Parallel AGV Scheduling Problem with Battery Constraints (ASP-BC)

Implementation from: "The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach", European Journal of Operational Research 307 (2023)

Alessandro Querenghi & Cristina Visentin

Mathematical Optimization, 15/09/2025

Problem Description

GENERAL INTRO

ASP-BC: scheduling of resource transferring from a location to many others through Automated Guided Vehicles (AGVs)

- **Considering battery constrains.**

“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**”.

Problem Description

TRANSFER JOBS AND CHARGING OPERATIONS

Transfer jobs consists in moving packages of material to a workstation.

- Initially positioned at the **center** of the warehouse (pickup point)
- Each package contains all and only what is needed by a workstation (**atomic**)
- Match 1:1 between transfer jobs and workstations
- Two parameters: **duration** (round trip + loading/unloading operations) and **weight** of the package

The main aim is to transfer **the right amount** of **the right material** to **the right place** at **the right time.**

Problem Description

AGVs

An AGV fleet is a part of an AGV system used for horizontal movement of materials from a pickup point to delivery ones.

- Used within a warehouse to deliver materials from the center of the warehouse (pickup points) **to a set of workstations** (delivery points)
- Deliver **one and only package per travel**
- #AGVs $\ll$ #workstations
- Travel at a constant speed

Problem Description

AGVs BATTERY

- AGVs charge in the pickup point
- Completely **charged at the beginning** of the transfer operations
- Battery is affected by travel time and weight carried
- Charges have a **predefined duration** (charges seen as special transfer jobs)
- Vehicles must be **completely recharged**, but never completely depleted

Problem Description

MAKESPAN

Defined as the **time to supply** a set of packages to a set of workstations

- AGVs are independent form each other and work in **parallel**
- Not the summation of the work duration of each AGV
- It is the **work duration of the AGV that takes the longest to complete its transfer jobs**

Minimizing the makespan through job and charges scheduling leads to a better management that improvs the overall productivity and avoids delays in production… aka minimize costs for the manufacturer.

Outline

Exact method

- Mathematical formulation

Matheuristic

- Three steps matheuristic

Variant of the exact method and matheuristic

- Allowing for variable charge time depending on remaining battery

Results and scalability analysis

Mathematical Formulation

Formulation

SETS AND PARAMETERS

Let

- J be the set of **transfer jobs**,
- M the set of **AGVs** and
- R the set of **charging jobs**

Each job $j \in J$ has two parameters

- e_j which denotes the **energy consumption**
- d_j which is the **job duration**

Vehicles $m \in M$ have all the same **battery capacity** b and **charging time** t.

Formulation

DECISION VARIABLES

$x_{jm} \in \{0,1\}$	$\forall j \in J,$ $\forall m \in M$	Transfer job j performed by AGV m
$q_{rm} \in \{0,1\}$	$\forall r \in R,$ $\forall m \in M$	Charging job r performed by AGV m
$y_{jrm} \in \{0,1\}$	$\forall j \in J,$ $\forall r \in R,$ $\forall m \in M$	Transfer job j performed by AGV m right after charging job r
$c_{\max} \in \mathbb{R}^+$		Makespan of the transfer process

Formulation

OBJECTIVE FUNCTION AND CONSTRAINTS

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

Minimize the makespan of the whole transfer process

$$c_{\max} \geq \sum_{j \in J} d_j x_{jm} + \sum_{r \in R \setminus \{1\}} t q_{rm} \qquad \forall m \in M \tag{1}$$

For each vehicle m, the time spent by m to perform its jobs and the time spent to recharge between jobs must be consistent with the makespan.

Since AGVs are already charged at the beginning the first charge operation $r = 1$ is not considered.

Formulation

CONSTRAINTS

$$\sum_{m \in M} x_{jm} = 1 \qquad \forall j \in J \tag{2}$$

Each transfer job j must be assigned to a single AGV m

$$\sum_{r \in R} y_{jrm} = x_{jm} \qquad \forall j \in J, m \in M \tag{3}$$

If a transfer job j is performed by AGV m, it must be preceded by a charging job r on the same AGV

Formulation

CONSTRAINTS

$$2y_{jrm} \leq x_{jm} + q_{rm} \qquad \forall r \in R, j \in J, m \in M \tag{4}$$

Variable upper bound on variables y_{jrm}.

It ensures consistency between the three variables

$$\sum_{j \in J} y_{jrm} \cdot e_j \leq b \qquad \forall m \in M, r \in R \backslash \{1\} \tag{5}$$

Battery constraint: the summation of all jobs performed after a charging job r must not exceed the battery limit b of the respective AGV m

Formulation

CONSTRAINTS

$$q_1^m = 1 \qquad \forall m \in M \tag{6}$$

Each AGV starts fully charged

$$q_r^m \leq q_{r-1}^m \qquad \forall r \in R \backslash \{1\}, m \in M \tag{7}$$

Symmetry breaking constraints: charging job r on vehicle m must be preceded by charging job $r-1$

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

$$y_{jrm} \in \{0,1\} \qquad \forall j \in J, \forall r \in R, \forall m \in M \tag{9}$$

$$q_{rm} \in \{0,1\} \qquad \forall r \in R, \forall m \in M \tag{10}$$

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

Matheuristic

3 Step Matheuristic

1. **Bin Packing Problem**: find the minimum number of charging operations required to perform all transfer jobs
2. **Bottleneck Generalized Assignment Problem**:
   - **Constrained**: assign transfer jobs to AGVs
   - **With charges**: assign charging operations to AGVs by using the solution from BPP
3. **Local search**

Bin Packing Problem

Bin Packing Problem

- Find a **partition** of items (transfer jobs) into bins (charging operations) such that the **number of bins is minimized**

Bin Packing Problem

VARIABLES

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

Bins (charging operations) used

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

Assignment of item j (transfer jobs) to bin r

Bin Packing Problem

OBJECTIVE

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

Minimize the number of bins (charging operations) used

Bin Packing Problem

CONSTRAINTS

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

Each transfer job j is assigned once and only once

$$\sum_{j \in J} e_j \cdot \chi_{rj} \leq b \cdot \gamma_r \qquad \forall r \in R$$

Transfer job assignment doesn't exceed battery capacity.

Transfer job assignment is consistent with partition

$$\gamma_r \leq \gamma_{r-1} \qquad \forall r \in R \backslash \{1\}$$

Symmetry breaking constraint

Bin Packing Problem

LOWER BOUND

- The BPP solution allows to compute a **lower bound** for the ASP-BC:

$$LB = \max\left\{ \left\lceil \frac{\max\{0, \zeta - |M|\} \cdot t + \sum_j d_j}{|M|} \right\rceil, \left\lceil \frac{\max\{0, \zeta - |M|\}}{|M|} \right\rceil \cdot t \right\}$$

- The first part is the total time required to perform all transfer operations in parallel
- The second part takes into account the indivisibility of the charging operations

BGAP

BGAP Constrained

(BGAP-C)

- Assign all transfer jobs to AGVs **without using additional charging operations**, while minimizing the makespan.

- If the number of charging operations doesn't exceed the number of AGVs, that is:

$$\zeta \leq |M|$$

- A **feasible solution for ASP-BC** is obtained by solving a Constrained Bottleneck Generalized Assignment Problem: **BGAP-C**

BGAP - C

VARIABLES

$$x_{jm} \in \{0,1\} \qquad \forall j \in J, m \in M$$

Transfer job j assigned to AGV m

$$c_{\max} \in \mathbb{R}^+$$

Makespan of the transfer process

BGAP - C

OBJECTIVE

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

Minimize makespan of the whole transfer process

Find the best solution without using additional charging operations

BGAP - C

CONSTRAINTS

$$c_{\max} \geq \sum_{j \in J} d_j \cdot x_{jm} \qquad \forall m \in M$$

Ensure consistency of the makespan

$$\sum_{m \in M} x_{jm} = 1 \qquad \forall j \in J$$

Jobs are assigned once and only once

$$\sum_{j \in J} e_j \cdot x_{jm} \leq b \qquad \forall m \in M$$

Job assignment doesn’t exceed battery limit

BGAP With Charging Operations

(BGAP-R)

- **Assign charging operations** to AGVs (from the solution of BPP)

- If the number of charges exceeds the number of AGVs, that is:

$$\zeta > |M|$$

- A **feasible solution for ASP-BC** is obtained by solving a Bottleneck Generalized Assignment Problem with Charging operations: **BGAP-R**

BGAP - R

VARIABLES

$$\theta_{rm} \in \{0,1\} \qquad \forall r \in R, m \in M$$

Assign charge operation r to AGV m

$$c_{\max} \in \mathbb{R}^+$$

Makespan of the transfer process

BGAP - R

OBJECTIVE

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

Minimize makespan of the transfer process

BGAP - R

CONSTRAINTS

$$c_{\max} \geq \sum_{r \in R} D_r \cdot \theta_{rm} - t \quad \forall m \in M$$

Ensure consistency of the makespan

where $$D_r = \sum_{j \in J} d_j \cdot \chi_{jr} + t$$

Duration of each charge operation

$$\sum_{r \in R} \theta_{rm} = 1 \qquad \forall m \in M$$

Charging operations are assigned once and only once

Local Search

Local Search

MECHANISM

- Starting from the optimal solution of the BGAP
- Compute the **saving** obtained by performing **one of three operations**:
  - **Add** s_a: take job j from AGV m_1 and give it to AGV m_2 after a newly added charging operation
  - **Remove** s_r: take job j from AGV m_1 and give it to AGV m_2 on an available charging operation
  - **Swap** s_s: swap two jobs from two different AGVs

Local Search

ALGORITHM

1. Set the saving $s^* = 0$ and compute $c_{\max}$
2. Compute each s_a. If $s_a > s^*$ update s^* and (x, y)
3. Compute each s_r. If $s_r > s^*$ update s^* and (x, y)
4. Compute each s_s. If $s_s > s^*$ update s^* and (x, y)
5. If $s^* > 0$:

    apply the update,

    then go back to first step,

   else stop

- Small variant

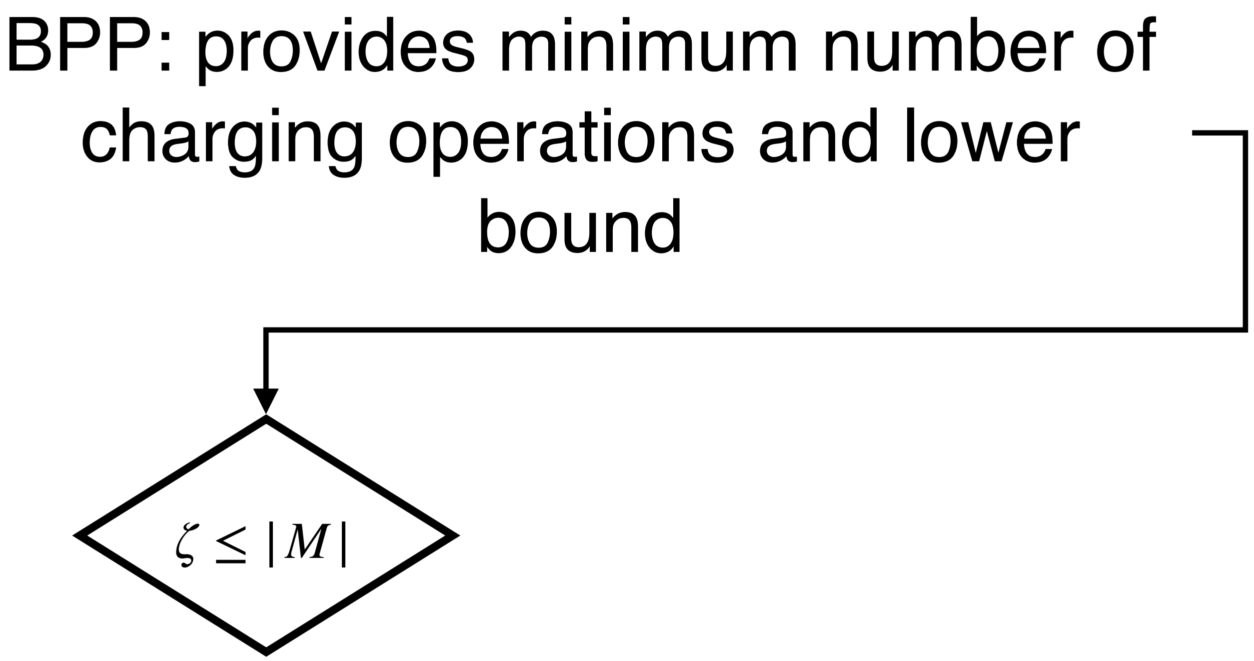
Matheuristic

ALGORITHM

BPP: provides minimum number of charging operations and lower bound

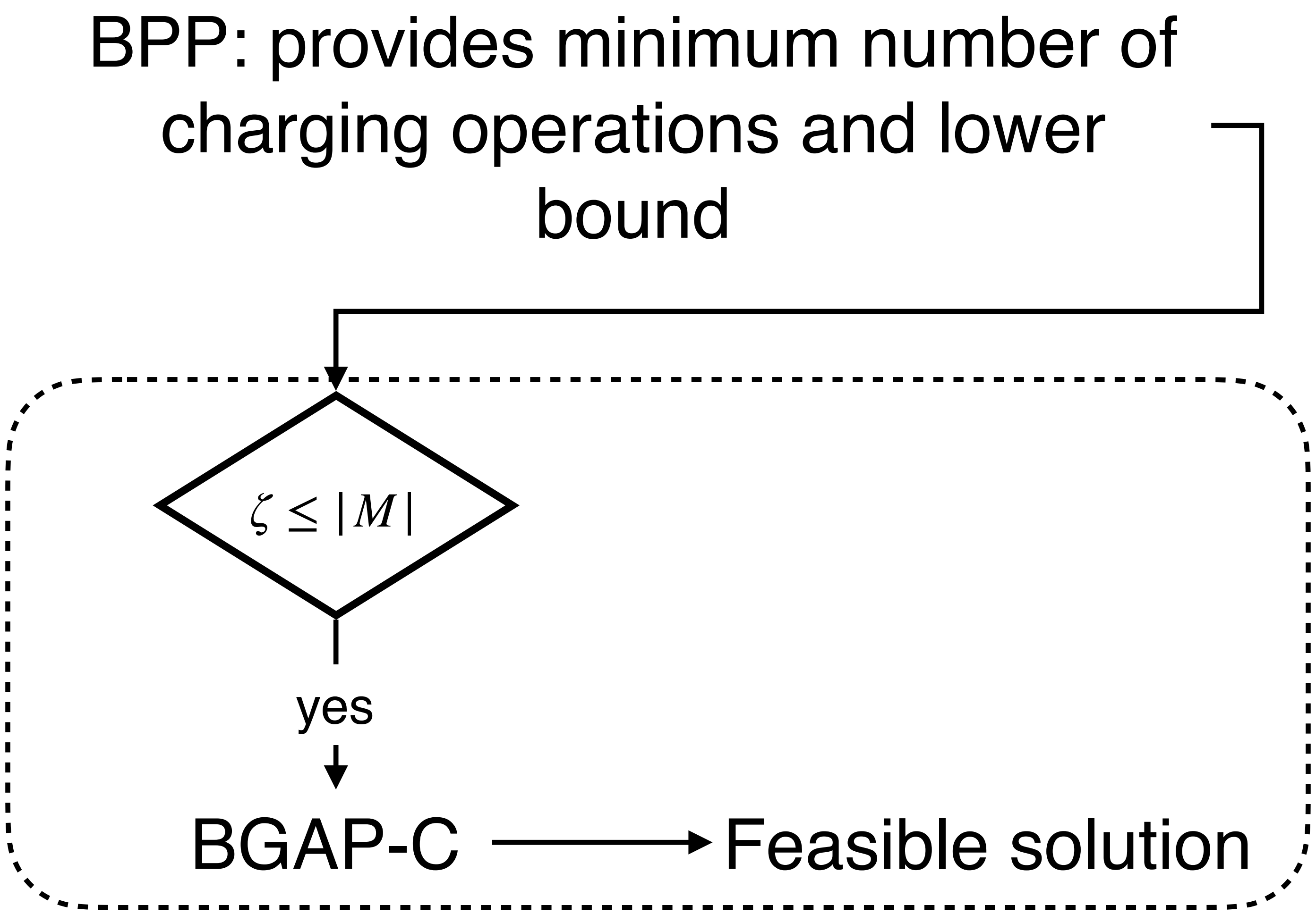
Matheuristic

ALGORITHM



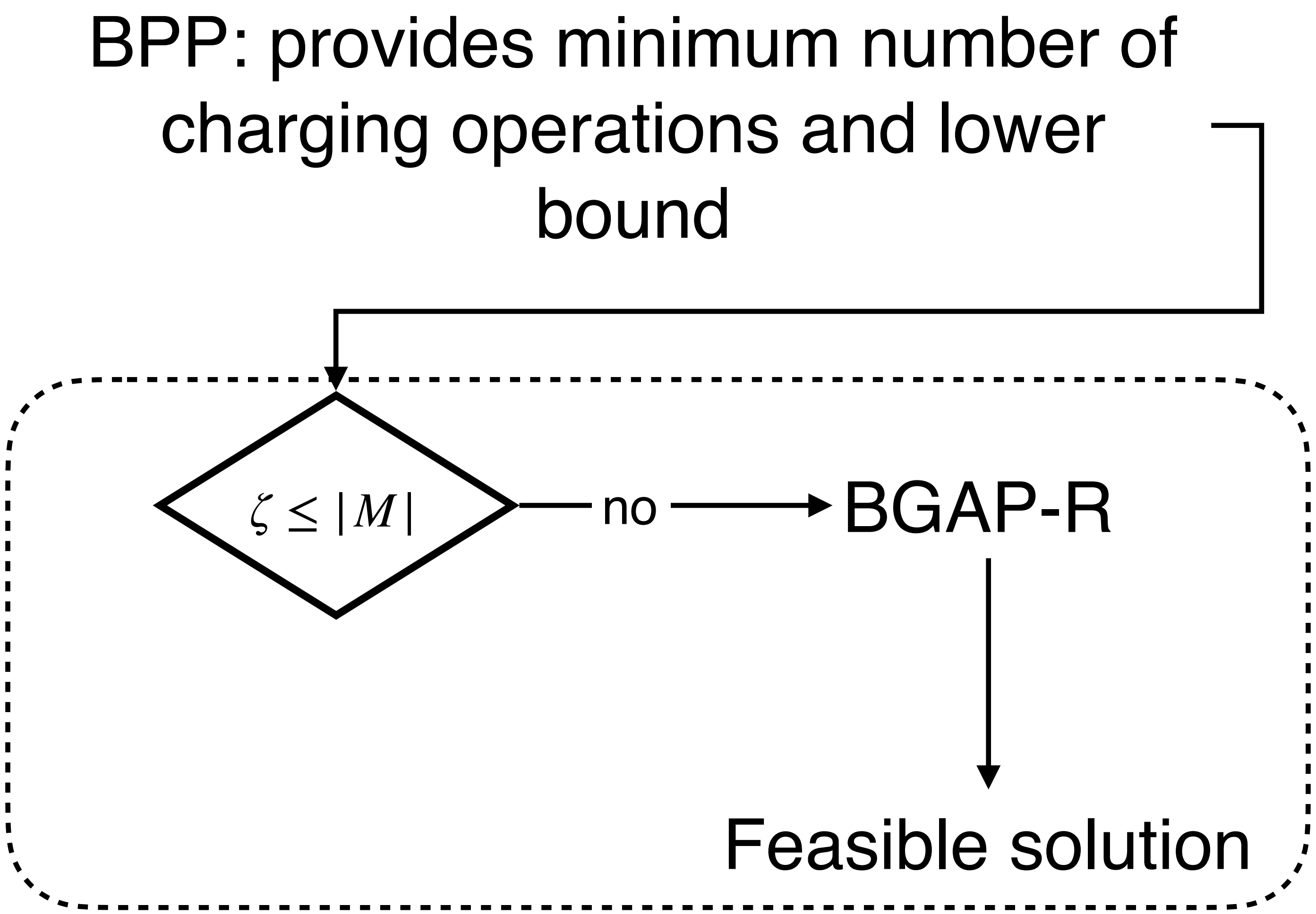
Matheuristic

ALGORITHM



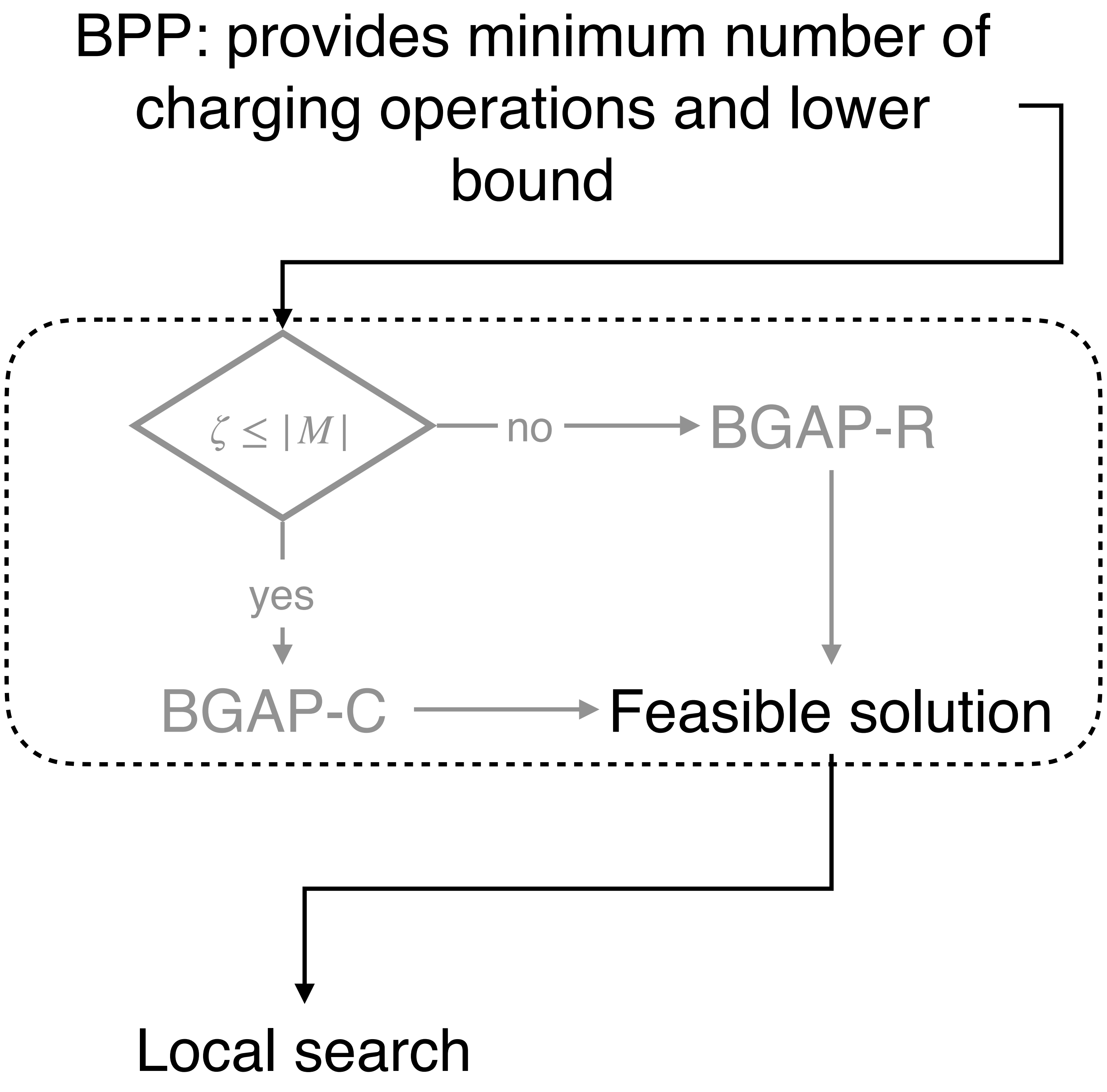
Matheuristic

ALGORITHM



Matheuristic

ALGORITHM



Variable Charge Approach

ASP-BC Variable Charge

Computational results sustain the **assumption of the fixed charging time** because when AGVs return to the pickup point to charge, the residual energy is, on average, about 5% of the battery capacity.

- Both the exact method and the matheuristic can be adapted to the case in which the charging time depends on the **residual energy** of a vehicle.

ASP-BC Variable Charge

DIFFERENCES WITH EXACT METHOD

- Remove parameter t (charging time)
- Add parameter τ which indicates the **charging time per energy unit**

- Update makespan constraint:

$$c_{\max} \geq \sum_{j \in J} d_j x_{jm} + \tau \sum_{j \in J} \sum_{r \in R \setminus \{n\}} e_j y_{jrm} q_{r+1m} \qquad \forall m \in M$$

For each vehicle m, the time spent by m to perform its jobs and the time of each charging operation which depends on the energy expenditure must be consistent with the makespan.

- Ignore the last charging operation since AGVs are charged at night.
- Not linear.

ASP-BC Variable Charge

DIFFERENCES WITH MATHEURISTIC

- Remove parameter t (charging time)
- Add parameter τ which indicates the **charging time per energy unit**

- **New lower bound formula** no longer requiring to solve BPP.

- Charge job duration has to be modified in BGAP-R to **ignore the charging time** of the **last charging job**.

- Saving expressions must be modified to **consider the variable charging times**.

Scalability Test

Scalability Test

INSTANCES AND AXIS OF ASSESSMENT

- **1080 instances**
- Parameters:
  - Number of **jobs** $J \in [50,100,150,200]$
  - Number of **AGVs** $M \in [2,5,10]$
  - Transfer job time **duration** $d \in \mathbb{N}^+$
  - Transfer job **energy cost** $e \in \mathbb{R}^+, \mu_e \in [1,2,4]$

Considerations on:

- **Percentage gap from optimum**
- **Computational time**

Scalability Test

MATHEURISTIC - % GAP FROM LOWER BOUND

AGVs/Jobs	50	100	150	200
2	0.00	0.00	0.00	0.00
5	0.92	0.01	0.00	0.00
10	5.97	0.81	0.11	0.04

- As the instance size **grows**, so does the matheuristic's **performance**.
- The number of AGVs **negatively** impacts results.

Scalability Test

MATHEURISTIC - COMPUTATIONAL TIME (s)

AGVs/Jobs	50	100	150	200
2	0.12	13.60	58.97	75.39
5	5.38	15.88	55.29	80.65
10	2.96	21.21	52.76	74.16

- Computational time **increases** with the number of jobs, but…

Scalability Test

MATHEURISTIC - COMPUTATIONAL TIME (s)

Average job cost/Jobs	50	100	150	200
1	0.05	0.16	0.45	3.94
2	0.05	4.83	5.86	4.89
4	9.15	48.21	162.01	220.92

- It mostly **depends on the average job cost**.
- This happens because **more expensive** jobs require **more charging operations.**

Scalability Test

EXACT METHOD - % GUROBI GAP

AGVs/Jobs	50	100	150
2	0.00	0.22	1.21
5	0.06	0.88	8.01
10	0.20	2.09	15.12

- As the instance size **grows**, the **performances** of the exact method **decrease**.
- Opposite with respect to the matheuristic.

Scalability Test

EXACT METHOD - COMPUTATIONAL TIME (s)

AGVs/Jobs	50	100	150
2	16.13	196.38	405.35
5	70.73	326.57	512.53
10	173.68	416.86	576.95

- Computational time **increases** with the **number of jobs** and also with the **number of AGVs.**

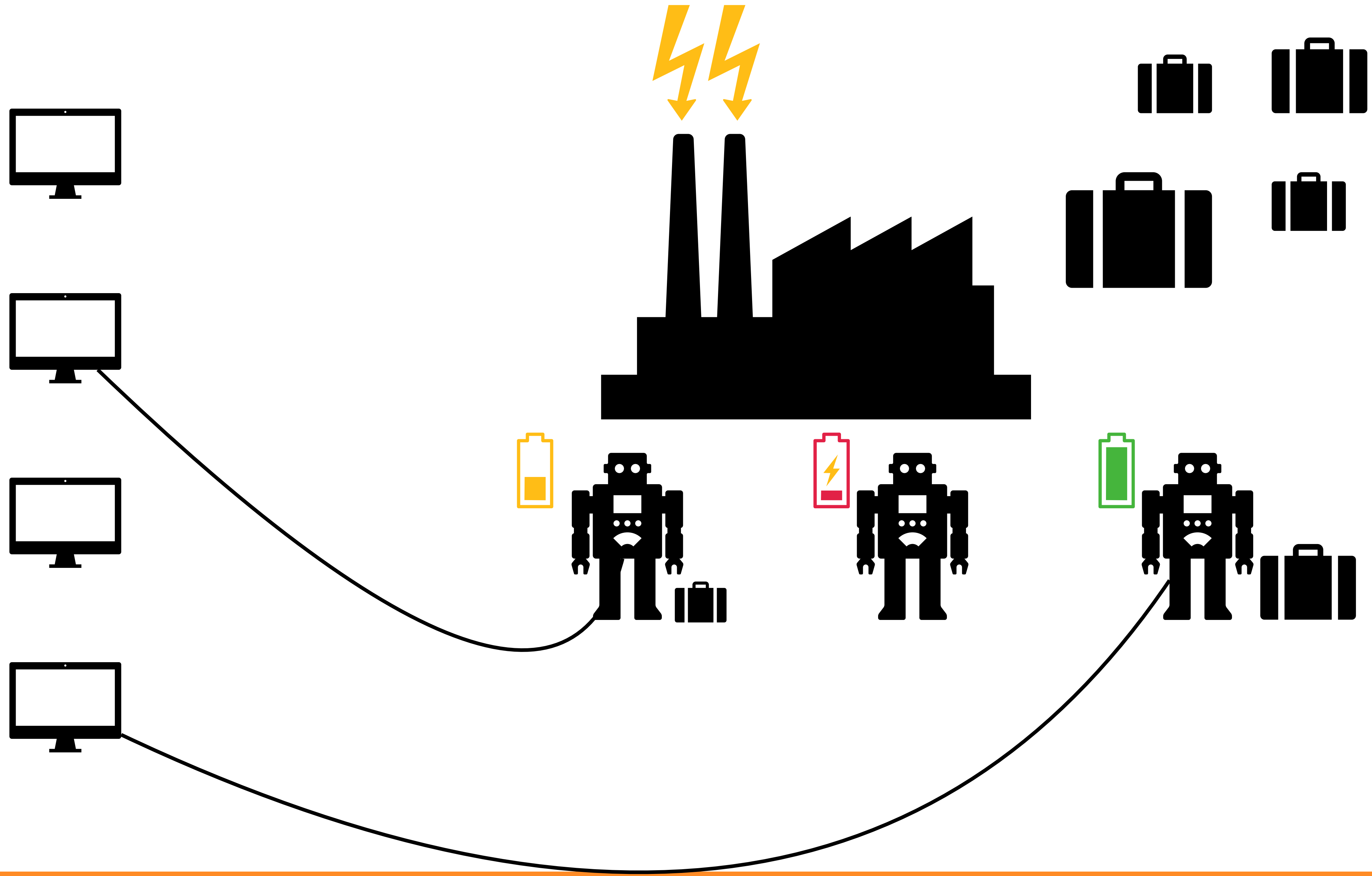
Scalability Test

EXACT METHOD - COMPUTATIONAL TIME (s)

Average Job Cost/Jobs	50	100	150
1	3.50	80.08	364.04
2	67.66	337.35	537.32
4	213.77	515.19	573.39

- As seen for the matheuristic, computational time depends heavily on the **average job cost.**

Mancava il confronto tra euristica e metodo esatto



Local Search

SAVE ADD

$$s_a(m_1, j, m_2) = \max\{0, c_{\max} - \max_{m \notin m_1, m_2} \{c_{m_1} - d_j, c_{m_2} + d_j + t, c_m\}\}$$

$$c_m = \sum_{j \in J} d_j \cdot x_{jm} + \sum_{r \in R \setminus \{1\}} t \cdot q_{rm}$$

Duration of the transfer process on AGV m

Local Search

SAVE REMOVE

$$s_r(m_1, j, m_2, r) = \max\{0, c_{\max} - \max_{m \notin m_1, m_2} \{c_{m_1} - d_j, c_{m_2} + d_j, c_m\}\}$$

$$\sum_{i \in J} e_i \cdot y_{rim_2} + e_j \leq b$$

If charge operation r can accommodate job j

Local Search

SAVE SWAP

$$s_s(m_1, r_1, j_1, m_2, r_2, j_r) = \max\{0, c_{\max} - \max_{m \notin m_1, m_2} \{c_{m_1} - d_{j_1} + d_{j_2}, c_{m_2} + d_{j_1} - d_{j_2}, c_m\}\}$$

$$b \geq \sum_{i \in J} e_i \cdot y_{r_1 i m_1} - e_{j1} + e_{j_2}$$

and

$$b \geq \sum_{i \in J} e_i \cdot y_{r_2 i m_2} + e_{j1} - e_{j_2}$$

ASP-BC Variable Charge

DIFFERENCES WITH MATHEURISTIC

- New lower bound formula:

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

- Charge job duration in BGAP-R

$$D_r = \sum_{j \in J} (d_j + \tau e_j) \chi_{rj}$$

ASP-BC Variable Charge

CONSTRAINT LINEARIZATION

Define a new variable

$$w_{jrm} \in \{0,1\}$$

such that

$$w_{jrm} \leq y_{jrm}$$
$$w_{jrm} \leq q_{r+1m}$$
$$w_{jrm} \geq y_{jrm} + q_{r+1m} - 1$$

$$\forall r \in R \backslash \{n\}, j \in J, m \in M$$