systems of transport, control, etc. They can provide all kinds of loads (tools, materials, pallets, products, support, etc.) with fast response times in order to meet requests of production and warehousing, and also distribution applications. In addition, they use a minimum space.

Various types of AS/RS have been developed, i.e. the unit-load AS/RS, multi-aisles AS/RS, AS/RS with sliding racks, the mini-load or reduced load AS/RS, carousel AS/RS, man-on-board AS/RS, deep-lane AS/RS, gravity conveyor AS/RS, etc. In this study, automated storage and retrieval systems (AS/RS) fitted with gravity conveyors are investigated.

Many studies have been carried out on AS/RS. Roodbergen et al. (2009) presented a literature review in the fields of design [2], estimation of cycle times, storage policies, and dwell point position of storage/retrieval machines as well as the scheduling of storage/retrieval requests. Azzouz et al. [3] used the Branch and Bound method to optimize the dimensions of an AS/RS



fitted with a gravity conveyor. In another study [4], the author optimized the dimensions of an AS/RS with a gravitational conveyor for a minimum cycle. On the other hand, Azzouz et al. [5] performed a synthesis on dimension optimization of the same kind of AS/RS. Gaouar [6] proposed a heuristic to improve the performance of that type of AS/RS. Sari et al. [7] developed a continuous mathematical expression for the cycle time in case of an AS/RS with a gravity conveyor. Ouhoud et al. [8] validate travel time models for single cycle displacements of the S/R machine in the multi-aisle automated storage and retrieval system (AS/RS) by using experimental computer simulation.

Several studies have been carried out in the field of multi-agent systems applied to industry. Mansour et al. [9] presented a flexible software solution to control intelligent multi-agent manufacturing systems "FMCC". This software allows users to edit various manufacturing sequences for each product with a multitude of possible combinations (flexibility in the order of manufacturing operations, the maximum waiting time for a given product, the minimum waiting time between two production operations, etc.). The main elements of the developed software are the OPC protocol, RFID technology, and multithreading. Indriago et al. [10] demonstrated the possibility to apply almost directly a holonic discrete-event based reference architecture to hybrid control systems. A case study of industrial electricity generation process was taken, specifically a combined cycle plant (CCP), to verify the proper operation of the proposed architecture. In another study, authors [11] modeled the coordination of different collaborative areas, both inside and outside a supply chain, using coordination models of coalition formation as proposed in MAS. They suggested an agent approach, a coalition formation algorithm, and a protocol of interaction between agents necessary to implement the distributed coordination. They illustrated their approach by an example of the avionics industry field. In his paper, Atalla F. Sayda [12] addressed the different aspects of designing a multi-agent system (MAS) to manage and control complex industrial facilities from conceptualization to deployment. A thorough literature survey was done to provide a complete illustration of what had been accomplished in academia and industry for this topic. Pannequin et al. [13] suggested a platform based on the emulation of the operating system, therefore, it is possible to represent complex systems with realistic scale. They introduced modeling primitives that allow representing aspects that are specific to be controlled by the product. Then, they specified the way a control system (possibly multi-agents) can be integrated into their platform. Finally, they used their approach to develop the emulation model of an industrial site. Clair [14] developed a production management simulation platform to compare different approaches. The problem consists of allocating tasks to operators of machines to arrange work pieces according to a list of commands. The author wished to model the problem as a distributed constraint satisfaction problem (DCSP) and solve it using self-organization techniques in order to offer solutions capable of responding to the environmental dynamics. This approach was supposed to put agents into action in order to work together and achieve the best possible solution to the proposed scenario. After analyzing the problem, several developed standard approaches were presented and dealt with the proposed self-organizing approach. To compare the advantages and limitations of this approach, the results of some tests were examined. Reguieg [15] proposed an approach for a multi-agent architecture realized on the Jade platform based on the development of a coordination protocol. The suggested model consists essentially of five components: Agent Resource (AR), Interface Agent (IA), Agent Supervisor (AS), Data Manager Agent (DMA), and Agent Coordinator (AC). The coordination protocol ensures the exchange of messages between agents; it provides them with predefined behaviors as well. Since the objective of our approach is to manage a dynamic production, the interaction between agents is needed to solve the problems of resource allocation and conflict management, and also to have a better reaction to a logistics failure. The agent coordinator (AC) is provided with a learning mechanism. The learning of AC facilitates decision-making for the production system and requires less execution time. This allows the agent AC to be free for potential treatments, and makes the system faster, flexible and more responsive. Taghezout [16] proposed the integration of agents into interactive Decision Support Systems (DSS) to solve some uncertainty problems in the system of Dynamic Production Scheduling. The developed system gives decision-making centers the opportunity to make decisions in a dynamic environment. More precisely, Integrated Production Stations (IPS) are supposed to be given the appropriate behavior to perform practical operations and, at the same time, react to the complexity of problems caused by the dynamic scheduling in real situations. These agents express their preferences, based on ELECTRE III method, to solve the differences. The negotiation mechanism is based on the contract net protocol (CNP). The protocol developed in JADE provides message exchanges between agents and gives them predefined behaviors. The approach has been tested through simple scenarios. In a typical study, [17] the author dealt with the way multi-agent systems (MAS) could help in the management of production, distribution and inventory. He used the CDPD approach (Demand Driven Production and Distribution) to examine the way MAS could solve the main problem of this approach, i.e. the centralization of decisions. For this purpose, MASs and the distributed planning were presented. Moreover, an idea was conducted on multi-agent architectures and on the types of inter-agent interactions that are capable to allow for the planning of production, storage and distribution in the supply chain. Kouloughli et al. [18] proposed a model implemented in the multi-agent platform Jade (Java Agent Development Framework) which was based on three kinds of agents: DF agent (Directory Facilitator), the product agent to retrieve and bin agent. The retrieval of products was generally based on a specific retrieval rule; each rule was founded on one mathematic metric. The objective was to bring out communications between different agents of their system to choose the best bin from where the product would be retrieved (the retrieval rule with the minimal metric). Wenrong et al. [19] examined the potential adoption of DI Approaches in warehouse management systems (WMS). In their research, they discussed the challenges in the warehouse management and compared these challenges to the characteristics of manufacturing problems to which the DI approach was suited. A new approach, called COSAH (Container Stacking via multi-agent Approach and Heuristic method), was proposed by Gazdar [20] to simulate, solve, and optimize the available storage space in order to handle departures and arrivals of containers in a river or a sea port. In general, the obtained results which were clearly presented and illustrated, show the effectiveness of COSAH in particular and also that of a distributed heuristic optimization method by combining the two concepts of Agent and Heuristic. Thomas et al. [21] highlighted that already several industrial and scientific applications in the domain of holonic

manufacturing were implemented and proved their reliability and capacity to render manufacturing processes sustainable. Nevertheless, a lot of challenges (scientific as well as applicative), especially concerned with the joint use of predictive and reactive distributed control systems, still exist. The future of these research works will have to determine how, when, and for what the control should switch between traditional ERP systems and new control systems with distributed intelligence. Parera [22], by using an example, demonstrated how agent-based modeling can be used for the health service supply chain design. Naveh et al. [23] worked on multi-task systems in general, and allocated tasks to agents in a multi-task system in particular. So far, the reduction of the retrieval time in an AS / RS with gravitational conveyor has been treated by heuristics, analytic methods, etc., but never with Multi-Agent Systems (MAS). In the present study, a new technique dedicated to automated storage and retrieval (multi-agent systems for storage and retrieval) is proposed. It aims to continually reduce the retrieval time of a list of products.

2. AS/RS with a Gravity Conveyor

An AS/RS fitted with a gravity conveyor (GC) consists of a single deep rack containing a number of bins, each equipped with an inclined gravity conveyor. A bin has several locations and each location is used for storing a single product. The system includes two machines: one on the front side of the rack for storing products, and the other one on the rear side of the rack for retrieving the products. Both sides of the rack are connected by a restoring conveyor inclined in the opposite direction of the racks. Products are stored by the storage machine on the first face of the rack and slide along the gravity conveyor until they reach the first empty location of the bin. Fig. 1 shows an AS / RS supplied with a gravity conveyor.

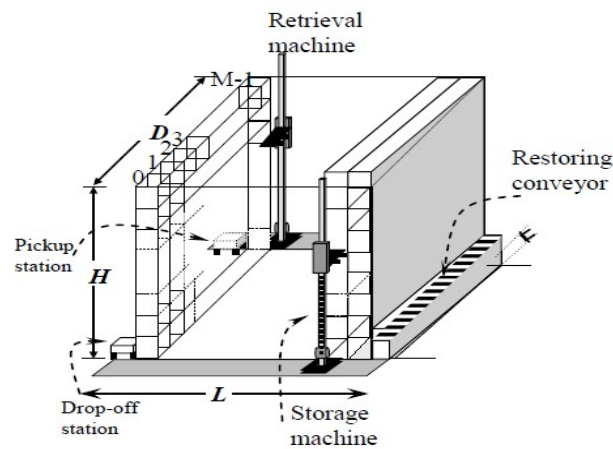


Fig. 1. Configuration and components of an AS/RS with a gravity conveyor.

Notations:

L, H, D: length, height, and depth of the rack of the gravity conveyor AS/RS;
M: number of locations in a bin, and number of layers in a rack;
N: number of bins in the rack;
Nl: number of bins per row;
Nh: number of bins per column;

3. Multi-Agent Systems (MAS)

An agent can be defined as a physical or abstract which is capable of acting on itself and on its environment having a partial representation of that environment (Fig. 2); it can communicate with other agents and shows a behavior that results from its observations, knowledge, and interactions with other agents [16].

Agents have two tendencies:

- A social tendency addressing the community, i.e. the mechanisms and associated knowledge about the group's activities
- An individual tendency, i.e. mechanisms and knowledge related to the rules of internal functioning of the agent. An agent can therefore be characterized by its role, specificity, objectives, and functions (its behaviors in general).

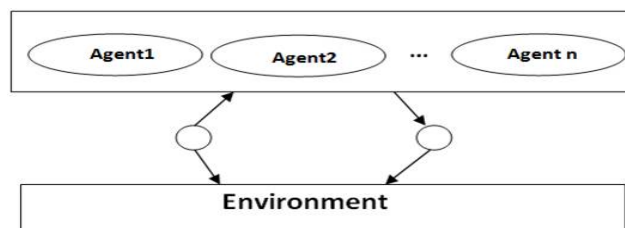


Fig. 2. Multi-Agent System.

4. Netlogo Simulator

NetLogo is a multi-agent programming language and modeling environment for simulating natural and social phenomena. It is particularly well suited for modeling complex systems evolving over time. Modelers can give instructions to hundreds or thousands of independent "agents" which all operate concurrently [24]. This makes it possible to explore connections between micro-level behaviors of individuals and macro-level patterns that emerge from their interactions. NetLogo is particularly suitable for modeling complex systems evolving over time. "Modelers" can give instructions to hundreds or thousands of "agents" operating independently from each other. This allows exploring the links between the behavior of agents as well as the general patterns (group or mass behavior) that emerge from the interactions between many individuals.

These simulations cover several areas of natural and social sciences including biology and medicine, physics and chemistry, mathematics and computer engineering as well as economics and social psychology. Many learning sessions are currently under development based on NetLogo models. Moreover, NetLogo offers a powerful tool for participatory simulations in the classroom, called HubNet which represents the new generation of a whole series of multi-agent modeling languages that started with StarLogo. It was developed from the bases provided by the StarLogoT software which supplies a series of significant new features and a completely redesigned language and user interface. NetLogo is written in Java and can run on all major systems (Mac, Windows, Linux and others); it runs as an independent application. Models can even be saved as Java applets and run in all modern web browsers.

5. Development of Our Model

In our system, the storage and retrieval of products in a rack with a single row of bins is merely studied. This means that the length (L) of the rack is equal to 1 (Fig. 3). This choice is made because the same scenario is repeated in the other rows. The arbitrary values of 35 and 46 are chosen for the height and depth of the rack, respectively. It is worth mentioning that depending on the studied configuration of the rack, our program allows easily changing the values of the past two dimensions.

Hypotheses:

It is clear that all the following values can be changed at any time in our program.

Number of product types = 140 colors.

Rack width = number of rows in the rack = 1 row.

Rack height = number of bins in the rack = 19 bins.

Rack depth = number of layers in the rack = 47 layers.

Initial fill rate of the rack = 48%.

Initial state of the rack: E0.

Horizontal displacement speed of the retrieval machine = vertical displacement speed of the retrieval machine = 1.

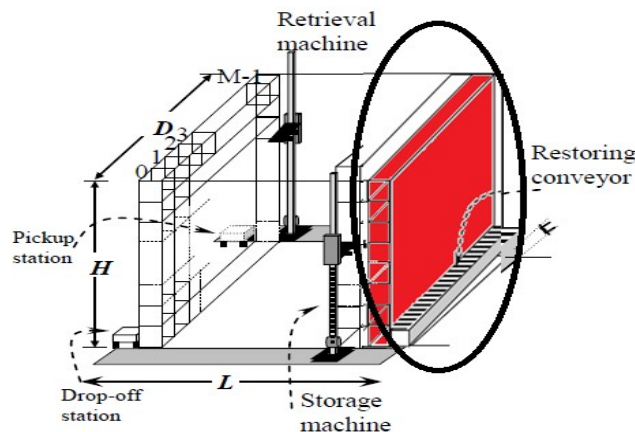


Fig. 3. The AS/RS with a gravity conveyor in the SRMAS model

This model (Fig. 4) consists of a rack containing 19 bins; 18 for the storage of products and 1 for their retrieval. Each type of product is assigned a color. In our model, 140 colors are used. Our program offers the possibility to modify the previously selected values, at any time.

Our system is regarded as a reactive multi-agent system where there are product agents (agents called turtles in Netlogo) (Fig. 5) that move according to the "If Action then Reaction" rule. These agents communicate their distances from the exit station (which is considered as a patch agent in Netlogo) in order to minimize the retrieval time of products to the maximum.

Product agents operate according to the following rule:

If <internal state> and <perception state> then <action>.

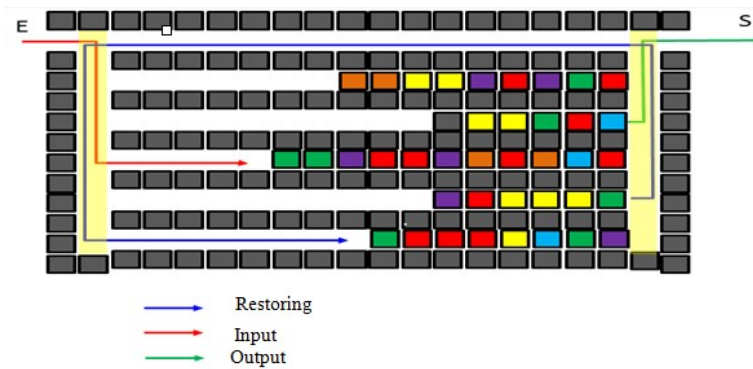


Fig. 4. Principle of operation of the SRMAS model.

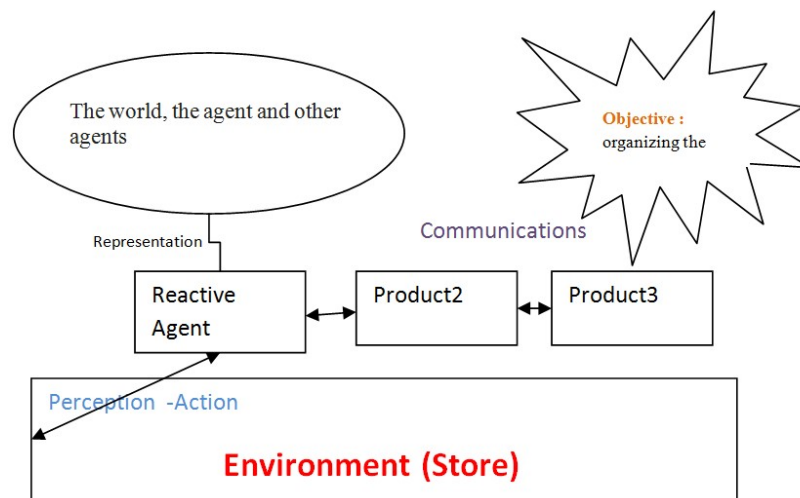


Fig. 5. Representation of a Multi-Agent System (MAS).

6. Model Functioning

The “Prepare” procedure in the following figure helps to erase all turtle agents that exist; “Init” puts the rack back in the proper initial state. For example, if “SaveStoreState” is put on, “True” and storage and/or retrieval actions are performed; if “Init” is pushed, then the last state of the store will be set as the initial state.

The input box “RuleS” allows choosing the one among the possible storage rules to be applied. Similarly, the input box “RuleD” allows choosing the one among the possible retrieval rules to be applied. Combining these two rules, a simulation is carried out using the Sim procedure. This means that there will be a text file input, called “D.txt”, which contains the list of colors of products to be retrieved. This list of products is retrieved according to the “RuleD” rule. Then, similar products (same color) are stored again according to the “RuleS” storage rule in order to have a larger variation in the types of products (Fig. 6).

The time required to retrieve this list of products is calculated, and then, the storage is rearranged by storing the products of the same type at each single time. After a certain time (previously fixed), one has to check if the retrieval time decreases as the store is arranged according to the “RuleS” storage rule.

7.1 Storage rules

Three storage rules were defined in our program:

Storage rule RE1

RE1 corresponds to random storage where the predefined NetLogo function is used. It returns an integer random number between 0 and 17.

Storage rule RE2

This rule is used to store a product in the bin that contains at least one product of the same type. If this is not possible, then, the product is stored in an empty bin.

Storage rule RE3

This rule tries to find the bin that contains the minimum of products of the same type as the one that is supposed to be stored; the objective is to have the greatest variation of products in each bin.

7.2 Retrieval rules

Retrieval rule R1

This rule checks whether a product of the same color, as that which it is desired to retrieve, exists in the first layer; if there are many, it chooses the one closest to the storage station; if none exists, then, it goes to the second layer.

Retrieval rule R2

In this rule, the rack is divided into four squares; the first one is examined and if the right product to be retrieved is not found in the first square, the second one is examined.

Retrieval rule R3

First, the aim is to seek all products of the same type as the one that is supposed to be retrieved. Then, the distance between the product and the output station is calculated (this is done by drawing a straight line between two points) and the product having the minimum distance is selected (Figure 7).

Retrieval rule R4

This rule takes the number of products before retrieval and also the distance between the latter and the exit station into account.

Retrieval rule R5

This rule takes into account the number of products existing before the one that is supposed to be retrieved; it also considers the maximum between the horizontal and the vertical distances travelled by the retrieval machine (Figure 7). It is assumed that as the system comes down to a single row, the horizontal displacement is equal to a unit of distance.

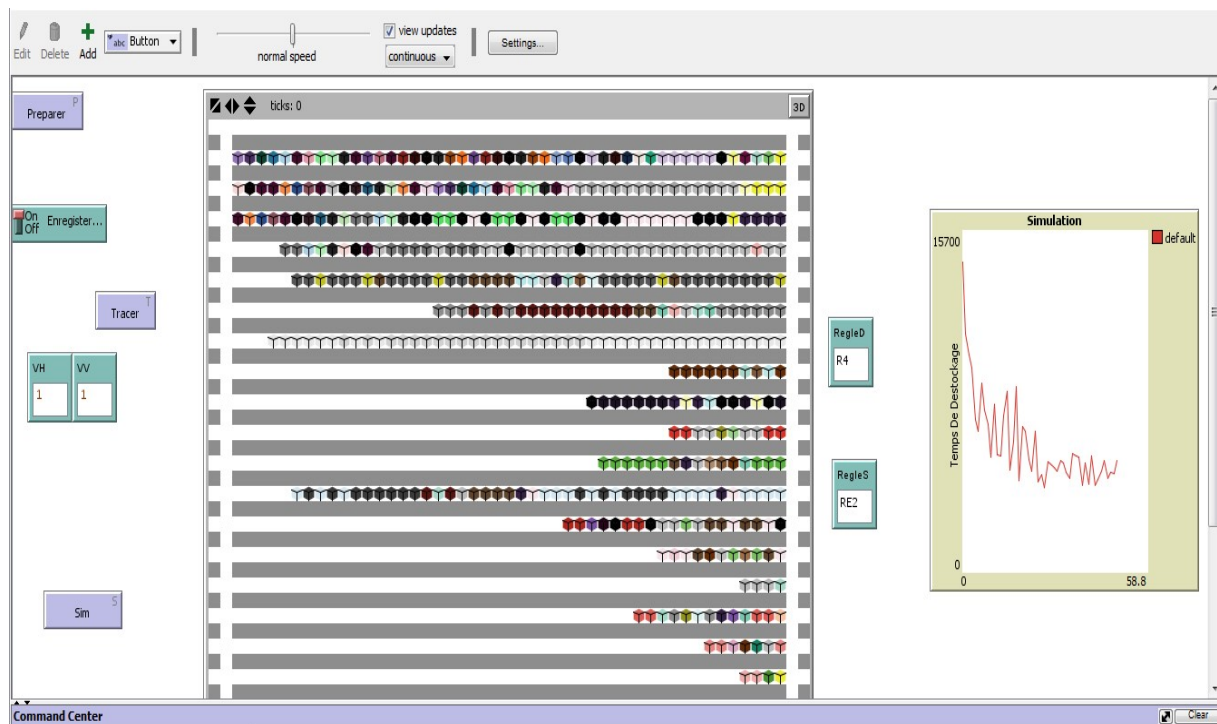


Fig. 6. Main interface of a model SRMAS.

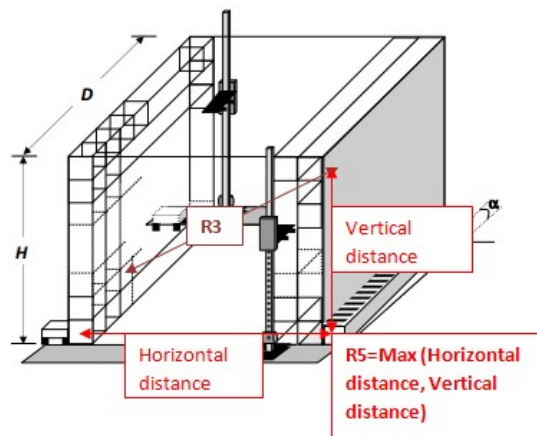


Fig. 7. Operating principle of retrieval rules R3 and R5.

7. Travel Time of the Retrieval Machine

It seemed interesting and appropriate to sketch and measure the travel time of the retrieval machine. Therefore, the time of horizontal displacements of the retrieval machine must be calculated as well as that of the vertical ones. The rest position of the retrieval machine was set right next to the output station (Fig. 8).

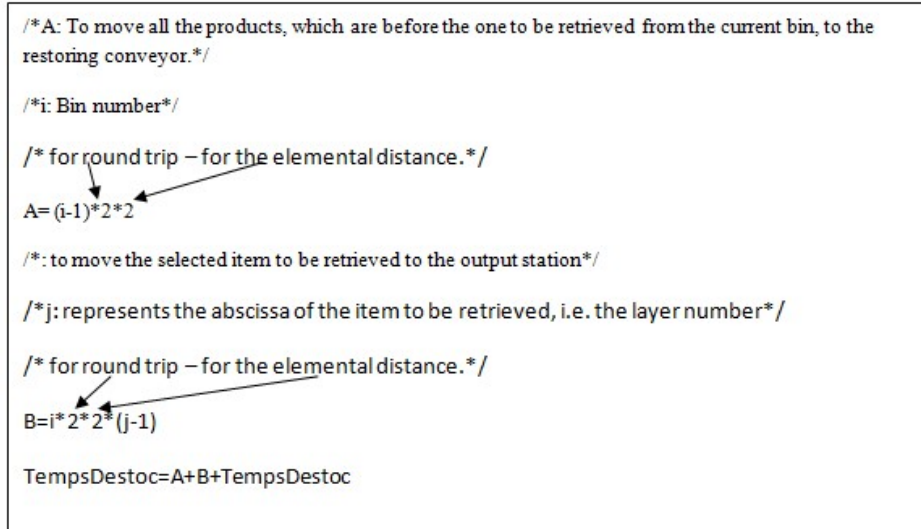


Fig. 8. Travel time of the retrieval machine.

8. Results

The average travel time of the retrieval machine is measured, each time starting from an initial state of the store. Each curve, in what follows, models fifty travel times of the retrieval machine (since there are fifty iterations in the “Repeat” loop of Sim procedure). Fifty travel times are simply calculated because they are sufficient to have interesting and interpretable results. However, these numbers can be increased.

The horizontal and vertical speeds of the S/R machine are represented by global variables in the main interface of our program. Their values may be changed at any time. For example, they were all assigned the value 1.

9.1 Initial state E0 of the store

The initial state (E0) of the store with a fill rate of almost equal to 48%. (Fig. 9) is chosen. The storage and retrieval rules are applied, starting from that state E0. Then, the retrieval of a predefined list of products is carried out.

The results are summarized in the following Table1.

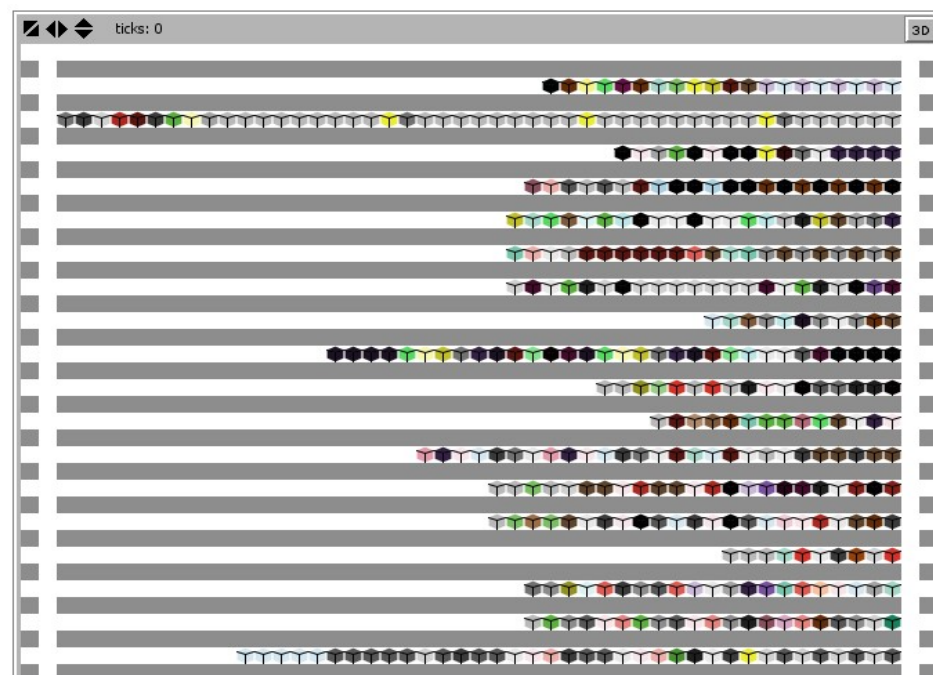


Fig. 9. Initial state of the rack

Table 1. Average retrieval time achieved by combining two rules

Storage rule	RE1	RE1	RE1	RE1	RE1	RE2	RE2	RE2	RE2	RE2	RE3	RE3	RE3	RE3	RE3
Retrieval rule	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
Average travel time of retrieval machine × 10 ⁻³	32,38	39,63	42,17	37,61	43,95	8,99	13,98	15,81	5,9	6,08	11,47	8,81	8,88	10,27	7,21

9.2 Modeling curves of the travel time of retrieval machine

Storage rule "RE1" combined with retrieval rules "R1, R2, R3, R4"

Random storage is first studied as a reference; it is then compared to the heuristic storage. It is well noted that for random storage, the retrieval time is unstable over time (Fig. 10).

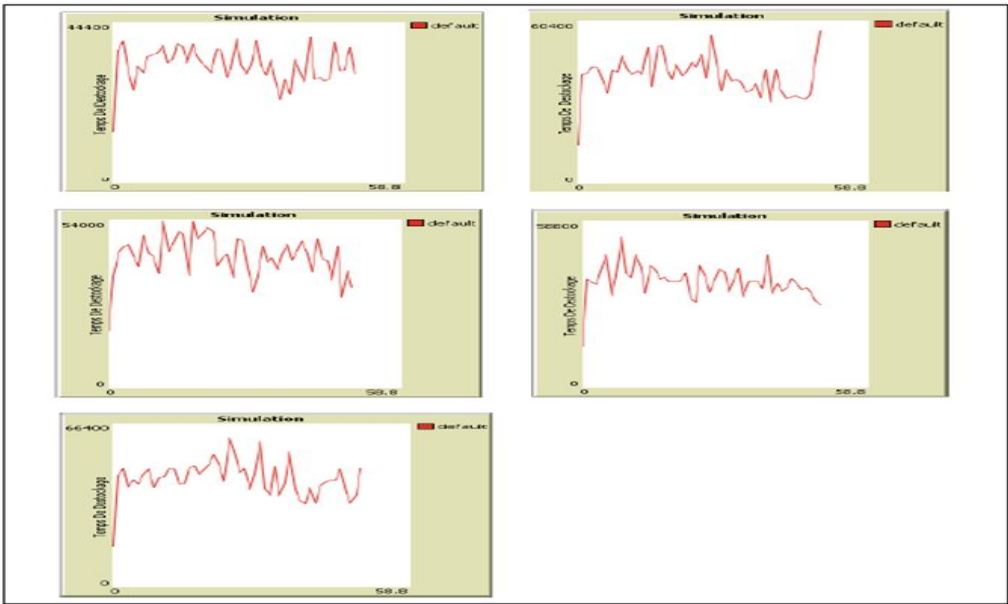


Fig. 10. Retrieval time resulting from random storage combined with the 5 rules of heuristics.

Storage rule "RE2" combined with retrieval rules "R1, R2, R3, R4"

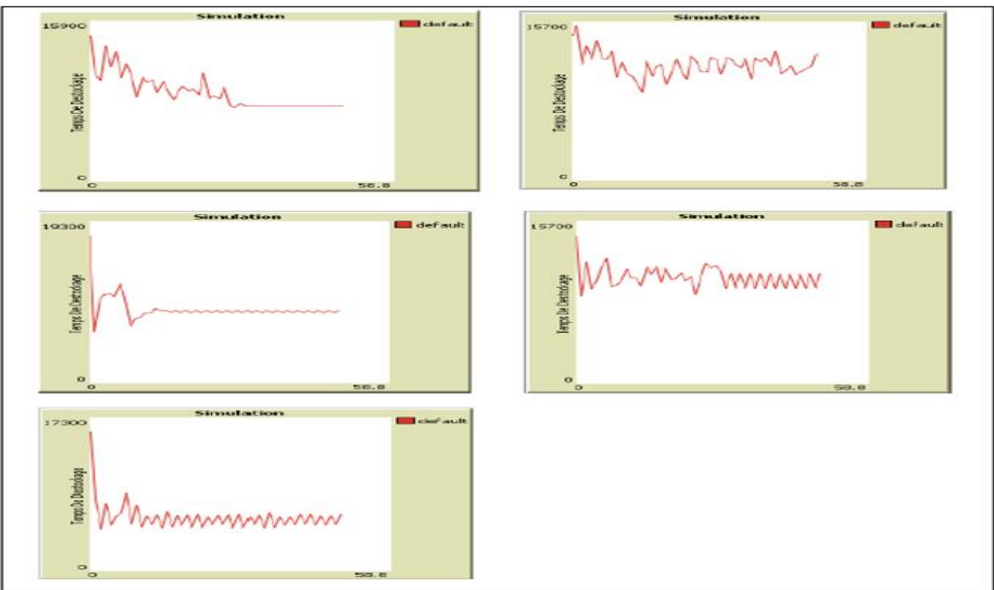


Fig. 11. Retrieval time from storage by RE2 combined with the 5 heuristic rules

It is well noted that when rule RE2 is applied, the 5 obtained average retrieval times are much smaller than those obtained when random storage is used. The average retrieval time that results from the combination of RE2 and R4 turns out to be the best, compared to that resulting from the combination of RE2, R2, RE2, R3, and R5 (Fig. 11). The retrieval rule R3 may achieve good results if the average number of products, which is stored before the one that is supposed to be retrieved, is small. The opposite situation would not be interesting because a great number of products should be restored each time.

Storage rule "RE3" combined with retrieval rules "R1, R2, R3, R4"

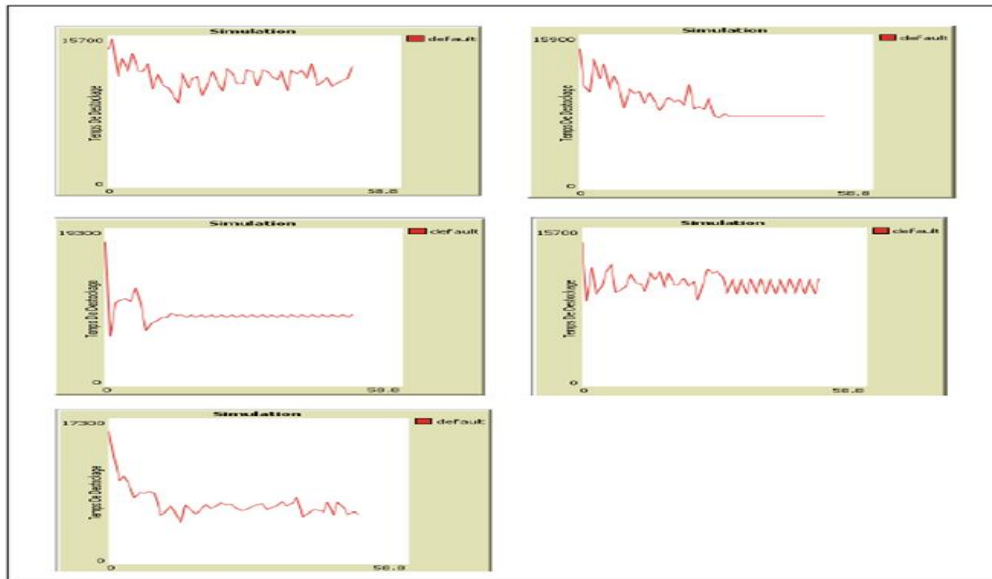


Fig. 12. Retrieval time from storage by RE3 combined with the 5 heuristic rules.

The best obtained average retrieval times are resulted from the combination of the storage rule RE3 with the retrieval rules R2 and R3. A very small number of similar products are found in the same bin when the rule RE3 is applied. Therefore, during retrieval, the number of products to be restored will be small (Fig. 12).

9.3 Interpretation of results

Storage rule RE1

It is clear that based on random storage (RE1), the travel time of the retrieval machine gives poor results compared with those obtained when the storage rules are applied.

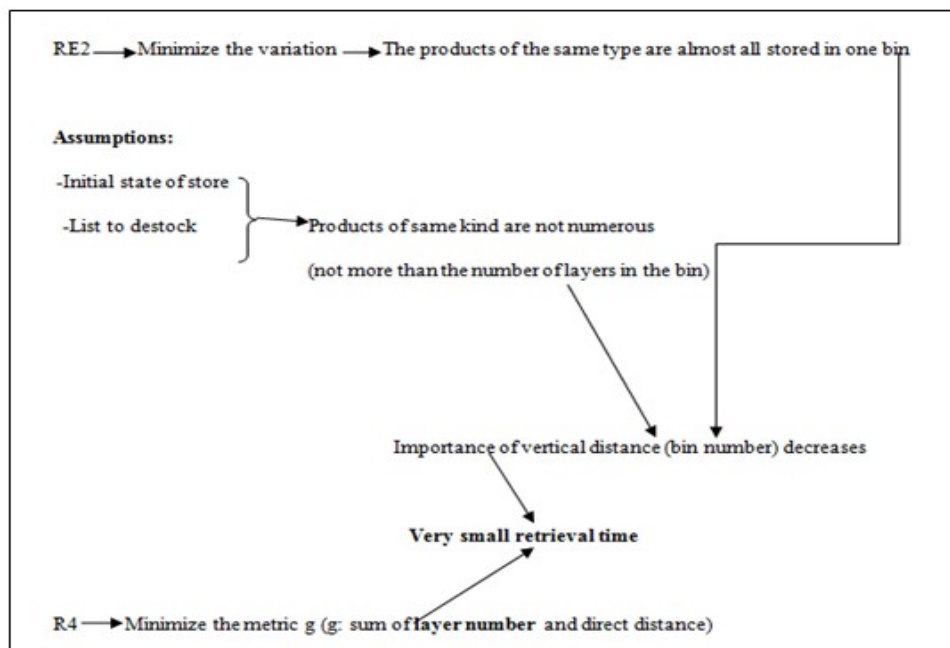


Fig. 13. The combination (RE2, R4).

Storage rule RE2

When the storage rule RE2 is combined with the various retrieval rules R1, R2, R3, R4 and R5, it is found that the minimum obtained retrieval time corresponds to the combination of (RE2 & R4); this may be explained as follows:

Rule RE2 minimizes the variation of product types in the same bin, and this allows finding the product to be retrieved in the first layers. Rule R4 computes the average of the number of products to be restored and the direct distance between the product to be retrieved and the output station. The smaller the number of layers, the shorter the direct distance between the product to be retrieved and the output station, and the smaller the number of products before the product is retrieved. This results in the best retrieval time for the combination of (RE2 & R4) (Fig. 13). The combination of (RE2 & R5) gives a good result because the retrieval rule R5 which is combined with any storage rule gives a good result, i.e. if the storage procedures are changed at any time, then it is preferable to apply the retrieval rule R5 (Fig. 14).

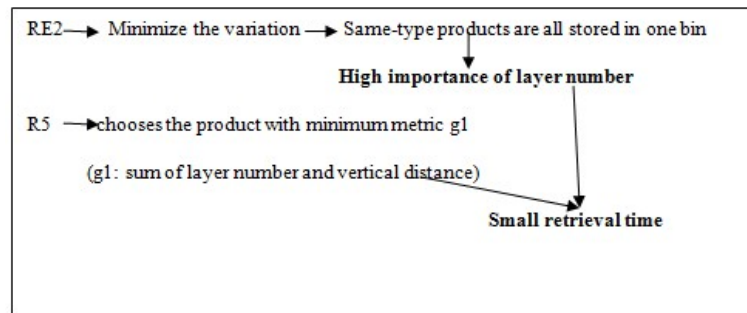


Fig. 14. The combination (RE2, R5).

Combination of (RE2 & R1) gives a good result because R1 tends to look for the desired product by examining all the layers in a row of the rack (i.e. the product is searched in the first layer of the first bin, then, in the first layer of the second bin, ...). As RE2 minimizes the variation of products in a single bin, then, there is a good possibility that the product is found in the first layers and this should give a good result (Fig. 15).

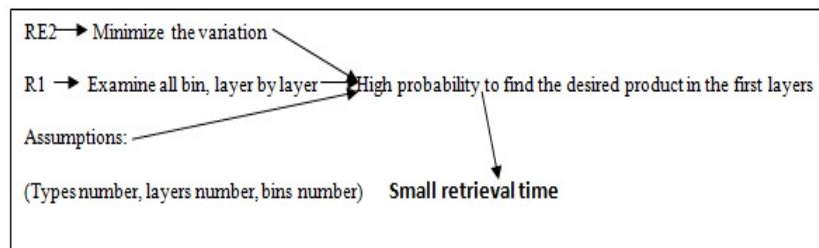


Fig. 15. The combination (RE2, R1).

It is not recommended to combine RE2 with R2 because it is likely to find the desired product in the last bins of the rack (for example, when one tries to find a yellow item and bin 17 is the one with most yellow items). Therefore, several squares of the rack should be examined before reaching the right one (Fig. 16).

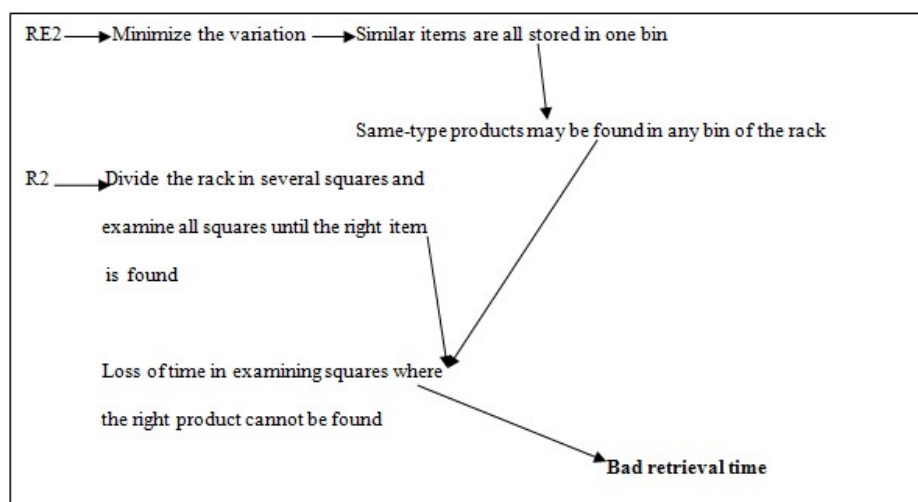


Fig. 16. The combination (RE2, R2).

The Combination of (RE2 & R3) must be avoided as well because R3 does not take the layer that contains the product to be retrieved into account; however, if RE2 is applied as a storage rule, then, there is more possibility to find all the desired products stored in the same bin. This shows the importance of the number of the layers containing the product (Fig. 17).

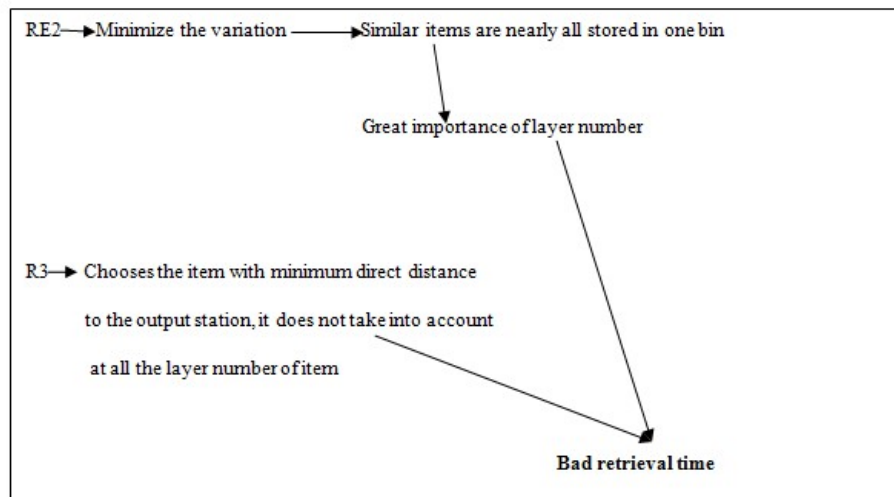


Fig. 17. The combination (RE2, R3).

Storage rule RE3

When the storage rule RE3 is combined with the various retrieval rules R1, R2, R3, R4 and R5, it is found that the minimum obtained retrieval time corresponds to the combination of (RE3 & R5); this may be explained as follows:

According to the storage rule RE3, it is likely to find the right product to be retrieved in any layer. This shows the great importance of the layer number. Rule RE3 maximizes the variation of products in each bin, and given the assumptions at hand, there is a good possibility to find the right product in the first bins. This means that the bin number is still important. As the retrieval rule R5 combines these two important factors (layer number and bin number), it leads to the minimum storage time (Fig. 18).

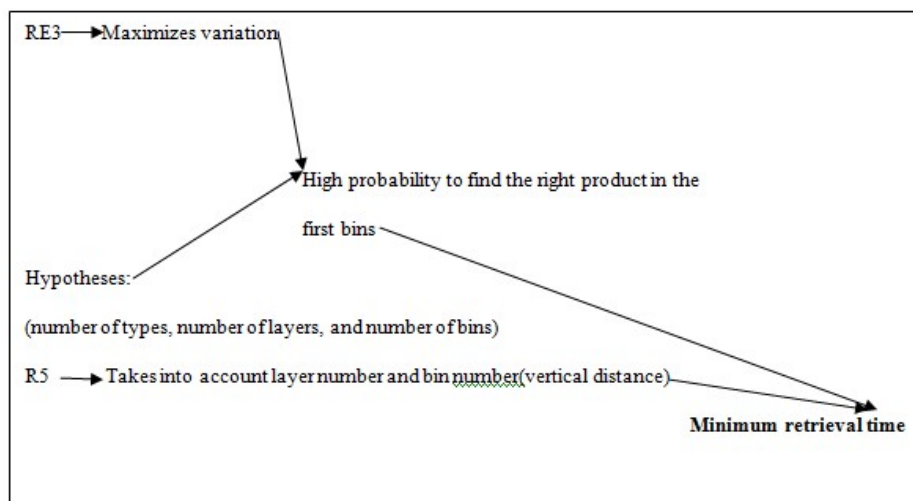


Fig. 18. The combination (RE3, R5).

The combination of (RE3 & R2) in (Figure 19) gives a good result because as previously noted, there is a great possibility to find the right product in the first bins; indeed R2 divides the rack into four squares and begins to examine the first one (the square that contains the first bins), then the second, This should lead to a good result.

Combination of (RE3 & R3) in (Figure 20) also gives a good result; this may be explained as follows:

When the storage rule RE3 is used, a great importance is given to the bin number and the layer number. According to the Pythagorean Theorem, it can be concluded that the direct distance between a product and the output station is equal to the square root of the sum of the square of the bin number (vertical distance) and that of the layer number.

$$\text{Direct distance} = \sqrt{(\text{layer number})^2 + (\text{bin number})^2} \quad (1)$$

Moreover, it is well known that the sum of (layer number + bin number) evolves (increases or decreases) in the same way as the direct distance; for this reason, a good result is obtained for the combination of RE3 with R3 (Fig. 20).

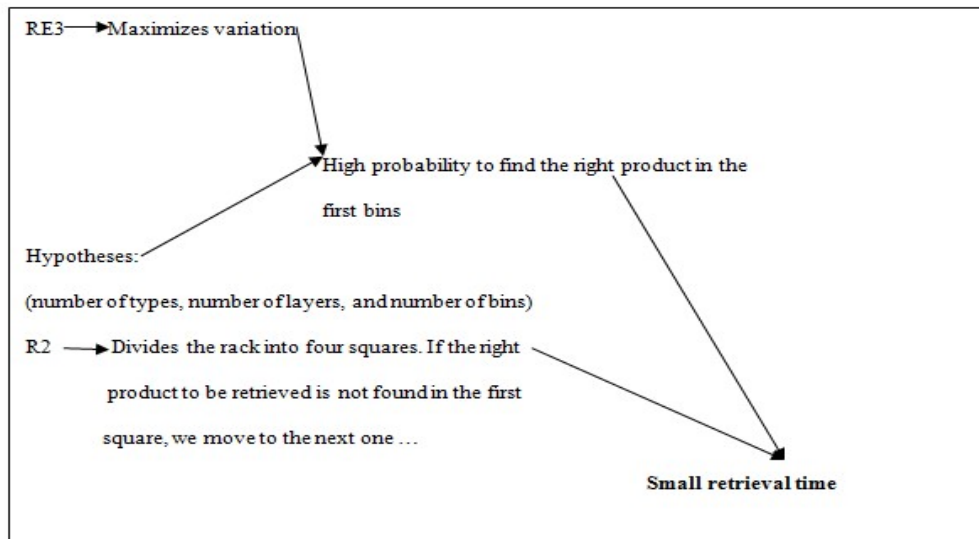


Fig. 19. The combination (RE3, R2).

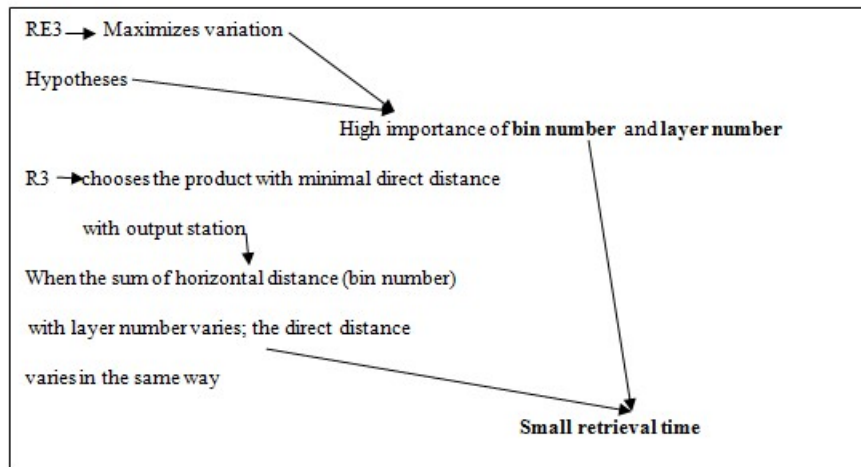


Fig. 20. The combination (RE3, R3).

Unlike RE2, and regarding RE3, one cannot be sure that the right kind of product is in the first layers. As R1 is supposed to examine each bin layer by layer, there is a risk of losing a lot of time going through layers unnecessarily (Fig. 21).

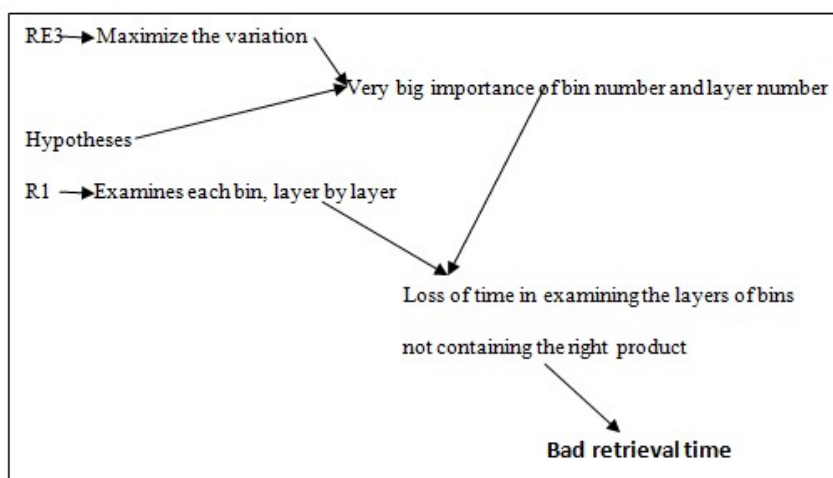


Fig. 21. The combination (RE3, R1).

Obviously, and regarding RE3, two things are to be considered, i.e. the bin number and the layer number. Even if the Journal of Applied and Computational Mechanics, Vol. 4, No. 1, (2018), 55-68

retrieval rule R4 takes the layer number into account, it does not combine it with the right distance (i.e. the vertical distance). This leads to a bad retrieval time (Fig. 22).

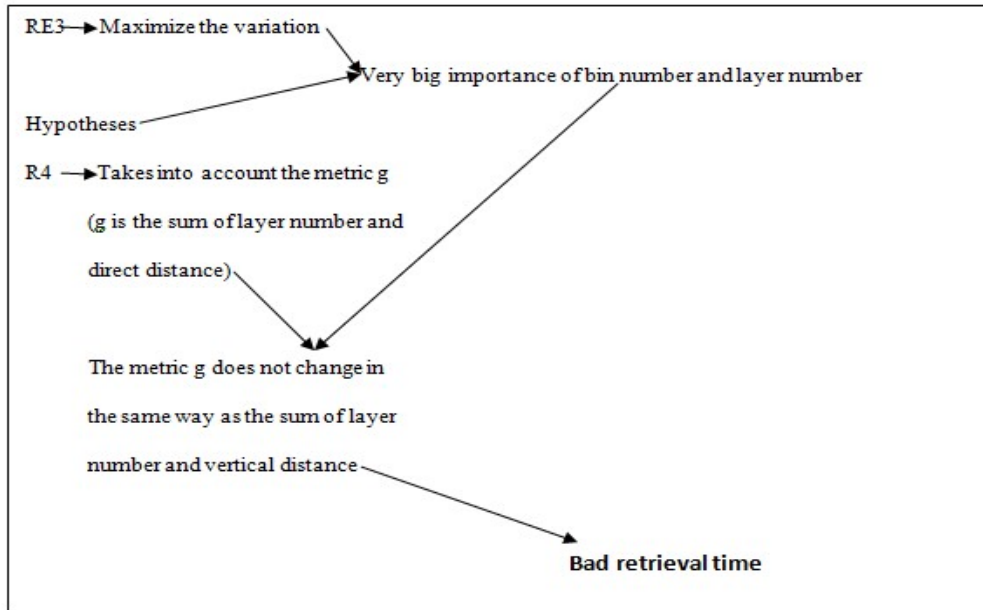


Fig. 22. The combination (RE3, R4).

Choosing the right combination of storage and retrieval rules depends on the following two important factors, namely the fill rate of the rack and also the way it was originally filled; i.e. if the initial state (E0) of the store is changed into E1, then the choice of the best combination of rules will certainly change. Therefore, it is highly recommended to add these two parameters in the main interface. The abscissa represents the real time which means any moment where the destocking operation begins. On the ordinate, we make correspondents to each time of the beginning of an operation of destocking which corresponds to the time required for the destocking in question that gives this zigzag allure. The results obtained depend on the initial state of the magazine, the rack fill rate, and the number of product types in the store. Viewing this system as a multi-agent system (MAS) is very important because when the product-type agents use the best combinations of storage and retrieval rules in the environment (the store), this should gradually improve the average retrieval time (which is the desired objective).

The product-type agents communicate their metric (in accordance with the applied retrieval rule) to the agent of the output station. It becomes possible, from there, to deduce the best choice for the product-type agent to be retrieved. This intelligence is shared between many reactive agents (products). Such an intelligent behavior emerges from the interaction between these reactive agents and the environment.

9. Conclusion

A completely new model, named SRMAS, was developed in this article. This model is novel and original because it merges two techniques carefully chosen and adapted to AS / RS gravitational conveyors. A storage and retrieval strategy (storage and retrieval rules) was first developed. The best combination of these rules was then selected in order to keep our store tidy. Then, our AS/RS fitted with a gravity conveyor was considered as a MAS (multi-agent system), where the agents are autonomous and can interact with each other. They may exchange information such as their type, distance to the exit station, etc. The merging of the two techniques led to a tidy rack. Our future perspectives will be to compare results between reactive MAS and cognitive MAS developed with JADE, securing our MAS with crypto algorithms and to face set of unpredicted events that can arrived. Finally, we will develop our own anthology.

Acknowledgment

This work was supported by the Manufacturing Engineering Laboratory of Tlemcen (Algeria); and the IUT (University Institute of Technology) of Nantes (France) with the professors Pierre Castagna and Zaki Sari.

References

- [1] Material Handling Institute, *Considerations for planning and installing an automated storage/retrieval systems*, Material Handling Institute, 1977.
- [2] Roodbergen, K.J., Vis, I.F. A survey of literature on automated storage and retrieval systems, *European Journal of Operational Research*, 194(2), 2009, pp. 343-362.
- [3] Azzouz, A., Sari, Z., Ghouali, N., La méthode de Branch & Bound appliquée à l'optimisation des dimensions d'un AS/RS à convoyeur gravitationnel, *Conférence Internationale sur la production, CIP2001*, June 2001, Alger pp. 9-11, 2001.
- [4] Azzouz, A., *Optimisation des dimensions d'un AS/RS à convoyeur gravitationnel pour un temps de cycle minimum*, Journal of Applied and Computational Mechanics, Vol. 4, No. 1, (2018), 55-68

Doctoral Dissertation, Thèse de magister, Université de Tlemcen Algérie, 2001.

- [5] Azzouz, A., Sari, Z., Ghouali, N., Une synthèse sur l'optimisation des dimensions d'un AS/RS à convoyeur gravitationnel, *Conférence Internationale sur la conception et la production intégrée*, October 2001, pp. 24-26.
- [6] Gaouar, N., *Amélioration des performances d'un AS/RS à convoyeur gravitationnel*, Doctoral Dissertation, Thèse de magister, Université de Tlemcen Algérie, 2001.
- [7] Sari, Z., Saygin, C., Ghouali, N., Travel-time models for flow-rack automated storage and retrieval systems, *International Journal of Advanced Manufacturing Technology*, 25(9-10), 2005, pp. 979-987.
- [8] Ouhoud, A., Guezzen, A., Sari, Z., Experimental Validation of Travel Time Models for Multi Aisle Automated Storage and Retrieval System in Class-Based Storage. *Electrotehnica, Electronica, Automatica*, 64(1), 2016, pp. 150.
- [9] Mansour, W., Jelassi, K., Flexible Software solution for intelligent multi-agent manufacturing systems. In *Electrical Sciences and Technologies in Maghreb (CISTEM), 2014 International Conference on*, November 2014, pp. 1-4.s
- [10] Indriago, C., Cardin, O., Rakoto, N., Castagna, P., Chacòn, E., H 2 CM: A holonic architecture for flexible hybrid control systems, *Computers in Industry*, 77, 2016, pp. 15-28.
- [11] Anane, D., Pinson, S., Aknine, S., Les approches agents pour la coordination d'activités dans les chaînes logistiques, *CAHIER DU LAMSADE 291*, University Paris-Dauphine, 2009.
- [12] Sayda, A.F., *Multi-agent Systems for Industrial Applications: Design, Development, and Challenges*, *Multi- Agent Systems - Modeling, Control, Programming, Simulations and Applications*, Dr. Faisal Alkhateeb (Ed.), InTech, 2013.
- [13] Pannequin, R., Thomas, A., Proposition d'une plateforme d'expérimentation sur le contrôle par le produit des flux de production. In 6ème Conférence Francophone de Modélisation et Simulation, MOSIM'06, April, 2006.
- [14] CLAIR, G., *Gestion de production par système multi-agent auto-organisateur*, Rapport de Master, 2, 2008.
- [15] Seddik, R., Elaboration d'un protocole de coordination dans un SMA pour la gestion de production dynamique: Vers une approche décentralisée, Doctoral Dissertation, Université Ahmed Ben Bella d'Oran1 Es Senia, 2011.
- [16] Taghezout, N., Conception et développement d'un système multi-agent d'aide à la décision pour la gestion de production dynamique, Doctoral dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2011.
- [17] Moyaux, T., Comment les syst mes multi-agents pourraient aider la gestion de la production, de la distribution et des stocks, 2001.
- [18] Kouloughli, I., Castagna, P., Sari, Z., Development of a Multi-Agent System (MAS) to optimize the retrieval time within an automated storage/retrieval system, *Electrotehnica, Electronica, Automatica*, 64(3), 2016, pp. 115.
- [19] Lu, W., Giannikas, V., McFarlane, D., Hyde, J., The role of distributed intelligence in warehouse management systems. In *Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics*, 2014, pp. 63-77.
- [20] Gazdar, M.K., Optimisation Heuristique Distribuée du Probleme de Stockage de Conteneurs dans un port, *Ecole Nationale des sciences de l'informatique*, Université de Manouba, 2008.
- [21] Thomas, A., Borangiu, T., Trentesaux, D., Holonic and multi-agent technologies for service and computing oriented manufacturing, *Journal of Intelligent Manufacturing*, 28, 2017, pp. 1501-1502.
- [22] Perera, L.C.M., Karunananda, A.S., Using A Multi-agent System For Supply Chain Management, *International Journal of Design & Nature and Ecodynamics*, 11(2), 2016, pp. 107-115.
- [23] Naveh, Y., Ronen, A., U.S. Patent No. 9, 430, 299. Washington, DC: U.S. Patent and Trademark Office, 2016.
- [24] Tisue, S., Wilensky, U., Netlogo: A simple environment for modeling complexity, *International Conference on Complex Systems*, 21, May 2004, pp. 16-21.