Int J Adv Manuf Technol (2010) 46:1239–1251
DOI 10.1007/s00170-009-2176-7

ORIGINAL ARTICLE

# A constructive heuristic for the integrated scheduling of machines and multiple-load material handling equipment in job shops

**Tamer F. Abdelmaguid · Ashraf O. Nassef**

Received: 27 March 2009 / Accepted: 15 June 2009 / Published online: 3 July 2009


**Abstract** This paper extends the traditional job shop scheduling problem (JSP) by incorporating the routing and scheduling decisions of the material handling equipment. It provides a generic definition and a mixed integer linear programming model for the problem considering the case of heterogeneous multiple-load material handling equipment. A constructive heuristic is developed for solving the problem. This heuristic is based on the well-known Giffler and Thompson's algorithm for the JSP with modifications that account for the routing decisions of the material handling equipment and their effect on the start times of the manufacturing operations. Different dispatching rules are integrated into the heuristic, and experiments are conducted to study their effect on the makespan along with the determination of the computational time requirements of the developed heuristic.

**Keywords** Job shop scheduling · Material handling · Vehicle routing · Pickup and delivery problem

T. F. Abdelmaguid
Mechanical Design and Production Dept., Faculty of Engineering, Cairo University,
Giza 12613, Egypt

T. F. Abdelmaguid (✉)
Industrial Engineering Dept., College of Engineering, King Saud University,
P.O. Box 800, Riyadh, Kingdom of Saudi Arabia
e-mail: tabdelmaguid@alumni.usc.edu

A. O. Nassef
Mechanical Engineering Dept, American University in Cairo,
Cairo, Egypt

## 1 Introduction

The job shop scheduling problem (JSP) is a traditional decision making problem that is encountered in low volume–high variety manufacturing systems which are known as job shops. Since it had been introduced in the literature in the 1950s, a large number of solution techniques have been proposed aiming to conquer this NP-hard problem [1]. The traditional mathematical models and solution techniques that are developed for the JSP consider only the scheduling decisions for operations on machines and neglect the material transportation tasks. This is unjustified since the capacities of the material handling equipment used in job shops are limited, and the transportation times of parts are dependent on their routings which differ considerably. Therefore, as shown by Smith et al. [2], the routing and scheduling decisions of material handling equipment in job shops have significant impact on the objectives of the scheduling of manufacturing operations.

According to the Material Handling Industry of America, material handling in manufacturing systems refers to all operations related to the movement, storage, control, and protection of materials throughout the manufacturing processes. In the literature, issues related to the proper selection of material handling systems at the design stage have been studied by many researchers [3–6]. At the operational planning level, issues related to the routing and scheduling of the material handling equipment (MHEs) during the manufacturing process have been investigated mostly for automated systems in which these decisions are conducted via a central computer. Examples of the MHEs used in automated manufacturing systems are robots, automated guided vehicles (AGVs), and automated hoists.

Early studies of material handling tasks and their effect on the scheduling decisions considered only a transportation time lag for each job. A transportation time lag can be viewed as the time needed to move a job between two machines assuming that material handling equipment is always ready and available. Examples of such an early consideration are found in [7, 8]. They extend the well know Johnson's algorithm [9] for solving the two-machine flow shop scheduling problem with transportation time lags for jobs. In the flow shop environment where parts follow the same processing sequence, Raman et al. [10] studied the simultaneous scheduling of machines and AGVs. Their model contains a restriction that AGVs always return to the load or unload station after transferring a load, which simplifies the scheduling decisions. They formulated the problem as an integer program and presented a solution algorithm based on project scheduling concepts under resource constraints. Another study considered the case of simultaneous scheduling of machines and a single robot in a flow shop environment [11].

In the electronics industry, the electroplating process of printed circuit boards represents another form of combined scheduling of manufacturing operations and material handling tasks in a flow-shop environment. In these systems, there is an interface between the manufacturing process, which is in this case the electroplating operations conducted in tanks and the movement and control of automated hoists which are the material handling equipment used in these systems [12–17].

In this paper, the integrated problem of scheduling manufacturing operations and material handling tasks in job shops is addressed with the consideration of multiple-load MHEs that are capable of transporting the materials of more than one job simultaneously. This study is motivated by real applications in flexible manufacturing systems (FMSs) which utilize multiple-load AGVs and in which part routes are dissimilar as in job shops. In such FMSs, machines and AGVs are controlled by a central computer through which manufacturing and AGV routing orders are dispatched. The existence of such central control system facilitates the implementation of pre-prepared offline schedules. This paper attempts to provide an efficient solution methodology that can provide such schedules in small computational time. However, this study is not confined to AGVs as the methodology used can be easily suited for other types of MHEs as well.

In the literature, there are two main lines on how such an integrated scheduling problem in a job shop environment is addressed. The first one focuses on studying dispatching rules for the MHEs. Simulation models are commonly used in this approach to imitate the online scheduling decisions and study the effect of different MHE dispatching rules on the overall objectives of the scheduling problem [18, 19]. The second line is an offline approach that deals with the static, deterministic version of the problem through defining mathematical programming models and algorithms to formulate and solve it. We focus on the literature of this line of research as we follow it in this study.

Bilge and Ulusoy [20] studied the simultaneous scheduling of machines and identical single-load AGVs in a job shop framework. They introduced a mixed integer nonlinear programming model with the objective of minimizing the makespan and proposed a solution heuristic based on the sliding time window (STW) approach which is developed by Ferland and Fortin [21] for solving the vehicle scheduling problem. Ulusoy et al. [22] developed a genetic algorithm (GA) approach for solving the same problem. Their GA implementation generates better results compared to the STW heuristic. An improvement to this GA representation was proposed by Abdelmaguid et al. [23]. They employed a greedy search algorithm for solving the vehicle routing and scheduling part of the problem, while the machine scheduling part has a similar GA representation as found in the operations-based GA coding scheme for the job shop scheduling problem. This problem has been studied further by Reddy and Rao [24] for the case of multiobjective optimization. They developed a GA approach that can provide a set of nondominated solutions for the minimization of makespan, mean flow time, and mean tardiness simultaneously.

Jawahar et al. [25] developed a heuristic approach that integrates vehicle dispatching rules into the scheduling decisions. Their study focuses on a specific FMS configuration that contains six work centers and a single AGV. Anwar and Nagi [26] studied the simultaneous scheduling problem of machines and AGVs in a job shop framework with the consideration of multilevel assembled products. With the objective of minimizing the makespan, they provided an MILP model and developed a heuristic search technique based on the critical path method found in the project scheduling literature. Their model considers only the case of single-loaded transporters.

Khayat et al. [27] presented a constraint programming model for the simultaneous scheduling problem of machines and identical single-load AGVs. Hurink and Knust [28] studied the job shop scheduling problem with the consideration of transportation tasks performed by a single robot. They extended the disjunctive graph model [29] for representing the problem and developed two different tabu search heuristics for solving it. Caumond et al. [30] developed an MILP model for the scheduling problem in flexible manufacturing systems with one vehicle. They considered the case of limited input and output buffer capacities.

Recently, more interest has been given to the case of multiple-load AGVs. Different dispatching rules were investigated and tested through simulation models in an online approach [31, 32]. Since the offline approach is easier to implement as the timing for all manufacturing processes and the routing of material handling equipment are made available at the beginning of the production plan, using it in such a complex problem would be appealing in real-life applications. Accordingly, we address the offline scheduling problem for both machines and multiple-load MHEs in a job shop environment. This integrated machine and material handling scheduling problem is referred to as the Job Shop Scheduling Problem with Material Handling (JSPMH).

The contribution of this paper is twofold. First, an MILP model is developed for the studied problem. This model can be solved using commercial MILP solvers for small-sized cases to develop an offline optimal solution. Second, a constructive heuristic is developed for providing efficient solutions to large instances of the problem. To the best of our knowledge, this is the first attempt to formulate a mathematical model and to provide an offline scheduling algorithm for the JSPMH. The developed model and solution algorithm are generic in the sense that they are not restricted to the case of AGVs as they can be suited easily to any type of MHEs.

The rest of this paper is organized as follows. In Section 2, a description of the problem is provided followed by the elements of the developed MILP model in Section 3. In Section 4, the constructive heuristic is presented. Numerical results are provided in Section 5 and the conclusion and directions for future research are presented in Section 6.

## 2 Problem description

The studied job shop consists of a set $M$ of different machines that perform the manufacturing tasks in addition to load and unload stations which respectively serve as the input and output ports for the job shop. There is a set $J$ of jobs waiting to be processed, where each job corresponds to a batch for a specific product with a predetermined batch size. Raw materials of all jobs are available at the beginning of the schedule at the load station, and finished parts should be delivered to the unload station. The generic term *processing center* is used to refer to either a machine or a load/unload station, and set $PC$ is defined as the set of all processing centers. Jobs follow different processing routes among the machines; no preemption is allowed, and no more than one job can be processed on a machine at the same time. We assume a static case in which at the beginning of the schedule, all manufacturing and material handling resources do not have current or pre-assigned tasks, and no additional jobs are expected to arrive during the implementation of the schedule.

Each job consists of an ordered list of operations that represents its predetermined processing route on machines. In addition, two dummy operations with zero times are placed at the beginning and at the end of each job's ordered list of operations. They respectively represent the start of the loading task of the job's raw material to the MHE at the load station and the end of the unloading task of the job's finished parts from the MHE at the unload station. These dummy operations are defined for more convenient modeling. Each operation is identified by an index number. We denote $I = \{1, 2, \ldots, n\}$ as the set of all operations' indexes. The operations' indexes are assigned such that for job $k \in J$, the subset of consecutive indexes $I_k = \{\alpha_k, \alpha_k + 1, \ldots, \omega_k\} \subseteq I$ includes the indexes of all the operations belonging to that job, and the operation with the lower index is to be processed first. For operation $i$, the job to which it belongs is denoted $jb(i)$, and its processing center is denoted $pc(i)$. In addition, we define subsets of indexes that include only machining operations and exclude the first and last dummy operations. These subsets are defined as $\underline{I}_k = \{i : \alpha_k < i < \omega_k\}$ for job $k \in J$ and $\underline{I} = \cup_{k \in J} \underline{I}_k$.

Each processing center is associated with a buffer having sufficiently large capacity. Buffers work as temporary storage locations for materials before and after processing. In addition to the machine setup and machining times, each operation $i$ requires a loading time for its raw material from the buffer associated with its designated processing center, $pc(i)$, and an unloading time for its machined material from $pc(i)$ to the buffer. The machine setup, loading, machining, and unloading tasks are assumed to be consecutive with no preemption and the summation of these times for operation $i$ is referred to as its processing time and denoted $p_i$. Each element of the processing time of an operation represents the summation for all parts in the batch, and machine setup times are assumed to be independent on the operations' processing sequence. We note that the processing centers of the first and last operations of every job are the load and unload stations respectively and they have zero $p_i$ values.

There is a set $H$ of MHEs that are ready to perform the material handling and transportation tasks between the processing centers. Each $h \in H$ is originally located at $O(h)$ and should park at its final destination $F(h)$ after completing its assigned trips. $O(h)$ and $F(h)$ denote physical locations in the shop or tool center point positions in the case of robots. They may refer to the same location, and they could be the location of any processing center in the shop or rather special parking/charging station.

Each operation $i \in \underline{I}$ is associated with unloading (delivery of raw material) and loading (pickup of machined material) tasks from the MHE to the buffer at $pc(i)$ and vice versa. The unloading and loading times of MHE $h$ for operation $i$ are denoted $u_i^h$ and $l_i^h$, respectively. The first dummy operation of each job, $i \in \{\alpha_k : k \in J\}$, is associated only with a loading task for picking up the raw material from the load station, and the associated loading time is denoted $l_i^h$ for MHE $h$. Similarly, the last dummy operation of each job, $i \in \{\alpha_k : k \in J\}$, is associated with an unloading task for delivering the finished parts to the unload station and the associated unloading time is denoted $u_i^h$. The assumption of sufficiently large capacity for buffers means that a buffer can serve any number of MHEs in addition to its associated processing center simultaneously.

Each MHE $h$ has a capacity denoted $C_h$ which can be defined in terms of weight, volume, or number of containers or holding places available. The capacities of the MHEs are not necessarily identical. An MHE is said to be used in transporting the material for operation $i$ of job $k$, where $i > \alpha_k$, if it performs the material pickup from $pc(i–1)$ and transports and delivers it to $pc(i)$. During that trip, operation $i$ consumes $c_i^h$ units from the capacity of MHE $h$. Each MHE can perform multiple pickups for any number of different jobs given that its capacity limit is not exceeded. It is assumed that each MHE has zero load at the beginning of the schedule. It is also assumed that the number of parts in any batch is predetermined in such a way that there exists an MHE with sufficient capacity to transport its material in one trip, and there is a restriction that the transportation of a batch cannot be split into more than one trip.

Due to certain technological constraints, an MHE may not be able to transfer the material for certain operations. These technological constraints may be related to the layout of the shop and the design of the guided paths or zones for the material handling equipment which may prevent an MHE from reaching specific processing centers or locations in the shop. They may also be related to the shape or size of the transported parts, where some MHE may not be suitable for handling them. We assume that for every operation, there is at least one MHE that can conduct its material handling tasks.

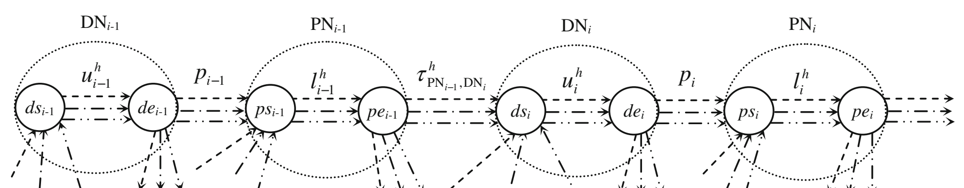
In the JSPMH, the MHEs are not necessarily identical, which means that different speeds and consequently different travel times between similar locations may exist. The MHE travel times are first defined between physical locations, for instance between the load station and the milling machine, and then based on the operations' assigned processing centers, they are converted into arc lengths between nodes in the network representation described in the following section. It is assumed that the MHE system is deadlock and accident-free.

It is required to determine the processing sequence of operations on each machine and the MHE routings such that all the aforementioned constraints are satisfied. The main objective is to minimize the makespan which is precisely defined in the following section. The processing sequence part of the problem has the same structure as the traditional job shop scheduling problem [1], while the MHE routing part is similar to the capacitated vehicle routing problem with pickup and delivery [33]. Both parts are known to be NP-hard leading to a stronger NP-hard JSPMH problem. A network model for the studied problem along with a branch-and-bound algorithm was introduced earlier in [34]. In the following section, we build upon that network model and develop an MILP model which can be used to solve small-sized instances.

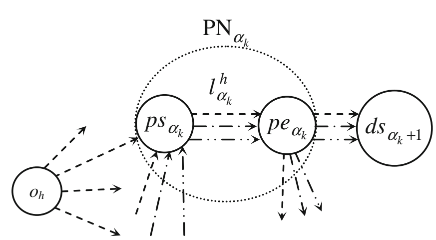
## 3 Mathematical model

Figure 1 illustrates the main elements of the network model introduced earlier in [34] for the studied problem. The developed MILP model is based on this network representation. In the network model, each operation $i \in \underline{I}_k$ of job $k$, as illustrated in Fig. 1a, is associated with two nodes $\mathrm{DN}_i$ and $\mathrm{PN}_i$ to respectively represent the material delivery and

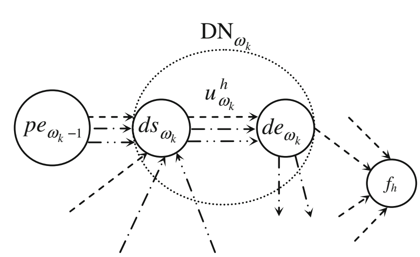
**Fig. 1** Elements of the network model for the JSPMH



(a) Operations processing sequence for a given job $k$, and their corresponding pickup and delivery nodes, where $\alpha_k < i$ & $i\text{-}1 < \omega_k$



(b) Nodes associated with the first dummy operation at the beginning of job $k$



(c) Nodes associated with the last dummy operation at the end of job $k$

pickup events, before and after the processing at its designated processing center $pc(i)$. To further show the details of the unloading (loading) times at each delivery (pickup) node, two nodes are defined to be subsumed by each node $\mathrm{DN}_i$ ($\mathrm{PN}_i$) and connected by arcs with arc lengths equal to the unloading (loading) times for every MHE that can reach that node. Accordingly, each delivery node $\mathrm{DN}_i$ encompasses two nodes, $ds_i$ and $de_i$ to signify the start and end of the delivery process, respectively. These two nodes are connected by a set of arcs, where each arc is defined for an MHE $h$ that can handle the delivery task of operation $i$, and the arc has a length equals the unloading time $u_i^h$. In addition to signifying the end of the delivery process, node $\mathrm{de}_i$ represents the start of the processing of operation $i$ on $pc(i)$. Similarly, each pickup node $\mathrm{PN}_i$ encompasses two nodes, $ps_i$ and $pe_i$, to represent the start and end of the pickup process. Each arc in the set of arcs that connects these two nodes has a length equals the loading time $l_i^h$ for the MHE $h$. The first dummy operation of a job $k$, as illustrated in Fig. 1b, is associated with only a pickup node, denoted $\mathrm{PN}_{\alpha_k}$; where in this case, node $ps_{\alpha_k}$ signifies the start and the end times of this operation. Meanwhile, as illustrated in Fig. 1c, the last dummy operation is associated with only a delivery node, denoted $\mathrm{DN}_{\omega_k}$, and node $de_{\omega_k}$ signifies the start and end times of this operation. Integrating the start and end nodes for both delivery and pickup operations is necessary to facilitate the modeling of the MHE routing constraints.

Due to some technological constraints that may prevent an MHE from reaching a specific machine or delivering the material of a specific job, two sets, denoted $P^h$ and $D^h$, are defined respectively to represent the sets of pickup and delivery nodes that can be visited by MHE $h$. Furthermore, we define $N^h = P^h \cup D^h$, $P = \cup_{h \in H} P^h$ and $D = \cup_{h \in H} D^h$. We denote by $A^h$ the set of all feasible arcs (illustrated by discontinuous lines in Fig. 1, where each line pattern corresponds to one MHE) on the set of nodes $N^h \cup \{o_h, f_h\}$, where $o_h$ and $f_h$ are two nodes that correspond to the original location and final destination of MHE $h$, respectively. The definition of set $A^h$ is governed by the processing sequence of operations within each job, as for instance, it is not feasible to have an arc connecting nodes $\mathrm{PN}_{i-1}$ and $\mathrm{PN}_i$, where $i$ and $i+1 \in \underline{I}_k$ for some job $k$.

By knowing the physical processing location of each operation, the travel time of an MHE between two nodes in the network can be determined. We denote by $\tau_{q,r}^h$ the travel time from node $q$ to node $r$ for MHE $h$, where $(q,r) \in A^h$. We note that the term $w_q^h$, which represents the amount of load that will be added to MHE $h$ after visiting node $q$, takes a positive value if $q \in P$ and a negative value if $q \in D$. Here, the value of $w_q^h$ is based on the consumption units $c_i^h$ of the operation $i$ for which node $q$ is defined.

The decision variables in the JSPMH are classified into two sets: processing centers scheduling variables and MHE routing and scheduling variables. Processing centers scheduling variables are basically two sets. The first one includes the binary variables $y_{ij}^m$ which define the processing sequence of operations on machines. The variable $y_{ij}^m$ takes the value of one if operation $i$ precedes operation $j$ on machine $m \in M$ and zero otherwise. The second set includes the variables $t_i$ which represents the start time of operation $i$ on its processing center $pc(i)$.

The MHE routing and scheduling variables are defined using indexes that correspond to nodes on the network model of the problem. We denote $x_{q,r}^h$ as a binary variable that takes the value of one if MHE $h$ moves from node $q$ to node $r$, where $(q,r) \in A^h$. The variable denoted $T_q^h$ refers to the earliest time at which MHE $h$ becomes available at node $q \in N^h$. Specifically, $T_q^h$ equals the time at which MHE $h$ reaches the pickup start ($ps$) or the delivery start ($ds$) nodes defined at node $q$. The variable denoted $W_q^h$ defines the load of MHE $h$ after the loading or unloading at node $q \in N^h$ has been completed. In addition, we use the decision variable $\mathrm{X}_{q,r}^h$ which equals $T_r^h$ if MHE $h$ moves from node $q$ to node $r$, where $(q,r) \in A^h$ and zero otherwise. The $\mathrm{X}_{q,r}^h$ decision variables are needed in the model to eliminate sources of nonlinearity when the relationships between processing locations scheduling variables $t_i$ and their corresponding $T_r^h$ are defined.

The following is a list of the constraints that define the relationships between the decision variables and the different parameters of the studied system. These relationships represent the characteristics and the structure of the studied problem which must be satisfied for any given values of the decision variables in order to obtain a feasible solution that can be implemented. We first note that all the above-mentioned decision variables are not allowed to take negative values. This non-negativity constraint is defined by default in most commercial MILP solvers, and they are necessary for the definition of the forthcoming sets of constraints.

$$t_i - t_{i-1} \geq p_{i-1} + \sum_{h \in H} l_{i-1}^h \sum_{(q,\mathrm{PN}_{i-1}) \in A^h} x_{q,\mathrm{PN}_{i-1}}^h + \sum_{h \in H} u_i^h \sum_{(q,\mathrm{DN}_i) \in A^h} x_{q,\mathrm{DN}_i}^h \quad \forall i, i-1 \in I_k \ \forall k \in J \tag{1}$$

$$\mathrm{X}_{q,r}^h - T_r^h \geq G\left(x_{q,r}^h - 1\right) \quad \forall (q,r) \in A^h \ \forall h \in H \tag{2}$$

$$\mathrm{X}_{q,r}^h - T_r^h \leq 0 \quad \forall (q,r) \in A^h \ \forall h \in H \tag{3}$$

$$X_{q,r}^h \leq G x_{q,r}^h \quad \forall (q,r) \in A^h \ \forall h \in H \tag{4}$$

$$t_{\alpha_k} - \sum_{h \in H} \sum_{\left(q, \mathrm{PN}_{\alpha_k}\right) \in A^h} \mathrm{X}^h_{q, \mathrm{PN}_{\alpha_k}} = 0 \quad \forall k \in J \tag{5}$$

$$t_i - \sum_{h \in H} \sum_{(q, \mathrm{DN}_i) \in A^h} \mathrm{X}^h_{q, \mathrm{DN}_i} = \sum_{h \in H} u^h_i \sum_{q \in N^h} x^h_{q, \mathrm{DN}_i} \quad i \in I_k \backslash \{\alpha_k\} \quad \forall k \in J \tag{6}$$

$$t_i + p_i \leq T^h_{\mathrm{PN}_i} \quad \forall i \in \cup_{k \in J} I_k \backslash \{\omega_k\} \quad \forall h \in H \text{ and } \mathrm{PN}_i \in N^h \tag{7}$$

$$G y^m_{ij} + t_i - t_j \geq p_j \quad \forall i, j \in \underline{I} \text{ where } j > i, pc(i) = pc(j) = m \text{ and } jb(i) \neq jb(j) \tag{8}$$

$$G\left(1 - y^m_{ij}\right) + t_j - t_i \geq p_i \quad \forall i, j \in \underline{I} \text{ where } j > i, pc(i) = pc(j) = m \text{ and } jb(i) \neq jb(j) \tag{9}$$

$$\sum_{(o_h, r) \in A^h} x^h_{o_h, r} = 1 \ \forall h \in H \tag{10}$$

$$\sum_{(q, f_h) \in A^h} x^h_{q, f_h} = 1 \ \forall h \in H \tag{11}$$

$$\sum_{(q, r) \in A^h} x^h_{q, r} - \sum_{(r, q) \in A^h} x^h_{r, q} = 0 \ \forall r \in N^h \forall h \in H \tag{12}$$

$$\sum_{h \in H} \sum_{(q, r) \in A^h} x^h_{q, r} = 1 \ \forall q \in P \tag{13}$$

$$\sum_{(\mathrm{PN}_i, r) \in A^h} x^h_{\mathrm{PN}_i, r} - \sum_{(q, \mathrm{DN}_{i+1}) \in A^h} x^h_{q, \mathrm{DN}_{i+1}} = 0 \ \forall i \in \cup_{k \in J} I_k \backslash \{\omega_k\} \ \forall h \in H \tag{14}$$

$$T^h_r - T^h_q \geq G\left(x^h_{q, r} - 1\right) + s^h_q + \tau^h_{q, r} \quad \forall (q, r) \in A^h \ \forall h \in H \tag{15}$$

$$T^h_{\mathrm{PN}_i} + l^h_i + \tau^h_{\mathrm{PN}_i, \mathrm{DN}_{i+1}} \leq T^h_{\mathrm{DN}_{i+1}} \forall i \in \cup_{k \in J} I_k \backslash \{\omega_k\} \text{where } (\mathrm{PN}_i, \mathrm{DN}_{i+1}) \in A^h \forall h \in H \tag{16}$$

$$W^h_q \leq C_h \quad \forall q \in N^h \quad \forall h \in H \tag{17}$$

$$W^h_{o_h} = 0 \quad \forall h \in H \tag{18}$$

$$W^h_r - W^h_q \geq w^h_r - G\left(1 - x^h_{q, r}\right) \quad \forall (q, r) \in A^h \quad \forall h \in H \tag{19}$$

$$W^h_r - W^h_q \leq w^h_r + G\left(1 - x^h_{q, r}\right) \quad \forall (q, r) \in A^h \quad \forall h \in H \tag{20}$$

The first set of constraints (1) defines the precedence relationships among operations belonging to the same job as illustrated in the network model in Fig. 1a. That is the difference between the start times of two consecutive operations, $i-1$ and $i$, which belong to the same job must be at least equal to the processing time of the operation $i-1$ plus the MHE loading and unloading times of operations $i-1$ and $i$, respectively.

In order to be able to represent the relationships between the operations start time decision variables, $t_i$, and the MHE scheduling variables, $T^h_r$, the product $x^h_{q,r}.T^h_r$ is needed as it defines the time at which an MHE will reach a given node given that this node is visited by that MHE. Since this product will turn the mathematical programming model to be nonlinear, which is known to be hard to solve as compared to linear programming models, it is imperative to provide a linearization of this product in the mathematical model. In order to do that, the decision variables $\mathrm{X}^h_{q,r}$ which are defined as $\mathrm{X}^h_{q,r} = x^h_{q,r}.T^h_r$ are introduced. This definition is linearized by using the set of inequalities (2) to (4), where $G$ is a sufficiently large number.

The next two sets of constraints, represented by Eqs. 5 and 6, define the relationships between the operations start time variables and the scheduling variables of the MHEs. Constraints (Eq. 7) are necessary to make sure that a pickup trip will not start before the processing of an operation has been finished. In order to represent the operations scheduling constraints on machines which state that any two operations cannot be processed on the same machine simultaneously and once an operation is scheduled it cannot be preempted; two sets of constraints, (8) and (9), are introduced. These sets of constraints are equivalent to the disjunctive constraints found in the JSP literature [1]. The binary decision variables $y^m_{ij}$ are used to provide a linearization of these logical constraints.

Equations 10–14 define the MHE routing constraints which are based on the mathematical model used for the vehicle routing problem with pickup and delivery [33].

Equations 10 and 11 represent, respectively, the conditions that the tour of MHE $h$ must start at its origin node $o_h$ and finally end at its destination node $f_h$. Equation 12 ensures the continuity of the flow of the MHEs. That is, if a node is visited by an MHE through an incoming arc, it has to leave that node through an outgoing arc. Constraint 13 ensures that each pickup node will be visited exactly by one MHE. Constraint 14 mandates that the material picked up should finally reach its delivery point. We note here that in order for these constraints to work properly and provide feasible solutions, the definition of the network elements (nodes and arcs) should be made accurately.

Constraints 15 and 16 represent the MHE scheduling constraints which define the restrictions on the times at which an MHE visits nodes on the network. Constraints 17–20 define the MHE loading restrictions as related to the routing variables. These constraints are necessary for the representation of multiple-load MHEs that can conduct more than one pickup operation before making any delivery as the capacity of an MHE permits.

There are several different objective functions that can be sought. In this paper, we consider the objective of minimizing the makespan. The minimization of the makespan objective is a representative of the target of maximizing the utilization of the machines and the MHEs. The makespan, denoted $C_{\max}$, is defined as the time at which all finished parts become available at the unload station. That is to say $C_{\max} = \max_{k \in J}(t_{\omega_k})$. To represent this objective, an additional set of constraints needs to be added along with the definition of the objective function as shown below.

$$\text{Minimize} \quad C_{\max} \tag{21}$$

$$C_{\max} \geq t_{\omega_k} \quad \forall k \in J \tag{22}$$

The above MILP model is intended to provide a clear definition of the studied problem in precise mathematical terms, and it may help in future algorithmic developments. In order to verify this model, small-sized test problems of three machines, two MHEs, and three jobs are designed and solved using a commercial MILP solver—CPLEX [35]. The optimal solution was obtained within 10 min framework when running on a personal computer with dual 1.5 GHz Intel Xeon processors and Windows 2000 operating system. Clearly, MILP solvers are capable of providing solutions for small size problems. Larger problems with eight jobs and four machines and two MHEs were tested on CPLEX with no feasible solution obtained in three days running time. This makes using heuristic approaches for practical cases inevitable. In the following section, we present a constructive heuristic.

## 4 Constructive heuristic

Due to the complex nature of the studied problem which stems from the complexity of its constituent sub-problems and their interaction with each other, finding an efficient solution in a short processing time becomes a priority for a real-life application. In this section, we present a constructive heuristic intended to serve for that purpose. The other purpose that this constructive heuristic is intended for is to provide an efficient method for generating initial feasible solutions that are required by metaheuristic search techniques such as genetic algorithm, simulated annealing and tabu search. It is known that the performance of such metaheuristic search techniques is highly affected by the quality of the initial solutions they use. Accordingly, an efficient constructive heuristic becomes a major constituent for the success of such metaheuristics.

The main scheme of the developed heuristic is based on the well known Giffler and Thompson's algorithm (GTA) [36]. GTA is a traditional JSP constructive algorithm developed for generating feasible active schedules which was further extended to generate non-delay schedules. One of the main advantages of this algorithm is that different dispatching rules can be incorporated into it to select an operation from an outstanding set to be scheduled. Examples of such dispatching rules are shortest processing time (SPT), longest processing time (LPT), minimum completion time (MCT), most work remaining (MWR), and least work remaining (LWR). Panwalkar and Iskander [37] provide a literature review on the different dispatching rules that can be used.

The idea behind the developed constructive heuristic is to insert the MHE selection and routing decisions into the traditional GTA non-delay search structure. This is done by considering the impact of selecting an MHE for conducting the pickup and delivery tasks of an operation on its start time which in turn affects the operation selection decisions within the traditional GTA. The selection of an MHE to handle the pickup and delivery tasks of an operation follows a greedy scheme that base its decision on the minimum time at which an operation could start for a given MHE selection. The following list illustrates the main steps of this algorithm.

Procedure JSPMH-GTA

1. Start with the set of next schedulable operations $\Psi = \{i : i = \alpha_k + 1 \ \forall k \in J\}$, and let $t_{\alpha_k} = 0 \ \forall k \in J$
2. For every operation $i \in \Psi$ do
   2.1. Let $\Upsilon_i = \{h : \mathrm{DN}_i \in D^h \text{ and } \mathrm{PN}_{i-1} \in P^h \ \forall h \in H\}$
   2.2. For each MHE $h \in \Upsilon_i$, let $\Theta_i^h = OPTPATH(h, i)$
   2.3. Select $h_i^* = \underset{h \in \Upsilon_i}{\operatorname{argmin}}\{mst(\Theta_i^h)\}$, tie-breaking is arbitrary

2.4. Let $t_i = mst\left(\Theta_i^{h_i^*}\right)$
3. Select $i^* = \underset{i \in \Psi}{\operatorname{argmin}} \{t_i\}$, tie-breaking is arbitrary. Let $m^* = pc(i^*)$
4. Let the conflict set $\Gamma = \{j : j \in I, pc(j) = m^* \text{ and } t_j \leq t_{i*} + p_{i*}\}$
5. Select an operation $j^*$ from $\Gamma$ using an appropriate dispatching rule
6. For operation $j^*$ and MHE $h_{j*}^*$, schedule the necessary transportation tasks and update the start times for all the operations that have scheduled deliveries by $h_{j*}^*$ on $\Theta_{j*}^{h_{j*}^*}$ and determine the corresponding variables $T_q^{h_j^*}$ and $W_q^{h_j^*}$ for every node $q$ in the determined path $\Theta_{j*}^{h_{j*}^*}$
7. Let $\Psi = \Psi \backslash \{j^*\}$
8. If $j^* < \omega_{jb(j^*)}$, let $\Psi = \Psi \cup \{j^* + 1\}$
9. If $\Psi \neq \emptyset$ go to step 2; otherwise STOP

As in the traditional non-delay GTA, from a given set of next schedulable operations, an operation with the minimum start time is selected as shown in step 3. Here, the start time of an operation depends on the delivery schedule of its unprocessed part along with the time at which its processing center becomes available. In step 2, the minimum possible start time of an operation is determined by considering every MHE that can handle the pickup and delivery tasks for it. The minimum start time among those MHEs is determined and assigned to the start time of the operation after considering the time at which its processing center becomes available. In steps 4 and 5, from the set of all schedulable operations that need to be processed on the same machine of the selected operation and having start times not larger than its completion time (called conflict set), an operation is selected using a prespecified dispatching rule. The selected operation is then scheduled in step 6 along with any other operations that may be affected by the changes that have been made along the delivery path of the selected operation. In steps 7 and 8, the successor operation to the scheduled one in the job sequence is added to the set of next schedulable operations. The search continues until all the operations including the last dummy operation of each job are scheduled.

In step 2, the algorithm calls the subroutine *OPTPATH* ($h$, $i$) to search for an efficient path for every MHE $h$ that can handle the material transportation tasks for operation $i$. This subroutine returns a path $\Theta_i^h$ in the network shown in Fig. 1. This path starts at node $o_h$ and ends at node $de_i$. For the sake of brevity, we do not provide a list of the steps of this subroutine; instead, we briefly describe its main idea. The subroutine *OPTPATH* ($h$, $i$) finds the minimum time possible at which operation $i$ can start whenever MHE $h$ is used to conduct its associated pickup and delivery tasks while considering its previously scheduled transportation tasks and the possibility of combining trips with other prescheduled operations, i.e., conducting consecutive pick-up (delivery) trips. Every time this subroutine is called, it checks if there is available capacity to conduct multiple pickups for the last scheduled operation(s) and operation $i$ such that there is saving in the transportation time. If there is saving, combined trips are decided; otherwise, separate pickup and delivery trips for operation $i$ are determined starting from the last location (node) the MHE has reached. The sub-problem of conducting multiple pickup and delivery trips of a set of operations is formulated as a traveling salesman problem with precedence and time window constraints. This sub-problem is solved using a simple nearest neighbor algorithm.

After determining the efficient path for each MHE in step 2.2, the MHE that provides the minimum start time for operation $i$ is selected in step 2.3 and denoted $h_i^*$. Here, the subroutine $mst\left(\Theta_i^h\right)$ returns the minimum start time of operation $i$ given the path $\Theta_i^h$ and considering the finish time of the last scheduled operation on $pc(i)$. The value of the minimum start time is then assigned to the decision variable $t_i$ as indicated earlier.

## 5 Computational results

The purpose of the experimentations conducted in this paper is to compare the results obtained by implementing different dispatching rules within the developed constructive heuristic and provide some guidance on selecting the most efficient ones. Another purpose of the experimentations is to have an estimation of the average computational time requirements by the developed heuristic as related to the size of the problem.

From our preliminary experiments, it has been recognized that there are two main factors that define the structure of the studied problem and seem to have impact on the performance of the developed heuristic. The first factor which has been reported earlier in the experimental design of Bilge and Ulusoy [20] is the ratio between the average transportation time and the average operations processing time. This ratio is denoted $\tau/p$ and evaluated as $\left(\sum_{h\in H}\sum_{(q,r)\in A^h}\tau_{q,r}^h/|H|\right)/\left(\sum_{i\in \underline{I}}p_i/|\underline{I}|\right)$, where $|.|$ denotes the size of a set. The higher the $\tau/p$ ratio is, the larger the effect the decisions of the material handling tasks has on the scheduling objectives. The second parameter is the ratio between the average capacity for all MHEs and the average load consumed by all operations. This ratio is denoted $C/c$ and evaluated as $\left(\sum_{h\in H}C_h\right)/\left(\sum_{h\in H}\sum_{i\in \underline{I}}c_i^h/|\underline{I}|\right)$. Similarly, the higher the value of this ratio is, the higher the effect of the decisions for combined pickup and delivery tasks on the scheduling objectives.

Since the studied problem is seen as an extension to the JSP, the designed experiments are built upon the standard benchmark problems of the JSP (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/jobshopinfo.html). A selected 40 JSP benchmark problems are used for defining the number of machines, number of jobs, processing sequence of each job, and processing time of each operation. The number of MHEs is selected to be a quarter of the number of machines rounded to the nearest higher integer, and all the MHEs reside at a parking station at the beginning of the schedule. The loading and unloading times of operations by each MHE are randomly generated using uniform distribution covering the range from 0.05 to 0.1 of the average processing time among all operations ($p$) rounded to the nearest higher integer.

There are two levels selected for the $C/c$ ratio: they are 2.0 and 6.0. All operations' consumption values of MHE load ($c_i^h$) are set at unit value. Accordingly, the selected values of the $C/c$ ratio directly define the mean value of the MHE capacities. The values of the MHE capacities are randomly generated using uniform distribution with two ranges defining its lower and upper limits. The values of the lower and upper limits of those ranges are selected to result in a mean value for the MHE capacities close to the selected level. The selected ranges are [1.0, 3.0] and [3.0, 9.0].

To randomly generate transportation times, two ranges for the $\tau/p$ ratio are used. These ranges are [0.05, 0.25] and [0.25, 0.75]. By randomly generating a value for the $\tau/p$ ratio using uniform distribution with the upper and lower limits provided by the given range, the mean transportation time ($\tau$) is determined by multiplying that value with the average processing time for all the operations. Then, a transportation time is randomly generated using uniform distribution with the limits $\tau \pm 0.1 \times p$ and $\tau \pm 0.25 \times p$ for each range of the $\tau/p$ ratio, respectively.

For each standard JSP benchmark problem and combination of levels for the $\tau/p$ and $C/c$ ratios, ten problems are randomly generated. The developed constructive heuristic is applied to each problem with five different dispatching rules, namely SPT, LPT, MCT, MWR, and LWR. The objective value under consideration, $C_{\max}$, is evaluated for each problem-dispatching rule combination. The percentage difference of the objective value obtained by SPT and the one obtained by a given dispatching rule divided by the SPT objective and multiplied by 100 is evaluated for each randomly generated problem. The averages of these percentage differences are evaluated for the ten randomly generated problems and their values are reported in Tables 1 and 2.

The computational results indicate that on average, the performance of the MWR dispatching rule is better than the others followed by MCT. There is also indication that the gap between the dispatching rules and the SPT rule increases with the increase of the level of $\tau/p$ ratio, while it decreases with the increase of the level of the $C/c$ ratio. On the other hand, statistical analyses do not show any evidence for the effect of the number of jobs or the number of machines on that gap.

Statistical results for the computational time does not show any significant effect for the problem settings ($\tau/p$ and $C/c$ ratios) or the selection of the dispatching rule on the computational time. Meanwhile, the size of the problem represented by the number of operations does affect the computational time. To show how the computational time of the developed constructive algorithm evolves with the increase of the problem size, the average computational time calculated for all replicates, dispatching rules, and problem settings generated for all JSP benchmark problems having the same size is evaluated. The calculated values are provided in Table 3. Figure 2 provides a graphical representation for the average computational time versus the number of operations. From Fig. 2, it is evident that the average computational time can be represented as a polynomial function of the number of operations. This is supported by statistical regression analysis which shows that a polynomial function of the order three can properly express the relationship between the number of operations and the computational time. Such an average polynomial time complexity makes the developed constructive algorithm suitable for practical applications as opposed to any mathematical programming solver which generally has an exponential time complexity.

## 6 Conclusion and future research

This paper addresses the integrated scheduling of machines and material handling equipment in a job shop environment (JSPMH). A generic definition for the problem that takes into consideration the case of multiple-load material handling equipment is presented along with a mathematical mixed integer linear programming model. The studied problem is composed of two interrelated decision problems known in the literature, the job shop scheduling problem [1] and the vehicle routing problem with pickup and delivery [33]. Both problems are known to be NP-hard, meaning that there is no solution algorithm that can find an optimal solution in a computational time that can be expressed as a polynomial function of the problem size. The result is an even stronger NP-hard JSPMH problem. Due to this complex nature, it is necessary to develop efficient solution algorithms that can provide relatively good solutions in a reasonable computational time.

A constructive heuristic based on the well-known JSP GTA is presented. The idea of this heuristic is to incorporate the material handling routing and scheduling decisions into the mechanism of operations' selection and

**Table 1** Average relative performance for the set of problems having $0.05 \le \tau/p \le 0.25$

| JSP bench | $\|J\|$ | $\|M\|$ | $C/c = 2$ | | | | $C/c = 6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | LPT | MCT | MWR | LWR | LPT | MCT | MWR | LWR |
| la01 | 10 | 5 | −1.25 | 1.89 | 9.47 | −16.30 | 0.98 | 5.81 | 11.02 | −7.54 |
| la02 | 10 | 5 | −8.80 | 5.13 | −3.14 | −11.27 | −4.48 | 3.85 | −1.64 | −7.27 |
| la03 | 10 | 5 | −2.51 | 6.47 | 11.29 | −10.31 | −0.19 | 5.56 | 12.59 | −10.23 |
| la04 | 10 | 5 | −2.61 | 3.72 | 6.19 | −7.75 | −11.47 | −1.85 | −4.98 | −11.43 |
| la05 | 10 | 5 | 9.40 | 11.56 | 16.73 | −4.70 | 6.91 | 10.54 | 11.36 | 2.74 |
| la06 | 15 | 5 | 7.26 | 5.91 | 12.69 | −9.25 | 0.04 | 4.09 | 10.99 | −4.55 |
| la07 | 15 | 5 | −2.65 | 7.86 | 5.66 | −5.52 | −7.65 | 6.37 | 1.89 | −5.93 |
| la08 | 15 | 5 | 6.50 | 7.23 | 14.11 | −8.13 | −3.56 | −0.48 | 0.89 | −11.61 |
| la09 | 15 | 5 | 8.40 | 6.89 | 11.38 | −11.58 | 4.84 | 2.62 | 5.51 | −5.76 |
| la10 | 15 | 5 | 7.79 | 7.99 | 11.52 | −7.77 | 5.05 | 4.49 | 6.69 | −6.54 |
| orb1 | 10 | 10 | 4.14 | 2.88 | 1.83 | −6.55 | 5.86 | 11.84 | 3.57 | −5.09 |
| orb2 | 10 | 10 | −4.65 | 6.63 | 7.03 | −8.75 | −10.26 | 6.61 | 1.69 | −15.79 |
| orb3 | 10 | 10 | −6.71 | 2.13 | 2.87 | −10.32 | −9.11 | 4.36 | 5.33 | −9.98 |
| orb4 | 10 | 10 | −0.78 | 6.34 | 15.88 | −13.77 | −2.91 | 7.07 | 14.22 | −14.31 |
| orb5 | 10 | 10 | −3.37 | 4.53 | 2.16 | −0.01 | −6.38 | 5.07 | 4.22 | 0.33 |
| la26 | 20 | 10 | 10.43 | 4.92 | 13.69 | −8.99 | 2.02 | 5.58 | 7.61 | −7.11 |
| la27 | 20 | 10 | 1.23 | 4.67 | 7.31 | −9.43 | −1.96 | 1.28 | 3.60 | −4.70 |
| la28 | 20 | 10 | 7.09 | 10.03 | 11.83 | −2.85 | 1.30 | 5.09 | 4.65 | −7.23 |
| la29 | 20 | 10 | 6.07 | 9.11 | 15.52 | −9.58 | 5.20 | 7.03 | 13.23 | −6.91 |
| la30 | 20 | 10 | 9.27 | 10.72 | 15.61 | −2.82 | 7.03 | 9.31 | 12.28 | −7.83 |
| ta06 | 15 | 15 | 0.49 | 5.86 | 9.47 | −9.02 | −0.20 | 6.10 | 9.10 | −7.66 |
| ta07 | 15 | 15 | −2.63 | 3.25 | 4.60 | −14.20 | 1.97 | 6.47 | 8.32 | −12.72 |
| ta08 | 15 | 15 | 0.29 | 6.51 | 3.08 | −14.01 | −0.23 | 7.43 | 4.60 | −12.97 |
| ta09 | 15 | 15 | 0.29 | 4.29 | 10.61 | −15.34 | −3.06 | 3.73 | 8.64 | −18.04 |
| ta10 | 15 | 15 | 3.73 | 9.39 | 7.98 | −9.30 | −1.91 | 6.34 | 2.92 | −12.56 |
| ta16 | 20 | 15 | 4.93 | 8.23 | 11.46 | −6.82 | 2.76 | 7.33 | 10.45 | −5.41 |
| ta17 | 20 | 15 | 4.25 | 7.29 | 13.27 | −2.91 | 4.11 | 5.43 | 12.04 | −5.88 |
| ta18 | 20 | 15 | 1.51 | 8.49 | 6.78 | −8.57 | −1.97 | 5.33 | 4.90 | −6.96 |
| ta19 | 20 | 15 | 5.11 | 3.10 | 8.98 | −8.69 | 1.19 | 1.40 | 8.30 | −6.09 |
| ta20 | 20 | 15 | 6.03 | 9.44 | 11.63 | −3.95 | 4.57 | 9.87 | 11.39 | −4.54 |
| ta21 | 20 | 20 | −1.18 | 1.61 | 5.94 | −11.33 | −2.17 | 2.08 | 5.75 | −12.42 |
| ta22 | 20 | 20 | 0.16 | 8.35 | 7.95 | −9.09 | 2.25 | 8.09 | 8.04 | −11.20 |
| ta23 | 20 | 20 | −5.93 | 4.15 | 4.61 | −12.36 | −4.07 | 4.25 | 6.66 | −10.93 |
| ta24 | 20 | 20 | 0.40 | 8.44 | 9.55 | −7.71 | −1.96 | 5.71 | 6.61 | −7.59 |
| ta25 | 20 | 20 | 11.02 | 12.55 | 12.90 | −5.62 | 6.57 | 9.31 | 9.88 | −5.13 |
| ta41 | 30 | 20 | 1.16 | 6.07 | 6.13 | −6.29 | −4.46 | 3.32 | −0.74 | −9.57 |
| ta42 | 30 | 20 | −1.16 | 7.18 | 7.45 | −9.43 | −2.25 | 5.64 | 6.74 | −10.77 |
| ta43 | 30 | 20 | 3.83 | 8.98 | 11.12 | −7.65 | 0.17 | 6.66 | 7.75 | −8.40 |
| ta44 | 30 | 20 | 0.05 | 3.55 | 5.10 | −8.11 | 2.52 | 7.38 | 7.04 | −3.15 |
| ta45 | 30 | 20 | 5.23 | 9.71 | 9.48 | −8.62 | 0.73 | 5.85 | 5.57 | −6.23 |

dispatching within GTA. The developed constructive heuristic makes use of the GTA's flexibility of incorporating different dispatching rules. Experimentation has been conducted to show the effect of different dispatching rules on the performance measured by the makespan. The computational results indicate that the MWR dispatching rule performs better than the others on average, followed by the MCT dispatching rule. Regarding the computational

**Table 2** Average relative performance for the set of problems having $0.05 \leq \tau/p \leq 0.25$

| JSP bench | $\|J\|$ | $\|M\|$ | $C/c = 2$ | | | | $C/c = 6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | LPT | MCT | MWR | LWR | LPT | MCT | MWR | LWR |
| la01 | 10 | 5 | 4.58 | 3.43 | 9.17 | −10.12 | 0.20 | 1.55 | 10.81 | −11.73 |
| la02 | 10 | 5 | 1.14 | 3.69 | 7.97 | −6.38 | −0.75 | 2.66 | 1.21 | −9.81 |
| la03 | 10 | 5 | −1.07 | 5.25 | 10.67 | −2.71 | 1.92 | 1.48 | 10.27 | −7.61 |
| la04 | 10 | 5 | 6.68 | 2.27 | 8.61 | 0.38 | −3.97 | 1.46 | −0.09 | −5.62 |
| la05 | 10 | 5 | 12.76 | 6.80 | 14.39 | −3.02 | 2.13 | 2.75 | 8.69 | −6.50 |
| la06 | 15 | 5 | 10.49 | 5.69 | 14.98 | −2.95 | 7.04 | 7.70 | 10.34 | 0.25 |
| la07 | 15 | 5 | 4.00 | 6.57 | 11.29 | −5.97 | 2.29 | 7.71 | 9.07 | −3.24 |
| la08 | 15 | 5 | 7.32 | 7.28 | 13.72 | −6.03 | 1.81 | 6.76 | 10.20 | −5.62 |
| la09 | 15 | 5 | 10.19 | 9.16 | 16.30 | −3.76 | 4.89 | 4.79 | 6.18 | −3.86 |
| la10 | 15 | 5 | 9.72 | 4.77 | 12.36 | −4.73 | 1.06 | 5.09 | 3.96 | −9.27 |
| orb1 | 10 | 10 | 3.14 | 0.58 | 3.19 | −7.01 | 3.44 | 7.68 | 4.57 | −4.81 |
| orb2 | 10 | 10 | 2.31 | 2.66 | 7.86 | −4.02 | −3.85 | 4.66 | 5.87 | −3.66 |
| orb3 | 10 | 10 | −5.09 | 4.90 | 2.96 | −9.26 | −0.64 | 3.86 | 5.31 | −7.13 |
| orb4 | 10 | 10 | 1.48 | 6.46 | 12.59 | −7.72 | −2.91 | 5.43 | 10.48 | −5.85 |
| orb5 | 10 | 10 | −1.05 | 5.22 | 4.08 | −1.73 | −2.05 | 1.12 | 1.58 | −1.35 |
| la26 | 20 | 10 | 17.95 | 10.84 | 19.97 | 3.43 | 6.22 | 6.84 | 10.71 | −2.90 |
| la27 | 20 | 10 | 5.35 | 5.62 | 9.49 | −4.00 | 5.92 | 5.54 | 7.77 | −2.35 |
| la28 | 20 | 10 | 11.31 | 10.26 | 17.08 | −2.30 | 3.43 | 4.58 | 6.36 | −3.12 |
| la29 | 20 | 10 | 5.98 | 5.50 | 10.67 | −6.76 | 5.41 | 4.90 | 9.01 | −2.88 |
| la30 | 20 | 10 | 9.25 | 8.16 | 15.22 | −2.16 | 5.50 | 4.29 | 9.46 | −3.49 |
| ta06 | 15 | 15 | 2.70 | 3.78 | 7.16 | −6.91 | 1.37 | 4.62 | 4.75 | −6.09 |
| ta07 | 15 | 15 | −0.71 | 2.72 | 6.98 | −10.91 | 2.30 | 6.46 | 8.89 | −7.02 |
| ta08 | 15 | 15 | 4.19 | 6.29 | 9.25 | −8.03 | 1.65 | 5.00 | 7.43 | −3.86 |
| ta09 | 15 | 15 | 7.46 | 3.61 | 10.58 | −6.13 | 0.83 | 4.89 | 7.29 | −7.31 |
| ta10 | 15 | 15 | 4.21 | 6.31 | 5.81 | −7.22 | 1.64 | 4.59 | 4.61 | −7.02 |
| ta16 | 20 | 15 | 7.33 | 5.48 | 11.01 | −7.92 | 4.25 | 3.03 | 5.11 | −3.94 |
| ta17 | 20 | 15 | 2.66 | 5.79 | 8.71 | −9.16 | 2.08 | 5.03 | 7.51 | −5.58 |
| ta18 | 20 | 15 | 2.77 | 3.93 | 8.43 | −7.37 | 0.85 | 4.51 | 5.88 | −5.92 |
| ta19 | 20 | 15 | 3.77 | 2.76 | 9.16 | −5.61 | 2.79 | 6.18 | 7.38 | −3.55 |
| ta20 | 20 | 15 | 6.26 | 5.85 | 9.20 | −6.37 | 4.26 | 5.89 | 6.78 | −0.57 |
| ta21 | 20 | 20 | 4.18 | 7.61 | 9.02 | −4.09 | −0.33 | 2.39 | 4.53 | −6.26 |
| ta22 | 20 | 20 | 3.92 | 7.50 | 9.29 | −4.95 | 1.06 | 5.48 | 7.92 | −5.62 |
| ta23 | 20 | 20 | 3.44 | 6.74 | 9.12 | −4.63 | −0.18 | 4.76 | 5.03 | −5.67 |
| ta24 | 20 | 20 | 3.57 | 6.14 | 8.35 | −5.61 | 0.99 | 5.20 | 6.92 | −4.97 |
| ta25 | 20 | 20 | 4.46 | 7.81 | 9.37 | −3.00 | 2.58 | 7.60 | 6.97 | −5.60 |
| ta41 | 30 | 20 | 5.28 | 7.47 | 9.16 | −6.48 | 2.87 | 3.73 | 4.49 | −3.55 |
| ta42 | 30 | 20 | 6.14 | 7.66 | 10.47 | −4.35 | 1.43 | 3.40 | 5.63 | −5.04 |
| ta43 | 30 | 20 | 9.10 | 6.34 | 12.70 | −1.75 | 5.84 | 4.41 | 6.39 | −3.16 |
| ta44 | 30 | 20 | 6.79 | 6.40 | 10.18 | −3.62 | 2.59 | 3.35 | 5.31 | −5.77 |
| ta45 | 30 | 20 | 7.46 | 6.83 | 12.19 | −2.51 | 4.87 | 5.78 | 7.35 | −3.30 |

time complexity of the developed algorithm, experimentations show that it follows a polynomial time function of the problem size which is represented by the number of operations.

An obvious direction for future research is to develop improvement techniques to improve the solutions obtained by the developed constructive heuristic. One can benefit from the presented network or mathematical model in

**Table 3** Average computational time

| $\lvert J\rvert$ | $\lvert M\rvert$ | No. of operations = $\lvert J\rvert \times \lvert M\rvert$ | Average computational time (seconds) |
|---|---|---|---|
| 10 | 5 | 50 | 0.0485 |
| 15 | 5 | 75 | 0.1717 |
| 10 | 10 | 100 | 0.2053 |
| 20 | 10 | 200 | 1.9404 |
| 15 | 15 | 225 | 1.9122 |
| 20 | 15 | 300 | 5.045 |
| 20 | 20 | 400 | 10.222 |
| 30 | 20 | 600 | 43.557 |

developing local search mechanisms similar to those used for the job shop scheduling problem. This can also be extended further by applying metaheuristic search techniques.

$A^h$ Set of all feasible arcs on the set of nodes $N^h \cup \{o_h, f_h\}$

$C_h$ Capacity of MHE $h$

$c_i^h$ Units consumed by operation $i$ from the capacity of MHE $h$ During the delivery of the operation's unprocessed material to pc($i$)

$D$ $\cup_{h \in H} D^h$

$D^h$ Set of delivery nodes that can be visited in a feasible path of MHE $h$

$\mathrm{DN}_i$ Delivery node of operation $i$

$f_h$ Final (terminal) node of MHE $h$

$H$ Set of material handling equipment (MHEs)

$I$ Set of all operations' indexes

$\underline{I}$ $= \cup_{k \in J} \underline{I}_k$

$I_k$ $= \{\alpha_k, \alpha_k + 1, \ldots, \omega_k\} \subseteq I$ Set of indexes of operations belonging to job $k \in J$

$\underline{I}_k$ $= \{i : \alpha_k < i < \omega_k\}$ Set of indexes of manufacturing operations belonging to job $k \in J$

$J$ Set of jobs ready for processing

$jb(i)$ Job to which operation $i$ belongs

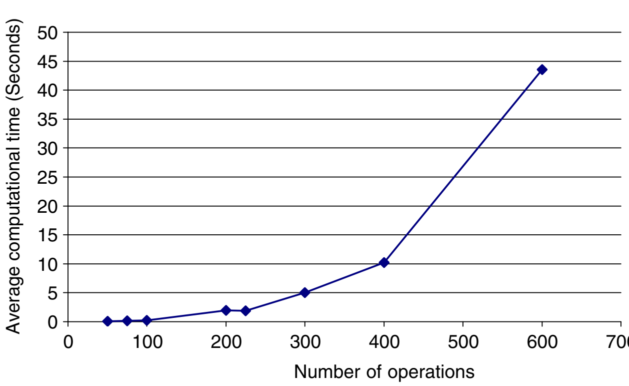



**Fig. 2** Average computational time versus the number of operations

$l_i^h$ Loading time of processed material of operation $i$ picked up by MHE $h$

$M$ Set of machines

$N^h$ $= P^h \cup D^h$

$o_h$ Origin node for MHE $h$

$P$ $= \cup_{h \in H} P^h$

PC Set of processing center $= M \cup$ {Load station, Unload station}

$pc(i)$ Processing center of operation $i$

$p^h$ Set of pickup nodes that can be visited in a feasible path of MHE $h$

$p_i$ Processing time of operation $i$

$\mathrm{PN}_i$ Pickup node of operation $i$

$s_q^h$ $= u_i^h$ when $q = \mathrm{DN}_i$ and $s_q^h = l_i^h$ when $q = \mathrm{PN}_i$

$T_q^h$ A decision variable that represents the time at which MHE $h$ becomes available at node $q \in N^h$

$t_i$ Decision variable that represents the start time of operation $i$

$u_i^h$ Unloading time of the material of operation $i$ prior to processing delivered by MHE $h$

$W_q^h$ A decision variable that equals the load of MHE $h$ after the loading or unloading at node $q \in N^h$ has been completed

$w_q^h$ Amount of load that will be added to MHE $h$ after visiting node $q$

$X_{q,r}^h$ A decision variable that equals $T_r^h$ if MHE $h$ moves from node $q$ to node $r$, where $(q, r) \in A^h$ and zero otherwise

$x_{q,r}^h$ A binary decision variable that takes the value of one if MHE $h$ moves from node $q$ to node $r$, where $(q, r) \in A^h$

$y_{ij}^m$ A binary decision variable that takes the value of one if operation $i$ precedes operation $j$ on machine $m$, and zero otherwise

$\alpha_k$ Index of the first operation of job $k$, which is a dummy operation used to represent the loading of raw material at the load station

$\tau_{q,r}^h$ Travel time from node $q$ to node $r$ for MHE $h$, where $(q, r) \in A^h$

$\omega_k$ Index of the last operation of job $k$, which is a dummy operation used to represent the unloading of finished parts at the unload station

## References


1. Jain AS, Meeran S (1999) Deterministic job-shop scheduling: Past, present and future. Eur J Oper Res 113:390–434. doi:10.1016/S0377-2217(98)00113-1
2. Smith JS, Peters BA, Srinivasan A (1999) Job shop scheduling considering material handling. Int J Prod Res 37:1541–1560. doi:10.1080/002075499191968
3. Fisher EL, Farber JB, Kay MG (1988) MATHES: An expert system for material handling equipment selection. Eng Costs Prod Econ 14(4):297–310. doi:10.1016/0167-188X(88)90034-1

4. Ganesharajah T, Hall NG, Sriskandarajah C (1998) Design and operational issues in AGV-served manufacturing systems. Ann Oper Res 76:109–154. doi:10.1023/A:1018936219150
5. Deb SK, Bhattacharyya B, Sorhkel SK (2003) Facility layout and material handling system selection planning using hybrid methodology. Int J Ind Eng Theory Appl Pract 10(3):289–297
6. Cho C, Egbelu PJ (2005) Design of a web-based integrated material handling system for manufacturing applications. Int J Prod Res 43(2):375–403. doi:10.1080/0020754042000268866
7. Maggu PL, Das G (1980) On the $2\times n$ sequencing problem with transportation time of jobs. Pure Appl Math Sci 12:1–6
8. Maggu PL, Das G, Kumar R (1981) On equivalent-job for job-block in $2\times n$ sequencing problem with transportation-times. J Oper Res Soc Jpn 24:136–146
9. Johnson SM (1954) Optimal two and three stage production schedules with setup times included. Nav Res Logistics Q 1:61–67. doi:10.1002/nav.3800010110
10. Raman N, Talbot FB, Rachamadugu RV (1986) Simultaneous scheduling of machines and material handling devices in automated manufacturing. Proceedings of 2nd ORSA/TIMS Conference on FMS: OR Models and Applications. Elsevier Science Publishers B.V, Amsterdam, pp 321–332
11. Hurink J, Knust S (2001) Makespan minimization for flow-shop problems with transportation times and a single robot. Discrete Appl Math 112:199–216. doi:10.1016/S0166-218X(00)00316-4
12. Che A, Chu C, Chu F (2002) Multicyclic hoist scheduling with constant processing times. IEEE Trans Robot Autom 18(1):69–80. doi:10.1109/70.988976
13. Liu J, Jiang Y, Zhou Z (2002) Cyclic scheduling of a single hoist in extended electroplating lines: a comprehensive integer programming solution. IIE Trans 34:905–914
14. Che A, Chu C (2005) A polynomial algorithm for no-wait cyclic hoist scheduling in an extended electroplating line. Oper Res Lett 33(3):274–284. doi:10.1016/j.orl.2003.10.012
15. Che A, Chu C (2005) Multi-degree cyclic scheduling of two robots in a no-wait flowshop. IEEE Trans Autom Sci Eng 2(2):173–183. doi:10.1109/TASE.2004.835600
16. Che A, Chu C (2007) Cyclic hoist scheduling in large real-life electroplating lines. OR Spectrum 29(3):445–470. doi:10.1007/s00291-006-0040-9
17. Che A, Chu C (2008) Optimal scheduling of material handling devices in a PCB production line: Problem formulation and a polynomial algorithm. Math Probl Eng 2008:364279
18. Wu SD, Wysk RA (1989) An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing. Int J Prod Res 27:1603–1623. doi:10.1080/00207548908942642
19. Sabuncuoglu I, Hommertzheim DL (1992) Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system. Int J Prod Res 30:1059–1079. doi:10.1080/00207549208942943
20. Bilge U, Ulusoy G (1995) Time window approach to simultaneous scheduling of machines and material handling system in an FMS. Oper Res 43:1058–1070. doi:10.1287/opre.43.6.1058
21. Ferland JA, Fortin L (1989) Vehicles scheduling with sliding time windows. Eur J Oper Res 38:213–226. doi:10.1016/0377-2217(89)90106-9
22. Ulusoy G, Sivrikaya-Serifoglu F, Bilge U (1997) A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Comput Oper Res 24:335–351. doi:10.1016/S0305-0548(96)00061-5
23. Abdelmaguid TF, Nassef AO, Kamal BA, Hassan MF (2004) A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. Int J Prod Res 42:267–281. doi:10.1080/0020754032000123579
24. Reddy BSP, Rao CSP (2006) A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. Int J Adv Manuf Technol 31:602–613. doi:10.1007/s00170-005-0223-6
25. Jawahar N, Aravindan P, Ponnambalam SG, Suresh RK (1998) AGV schedule integrated with production in flexible manufacturing systems. Int J Adv Manuf Technol 14:428–440. doi:10.1007/BF01304622
26. Anwar MF, Nagi R (1998) Integrated scheduling of material handling and manufacturing activities for just-in-time production of complex assemblies. Int J Prod Res 36:653–681. doi:10.1080/002075498193624
27. Khayat GE, Langevin A, Riopel D (2006) Integrated production and material handling scheduling using mathematical programming and constraint programming. Eur J Oper Res 175(3):1818–1832. doi:10.1016/j.ejor.2005.02.077
28. Hurink J, Knust S (2005) Tabu search algorithms for job-shop problems with a single transport robot. Eur J Oper Res 162:99–111. doi:10.1016/j.ejor.2003.10.034
29. Roy B, Sussmann B (1964) Les problemes d'ordonnancement avec constraints disjonctives. Technical report 9, SEMA, Note D.S., Paris
30. Caumond A, Lacomme P, Moukrim A, Tchernev N (2008) An MILP for scheduling problems in an FMS with one vehicle. Eur J Oper Res . doi:10.1016/j.ejor.2008.03.051
31. Ho Y-C, Chien S-H (2006) A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs. Int J Prod Res 44(20):4193–4222. doi:10.1080/00207540500442401
32. Lee J, Srisawat T (2006) Effect of manufacturing system constructs on pick-up and drop-off strategies of multiple-load AGVs. Int J Prod Res 44(4):653–673. doi:10.1080/00207540500251646
33. Desaulniers G, Desrosiers J, Erdmann A, Solomon MM, Soumis F (2002) VRP with pickup and delivery. In: Toth P, Vigo D (eds) The vehicle routing problem. Siam, Philadelphia, pp 225–242
34. Abdelmaguid TF (2007) Integrated scheduling of manufacturing operations and material handling tasks in the job shops: A network model. Proceedings of the 37th international conference on computers and industrial engineering. Alexandria, Egypt, pp 1951–1960
35. Anonymous (2003) CPLEX 9.0 User's Manual, ILOG, Mountain View, CA
36. Giffler B, Thompson GL (1960) Algorithms for solving production scheduling problems. Oper Res 8:487–503. doi:10.1287/opre.8.4.487
37. Panwalkar SS, Iskander W (1977) A survey of scheduling rules. Oper Res 25(1):45–61. doi:10.1287/opre.25.1.45