



Creative Construction Conference 2018, CCC 2018, 30 June - 3 July 2018, Ljubljana, Slovenia

A simulation-based approach for optimal construction planning and scheduling

B. Naticchia^a, A. Carbonari^{a*}, M. Vaccarini^a

^a*Department of Civil and Building Engineering and Architecture (DICEA), Polytechnic University of Marche, 60131, Ancona, Italy*

Abstract

Project schedules in construction are responsible for an efficient deployment of resources on the job-site and for the overall efficiency of work progress. Current approaches too often lead towards sub-optimal work plans or, sometimes, even scarce productivity. For that reason, a lot of research was devoted to the development of automated scheduling tools, which can provide optimal solutions while requiring reasonable computational effort. As a consequence, planners can save their time and involved resources can benefit from the efficient organization of work packages and tasks. However, automation in construction scheduling is a tough challenge, because it requires to generate and optimize multi-objective problems, which usually include several parameters. In addition, deviations from what expected is quite frequent, and these algorithms should be able to quickly revise the previous plan, in fact performing dynamic planning. Hence, this paper presents an agent-based approach, which can be integrated in a BIM-based platform to perform automated scheduling of construction works. The BIM component can provide instant access to relevant information, which must be integrated with some user defined inputs, in order to feed the optimization algorithm. This algorithm was based on the multiple ant colony system for vehicle routing problems with time windows, because it can handle several resources travelling through many locations, each one performing its task, even in the presence of time constraints. The optimization was performed with respect to both overall makespan and total costs. An application to the case of bored piles execution will be presented in this paper.

© 2018 The Authors. Published by Diamond Congress Ltd.

Peer-review under responsibility of the scientific committee of the Creative Construction Conference 2018.

Keywords: construction planning; scheduling automation; agent-based simulation.

1. Introduction

Project schedules in construction projects help project management teams handle interrelated critical aspects of management, such as time, cost, resources. It requires availability of information both at the pre-execution stage and during execution. The standard construction practice is commonly done applying diagram methods to sequence activities that are based on work packages defined according to a work-breakdown-structure. In this phase, the planner's background and experience play a vital role in the creation of a construction schedule, because she/he is in charge of assuring consistency with the scope, plan efficiency and proper deployment of resources. Then, progress management usually involves deviation analyses between the actual progress and the initial plan that is constantly being updated and revised. The often scarce communication and the absence of automation requires the adoption of standard methods, such as face-to-face meetings and paperwork. This approach has been identified as an inhibitor to increasing productivity and a frequent source of miscommunication and rework [1]. To solve the problem of insufficient information and sub-optimal planning, researchers turned towards automating the process of generating schedules. Research has been carried out through the past decades in several fields, where automated tools demonstrated their ability to provide a variety of advantages, such as quickly generation and updating of schedules, optimization of project resource allocation and levelling, of the makespan and/or total costs.

However, there are several research challenges that are still open. In this paper, we will mainly focus on computational tools and on information handling. As far as computation is concerned, in the past decades a number of solutions were suggested, ranging from case-based and knowledge-based reasoning to model-based simulation,

until the latest approaches inspired from artificial intelligence [1]. More recently, thanks to the advent of powerful computational tools, the potential of agent-based simulation was highlighted, and first applications in the field of optimal planning were produced. This approach manages to incorporate the inherent variability that arises from the independent construction subjects' behaviour as they interact on a construction site [2]. In fact, multi-agent simulation is suitable for modelling resources' behaviour and interactions in complex settings, like in construction. These models are capable of specifying the characteristic of trade crews, their work methods, the amount of work, workspaces and dependencies between tasks. As a result, simulations encapsulate both variability and uncertainty of the construction workflow [3].

The next section 2 of this paper reports the relevant background in the field, section 3 concerns a new algorithm for agent-based simulation, section 4 first suggests an overall framework for automated planning of construction schedules, then details about one scenario where it was implemented, finally it discusses its main findings. Conclusions and references sections close this paper.

2. Scientific Background

Automation in scheduling has been studied mainly in the last three decades and several approaches were experimented [1]. Case-based reasoning (CB) is able to exploit the specific knowledge of formerly practiced situations and compares present problems with an earlier situation, which is then used to solve and explain the new problem [4]. Similarly, knowledge-based reasoning integrates knowledge from several sources (e.g. construction rules, basic physics etc..) for the purpose of generating schedules, even retaining learned experience. Genetic algorithms (GA) is an optimization tool that uses a heuristic search which mimics the natural evolutionary process. Using a mathematically defined fitness function as the objective function, the initial randomly generated genomes can evolve into optimized solutions for a given problem [5]. An expert system, which belongs to the artificial intelligence tools, is defined as a computer based algorithm that imitates human decision-making skills. Expert systems are designed mainly using if-then structures instead of regular practical codes. Neural networks are inspired by the brain of animals and are able to perform pattern recognition using "all-or-none" (i.e. a type of binary language) rule of the nerves. Several calculators are known, e.g. the "Hebb's rule". They need to perform optimization relying on a huge database, so their learning process must be fed by thousands of records. Model-based systems use formalized construction method models to perform the scheduling. These methods usually decompose higher level activities of the schedule into lower level activities to ease the linking of the schedule with diverse level of details.

However, none of these approaches can tackle all the relevant aspects of the scheduling problem. Indeed, case-based and knowledge-based are good at identifying tasks and sequencing them [4]; although genetic algorithms is known as a meta-heuristic optimization method that is mainly suitable for solving multi-objective problems [5], they were applied in scheduling mainly to optimize resource utilization and perform resource levelling to come up with better project schedules compared to heuristic methods [1, 6]; expert systems provided an ample flexibility of applications, but their development procedure is not standardized, yet, and relies on the experience and judgement of the researcher [1]; neural networks, besides requiring a huge dataset as previous knowledge, they were shown to be a powerful tool for dealing with some specific scheduling problems (e.g. job-shop scheduling, single-machine scheduling, timetable scheduling), it was not used for construction sequencing and scheduling [7]; model-based approaches requires a great effort in the development of rules (e.g. spatial reasoning) for automating, which some authors tried to simplify through the use of templates stored as a knowledge base [8]. As a result, one of the most important gaps to be addressed is the development of a hybrid approach that could address multiple objectives associated with scheduling [1].

Another important challenge is the development of adaptive algorithms. In fact, project planning is usually done at different levels. While a master schedule provides a global view of project milestones and the overall execution strategy, it must be always specified by a short-term schedule, which is a more detailed plan listing work to be done within a relatively short time window based on the most up-to-date site conditions and performance [9]. In the last planner system this second level is called look-ahead schedule. But the construction site is not stationary, and the plan must react to system changes on a real-time or near-real-time basis. So, when it is generated by a simulation model, it requires that such a model is able to capture site condition changes constantly and be updated accordingly so that the changes and their impacts can be evaluated in a timely manner [9]. In other words, adaptive framework must include real-time data acquisition modules, process interpretation modules, adaptive modelling and optimization algorithms. The long term goal of the adaptive modelling component is to streamline the model-updating procedure by taking advantage of inputs from the data acquisition and the process knowledgebase components. For example, real-time and most-recent values of an activity duration can detect changes in the duration pattern based on past measurements.

Further automation can be exploited in this process if a BIM repository is used as the archive of all the information required by the optimization algorithms to perform its estimations. To date, BIM was mainly used to generate quantity take-offs, 4D scheduling and building simulations. However, even the generation of construction schedules can retrieve data (e.g. spatial, geometric, quantity, relationships and material set of information) from what is stored in BIM models [10]. This approach would achieve significant time reductions in scheduling, compared to the traditional manual methods.

For the reasons stated above, in this paper we highlight that automated scheduling can take advantage of BIM-based structured information [10]. Then, an algorithm based on multi agent-based simulation will be suggested and its performances analysed, showing that it is able to imitate real world process of systems, where the global behaviour emerges as a result of interactions of single agents [11]. Agents can be active, proactive, autonomous, cooperative, adaptive and mobile. They interact to reach a global objective. This tool allows us to evolve from single-objective optimization towards a multi-objective optimization problem based on ant-colony simulation. In this approach, every objective can be optimized by a different colony, that is cooperating and contributing to the whole process. In addition, the framework proposed in this paper was designed so as to be capable of continuously re-planning the schedule as a result from unexpected occurrences, in fact performing dynamic and adaptive scheduling. In the implementation, several scenarios will be proposed in order to show the sensitivity of the algorithm to several inputs, such as the ratio between direct and indirect costs.

3. Development of the optimal planning algorithm

3.1. Overview

According to some research findings from the manufacturing field, when the global behavior of a complex system is the result of interaction among many actors, it resembles the behaviour of food-foraging ants, which is called stigmergy. Even from the computational point of view, stigmergy is a very efficient approach because it is capable of incorporating nonlocal information while employing only local reality-mirroring components [12]. To sum up, the following steps are performed in a stigmergic approach:

- In absence of any signs in the environment, ants perform a randomized search for food;
- When an ant discovers a food source, it drops a smelling substance, called pheromone, on its way back to the nest while carrying some of the food. The pheromone trail evaporates if no other ant deposits fresh pheromone;
- When an ant senses a pheromone trail, it will be urged by its instinct to follow this trail to the food source and will deposit pheromone itself on its way back to the nest.

This pattern is an emergent behavior of the ant colony, that is ordered and is robust against the uncertainty and the complexity of the environment. Although information about the presence of food is made available locally, it affects the global behavior of the colony and the state of the environment.

The algorithm presented in this paper is an extension of the multiple ant colony system for vehicle routing problems with time windows (MACS-VRPTW), which is reported in sub-section 3.2. The enhancement proposed by the authors is detailed in sub-section 3.3 and concerns some steps for overall cost optimization. For the sake of clarity, Fig. 1-a depicts the whole logic: the grey boxes framed with a solid line represent the typical steps found in a MACS-VRPTW algorithm, whereas the white dashed-line boxes depict those parts of the algorithm that were added to perform what is described in sub-section 3.3.

3.2. Multi-objective optimization through MACS-VRPTW

The basic algorithm implemented to perform optimization was the vehicle routing problem. Indeed, driving crews along the most cost-effective path recalls problems about routing optimization. More specifically, we refer to the multiple ant colony system for vehicle routing problems with time windows (MACS-VRPTW), which performs ant colony optimization [13].

Basically, MACS-VRPTW is organized with a hierarchy of artificial ant colonies designed to successively optimize a multiple objective function: the first colony minimizes the number of vehicles while the second colony minimizes the travelled distance. A VRP is composed of n customers served from a unique depot c_0 . Each customer $c_i, i = 1, \dots, n$ asks for a quantity q_i of goods and a vehicle capacity Q is available for delivery. Each delivery cannot be split and the vehicle has to periodically return to the depot for reloading. On the overall, the problem is represented as a graph made of a node set $C = \{c_0, c_1, \dots, c_n\}$ and arcs $L_{ij} = (c_i, c_j): i \neq j$ to which a matrix of travel time values t_{ij} is associated.

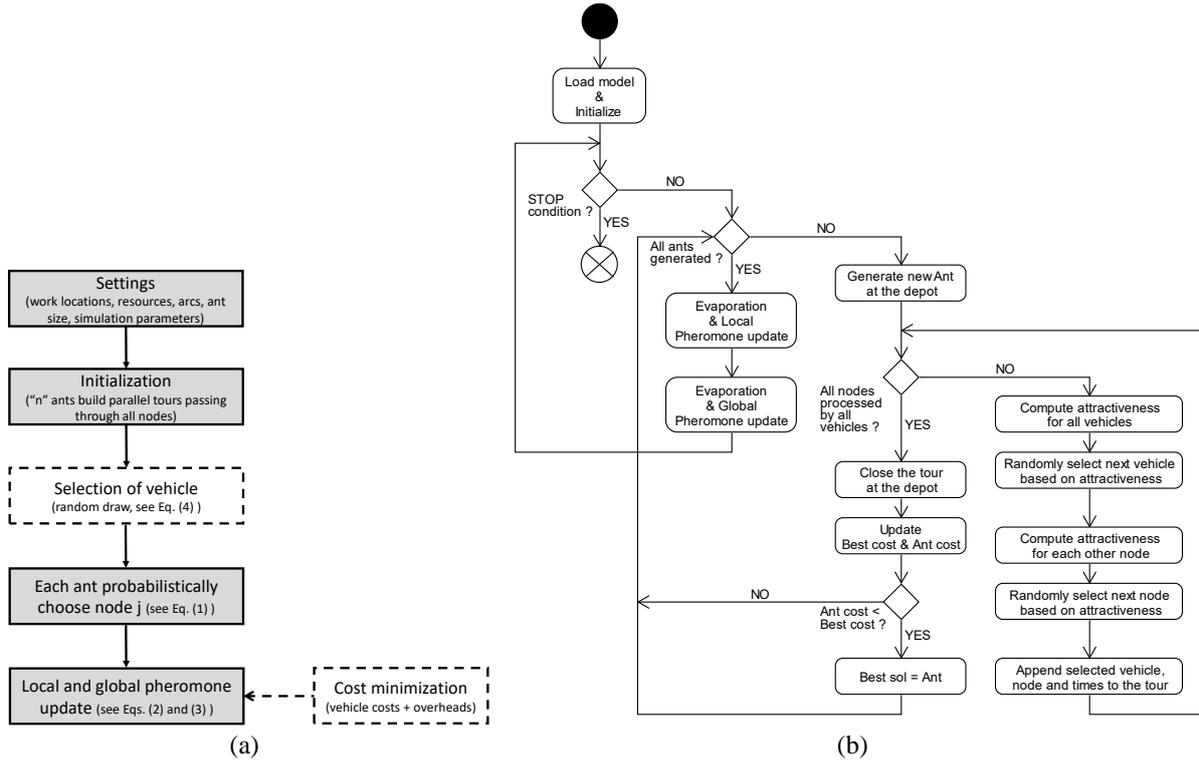


Fig. 1. Overview of the enhanced MACS-VRPTW algorithm (a) and details of the MACS-VRPTW implemented in this paper (b).

The goal is to find a set of tours of minimum total travel time, where each tour starts and ends at the depot c_0 . Extensions to the basic problem include: service time for each customer; duration limit of each tour.

In this paper the VRP with time windows, i.e. VRPTW, was applied. This problem includes for the depot and each customer c_i a time window $[b_i; e_i]$, during which each customer must be served (i. e. it must be served between starting time b_i and end time e_i). The VRPTW solution approach proposed in [13] assumes that the tours are performed by a fleet of identical vehicles and that the optimization is based on the Ant Colony System (ACS) that is briefly described in the following. ACS is applied for minimizing both the number of vehicles and travel time (i.e. to achieve a multi-objective optimization). To this purpose, two measures (i.e. heuristics) are associated to each arc: closeness (η_{ij}) and pheromone trail (τ_{ij}). The first one is the inverse of the distance, the second one is dynamically changed by ants at runtime. Pheromone trails are used in conjunction with the objective function to construct new solutions: a higher attractiveness is given to the arcs with a stronger pheromone trail. Pheromone levels give a measure of how desirable (attractive) it is to add a given arc in a partial solution. At runtime, n_a ants build their own tours in parallel. Each ant is assigned to the depot c_0 and must build a feasible solution, by iteratively adding new nodes until all nodes have been visited just once. When ant k (for $k = 1, \dots, n_a$) is located at node i , it chooses the next node j probabilistically in the set of feasible nodes N_k^i (i.e. the nodes that have not been visited yet and that comply with the given time window $b_i \leq t_i \leq e_i$).

The attractiveness of a node is defined by Eq. 1:

$$p_{ij} = \begin{cases} \frac{\tau_{ij} \cdot [\eta_{ij}]^\beta}{\sum_{c_l \in N_k^i} \tau_{il} \cdot [\eta_{il}]^\beta} & \text{if } c_j \in N_k^i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Parameter β weighs the relative importance of the heuristic evaluation based on the distance, with respect to the pheromone trail.

The probabilistic rule, at each iteration, randomly decide between two alternative selection criteria: exploitation and exploration. This is done based on a parameter $q_0 \in [0; 1]$ that determines the relative importance of exploitation versus exploration: the higher q_0 is the more likely the exploitation criterion is with respect to the exploration criterion. Ant k with probability q_0 sorts out the next node as that one with the highest $\tau_{ij} [\eta_{ij}]^\beta$, while with probability $(1 - q_0)$ it selects the node as an observation of the discrete random variable with probability distribution $p_{ij}, \forall j = 1, \dots, n$ (see Eq. 1).

In the MACS-VRPTW, it is optional to include that, once each ant has built a complete solution, it is tentatively improved using a local search procedure.

The best solution is used to modify the pheromone trail matrix (τ_{ij}) as follows:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \frac{\rho}{J_{\Psi}^{gb}}, \quad \forall (i, j): c_i, c_j \in \Psi^{gb} \quad (2)$$

Where $0 < \rho < 1$ and J_{Ψ}^{gb} is the length of J^{gb} , i.e. the shortest path generated by ants since the beginning of computation. Future ants will use this information to generate new solutions around the best solution.

Locally, when an ant moves from node i to node j , the amount of pheromone trail on arc L_{ij} is decreased by the amount:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (3)$$

where τ_0 is the initial value of trails.

Then, the process is iterated generating again m ants until a termination condition is met. This algorithm was used in MACS-VRPTW with two objectives: minimization of the number of tours and minimization of the total travel time. To this purpose, two independent colonies are used, one per each objective, but both share the variable Ψ^{gb} .

3.3. Extension towards multiple activities

In the final version of the MACS-VRPTW we implemented in this paper (Fig. 1-b), some changes with respect to the reference version detailed in sub-section 3.2 were made. First, only the ACS-Time colony was necessary. There was no need to use another ant colony (although it can be implemented in the MACS-VRPTW), because the objectives of reducing the number of vehicles and minimizing the total travel time can be done by means of cost optimization. Secondly, the local search option of MACS-VRPTW was not implemented but it could be used for fine tuning the solution. Finally, no capacity limit was set (i.e. the Q value was not inputted and infinite capacity is assumed); however, this option was kept open and it is expected to be implemented in future versions for handling activities that require to return periodically to the stocking area.

Rather, additional functionalities were added in the final algorithm, that are necessary for the construction field, where multiple different activities are involved in the planning task. As shown in Fig. 1-b, each ant consists of all activities (vehicles) performed on all customers (nodes) by following the correct operation sequence. The time window can be differentiated for each vehicle according to the nodes and operation that must be served and the specific productivity and costs can be set for each vehicle. The computation of costs was generalized: direct costs include those ones that are generated by travelling between nodes, waiting for work (e.g. because the previous task was not accomplished in the next node) and operation (e.g. according to productivity). Also, indirect costs were added, which depend on the total elapsed time. Constraints that take into account for prerequisite activities have been added in order to enforce the correct operation sequence, hence the relationships between vehicles match the ones between activities. Moreover, the best solution is re-evaluated, compared and updated at each iteration to allow for dynamic re-planning when unexpected events occur.

The generalization to multiple activities required a double selection process for making each movement of an ant: not only the next node have to be selected but also the next vehicle must be chosen for operating in that node. When ant k (for $k = 1, \dots, n_a$) is located at node i , it chooses the next vehicle v probabilistically in the set of feasible vehicles V_k^i (i.e. the vehicles that have not visited all nodes yet) based on the number of feasible nodes n_k^{iv} for that vehicle:

$$p_v^i = \begin{cases} \frac{n_k^{iv}}{\sum_{u \in V_k^i} n_k^{iu}} \cdot \gamma & \text{if } v \in V_k^i \text{ and } v \text{ was operating in node } n_i \\ \frac{n_k^{iv}}{\sum_{u \in V_k^i} n_k^{iu}} & \text{if } v \in V_k^i \text{ and } v \text{ was not operating in node } n_i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\gamma > 1$ is a factor used to increase attractiveness of vehicles that was already active and, therefore, to foster solutions with continuous vehicle operations. Similarly to what is done in the classic MACS-VRPTW, the next vehicle is then selected by an observation of the discrete random variable with probability distribution $p_v^i, \forall v = 1, \dots, n_v$ (see Eq. 4).

4. Simulation and testing

4.1. General framework and scenarios

The general framework we suggest in this paper is meant to be able to exploit information stored in BIM (e.g. through BIMserver) to assist in generating schedules. Indeed, automatically retrieving information from a BIM model could help achieve significant time reductions in scheduling as compared with traditional manual methods. In general, the data that can be retrieved from a BIM model are those ones regarding the material of construction components, their locations on site and quantities. On the contrary, the remaining information must be defined through inputs from the user, who is in charge of defining the maximum number and types of available vehicles and crews to perform on-site activities, the location of depots where equipment is parked when not in use, constraints about sequencing for activities. Once the algorithm is fed with these data, it can perform optimization and work out the optimal schedule, as shown in Fig. 2.

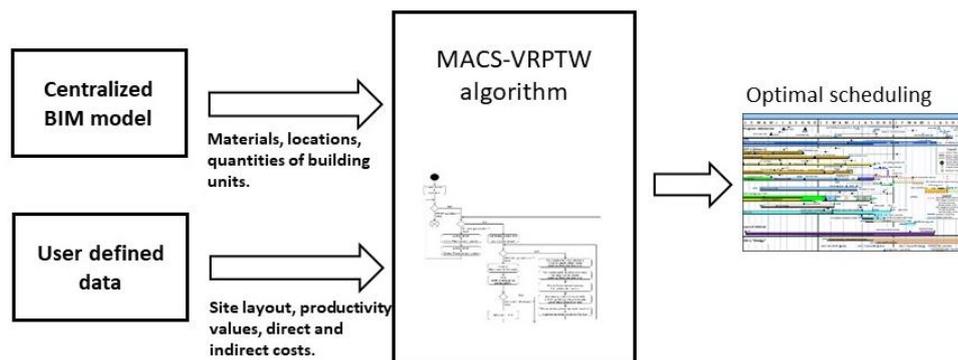


Fig. 2. Framework of the planning system tested in this paper and based on the extended MACS-VRPTW algorithm

The case study developed in this paper concerns bored piles execution, which includes a series of tasks well described in the work by Zayed and Halpin [14]. The main tasks to be performed are, as a first step, adjusting the machine on the pile axis and drilling the pile; as a second step, erecting the rebar cage using a crane; as a final step, erecting the concrete pouring tool and pouring concrete until the pile is finished. After this process is accomplished for one pile, the machines need to be relocated and the process is repeated again for the next one.

Hence, the outcome of our planning process was a schedule made of three activities (i.e. drilling, rebar, pouring), which are accommodated so as to provide the sequence of piles that must be worked by each machine and its crew (i.e. drilling machine, crane and pouring tool). Of course, at every pile the possibility for a machine to perform its task was constrained by the sequence of technical steps typical of bored piles, i.e. drilling-rebar cage-pouring. As a result, if we focus on a particular machine, the algorithm will determine the ordered list of piles where it must perform its task. To be noticed that the ordered list that is optimal for one machine can be different from the optimal ordered lists for the machines that follow the first one. Instead, the depot location was determined by the user. No restrictions in terms of time windows was set in the first application shown in this paper, which means that the algorithm was left free to find out the best solution in terms of overall costs and makespan.

4.2. Implementation

The specific layout of piles chosen to test the approach shown in this paper is the one depicted on Fig. 3, that is the plan view of the BIM model containing the project to be executed. At this point, the BIM model was made of just 32 piles on three curved rows, numbered as shown in the figure. Although not fully automated, the quantity take-offs were extracted directly from the corresponding BIM model. These datasets included piles diameter, depth and volume.

Productivity of drilling activities were estimated according to international literature [15], because it provides values averaged over the most common types of soil and machines that can be adopted. The other productivity values (i.e. rebar cage positioning and concrete pouring), direct and indirect costs were derived from databases and software tools for the estimation of unit costs of activities referred to the Italian context [16]. The productivity and costs values inserted in the algorithm are shown in Table 1, where it can be noticed that overheads are high as 50 €/h and were attributed to the whole construction site. Additional indirect costs were accounted to the equipment. Denoting \bar{d} as the mean distance between nodes, the other parameters used to run the algorithm are: $n_a = n$, $\tau_0 = 1/(n \cdot \bar{d})$, $q_0 = 0$, $\beta = 2.5$, $\rho = 0.5$, $\gamma = 15$.

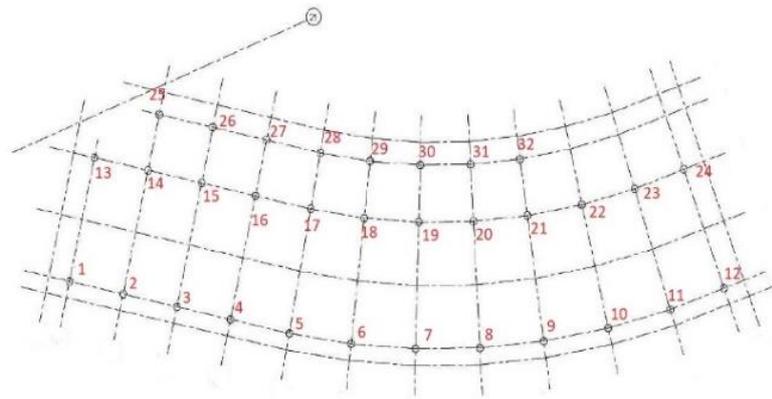


Fig. 3. Layout of bored piles whose optimal planning was automated by means of the extended MACS-VRPTW.

Table 1. Productivity and costs values inserted as inputs in the MACS-VRPTW.

Type of value	Drilling machine	Rebar cage	Concrete pouring
Productivity	10.00 m/h	0.16 t/h	2.86 m/h
Direct costs	83.32 €/h	74.96 €/h	162.00 €/h
Indirect costs	23.74 €/h	10.86 €/h	10.32 €/h
Travelling speed	1.50 km/h	1.50 km/h	1.50 km/h
Overheads (job-site)		50.00 €/h	

The MACS-VRPTW applied to this case study gave back the optimal schedule shown in Fig. 4-a, where the y-axis reports the list of activities: no. 1 is drilling, no. 2 is rebar cage erection and no. 3 is concrete pouring; the x-axis reports the elapsed time. Fig. 4-b depicts the number of iterations vs the value of the cost function related to the work execution.

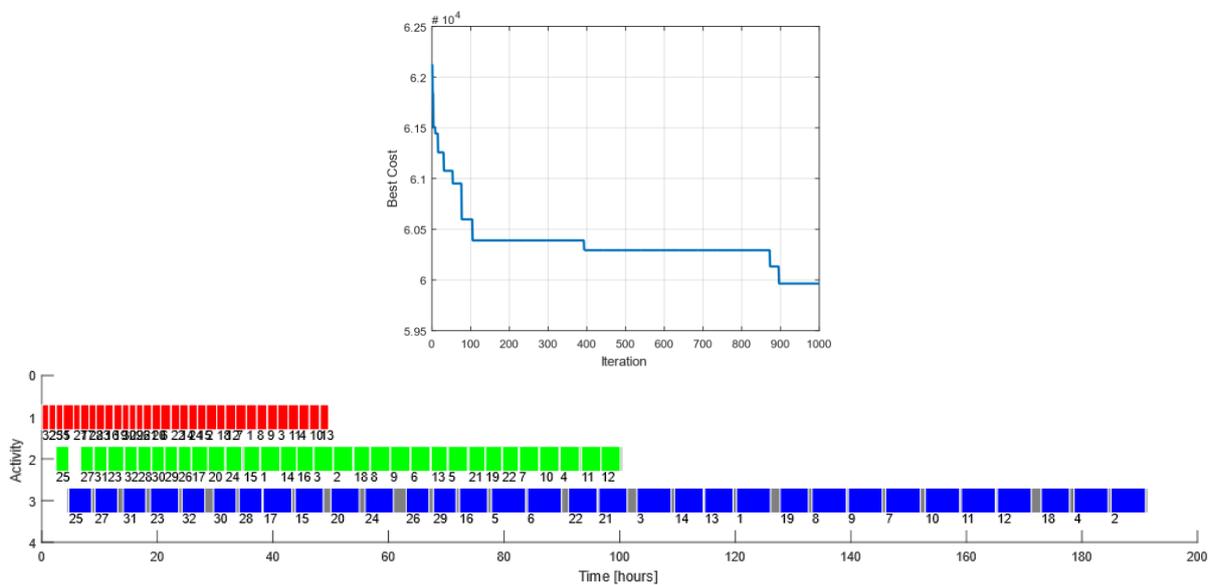


Fig. 4. Trend of the best cost (top) and final optimal plan for the execution of bored piles (bottom).

4.3. Discussion

As expected, the ordered lists of piles suggested for the execution of the three activities by the respective involved machines is not invariable. In fact, the sequence of piles worked within the first activity is different from the sequence of tasks worked within the second activity which is, in turn, different from the third one. In addition, the work performed by the three machines is interrupted by waiting times, that are necessary when a machine cannot perform its task until the previous one is not accomplished at the pile it has ordered as the next one on the

list. Indeed, waiting times increase indirect costs, but sometimes they can be accepted if the total travelled distance and related costs decrease. For example, the average waiting time for the second machine is equal to 0.06 h. Hence, this algorithm performs a trade-off between direct and indirect costs. The overall time to perform all the activities (i.e. makespan) is equal to 191.4 h. The total cost of the work performed is equal to 59,964 €. Finally, it must be noticed that this solution is the one that the algorithm worked out after 1000 iterations, as depicted on Fig. 4-b. Due to the nature of the algorithm, it could be capable of improving the suggested solution while increasing the number of iterations, until it stabilizes to the best one. However, when the optimization process must be quick, like in dynamic planning, this algorithm is always able to provide a sub-optimal solution within a pre-determined time limit, e.g. if we had stopped after 800 iterations, the solution provided at that point would have been slightly worse than the one suggested after 1000 iterations.

5. Conclusions

An overall framework for automated planning of construction works was presented in this paper. It consists of several units: one unit is expected to retrieve information about building component materials, locations and geometry directly from the BIM model, without human intervention; one unit will manage information about the construction sites (e.g. locations of depots); another unit will implement the MACS-VRPTW algorithm to automatically generate the optimal schedule; the last unit will visualize the resulting schedule. The MACS-VRPTW algorithm developed in this paper performs optimization through the application of the stigmergic technique, which is based on ant colony search for the optimal solution. It belongs to the wider category of multi agent-based simulation, which looks very suitable for construction, where the overall behavior can be interpreted as the result of the interaction between several agents (i.e. crews), each one pursuing its own objective but obliged to interact with the remaining resources and with the context.

In the algorithm presented in this paper, not only the overall makespan, but even the total costs of the construction site under analysis were optimized. The results show that after 1000 iterations the algorithm was able to optimize the overall cost function and to work out a solution that could limit the average waiting time of each crew. Also, it visualized the optimal schedule for the case study under analysis, according to the provided inputs. Among the main benefits, we cite the very efficient computational capabilities and the dynamicity of the algorithm, that can be run even in real-time during work execution, if revised planning is requested after deviations occurring from what expected.

References

- [1] Vahid Faghihi, Ali Nejat, Kenneth F. Reinschmidt, Julian H. Kang, Automation in construction scheduling: a review of the literature, *Int. J. Adv. Manuf. Technol.* 81:9-12 (2015) 1845–1856.
- [2] M. Watkins, A. Mukherjee, N. Onder, K. Mattila, Using agent-based modelling to study construction labor productivity as an emergent property of individual and crew interactions, *Journal of Construction Engineering and Management.* 135:7 (2009) 657-667.
- [3] Ling Ma and Rafael Sacks, Agent-based simulation of construction workflows using a relational data model, In: *Proceedings of the 24th Annual Conference of the International Group for Lean Construction.* Boston, MA, USA, (2009) 73-82.
- [4] A. Morad, Y. Beliveau, Knowledge-based planning system. *J Constr Eng Manag.* 117:1 (1991) 1-12.
- [5] Konda A., Coit D.W., Smith A.E., Multi-objective optimization using genetic algorithms: a tutorial, *Reliab EngSyst Saf.* 91:9 (2006) 992-1007.
- [6] Bai Y., Zhao Y., Chen Y., Chen L., Designing domain work breakdown structure (DWBS) using neural networks, *Proceedings of: 6th international symposium on neural networks, Wuhan, China.* Springer Berlin Heidelberg (2009) p. 1146-1153.
- [7] Sabuncuoglu I., Scheduling with neural networks: a review of the literature and new research directions. *Prod Plann Control: Manag Oper.* 9:1 (1998) 2-12.
- [8] Kataoka M., Automated generation of construction plans from primitive geometries, *J Constr Eng Manag* 134:8 (2008) 592-600.
- [9] Lingguang Song, Neil N. Eldin, Adaptive real-time tracking and simulation if heavy construction operations for look-ahead scheduling, *Automation in Construction* 27 (2012): 32-39.
- [10] Hyunjoo Kim, Kyle Anderson, SangHyun Lee, John Hildret, Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology, *Automation in Construction* 35 (2013) 285-295.
- [11] Ling Ma and Rafael Sacks, Agent-based simulation of construction workflows using a relational data model, in *Proceedings of the 24th Annual Conference of the International Group for Lean Construction*, Boston, MA, USA (2009) p. 73-82.
- [12] Paul Valckenaers, Hendrik Van Brussels, *Design for the unexpected*, Elsevier, 2016, ISBN: 978-0-12-803662-4.
- [13] Luca Maria Gambardella, Eric Taillard and Giovanni Agazzi, MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows, In: D. Corne, M. Dorigo and F. Glover, *New ideas in optimization*, McGraw Hill, London, UK (1999) 63-76.
- [14] T.M. Zayed and D.W. Halpin, Pile Construction Productivity Assessment, *Journal of Construction Engineering and Management*, vol. 131:6 (2005) 705-714.
- [15] Guild of Project Control – Bored Piling, 2009-02-22. *Planning Planet*, available online at: <http://www.planningplanet.com/wiki/422511/bored-piling>, last access on: 2018/02/09.
- [16] Matteo Gieri, *Analisi dei prezzi unitari in edilizia*, Maggioli Editore (2011) ISBN: 978-88-387-6590-1.