Management and Production Engineering Review
Volume 5 • Number 2 • June 2014 • pp. 33–44
DOI: 10.2478/mper-2014-0015

DE G DE GRUYTER OPEN

mper

IMPULSE ARTIFICIAL NEURAL NETWORKS IN INTERNAL TRANSPORT

Joanna Ochelska-Mierzejewska

Lodz University of Technology, Institute of Information Technology, Poland

Corresponding author:
Joanna Ochelska-Mierzejewska
Lodz University of Technology
Institute of Information Technology
Wolczanska 215, 90-924 Lodz, Poland
phone: +48 (42) 632-97-57
e-mail: joanna.ochelska-mierzejewska@p.lodz.pl

Received: 15 January 2014
Accepted: 12 May 2014

Abstract

The second most important function of a warehouse, apart from the storing of goods, is internal transport with a focus on time-effectiveness. When there is a time gap between the production and export of products, the goods need to be stored until they are dispatched to the consumers. An important problem that concerns both large and small warehouses is the selection of priorities, that is handling the tasks in order of importance. Another problem is to identify the most efficient routes for forklift trucks to transport goods from a start-point to a desired destination and prevent the routes from overlapping. In automated warehouses, the transport of objects (the so called pallets of goods) is performed by machines controlled by a computer instead of a human operator. Thus, it is the computer, not the man, that makes the difficult decisions regarding parallel route planning, so that the materials are transported within the warehouse in near-optimal time. This paper presents a method for enhancing this process.

Keywords
internal transport, impulse artificial neural networks, dynamic path planning.

Introduction

There is no single standard definition of *intelligence*. Modern definitions of this concept refer to a variety of mental capabilities that characterize human beings, i.e. reasoning, learning, problem solving. In short, the term *intelligence* can be referred to as the ability to use knowledge. Artificial intelligence is a field that can completely change our world. It is clear that during the study of artificial intelligence a man is at the beginning of a long journey – trying to integrate a symbolic description of the mind and the brain at the level of the neural network. A lot of research has been carried out on the usage of artificial neural networks in various areas of human life. The purpose of those studies is to forecast activities which could not be logically justified, parallelize them in an appropriate time period and accelerate the process of decision making.

Today, there is hardly any sphere of social or economic life, in which the principles of logistics management are not applied. Technical progress contributes significantly to the development of logistics. IT revolution provides the means to monitor, coordinate and synchronize the flow of people, money and goods. However, due to the variations in demand and supply, it is difficult to establish a balance between them, which creates a need for a warehouse.

The need for storing arises due to the differences between the supply and demand. Warehousing provides protection for the stored supplies and enables a firm to keep track of their status. The need for storage arises also because some goods are demanded only during a particular season. Thus, overtime production during some seasons is unprofitable. On the other hand, low-yield production is not profitable in other seasons. It is important to ensure a reasonable distribution of production costs, while maintaining good quality of the produced material.

The most important function of storage, apart from holding the goods, is inner transport, with the emphasis on time-effectiveness. There might be a need for storing the goods so as to make them available to clients when required. Both large and small stores might have difficulty with the selection of priorities, that is sorting the tasks by order of importance. Another important problem concerns identifying the right routes for a truck to transport goods from one location to another. Of course, the routes should not overlap each other. In automated warehouses, the transport of objects (the so called pallets of goods) is performed by computer-controlled machines. The computer replaces a human agent in taking the difficult decisions regarding parallel route planning so that the materials are transported within the warehouse in near-optimal time.

The paper is organized as follows: section “Artificial Neural Networks” describes the artificial neural networks with a special focus on impulse neural networks. The next section, titled “The Idea of Dynamic Path Planning”, contains a description of the idea of dynamic path planning which is used in internal transport in the warehouse. Section “The Idea of Dynamic Path Planning” defines considered logistic problem from the viewpoint of the implemented system. This is followed by the presentation of a possible solution, description of the input data and warehouse structure. The section “Experiments and conclusions” presents two tests and discussion of the results.

Artificial neural networks

The basic constitutive element of a neural network model is an artificial neuron. The general model of an artificial neuron was developed by McCulloch and Pitts in 1943. It was based on the construction which simulated the basic functions of a nerve cell. An artificial neuron is considered as a signal converter which operates according to the principles of automatic control systems [1, 2].

Like a nerve cell, an artificial neuron receives n inputs $(x_1, x_2, ..., x_n)$ representing biological dendrites. It produces one output denoted by s, which corresponds to a biological axon. The input values are original data from the outside or already processed signals from other neurons’ outputs.

The output signal y depends on the values given to the inputs $(x_1, x_2, ..., x_n)$ which are assigned to the corresponding weights $(w_1, w_2, ..., w_n)$ and the activation function f. The sum of the products of input signals and their weights φ, which is the argument of the activation function is determined by the formula (1)

$$\varphi = \sum_{i=1}^{n} (x_i \cdot w_i), \tag{1}$$

where i is the number of the next input, n is the number of inputs to the neuron, xi is the value given for the i-th input, and w_i is the weight of the i-th input.

The parameters w_i, called synaptic weights, determine the importance of the value given for the i-th input x_i on the result returned at the output neuron y. After determining the φ value, the activation function f is calculated. The dependence can be described by the following formula:

$$y = f \cdot \varphi = f \cdot \sum_{i=1}^{n} (x_i \cdot w_i). \tag{2}$$

The activation may take the form of a linear, nonlinear or unit step function.

A single neuron can already be called a very simple neural network, but in order to take full advantage of what an artificial network offers, it is necessary to use a large number of neurons. This provides one with the ability to solve more complex problems. The capability of a single neuron to remember certain information is not very high. Thus, in order to increase the information capacity, neurons are combined in layers, and only a set of neuron layers can be regarded a complex neural network [3].

Impulse binary neuron

A modified version of the general model of an artificial neuron is referred to as an impulse neuron. It can be applied in impulse artificial networks. An impulse neuron is a more complex neuron, which communicates with other neurons by exchanging impulses [4, 5]. Its condition can be described as active or resting. The impulses are propagated in a discrete space. Thus, the i-th input x_i may store the values $\{0, 1\}$. A similar situation applies to the output. The values in this area can be considered as binary, where 1 is the active state and 0 represents the resting state. The impulses passing through a neuron are adequately processed and produce different potentials in the synapse [4, 5]. Such a neuron should know the concept of time to be able to remember its previous states of activation. Therefore, the output of a given neuron can function as an input for this neuron.

If we assume that the binary space is part of the discrete space, then a binary neuron can be regarded as an object devoid of weight, where the combiner and its activation function will refer to an algorithm which compares to the corresponding inputs of such a neuron. We can distinguish several types of inputs: the output from the neighboring neurons, outside inputs which inhibit the neuron, outside inputs which enhance the neuron, the output of the neuron treated as an inhibitory input which equips the neuron in a short-term memory. The scheme of the reasoning of an impulse neuron is shown in Fig. 1.

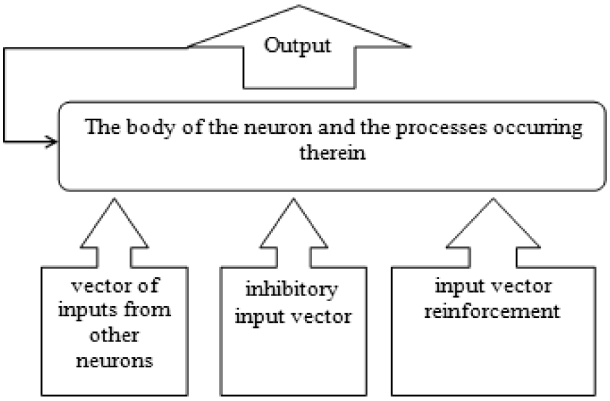
Output
The body of the neuron and the processes occurring therein
vector of inputs from other neurons
inhibitory input vector
input vector reinforcement

Fig. 1. The construction of an impulse neuron.

The functioning of an impulse neuron acting in the space $\{0, 1\}$ and using different types of inputs can be determined on the basis of the validity of the inputs. For example, if any inhibitory input receives a pulse, then a neuron becomes dormant automatically and sends information about its inactive state. Then, one can analyze the vector of strengthening inputs and, in the case of relevant information, reactivate the neuron from its dormant state and assign the output value 1. The next step is the interpretation of impulses transmitted from the neighboring neurons. While modeling this type of neuron, we can ignore the process of learning, because the neuron knows what to do thanks to the impulses it receives.

Impulse artificial neural networks

Analyzing the structure of the neural network and using binary impulse neurons, we are able to create a network, which will spread impulses thanks to the input data received from the outside. The biggest problem consists in proper adjustment of the network to the way those impulses are propagated through it, so as to receive the expected result. It is also important that the connections between the cells need not be limited only to the nearest neighboring cells. A crucial element in this type of network theory is the description of the neighborhood [4, 5]. Figure 2 shows a few examples of possible neighborhoods of neurons arranged in the plane of a varying parameter r which denotes the distance between them. The input data fed to the neural network layer is received simultaneously by all of its cells. As a result of the dynamic interactions between neurons, a certain state of the network is generated until the final determination of output values in all cells, or until the programmer implements a certain phenomenon occurring between specific cells. Although the connections between neighbors are local by nature, the changes occurring between them, namely the spread of the impulse effect, result in global changes across the network (the network layer) [4, 5]. Therefore, such a network enables parallel distributed information processing.

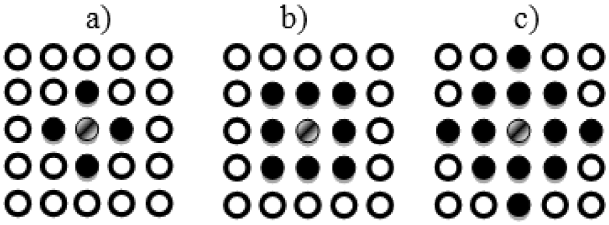

Fig. 2. Examples of neighborhood types for impulse neural networks a) $r = 1$, b) $r \leq \sqrt{2}$, c) $r \leq 2$.

Figure 3 helps to examine the structure of such a network. It illustrates a horizontal cut of a few selected neurons, with division into the entrances.

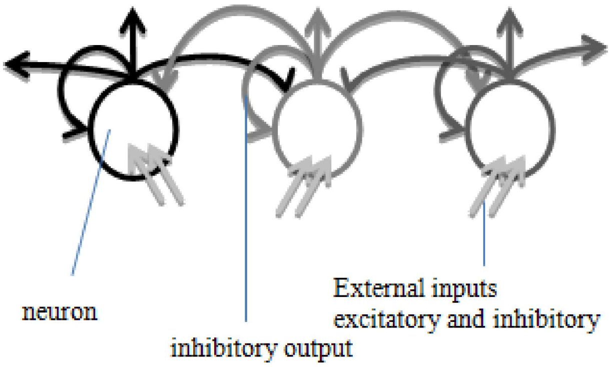

Fig. 3. A horizontal cut of an impulse neural network.

As shown in Fig. 3, the output of each neuron is connected to the inputs of neighboring neurons, with the distance between them denoted as r. It can also be connected to further layers of the impulse neural network. Such a structure of the impulse neural network is suited particularly for systems aimed at time-effective operation performance and achievement of real-time results. Thanks to the ability of parallel data processing, the network is capable of dealing with those difficult tasks. In the case of networks operating in a non-logical space, each cell manufactures the data in the same way, so it should have the same set of weighting factors. Thus, such networks are not subjected to the processes of learning and are analytically suited for a particular task. However, in the case of the binary space, the equivalent of the weighted inputs is the hierarchy of those inputs which is also determined analytically during the design.

The idea of dynamic path planning

The previously discussed impulse neural network can be applied to route planning within a certain location. Route planning is a practical problem which dates back to the time when humans started planning their first trips. Vehicle motion planning is a complex and difficult process, as it takes place in a changing environment and there is a need for parallel, quick decision making in a relatively short period of time. A solution to the above problems is provided by heuristics based on a neural network. It is an alternative to existing traditional algorithms. Heuristics can be understood as a method of an efficient algorithm that can find the best possible solution in near-optimal time. The traditional methods are often cumbersome and time-consuming, whereas the heuristic methods can drastically shorten the search for a solution [6, 7].

Before trying to plan the route for a vehicle, we have to identify its environment first. The environment may be static or dynamic (changing in a given interval of time). The present study deals with the dynamic type of environment, as it may turn out that a path planned for one vehicle has already been taken by another vehicle. Such a situation might result in a collision and congestion of the route. This paper endeavors to prevent such problems by introducing a novel path planning solution for the navigation of a forklift truck.

Definition of logistics task

The technical and technological revolution provides tools for the monitoring, coordination and synchronization of commodity flow. The example mentioned in the previous chapter can significantly optimize logistics processes by reducing the costs and time required to complete particular tasks.

The IT revolution provides a means to coordinate the supply and demand. This, in turn, creates the need to make supplies and store them in a separate building, i.e. a warehouse. The terms warehouse and storage are used interchangeably. The use of a warehouse is dictated by customer demand. The process of order fulfillment involves the flow of goods by various streams in an unsynchronized discontinuous manner. In order to avoid a downtime caused by the lack of certain commodities, it is necessary to store inventory.

Inventory can be defined as a physical quantity of goods expressed in quantitative units with a precisely defined location. Where there are business processes, there is inventory. It is an indispensable, constant element of a production process. Supplies are made at every phase of the company's activities and are subject to constant renewal through the inlet and outlet streams. We can distinguish a few types of inventories, namely those in the distribution networks, in the central warehouse, inventory of materials for assembly and manufacturing, which are maintained by the manufacturer or supplier. All of the above-mentioned inventory types have one common feature, namely they all need storing. The operations occurring in the warehouse are not always performed in an optimal way, which increases production costs. The productivity and efficiency of this logistics system (storage) in determined by proper design and layout of the building. An appropriate approach to this problem can help to:

- Increase productivity.
- Synchronize the inflow of products.
- Reduce the costs.
- Improve customer service.
- Provide a better working environment.

Effective warehouse planning requires following certain rules, some of which are listed below:

- Storage space requirements for all the products should be rationally calculated.
- If possible, a multi-level storage should be used. Using multi-level shelving provides the ability to keep more than one product within a given space unit.
- Passageways in the warehouse should be carefully designed. If a passageway is too narrow, it will restrict the flow of goods and the efficient use of equipment, and cause congestion. Too broad passageways, on the other hand, are considered as a waste of space which could be utilized for the purpose of storing.
- It is important to use the upper space of the warehouse for holding the goods that will be used less frequently. The lower rackspace should be used for the commodities that are planned to be used more often.
- Organization of landings and mezzanines for office space is necessary in order to maximize the storage area.
- The possibility to expand the storage space by following certain standards of storage, which are based on the ratio of utilized space measured in

cubic meters to the total usable area of the store given in square meters.

Depending on the solution chosen by a given company, it is necessary to maximize the available storage space and provide a system for material transport and reloading in a warehouse [8–10].

Internal transport in the warehouse

To maximize the efficiency of such operations as goods receipt, issuing and storing, the warehouse is divided into several parts – zones, where the import and export of goods takes place. We can distinguish several types of arrangement of these zones: transit layout, angled layout or sack layout.

Another important issue, apart from the organization of input/output zones where the transport of goods takes place, is the management of the storage area. The most common methods of storage area management are: row storage and storage in blocks.

Forklift trucks or automatic machines move along the planned routes between pallet racks. Warehouse Management System (WMS) generates tasks for workers or machines, which are represented as the start and end point on the path adjacent to the pallet rack where the object is located, or adjacent to the intermediate storage area. Then, the level of the item's location is generated. Regardless of the type of operation, whether it is the import, export or relocation of goods in a warehouse, the work is performed according to the same pattern:

- First, the task is to select a machine or employee that is currently inactive and located in close proximity to the point designated by the system.
- Next, the shortest possible and collision-free route is calculated and transmitted to the machine or worker responsible for completing the task. Meanwhile, the route might be re-planned on the basis of update information from a transmitter.
- Finally, after arriving at the designated place and picking up the item, the machine or agent receives information about the point where the item should be delivered, as well as the optimal route.

Most warehouses where goods are moved by non-automatic machines do not apply automatic route planning systems. The scheduling of individual forklift routes is left to a human agent. In larger storage facilities, this may cause significant prolongation of transport time, since a human operator is not always able to determine the fastest route and take into account all possible circumstances. For example, consider a situation when a narrow path in a warehouse is shared by two vehicles at the same time. When one of the workers is picking the goods from the rack, the other one has to wait for them to complete their task and only then can they move on.

The majority of warehouses apply software solutions that provide information about the current state of the stock and help to monitor the location of inventory items. However, the problem of dynamic scheduling of the most optimal route remains unresolved. One of the difficulties is definitely tracing the forklift's current location, but this can be solved by the use of a transmitter and a small-sized detection device. In order to determine an accurate position of a forklift in a three-dimensional space, it is enough to use three detection devices. Another problem concerns route scheduling in real time. This paper endeavors to find a solution to this problem.

Background of the problem

A typical method of planning a route in a defined environment is to highlight the points which are the vertices of paths and the parameters, such as the number of vehicles that may share the same path, maximum speed, level of congestion, etc. In the case of a warehouse, a vehicle is tasked to move between path vertices and get to the designated internal point of the road in order to pick or deliver goods. An example of the approach based on the vertices is shown in Fig. 4.

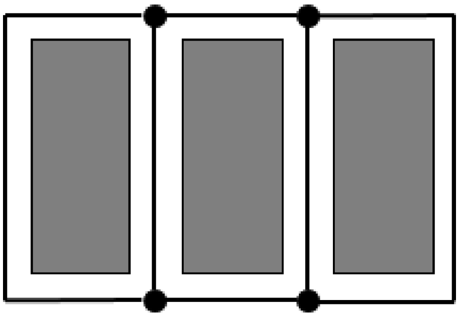

Fig. 4. A warehouse model in which the route is based on the vertices and the roads stretched between them.

The solution can be also expanded with additional vertices stretched on the road and located at the places where the vehicle carrying the goods may stop. Then, it is possible to establish a detailed route to the requested location. However, such an approach yields a huge number of vertices and even more paths. As we try to take into account all the parameters of the path, the computational complexity of the task increases. Another possible approach is to span a vertical-horizontal grid over the warehouse area (Fig. 5). Then, each point of the grid represents a possible location of a single vehicle or the stored goods.

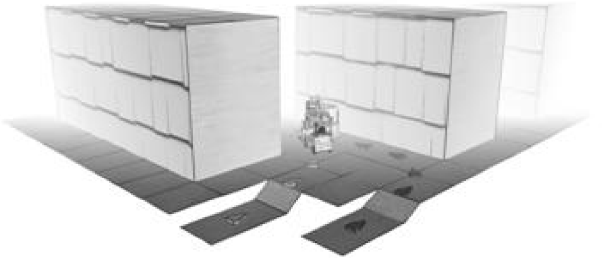

Fig. 5. Visualization of a storage based on a horizontal-vertical grid structure.

This approach ensures that no two vehicles share the same field at the same time, thus reducing the risk of a collision. Also, the roads in the warehouse run only horizontally or vertically, and the vehicle located at a point of such a grid is allowed to move in one of eight directions, see Fig. 6. Thus, each field in the warehouse is depicted as a square which can represent either part of a road or a rack where the goods are stored.

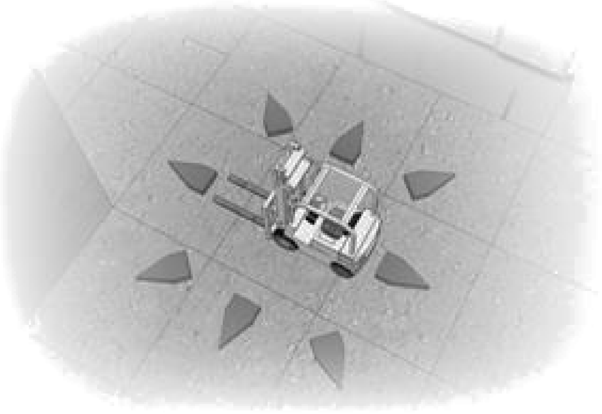

Fig. 6. Vehicle's mobility in a warehouse.

Methods of solution

A possible solution to the problem is the use of neural networks. The other possible methods include fuzzy logic (an expert system), ant colony systems or evolutionary algorithms. Below, there is a short description of the methods, with a particular focus on their advantages and disadvantages.

Fuzzy control is aimed to reflect the reasoning of a human being who is unwilling to deal with a problem mathematically. The human mind is capable of creating many different interpretations of a situation, while mathematics is limited to one particular outcome. The theory of fuzzy sets allows the formulation of a selected term in many ways, which helps in describing more complex systems. The application of fuzzy control also makes sense in situations where information is incomplete or inaccurate. On the basis of certain assumptions collected in the database, an inference machine generates the best possible solution. The fuzzy reasoning process is based on several rules for generating a logical sum. This process involves the use of resulting and decision-making blocks. The structure of the magazine should be kept in the database while using an expert system. The database also stores information about the possible decisions made by the system when generating a route from the starting point to the target. During application development, it is important to generate a knowledge base containing various possible circumstances, such as path blockade caused by another vehicle. In such a case, the expert system should determine another way, providing the ability to bypass the obstacle. There are many more instances similar to the above. Therefore, the establishing of such a system would not provide full integrity of the application and a potential expansion of the warehouse would entail the need for a lot of new changes in the knowledge base.

The method of genetic algorithms was invented to mimic the natural phenomena observed in genetics that lead to the evolution of species. Genetic algorithms are based on the mechanism of natural selection and inheritance, leading to the survival of the fittest individuals. A genetic algorithm starts with the formation of a random population of individuals which can be crossed with each other or subject to mutation. The next population is generated by selecting mostly the fittest individuals. This operation is repeated in a cyclic manner until we find an individual that meets the intended purpose. A fitness function is applied to assess the degree of adaptation of each individual and select the strongest individuals from the population for further reproduction.

The use of genetic algorithms consists in searching for the optimal solution by generating multiple random answers, checking the correctness of the results, making possible mutations or crossing one route with another. The operations described above might give the correct result, but an attempt to create multiple paths rather than just one path, increases significantly the computational complexity of the algorithm.

The ant colony approach is based on the natural behavior of insects – ants. It turns out that a colony of ants seeking a path between the food source and the nest is always able to find the optimal or near-optimal route. Each ant lays down a characteristic fragrance, called a pheromone, by which it communicates with other ants. If the route is blocked by an obstacle, the ants try to avoid it by taking different paths. When the first individuals reach the nest, they will return to the destination by the same path laying a stronger pheromone trail. More individuals on the route will intuitively choose just the same way until, after some time, all the ants from the colony follow the same, most optimal way. The pheromone evaporates over time and finally disappears as the source of food gets exhausted. Then, the ants no longer follow this route and begin the search for a new food source. Again, the first few ants begin reinforcing a new route with the pheromone trail, making it more appealing for the next ants.

The ant colony approach can be applied to route scheduling by creating a certain ant colony. Thus, the requested destination denotes the source of food and the starting point of a vehicle represents the ants nest. The ants circulate in a virtual warehouse until all of them start to follow the same route, which means that an optimal solution has been reached. The use of such an approach is usually time-consuming, as it takes some time until the ants fall on the right track and are joined by others. As the situation in a warehouse changes dynamically, there is a need to introduce a new colony of ants once in a while, so as to determine a new route in case of any obstacles, such as another vehicle on the route. In conclusion, ant colony systems give the most optimal results, although they require a certain time complexity.

Another solution to the task set in this paper is to use the previously described neural networks. They are susceptible to environment changes and can generate the desired solution in a very short time. The way of obtaining the result is not always optimal – not always close to the most optimal solution. The next sections of the article describe the use of neural networks for optimal route planning in a warehouse.

Solution with the use of impulse artificial neural networks

This section describes the use of impulse artificial neural networks to solve the problem of optimal route determination in the real-time environment of a warehouse.

Description of the problem

The purpose of the study is trajectory planning in a warehouse:

- A vehicle that is currently not in use is assigned the task of picking the items from the intermediate storage area or shelf.
- After loading the goods on a truck or an automated machine, the task is to transport it to a desired destination located directly near the shelf or the intermediate storage area.

The warehouse used for testing is based on a horizontal-vertical grid structure [11, 12], as shown in Fig. 4.

Each field of the storage structure belongs to one of the following types:

- The road on which the vehicle can move.
- Intermediate storage area from which the goods are imported to the store.
- Intermediate storage area into which the goods are imported from shelves.
- Multi-story bookcase in which goods are stored.
- Vehicle parking place.

The vehicles can move only along a pre-defined road, and only one vehicle at a time is allowed on a single field of the road. The vehicle can change its location to one of the eight neighboring fields or stay in the same field. The visualization of this assumption is shown in Fig. .

The next step is to apply a collision avoidance system. The system enables dynamic generation of a new collision-free route for a vehicle that encounters another entity on its path.

The tasks are generated and assigned to vehicles by an external system in a random way. The task is assigned to a vehicle which is currently idle. If all available vehicles are in use at a given moment, the task is deposited on the stack queue and waits for a vehicle that is currently inactive. The queue works according to the FIFO method – the task which was first placed in the queue is handled first.

Implementation of the method

The impulse artificial neutral network can be regarded as a good solution because of its parallel processing capabilities of calculation and communication between adjacent neurons of the component corresponding to the grid, and also the same field where each point of the impulse is implemented as a binary neuron. The structure of the grid is shown in Fig. 2. Each neuron in the final stage of the calculation has only two states – active or inactive. This is a key feature taken into consideration while analyzing the result of the implemented process. The neurons of the implemented network must have a temporary memory in order to remember their previous activity. Activation of a neuron depends solely on the incoming pulses, which do or do not induce the potential of the synapse. The inputs to the neuron are divided into external (stimulating and inhibitory) and internal (information about the status of neighbors). The implementation of an impulse artificial neural network algorithm requires two layers:

- the broadcast layer,
- the path layer (end result implementation layer).

The broadcast layer works by providing a stimulus to the end point of the planned route using an external stimulating input of a neuron. This initiates the formation of a wave seeking for possible paths of the vehicle. The neuron which is currently active, in this case the neuron corresponding to the destination, will send an impulse which will be received by the horizontally and vertically situated neighbors. It will also send the impulse to itself and receive it by an inhibitory input. In this case, the neuron, which has been active in the last few steps, is required to fall asleep to prevent the wavefront from stepping back. The neighboring neurons can continuously be bombarded with inhibitory impulses from the external input. When a neuron does not correspond to the field of a warehouse path, the pulse will cause a permanent weakening of the neuron. The neuron will not be inhibited by itself and will not receive impulses from the external inhibitory input. When one of the inputs receives a stimulus from its neighbors, it becomes energized itself and sends a signal to others. When a neuron, which in the previous step was excited, gets back a signal from its neighbors (the wave is trying to step back), it uses the temporary memory. This excludes neuron reactivation and makes the neuron inhibit itself on the basis of three previous states recorded in its memory. The construction of a neuron located in the broadcast layer and the organization of inputs is shown in Fig. 7.

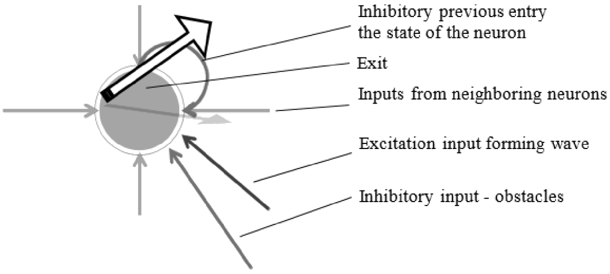
Inhibitory previous entry
the state of the neuron
Exit
Inputs from neighboring neurons
Excitation input forming wave
Inhibitory input - obstacles

Fig. 7. Structure of the input and the output of a neuron in the broadcast layer.

A single output of a neuron is cloned and the impulse reaches neurons with a corresponding location in the path layer. The steps of wave propagation in the broadcast layer are shown in Fig. 8.

The neurons in the path layer have inputs from their counterparts in the propagation layer, as well as external stimulating inputs. The signal from the external stimulating input is fed to the neuron corresponding to the current position of the vehicle and also relates to the neuron corresponding to the desired destination. In addition, these neurons have inputs which are characterize by eight neighboring neurons. The number of inputs determines the mobility possibilities of a vehicle, which is illustrated in Fig. 6.

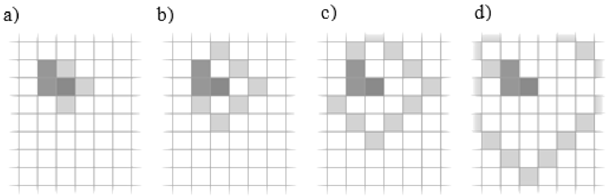

Fig. 8. Visualization of the stages of a broadcast wave propagation in a layer.

When two neighboring neurons in the path layer are active, one of them additionally receives an external impulse denoting the current position of a vehicle. Wave propagation in the broadcast layer ceases and the vehicle can move to a location corresponding to the position of a neighbor. At this point, a wave is sent to another vehicle. After all the vehicles have made a move, the operation is repeated in order to find the next possible position of the vehicle and, consequently, to reach the designated position. If the vehicle does not have a possibility to make any movement, it remains at its location, waiting until one of the roads is unblocked. A comparison of states of neurons in the path layer and the propagation layer is presented in Fig. 9. The application of an impulse neural network in steps allows the network to adapt to the rapidly changing conditions of the warehouse. While searching for the most optimal path, the vehicles located at various points of the route automatically restrict the maneuver possibilities of other vehicles. Then, the neurons in the propagation layer (except for the currently processed one) receive a pulse from the external inhibitor input informing about an obstacle. This might be also useful when a particular fragment of the route, is renovated and thus unavailable to any vehicle. Then, a pulse is sent, providing information about inaccessibility of this point and thus excluding it from the route.

Figure 9 presents selected steps of an impulse neural network operation. Situation *a* represents a selected step of wave propagation. Situation *b* presents a step preceding the finding of two adjacent neurons in the path layer, between which where the vehicle will move. An arrow marks a vehicle movement made in the next step. Situation *c* presents the moment when the vehicle changes its location, which is characterized by the excitation pulse transmission to another neuron. Situation *d* illustrates a step preceding the execution of the next move of a vehicle.

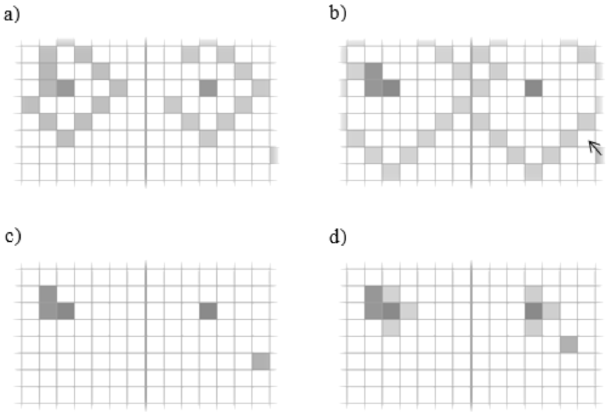

Fig. 9. Comparison of the states of neurons in several situations in the path layer and the propagation layer.

The above-mentioned steps are performed in a cyclic manner. However, where a vehicle completes its task, a suitable algorithm has to plan its return to the designated parking space so that it does not block the road, which can be used by another vehicle.

The neurons used both in the propagation and in the path layer are adapted to the dynamically changing environment and have a predetermined functionality. Thus, the neurons do not have to learn to determine their states, since they are pre-programmed. Another important aspect of using this type of neurons is the fact that the parameters stored by them and passed between them are binary values. This affects greatly the performance of the network, even in the case of a computer with average parameters, and provides a comparably faster work pace in a large stock. Also, in the case of expansion of the storage area, the network provides the ability to react to dynamic environment changes and thus allows trouble-free operation. Since the network operates in a step-based way, it takes into account a dynamically changing environment, allowing large numbers of vehicles to operate in a collision-free manner.

Experiments

The use of impulse neural network structure allows one to generate the results in real time. It also allows one to compare task completion times for different numbers of concurrent vehicles, and then translate the results into the costs related to the employment of workers, exploitation of transporters and their number.

The algorithm presented in this study is aimed to carry out research related to the simulation of a warehouse operation. The application allows one to load a warehouse structure, determine the number of vehicles (after a change in the number of vehicles, they return to their parking spaces) allowed to operate at the same time. It also allows random assignment of tasks:

1. A vehicle is tasked to pick up an item from the intermediate storage area and to allocate goods into the warehouse. Next, the task is to determine an optimal route from the start point to the intermediate storage area. When the vehicle arrives at the intermediate storage area, the application generates the location of a requested drop-off station and an optimal route to that point is dynamically planned.
2. A vehicle is tasked to pick up an item from a randomly selected field which denotes a rack. Then, the task is to plan a route from the start point to the requested drop-off point. After loading the goods, the requested drop-off point is generated, which is followed by dynamic generation of the route.

Experiments and conclusions

To the test the application, a list of 75 tasks was prepared (Table 1). A study was carried out in or-

Table 1
List of test tasks (type of tasks to do – coordinates (x, y) used in a specific field).

TO	20.9	FROM	20.8	TO	16.15
TO	13.9	TO	26.2	FROM	12.12
TO	8.7	TO	16.12	TO	14.7
FROM	2.7	FROM	26.17	TO	14.8
FROM	9.18	TO	26.9	FROM	11.8
TO	17.2	FROM	20.1	FROM	5.7
TO	26.8	TO	17.6	FROM	16.2
FROM	23.14	TO	11.7	FROM	22.5
FROM	17.7	TO	5.7	FROM	18.5
TO	8.7	TO	9.12	FROM	11.7
FROM	32.8	TO	15.12	FROM	14.12
TO	32.9	FROM	15.15	TO	20.17
TO	33.7	FROM	16.15	TO	23.1
FROM	33.2	TO	10.18	TO	26.1
TO	26.1	TO	20.2	FROM	29.17
FROM	17.15	FROM	23.8	TO	29.9
FROM	11.12	FROM	23.12	FROM	32.7
TO	11.7	TO	29.12	TO	33.16
FROM	11.2	FROM	33.15	FROM	17.15
FROM	2.7	TO	29.5	TO	11.8
FROM	2.8	FROM	29.2	TO	5.6
TO	8.15	FROM	10.2	FROM	2.9
FROM	11.18	TO	2.7	FROM	9.15
TO	14.7	FROM	8.8	TO	7.18
TO	8.6	FROM	5.6	TO	6.2

der to analyze the work of a warehouse, depending on the number of simultaneously moving carriages. The number of forklift trucks was a variable ranging from 1 to 11. The tasks are divided into two groups: TO – the task requires the forklift truck to move to a given field to take and transport the goods to the field belonging to the path of the given coordinates, FROM – the task requires transporting the product from a given field to the intermediate storage area.

The structure of the warehouse used for testing is shown in Fig. 10. Two tests were carried out:

- For the same set of tasks, but using a different number of moving vehicles.
- For dynamically generated tasks, using a maximal number of vehicles (the number of tasks to be performed was always larger than the number of vehicles) during the 10000 general steps of warehouse operation.

Test 1 was conducted for the same set of data, namely the coordinates of fields FROM/TO which the item had to be picked up/delivered using a varying number of resources, i.e., vehicles.

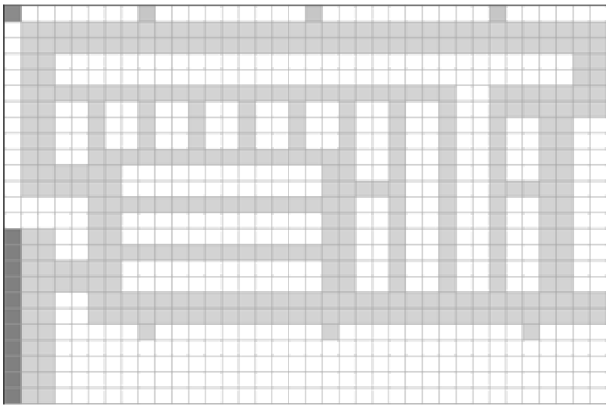

Fig. 10. The structure of the warehouse used for testing.

Figure 11 illustrates the dependence of the working time of the warehouse until the execution of all tasks. As shown in the chart, with the increase of the number of vehicles, we can observe an exponential working time dependence with a negative exponent. The higher the number of vehicles, the shorter the working time. However, this increases the maintenance and operating costs of the vehicles, as well as the employment costs of vehicle operators. This dependence can be described by the exponential function, the general form of which is presented below (3).

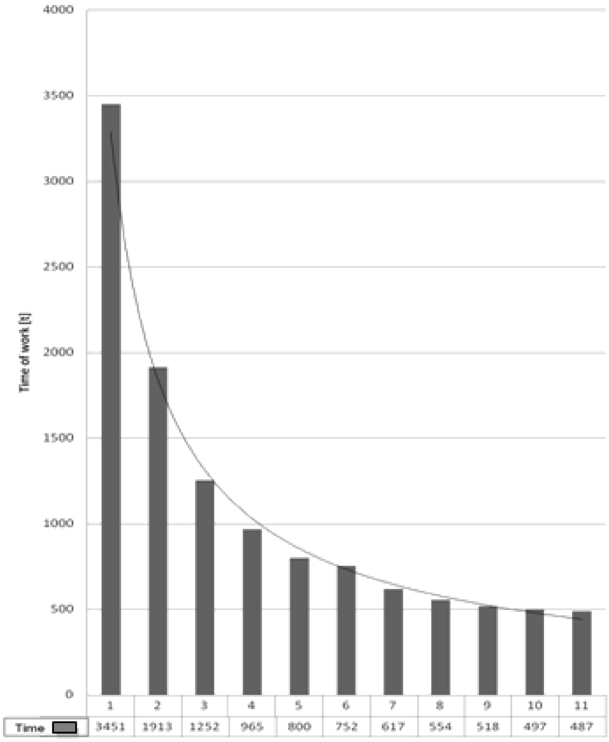

Fig. 11. Time magazine work depending on the number of working forklift trucks.

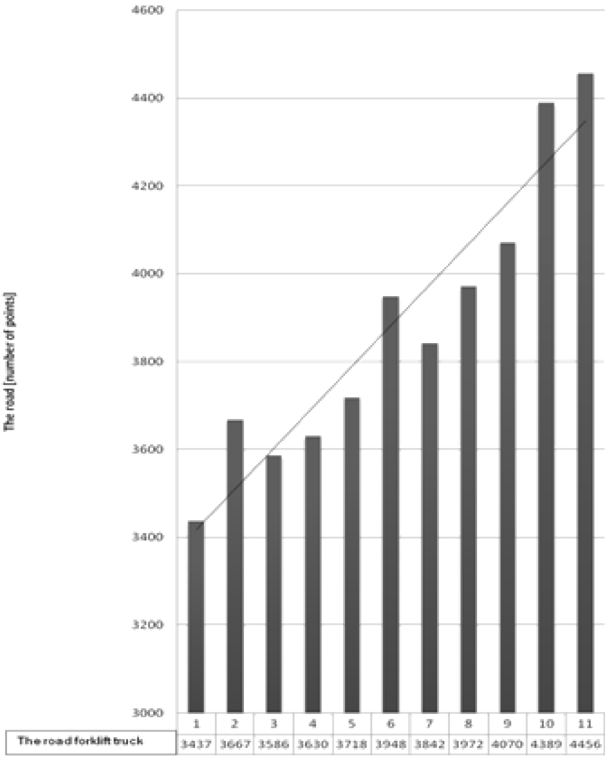

Fig. 12. Test 1: Total number of vehicle movements during the working, depending on the number of vehicles.

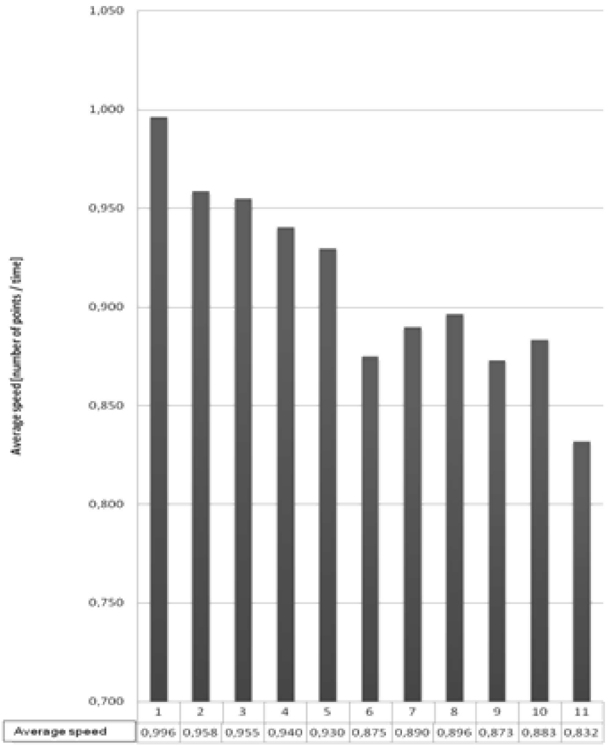

Fig. 13. Test 1: Average speed of the forklift as the ratio of the number of way-points to the total working time of the warehouse with the same set of tasks.

The function has been designated by the trend line in Microsoft Excel.

$$t = 3287 \cdot x^{-0.83}, \tag{3}$$

where t is given unit of time – a move by each vehicle, and x is the number of concurrent vehicles.

Thus, for the purpose of this experiment, we can determine the number of vehicles needed to complete the work in a timely manner. An increase in the number of vehicles results in an increase in the number of movements made by each vehicle (see Fig. 12. However, this relationship is not quite linear, as it affects road pass-ability, reducing the average speed (the number of vehicle movements in a unit of time) as illustrated in Fig. 13.

Test 2 was carried out using an identical warehouse structure, with a number of steps equal to 10000. As shown in Fig. 14, which illustrates the total number of tasks, the higher the number of parallelly working vehicles, the more tasks are completed. However, like in the first test, the number of tasks grows exponentially. Thus, by using a growing number of vehicles, we can increase the number of tasks completed. However, the rate of increase diminishes.

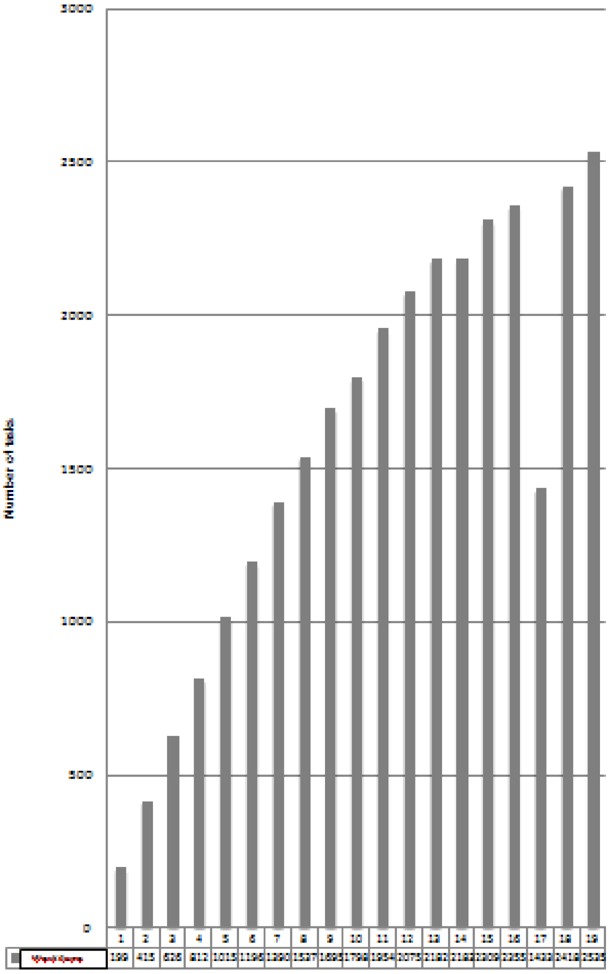

Fig. 14. Test 2: Total number of tasks performed by all vehicles used in extreme conditions, depending on the number of vehicles.

It is also worth mentioning that an increase in the number of vehicles entails a decrease in the number of tasks performed by a single vehicle, which in turn reduces the efficiency of each vehicle. However, the total number of tasks completed is growing – Fig. 15.

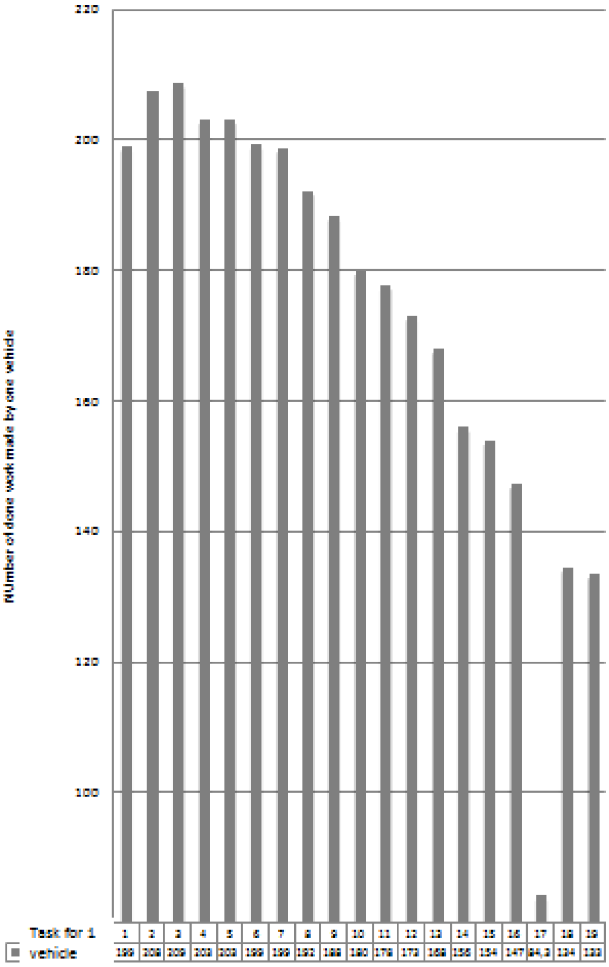

Fig. 15. Test 2: Average number of tasks performed by one vehicle in extreme conditions depending on the number of vehicles.

Summary

The aim of this paper was to present the possibility of using neural networks to generate real-time routes for vehicles in a dynamic environment of a warehouse. Impulse neural networks can in fact carry out activities related to the appointment of a route in optimal time. Thus, the method provides the ability to organize the work in the best possible way, regardless of the structure of a warehouse. The algorithms used for developing this application can be applied to the following purposes:

- To plan the construction of a warehouse structure, in order to optimize it by means of simulation.
- To plan the expansion or reconstruction of an existing storage in order to optimize the work.
- For automatic vehicles moving without human assistance, in order to determine a collision-free path between the respective fields of their trajectory
- For vehicles moving with the help of human operators equipped with a position transmitter, in order to determine a collision-free path.

References

[1] Harmon L.D., *Artificial Neuron*, Science, 129, 3354, 962–963, 1959.

[2] Hassoun M., *Fundamentals of Artificial Neural Networks*, Massachusetts Institute of Technology, 2003.

[3] Braspenning P.J., Thuijsman F., Weijters A.J.M.M., *Artificial Neural Networks: An Introduction to ANN Theory and Practice*, Lecture Notes in Computer Science, Springer, 1995.

[4] Ma Y., Zhan K., Wang Z., *Applications of Pulse-Coupled Neural Networks*, Springer, 2011.

[5] Lindbland T., Kinser J.M., *Image Processing using Pulse-Coupled Neural Networks: Applications in Python*, Springer, 1997.

[6] Miao H., Tian Y., *Dynamic robot path planning using an enhanced simulated annealing approach*, Applied Mathematics and Computation, 222, 420–437, 2013.

[7] Lee M., Lee Y., Rittenhouse R.G., *A genetic algorithm for PDA optimal path generation using GPS*, Applied Soft Computing, 12, 2379–2386, 2012.

[8] Roth U., Walker M., Hilman A., Klar H., *Dynamic Path Planning with Spiking Neutral Networks*, Lecture Notes in Computer Science, Computer Science Logic, 2007.

[9] Effati S., Jafarzadeh M., *Nonlinear neural networks for solving the shortest path problem*, Applied Mathematics and Computation, 189, 567–574, 2007.

[10] Yang S.X., Meng M., *An efficient neural network method for real-time motion planning with safety consideration*, Robotics and Autonomous Systems, 32, 115–128, 2000.

[11] Davoodi M., Abedin M., Banyassady B., Khanteimouri P., Mohades A., *An optimal algorithm for two robots path planning problem on the grid*, Robotics and Autonomous Systems, 61, 1406–1414, 2013.

[12] Chong J.W.S., Ong S.K., Nee A.Y.C., Youcef-Youmi K., *Robot programming using augmented reality: An interactive method for planning collision-free paths*, Robotics and Computer-Integrated Manufacturing, 25, 689–701, 2009.