DOI: 10.2195/lj_Proc_pagani_en_201811_01
URN: urn:nbn:de:0009-14-47433

A Neural Network-Based Algorithm with Genetic Training for a Combined Job and Energy Management for AGVs

Auftragsvergabe und Energieverwaltung basierend auf mit genetischen Algorithmen trainierten neuronalen Netzen für Fahrerlose Transportsysteme

M.Sc. Paolo Pagani
M.Sc Dominik Colling
Prof. Dr.-ing. Kai Furmans

Institute for Material Handling and Logistics (IFL)
Karlsruhe Institute of Technology

Automated guided vehicles are designed for internal material transport in production and warehouse environments. To do this, transport orders must be assigned to the vehicles. In addition, the vehicles often have an electric drive. The batteries required for this are discharged during operation. Therefore, it must be decided when the vehicles must go to a charging station. This control option is often ignored and the vehicles are only sent for loading when the battery has (almost) completely discharged. In this work, a procedure that simultaneously solves the assignment of jobs and the decision when a vehicle should drive to a charging station is presented and evaluated. It is based on neural networks trained by genetic algorithms. The evaluation shows that the presented method delivers better results than a method that combines the "First-Come-First-Served" and the "Nearest-Vehicle-First" methods and in which the charging processes are controlled by a fixed battery threshold.

[Keywords: Automated guided vehicles, AGV, job assignment, energy management, neural networks, genetic algorithms]

Fahrerlose Transportsysteme dienen dem innerbetrieblichen Materialtransport im Produktion- und Lagerumfeld. Dafür müssen den Fahrzeugen Transportaufträge zugeordnet werden. Außerdem haben die Fahrzeuge oft einen elektrischen Antrieb. Die dafür nötigen Akkubatterien werden im Betrieb entladen, sodass zusätzlich entschieden werden muss, wann die Fahrzeuge zu einer Ladestation fahren sollen. Diese Steuerungsmöglichkeit wird oft ignoriert, sodass die Fahrzeuge nur zum Laden geschickt werden, wenn sich die Batterie (fast) vollständig entladen hat. In dieser Veröffentlichung wird ein Verfahren vorgestellt und evaluiert, das die Auftragszuordnung sowie die Entscheidung, wann ein Fahrzeug zu einer Ladestation fahren soll, gleichzeitig löst und auf von genetischen Algorithmen trainierten neuronalen Netzen basiert. Die Versuche zeigen, dass das vorgestellte Verfahren bessere Ergebnisse liefert als ein Verfahren, das eine Kombination aus „First Come First Served"- und dem „Nearest Vehicle First"-Verfahren darstellt und bei dem Aufladevorgänge nur eingeleitet werden, wenn die Fahrzeugbatterie einen Grenzwert unterschreitet.

[Schlüsselwörter: Fahrerlose Transportsysteme, FTS, Auftragsvergabe, Energieverwaltung, neuronale Netze, genetische Algorithmen]

1 INTRODUCTION

Due to advances in laser scanner-based navigation and safety technology, automated guided vehicle systems (AGV) are becoming increasingly common in factories and warehouses. AGVs, such as KARIS PRO [Col16], can transport a wide variety of goods by using exchangeable modules (as Figure 1 shows). Today, the vehicles can drive freely and without additional infrastructure within the facilities. This significantly simplifies the installation and (re-)configuration of such systems. The increasing flexibility means that AGVs are more and more applicable to a wide range of scenarios.

Figure 1: Representation of possible applications of the vehicles of the project KARIS PRO in a production environment

However, the controls for transport job assignment are system-specific. This is why special strategies must be found for each system. These are usually heuristics that do not offer an optimal solution. Finding a suitable heuristic is difficult. A new approach is the use of neural networks to control the job assignment [Pag17].

On the other hand, many AGVs have built-in batteries which feed their control boards, the motors and the sensors. As a result, the batteries are discharged during operation. This is why at some point, the vehicles must go to a charging station to recharge the batteries. Once the desired battery level has been restored, they can take new transportation jobs. While charging vehicles can't be used for job processing and the throughput of the entire system will decrease. This is why a mechanism for controlling the charging cycles of the vehicles is required but usually ignored [LeA06].

A better job assignment policy and energy management can lead to a higher throughput, lower operating costs and/or a lower number of required vehicles. This is why we have developed a combined job and energy management policy which is based on neural networks which have been trained by genetic algorithms.

This research paper is organized as follows: In section 2 the standard job assignment and energy management methods are discussed. In section 3 we will explain how the system has been modelled. In section 4 we will describe how the neural network works and how it has been trained. In section 5 we will carry out a case study. Finally, we will describe the conclusions in section 6.

2 MANAGEMENT POLICIES

As mentioned in chapter 1, the management of AGVs has to tackle two main problems simultaneously, namely the job allocation and the energy management.

2.1 JOB ALLOCATION

As defined in [Pag17], the loading and unloading process of AGVs on the vehicles occurs at dedicated stations, which will be denoted in this paper as material sources and material destinations. The goal of AGVs is to pick up the material units from material sources and bring them to the correspondent material destinations. For each transportation, these two tasks are summarized in a job which has to be assigned to a vehicle before its execution. Whenever a job can be assigned to more than one idle vehicle or an idle vehicle can get more than one available job, a decision policy is required. A range of decision policies for this kind of dispatching problem can be found in [LeA06].

2.2 ENERGY MANAGEMENT

Although energy management has a great influence on the performance of an AGV system, it has so far been widely neglected in research.

Both [McH95] und [Ebb01] show the negative impact of a missing energy management by using a simulation experiment.

As defined in [McH95] two fundamental distinctions as to how the energy supply of the vehicles can be ensured:

- Changing the batteries (manually or automatically)
- Charging the batteries at stations or while driving

According to him there are three situations where an energy management can be omitted:

- There are sufficiently long breaks between shifts and the shifts are so short that the batteries are not completely discharged.
- The vehicles are so underutilized that they always have the possibility to charge energy without orders therefore not being able to be processed.
- The vehicles can be charged sufficiently with energy while driving.

As the vehicles we consider have a built-in battery and navigate freely (i.e. have no fixed routes), we will focus on the charging of batteries at dedicated stations. As the charging time is relatively long in comparison to the transport times, charging will only take place between to jobs. We also have no breaks, the vehicles are used constantly and the shifts are sufficient long so that all vehicles will have to go to a charging station regularly.

2.3 META-HEURISTIC MANAGEMENT POLICIES

In most recent literature, many meta-heuristic management strategies [Abd14] have been developed and applied in different scheduling problems like project scheduling or job shop scheduling. They have generally outperformed heuristics in most of the cases [Leu15] by finding good solutions with less computational effort [Blu03]. One of the most promising group of meta-heuristic algorithms are the neural network-based strategies, like for example [Fen03], [Foo88] and [Par00]. They are able to link the current system state to the decision to be taken with very complex relations. The description of this relations by heuristic strategies is very difficult. The decisions are identified by topology, the weights and thresholds of the neural network itself. As a result, they perform well in comparison to other heuristic and meta-heuristic methods, when the relations between the current system state and the best decision are complex and, sometimes, not intuitive. Moreover, those strategies can also be combined with other heuristics or meta-heuristics [Hai01].

In this paper, the combination between a neural network-based policy and a genetic algorithm, in a similar way to the one applied to the project scheduling in [Aga11], has been proposed and tested to design a neural network-based decision tool for the job assignment of AGVs

DOI: 10.2195/lj_Proc_pagani_en_201811_01
URN: urn:nbn:de:0009-14-47433

3 SYSTEM MODELLING

In this section, the modelling of the problem under investigation is explained. The model is composed by five main modelling objects: material units, material sources, material destinations, charging stations and vehicles. For instance, Figure 2 represents an example of system composed by two material sources, two material destinations, two charging stations and two vehicles.

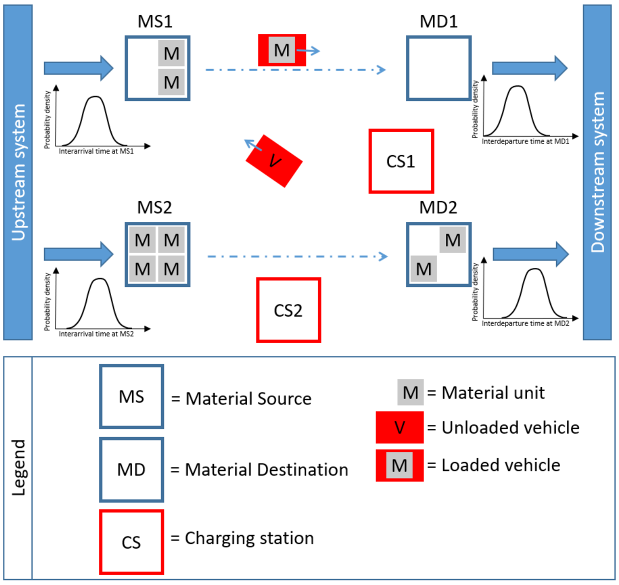

Figure 2: Example of system with two material sources (four buffer places each), two material destinations (four buffer places each), two charging stations and two vehicles.

The material units (in Figure 2 depicted as full squares and denoted by the letter "M") represent some group of goods, which can be identified by a common container or carrier, for example a box, a transportable shelf or a pallet. They are supplied from the upstream part of the logistic chain to the material sources. Once they lay on the material source, the AGVs are in charge to bring them to the right material destination. Once they arrive at the material destinations, they can be withdrawn to continue in the downstream part of the logistic chain.

The material sources (in Figure 2 depicted as empty squares and denoted by the letters "MS" followed by a number) are dedicated stations where the material units constantly arrive with an arrival process described by a Gaussian distributed interarrival time. Those stations are also characterized by a maximum number of buffer places and by (x,y) coordinates in a 2D layout. When a new material unit is about to arrive, and all buffer places are busy (see MS2 in Figure 2), it is assumed that this material unit does not enter the considered system. As a result, each material unit that cannot be delivered to a material source contributes to lower the total system throughput and thereby lowers the objective function of the model.

The material destinations (in Figure 2 depicted as empty squares and denoted by the letters "MD" followed by a number) are dedicated stations where the material units are constantly withdrawn with a Gaussian distributed interdeparture time. As for the material sources, those stations are also characterized by a maximum number of buffer places and by (x,y) coordinates in a 2D layout. When a material unit is required to be withdrawn at a material destination and no material unit are available in it (see MD1 inFigure 2), it is assumed that the demand is lost. As a result, each material unit that cannot be withdrawn from a material destination also contributes to lower the total system throughput.

The vehicles are the AGVs which the transportation jobs, i.e. the transportation of a material unit from one material source to one material destination, can be assigned to. Once they have completed a transportation job, they wait in an idle state at the material destination where they have delivered the last transported material unit. They represent the limiting resource, i.e. the higher their number, the higher the throughput, if the saturation has not been reached yet and if there are no blocking effects due to high traffic. They are characterized by the moving speed and a loading/unloading time, i.e. an additional time that is required to perform the loading/unloading process of the material unit on the vehicle, e.g. fine positioning, load transfer, etc.

The dispatching of the material is defined by a dispatching matrix DM, whose element $DM_{i,j}$ identifies the probability that a material unit arriving at the material source i must be delivered at the material destination j. As a result, it has a number of rows equal to the number of material sources and a number of columns equal to the number of material destinations. In the case depicted in Figure 2, where each material source is coupled to a material destination, the dispatching matrix is as follows:

$$\mathrm{DM} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

In addition to [Pag17], the charging stations (in Figure 2 depicted as empty squares and denoted by the letters "CS" followed by a number) are also included. They represent the location where the AGVs can recharge their batteries. Those stations are characterized by a maximum number of simultaneously loading vehicles and by (x,y) coordinates in a 2D layout.

4 STRUCTURE OF THE NEURAL NETWORK-BASED GENETIC ALGORITHM

In this section, it is explained how the neural network-based genetic algorithm works and how it can be trained without training samples by using a genetic algorithm.

DOI: 10.2195/lj_Proc_pagani_en_201811_01
URN: urn:nbn:de:0009-14-47433

4.1 Neural network as decision tool

Artificial neural networks (ANN) are inspired by biological neural networks. As stated in [DaS16], ANN consist of multiple computational components (artificial neurons). These neurons receive numeric signals as input and transform them (if an activation potential is exceeded) by a specific function into a numeric output signal.

Each ANN is organized in layers which are connected. There are three different types of layers.

- The input layer consists of neurons which receive a state vector (one input node for each element of the state vector) representing the current state of the system (e.g. $x_1, x_2, \ldots, x_n$, see Figure 3).
- The output layer consists of neurons which transmits a vector representing the output value. (e.g. $y_1, y_2, \ldots, y_m$, see Figure 3). Each output neuron is assigned to a possible decision.
- The neurons in the hidden layer receive weighted signals from neurons of the previous layer (input layer or previous hidden layer), transform them and transmit them to neurons of the next layer (next hidden layer or output layer).

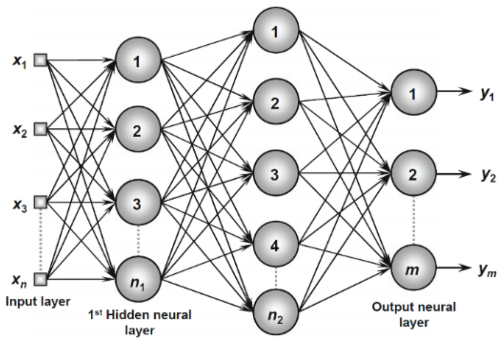

Figure 3: Example of multi-layer artificial neural network [DaS16]

An ANN has one input layer, one output layer and a variable number of hidden layers.

The ANN we use in our work performs as follows: Each time a decision is necessary (e.g. a vehicle has finished a job and gets idle or a new job is created) the current system state is transformed to a state vector and is given to the ANN. The neurons of the input layer transform the input signals to output signals and transmit them to the next layer where the signals are then again processed and transmitted to the next layer and so on. In the end the signals arrive at the neurons of the output layer where each neuron representing a possible decision provides a value. Based on this value the decisions are ranked and the decision with the highest rank which is currently possible (e.g. there are still idle AGVs or material units to be assigned) is executed.

In this work, the following parameters have been chosen as input information:

- Number of material units at each material source
- Number of material units at each material destination
- Number of idle vehicles at each material destination
- Number of loaded vehicles heading to each material destination
- Battery level of each vehicle

As a result, if we call N_{MS} the number of material sources, N_{MD} the number of material destinations, N_{CS} the number of charging stations and N_V the number of vehicles, the number input nodes (elements of input state vector) is equal to N_{MS} + 3* N_{MD} + N_V. For what concerns the number of output neurons, it is equal to the number of decisions. For this case, two types of decisions can be taken, i.e. each idle AGV can be either get a transportation job or can be sent to a charging station. As a result, the number of possible decisions, is equal to N_V*N_{MS}*N_{MD} + N_V*N_{CS}.

Figure 4 and Figure 5 show how the neural network is structured in the case depicted in Figure 2 and assuming one additional hidden layer with 11 neurons. In particular, in Figure 4 represents the chosen state vector and the structure of the input layer, while Figure 5 focuses on the output layer and how on the relation between output values and decisions are related. In Figure 4, the following abbreviations are used:

- "n_M in MS_i" denotes the number of material units in the i^{th} material source.
- "n_M in MD_j" denotes the number of material units in the j^{th} material destination.
- "n_{IV} in MD_j" denotes the number of idle vehicles waiting in the j^{th} material destination.
- "n_V to MD_j" denotes the number of loaded vehicles heading to the j^{th} material destination.
- "$\%_B$ of V_k" denotes the battery percentage of the k^{th} vehicle.

DOI: 10.2195/lj_Proc_pagani_en_201811_01
URN: urn:nbn:de:0009-14-47433

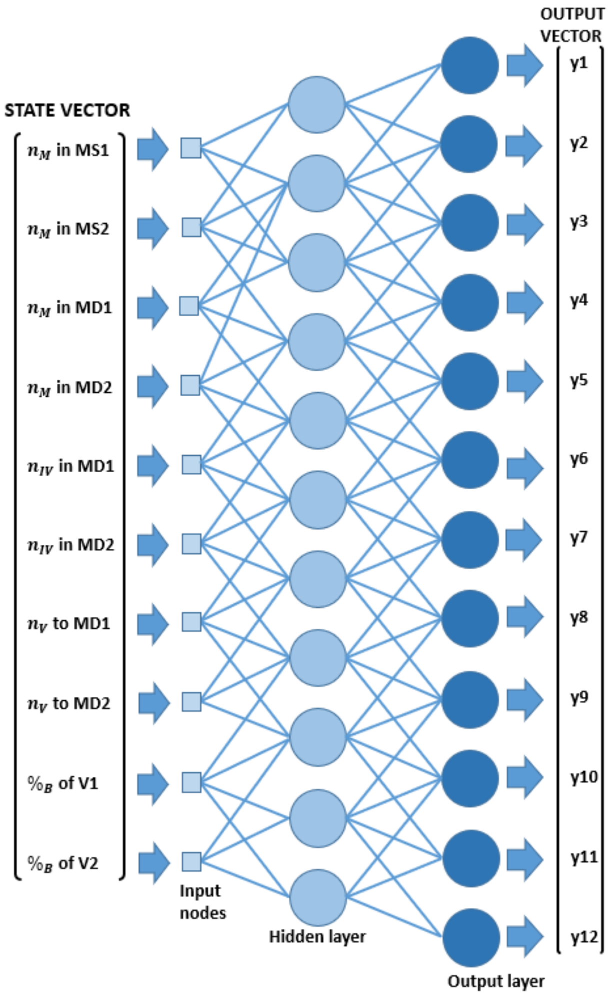

Figure 4: Input structure of the artificial neural network in the case represented in Figure 3.

In Figure 5, the values of the output vector are output values of the last layer of neurons, while the correspondent decisions are denoted either with the abbreviation "V_k transports from MS_i to MD_j" (the transportation job of a material unit, whose destination is the j^{th} material destination and which lays in the i^{th} material source, is assigned to the k^{th} vehicle) or with "V_k go to charging station CS_l", where V_k stands for the k^{th} vehicle.

The position and state of the material units and of the vehicles changes during operation and, as a result, the state vector changes as well. This is why the ANN may suggest a different decision for each different state of the system. If a decision cannot be executed (e.g. the suggested vehicle is already occupied) the decision with the second highest ranking will be chosen and so on.

In order to avoid that the vehicles go charging with high battery levels, it has been set that, if the battery level is greater than 50%, a vehicle is never sent to a charging station. Secondly, it must be assured that the AGVs with a too low battery level do not take any transportation job, but they are forced to charge the batteries. The battery level threshold, under which the vehicles are forced to go to a charging station has been set to 5%.

By using this rules, the ANN is able to compute a dynamic ranking, which depends from the current system state, for the possible decisions and can be used as a decision tool.

In order to evaluate the goodness of a decision tool, in this case the ANN, the logistical simulation approach is used. It means that the case study under investigation is reproduced in a discrete-event simulation environment. Similar to [Pag17], the system performances, in this case the

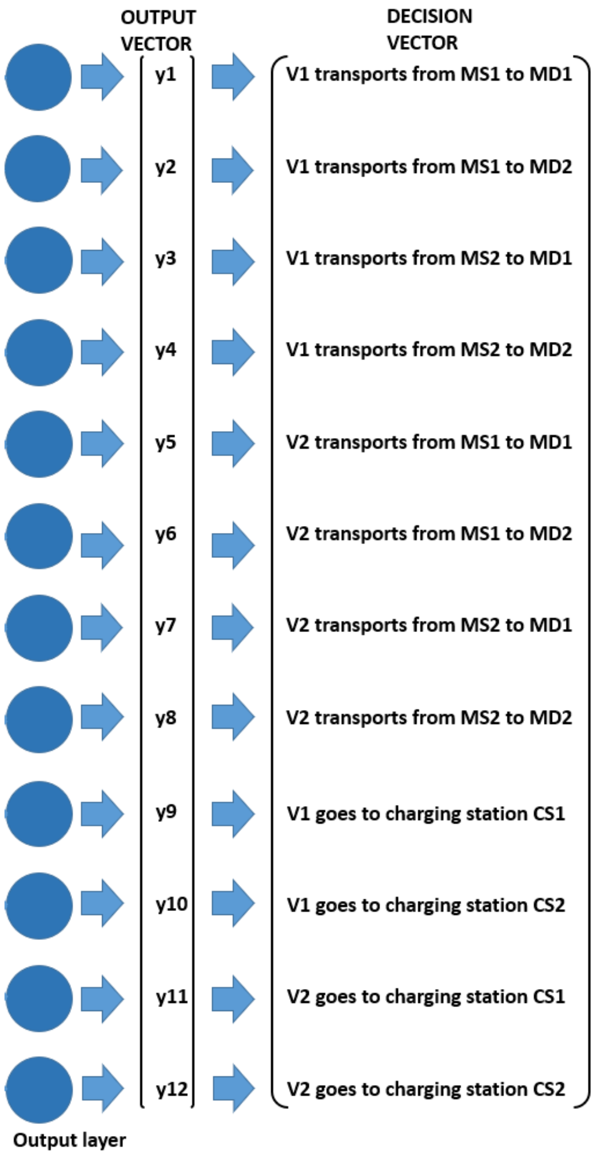

Figure 5: Output structure of the artificial neural network the case in Figure 3.

percentage of throughput denoted as $\%TH$ (see the formula below), applying the current decision tool are measured.

$$\%TH = \frac{TH_{out}}{TH_{out} + TH_{out,lost}}$$

TH_{out} = *Total number of material units that exited the system*

$TH_{out,lost}$ = *Total number of material units that could not be withdrawn from the material destinations (lost demand)*

4.2 Training with genetic algorithms

The output and thereby the performance of an ANN is highly dependent on the network topology (number of hidden layers and number of neurons in each layer), on the connection weights between the layer and on the activation potential and function of each neuron.

To find a good solution the artificial neural networks are usually trained with a large set of training samples, a set of samples where the right output values (in this case the right decision) for a given set of inputs (state vector) is known. One possibility is applying a set of known heuristic rules. However their goodness is sometimes limited by their simplicity. On the contrary, the purpose of this paper is to investigate the use of genetic algorithms.

The idea of genetic algorithms is to randomly generate a set of ANNs, test their goodness with a simulation tool and to proceed with the best subset of ANNs. The changeable parameters, like number of hidden layers, number of neurons on each hidden layer, connection weights, activation potential and activation function, of the ANNs of this subset, called parents, will be randomly changed. Therefore the parameters are combined and mutated so that new ANNs (children) are created. Parents and children then form a new set that will be evaluated and so on. As a result, it is possible to move in the solution space by generating new solutions. These steps are repeated until a target criteria is met.

The genetic algorithm for neural network-based decision tools proposed in this paper works as follows:

1. Initialize the algorithm parameters:
   a. Define a maximum number of generated neural networks, i.e. solutions, (computational constraint). This parameter is denoted as N_{max}.
   b. Define an initial population size of the solutions. This parameter is denoted as N_p.
   c. Define a number of generated combined solutions, i.e. a number of solutions that are created by combining random preexisting solutions at each iteration. This parameter is denoted as N_c.
   d. Define a number of generated mutated solutions, i.e. a number of solutions that are created by mutating random preexisting solutions at each iteration. This parameter is denoted as N_m.
   e. Define the topology of the neural networks (number of layers and number of neurons on each hidden layer). The topology is kept fixed for all the generated solutions. The number of input nodes and output neurons is based on the number of parameters of the input vector and to the number of possible decisions.
2. Generate the initial population of N_p solutions.
3. Add N_c combined solutions to the population. A combined solution is obtained by randomly choosing 2 parent solutions and by transferring some parameters of the neural network from one parent solution and some from the other.
4. Add N_m mutated solutions to the population. A mutated solution is obtained by randomly choosing one parent solution. Some parameters of the neural network are transferred from the parent solution, while others are randomly mutated.
5. Evaluate the goodness of the "N_p+N_c+N_m" individuals (neural networks) of the population as decision tool for the job assignment in the event-driven simulation environment.
6. Remove the worst performing neural networks and keep only the best N_p ones.
7. If the number of total tested solutions (initial + combined + mutated) is lower than N_{max}, repeat from step 3, else stop algorithm.

5 Case study

In this section, a case study representing the material supply of a car manufacturing plant is introduced. It is then used to test and to compare the benefit of a neuro-genetic algorithm in comparison to the commonly used heuristic policy "FCFS-nearest vehicle first" on a concrete example.

In the case study, the KARIS PRO vehicles [Col16] are responsible for the transportation of goods from the component supermarket to the assembly line.

5.1 Layout

In the layout presented in Figure 6, it is possible to see 6 stations in the supermarket area (marked with blue circles

DOI: 10.2195/lj_Proc_pagani_en_201811_01
URN: urn:nbn:de:0009-14-47433

numbered from 1 to 6), i.e. locations where the components from the supermarket are put on movable shelves (material and transportation unit) and get ready for the transportation, and 6 stations in the assembly line area (marked with blue circles numbered from 7 to 12), which are the locations where the movable shelves must be transported to. Moreover, 4 charging stations (CS) are available.

The distances between the relevant points of the layout are also given in Figure 6. The shortest segments without any length indication are negligible. The AGVs can only move on the dashed green lines.

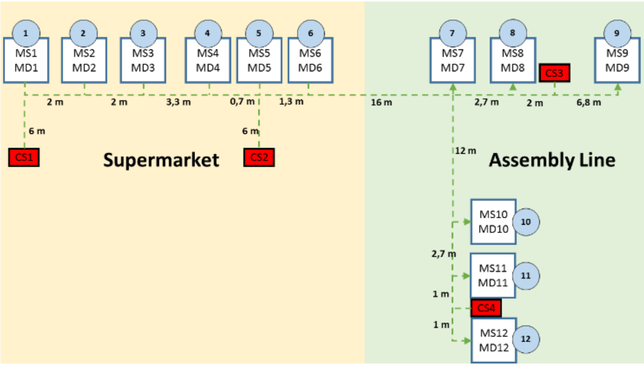

Figure 6: Layout of the use case

5.2 MODELLING

Since the assembly line is a timed process with a period of 27,7 minutes, one full movable shelf must be delivered at each station of the assembly line and one empty movable shelf must be returned at each station of the supermarket during each period. As a result, each station represents both a material source and a material destination. The interarrival time and interdeparture time are assumed to be normal distributed with expected value equal to the time period (27,7 minutes = 1662 seconds) and standard deviation equal to 415 seconds. Moreover, the material sources and destinations are pairwise coupled in such a way that the movable shelves are only transported back and forth from station 1 to 7, from 2 to 8, from 3 to 9, from 4 to 10, from 5 to 11 and from 6 to 12.

The system is served by four vehicles with a speed of 1 m/s and a loading/unloading time, i.e. an extra time required to autonomously load and unload the vehicles with the material units, of 30 seconds. The batteries are discharged with a rate of 0,75 $\frac{\%}{min}$ during the transportation and 0,19 $\frac{\%}{min}$ in all other cases.

For what concerns the genetic algorithm used to generate new neural networks, the following parameters have been assumed:

Table 1: list of parameters for the neuro-genetic training used for the case study

Parameter	Value
N_{max}	500
N_p	5
N_c	5
N_m	5
Mutation probability of the neural network parameters (weight and thresholds)	0,5
Expected value of new mutated parameters	0
Standard deviation of new mutated parameters	1
Distribution of new mutated parameters	Gaussian
Number of generated material units in the simulation to stop the evaluation	500

For what concerns the neural networks used as decision tool, a single hidden layer with 400 neurons has been considered.

5.3 COMPARISONS WITH STATE-OF-THE-ART STRATEGIES

In order to test the effectiveness of the neuro-genetic algorithm for the job assignment for AGVs, this algorithm has been tested against one of the most common assignment policies. This policy will be called “First-Come-First-Served and Nearest-Vehicle-First”, shortly “FCFS and NVF”. Applying this policy, the material units that are waiting for a longer time in the material sources get a higher priority. The transportation job of the material units with the highest priorities will be assigned to the nearest idle vehicle. This combined policy aims to transport the material with a FIFO policy and to minimize the paths of the AGVs at the same time. For what concerns the energy management, vehicle will only go to a charging station when their battery level is 5% or less.

As shown in Figure 7, the neural network-based policy, which has been trained with the neuro-genetic algorithm, outperforms the “FCFS and nearest vehicle” policy with a percentage of successfully provided material units at the material destinations of 80,66%, which is higher than the 72,51% of the “FCFS and Nearest Vehicle First” policy. The error bars in Figure 7 represent the 95% confidence interval of the data with 10 samples per policy and

DOI: 10.2195/lj_Proc_pagani_en_201811_01
URN: urn:nbn:de:0009-14-47433

shows that the results of the comparisons are statistically robust.

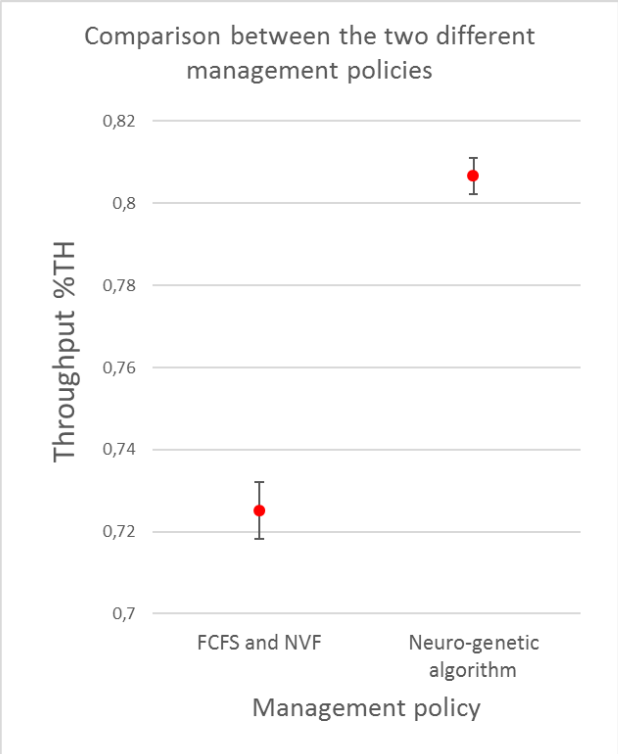

Figure 7: Results of the comparison between the two investigated assignment policies (neural network-based after the genetic training and "First Come First Served and Nearest Vehicle First").

It is important to underline that 80,66% of throughput is too low for a real application. In reality, more vehicles would have been used and also backup transportation, for instance manually operated by workers, would have been available. As we tried to simulate a fully utilized system, we only used four vehicles and excluded backup transports.

6 Conclusions

In this work, a neural network-based genetic algorithms have been introduced as a new methodology to design decision tools for the job assignment combined to the energy management for automated guided vehicles, shortly AGVs. In comparison to previous work in this direction (e.g. [Pag17]), the consideration of the energy management in the modelling provides the possibility to model the reality even better.

In particular, it has been shown how an artificial neural network can be used to link the set of states that a system can have and the set of state-dependent decisions that can be taken in operation. The training of the neural network is done with a genetic algorithm which generates new neuronal networks by combining and mutating the parameters of the best selected neural networks at each iteration. For what concerns the evaluation of the best solutions, an event-driven simulation is used.

The results have shown how this new methodology can quickly and effectively find a good solution, which outperforms another commonly used management policy for AGV, i.e. the "First-Come-First-Served and Nearest-Vehicle-First" policy, in the proposed case study representing the component supply to the assembly line in a car manufacturing plant.

In order to get an even deeper understanding of the use of the neuro-genetic algorithm, further research steps are required. For example, it is necessary to test them on a large set of systems with different layouts and vehicles and to test them with different parameters used for the combination and mutation of the solutions. Another important topic to be further investigated is the influence of a different topology of the artificial neural network, i.e. mainly number of hidden layers and number of neurons on each layer, on the training process.

Literature

[Abd14] M. Abdolshah, „A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions," *International Transaction Journal of Engineering, Management, Applied Sciences and Technologies,* pp. 50-84, 2014.

[Aga11] Anurag Agarwal, Selcuk Colak, Selcuk Erenguc, „A Neurogenetic approach for the resource-constrained project scheduling problem," *Computers & Operations Research,* Bd. 38, Nr. 1, pp. 44-50, 2011.

[Blu03] Blum, Christian and Roli, Andrea, „Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR),* Bd. 35, Nr. 3, pp. 268-300, 2003.

[Col16] Colling D, Ibrahimpasic S, Trenkle A, Furmans K, „Dezentrale Auftragserzeugung und -vergabe für FTF," *Logistics Journal: Proceedings,* Bd. Vol 2016, 2016.

[DaS16] Da Silva, Ivan Nunes and Spatti, Danilo Hernane and Flauzino, Rogerio Andrade and Liboni, Luisa Helena Bartocci and dos Reis Alves, Silas Franco, Artificial Neural Networks: A Practical Course, Springer, 2016.

[Ebb01] Ebben, Mark. „Logistic control in automated transportation networks”. PhD diss., Twente University Enschede, 2001.

[Fen03] S. Feng, „Using MLP networks to design a production scheduling system,“ *Computers and Operations Research,* pp. 821-832, 2003.

[Foo88] Foo, Simon, Yoon-Pin & Takefuji, Yoshiyasu, „Stochastic neural networks for solving job-shop scheduling. I-Problem representation,“ in *IEEE Int Conf on Neural Networks*, 1988, pp. 283-290.

[Hai01] Haibin Yu, Wei Liang, „Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling,“ *Computers & Industrial Engineering,* Bd. 39, Nr. 3, pp. 337-356, 2001.

[LeA06] Le-Anh, Tuan and De Koster, MBM, „A review of design and control of automated guided vehicle systems,“ *European Journal of Operational Research,* Bd. 171, Nr. 1, pp. 1-23, 2006.

[Leu15] R. Leus, S. Rostami and S. Creemers, „New benchmark results for the stochastic resource-constrained project scheduling problem,“ *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),* pp. 204-208, 2015.

[McH95] McHaney, Roger. “Modelling battery constraints in discrete event automated guided vehicle simulations.” *International Journal of Production Research,* Bd. 33, Nr. 11: pp. 3023–40, 1995.

[Pag17] Pagani, Paolo, Dominik Colling, and Kai Furmans. "Neural Network-Based Genetic Job Assignment for Automated Guided Vehicles." *Logistics Journal: Proceedings* 2017.10 (2017).

[Par00] Park, Youngshin and Kim, Sooyoung and Lee, Young-Hoon, „Scheduling jobs on parallel machines applying neural network and heuristic rules,“ *Computers and Industrial Engineering,* pp. 189-202, 2000.

M.Sc. Paolo Pagani is working as a Research Assistant at the chair of Robotics and Interactive Systems, Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT).
E-Mail: paolo.pagani@kit.edu

M.Sc. Dominik Colling is working as a Research Assistant at the chair of Robotics and Interactive Systems, Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT).
E-Mail: colling@kit.edu

Prof. Dr.-Ing. Kai Furmans is Head of the Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT).
E-Mail: kai.furmans@kit.edu

Address: Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT), Gotthard-Franz-Str. 8, 76131 Karlsruhe, Germany