European Journal of Operational Research 307 (2023) 590–603

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

# The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach

Maurizio Boccia[a], Adriano Masone[a,*], Claudio Sterle[a], Teresa Murino[b]

[a] Department of Electrical Engineering and Information Technology, University of Naples "Federico II", Italy
[b] Department of Chemical, Materials and Production Engineering, University of Naples "Federico II", Italy

A R T I C L E I N F O

*Article history:*
Received 15 November 2021
Accepted 13 October 2022
Available online 20 October 2022

*Keywords:*
Assignment
Scheduling
AGV
Battery constraints
MILP
Matheuristic

A B S T R A C T

Nowadays, automated guided vehicles (AGVs) are frequently used in larger systems, known as AGV-based transportation systems, for the movement of goods and materials from one location to another. The design of an efficient and effective AGV-based transportation system requires to address many tactical and operational issues. Among the others, the scheduling of transfer jobs on the AGVs represents one of the main operational issues that has to be solved to overcome delays in production and material handling processes. In this context, a significant research activity on AGV systems and on related scheduling problems has been conducted in the last twenty years. However, most of the contributions neglected the issues related to the AGV battery depletion and recharge. Thus, in this work, we study the AGV scheduling problem with battery constraints (*ASP-BC*). It consists in determining the scheduling of transfer jobs and charging operations of a fleet of homogeneous AGVs such that the makespan of the handling process is minimized. The methodological contribution of our work is twofold. On one side, we propose an original mixed integer linear programming formulation based on the bottleneck generalized assignment problem. On the other side, we propose a three step matheuristic based on the sequential solution of the two subproblems arising from the natural decomposition of the *ASP-BC* and a local search heuristic. The proposed approaches have been tested and validated on simulated and real instances provided by a manufacturing company. The results show the effectiveness and the scalability of the proposed solution methods.



## 1. Introduction

Automated Guided Vehicles (AGVs), firstly introduced in 1955, are driverless transportation systems used for horizontal movement of materials. Since then, the use of AGVs has grown enormously due to the wide range of benefits they produce. In particular, the increased productivity, the enhanced safety, the reduced labor cost, energy consumption, and emissions, lead the AGVs on the rise in several application fields, such as manufacturing, logistics, and healthcare, to cite only a few (Bechtsis, Tsolakis, Vlachos, & Iakovou, 2017; Fragapane, de Koster, Sgarbossa, & Strandhagen, 2021).

In these fields, AGVs might be part of a larger system used for the transportation of goods and materials from one location to another. This system is known as AGV-based transportation system (AGV system). It consists of a transportation network, a physical interface between the production/storage system, an AGV fleet, and a control system. The transportation network connects all the pick-up and delivery points. These points operate as interfaces between the production/storage system and the transportation system. Finally, AGVs move from one pick-up/delivery point to another on fixed or free paths. The main aim of an AGV system is to transfer the right amount of the right material to the right place at the right time (Masone, Murino, Sterle, & Strazzullo, 2021). To this end, many tactical (e.g., fleet sizing and flowpath design) and operational (e.g., dispatching, routing and scheduling) issues have to be addressed when designing an AGV system (Vis, 2006).

In this paper, we focus on the AGV scheduling issues whose effective and efficient management allows to improve the overall productivity of transfer operations and to avoid delays in production and material handling processes (Luo, Wu, & Mendes, 2016). A significant research activity on AGV systems and on related scheduling problems has been conducted in the last twenty years. In particular, different scheduling problems were proposed on the basis of the considered AGV requirements and capabilities (e.g., capacity, single/multi-load, and schedules of additional equipment)

* Corresponding author.
*E-mail addresses:* maurizio.boccia@unina.it (M. Boccia), adriano.masone@unina.it (A. Masone), claudio.sterle@unina.it (C. Sterle), teresa.murino@unina.it (T. Murino).

https://doi.org/10.1016/j.ejor.2022.10.023
0377-2217/

(Fazlollahtabar & Saidi-Mehrabad, 2015). However, most of the contributions on the topic frequently neglected the issues related to the battery depletion and recharge, as noticed in De Ryck, Versteyhe, & Shariatmadar (2020), Fatnassi & Chaouachi (2015) and McHaney (1995).

This work addresses the Automated Guided Vehicle Scheduling Problem with battery constraints (*ASP-BC*). The *ASP-BC* was firstly defined in Masone et al. (2021) to address a real problem of a manufacturing company implementing the Industry 4.0 paradigm for internal logistics (Mehami, Nawi, & Zhong, 2018). The manufacturing company aims to minimize the time to supply a set of packages to a set of workstations (i.e., the makespan) and uses an AGV system where AGVs are subject to battery constraints. In a nutshell, the AGV system performs as follows: a set of materials has to be moved to different workstations; the materials are organized in packages where each package contains all the materials of a single workstation; all the packages are located at the central warehouse; an AGV can be loaded with only one package on each trip; the AGV has to come back to the warehouse after delivering each package; the AGV battery consumption depends on the travel time and the weight carried; the AGV has to be recharged fully before its battery is completely depleted; the AGV charging time is fixed; the AGV charging stations are located at the central warehouse. The *ASP-BC* determines the AGV scheduling that minimizes the makespan, which also includes the AGV charging time. In this work, we present exact and heuristic approaches to tackle the *ASP-BC*. However, we also show how to adapt the proposed approaches to the case where the charging time depends on the residual energy.

In particular, the main contributions of this work can be summarized as follows:

1. We propose an original mixed-integer linear programming (MILP) model for the *ASP-BC*. In particular, we formulate the problem as a variant of the Bottleneck Generalized Assignment Problem (*BGAP*) (Martello & Toth, 1995). The BGAPs in the literature consider only assignment decisions. In our model, considering a round trip from the warehouse to the workstation as a job, two kinds of decisions arise: assignment of jobs to AGVs and sequencing of jobs on each AGV. In particular, the sequence of jobs impacts when recharging operations must occur on each AGV. The proposed formulation is more compact than other formulations proposed in the literature for problems similar to the *ASP-BC* since it implicitly considers the time dimension.
2. We develop a three-step matheuristic to solve large size instances of the *ASP-BC*. A bin-packing problem (*BPP*) with the aim of determining the minimum number of required battery charges is solved at the first step. The *BPP* solution determines a lower bound for the *ASP-BC* and subsets of jobs that can be grouped together without fully depleting the battery. Then, a feasible solution is obtained by solving a bottleneck generalized assignment problem in the second step. Finally, the solution is improved through a local search performing reassignment and swap moves of transfer jobs among available AGVs.
3. We solve instances built from real data provided by a manufacturing company. The results on different kinds of instances, in terms of AGV fleet size, number of jobs, job duration, and related energy consumption, show that the complexity of the *ASP-BC* is strongly related to the AGV energy consumption. Moreover, the computational experiments also show the effectiveness and the efficiency of the proposed approaches. The new formulation allows us to solve instances with up to 150 jobs and 10 AGVs outperforming the results of the formulation proposed in Masone et al. (2021). The matheuristic is able to solve instances with up to 200 jobs and 10 AGVs, with a limited computation burden and an average optimality gap lower than 1%.

The remainder of the paper is organized as follows: Section 2 contains an overview of the related literature. In Section 3, the *ASP-BC* features and assumptions, and the proposed *MILP* formulation are described. Section 4 presents the proposed matheuristic. The results of extensive computational tests to assess the performance of the proposed solution methods are reported in Section 5. Finally, Section 6 presents conclusions and gives future research directions on this topic.

## 2. Literature review

The literature on AGV scheduling problems (*ASPs*) is very rich as witnessed by the survey in Kaoud, El-Sharief, & El-Sebaie (2017), Qiu, Hsu, Huang, & Wang (2002) and Xie & Allen (2015). However, despite the large number of published papers, scarce attention has been given to the AGV battery depletion and recharge issues. To the best of authors' knowledge, the *ASP-BC* has been defined and tackled by a *MILP* formulation for the first time only in the recent work in Masone et al. (2021), where the authors demonstrate the impact of the AGV battery recharge times on the completion time of the overall material handling process.

For this reason, in this section, we discuss papers that can be considered as closely related to the *ASP-BC* even if they do not specifically consider AGVs battery depletion and/or charging operations. To this aim we extend our review to scheduling problems with maintenance activities since the battery charging operations of an AGV can be reckon as maintenance activities capable of restoring the vehicle initial utility conditions. For a complete review of production, maintenance, and resource scheduling problems, we address the interested reader to Geurtsen, Didden, Adan, Atan, & Adan (2022).

Scheduling problems with maintenance activities can be classified into "fixed" and "coordinated" problems according to Yoo & Lee (2016). In the first class of problems, jobs have to be scheduled coherently with planned maintenance activities, whose starting and completion times are known in advance. Instead, in the second class of problems, both jobs and maintenance activities with related starting times have to be scheduled. In the following, we focus on the second class of problems since, regarding the battery recharge as a maintenance activity, the *ASP-BC* can be considered as a variant of the "coordinated" case. "Fixed" problems are out of the scope of this work, thus we just address the interested reader to Avalos-Rosales, Angel-Bello, Álvarez, & Cardona-Valdés (2018), Fu, Huo, & Zhao (2011), Gedik, Rainwater, Nachtmann, & Pohl (2016), Liao & Sheen (2008), Mellouli, Sadfi, Chu, & Kacem (2009) and Wang & Cheng (2015). The contributions on "coordinated" problems are quite heterogeneous since they generally arise from real applications each of them entailing very specific features. However, we can further classify "coordinated" problems into two groups. The first group includes problems assuming that a single maintenance activity must be performed within each maintenance interval (i.e., a time window) regardless of the working time: a single machine scheduling problem is tackled in Graves & Lee (1999); a parallel machine scheduling problem with the aim of minimizing the job total weighted completion time is addressed in Lee & Chen (2000); a flow shop scheduling problem is studied in Aggoune (2004); a complexity study of the parallel machine scheduling problem considering different objective functions and/or the operating conditions is reported in Yoo & Lee (2016). The second group includes problems where machines cannot continuously operate longer than a pre-defined working time without performing a maintenance activity. Specifically, a single machine scheduling problem with release date is addressed in Cui & Lu (2017). A two identical machine scheduling problem aimed at minimizing the sum of the job completion time is tackled in Sun & Li (2010). A similar problem with more than two machines is investigated in Costa, Cappadonna, & Fichera (2016). The benefits of an integrated scheduling-preventive maintenance approach based on the minimization of the job tardiness and the machine failure rate are shown in Cassady & Kutanoglu (2003).

The *ASP-BC* clearly falls within the second group of "coordinated" problems, but it has several differentiating features. First, we highlight that the *ASP-BC* envisages a parallel machine scheduling problem aimed at minimizing the makespan instead of the completion time, as generally done in previously discussed literature. Moreover, the *ASP-BC* clearly stands out for another key feature. Maintenance activities are scheduled considering either the elapsed time from the beginning of the process or the time passed from the last performed maintenance. In other words, they are planned considering only the time dimension of the process. Instead, in the *ASP-BC* each job, depending on its features (e.g., carried weight), consumes a specific amount of energy. Thus, recharge activity cannot be planned only considering the time dimension, but the actual battery depletion.

Furthermore, from a methodological point of view, we underline that, even if there is a bunch of exact and heuristic solution methods dealing with scheduling problems with maintenance activities, such methods cannot be straightly used or adapted to solve the *ASP-BC*. Indeed, on one hand, the exact methods suffer from dimensional drawbacks which make their solution impracticable even on small instances (few tens of jobs and less than ten machines). On the other hand, proposed heuristic methods are designed to tackle very specific variants of the problem, so preventing their usage except for a significant rethinking of the method with no effectiveness guarantee.

On the basis of these considerations, the solution of the *ASP-BC* requires the development of ad-hoc exact and heuristic methods exploiting its own features.

In conclusion, we highlight that *ASP-BC* can appear as a variant of the AGV routing problems with battery constraints (Dang, Singh, Adan, Martagan, & van de Sande, 2021; Fatnassi & Chaouachi, 2015; Singh, Dang, Akcay, Adan, & Martagan, 2021). However, these two classes of problems are very different. Indeed, in the *ASP-BC*, given a subset of jobs assigned to a charging operation, the sequence of jobs does not affect the amount of energy consumed and, consequently, does not change the value of the objective function. On the contrary, job sequencing plays a crucial role in AGV routing problems. Moreover, unlike the ASP-BC, transfer jobs with AGV routing problems may have multiple pickup and drop-off points, and the energy expenditure depends only on the travel time. Finally, we point out that each AGV routing problem has its own distinguishing features that make these problems very different from each other. Therefore, for clarity, we provide a brief description of these problems in the following.

In Fatnassi & Chaouachi (2015), the problem involves finding a set of routes to perform the transfer jobs. The routes have to start and end at the charging location, and each route length does not have to exceed the AGV battery capacity. The authors assume an unlimited number of AGVs, and the objective is to minimize the sum of the route lengths. The problem is solved by a heuristic algorithm that is able to tackle instances of up to 100 jobs.

In Singh et al. (2021), each transfer job is also characterized by a time window. If the AGV arrives at a destination node after the latest delivery time of the time window, tardiness occurs. Each AGV also has a travel cost per time unit, and the AGV battery charge level decreases proportionally with the traversed distance. The AGV must recharge its battery if the charge level drops below a critical threshold, and the recharging duration should be long enough to allow the charge level to reach at least a fixed threshold. Moreover, an AGV cannot carry more than a single load. The aim is to determine the set of AGV routes that minimizes the weighted sum of the tardiness of the transfer jobs and travel costs of AGVs. The authors propose a *MILP* formulation able to optimally solve instances with up to 20 transfer jobs and 6 AGVs. Therefore, the authors integrate the *MILP* formulation within an adaptive large neighborhood search framework to solve instances of up to 30 transfer jobs and the same number of AGVs.

Finally, the study presented in Dang et al. (2021) extends the problem tackled in Singh et al. (2021) considering that AGVs can carry multiple loads simultaneously. The authors adapt the exact and heuristic approaches proposed in Singh et al. (2021) to tackle the extended problem. The exact approach can solve instances of up to 12 jobs and 3 AGVs. On the other hand, the heuristic approach can solve instances of up to 100 jobs and 9 AGVs. Based on this discussion, it is clear that the solution of AGV routing problems integrates scheduling decisions with routing decisions related to the movement of each job from a pick-up to a delivery point. Therefore, the related solution approaches are mainly focused on the routing aspects, which, as said above, are not considered in the *ASP-BC*. On this basis, AGV routing problems can be conceived as a particular case of the Electric Vehicle Routing Problems (*EVRPs*). For the sake of completeness, we address the interested reader on the *EVRP* literature to Bongiovanni, Kaspi, & Geroliminis (2019), Lin, Zhou, & Wolfson (2016) and Masmoudi, Hosny, Demir, Genikomsakis, & Cheikhrouhou (2018) for some of the most recent works and to Erdelić & Carić (2019) and Kucukoglu, Dewil, & Cattrysse (2021) for some surveys on the topic.

## 3. The AGV scheduling problem with battery constraints

In this work, we study the *ASP-BC* as defined in Masone et al. (2021). Hence, we consider the same AGV system features, assumptions, and operating conditions, which are a specification of the ones hinted in Section 1. In this section, we first describe the *ASP-BC*. Then, we present our original *MILP* formulation.

### 3.1. Problem description

The manufacturing company uses an AGV system for supplying the materials required for the production tasks to a set of workstations. The company aims at minimizing the time required to supply the materials to all the workstations so as to avoid delays in the production operations. For each workstation, the company organizes all the required materials within a single dedicated package (no splitting is possible). In other words, all the materials requested by a workstation are included in one package only and a package contains all the materials of one workstation only. Thus, a one-to-one matching between packages and workstations exists.

The packages are stored in a central warehouse, and they have to be carried to the workstations by a fleet of homogeneous AGVs that is initially located at the central warehouse. All the AGVs travel at a constant speed. Each AGV generally serves more than one workstation since the number of AGVs is much smaller than the number of workstations/packages. However, since each AGV can be loaded with a single package at a time, an AGV serving multiple workstations has to perform multiple round trips, consisting of a loaded trip from the warehouse to a workstation and the empty trip back. In the following, coherently with scheduling terminology, such trips are referred to as transfer jobs. Each transfer job is characterized by two parameters: *duration*, given by the sum of the round trip travel time and the time needed for the package loading and unloading operations; *weight*, given by the sum of the weight of the carried materials.

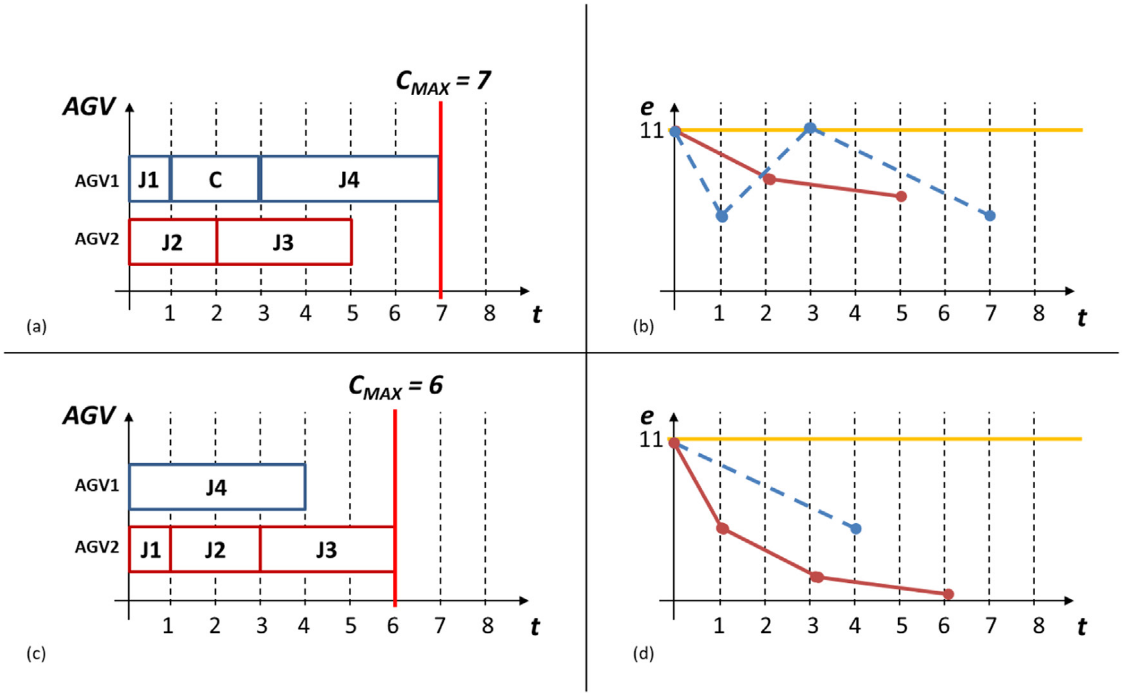


**Fig. 1.** Two *ASP-BC* solutions on a small instance with 4 jobs and 2 AGVs.

Then, each AGV is characterized by a battery with limited capacity, whose charge level decreases in function of the transfer job duration and carried weight. Thus, an AGV may need to visit a charging station to complete all the transfer jobs assigned to it. A set of charging stations is located at the central warehouse and a charging operation has to be performed before the AGV battery is completely depleted. A charging operation fully recharges the AGV battery and the charging time is fixed and does not depend on the residual energy. This assumption is sustained by the manufacturer experience and practice with respect to the real problem under investigation. However, we highlight that this assumption can be further sustained considering the battery life cycle and the characteristics of the charging process. Indeed, it is important to minimize the number of charges since a limited number of charge cycles characterizes a battery. This can be achieved by performing full-charge operations. Furthermore, charging time represents an overhead for the transfer process. On this basis, it is quite natural to perform as many jobs as possible, to get near to full battery depletion, and then charge a battery. Therefore, the charging time can be considered fixed when the battery is almost depleted. Finally, we highlight that transfer jobs are generally characterized by small energy consumption values with respect to battery capacity. This leads to solutions performing as many jobs as possible on an AGV battery charge, so pushing towards full battery depletion.

On this basis, it is possible to assume a charging operation as a special job with a duration equal to the charging time and null weight. This assumption forms the basis of the *MILP* formulation described in the next section.

The *ASP-BC* consists in determining the scheduling of transfer job and charging operations on an AGV fleet such that the transfer process makespan is minimized. Such makespan, as in parallel machine scheduling problem, is defined as the maximum completion time of the last transfer job over all the AGVs. Therefore, the *ASP-BC* solution involves assignment decisions (matching jobs to machines) and scheduling decisions (sequencing jobs on each machine).

**Table 1**
Job duration and energy consumption of a small instance with 4 jobs and 2 AGVs.

| Job ID | Time | Energy |
|---|---|---|
| J1 | 1 | 6 |
| J2 | 2 | 3 |
| J3 | 3 | 1 |
| J4 | 4 | 6 |

Two solutions of the *ASP-BC* on a small instance with 4 transfer jobs and 2 AGVs are reported in Fig. 1 to better explain the problem. The job duration in time units and the related consumption in terms of energy units are reported in Table 1. Each AGV has a battery capacity equal to 11 energy units and a charging operation requires 2 time units. Fig. 1(a) shows a feasible *ASP-BC* solution requiring a charging operation for the AGV1, whose corresponding makespan is equal to 7 time units. We highlight that if we had neglected the battery constraints, the scheduling considered in this solution would have produced the optimal solution of the resulting parallel machine scheduling problem with a makespan equal to 5 time units. Fig. 1(b) shows the battery level of the two AGVs throughout the planning horizon, where the dashed and the continuous lines correspond to AGV1 and AGV2, respectively. We can observe that if the battery of AGV1 had not been charged after job J1, then it would have been completely depleted during the execution of job J4. Fig. 1(c) shows the optimal *ASP-BC* solution requiring no charging operations, whose corresponding makespan is equal to 6 time units. Fig. 1(d) shows the battery level in the optimal solution of the available AGVs.

The provided example allows to better understand the difference in terms of makespan minimization between the majority of the parallel machine scheduling problems and the *ASP-BC*. Indeed, in the first case, being battery issues neglected, the makespan of each machine is computed as the sum of the durations of the jobs assigned to it, independently of the sequence. Instead, in the second case, given a jobs-to-machine assignment, each job sequence provides a different battery level trend and consequently a different need of charging operations. Such operations represent an overhead that has to be added to the makespan of each machine.

### 3.2. Problem formulation

On the basis of the previous discussion, we develop an assignment-based formulation for the *ASP-BC* which implicitly considers the sequence dimension.

In particular, in the following, we present an original *MILP* formulation for the *ASP-BC* based on the Bottleneck Generalized Assignment Problem (*BGAP*) and therefore named *A-BGAP*.

The *BGAP* is the min-max version of the generalized assignment problem. Given $n$ items and $m$ units, the penalty, $p_{ij}$, and the resource requirement, $q_{ij}$, corresponding to the assignment of item $j$ to unit $i$ ($j = 1, .., n, i = 1, .., m$), and the amount of resource $a_i$ available at unit $i$ ($i = 1, .., m$), the *BGAP* is to assign each item to one unit so that the total resource requirement for any unit does not exceed its availability and the maximum penalty incurred is minimized (Martello & Toth, 1995).

In our case, the *ASP-BC* can be conceived as a variant of the *BGAP* where items and units are represented by jobs and AGV charging operations, respectively. On this basis, the resource requirement of an item is the energy consumption of a transfer job, and the resource is the battery capacity. However, the penalty depends on the subset of jobs performed on an AGV. In particular, it can be computed as the sum of the duration of the transfer and charging jobs. Therefore, unlike the *BGAP*, the *ASP-BC* requires a double level of assignment: a first level where transfer jobs are assigned to the charging operations; a second level where the charging operations and the related transfer jobs are assigned to the AGVs. On this basis, let $J$ be the set of transfer jobs, where each job $j, j \in J$, has an energy consumption $e_j$ and a processing time $d_j$ independent from the potential AGV managing its transfer. Let $M$ be the set of AGVs, where each AGV $m, m \in M$, has a battery capacity equal to $b$. Finally, being $n$ the number of charging operations, we define $R = \{1, \dots, n\}$ as the set of the charging operations. In the worst case, when all the job duration assume values near to the battery capacity, the AGV batteries should be charged after the execution of each transfer job. Thus, we set the value of $n$ equal to the number of transfer jobs ($n = |J|$). As previously discussed, the charging operations can be considered as special jobs that have to be assigned to the AGVs likewise transfer jobs. The charging time of all the charging operations is equal to $t$. However, the charging time of the first charging operation on each AGV can be neglected since AGVs are completely charged at the beginning of the transfer operations.

On the basis of this notation, we introduce the following decision variables for the mathematical formulation:

- $x_j^m \in \{0, 1\}$: equal to 1 if transfer job $j, j \in J$, is performed by AGV $m, m \in M$, 0 otherwise.
- $q_r^m \in \{0, 1\}$: equal to 1 if charging job $r, r \in R$, is performed by AGV $m, m \in M$, 0 otherwise.
- $y_{jr}^m \in \{0, 1\}$: equal to 1 if job $j, j \in J$, is performed by AGV $m, m \in M$, after the charging job $r, r \in R$, and before the next one (if a next charging job on the AGV $m$ exists).
- $C_{\max} \in \mathbb{Z}^+$: completion time of the transfer process.

On this basis, we can introduce the A-BGAP formulation:

$$z = \min C_{max}$$

$$C_{max} \geq \sum_{j \in J} d_j x_j^m + \sum_{r \in R \setminus \{1\}} t q_r^m \qquad \forall m \in M \tag{1}$$

$$\sum_{m \in M} x_j^m = 1 \qquad \forall j \in J, \tag{2}$$

$$\sum_{r \in R} y_{jr}^m = x_j^m \qquad \forall j \in J, m \in M, \tag{3}$$

$$2y_{jr}^m \leq x_j^m + q_r^m \qquad \forall j \in J, m \in M, r \in R, \tag{4}$$

$$\sum_{j \in J} e_j y_{jr}^m \leq b \qquad \forall m \in M, r \in R, \tag{5}$$

$$q_r^m \leq q_{r-1}^m \qquad \forall m \in M, r \in R \setminus \{1\}, \tag{6}$$

$$q_1^m = 1 \qquad \forall m \in M, \tag{7}$$

$$x_j^m \in \{0, 1\} \qquad \forall m \in M, j \in J, \tag{8}$$

$$q_r^m \in \{0, 1\} \qquad \forall m \in M, r \in R \setminus \{1\}, \tag{9}$$

$$y_{jr}^m \in \{0, 1\} \qquad \forall j \in J, m \in M, r \in R, \tag{10}$$

$$C_{max} \in \mathbb{R}^+ \tag{11}$$

The objective function minimizes the makespan of the whole transfer process. Constraints (1) ensure the consistency between the duration of the transfer process of each AGV and the makespan. We point out that the duration of the transfer process on each AGV is given by the sum of the duration of the performed transfer jobs and the required charging operations. Constraints (2) requires that each transfer job must be assigned to a single AGV. According to constraints (3), if a transfer job $j$ is assigned to an AGV $m$ then at least a charging job $r$ foregoing job $j$ must be on the same AGV. Constraints (4) are variable upper bounds on the variables $y_{jr}^m$. Constraints (5) are battery capacity constraints for each charging job. Constraints (6) are symmetry breaking constraints. Constraints (7) set the first charging job on each AGV equal to 1 since all the AGVs are supposed to be fully charged at the starting time. Constraints (8–11) define the nature of the variables.

For the sake of completeness, we highlight that the proposed model could be easily modified to consider a charging time that depends on the residual energy. In particular, if we consider a parameter $\tau$ which indicates the charging time per energy unit, then it is possible to modify constraints 1 as follows:

$$C_{max} \geq \sum_{j \in J} d_j x_j^m + \tau \sum_{j \in J} \sum_{r \in R \setminus \{n\}} e_j y_{jr}^m q_{r+1}^m \qquad \forall m \in M \tag{12}$$

The first term on the right-hand side represents the transfer time, while the second term corresponds to the time of each charging operation which depends on the energy expenditure. We point out that constraints (12) are non linear since they present a sum of products of binary variables ($y_{jr}^m q_{r+1}^m$), that can be easily linearized (Asghari, Fathollahi-Fard, Mirzapour Al-e hashem, & Dulebenets, 2022). These products are necessary to ignore the charging time of the last charging operation since AGVs are charged during the night for the transfer operations of the following day.

## 4. A matheuristic for the *ASP-BC*

In this section, we present a matheuristic based on the “natural” decomposition of the *A-BGAP* into its two assignment levels previously mentioned. The matheuristic consists in three steps where in the first two steps the two assignment levels are solved sequentially determining a lower and an upper bound, respectively. Then, the solution obtained is improved through a local search based on the reassignment and swap of the transfer jobs in the last step. In the following, a detailed description of each step is provided.

### 4.1. Step 1: transfer job-charging operation assignment

The first step determines the assignment of the transfer jobs to the charging operations. This assignment is obtained through the solution of a one-dimensional Bin Packing Problem (*BPP*) (Wei, Luo, Baldacci, & Lim, 2020), where the bins are the charging operations, and the items are the transfer jobs. As in the *A-BGAP* formulation, let $J$ and $R$ be the sets of transfer jobs and charging operations, respectively. In this setting, let $\gamma_r, r \in R,$ be a binary variable equal to 1 if the $r$-th charging operation is performed, 0 otherwise, and let $\chi_{rj}, r \in R, j \in J,$ be a binary variable equal to 1 if the transfer job $j$ is assigned to the $r$-th charging operation, 0 otherwise. On the basis of this notation, the resulting *BPP* formulation is the following:

$$\zeta = \min \sum_{r \in R} \gamma_r$$

$$\sum_{r \in R} \chi_{rj} = 1 \qquad \forall j \in J, \tag{13}$$

$$\sum_{j \in J} e_j \chi_{rj} \le b\ \gamma_r \qquad \forall r \in R, \tag{14}$$

$$\gamma_r \le \gamma_{r-1} \qquad \forall r \in R \setminus \{1\}, \tag{15}$$

$$\chi_{rj} \in \{0, 1\} \qquad \forall r \in R,\ \ j \in J, \tag{16}$$

$$\gamma_r \in \{0, 1\} \qquad \forall r \in R, \tag{17}$$

The objective function minimizes the number of performed charging operations. Constraints (13) guarantee that each job is assigned to a charging operation. Constraints (14) ensure that if a charging operation is used, then the sum of the energy consumption of the jobs assigned to it is lower than the battery capacity. Constraints (15) are symmetry breaking constraints. Finally, constraints (8–11) express the integrality of the variables. We highlight that if $\bar{\zeta}$ is the value of the optimal solution of the previous *BPP*, then $\max(0, \bar{\zeta} - |M|)$ is the minimal number of required charging operations to process all the job since the AGVs are fully charged at the beginning of the operations. On the basis of this observation, a valid lower bound for the *ASP-BC* can be computed as follows:

$$LB = \max\left\{ \left\lceil \frac{[\max(0, \bar{\zeta} - |M|)] * t + \sum_{j \in J} d_j}{|M|} \right\rceil, \left\lceil \frac{[\max(0, \bar{\zeta} - |M|)]}{|M|} \right\rceil * t \right\}.$$

The numerator of the first term, $[\max(0, \bar{\zeta} - |M|)] * t + \sum_{j \in J} d_j$, represents the minimum total time required by all the AGVs to perform all the charging and transfer operations. Instead, the second term allows taking into account the indivisibility of the charging operations in the computation of the charging time.

Being charging operations scheduled only to avoid that AGV batteries are completely depleted, one might expect that the *ASP-BC* optimal solution uses the minimum number of charging operations. However, the following counterexample with 2 AGVs and 3 jobs shows that an *ASP-BC* optimal solution could use a number of charging operations greater than $\bar{\zeta}$. Let us consider two AGVs with a charging time of 10 time units and a battery capacity of 10 energy units. Moreover, let the job duration and energy consumption be $D = \{25, 5, 5\}$ and $W = \{3, 5, 6\}$, respectively. The minimum number of charging operations is equal to two since it is possible to pack into one charging operation jobs J1 and J2 and into a second charging operation the job J3. The solution resulting from this assignment is shown in Fig. 2(a). The makespan is equal to 30 time units given by the sum of the duration of jobs J1 and J2. Fig. 2(b) shows the *ASP-BC* optimal solution with a makespan equal to 25, where 3 charging operations, one for each job, are scheduled.

### 4.2. Step 2: charging operation-AGV assignment

The solution of the *BPP* is used as input in the second step of the matheuristic, where a first feasible solution for the *ASP-BC* is determined. Specifically, let $\bar{\zeta}$ be the optimal objective function value of the previously defined *BPP*. If $\bar{\zeta} \le |M|$, then a feasible solution for the *ASP-BC* without charging operations exists. On the other hand, if $\bar{\zeta} > |M|$, then at least one charging operation has to be performed to obtain an *ASP-BC* feasible solution. For each case, we set up a different method to determine a feasible *ASP-BC* solution. In the following, each method is described in detail.

#### 4.2.1. ASP-BC solution without charging operations

In this case, we can determine an *ASP-BC* solution setting up a *BGAP* with resource constraints (*BGAP-C*) where the items are the transfer jobs and the bins are the AGVs. The *BGAP-C* solution corresponds to the best *ASP-BC* solution without charging operations (not necessarily the optimal solution of the *ASP-BC*, as shown in the previous example). On the basis of the same notation used for the *A-BGAP*, it is possible to formulate the *BGAP-C* as follows:

$$z = \min C_{max}$$

$$C_{max} \ge \sum_{j \in J} d_j x_j^m, \qquad m \in M, \tag{18}$$

$$\sum_{m \in M} x_j^m = 1 \qquad j \in J, \tag{19}$$

$$\sum_{j \in J} e_j x_j^m \le b \qquad m \in M, \tag{20}$$

$$x_j^m \in \{0, 1\} \qquad j \in J, m \in M, \tag{21}$$

$$C_{max} \in \mathbb{R}^+ \tag{22}$$

The objective function minimizes the makespan. Constraints (18) guarantee the consistency of the makespan and the duration of the transfer jobs. Constraints (19) ensure that all the transfer jobs are performed. The maximum energy consumption on each AGV is guaranteed by constraints (20). On the basis of some preliminary experiments, we highlight that this case (i.e., $\bar{\zeta} \le |M|$) might arise when the number of jobs is relatively small compared to the number of AGVs and/or the job energy consumption are relatively small compared to the battery capacity.

#### 4.2.2. ASP-BC solution with charging operations

In this case, we exploit the *BPP* solution setting up a *BGAP* which assumes the charging operations containing the transfer jobs as items and the AGVs as bins (*BGAP-R*). To this end, let $(\overline{\chi}, \overline{\gamma})$ be the *BPP* optimal solution, we define $\overline{R}$ as the smallest subset of charging jobs in a feasible solution ($\overline{R} = \{r \in R : \overline{\gamma}_r = 1\}$). Each charging job $\bar{r}, \bar{r} \in \overline{R}$, is characterized by a duration $D_{\bar{r}}$ given by the

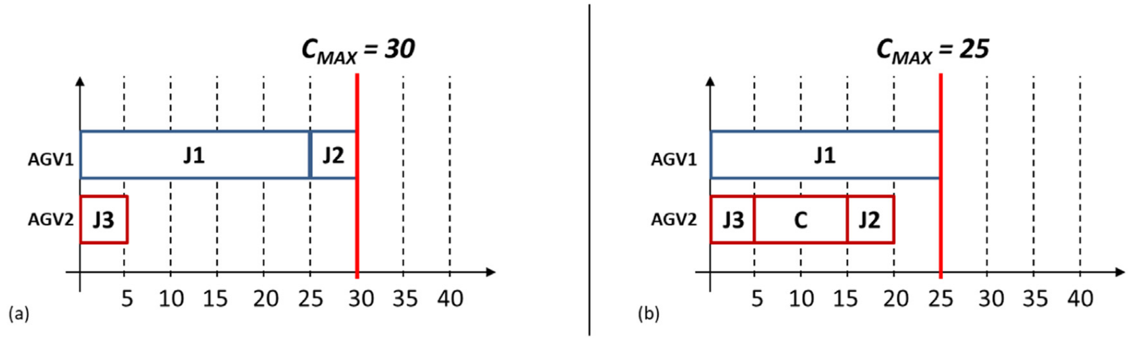


**Fig. 2.** Two *ASP-BC* solutions on a small instance with 3 jobs and 2 AGVs.

total processing time of all the jobs assigned to that charging operation and the charging time, $D_{\bar{r}} = (\sum_{j\in J} d_j \overline{\chi}_{\bar{r}j}) + t$. On the basis of this notation, we set up the following *BGAP-R* to determine a feasible solution for the *ASP-BC*:

$$z = \min C_{max}$$

$$C_{max} \geq (\sum_{\bar{r}\in\overline{R}} D_{\bar{r}}\theta_{\bar{r}m}) - t, \qquad \forall m \in M, \tag{23}$$

$$\sum_{m\in M} \theta_{\bar{r}m} = 1 \qquad \forall \bar{r} \in \overline{R}, \tag{24}$$

$$\theta_{\bar{r}m} \in \{0, 1\}, \qquad \forall \bar{r} \in \overline{R},\ m \in M, \tag{25}$$

$$C_{max} \in \mathbb{R}^+ \tag{26}$$

where $\theta_{\bar{r}m}$ is a binary variable equal to 1 if the charging operation/bin $\bar{r}$ is assigned to AGV $m$, 0 otherwise. The objective function minimizes the makespan. The set of constraints (23) ensures the consistency between the duration of the transportation and charging operations among the AGVs and the makespan. We point out that we subtract the charging time to the right hand side of constraints (23) due to the AGV initial battery charge. The set of constraints (24) guarantees that all the bins are assigned to an AGV. Finally, the sets of constraints (25,26) express the nature of the variables.

### 4.3. Step 3: local search

The local search explores the neighborhood of a given solution obtained considering three moves that can be considered as the well-known add, swap, and remove local search operations. The add operation consists in moving a transfer job from a charging operation of an AGV to a new (additional) charging operation of a different AGV. The remove operation consists in moving a transfer job from a charging operation of an AGV to an already scheduled charging operation of a different AGV. Finally, the swap operation consists in replacing a transfer job of an AGV with a transfer job of a different AGV. In the following, we first describe the three operations and then we provide the implementation details of the local search.

*First move – add operation*

Let $(\hat{x}, \hat{y})$ and $\hat{C}_{\max}$ be the solution obtained at the end of the second step and the corresponding objective function value, respectively. We define $\hat{c}_m$ as the duration of the transfer process on the $m$-th AGV ($\hat{c}_m = \sum_{j\in J} d_j \hat{x}^m_j + \sum_{r\in R\setminus\{1\}} t\hat{q}^m_r$). Moreover, let $\hat{J}_m$ be the set of transfer jobs assigned to the $m$-th AGV in the current solution ($\hat{J}_m = \{j \in J : \hat{x}^m_j = 1\}$).

On this basis, we define $s_a(m_1, j, m_2)$ as the saving which arises adding a job $j, j \in \hat{J}_{m_1}$, from the AGV $m_1, m_1 \in M$, to a new charging operation of the AGV $m_2, m_2 \in M \setminus \{m_1\}$. Then, the formal expression of $s_a(m_1, j, m_2)$ is the following:

$$s_a(m_1, j, m_2) = \max\{0, \hat{C}_{max} - \max_{m\in M\setminus\{m_1,m_2\}}\{\hat{c}_{m_1} - d_j, \hat{c}_{m_2} + d_j + t, \hat{c}_m\}\}.$$

We point out that if a job $j^*$ is the only one assigned to a charging operation $r^*$ of the AGV $m^*_1$ (i.e., $y^{m^*_1}_{j^*r^*} = 1 \wedge \sum_{i\in J_{m^*_1}\setminus\{j^*\}} y^{m^*_1}_{ir^*} = 0$) then the expression of the new duration of the transfer process on the $m^*_1$-th AGV is equal to $\hat{c}_{m^*_1} - d_j - t$. Finally, we would like to highlight that this move is developed to consider the case in which the optimal solution uses a number of charging operations greater than the minimum one.

*Second move - Remove operation*

Let $\hat{R}_m$ be the set of charging operations assigned to the $m$-th AGV in the current solution ($\hat{R}_m = \{r \in R : \hat{q}^m_r = 1\}$). Moreover, let $\hat{b}^m_r$ be the energy required by the transfer jobs assigned to the $r$-th charging operation of the $m$-th AGV (i.e., $\hat{b}^m_r = \sum_{j\in\hat{J}_m} e_j \hat{y}^m_{jr}$).

On this basis, we define $s_r(m_1, j, m_2, r)$ as the saving which arises removing a job $j, j \in \hat{J}_{m_1}$, from the AGV $m_1, m_1 \in M$, and assigning it to the charging operation $r, r \in \hat{R}_{m_2}$ of the AGV $m_2, m_2 \in M \setminus \{m_1\}$, without exceeding the battery capacity (i.e., $b - \hat{b}^m_r \geq e_j$). Then, the formal expression of $s_r(m_1, j, m_2, r)$ is the following:

$$s_r(m_1, j, m_2, r) = \max\{0, \hat{C}_{max} - \max_{m\in M\setminus\{m_1,m_2\}}\{\hat{c}_{m_1} - d_j, \hat{c}_{m_2} + d_j, \hat{c}_m\}.$$

Also in this case, if a job $j^*$ is the only one assigned to a charging operation $r^*$ of the AGV $m^*_1$, then the expression of the new duration of the related transfer process has to be modified accordingly.

*Third move – Swap operation*

We define $s_s(m_1, r_1, j_1, m_2, r_2, j_2)$ as the saving which arises swapping a job $j_1, j_1 \in \hat{J}_{m_1}$, from the charging operations $r_1 \in \hat{R}_{m_1}$ of the AGV $m_1, m_1 \in M$, with the job $j_2, j_2 \in \hat{J}_{m_2}$, from the charging operations $r_2 \in \hat{R}_{m_2}$ of the AGV $m_2, m_2 \in M \setminus \{m_1\}$, without exceeding the battery capacity of the two AGVs (i.e., $b - \hat{b}^{m_1}_{r_1} \geq e_{j_2} - e_{j_1} \wedge b - \hat{b}^{m_2}_{r_2} \geq e_{j_1} - e_{j_2}$). Then, the formal expression of $s_s(m_1, r_1, j_1, m_2, r_2, j_2)$ is the following:

$$s_s(m_1, r_1, j_1, m_2, r_2, j_2) =$$

$$= \max\{0, \hat{C}_{max} - \max_{m\in M\setminus\{m_1,m_2\}}\{\hat{c}_{m_1} - d_{j_1} + d_{j_2}, \hat{c}_{m_2} + d_{j_1} - d_{j_2}, \hat{c}_m\}\}$$

*Implementation details*

Let $S$ be the solution obtained at the end of the second step, the proposed local search computes the maximum among the three kinds of savings previously described at each iteration. In particular, at the beginning of the local search, we set the maximum saving $s^*$ equal to 0. Then, we compute the add, remove and swap savings in this order. Each time a saving is computed, if it is greater

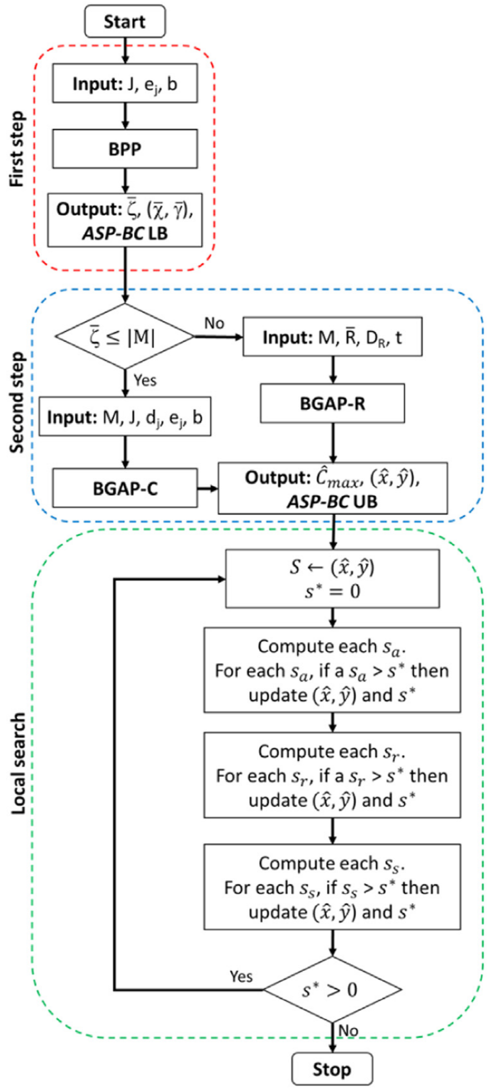


**Fig. 3.** Flow-chart of the proposed matheuristic.

than $s^*$ we save the information about the corresponding move and we update the value of $s^*$. Once all the possible savings are computed, if $s^* > 0$ (i.e., it is possible to reduce the makespan), then $S$ is modified accordingly to the corresponding move and the local search is iterated again. If $s^* \leq 0$, the local search stops.

For the sake of completeness, we point out that the local search could compute the savings for each couple of AGVs $m_1, m_2 \in M : m_1 \neq m_2$. However, it is possible to reduce the computational effort obtaining the same result considering only the job assigned to the AGV which corresponds to the largest makespan. Indeed, it is clear that it possible to determine a saving (i.e., a makespan reduction) only through the decrease of the completion time of the most loaded AGV. Therefore, at the beginning of each iteration of the local search, we identify the most loaded AGV $m^*$, $m^* = \arg\max_{m \in M} \hat{c}_m$. Then, we compute only the savings with $m_1$ equal to $m^*$.

For the sake of clarity, a flow-chart of the whole matheuristic is presented in Fig. 3.

Finally, for the sake of completeness, we point out that also the matheuristic approach can be adapted for the case of variable charging time. In particular, if we consider a charging time for energy unit, $\tau$, different modifications have to be performed to all the steps. Therefore, in the following, we discuss only the main modifications that have to be performed.

Concerning the first step, the lower bound expression has to be modified as follows:

$$LB = \left\lceil \frac{\sum_{j \in J}(d_j + \tau e_j)}{|M|} - \tau b \right\rceil,$$

where the numerator of the first term represents the total time required by all the machine to perform all the transfer jobs and charging operations with variable time. The second term is required for considering the initial AGV charge. Given the new lower bound expression, it is clear that there is no need to solve the *BPP* to determine it. However, its solution is still necessary to determine a first assignment of transfer jobs to charging jobs.

Concerning the second step, the expression of the charging job duration has to be modified to take into account the processing time of the jobs and the variable charging time ($D_{\bar{r}} = \sum_{j \in J}(d_j + \tau e_j)\overline{\chi}_{\bar{r}j}$). Then, constraints (23) have to be adapted to ignore the charging time of the last charging job on each AGV.

Finally, concerning the final step, the savings expression must be modified to consider the variable charging time.

## 5. Computational experiments

In this section, we present and discuss the computational results of the experimentation performed to evaluate and validate the proposed *A-BGAP* formulation and the matheuristic approach (*3S-MHA* in the following). The experiments have been performed on an Intel(R) Core(TM) i7-6500U, 2.50 GHz, 8.00 GB of RAM. The *3S-MHA* has been coded in Python language. The *A-BGAP*, the *BPP*, the *BGAP-C* and the *BGAP-R* formulations are solved using Gurobi 9.1. We set a time limit equal to 1800 s for the *A-BGAP* formulation. On the other hand, we set the time limit for all the other formulations within the *3S-MHA* to 720 s.

In the following, we first describe the problem instances built from real data coming from a manufacturing company. Next, we show the results of the *A-BGAP* formulation and the *3S-MHA* on small and medium instances (with up to 150 jobs and 10 AGVs). Moreover, we provide the results of the *3S-MHA* on a set of large instances (200 jobs and with up to 10 AGVs). Finally, we show the results of a comparison with other solution approaches for the *ASP-BC*. For the sake of completeness, we highlight that all the instances and the detailed results are given in Boccia, Masone, Sterle, & Murino (2021).

### *5.1. Test instances*

The manufacturing company provided us with a set of few real *ASP-BC* instances of different size (from few dozens to almost two hundreds jobs and from 2 to 10 AGVs). Starting from this information, we generated a larger set of instances considering the same features of the real instances. In particular, we considered instances with $|M|$ equal to 2, 5, 10 and $|J|$ equal to 50, 100, 150, and 200. Then, the values of the duration and energy consumption were generated using normal distributions derived from data provided by the company. Job duration depends on the travel time and the time for the loading and unloading operations. In particular, the time required for these operations depends on several factors, such as package position within the warehouse, type of workstation, and package weight and volume. Therefore, the set of job duration is very heterogeneous. On this basis, we generated the job processing times, $(d_j)$, following a normal distribution $\mathcal{N}(\mu_d, (\mu_d/2)^2)$ with three different average values suggested by the company ($\mu_d \in \{10, 20, 30\}$). In this way, the average duration of an *AGV* round trip from the central warehouse to the working station is about 10, 20 and 30 minutes, as in the

**Table 2**
Exact and matheuristic approach results on the instances with 50 jobs.

| | Exact approach | | Matheuristic approach | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Instances** | **%Gap** | **Time** | **Time Step 1** | **Initial %Gap** | **Time Step 2** | **Final %Gap** | **Time Step 3** | **Total Time** | **Best %Gap** |
| **M2_J50_d10_E1** | 0,00 | 3,90 | 0,21 | 1,17 | 0,01 | 0,02 | 0,00 | 0,22 | 0,02 |
| **M2_J50_d10_E2** | 0,00 | 10,85 | 0,24 | 0,00 | 0,01 | 0,00 | 0,00 | 0,25 | 0,00 |
| **M2_J50_d10_E4** | 0,00 | 14,78 | 0,49 | 0,00 | 0,01 | 0,00 | 0,00 | 0,50 | 0,00 |
| **M2_J50_d20_E1** | 0,00 | 3,76 | 0,20 | 24,28 | 0,02 | 1,51 | 0,00 | 0,22 | 0,02 |
| **M2_J50_d20_E2** | 0,00 | 28,61 | 0,28 | 0,00 | 0,01 | 0,00 | 0,00 | 0,29 | 0,00 |
| **M2_J50_d20_E4** | 0,44 | 425,94 | 72,54 | 0,23 | 0,01 | 0,23 | 0,00 | 72,56 | 0,00 |
| **M2_J50_d30_E1** | 0,00 | 5,76 | 0,20 | 0,39 | 0,01 | 0,03 | 0,00 | 0,22 | 0,03 |
| **M2_J50_d30_E2** | 0,00 | 17,52 | 0,23 | 0,01 | 0,01 | 0,00 | 0,00 | 0,24 | 0,00 |
| **M2_J50_d30_E4** | 0,57 | 597,24 | 72,79 | 0,18 | 0,01 | 0,18 | 0,00 | 72,80 | 0,00 |
| **M5_J50_d10_E1** | 0,00 | 5,70 | 0,21 | 33,76 | 0,01 | 1,33 | 0,01 | 0,23 | 1,33 |
| **M5_J50_d10_E2** | 0,00 | 11,04 | 0,23 | 9,68 | 0,04 | 0,51 | 0,01 | 0,27 | 0,51 |
| **M5_J50_d10_E4** | 1,22 | 1092,78 | 73,79 | 0,35 | 0,17 | 0,35 | 0,00 | 73,96 | 0,35 |
| **M5_J50_d20_E1** | 0,00 | 6,97 | 0,19 | 24,28 | 0,02 | 1,51 | 0,00 | 0,21 | 1,51 |
| **M5_J50_d20_E2** | 0,00 | 25,37 | 0,22 | 2,41 | 0,05 | 0,34 | 0,00 | 0,27 | 0,34 |
| **M5_J50_d20_E4** | 0,61 | 851,94 | 72,67 | 0,26 | 0,18 | 0,26 | 0,00 | 72,86 | 0,26 |
| **M5_J50_d30_E1** | 0,00 | 8,76 | 0,20 | 29,87 | 0,01 | 1,57 | 0,01 | 0,21 | 1,57 |
| **M5_J50_d30_E2** | 0,02 | 294,33 | 0,27 | 2,36 | 0,05 | 0,42 | 0,01 | 0,32 | 0,42 |
| **M5_J50_d30_E4** | 1,02 | 1002,00 | 72,47 | 0,24 | 0,28 | 0,24 | 0,00 | 72,75 | 0,24 |
| **M10_J50_d10_E1** | 0,00 | 1,81 | 0,30 | 0,00 | 0,00 | 0,00 | 0,00 | 0,30 | 0,00 |
| **M10_J50_d10_E2** | 0,00 | 146,43 | 0,21 | 54,14 | 0,02 | 22,02 | 0,01 | 0,24 | 10,57 |
| **M10_J50_d10_E4** | 1,51 | 1452,01 | 3,25 | 10,94 | 73,45 | 8,88 | 0,01 | 76,70 | 6,39 |
| **M10_J50_d20_E1** | 0,00 | 2,26 | 0,42 | 0,00 | 0,00 | 0,00 | 0,00 | 0,42 | 0,00 |
| **M10_J50_d20_E2** | 0,34 | 860,70 | 0,22 | 51,46 | 0,02 | 20,55 | 0,01 | 0,25 | 20,37 |
| **M10_J50_d20_E4** | 1,38 | 1455,32 | 1,76 | 4,15 | 17,32 | 2,85 | 0,00 | 19,09 | 2,85 |
| **M10_J50_d30_E1** | 0,00 | 2,36 | 0,62 | 0,00 | 0,00 | 0,00 | 0,00 | 0,62 | 0,00 |
| **M10_J50_d30_E2** | 0,44 | 1072,95 | 0,25 | 35,35 | 0,03 | 10,79 | 0,01 | 0,28 | 10,79 |
| **M10_J50_d30_E4** | 1,62 | 1715,45 | 0,90 | 3,05 | 3,22 | 2,43 | 0,00 | 4,12 | 2,43 |
| **Average** | 0,34 | 411,72 | 13,90 | 10,69 | 3,52 | 2,82 | 0,00 | 17,42 | 2,22 |

real instances. Similarly, the energy consumption depends mainly on the travel time and package weight, but it is also affected by other factors (e.g., package volume). Therefore, the job energy consumption, $(e_j)$, are generated following a normal distribution $\mathcal{N}(\mu_e, (\mu_e/2)^2)$, with $\mu_e \in \{1, 2, 4\}$. Then, we set the battery capacity $b$ equal to 10. The values of the energy consumption and battery capacity were chosen, according to some guidelines provided by the company, to realize three scenarios: an AGV can perform about 10 jobs ($\mu_e = 1$), 5 jobs ($\mu_e = 2$), and 2–3 jobs ($\mu_e = 4$). We point out that each job duration and energy consumption are generated independently. This choice is motivated by the fact that our aim is to represent the highly heterogenous situations tackled by the company (e.g., all the combinations of long/short travel time with a light/heavy package). However, we highlight that preliminary experiments showed that the performance of the proposed approaches is not affected by any correlation between job duration and energy consumption. The charging time $t$ of each machine is equal to 1 h based on the technical specification of the AGVs used by the company. The charging time and the battery capacity are assumed to be the same for all the AGVs. Finally, for each combination of $|M|$, $|J|$, $\mu_d$, and $\mu_e$, we generated a set of 10 instances so obtaining 1080 instances. In particular, given a certain combination of $|M|$, $|J|$ and $\mu_d$, we fix the values of the processing times, and we generate the energy consumption according to the three distributions of $\mu_e$, so obtaining three instances with the same processing times but with different energy consumption.

### 5.2. Results on small and medium instances

First, we carry out a comparative evaluation of the two solution methods using the set of instances with up to 150 transfer jobs. In particular, we summarize the percentage optimality gap (%Gap) and computation time of the exact approach (i.e., the *A-BGAP* formulation) and of each step of the *3S-MHA* on the instances with 50, 100, and 150 jobs inTables 2–4, respectively (each row gives an average value calculated over 10 instances). The *A-BGAP* percentage gap is calculated considering the lower bound and the upper bound provided by Gurobi. Therefore, if the relaxation is not solved within the time limit, then the lower bound will be the one of the linear relaxation, and the upper bound will be the one computed by the general-purpose heuristic implemented in Gurobi. The matheuristic initial and final percentage gaps correspond to the gap after the second step and the local search, respectively. They are computed considering the lower bound determined in the first step. Moreover, we recall that it is not possible to compute a percentage gap related to the first step of the *3S-MHA* since it determines only a valid lower bound for the *ASP-BC*. Finally, for the *3S-MHA*, we report the best percentage gap computed considering the solution obtained by the *3S-MHA* and the largest lower bound value between the one obtained by the *A-BGAP* and the one of the first step of the matheuristic.

In Table 2, we can observe that the exact approach is able to solve to optimality almost all the instances with an average job energy consumption lower than or equal to 2. On the other hand, the instances with an average consumption equal to 4 result the most difficult ones. Nevertheless, the maximum average gap on these instances is lower than 2%. These results hint that the *ASP-BC* complexity is strictly related to the battery constraints. Specifically, the *ASP-BC* complexity increases when the average job consumption increases since more charging operations are required to obtain a feasible solution. The running times of the *A-BGAP* confirm this relationship. Indeed, the higher the average energy consumption, the larger is the order of magnitude of the running times.

On the other hand, the quality of the solution determined by the *3S-MHA* is unaffected by the average consumption and/or duration. However, the results of the *3S-MHA* are slightly worse than those of the exact approach with an average optimality gap around 2%. Moreover, we can observe that for several instances the *3S-MHA* is able to determine good solutions already in the second step (i.e., the first upper bound determined by the *3S-MHA*). Furthermore, the local search results to be very effective being able to reduce the average gap of up to the 30%. Finally, we point out that

**Table 3**
Exact and matheuristic approach results on the instances with 100 jobs.

| | Exact approach | | Matheuristic approach | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Instance Name | %Gap | Time | Time Step 1 | Initial %Gap | Time Step 2 | Final %Gap | Time Step 3 | Total Time | Best %Gap |
| **M2_J100_d10_E1** | **0,00** | **16,49** | 0,86 | 0,00 | 0,01 | 0,00 | 0,01 | 0,88 | 0,00 |
| **M2_J100_d10_E2** | **0,00** | **121,71** | 1,11 | 0,00 | 0,01 | 0,00 | 0,01 | 1,13 | 0,00 |
| **M2_J100_d10_E4** | **0,79** | **914,51** | 148,39 | 0,30 | 0,01 | 0,30 | 0,01 | 148,42 | 0,30 |
| **M2_J100_d20_E1** | **0,00** | **16,58** | 0,78 | 2,69 | 0,05 | 0,47 | 0,02 | 0,85 | 0,00 |
| **M2_J100_d20_E2** | **0,15** | **304,65** | 4,93 | 0,00 | 0,01 | 0,00 | 0,01 | 4,95 | 0,00 |
| **M2_J100_d20_E4** | **0,83** | **926,39** | 169,24 | 0,23 | 0,01 | 0,23 | 0,01 | 169,26 | 0,23 |
| **M2_J100_d30_E1** | **0,00** | **50,71** | 0,80 | 0,00 | 0,01 | 0,00 | 0,01 | 0,83 | 0,00 |
| **M2_J100_d30_E2** | **0,00** | **391,61** | 3,53 | 0,00 | 0,01 | 0,00 | 0,01 | 3,55 | 0,00 |
| **M2_J100_d30_E4** | **1,68** | **1307,76** | 169,72 | 0,19 | 0,01 | 0,19 | 0,01 | 169,74 | 0,10 |
| **M5_J100_d10_E1** | **0,00** | **87,45** | 0,83 | 5,40 | 0,04 | 0,06 | 0,02 | 0,88 | 0,06 |
| **M5_J100_d10_E2** | **0,23** | **810,63** | 5,68 | 0,00 | 0,16 | 0,00 | 0,01 | 5,85 | 0,00 |
| **M5_J100_d10_E4** | **0,91** | **1394,70** | 83,23 | 0,17 | 0,18 | 0,17 | 0,02 | 83,42 | 0,17 |
| **M5_J100_d20_E1** | **0,00** | **139,93** | 0,82 | 2,69 | 0,05 | 0,47 | 0,02 | 0,89 | 0,47 |
| **M5_J100_d20_E2** | **0,24** | **950,68** | 4,82 | 0,00 | 0,25 | 0,00 | 0,01 | 5,09 | 0,00 |
| **M5_J100_d20_E4** | **1,90** | **1631,26** | 248,60 | 0,36 | 0,18 | 0,36 | 0,01 | 248,79 | 0,30 |
| **M5_J100_d30_E1** | **0,00** | **144,69** | 0,89 | 3,51 | 0,05 | 0,27 | 0,02 | 0,96 | 0,27 |
| **M5_J100_d30_E2** | **0,34** | **1363,40** | 4,14 | 0,00 | 0,39 | 0,00 | 0,02 | 4,54 | 0,00 |
| **M5_J100_d30_E4** | **1,82** | **1546,11** | 223,50 | 0,47 | 0,21 | 0,47 | 0,02 | 223,74 | 0,47 |
| **M10_J100_d10_E1** | **0,00** | **134,75** | 0,79 | 48,34 | 0,03 | 3,70 | 0,06 | 0,88 | 3,70 |
| **M10_J100_d10_E2** | **1,89** | **1547,35** | 4,58 | 3,96 | 196,13 | 1,52 | 0,03 | 200,74 | 1,52 |
| **M10_J100_d10_E4** | **4,03** | **1805,11** | 24,69 | 0,09 | 78,17 | 0,09 | 0,02 | 102,88 | 0,09 |
| **M10_J100_d20_E1** | **0,00** | **400,86** | 0,80 | 48,28 | 0,02 | 2,70 | 0,04 | 0,85 | 2,70 |
| **M10_J100_d20_E2** | **1,38** | **1592,00** | 3,43 | 2,73 | 53,96 | 0,82 | 0,02 | 57,40 | 0,82 |
| **M10_J100_d20_E4** | **3,14** | **1776,97** | 164,83 | 0,25 | 2,27 | 0,25 | 0,02 | 167,12 | 0,12 |
| **M10_J100_d30_E1** | **0,00** | **288,56** | 0,86 | 36,07 | 0,02 | 2,73 | 0,03 | 0,91 | 2,73 |
| **M10_J100_d30_E2** | **1,31** | **1805,24** | 0,80 | 1,07 | 100,77 | 0,58 | 0,02 | 101,59 | 0,58 |
| **M10_J100_d30_E4** | **4,07** | **1752,79** | 113,70 | 0,10 | 2,80 | 0,10 | 0,02 | 116,52 | 0,10 |
| **Average** | **0,92** | **860,11** | 51,35 | 5,81 | 16,14 | 0,57 | 0,02 | 67,51 | 0,55 |

**Table 4**
Exact and matheuristic approach results on the instances with 150 jobs.

| | Exact approach | | Matheuristic approach | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Instance Name | %Gap | Time | Time Step 1 | Initial %Gap | Time Step 2 | Final %Gap | Time Step 3 | Total Time | Best %Gap |
| **M2_J150_d10_E1** | **0,00** | **107,40** | 2,31 | 0,00 | 0,02 | 0,00 | 0,03 | 2,36 | 0,00 |
| **M2_J150_d10_E1** | **0,00** | **107,40** | 2,31 | 0,00 | 0,02 | 0,00 | 0,03 | 2,36 | 0,00 |
| **M2_J150_d10_E2** | **0,15** | **739,41** | 16,31 | 0,00 | 0,01 | 0,00 | 0,03 | 16,35 | 0,00 |
| **M2_J150_d10_E4** | **3,60** | **1662,23** | 589,13 | 0,82 | 0,02 | 0,82 | 0,03 | 589,18 | 0,82 |
| **M2_J150_d20_E1** | **0,00** | **132,16** | 2,63 | 0,06 | 0,72 | 0,00 | 0,04 | 3,39 | 0,00 |
| **M2_J150_d20_E2** | **1,95** | **1027,11** | 38,15 | 0,00 | 0,02 | 0,00 | 0,03 | 38,20 | 0,00 |
| **M2_J150_d20_E4** | **3,87** | **1657,06** | 429,02 | 0,39 | 0,02 | 0,39 | 0,03 | 429,07 | 0,39 |
| **M2_J150_d30_E1** | **0,00** | **423,90** | 2,32 | 0,00 | 0,02 | 0,00 | 0,03 | 2,36 | 0,00 |
| **M2_J150_d30_E2** | **6,61** | **1564,06** | 182,76 | 0,15 | 0,02 | 0,15 | 0,03 | 182,80 | 0,15 |
| **M2_J150_d30_E4** | **4,05** | **1802,96** | 591,15 | 0,61 | 0,03 | 0,61 | 0,03 | 591,21 | 0,61 |
| **M5_J150_d10_E1** | **0,00** | **554,16** | 2,36 | 0,09 | 0,19 | 0,00 | 0,03 | 2,59 | 0,00 |
| **M5_J150_d10_E2** | **10,72** | **1713,33** | 17,23 | 0,00 | 0,22 | 0,00 | 0,02 | 17,48 | 0,00 |
| **M5_J150_d10_E4** | **25,48** | **1807,04** | 650,87 | 1,29 | 0,28 | 1,29 | 0,03 | 651,18 | 1,29 |
| **M5_J150_d20_E1** | **0,00** | **893,73** | 2,92 | 0,06 | 0,72 | 0,00 | 0,04 | 3,68 | 0,00 |
| **M5_J150_d20_E2** | **16,54** | **1746,76** | 6,01 | 0,00 | 0,39 | 0,00 | 0,04 | 6,44 | 0,00 |
| **M5_J150_d20_E4** | **16,36** | **1807,02** | 676,53 | 0,92 | 0,40 | 0,92 | 0,04 | 676,98 | 0,92 |
| **M5_J150_d30_E1** | **2,92** | **1565,71** | 2,25 | 0,06 | 0,93 | 0,01 | 0,03 | 3,21 | 0,01 |
| **M5_J150_d30_E2** | **19,26** | **1807,11** | 14,31 | 0,00 | 0,38 | 0,00 | 0,03 | 14,72 | 0,00 |
| **M5_J150_d30_E4** | **4,21** | **1718,78** | 649,87 | 0,62 | 0,38 | 0,62 | 0,03 | 650,29 | 0,62 |
| **M10_J150_d10_E1** | **7,56** | **1480,65** | 2,09 | 11,02 | 0,08 | 0,00 | 0,07 | 2,24 | 0,00 |
| **M10_J150_d10_E2** | **70,26** | **1813,99** | 2,64 | 0,00 | 1,27 | 0,00 | 0,05 | 3,96 | 0,00 |
| **M10_J150_d10_E4** | **3225,86** | **1813,81** | 694,05 | 0,89 | 2,42 | 0,89 | 0,05 | 696,52 | 0,89 |
| **M10_J150_d20_E1** | **8,28** | **1805,38** | 2,43 | 6,79 | 0,11 | 0,16 | 0,06 | 2,61 | 0,16 |
| **M10_J150_d20_E2** | **44,66** | **1814,04** | 2,47 | 0,00 | 5,43 | 0,00 | 0,05 | 7,95 | 0,00 |
| **M10_J150_d20_E4** | **1254,57** | **1850,38** | 678,97 | 0,89 | 5,15 | 0,89 | 0,05 | 684,17 | 0,89 |
| **M10_J150_d30_E1** | **5,65** | **1794,88** | 2,37 | 7,89 | 0,07 | 0,43 | 0,07 | 2,51 | 0,43 |
| **M10_J150_d30_E2** | **32,73** | **1814,02** | 30,66 | 0,03 | 98,86 | 0,03 | 0,04 | 129,56 | 0,03 |
| **M10_J150_d30_E4** | **948,75** | **1813,95** | 531,61 | 0,53 | 7,39 | 0,53 | 0,07 | 539,06 | 0,53 |
| **Average** | **211,63** | **1434,48** | 215,68 | 1,23 | 4,65 | 0,29 | 0,04 | 220,37 | 0,29 |

the best percentage gap is almost equal to the final percentage gap proving the quality of the lower bound of the *3S-MHA*. Finally, we can observe that the most time consuming step of the *3S-MHA* is the first one while the running time of the local search is almost negligible.

In Table 3, we can observe that the exact approach is able to optimally solve all the instances with an average energy consumption equal to 1. Moreover, the average optimality gap for the other instances is still low even if they are not solved within the time limit.

**Table 5**
Comparison of the exact and matheuristic approaches: solution quality and computation time.

| | Exact approach | | | | | Matheuristic approach | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | # ins | # opt | Av %gap | Max %gap | Av Time | # ins | # opt | Av %gap | Max %gap | Av Time |
| **J = 50** | 270 | 218 | 0.34 | 3.92 | 411.72 | 270 | 166 | 2.22 | 40.69 | 17.42 |
| **J = 100** | 270 | 168 | 0.92 | 7.54 | 860.3 | 270 | 201 | 0.55 | 16.3 | 67.5 |
| **J = 150** | 180 | 79 | 6.43 | 189.12 | 1262.77 | 270 | 184 | 0.29 | 2.93 | 220.35 |
| **M = 2** | 270 | 211 | 0.91 | 27.79 | 528.71 | 270 | 237 | 0.1 | 1.96 | 92.63 |
| **M = 5** | 270 | 160 | 3.84 | 189.12 | 925.23 | 270 | 188 | 0.41 | 14.05 | 104.51 |
| **M = 10** | 180 | 93 | 1.17 | 7.54 | 989.9 | 270 | 125 | 2.55 | 40.69 | 108.13 |
| $\mu_d$ **= 10** | 240 | 170 | 2.11 | 189.12 | 673.15 | 270 | 197 | 1.04 | 20.45 | 99.24 |
| $\mu_d$ **= 20** | 240 | 159 | 2.05 | 119.09 | 777.67 | 270 | 181 | 1.2 | 40.69 | 99 |
| $\mu_d$ **= 30** | 240 | 135 | 2.08 | 26.7 | 927.29 | 270 | 172 | 0.822 | 28.57 | 107.03 |
| $\mu_e$ **= 1** | 240 | 234 | 0.12 | 10.91 | 208.26 | 270 | 210 | 0.56 | 16.3 | 1.29 |
| $\mu_e$ **= 2** | 240 | 154 | 2.57 | 45.75 | 831.37 | 270 | 190 | 1.71 | 40.69 | 29.8 |
| $\mu_e$ **= 4** | 240 | 76 | 3.55 | 189.12 | 1338.47 | 270 | 150 | 0.8 | 13.48 | 274.18 |
| **All** | 720 | 464 | 2.08 | 189.12 | 792.7 | 810 | 550 | 1.02 | 40.69 | 101.76 |

On the other hand, the *3S-MHA* scales well since the average percentage optimality gap is around 0.5% and the computation time is in the order of minutes. Moreover, we point out that the *3S-MHA* determine a good solution already at the second step for almost all the instances. However, the local search still improves significantly the solution with an average reduction of up to 45% in some cases.

In Table 4, we can observe that the exact approach is able to effectively solve only the instances with 2 AGVs and just few instances with 5 AGVs. Moreover, we point out that it is even not possible to solve the linear relaxation within the time limit for some of the larger instances with 10 AGVs and an average consumption equal to 4. Instead, the matheuristic keeps its good performance in terms of both solution quality and computation times. However, the *BPP* formulation of the first step is not solved to optimality on different instances with an average energy consumption equal to 4. This could have a twofold effect on the matheuristic percentage gap. On one side, the resulting lower bound can be lower than the one that would have been obtained optimally solving the *BPP* so producing a higher gap. On the other side, the resulting number of charging operations could be greater than the optimal one so producing lower quality solutions. Neverthless, the average percentage gap on the instances with 150 jobs is the lowest one compared to the average gaps obtained by the *3S-MHA* on the instances with a different number of jobs. This result still confirms the effectiveness of the matheuristic approach. Moreover, we can observe that the average initial gap decreases when the number of jobs increases. This can be explained considering that a larger number of jobs requires a large number of charging operations. Therefore, a low-quality bin-packing solution has a minor impact on the quality of the first upper bound as the instance size grows.

Finally in Table 5, we report a comparison of the results of the two proposed approaches in terms of number of optimal solutions obtained, average and maximum percentage gap, and average time. The results are grouped on the basis of different parameters: number of jobs, number of AGVs, average job duration and average energy consumption. We point out that Table 5 does not contain the results of the *A-BGAP* on the instances with 150 jobs and 10 AGVs since they would lead to biased average results. Concerning the exact approach, it can be seen that it finds the optimal solution for 464 out of 720 considered instances. Moreover, we can observe that, as expected, the average, the maximum gap, and the running times increase when the instances become more complex (i.e., larger number of jobs and AGVs, larger average energy consumption). Furthermore, we point out that the average gap computed is always lower than 4% even with a maximum gap of 189.12%. This proves that the exact approach is generally able to effectively solve instances with up to 150 jobs and 5 AGVs. Finally, we highlight that the average job duration does not affect the performance of the exact approach since for each tested value we obtain similar results.

Concerning the results of the *3S-MHA*, we first observe that, in percentage, it is able to determine a greater number of optimal solutions compared to the exact approach. Moreover, we point out that both the average percentage optimality gaps and the running times are very low considering all the analyzed dimensions (jobs, number of AGVs, average time and energy consumption). These results prove the scalability and robustness of the proposed *3S-MHA*.

### *5.3. Results on large instances*

In Table 6, we report only the results of the *3S-MHA* since it is not possible to solve the linear relaxation of the *A-BGAP* formulation on almost all the instances with 200 jobs. Moreover, we did not report the matheuristic best percentage gap since the only lower bound available is the one provided by the *BPP* solution at the first step. We can observe that the average percentage gap is similar to the one obtained on the instances with 150 jobs. Instead, the running times are still relatively low with only the solution of the *BPP* requiring a higher amount of time while the running times of the other two steps (i.e., the solution of the *BGAP-R*/*BGAP-C* and the local search) are almost negligible. These results prove that the matheuristic is able to scale well with the instance size. As a final general comment, we point out that in most solutions, the battery level is almost near to full depletion before the charging activity. In particular, we observed that the residual energy is, on average, about 5% of the battery capacity. Therefore, the computational results sustain the assumption on the fixed charging time.

### *5.4. Comparison of solution approaches for the ASP-BC*

In this section, we focus on comparing solution approaches for the *ASP-BC*. In particular, we compare our approaches with the formulation proposed in Masone et al. (2021) and the approach used by the company since, to the best of our knowledge, there are no other approaches in literature specifically designed to address the *ASP-BC*. On this basis, we set up two experiments. In the first one, we compare the exact approaches for the *ASP-BC*, i.e., our *A-BGAP* formulation with the one reported in Masone et al. (2021) (*MSS21* in the following). In the second one, we compare the heuristic approaches, i.e., the *3S-MHA* with the company solution approach (*CSA* in the following). For the sake of completeness, we point out that *MSS21* formulation is solved using Gurobi 9.1 with a time limit

**Table 6**
Matheuristic approach results on the instances with 200 jobs.

| | Matheuristic approach | | | | | |
|---|---|---|---|---|---|---|
| **Instance Name** | **Time Step 1** | **Initial %Gap** | **Time Step 2** | **Final %Gap** | **Time Step 3** | **Total Time** |
| **M2_J200_d10_E1** | 3,91 | 0,00 | 0,01 | 0,00 | 0,04 | 3,96 |
| **M2_J200_d10_E2** | 87,58 | 0,11 | 0,02 | 0,11 | 0,04 | 87,64 |
| **M2_J200_d10_E4** | 721,66 | 0,99 | 0,02 | 0,99 | 0,04 | 721,73 |
| **M2_J200_d20_E1** | 4,19 | 0,00 | 0,42 | 0,00 | 0,06 | 4,67 |
| **M2_J200_d20_E2** | 52,14 | 0,00 | 0,02 | 0,00 | 0,04 | 52,20 |
| **M2_J200_d20_E4** | 721,67 | 0,91 | 0,02 | 0,91 | 0,04 | 721,73 |
| **M2_J200_d30_E1** | 4,18 | 0,00 | 0,01 | 0,00 | 0,04 | 4,23 |
| **M2_J200_d30_E2** | 39,35 | 0,00 | 0,02 | 0,00 | 0,04 | 39,41 |
| **M2_J200_d30_E4** | 721,66 | 0,65 | 0,02 | 0,65 | 0,04 | 721,72 |
| **M5_J200_d10_E1** | 4,40 | 0,00 | 0,25 | 0,00 | 0,04 | 4,69 |
| **M5_J200_d10_E2** | 9,69 | 0,00 | 0,18 | 0,00 | 0,04 | 9,91 |
| **M5_J200_d10_E4** | 721,69 | 1,16 | 0,20 | 1,16 | 0,05 | 721,93 |
| **M5_J200_d20_E1** | 4,19 | 0,00 | 0,42 | 0,00 | 0,06 | 4,67 |
| **M5_J200_d20_E2** | 147,98 | 0,15 | 0,40 | 0,15 | 0,06 | 148,43 |
| **M5_J200_d20_E4** | 721,66 | 0,87 | 0,31 | 0,87 | 0,07 | 722,03 |
| **M5_J200_d30_E1** | 4,27 | 0,00 | 0,46 | 0,00 | 0,05 | 4,78 |
| **M5_J200_d30_E2** | 37,88 | 0,00 | 0,27 | 0,00 | 0,06 | 38,21 |
| **M5_J200_d30_E4** | 721,64 | 0,83 | 0,35 | 0,83 | 0,04 | 722,03 |
| **M10_J200_d10_E1** | 6,73 | 2,80 | 227,84 | 0,00 | 0,08 | 234,65 |
| **M10_J200_d10_E2** | 77,54 | 0,11 | 2,46 | 0,11 | 0,06 | 80,06 |
| **M10_J200_d10_E4** | 721,68 | 1,21 | 1,72 | 1,21 | 0,08 | 723,49 |
| **M10_J200_d20_E1** | 4,57 | 2,10 | 209,83 | 0,10 | 0,09 | 214,49 |
| **M10_J200_d20_E2** | 129,52 | 0,08 | 21,32 | 0,08 | 0,06 | 150,90 |
| **M10_J200_d20_E4** | 629,51 | 0,66 | 4,04 | 0,66 | 0,09 | 633,64 |
| **M10_J200_d30_E1** | 4,31 | 1,85 | 93,59 | 0,04 | 0,09 | 97,99 |
| **M10_J200_d30_E2** | 100,14 | 0,06 | 17,67 | 0,06 | 0,08 | 117,89 |
| **M10_J200_d30_E4** | 721,66 | 0,76 | 5,24 | 0,76 | 0,10 | 727,00 |
| **Average** | 263,90 | 0,57 | 21,74 | 0,32 | 0,06 | 285,71 |

**Table 7**
Comparison of the exact approaches for the *ASP-BC*.

| | $Diff_{LB}$ | | | $Diff_{UB}$ | | | Speed-up | | |
|---|---|---|---|---|---|---|---|---|---|
| **Instance name** | **Min** | **Avg** | **Max** | **Min** | **Avg** | **Max** | **Min** | **Avg** | **Max** |
| **M2_J50_d10_E1** | 3.85 | 4.27 | 4.59 | 1.89 | 2.21 | 2.60 | 314.23 | 485.50 | 732.66 |
| **M2_J50_d10_E2** | 5.94 | 6.55 | 7.66 | 1.29 | 1.43 | 1.68 | 71.38 | 230.76 | 511.30 |
| **M2_J50_d10_E4** | 8.29 | 10.38 | 12.34 | 0.46 | 0.68 | 0.99 | 60.25 | 200.57 | 543.40 |
| **M2_J50_d20_E1** | 5.42 | 6.05 | 6.67 | 1.34 | 1.42 | 1.51 | 340.09 | 499.87 | 644.20 |
| **M2_J50_d20_E2** | 7.25 | 7.85 | 8.65 | 0.90 | 1.01 | 1.08 | 9.83 | 198.26 | 445.43 |
| **M2_J50_d20_E4** | 9.02 | 10.72 | 11.76 | 0.41 | 0.54 | 0.71 | 1.00 | 113.19 | 509.46 |
| **M2_J50_d30_E1** | 6.04 | 6.97 | 7.99 | 0.95 | 1.09 | 1.18 | 185.17 | 349.20 | 573.36 |
| **M2_J50_d30_E2** | 7.56 | 8.22 | 9.24 | 0.69 | 0.82 | 0.89 | 44.78 | 151.74 | 438.56 |
| **M2_J50_d30_E4** | 8.80 | 10.52 | 12.23 | 0.37 | 0.48 | 0.64 | 1.00 | 92.22 | 576.48 |
| **M5_J50_d10_E1** | 2.63 | 3.40 | 4.22 | 2.42 | 2.85 | 3.11 | 182.90 | 329.09 | 385.56 |
| **M5_J50_d10_E2** | 4.02 | 5.31 | 6.75 | 1.48 | 1.82 | 2.08 | 88.41 | 179.88 | 251.54 |
| **M5_J50_d10_E4** | 6.57 | 9.07 | 11.98 | 0.41 | 0.79 | 1.06 | 1.00 | 46.54 | 345.72 |
| **M5_J50_d20_E1** | 2.02 | 2.98 | 3.63 | 1.51 | 1.70 | 1.88 | 186.38 | 264.75 | 329.34 |
| **M5_J50_d20_E2** | 2.94 | 3.98 | 5.05 | 1.10 | 1.22 | 1.36 | 27.39 | 105.48 | 201.18 |
| **M5_J50_d20_E4** | 4.28 | 5.73 | 7.63 | 0.50 | 0.67 | 0.81 | 1.00 | 23.54 | 86.82 |
| **M5_J50_d30_E1** | 2.31 | 2.70 | 3.42 | 1.14 | 1.31 | 1.51 | 63.24 | 255.38 | 342.49 |
| **M5_J50_d30_E2** | 3.00 | 3.50 | 4.28 | 0.82 | 0.94 | 1.09 | 1.00 | 50.32 | 106.41 |
| **M5_J50_d30_E4** | 3.85 | 4.67 | 5.87 | 0.39 | 0.55 | 0.67 | 1.00 | 32.64 | 188.37 |
| **M10_J50_d10_E1** | 0.73 | 1.43 | 3.99 | 3.15 | 3.73 | 4.71 | 923.04 | 994.52 | 1042.91 |
| **M10_J50_d10_E2** | 1.18 | 2.22 | 4.38 | 2.10 | 2.42 | 2.67 | 1.59 | 82.52 | 196.91 |
| **M10_J50_d10_E4** | 2.14 | 4.37 | 9.74 | 0.70 | 1.04 | 1.28 | 1.00 | 19.19 | 165.46 |
| **M10_J50_d20_E1** | 0.50 | 0.83 | 1.19 | 1.91 | 2.15 | 2.49 | 500.20 | 829.89 | 980.55 |
| **M10_J50_d20_E2** | 0.81 | 1.10 | 1.51 | 1.51 | 1.74 | 2.14 | 1.00 | 10.29 | 27.22 |
| **M10_J50_d20_E4** | 1.43 | 2.05 | 2.72 | 0.70 | 0.90 | 1.18 | 1.00 | 5.88 | 25.76 |
| **M10_J50_d30_E1** | 0.54 | 0.89 | 1.38 | 1.24 | 1.43 | 1.62 | 587.81 | 778.82 | 966.01 |
| **M10_J50_d30_E2** | 0.75 | 1.11 | 1.73 | 1.06 | 1.21 | 1.41 | 1.00 | 9.06 | 41.24 |
| **M10_J50_d30_E4** | 1.18 | 1.66 | 2.22 | 0.53 | 0.77 | 0.97 | 1.00 | 1.09 | 1.91 |
| **All** | 0.50 | 4.76 | 12.34 | 0.37 | 1.37 | 4.71 | 1.00 | 234.82 | 1042.91 |

of 1800 s. Moreover, the *CSA* is implemented in Python language. In particular, the *CSA* is based on the Longest Processing Time rule. The approach first sorts the jobs by duration in descending order. Then, according to this order, the approach assigns each job to the least loaded machine, where machine loads are computed considering potential charging operations. In particular, if adding a job to a machine requires an additional charging operation, the machine load is considered equal to the current load and the potential charging time.

Concerning the comparison of exact approaches, we report only the results on the instances with 50 jobs since our experiments showed us that *MSS21* formulation is not able to address larger instances effectively. The results of this comparison are reported in Table 7, where we summarize the minimum, the average, and the maximum of the difference between the lower bounds ($Diff_{LB}$), the difference between the upper bounds ($Diff_{UB}$), and the speed-up. In particular, $Diff_{LB}$ is computed as follows: $LB_{A_BGAP}/LB_{MSS21} - 1$, where $LB_{A-BGAP}(LB_{MSS21})$ is the lower bound of the *A-BGAP(MSS21)* formulation. On the other hand, $Diff_{UB}$ is computed as $UB_{MSS21}/UB_{A-BGAP} - 1$, where $UB_{A-BGAP}(UB_{MSS21})$ is the upper bound of the *A-BGAP(MSS21)* formulation. Finally, the speed-up is computed as $t_{MSS21}/t_{A-BGAP}$, where $t_{A-BGAP}(t_{MSS21})$ is the running time of the *A-BGAP(MSS21)* formulation.

The results show that from all points of view, the *A-BGAP* formulation outperforms the *MSS21*. Indeed, the *MSS21* formulation is not able to optimally solve any instance within the time limit, while the *A-BGAP* formulation is able to solve 218 out of 270 instances optimally. Moreover, even only considering the results of the instances not optimally solved, it is possible to observe that the average $Diff_{LB}$ and the $Diff_{UB}$ are 3.89 and 0.89, so confirming the superiority of the *A-BGAP* formulation.

Concerning the comparison of the heuristic approach, we do not report the detailed results for privacy reasons. However, we highlight that the company can achieve significant gains if it uses the *3S-MHA* instead of the *CSA*. Indeed, if we compute the percentage gain as $(UB_{CSA} - UB_{3S-MHA})/UB_{3S-MHA} \cdot 100$, then the maximum and the average gains are about 40% and 10%, respectively. Finally, we point out that in terms of computation time, the *CSA* is slightly faster than the *3S-MHA* (on average, a few seconds and a couple of minutes, respectively). However, the *3S-MHA* is still preferable to the *CSA* since it provides a lower bound for the *ASP-BC* which can be used to evaluate the quality of the obtained solution.

## 6. Conclusions

Nowadays, AGV systems are broadly used in different application fields due to their wide range of benefits. However, the design of an AGV system is a complex task due to the many tactical and operational issues that have to be addressed. Among the operational issues, the AGV scheduling is one of the most relevant, since it greatly affects the performance of the AGV system. In this paper, we study a particular scheduling problem named *ASP-BC*. It consists in the scheduling of a set of transfer jobs on a fleet of homogeneous AGVs such that the makespan is minimized and AGV battery constraints are satisfied.

We proposed an original *MILP* formulation based on the representation of the *ASP-BC* as a variant of the *BGAP*. The computational experiments showed that the *A-BGAP* formulation allows us to effectively solve medium size instances within a reasonable amount of time outperforming the state-of-the-art exact method for the *ASP-BC*. Moreover, we showed that the complexity of the *ASP-BC* is closely related to the job energy consumption.

We developed a three-step matheuristic which determines both an upper and a valid lower bound for the *ASP-BC*. The computational results showed the robustness of the *3S-MHA* since its performance does not depend on the instance features. Moreover, the results also proved the effectiveness and the scalability of the *3S-MHA* being it able to solve instances with up to 200 jobs and 10 AGV in few minutes with an average gap of lower than 1%.

Future research directions naturally include the extension of the *ASP-BC* to partial charging operations and the inclusion of residual energy in the computation of the speed of the AGVs. In addition, it may be worth considering different objective functions to investigate in which contexts an effective management of the AGV battery constraints provides the largest benefits.

## References


Aggoune, R. (2004). Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research, 153*(3), 534–543. https://doi.org/10.1016/S0377-2217(03)00261-3. EURO Young Scientists

Asghari, M., Fathollahi-Fard, A. M., Mirzapour Al-e hashem, S., & Dulebenets, M. A. (2022). Transformation and linearization techniques in optimization: A state-of-the-art survey. *Mathematics, 10*(2), 283.

Avalos-Rosales, O., Angel-Bello, F., Álvarez, A., & Cardona-Valdés, Y. (2018). Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times. *Computers & Industrial Engineering, 123*, 364–377. https://doi.org/10.1016/j.cie.2018.07.006.

Bechtsis, D., Tsolakis, N., Vlachos, D., & Iakovou, E. (2017). Sustainable supply chain management in the digitalisation era: The impact of automated guided vehicles. *Journal of Cleaner Production, 142*, 3970–3984. https://doi.org/10.1016/j.jclepro.2016.10.057.

Boccia, M., Masone, A., Sterle, C., & Murino, T. (2021). *ASP-BC* Dataset: Instances and detailed solution results. https://doi.org/10.5281/zenodo.5703131, [Online; accessed 15-November-2021].

Bongiovanni, C., Kaspi, M., & Geroliminis, N. (2019). The electric autonomous dial-a-ride problem. *Transportation Research Part B: Methodological, 122*, 436–456. https://doi.org/10.1016/j.trb.2019.03.004.

Cassady, C. R., & Kutanoglu, E. (2003). Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. *IIE Transactions, 35*(6), 503–513. https://doi.org/10.1080/07408170304416.

Costa, A., Cappadonna, F., & Fichera, S. (2016). Minimizing the total completion time on a parallel machine system with tool changes. *Computers & Industrial Engineering, 91*, 290–301. https://doi.org/10.1016/j.cie.2015.11.015.

Cui, W.-W., & Lu, Z. (2017). Minimizing the makespan on a single machine with flexible maintenances and jobs' release dates. *Computers & Operations Research, 80*, 11–22. https://doi.org/10.1016/j.cor.2016.11.008.

Dang, Q.-V., Singh, N., Adan, I., Martagan, T., & van de Sande, D. (2021). Scheduling heterogeneous multi-load AgVs with battery constraints. *Computers & Operations Research, 136*, 105517. https://doi.org/10.1016/j.cor.2021.105517.

De Ryck, M., Versteyhe, M., & Shariatmadar, K. (2020). Resource management in decentralized industrial automated guided vehicle systems. *Journal of Manufacturing Systems, 54*, 204–214. https://doi.org/10.1016/j.jmsy.2019.11.003.

Erdelić, T., & Carić, T. (2019). A survey on the electric vehicle routing problem: variants and solution approaches. *Journal of Advanced Transportation, 2019*.

Fatnassi, E., & Chaouachi, J. (2015). Scheduling automated guided vehicle with battery constraints. In *2015 20th international conference on methods and models in automation and robotics (MMAR)* (pp. 1010–1015). https://doi.org/10.1109/MMAR.2015.7284017.

Fazlollahtabar, H., & Saidi-Mehrabad, M. (2015). Methodologies to optimize automated guided vehicle scheduling and routing problems: A review study. *Journal of Intelligent and Robotic Systems, 77*, 525–545. https://doi.org/10.1007/s10846-013-0003-8.

Fragapane, G., de Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research, 294*(2), 405–426. https://doi.org/10.1016/j.ejor.2021.01.019.

Fu, B., Huo, Y., & Zhao, H. (2011). Approximation schemes for parallel machine scheduling with availability constraints. *Discrete Applied Mathematics, 159*(15), 1555–1565. https://doi.org/10.1016/j.dam.2011.06.007.

Gedik, R., Rainwater, C., Nachtmann, H., & Pohl, E. A. (2016). Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals. *European Journal of Operational Research, 251*(2), 640–650. https://doi.org/10.1016/j.ejor.2015.11.020.

Geurtsen, M., Didden, J. B., Adan, J., Atan, Z., & Adan, I. (2022). Production, maintenance and resource scheduling: A review. *European Journal of Operational Research*.

Graves, G., & Lee, C.-Y. (1999). Scheduling maintenance and semiresumable jobs on a single machine. *Naval Research Logistics, 46*(7), 845–863. Cited By 145. doi: https://doi.org/10.1002/(SICI)1520-6750(199910)46:7⟨845::AID-NAV6⟩3.0.CO;2-#

Kaoud, E., El-Sharief, M. A., & El-Sebaie, M. (2017). Scheduling problems of automated guided vehicles in job shop, flow shop, and container terminals. In *2017 4th international conference on industrial engineering and applications (ICIEA)* (pp. 60–65). https://doi.org/10.1109/IEA.2017.7939179.

Kucukoglu, I., Dewil, R., & Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering, 161*, 107650.

Lee, C.-Y., & Chen, Z.-L. (2000). Scheduling jobs and maintenance activities on parallel machines. *Naval Research Logistics, 47*(2), 145–165. https://doi.org/10.1002/(SICI)1520-6750(200003)47:2<145::AID-NAV5>3.0.CO;2-3. Cited By 177

Liao, L.-W., & Sheen, G.-J. (2008). Parallel machine scheduling with machine availability and eligibility constraints. *European Journal of Operational Research, 184*(2), 458–467. https://doi.org/10.1016/j.ejor.2006.11.027.

Lin, J., Zhou, W., & Wolfson, O. (2016). Electric vehicle routing problem. *Transportation Research Procedia, 12*, 508–521. https://doi.org/10.1016/j.trpro.2016.02.007. Tenth International Conference on City Logistics 17–19 June 2015, Tenerife, Spain

Luo, J., Wu, Y., & Mendes, A. B. (2016). Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container

terminal. *Computers & Industrial Engineering, 94*, 32–44. https://doi.org/10.1016/j.cie.2016.01.010.
Martello, S., & Toth, P. (1995). The bottleneck generalized assignment problem. *European Journal of Operational Research, 83*(3), 621–638. https://doi.org/10.1016/0377-2217(93)E0271-X.
Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., & Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research Part E: Logistics and Transportation Review, 118*, 392–420. https://doi.org/10.1016/j.tre.2018.08.005.
Masone, A., Murino, T., Sterle, C., & Strazzullo, M. (2021). A milp formulation for an automated guided vehicle scheduling problem with battery constraints. In A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, & D. Romero (Eds.), *Advances in production management systems. artificial intelligence for sustainable and resilient production systems* (pp. 15–23). Cham: Springer International Publishing.
McHaney, R. (1995). Modelling battery constraints in discrete event automated guided vehicle simulations. *International Journal of Production Research, 33*(11), 3023–3040. https://doi.org/10.1080/00207549508904859.
Mehami, J., Nawi, M., & Zhong, R. Y. (2018). Smart automated guided vehicles for manufacturing in the context of industry 4.0. *Procedia Manufacturing, 26*, 1077–1086. https://doi.org/10.1016/j.promfg.2018.07.144. 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA
Mellouli, R., Sadfi, C., Chu, C., & Kacem, I. (2009). Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times. *European Journal of Operational Research, 197*(3), 1150–1165. https://doi.org/10.1016/j.ejor.2008.03.043.
Qiu, L., Hsu, W.-J., Huang, S.-Y., & Wang, H. (2002). Scheduling and routing algorithms for AgVs: A survey. *International Journal of Production Research, 40*(3), 745–760. https://doi.org/10.1080/00207540110091712.
Singh, N., Dang, Q.-V., Akcay, A., Adan, I., & Martagan, T. (2021). A matheuristic for AgV scheduling with battery constraints. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2021.08.008.
Sun, K., & Li, H. (2010). Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines. *International Journal of Production Economics, 124*(1), 151–158. https://doi.org/10.1016/j.ijpe.2009.10.018.
Vis, I. F. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research, 170*(3), 677–709. https://doi.org/10.1016/j.ejor.2004.09.020.
Wang, X., & Cheng, T. (2015). A heuristic for scheduling jobs on two identical parallel machines with a machine availability constraint. *International Journal of Production Economics, 161*, 74–82. https://doi.org/10.1016/j.ijpe.2014.12.001.
Wei, L., Luo, Z., Baldacci, R., & Lim, A. (2020). A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. *INFORMS Journal on Computing, 32*(2), 428–443. https://doi.org/10.1287/ijoc.2018.0867.
Xie, C., & Allen, T. T. (2015). Simulation and experimental design methods for job shop scheduling with material handling: A survey. *The International Journal of Advanced Manufacturing Technology, 80*, 233–243. https://doi.org/10.1007/s00170-015-6981-x.
Yoo, J., & Lee, I. S. (2016). Parallel machine scheduling with maintenance activities. *Computers & Industrial Engineering, 101*, 361–371. https://doi.org/10.1016/j.cie.2016.09.020.