Optimal management of an AGV based internal distribution system: MILP formulation and heuristic approach

M. Boccia[a], A. Mancuso[a], A. Masone[a], A. Sforza[a] and C. Sterle[a,*]

[a] Department of Electrical Engineering and Information Technology,
University Federico II of Naples, Naples, Italy
{maurizio.boccia, andrea.mancuso, adriano.masone, antonio.sforza, claudio.sterle}@unina.it,
[*] Corresponding author

Extended abstract submitted for presentation at the 11th Triennial Symposium on
Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

March 31, 2022

Keywords: AGV transportation system; battery constraints; scheduling; MILP; heuristic

1 Introduction

Automated guided vehicles (AGVs) are driverless transportation systems used for horizontal movement of materials. Since their introduction in 1955, they had a great impact on logistics activities due to their dexterity, efficiency and flexibility. Generally, AGVs are part of larger system used for the internal transportation of goods and materials among various departments and locations within the same factory or for receiving, storage and sorting operations in shipment areas. Thus, the main aim of an AGV-based transportation system can be summarized as: transferring/handling the right amount of the right material to the right place at the right time.

It is straightforward to understand that such aim configures a scheduling problem where transportation tasks have to be effectively assigned to AGVs. Literature on this topic is quite rich, as witnessed by the survey works by Kaoud *et al.* (2017), Qiu *et al.* (2002) and Xie & Allen (2015), where several scheduling problems, differing for the considered AGV side constraints, are reviewed (Fazlollahtabar & Saidi-Mehrabad, 2015). However, as noted in De Ryck *et al.* (2020), most of the contributions do not deal with battery depletion and recharge issues. To the best of authors' knowledge, the only work tackling the AGV scheduling problem with battery contraints (*ASP-BC*) is described in Masone *et al.* (2021).

In a nutshell, the *ASP-BC* can be schematized on the basis of the following assumptions:
1) a set of AGVs must move a set of packages from a central warehouse to different workstations;
2) all the AGVs are initially located at the central warehouse with fully charged batteries;
3) an AGV can be loaded with a single package on each trip. Being the number of available AGVs much lower than the number of packages, an AGV should perform more than one trip;
4) the AGV battery consumption depends on the travel time and the weight carried;
5) an AGV battery has to be fully recharged before it is completely depleted. The charging time is fixed and does not depend on the residual energy;
6) the *ASP-BC* objective is to minimize the makespan of the handling process.

For the sake of the completeness, we have to say that this problem recalls the parallel machine scheduling problem with coordinated maintenance activities, where machines cannot continuously operate longer than a pre-defined working time without performing a maintenance activity

(Yoo & Lee, 2016). However, the *ASP-BC* has several differentiating features. From an operational point of view, we highlight that maintenance activities are scheduled considering the time dimension (i.e. the elapsed time from the beginning of the process or the time passed from the last performed maintenance). Instead, in the *ASP-BC*, the focus is on the battery depletion dimension, since each job consumes a specific amount of energy. From a methodological point of view, the *ASP-BC* is aimed at minimizing the makespan of the handling process instead of the sum of the job completion times. This does not allow to straightly use exact and heuristic methods proposed for scheduling problems with maintenance activities to solve the *ASP-BC*, except for a significant rethinking of the method with no effectiveness guarantee.

In this context, this work is aimed at filling this gap providing the following threefold contribution: 1) proposing an original mixed-integer linear programming (MILP) model for the *ASP-BC*; 2) developing a heuristic approach, integrated with a local search procedure, to solve large size instances of the problem; 3) validating the proposed methods on instances built from real data provided by a manufacturing company.
The rest of this paper is structured as follows: the proposed *MILP* formulation and the heuristic approach are sketched in 2; the computational results are reported and discussed in Section 3.

2 Problem Formulation and Heuristic Solution Method

In the following, we will refer the combination of an AGV loaded trip from warehouse to workstations and the empty trip back as a job. A job can be characterized by a duration given by the sum of the travel time from the warehouse to the workstation and back, and the time needed for the package load and unload operations. Moreover, each job requires a certain amount of energy to be performed by an AGV. Specifically, the required energy depends on the job duration and the package weight. Therefore, the number of consecutive jobs that an AGV can perform without battery recharge is limited. On this basis, let J and K be the set of jobs and AGVs, respectively. Moreover, we introduce a starting/ending dummy job, denoted with o/d, for each AGV. For the sake of readability, we also define J^+ as $J \cup \{o\}$ and J^- as $J \cup \{d\}$. The *ASP-BC* solution involves three kinds of decisions: 1) assignment of jobs to the AGVs; 2) sequencing of the jobs on each AGV; 3) scheduling of AGV battery recharge. The objective is to minimize the makespan that is given by the maximum completion time of the last job over all the AGVs.

On this basis, we introduce the following decision variables: $C_{\max}$ continuous variable equal to the completion time of handling process; c_j^k continuous variables equal to the completion time of job j on the AGV k; e_j^k continuous variables equal to the energy spent by the AGV k from its last charge to the completion of job j; y_{ij}^k binary variable equal to 1 if job j is performed immediately after job i on the k-th AGV, 0 otherwise: δ_j^k binary variable equal to 1 if the AGV k is recharged before performing job j, 0 otherwise.

Then, we indicate with t_j^k and w_j^k the processing time and energy required, respectively, for each job j and for each AGV k. Finally, being r the charging time and E the battery capacity, the *ASP-BC* can be formulated as follows:

$$\min C_{\max} \tag{1}$$

s.t.

$$C_{\max} \geq c_j^k \qquad \forall j \in J, k \in K \tag{2}$$

$$c_j^k \geq c_i^k + t_j^k + r\delta_j^k - M(1 - y_{ij}^k) \qquad \forall i \in J^+, j \in J, k \in K \tag{3}$$

$$c_o^k = 0 \qquad \forall k \in K \tag{4}$$

$$\sum_{i \in J^+} \sum_{k \in K} y_{ij}^k = 1 \qquad \forall j \in J \tag{5}$$

$$\sum_{j \in J} y^k_{oj} = \sum_{j \in J} y^k_{jd} = 1 \qquad \forall k \in K \tag{6}$$

$$\sum_{j \in J^+} y^k_{ji} = \sum_{j \in J^-} y^k_{ij} \qquad \forall i \in J, K \in K \tag{7}$$

$$e^k_j \geq e^k_i + w^k_j - \sum_{j \in J} M(1 - y^k_{ji}) - M\delta^k_j \qquad \forall i, j \in J, k \in K \tag{8}$$

$$e^k_j \geq w^k_j - M(1 - \sum_{i \in J^+} y^k_{ij}) \qquad \forall j \in J, k \in K \tag{9}$$

$$e^k_j + w^k_i \leq E + M(1 - y^k_{ji}) + M\delta^k_j \qquad \forall i \in J, j \in J^+, k \in K \tag{10}$$

The objective function (1) minimizes the makespan. Constraints (2) set the makespan of the handling process greater than the completion time of any processed job. Constraints (3) ensure that if a job j is performed on the AGV k after a job i then its completion time is greater than the completion time of i plus the processing time of j and the charging time (if a recharge is scheduled). Constraints (4) set to 0 the completion time of the initial dummy job. Constraints (5) guarantee that each job is performed. Constraints (6-7) ensure that each AGV can perform one job at a time. Constraints (8, 9) set the value of the energy consumption of each AGV after each performed job. Constraints (10) guarantee that an AGV is recharged if, performing a job, the energy consumption would exceed its battery capacity.

The proposed formulation can be used to solve small and medium size instances of the problem. However, for larger instances, the need of ad-hoc solution methods arises. Thus, we solved the *ASP-BC* by a heuristic approach exploiting a natural decomposition of the problem in its two assignment sub-problems: transfer jobs to charging operations; charging operations to AGVs. These two problems are solved sequentially and can be formulated starting from the proposed *ASP-BC* formulation with some specific settings. The first one is obtained solving the *ASP-BC* with only one AGV. Its solution allows us to compute a lower bound for the original *ASP-BC* and to determine an assignments of subset of jobs to the charging operations. The second sub-problem fixes these assignment of jobs to the charging operations and schedules them among the AGVs so determining an upper bound for the *ASP-BC*. Then, the solution is improved through a local search procedure based on job moves that can be conceived as the well-known add, swap, and remove operations. Thus, the heuristic configures a three-step matheuristic approach which is able to provide both a lower and an upper bound for the *ASP-BC*.

For the sake of the brevity, we do not provide the details of the method. However, we point out that the last two steps can be conceived as a single "wide" local search step which can be performed also exploiting different metaheuristic structures (e.g., genetic algorithm or others).

3 Computational experiments and Discussion

In this section, we present the experimentation performed to evaluate and validate the proposed three-step 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 *MILP* formulations are solved using Gurobi 9.1.

We considered instances with $|K|$ equal to 2, 5, 10 and $|J|$ equal to 50, 100 and 150. Then, we generated the job processing times, (t^k_j), following a normal distribution $\mathcal{N}(\mu_t, (\mu_t/2)^2)$, with $\mu_t \in \{10, 20, 30\}$. Similarly, the job energy consumption, (w^k_j), are generated following a normal distribution $\mathcal{N}(\mu_w, (\mu_w/2)^2)$, with $\mu_w \in \{1, 2, 4\}$. We point out that we considered two independent distributions for t^k_j and w^k_j to simulate different AGV technologies, each of them characterized by a specific function linking carried weight, travelling time and energy consumption. The charging time r of each AGV is equal to 1 hour and the battery capacity E is equal to 10 and it assumed to be the same for all the AGVs. Finally, for each combination of $|J|, \mu_t,$

μ_w, and K we generated a set of 10 instances.

In Table 1, we report the results of the proposed approach in terms of number of optimal solutions obtained, average and maximum percentage gap, and average time. For the sake of the brevity, we report only the results of the best *3S-MHA* implementation and setting. The results are grouped on the basis of different parameters: number of jobs, number of AGVs, average job duration and average energy consumption. We observe that *3S-MHA* is able to determine a great number of optimal solutions. 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*. We also highlight that preliminary results obtained exploiting a metaheuristic scheme in the implementation of the *3S-MHA*, are further confirming the effectiveness of the proposed approach in dealing large scale *ASP-BC* instances.

Table 1 – *Results of the three-step matheuristic approach*

	# inst	**# opt**	**Av %gap**	**Max %gap**	**Av CPU Time**
J = 50	270	166	2.76	40.69	17.42
J = 100	270	201	0.56	16.3	67.5
J = 150	270	184	0.29	2.93	220.35
M = 2	270	237	0.1	1.96	92.63
M = 5	270	188	0.41	14.05	104.51
M = 10	270	125	2.55	40.69	108.13
Av Time = 10	270	197	1.04	20.45	99.24
Av Time = 20	270	181	1.2	40.69	99
Av Time = 30	270	172	0.822	28.57	107.03
Av Weight = 1	270	210	0.56	16.3	1.29
Av Weight = 2	270	190	1.71	40.69	29.8
Av Weight = 4	270	150	0.8	13.48	274.18
All	810	550	1.02	40.69	101.76

The performed experimentation derived from real data, confirm the applicability and the effectiveness of the proposed approach. Therefore, future works will be aimed at investigating the possibility of taking into account other operational aspects affecting the battery consumption (e.g., the AGV routing). Indeed, for the sake of completeness, we highlight that *ASP-BC* can be considered as a sub-problem of the Electric Vehicle Routing Problem (*EVRP*), which is a special case of the AGV routing problems with battery constraints (Bongiovanni *et al.*, 2019), where scheduling decisions are integrated with routing decisions related to the movement of each job from a pick-up to a delivery point. Therefore, it would be interesting to extend the proposed approach to problems involving both routing and scheduling decisions.

References

Bongiovanni, Claudia, Kaspi, Mor, & Geroliminis, Nikolas. 2019. The electric autonomous dial-a-ride problem. *Transportation Research Part B: Methodological*, **122**, 436–456.

De Ryck, M., Versteyhe, M., & Debrouwere, F. 2020. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, **54**, 152–173.

Fazlollahtabar, H., & Saidi-Mehrabad, M. 2015. Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study. *J Intell Robot Syst*, **77**, 525–545.

Kaoud, E., El-Sharief, M. A., & El-Sebaie, M.G. 2017. Scheduling problems of automated guided vehicles in job shop, flow shop, and container terminals. *Pages 60–65 of: 2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*.

Masone, A., Murino, T., Sterle, C., & Strazzullo, M. 2021. A MILP Formulation for an Automated Guided Vehicle Scheduling Problem with Battery Constraints. *Pages 15–23 of:* Dolgui, et al. (ed), *APMS. Artificial Intelligence for Sustainable and Resilient Production Systems*. Springer Int. Publ.

Qiu, Ling, Hsu, Wen-Jing, Huang, Shell-Ying, & Wang, Han. 2002. Scheduling and routing algorithms for AGVs: A survey. *International Journal of Production Research*, **40**(3), 745–760.

Xie, C., & Allen, T. T. 2015. Simulation and experimental design methods for job shop scheduling with material handling: a survey. *Int J Adv Manuf Technol*, **80**, 233–243.

Yoo, Jaewook, & Lee, Ik Sun. 2016. Parallel machine scheduling with maintenance activities. *Computers & Industrial Engineering*, **101**, 361–371.