The Multi Crane Scheduling Problem: A Comparison Between Genetic Algorithm and Neural Network Approaches based on Simulation Modeling

(Article begins on next page)

26 December 2024

The Multi Crane Scheduling Problem: A Comparison Between Genetic Algorithm and Neural Network Approaches based on Simulation Modeling

Naomie Bartoli, Roberto Revetria, Emanuele Morra
Department of Mechanical Engineering (D.I.M.E.)
University of Genoa, UNIGE
Genoa, Italy
e-mail: nani.bartoli@live.it, roberto.revetria@unige.it,
emanuele.morra@edu.unige.it

Gabriele Galli, Edward Williams
Department of Industrial and Manufacturing System
Engineering (I.M.S.E.)
University of Michigan - Dearborn
Dearborn, USA
e-mail: {ggalli, williame}@umich.edu

Maurizio Schenone
Department of Management and Production Engineering (D.I.G.E.P.)
Polytechnic of Turin, PoliTO
Turin, Italy
e-mail: maurizio.schenone@polito.it

Abstract— **The internal logistics for warehouses of many industrial applications, based on the movement of heavy goods, is commonly solved by the installment of a multi-crane system. The job scheduling of a multi-crane system is an interesting problem of optimization, solved in many ways in the past. This paper describes a comparison between the optimization by the use of Genetic Algorithms (GA) and introduce a framework for the solution of the problem using machine learning driven by Neural Networks (NN). Even though this last approach is not implemented in this paper, performances very close to GA ones are expected with NN. A case-study for steel coil production is proposed as a test frame for two different simulation software tools, one based on a heuristic solution and one on machine learning; performances and data achieved from reviews and simulations are compared.**

Keywords - Crane; Genetic Algorithm; Neural Network; Simulation.

I. Introduction

To move materials, particularly heavy or bulky ones, in warehouses, bridge cranes are often used. These bridge cranes frequently share runway girders. The presence of more than one overhead bridge crane sharing the same runway girders, virtually over the same served area, represents an enormous advantage in terms of efficiency for the operations, but also markedly increases the complexity of job scheduling, with relevant consequences for plant and workers' safety. The digitalization of production processes opens numerous research themes with the ultimate aim of reaching a higher level of repeatability and efficiency of the plant. In the context of the whole manufacturing factor, the bridge crane is a fundamental component of every plant, as it must guarantee the movement of the processed objects or its components from one workstation to another, minimizing delays and other inefficiencies. To date, the management of overhead traveling cranes is still done manually for many reasons, mainly related to the safety and extreme conditions of these systems such as temperature, dust, and the handling of generally heavy or unwieldy objects. Nonetheless, the latest developments in the field of sensors and the implementation of interventions today invite exploration of completely automatic management of the movement of the bridge cranes, in particular applied in the park area. This area is organized in large stacks containing different types of scrap and is dominated by a bay on which at least two bridge cranes move; their exact number depends on the size and complexity of the system. Their hooks (mechanical or magnetic) enable picking up and depositing the material.

In general, the scheduling of overhead crane movements is an optimization problem, given the data of a list of jobs (withdrawal-deposit); the goal is to find the best sequence with the assignment of single jobs to complete all the jobs in the shortest possible time. This theme is particularly interesting in the scrap stock area for coil steel production plant, where, unlike the melt shop, at a given time it is typical to have very long job lists to get a basket, so dozens of missions are needed for one goal.

Crane transport is the most important logistical transport method in the production of steel; the cranes must catch, hoist, and deliver scrap loads. The planning of the bridge crane has become the key to any production procedure and represents a decisive factor to ensure the production process operates with stability, reliability, and high efficiency in the workshop.

The programming problems of metallurgical cranes are mainly focused on how to solve the spatial conflict between cranes. Research methods based on mathematical programming methods, heuristic methods, simulation methods and intelligent hybrid programming algorithms have been used. It is, therefore, necessary to study a method that could simultaneously and effectively eliminate the spatial and temporal conflict between cranes [1]. Finding

optimal solutions to planning problems is generally a difficult proposition, entailing combinatorial optimization and often being strongly nondeterministic polynomial (NP). The problem of programming cranes is no exception to this: it is an NP-hard problem. Therefore, simulation is a suitable tool for studying complex materials management systems.

In this paper, a comparison between two different approaches is proposed to be analyzed in parallel on a test frame offered by a simulation model: the first is given by the application of a wizard for optimization based on Genetic Algorithms and the second is the application of machine learning for piloting the multi crane by a Neural Network. The comparison is currently still on a qualitative level and this paper proposes a test frame for further performance comparisons. Quantitative results from the Genetic Algorithm solution are shown, while hypotheses on the Neural Network approach results are described, taken from general evaluation and results by commercial solutions. The next development of this paper will be focused on the set up of a model based on the same test frame, but in which multiple cranes are moved by an artificial intelligence (AI) application. Hence a quantitative comparison between both will be available in future work.

The paper is organized as follows. First, a literature review about the cranes problem is presented, followed by a description of the problem (both a general description and the scheduling problem). Second, the methodology used (i.e. simulation, GA solution, and NN solution – the last one to be implemented later in a future follow-up paper) is explained. Third, an evaluation analysis of the paper is introduced. Lastly, a conclusion is provided.

II. STATE OF THE ART

Due to the growing need to accelerate warehouse handling operations, many recent studies have focused on programming multiple cranes operating simultaneously within the same area (typically understood to mean "bay;" i.e., the cranes share rails). A literature review was carried out according to the following stages:

1. identification of keywords and their combinations;
2. selection of a source database;
3. analysis of the results.

Regarding phase one, the following words were chosen: simulation, optimization problem, objective function minimization, bridge crane scheduling, genetic algorithms. The research was applied to the article title, abstract and keywords. In research phase two, Scopus and Google Scholar have been selected as source databases. At the end of this phase, about twenty articles were collected. During phase three, a new systematic analysis of the basic specifications was undertaken and specific papers in the references have been selected based on their content and availability.

The topic was already covered in the early 2000*s,* during which proposals arose to study the programming of cranes in a port environment with spatial constraints with the goal of finding the crane-job combination that maximizes productivity. Zäpfel et al. [2] presented new ideas with the aim of increasing the efficiency of the steel supply chain, by improving warehouse management, concerning the handling of different types of steel coils with the help of overhead cranes subject to time constraints. This study aimed to minimize processing time, the last order completion time during the planning period and labor costs. To do this, due to the complexity of the study, a mixed-integer mathematical-nonlinear optimization model was created. Furthermore, the problem was formulated in terms of job shops to represent all the dynamic situations of the warehouse processes (logistics center). Graunke et al. [3] studied the sequencing problem that was based on a multi-crane programming problem in a coils warehouse. Indeed, for problems on a larger scale, the heuristic algorithm generates quick, robust and acceptable solutions. The genetic algorithms were also used for the optimization of the integrated programming of overhead cranes and storage platforms at the automated container terminals. This article introduced a genetic algorithm capable of performing integrated programming of quay cranes, automated guided vehicles and handling platforms; in particular, this work is oriented towards container terminal applications. In fact, this study aimed to optimize the container stacking process, which is the motivation to save space. The results obtained with the genetic algorithm method have shown higher performance when the programming horizon increases. The programming of the movements of the bridge cranes over time has therefore been studied with different methods and approaches; only some of the methods used in different work environments will be reported below.

The use of machine learning and AI in multi-crane driving is as yet a task unconsolidated in the literature, although early studies and commercial solutions based on model simulation are appearing on the market. On the contrary, job scheduling problems in manufacturing have been studied via machine learning for many years. The works by Jain et al. [4] and by Weckman et al. [5] are expressions of this and specific system architectures in this direction have been evaluated. Results emphasize that the solution through a Neural Network scheduler provides performances very close to a Genetic Algorithm scheduler, with the well-known capabilities of AI to learn and adapt to variations in initial conditions.

Xie et al. [6] considers a multi-crane programming problem commonly encountered in real warehouse operations in steel companies' warehouses of steel coils. A certain set of coils must be recovered from the designated places: if a requested coil is in the upper or lower level without being blocked, it can be taken directly to its designated place; otherwise, the blocking coils must first be moved to another position. The study is to consider at the same time the problem of mixing and the problem of programming the overhead crane since the simultaneous approach could accurately take into account the routing and the time of movement of the crane; therefore an improvement in the utilization of the crane could also

improve the completion time of the operations. To clearly describe the problem studied, it was first formulated as a mixed Model of Integer Linear Programming (MILP); after which some feasible and optimal properties were identified for the assignment of the bridge cranes to avoid collisions. Since a particular case of the problem has proved to be strongly NP-hard, a heuristic algorithm has also been proposed to solve the aforementioned problem efficiently. In the heuristic algorithm program, for each crane, a detailed sequence specifies the order of all the necessary coils and the position of each block coil to be repositioned. To evaluate the performance of the heuristic, three lower limits are obtained according to the different cases. Furthermore, the strategy of the composite constraint of the lower limits has been adopted to converge toward an optimal value as they are mutually complementary and no one lower limit can dominate the others. Based on the tighter lower limit, the worst performance limit of the proposed heuristic algorithm is analyzed. The results showed that for small-scale problems, MILP can achieve slightly better high-quality solutions than heuristic solutions. For problems on a larger scale, the proposed heuristic algorithm can quickly generate robust and acceptable solutions.

In order to solve the NP-hard problem of overhead crane programming, the work by Ma et al. [1] presents a method of simulating crane programming in the steelworks workshop based on the Multi-Agent System (MAS). The scheduling system mainly includes groups of agents for overhead cranes, groups of agents for workstations and agents for managing coordination. MAS can decompose a complex and large system into many simple agents that have a simple structure, can interact with each other and can be easily managed. MAS introduces the global objective to match the programming strategies and operating results of the overhead crane with the global objective and the production practice in the workshop. Based on the analysis of the production process, on the methods of organizing production and on the characteristics of the current programming of the cranes in the steelworks workshop, the MAS applied in the operation of the steel bridge cranes is feasible because:

1. The MAS uses the "bottom-up" design method, which agrees with the rule of formation of the overhead crane programming system;
2. The MAS corresponding to the structural characteristics of the different levels is in accordance with the distributed structure of the bridge crane programming.
3. The autonomy and adaptability of MAS agree with the intelligence of each unit in the overhead crane programming system;
4. MAS can combine with various types of advanced information technology, artificial intelligence technology and so on, which then complement each other.

Each agent has its corresponding rules. The research shows that the simulation method based on the MAS is feasible and can meet the demand of the bridge crane in a timely and effective way. Furthermore, it can reduce the average transit time of materials and also describe the characteristics of the overhead cranes in the steel mill efficiently [1].

The document by Lim at al. [7] examines the programming of overhead cranes for ports through the Tabu Search (TS) method, a search procedure that proceeds iteratively from one solution to another by moving to a "neighborhood space" with the help of adaptive memory. Probabilistic Search for Tabu (PTS) is a variant of the basic TS, which places more emphasis on randomization than the basic TS. TS operates through neighborhood shifts, which proceed from one solution to another at each iteration. In the basic TS method, some shifts are taboo-marked and prohibited, unless they lead to highly desirable results. Rather than using deterministic strategies, PTS can be used to select movements based on the status and ratings assigned to them by the basic TS principles. The basic approach is to create move assessments that include references to the taboo status and other relevant prejudices of TS strategies using penalties, modifying the decision criterion and selecting the next move among the neighborhood moves that are based on values of different evaluation. From the experiments, the comparison and the above analysis, it is, therefore, believed that the Tabu Probabilistic Research is a good heuristic method to find good solutions to the programming problem of the bridge cranes.

The paper by Yuan et al. [8] proposes an efficient network flow model based on events and time spaces with lateral constraints for the problem of programming the overhead crane in a coil warehouse when the overhead crane must undertake a series of requests for storage, recovery and mixing of the reels, and determine the sequence of management of these requests as well as the positions to which the reels are moved. The goal is to minimize the makespan (the time needed to execute *all* the requests); in fact, the problem of programming the overhead crane is optimized by minimizing the actual working time to avoid unnecessary movements. Not only optimization for the handling time of each request, but also for the space in which for storing the reels is required; so, a new space-time network flow model is required, that uses the strategy of continuous time modelling in a space-time network, which can be optimally solved by a conventional Mixed Integer Program (MIP) solver for small cases and medium size in one minute. Due to the NP-hard characteristic of the problem, a dynamic programming approach has been designed exploiting the structure of the problem, on the basis of which an Approximate Dynamic Programming algorithm (ADP) is developed to obtain almost optimal solutions. The results, through computational experiments, showed that the proposed model for space-time network flow can be solved much faster than the traditional planning model and the standard model for space-time network flow.

Xie et al. [9] solve the programming problem of a bridge crane as presented in the warehouse of cold rolling material in a steel company. The goal, as before, is to minimize the makespan. Since the problem is NP-hard, a genetic algorithm has been presented to offer a reasonable quality of the solution. The exploration process is performed by two genetic operators: the crossover and the mutation. GA starts with a population of individuals and each solution is called a chromosome, which is made up of genes that represent decision variables. The chromosome representation used in this work shows that each coil in the program is a chromosome gene. Ten instances have been resolved to test the performance of the proposed GA. For most smaller cases, GA can achieve an optimal solution. As the problem size or space usage increases, the GA solution time is longer. The performance of this algorithm has been demonstrated efficiently and effectively. It provides a solution to improve the efficiency of the bridge cranes and the productivity of the steel company in the warehouse. In the work proposed, simulation has been used together with GA in order to explore a sub-system of solution and to reduce the computational time accordingly. This is a crucial point since three cranes are assumed; hence, the computational complexity is markedly higher than that of reference.

III. MATERIALS AND METHODS

A. General Description

The steel plant is divided into several bays and, in this elaboration, we have focused on the analysis of the scrap park which represents the area where the material is stored awaiting loading into the kilns. The shed analyzed is characterized by several stalls (sheet pantograph, turning, pre-reduced, naval, etc.) and each of these represents a possible starting point for the scrap. In addition, there are three preparation points where the scrap baskets are then prepared.

To move, after loading the scrap in the various stalls, and unloading it subsequently in the baskets, there are three overhead traveling cranes that travel on a single track. The real dimensions for the shed are the length of runway (213m) and the respective width of runway (63m).

B. Scheduling Problem

In the scrap park, overhead cranes must manage mainly:

● The supply of scrap, moving it from trucks/flights to stacks in dedicated bays,
● Loading the scrap, moving it from the stacks to the baskets (or conveyor belts) which then deliver the scrap to the melting furnace, following a suitable and predetermined succession by type.

Three overhead cranes share the same runways and are engaged by the simulation scheduling system. The task of the system is to provide scrap materials to specific baskets, starting from different sources according to a specific recipe, needed for the final product (steel type) to be produced. The overall process flowchart is shown in Figure 1.

C. Simulation Model

As previously described, the plant studied is a scrap park and is characterized by several areas that represent the possible starting points for the scrap handling. The transport of the scrap from the different areas to the baskets takes place by means of three overhead cranes that move on one set of rails. A standard object from a common simulation software library is used. In particular, the software used is Tecnomatix Plant Simulation [10]. The real features of the shed have been attributed to the model frame, so the geometric scale is one-to-one relative versus the model. An initialization function called at the beginning of the simulation loads into a column of a data table all the portal cranes desired and creates their frame logic and 3D counterparts. The portals need other functions to take the mobile unit (hoist load), then the object, and take it from the originating station to the arrival station. In the specific case, it allows removing the scrap from the various stalls and taking it to the scrap baskets. The scrap park is made up of several stations, so there are more starting points where the scrap is available. These areas are modelled as buffers with an infinite capacity assigned. A structure is therefore built in and the starting points are arranged in matrix format. The same procedure is used to represent the scrap baskets which, unlike the scrap pick-up points, are connected to a drain block and then are to be deleted. The base is to create recipes and a function that reads recipes data from a table is provided. This table refers to one of the preparation points; therefore, in this specific case to the three scrap preparation points which are the baskets. This table consists of two columns: a first column where each row represents a location, i.e. a scrap pick-up point (i.e. the buffers), and a second column that contains a number, which indicates the number of catches that must be done in that location.

A queue is created for each portal crane with the following features:

● A function to take the first object of the queue, which is then removed;
● A function which allows the analyst to add an object;
● A function to take the first object at the top without removing it; therefore, the analyst can examine it as often as necessary.

The recipes are provided for more than one basket depending on the type of steel to be made. The scrap pick-up location and the number of pick-up tasks for each location will be provided for each recipe. The traveling salesman problem can be compared to the scheduling

problem for the loading of scrap baskets. In the traveling salesman problem, the aim is to find an optimal sequence of points to visit such that the sum of the distances to go from one point to another is minimal. The problem relating to the loading of the scrap baskets is similar, in the sense that there is a sequence of operations to be carried out, specifically the movements with the bridge crane, with the difference that the sum of the times necessary to load the baskets must be minimized, in contrast to minimizing the sum of the distances to be covered. To do this, genetic algorithms will be used for their properties in this first customization of the test frame for the comparison with the Neural Network approach. A table that indicates the sequence of tasks and a function that indicates the goodness of our solution are created. In the case studied, the function will be the completion time of the baskets which, to minimize it, must optimize the problem and then sequence the baskets. For the fitness function, a method will be created to communicate with the GA optimization tool from the software library described in the next paragraphs. The table, which will be used to configure the model to read the sequences, will define the sequence in which we will prepare the baskets. The assignment of the basket target is needed, since in the system studied there are three places to prepare them.
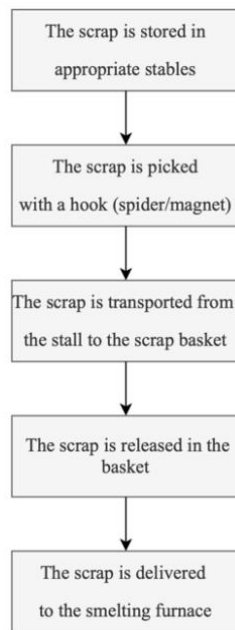

Figure 1. Process flowchart

D. Genetic Algorithm Solution

The use of Genetic Algorithm is an efficient solution because the optimization task at hand has a large number of variants of different solutions. In the specific case, a GA wizard is instantiated to integrate genetic algorithms into the test frame of the simulation model. Table I summarizes the inputs to and the outputs of the optimizer. In terms of inputs, we assume N missions (*OrderID* column) are carried

out in a time span. A recipe (*RecipeID* column) and a to-fill-basket (*Basket* column) are assigned to each mission. This recipe tells how many times the scrap must be taken from a stall point; hence, the mission will end only when all the scrap planned in the recipe is taken. After assigning a recipe to each mission, a deadline (*DueDate* column) is provided. This is the time the mission must be completed. The solutions generated by the Genetic Algorithms wizard object are passed to the simulation model, which in turn will be configured accordingly. Standard libraries from the simulation software are used. At the end of the simulation run, the test frame model transfers the resulting fitness value to the GA wizard object. When the Genetic Algorithms creates individuals, who define the same parameterization, the GA wizard recognizes it and uses the fitness values of the individuals already evaluated. Performing optimization therefore does not waste additional time for multiple assessments of an individual. A first important advantage compared to classical methods is the immunity of genetic algorithms to the number of variables, whereas classical numerical methods are often imprecise due to rounding errors which increase rapidly as the number of variables increases. GA are often used in problems where the objective function depends on a large number of variables, typically five or more. A second, but no less important, property of these algorithms is that the method of finding the solution does not presuppose any type of linearity of the problem, so it is also possible to face problems in which one or more non-linearities exist, possibly also in the form of constraints. Practically, the GA wizard object will do the sequencing, as well as the optimization, starting from the initial orders list; this sequencing represents the initialization of the simulation.

E. Neural Network Solution

The Neural Network approach under study requires specific work, intended as an extension of the present paper. The scheduler will be created using the system architecture in Figure 2.

The Neural Network suitable for this aim is the Multi-Layer Perceptron (MLP) with one hidden layer and trained using a back-propagation algorithm. Regarding the simulation model, the test frame will be implemented with the following scheme:

- Creation of a connection between the model and the external neural network, to be developed on a software-specific platform (data must be collected continuously in the simulations before specified actions);
- Use of the model test frame to test the NN training algorithm;
- Testing the NN driving system for cranes on the same data set used for the GA solutions.

EndTime	DueDate	Delay	Elapsed
2018/01/01 00:28:41	2018/01/01 01:00:00	0.00	26.04
2018/01/01 01:27:15	2018/01/01 01:10:00	17.25	86.31
2018/01/01 02:06:48	2018/01/01 02:30:00	0.00	93.45
2018/01/01 03:38:04	2018/01/01 03:40:00	0.00	87.56
2018/01/01 03:12:17	2018/01/01 04:55:00	0.00	100.56
2018/01/01 05:51:18	2018/01/01 05:40:00	11.30	131.55
2018/01/01 04:46:51	2018/01/01 04:50:00	0.00	89.93
2018/01/01 00:19:17	2018/01/01 00:40:00	0.00	17.87
2018/01/01 01:31:11	2018/01/01 01:35:00	0.00	65.32
2018/01/01 02:43:12	2018/01/01 02:45:00	0.00	67.72

IV. RESULTS AND DISCUSSION

A. Genetic Algorithm Performance Evaluation

The results from the model simulation for the case study are shown in Table I. The solution represents the best sequence of operations that minimize the fitness function which is intended as the sum of process and delay times contained in the column *Delay*. Intuitively, the shorter the delays, the better the configuration obtained. The objective function, also called the fitness function, is the sum of the *Delays* and the minimum is obtained for Generation Four which reaches a value of roughly 28.55, as shown by the red curve in Figure 3. More quantitative details of the optimization output are included in Table II.

To reach the optimal value, 15 generations with 20 individuals per population and five generations per individual have been chosen empirically. Multiple solutions for each generation are shown in Figures 4 through 8.
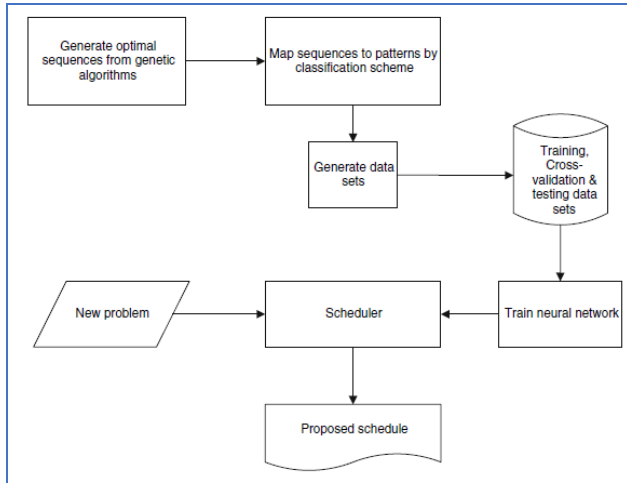

Figure 2. NN scheduler architecture system

TABLE I. OPTIMIZATION SCHEDULING

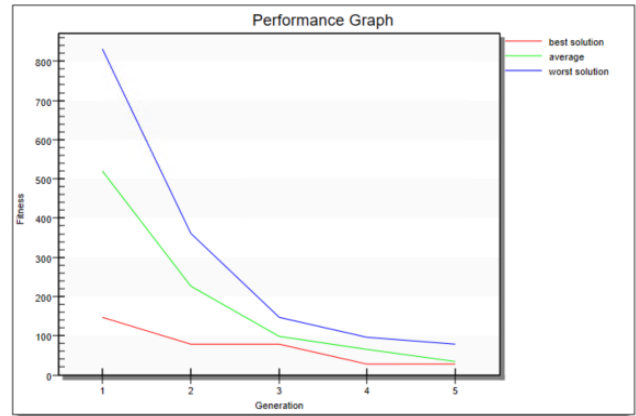| OrderID	RecipeID	Basket_Seq	StartTime
9	Recipe4	Basket1	2018/01/01 00:02:38
8	Recipe1	Basket3	2018/01/01 00:00:56
4	Recipe3	Basket1	2018/01/01 00:33:21
6	Recipe1	Basket1	2018/01/01 02:10:31
3	Recipe2	Basket3	2018/01/01 01:31:43
5	Recipe2	Basket1	2018/01/01 03:39:45
1	Recipe4	Basket3	2018/01/01 03:16:55
10	Recipe4	Basket2	2018/01/01 00:01:25
2	Recipe2	Basket2	2018/01/01 01:25:52
7	Recipe1	Basket2	2018/01/01 01:35:28

Figure 3. Fitness function trend for different solutions

B. Experimental NN performances evaluation

In the hypothesis of this paper, performances very close to GA ones are expected with the neural network approach. A peculiar expected result, to be quantitatively inspected, is the capability of the NN scheduler to quickly react to scenarios never seen before. From commercial feedback collected so far, a performance gain of 50% to 70% is expected, compared to other optimization methods.

V. CONCLUSIONS

The use of Genetic Algorithms has proven to be a good method for solving NP-hard problems, hence also the problem related to the scheduling of the movements of the overhead cranes for loading scrap baskets in the steel industry. Certainly, of fundamental importance is the creation of the model because in order to avoid errors or to avoid obtaining results that differ too much from reality, it is necessary to make the virtual model as equivalent as possible to the real one, creating a Digital Twin. This overhead crane scheduling problem has received attention in

the literature and different methods have been used (network flow model, Tabu research, MAS based), but the performance of the Genetic Algorithms is considered the best. Even in the case of multiple gantries, the method based on Genetic Algorithms provides a solution to improve their efficiency and the introduction of the GA Wizard object improves the preparation times of the scrap baskets for melting furnaces; consequently, also an improvement of the overall performance of the steel company is achieved.

The most important achievement of the paper is the development of a challenging test frame to compare GA results with the ones expected by the use of a neural network technique. Further studies will complete the comparison proposed.
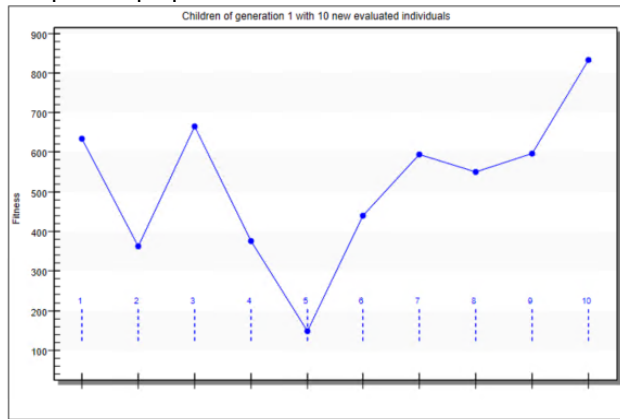

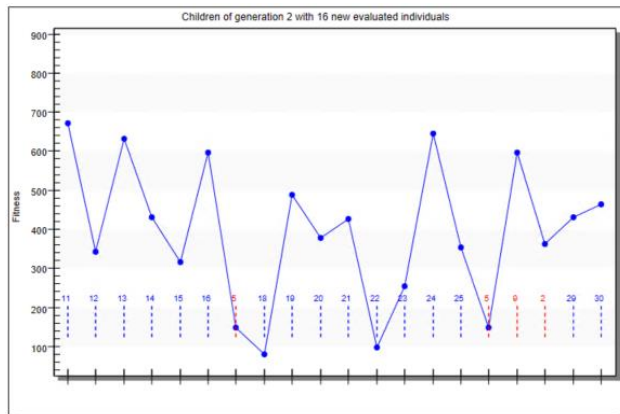Figure 4. Children of generation 1 with 10 new evaluated individuals

Figure 5. Children of generation 2 with 16 new evaluated individuals
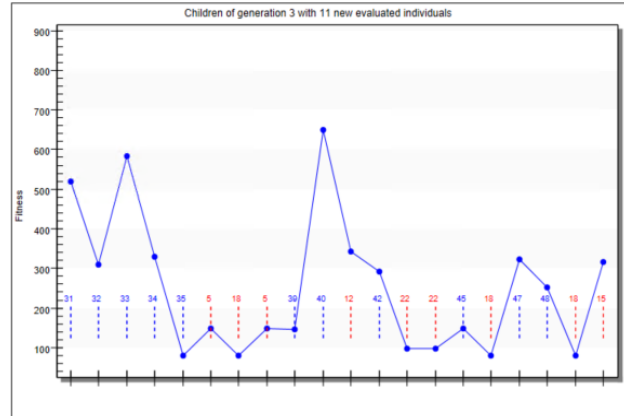

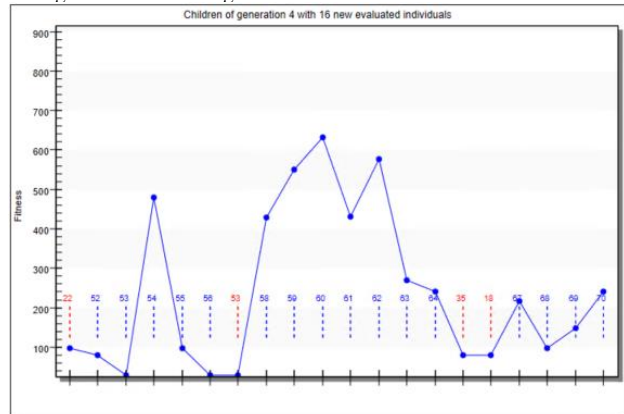Figure 6. Children of generation 3 with 11 new evaluated individuals

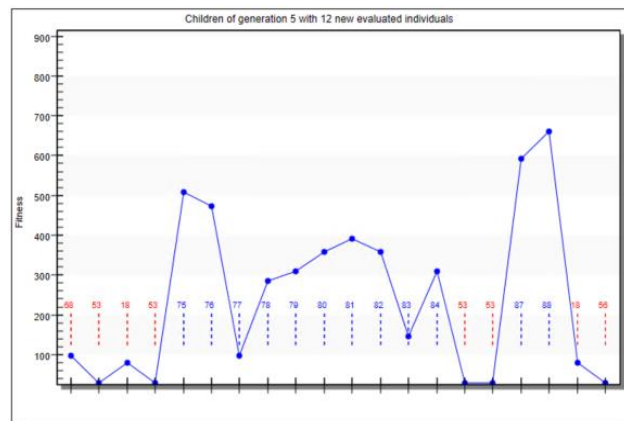Figure 7. Children of generation 4 with 16 new evaluated individuals

Figure 8. Children of generation 5 with 12 new evaluated individuals

TABLE II.   BEST INDIVIDUAL FOR GENERATION FOUR

Individual	Fitness	Chromosomes	Observations
Gen 4 Ind 3	28.55409	Chrom 1	Fitness 53
Gen 4 Ind 6	28.55409	Chrom 1	Fitness 56
Gen 2 Ind 8	78.70279	Chrom 1	Fitness 18
Gen 3 Ind 5	78.70279	Chrom 1	Fitness 35
Gen 4 Ind 2	78.70279	Chrom 1	Fitness 52

REFERENCES

[1] C.B. Ma, D.F. Zhu, H. Wang, Y.Q. Zhao, and J.X. Xu, "Simulation model for crane scheduling in workshop of steel-making plant based on MAS," *2010 International Conference on Computer Application and System Modelling* (ICCASM 2010), vol. 5, pp. V5-404, 2010.

[2] G. Zäpfel and M. Wasner, "Warehouse sequencing in the steel supply chain as a generalized job shop model," *International Journal of Production Economics*, vol. 104(2), pp. 482-501, 2006.

[3] A. Graunke, G. Burnett, C. Hu, and G. Wirth, "Decision support model to evaluate complex overhead crane schedules," *Proceedings of the Winter Simulation Conference* 2014, pp. 1608-1619, 2014.

[4] A.S. Jain and S. Meeran, "Job-shop scheduling using neural networks," *International Journal of Production Research*, vol. 36(5), pp.1249-1272, 1998.

[5] G.R. Weckman, C.V. Ganduri, and D.A. Koonce, "A neural network job-shop scheduler," *Journal of Intelligent Manufacturing*, vol. 19(2), pp.191-201, 2008.

[6] X. Xie, Y. Zheng, and Y. Li, "Multi-crane scheduling in steel coil warehouse," *Expert Systems with Applications*, vol. 41(6), pp. 2874-2885, 2014.

[7] A. Lim, B. Rodrigues, F. Xiao, and Y. Zhu, "Crane scheduling using tabu search," *14th IEEE International Conference on Tools with Artificial Intelligence*, pp. 146-153, 2002.

[8] Y. Yuan and L. Tang, "Novel time-space network flow formulation and approximate dynamic programming approach for the crane scheduling in a coil warehouse," *European Journal of Operational Research*, vol. 262(2), pp. 424-437, 2017.

[9] X. Xie, Y. Zheng, and Y. Li, "Genetic algorithm and its performance analysis for scheduling a single crane," *Discrete Dynamics in Nature and Society*, 2015.

[10] M.R.K. Mes, "Simulation Modeling using Practical Examples: A Plant Simulation Tutorial," University of Twente, 2017 https://www.utwente.nl/en/bms/iebis/staff/mes/plantsimulation/tutorialplantsimulation13v20170726.pdf .