Les Cahiers du GERAD ISSN: 0711–2440

Integrated Production and Material Handling Scheduling Using a Set of Conflict Free Alternative Paths

G. El Khayat, A. Langevin
D. Riopel

G–2003–32

May 2003

Integrated Production and Material Handling Scheduling Using a Set of Conflict Free Alternative Paths

Ghada El Khayat, André Langevin
Diane Riopel

GERAD and École Polytechnque de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) H3C 3A7

ghada.el-khayat@polymtl.ca
andre.langevin@polymtl.ca
diane.riopel@polymtl.ca

May, 2003

Les Cahiers du GERAD

G–2003–32

Abstract

Due to the increasing popularity of automated guided vehicles in modern industry and the valuable investment they require, their design and operational issues have been studied extensively in recent literature. On the other hand, machine scheduling remains an active research area since several decades. However, material handling operations should be synchronized with machine operations to get a better utilization of resources. In this article, we optimally solve an integrated production and material handling scheduling problem in a job-shop context. A mathematical programming model and a constraint programming model are presented for the problem and tested on problems from the literature. One of k-shortest paths between every two machines may be used for a material handling operation or a deadhead. When a path is chosen, all its segments are mobilized until the end of the handling operation or the deadhead. A commercial software (ILOG OPLStudio) was used for modeling and solving both models.

Keywords: Integrated scheduling, conflict-free routing, material handling, production, mixed integer programming, constraint programming.

Résumé

À cause de l'usage, de plus en plus répandu, des chariots automatiques dans l'industrie moderne et les investissements coûteux qu'ils requièrent, leurs problèmes de design et d'opération ont été largement étudiés dans la récente littérature. D'autre part, l'ordonnancement de la production demeure un domaine de recherche très actif depuis plusieurs décennies. Cependant les tâches de manutention et celles de production doivent être synchronisées pour réaliser une meilleure utilisation des ressources. Dans cet article, nous résolvons à l'optimalité un problème d'ordonnancement intégré de la production et de la manutention dans un contexte d'atelier multigamme. Un modèle en programmation linéaire mixte et un autre en programmation par contraintes sont présentés et testes sur des jeux de données de la littérature. Un des k-plus court chemins entre chaque deux machines peut être utilisé pour une tâche de manutention ou un voyage à vide. Quand un chemin est choisi, tous ses segments sont mobilisés jusqu'à la fin de la tâche de manutention ou le voyage à vide. Un logiciel commercial (ILOG OPLStudio) a été utilisé pour la modélisation et la résolution des deux modèles.

Mots clés : Ordonnancement intégré, routage sans conflit, manutention, production, programmation linéaire mixte, programmation par contraintes

1 Introduction

Production scheduling problems in real factories incorporate much more constraints than those generally studied in literature. Constraints vary according to the production environment. In manufacturing, material handling constraints are very important and should be considered to achieve a better resource utilization. Neglecting the interdependence of the machines and the material handling equipment in a shop may lead to separate incompatible production and material handling schedules. Recently, some researchers studied the problem of simultaneous scheduling of production and material handling. Contributions in this area proposed heuristic solution approaches. Moreover, most test problems correspond to very special cases in manufacturing where travel times for material handling equipment are comparable to processing times on machines. In this paper, based on common shop dimensions and vehicle speed (3 km/hr), we consider material handling times that are shorter than the processing times. This corresponds to real manufacturing settings.

In the problem studied in this article, we consider the production system with the material handling resources. In addition to machines, material handling vehicles, such as automated guided vehicles (AGVs), are considered. In order to optimize AGVs operation, routing conflicts should be avoided. Production scheduling problems are very complex because of their combinatorial nature, even if we do not consider additional constraints. In this article, we propose a modeling strategy that accounts for routing conflicts by using a set of k-shortest paths. The shortest paths, on a bidirectional network, are used for either the material handling operations or the deadheads. When a handling operation or a deadhead is assigned to a path, all the segments of that path can not be used by another vehicle in the same time. A mathematical programming model and a constraint programming model are proposed. The two formulations are solved to optimality on test problems from the literature.

This paper is organized as follows. Section 2 presents a literature review. Section 3 presents the problem and the models. Section 4 presents the experimentation and the conclusion follows.

2 Literature Review

Contributions in integrated scheduling started as extensions of the 2 machine flow-shop problem solved optimally by Johnson [1]. Material handling considerations were added to the problem. These include : equipment availability and capacity constraints for buffers. To name some researchers, Maggu, Das and Kumar [2], Stern and Vitner [3], Panwalker [4] and Levner, Kogan and Levin [5] studied the problem for the minimization of the makespan. Raman, Talbot and Rachamadugu [6] studied a larger size flow-shop problem where handling times were not sequence dependent.

Lee and DiCesare [7] study the integrated production and material handling scheduling in a job-shop context. Two shortest path routes (in opposite directions) exist between every pair of machines. Routes are constraining resources. A Petri-net is presented and a heuristic method proposed. The objective is to minimize the makespan. They consider a shop of 3 machines and 1 robot for transformation activities and 5 AGVs for material handling activities. Two cases are presented. In the first, the AGV is dedicated to a job, accompanying it until the end of processing. In the second, the AGV is dedicated to a machine to move the jobs after being processed by the machine. 2 AGVs are dedicated to the load/unload station. Therefore, no assignment decisions are considered. The method was applied to another problem with no a priori assignments (3 machines, 1 AGVs and 3 jobs). Operations per job are less than or equal to 3.

Integrated scheduling in a job-shop context was studied by Bilge and Ulusoy [8]. They study the problem considering the objective of minimizing the makespan in an FMS. Unlike in Raman, Talbot and Rachamadugu (1986), AGVs do not return to a load/unload station after every material handling operation. Hence, handling time is sequence dependent. A non linear formulation is presented. The problem is then decomposed into 2 sub-problems and solved by a "Time-window" heuristic. At every iteration, they obtain a new machine schedule this is used to determine time-windows for material handling operations. They then search for a feasible solution for the material handling scheduling sub-problem. If this is not possible, the machine schedule is revised and the heuristic continues.

Ulusoy, Funda and Bilge [9] study the same problem and propose a genetic algorithm to solve it. Results are better than those obtained using the "Time-window" heuristic. The algorithm solved optimally 60% of the test problems. The main difference between the two approaches is that a solution obtained by the genetic algorithm contains simultaneously information on the machine scheduling and the material handling scheduling in the chromosomes while the iterative approach considers the 2 systems separately. The shortest path between 2 machines determines the handling time and deadheads time. There are no buffer considerations.

Sabuncuoglu and Karabuk [10] study the integrated scheduling problem considering a limited buffer capacity for machines. They use a partial enumeration method (Filtered Beam Search). Their job-shop consists of 6 machines and a load/unload station. 3 AGVs are responsible for material handling operations. However, the limited buffer capacity constraint is not rigid and overcome by a design solution that supposes that a central infinite capacity buffer exists and that AGVs can be rerouted to it at any time to prevent blocking. Conflict avoidance is not clearly discussed in the paper. Each route segment measures 5 distance units. Test problems have up to 25 jobs, 5 or 6 operations per job. Processing times are determined by a 2-Erlang distribution. The performance of the algorithm is superior compared to scheduling rules. Several objective functions were tested.

A 2-stage assembly production system composed of 4 machine cells is studied by Anwar and Nagi [11]. The main distinction between single and 2-stage production is in the nature of the precedence constraints. In assembly shops, precedence relations exist between operations on different pieces that will ultimately be a part of one end-product. The problem does not consider a route network and conflicts are not taken into account. The authors present a formulation and a heuristic to solve the problem. Anwar and Nagi [12], study the same assembly shop. They propose a heuristic methodology that accounts for conflicts. No numerical results are reported.

Smith, Peters and Srinivasan [13] consider not only material handling activities but also explicitly loading and unloading activities. In their problem, a machine stays blocked, if there is no material handling resource available to free it. Two heuristic approaches are presented. One is a global random search procedure O (n^2 log n) that considers all operations to be scheduled: n being the number of operations to schedule. The second approach is hierarchical and considers machines in the first place.

El Khayat, Langevin and Riopel [14], propose a mathematical programming and a constraint programming formulation for the integrated production and material handling scheduling problem. They consider the use of shortest paths to accomplish a handling operation or a deadhead. Test problems presented in Bilge and Ulusoy [8] were used to validate the models. They also proposed larger size problem instances. The two models were solved to optimality. However, for large instances, the constraint programming model outperformed the mathematical programming model.

Previous contributions consider inter-machine distances to be the most influencing for the material handling activities. Lee and Maneesavet [15], however, present a contribution where material handling activities that influence the schedule take place between the load/unload station and the machine. Dispatching strategies are proposed and evaluated for rail-guided vehicles in a loading/unloading zone of a Flexible Manufacturing System (FMS).

Some contributions were also presented in a dynamic setting. They consider continuous arrival of jobs instead of a set of jobs available at the beginning of the scheduling horizon. Myopic scheduling rules are generally used. Sabancuoglo and Hommertzhein [16] studied scheduling rules for machines and AGVs in an FMS. Later, they studied integrated scheduling of production and material handling for an FMS with a job-shop production environment [17]. Simulation was used to evaluate the performance of different scheduling rules. The objective considered was to minimize average flow-time. They consider finite capacity for machines, material handling equipment and work-in-progress buffers. The number of machines is between 1 and 6, and the system has 2 AGVs. Sabuncuoglu and Hommertzheim [18] also proposed an on-line algorithm with a better vision compared to the scheduling rules. The algorithm considers more than one operation at a time. The same authors reconsider the experimental investigation of scheduling rules for a wide variety of objective functions [19]. They also carried out this investigation for the case of machine breakdown [20]. Jawahar et al. [21] also study the dynamic problem. They present a heuristic that uses dispatching rules, accounting for conflicts, for AGVs.

As presented, although different complementary aspects are integrated to the machine scheduling problem, solution methodologies in the literature are all heuristic. We optimally solve the integrated scheduling problem, in a job-shop setting. We also use mathematical programming and constraint programming for the first time to solve the problem. To our knowledge, this problem has not been formulated nor solved using the constraint programming technique. However, production scheduling problems implying only machine resources was studied by Jain and Grossmann [22] and solved using the constraint programming technique and results were promising. On the other hand, vehicle routing problems were solved by Pesant et al. [23] using constraint programming. However, in their work, conflicts were not accounted for.

3 Problem Statement And Models

Mathematical programming formulations proposed in the literature, for job-shop scheduling problems, are quite difficult to solve for large size problems. They are mixed integer and sometimes non-linear for disjunctive constraints. A rather new research area that has proven effective for scheduling applications is constraint programming. In section 3.1, we present a mixed-integer model for the integrated scheduling job shop. In section 3.2, we present a constraint programming model for the problem. First, we describe the operational system.

The operational system considered is a job-shop environment, with machines and AGVs referred to as workcentres. A job is composed of a certain number of pieces to be processed on and handled between machines in a predefined sequence. These pieces form a lot or several lots. A certain number of pieces forming a lot are gathered on a pallet and then handled by AGVs. We consider one lot to schedule for each job. Operation refers to processing on a machine or handling between two machines.

The number and types of machines are given. That is the lot assignment to machines is already determined together with the order in which machines will be visited. Sufficient input/output buffer space is available at each machine. Tools, pallets and resources for loading and unloading are sufficiently available. Machine operations are not preemptive and the set of operations to schedule together with relevant data is available at the beginning of the time horizon. That is all jobs have zero ready times.

Trips may follow one of the k-shortest paths between 2 machines to accomplish whether a material handling operation or an empty travel (deadhead). Material handling operations and deadheads are not preemptive. The duration of a material handling operation depends on the route to which it is assigned. The duration of the deadhead depends on the assignment sequence of the material handling operations to AGVs and consequently the assignment to routes. All data are deterministic.

The problem reads as follows: given the shop layout, k-shortest path routes between every origin-destination pair and job routes indicating precedence relations and processing times, determine the starting time of production and material handling operations for all jobs together with the assignment of material handling operations to AGVs and to routes that minimize the makespan.

3.1 Mathematical programming formulation

The model includes variables and constraints representing the job-shop production scheduling problem. To this, we add precedence constraints for material handling operations, assignment constraints of material handling operations to AGVs and to predetermined routes and connectivity constraints ensuring that empty travel times are considered. Empty travel times depend on assignment of the deadhead to predetermined route. A material handling operation corresponds to moving a job from a source machine to a destination machine on which the following processing will take place. An empty travel corresponds to the movement of the AGV from the machine destination of a material handling operation to the machine source of the following material handling operation on the same AGV. Disjunctive constraints are written for routes as these are considered constraining resources. Conflicts are obviously accounted for. However, a route used by a material handling operation or a deadhead is completely mobilized until the end of the operation.

3.1.1 Notation.

O	Set of operations (j∈O), including potential deadheads and an end dummy operation
O^R	Set of production and material handling operations $O^R \subset O$
O^M	Set of material handling operations $O^M \subset O$
O^P	Set of production operations $O^P \subset O$
O^F	Set of first operations of all jobs $O^F \subset O$
O^L	Set of last operations of all jobs $O^L \subset O$
O^V	Set of deadheads $O^V \subset O$
O^{MV}	Set of material handling operations and deadheads $O^{MV} \subset O$
W	Set of all workstations $w \in W$
M	Set of machines $M \subset W$
C	Set of AGVs $C \subset W$
R	Set of routes $r \in R$
R^c	Set of pairs of conflicting routes
O^w	Set of operations soliciting workstation w
O^{si}	Set of deadheads that might be generated upon accomplishing material handling task i
O^{pj}	Set of deadheads that might precede material handling task j
R^j	Set of routes that might host an operation j
O^r	Set of operations soliciting a route r

d_r	Time needed to travel a route r
n(j)	Operation following operation j
s	Start of the time horizon
t_j	Processing time of production or dummy operation j
H	A big value

Ψ_{ij}	Variable having the value 1 if operation i precedes operation j , 0 otherwise
Φ_{jw}	Variable having the value 1 if material handling operation $j$is assigned to AGV w, 0 otherwise
Π_{jr}	Variable having the value 1 if material handling operation or deadhead j is assigned to the route r
Ω_{ij}	Variable having the value 1 if material handling operation or deadhead i precedes material handling operation or dead j on a route, 0 otherwise
C_{max}	Variable indicating the end time of the schedule (makespan)
S_j	Variable indicating the start time of the operation or deadhead j

3.1.2 Mathematical programming model.

Min C_{max}		(1)
$C_{max} \geq S_j + t_j$	$\forall j \in O^L$	(2)
$S_{n(j)} \geq S_j + t_j$	$\forall j \in O \backslash O^L \mid j \in O^P$	(3)
$S_{n(j)} \geq S_j + \Sigma_r d_r \Pi_{jr}$	$\forall j \in O \backslash O^L \mid j \in O^M$	(4)
$S_j \geq s$	$\forall j \in O^F$	(5)
$\Psi_{ij} + \Psi_{ji} = 1$	$\forall w, \forall i,j \mid i \neq j, i,j \in O^w$	(6)
$\Omega_{ij} + \Omega_{ji} = 1$	$\forall r, \forall i,j \mid r \in R^j, r \in R^i, i \neq j, i,j \in O^{MV}$	(7)
$\sum_w \Phi_{jw} = 1$	$\forall w \in C, \forall j \mid j \in O^w$	(8)
$\Sigma_r \Pi_{jr} = 1$	$\forall j \in O^{MV} \mid r \in R^j$	(9)
$S_j \geq (S_i + t_i) + (\Psi_{ij} - 1) H$	$\forall w \in M, \forall i,j \mid i \neq j, i,j \in O^w$	(10)
$S_j \geq S_i + \Sigma_r d_r \Pi_{jr} + \Sigma_r d_r \Pi_{vr} + (\Pi_{jr} + \Pi_{vr} + (\sum_{k \neq i} \Psi_{ik} - \sum_{k \neq j} \Psi_{jk}) + \Phi_{iw} + \Phi_{jw} - 5) H$	$\forall w \in C, \forall i,j \mid i \neq j, i,j \in O^w, v \in O^{si} \cap O^{pj}$	(11)
$S_j \geq S_i + d_s + (\Pi_{jr} + \Pi_{is} + \Omega_{ij} - 3) H$	$\forall (r,s) \in R^C, \forall i,j \mid i \neq j, i,j \in O^{MV}, s \in R^i, r \in R^j$	(12)
$S_v \geq S_i + d_s + (\sum_{k \neq i} \Psi_{ik} - \sum_{k \neq j} \Psi_{jk}) + \Phi_{iw} + \Phi_{jw} + \Pi_{is} - 4) H$	$\forall w \in C, v \in O^{si} \cap O^{pj}, \forall s \in R^i, \forall i,j \mid i \neq j, i,j \in O^M$	(13)
$S_j \geq S_v + d_s + ((\sum_{k \neq i} \Psi_{ik} - \sum_{k \neq j} \Psi_{jk}) + \Phi_{iw} + \Phi_j + \Pi_{vs} - 4) H$	$\forall w \in C, \forall i,j, v \in O^{si} \cap O^{pj}, \forall s \in R^v \mid i \neq j, i,j \in O^M$	(14)
$\Psi_{ij} \in \{0,1\}$		(15)
$\Phi_{jw} \in \{0,1\}$		(16)
$\Pi_{jr} \in \{0,1\}$		(17)
$\Omega_{ij} \in \{0,1\}$		(18)
$S_j \geq 0$		(19)

Table 1: Complete model in mathematical programming

The model minimizes the makespan (1). Constraints (2) ensure that the makespan is greater than the end times of last operations. According to constraints (3), precedence relations between production operations and their successors are respected. Constraints (4) state that precedence relations for material handling operations are respected and that their duration depends on the assignment to one of the shortest path routes. Constraints (5) impose that first operations start after the beginning of the scheduling horizon. Constraints (6) determine the order in which operations are assigned to a machine or an AGV. Similarly, constraints (7) determine whether a material handing operation or a deadhead i precedes j on a route or vice versa. Constraints (8) state that a material handling operation is assigned to an AGV. Constraints (9) state that a material handling operation or a deadhead is assigned to a route. Constraints (10) model the disjunctive character of the machines. For two successive operations on a machine, the second starts after the first is finished.

Constraints (11) model the disjunctive character of AGVs. For two successive operations on an AGV, the second starts after or at the end of the deadhead resulting from the operations assignment. The duration of the deadhead and the material handling operation depends on the used route. Constraints (11) also account for connectivity of the AGVs routes since they consider empty travel time for two successive material handling operations on an AGV. Constraints (12) ensure that no conflicts take place on routes, for any two routes sharing one or several segments. According to constraints (13) and (14), deadheads occur between successive material handling operations assigned to the same AGV. Constraints (15-18,19) define binary and continuous variables respectively.

3.2 Constraint programming formulation

It is difficult to present standard notations and formulations for constraint programming models. Hence, we present the declarative names as in the model formulated with using the commercial software (OPLStudio). The choice of this software was based on the fact that incorporates both Cplex and constraint programming algorithms. Therefore, it was used with both models presented in this article. The constraint programming model uses software functions that enable solving by the solver (Scheduler), part of OPLStudio. This solver is specially designed for scheduling problems. The number of variables and constraints is significantly less than those of the mixed-integer formulation. The model is written in a more expressive manner and data structures are more compact.

First, we present main resource types in OPLStudio [24]. For an introduction to constraint programming the reader is referred to Marriott and Stucky [25]. For a presentation of logic-based methods for optimization, we refer the reader to Hooker [26].

Unary resources: a unary resource cannot be shared by two activities/operations at the same time.

Alternatives resources: alternative resources are equivalent from the activity/operation stand point. We can use one or the other. The instruction **ActivityHasSelectedResource** used with appropriate arguments holds if a resource is chosen by an activity. It can be used to formulate global constraints.

Discrete Resources: discrete resources are used to model equivalent and interchangeable resources.

In the model, we use a global constraint (alldifferent) to model assignment order to AGVs. For a detailed presentation on types of global constraints (deterministic and non-deterministic) and more generally on types of constraints in constraint programming (elementary and composite), we refer the reader to Van Hentenryk [27].

The software supports the definition of variable operation duration. The model developed is based on this aspect since material handling operations and deadheads have the duration of the route chosen which is not determined a priori

3.2.1 Notation. In this section we present the notation for the constraint programming model. Next, we present the model using usual logical constraints. In conjunction with the resource declaration, the OPLStudio keyword **requires** is used to model the resource disjunction constraints.

Machines	Set of machines
Tasks	Set of operations
Tasksmach	Set of production operations ⊂ Tasks
TasksMH	Set of material handling operations ⊂ Tasks
Vehicles	Set of vehicles
MHdum	Set of dummy ending operations for AGVs (End1..Endx)
MHorigin	Set of dummy starting operations for AGVs (Start1..Startx)
TasksMHdum	Set of material handling operations and dummy ending operations for AGVs TasksMH ∪ MHdum
TasksMHSour	Set of material handling operations and start dummy operations for AGVs TasksMH ∪ MHorigin
TasksMHSourDUM	Set of material handling operations and dummy starting and ending operations for AGVs TasksMH ∪ MHdum ∪ MHorigin
SetOfPrecedences	Set of pairs of operations having a precedence relation
deadheads	Set of potential deadheads
Routes	Set of routes
mayHost[t]	Set of routes that might host an operation t∈Tasks ∪ deadheads
ConflictingRoutes	Set of pairs of conflicting routes (r1,r2) sharing one or several segments r1 ∈ Routes, r2 ∈ Routes
durationR [r]	Time needed to travel route r ∈ Routes
resourcem [t]	"Machines" type data indicating which machine is needed to accomplish operation t ∈ Tasksmach
followsOnVehicle[t]	"Tasks" type variable indicating which operation follows another operation on an AGV t ∈ TasksMHSour
carries [t]	"Vehicles" type variable indicating to which vehicle a material handling or a dummy operation would be assigned t ∈TasksMHSourDUM
hosts [t]	"Routes" type variable indicating to which route an operation or a deadhead would be assigned t ∈ Tasks ∪ deadheads
duration [t]	Integer variable indicating the duration of an operation or a deadhead t ∈ Tasks ∪ deadheads
S[t]	Integer variable indicating the start time of a production, a material handling operation, a deadhead or the dummy ending operation "makespan" t ∈ Tasks ∪ deadheads ∪ makespan

3.2.2 Constraint programming model. The constraint programming formulation has the objective of minimizing the makespan. In the model, we define a set of production operations to accomplish as well as a set of material handling operations. Deadheads arise as material handling operations are assigned to AGVs. They are accounted for by the constraints that formulate the AGVs disjunction. However, deadheads use routes which are constraining resources. Consequently, when they occur they mobilize routes for which disjunction constraints should be respected.

Minimize S [makespan]	(20)
S[makespan] $\geq$ S[t] + duration[t] $\forall$ t $\in$ Tasks $\cup$ deadheads	(21)
S[q] $\geq$ S[t] + duration [t] $\forall$(t,q) $\in$ setOfPrecedences	(22)
followsOnVehicle [p]= q $\Rightarrow$ S[p] + duration [p] $\leq$ S[d] $\forall$ d $\in$ deadheads, $\forall$(p,q)$\vert\{$p $\in$ TasksMH, q $\in$ TasksMHdum, d after p & before q$\}$	(23)
followsOnVehicle [p]= q $\Rightarrow$ S[p] + duration [p] $\leq$ S[q] $\forall$ d $\in$ deadheads, $\forall$(p,q)$\vert\{$p $\in$ TasksMH, q $\in$ TasksMHdum, d after p & before q$\}$	(24)
followsOnVehicle [p]= q $\Rightarrow$ S[d] + duration[d] $\leq$ S [q] $\forall$ d $\in$ deadheads, $\forall$(p,q)$\vert\{$p $\in$ TasksMH, q $\in$ TasksMHdum, d after p & before q$\}$	(25)
alldifferent (followsOnVehicle)	(26)
followsOnVehicle [t] = q $\Rightarrow$ t $\neq$ q $\forall$(t,q)$\vert$p $\in$TasksMHSour & q $\in$ TasksMHdum	(27)
followsOnVehicle [p]= q $\Rightarrow$ q in TasksMHdum $\forall$ p $\in$ TasksMHSour, q $\in$ Tasks	(28)
carries [Start1] = carries[End1]	(29)
carries [Start2] $\neq$ carries[Start1]	(30)
carries[Start2] = carries[End2]	(31)
carries [End1] $\neq$ carries[End2]	(32)
DurationR [hosts [t]]=duration [t] $\forall$ t $\in$ Tasksmach $\cup$ TasksMH	(33)
followsOnVehicle[p]=q $\Rightarrow$ duration[d]=DurationR [hosts[v]] $\forall$ r $\in$ routes, $\forall$ d $\in$ deadheads, $\forall$(p,q)$\vert\{$p $\in$ TasksMH, q $\in$ TasksMHdum, d after p & before q$\}$	(34)
hosts[t]= r $\Rightarrow$ r in mayHost[t] $\forall$t $\in$ TasksMH $\cup$ deadheads, $\forall$ r $\in$ Routes	(35)
followsOnVehicle[p]=q $\Rightarrow$ carries [p] = carries [q] $\forall$ p $\in$ TasksMHSour, q $\in$ Tasks	(36)
S[q] $\geq$ S[t] + d[t] \/ S[t] $\geq$ S[q] +d[q] $\forall$ t $\in$ Tasksmach, $\forall$ m $\vert$ resourcem[t]= resourcem[q]= m	(37)
carriers[t]= c $\Rightarrow$ t **requires** c $\forall$ t $\in$ TasksMHSourDUM, $\forall$ c $\in$ Vehicles	(38)
followsOnVehicle[p]=q & carries [p]=c $\Rightarrow$ a[d] **requires** c $\forall$ p $\in$ TasksMHSour, $\forall$ q $\in$ TasksMHdum, d after p & before q,$\forall$ d $\in$ deadheads, $\forall$ c $\in$ Chariots	(39)
followsOnVehicle[p] = q & hosts [d]= r $\Rightarrow$ a[d] **requires** r $\forall$ r $\in$ routes, $\forall$ d$\in$deadheads, $\forall$(p,q)$\vert\{$p $\in$ TasksMH, q $\in$ TasksMH, d after p & before q$\}$	(40)
hosts[t]= r $\Rightarrow$ t **requires** r $\forall$ t $\in$ TasksMH, $\forall$ r $\in$ mayHost [t]	(41)
hosts [d] =r2 & hosts[t]= r1 $\Rightarrow$ a[t].start $\geq$ a[d].end \/ a[d].start $\geq$ a[t].end $\forall$ (r1,r2) $\in$ ConflictingRoutes, $\forall$ t $\in$ TasksMH $\cup$ deadheads $\forall$ d $\in$ TasksMH $\cup$ deadheads $\vert$ $\{$r1 $\in$ mayHost[t]& r2 $\in$ mayHost[d]$\}$	(42)
Sub-tour constraints as needed	(43)

Table 2: Complete model in constraint programming

In (20) we minimize the makespan. Precedence relations between makespan and all other operations or deadheads are modeled by constraints (21). Precedence relations between operations are accounted for by constraints (22). More precedence relations are generated according to assignment order to the AGVs and are modeled by constraints (23), (24), and (25). They state that the deadhead resulting from the assignment of two successive material handling operations to an AGV start at the end of the first material handling task or later. Constraints (24) are redundant and are verified by constraints (25) which define precedence relations between the resulting deadhead and the following material handling task on the AGV. Global constraint (26) imposes that each material handling task has a distinct time interval according to the assignment order to AGVs. Constraint (27) ensures that material handling operation cannot be a successor to itself. Constraints (28) indicate that a material handling operation on an AGV is followed by another material handling operation or a dummy ending operation. Constraints (29-32) define the start and the end of an AGV route and assign a dummy start and a dummy end operation to each AGV. Constraints (33) determine the duration of an operation which corresponds to the used route. Artificial routes are created for production operations. Constraints (34) determine the duration of deadheads according to the routes. Constraints (35) state that routes used for handling operations or deadheads are admissible. Constraints (36) ensure that two following operations are assigned to the same vehicle. Constraints (37) model the disjunctive character of machines. In the OPLStudio model, these are accounted for by the **unary** resource declaration for the machines. The declaration is used in conjunction with a **requires** constraint. The latter ensures that the resource requirements are met. Similarly, constraints (38-39) and (40-41) account for the disjunction of AGVs and routes, respectively. In the OPLStudio model, AGVs and routes are declared **unary** resources, as defined earlier. This declaration is also used in conjunction with a **requires** constraint to respect resource needs. Variables "carries" and "hosts" eliminates the need to define these entities as **alternative** resources. Constraints (42) account for conflicts. They ensure that two routes sharing one or several segments are not used at the same time.

By subtour constraints we refer to constraints that ensure that a number of routes equivalent to the number of AGVs is created. They prevent the occurrence of a solution where the following operations follow on an AGV: a1,a2, ..., ax,a1. To formulate this we use a logical constraint that reads as follows:

followsOnVehicle[a1]=a2 &. . . . . & followsOnVehicle[ay]= ax ⇒ not followsOnVehicle[ax]=a1

Subtour constraints for three operations are generally sufficient because lots of them are already prohibited by precedence constraints. Subtour constraints do not affect solving time negatively.

In the model, resources are also declared **discrete** to facilitate search for solutions. Once a solution is found for the **discrete** resource problem, it is then easier to find a solution that respects the **unary** resource constraint.

Complementary to the problem formulation, a good search procedure should be designed. Several procedures are available in OPLStudio. In our problem, we use the **generate** instruction which is one of the generation procedures in OPL. The instruction acts randomly. Generation procedures receive a discrete variable, or an arbitrary array of discrete variables, and generate values for all these variables. The **generate** instruction generates values for the variables in its arguments by first generating values for the variable with the smallest domain. In addition to the generate instruction, we use a dichotomic search combined with Limited Discrepancy Search (LDS). LDS is a search strategy that assumes the existence of a good heuristic that it incorporates. Its basic intuition is that the heuristic, when it fails, probably would have found a solution if it had made a small number of different decisions during the search. The choices where the search procedure does not follow the heuristic are called *discrepancies*. As a consequence, LDS systematically explores the search tree by increasing the number of allowed discrepancies. Initially, a small number of discrepancies is allowed. If the search is not successful or if an optimal solution is desired, the number of discrepancies is increased and the process is iterated until a solution is found or the whole search space has been explored. LDS has been shown to be effective for job-shop scheduling problems.

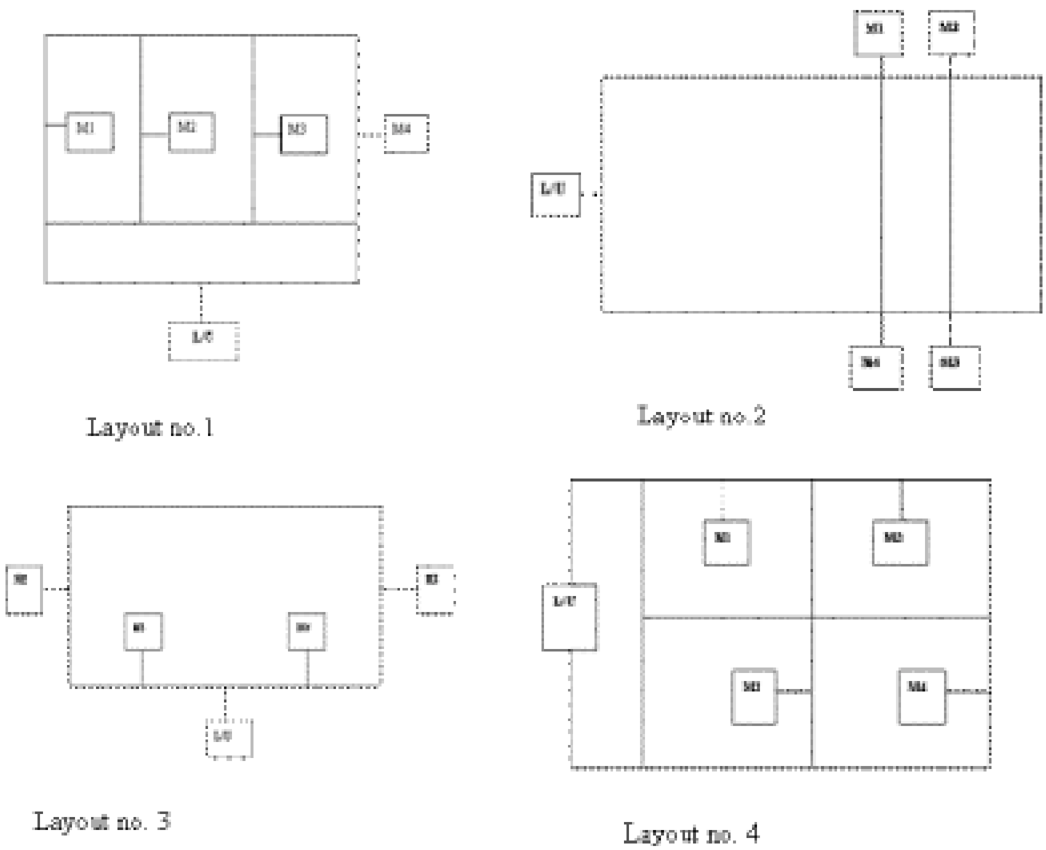
The constraint programming model presented in this article has a lot of variables and constraints that are sometimes redundant. These helped improve the performance. On the other hand, no numerical precision problems are experienced as in the case of the mixed integer formulation when big H is not carefully chosen.

4 Experimentation

Four layouts are considered in generating the test problems. They are presented in Figure 1. The layouts were presented in [8]. Each has 4 machines and one load/unload station. In the article, the authors presented the only literature test problems that can be regenerated. The test problems consist of 10 job sets with different routings (job-shop). Using these 10 sets with the four layouts, we generated 40 test problems.

In those problems, processing times on machines and material handling times are most of the time comparable. In many industrial settings, material handling times are much smaller than processing times on machines. Modifications were introduced to account for this reality. An AGV travels at an average speed of 3 km/h. Considering common dimensions of a shop and a lot production, handling times should be inferior to machining times. Handling times were consequently divided by 2. The 2-shortest paths required for testing the models can be generated by a k-shortest path generation algorithm such as Yen's algorithm [28].

A second modification is the use of a bidirectional network instead of a unidirectional network as in [8]. In their work, travel time between machines x and y is not the same as that between y and x. Detailed distance matrixes, job routes, processing times and route lengths are presented in Annex A.



L/U Load/Unload
M Machine

Figure 1: The four layouts

Tests were performed on a Pentium 4, 2.53GHz personal computer using OPLStudio version 3.6 that includes Cplex version 8. Result tables are presented hereafter. Tables 3 and 4 show results for the mathematical programming and the constraint programming model, respectively.

Data	nb. operations	nb. machines	nb. AGVs	Variables	Contstraints	Time (sec)	Objective	Nodes	Iterations
MP-CR-1-1	21	4	2	16995	6583	4,86	63	70	620
MP-CR-1-2	21	4	2	16995	9111	5,64	61	37	1232
MP-CR-1-3	21	4	2	16995	8711	5,45	66	70	1208
MP-CR-1-4	21	4	2	16995	8207	6,31	62	100	1780
MP-CR-2-1	24	4	2	25401	1896	1,94	74	47	287
MP-CR-2-2	24	4	2	25401	1972	2,36	71	55	427
MP-CR-2-3	24	4	2	25401	1968	2,17	73	37	266
MP-CR-2-4	24	4	2	25401	1940	2,14	73	35	209
MP-CR-3-1	27	4	2	35940	2271	2,66	75	5	163
MP-CR-3-2	27	4	2	35940	2373	3,17	72	2	111
MP-CR-3-3	27	4	2	35940	2363	3,13	74	1	81
MP-CR-3-4	27	4	2	35940	2327	2,91	74	11	160
MP-CR-4-1	33	4	2	113388	4465	20 *	62	—	—
MP-CR-4-2	33	4	2	113388	4686	15,31	56	40	385
MP-CR-4-3	33	4	2	113421	4671	25*	59	—	—
MP-CR-4-4	33	4	2	113408	4608	6*	57	—	—
MP-CR-5-1	21	4	2	16995	1438	1,55	52	0	148
MP-CR-5-2	21	4	2	16995	1502	1,86	49	0	149
MP-CR-5-3	21	4	2	16995	1490	1,80	53	0	111
MP-CR-5-4	21	4	2	16995	1460	1,73	50	9	142
MP-CR-6-1	30	4	2	66768	3371	9,27	95	446	4762
MP-CR-6-2	30	4	2	66768	3507	10,69	91	427	4214
MP-CR-6-3	30	4	2	66768	3491	10,84	94	508	3907
MP-CR-6-4	30	4	2	66768	3439	9,30	93	211	2376
MP-CR-7-1	31	4	2	51021	2876	6,95	66	645	2839
MP-CR-7-2	31	4	2	51021	2986	7,14	66	155	801
MP-CR-7-3	31	4	2	51021	2986	72	68	37831	175756
MP-CR-7-4	31	4	2	51021	2942	6,64	66	141	957
MP-CR-8-1	34	4	2	114556	4567	18,59	147	1151	9419
MP-CR-8-2	34	4	2	114556	4735	22,03	143	1124	9904
MP-CR-8-3	34	4	2	114556	4735	20,70	149	949	9886
MP-CR-8-4	34	4	2	114556	4643	28,97	148	2548	27267
MP-CR-9-1	29	4	2	65911	3329	8,11	88	271	2135
MP-CR-9-2	29	4	2	65911	3459	10,25	84	440	2926
MP-CR-9-3	29	4	2	65911	3459	10,22	89	501	3160
MP-CR-9-4	29	4	2	65911	3397	16,36	87	946	4230
MP-CR-10-1	36	4	2	146475	5232	36,66	121	1561	9768
MP-CR-10-2	36	4	2	146475	5446	29,94	117	1689	11488
MP-CR-10-3	36	4	2	146475	5394	28,77	119	1318	11337
MP-CR-10-4	36	4	2	146475	5310	38,14	120	2106	15098

* Time to find the solution without proof of optimality.

Table 3: Results for the mathematical programming formulation

The code in the first column of the table refers to a specific problem instance which is characterized by some information presented in the problems code that follows.

Problems Code
MP-CR-no.1-no.2

MP	Mathematical programming
CR	Choice of Routes
nb. 1	Number of the job set
nb.2	Number of the layout

The time noted in the table refers to time needed for compilation, solving and display of results. Results are obtained in acceptable time with proof of optimality (mean time of 13 seconds) except for problems PM-CR-4-1, PM-CR-4-3 et PM-CR-4-4, where time necessary for optimality proof is too long. Results for the two first problems, PM-CR-4-1 and PM-CR-4-3, are not optimal. The optimal solution for the first has the value of 61 and is obtained by solving the constraint programming formulation. For the second, the constraint programming formulation gives an objective of 58 without proof of optimality. For the third, PM-CR-4-4, we obtain an objective of 57 without proof of optimality. We note that these problem instances; present a recirculation phenomenon where jobs visit the same machine more than once.

Problems include a large number of variables and constraints and results are encouraging. We have tested machine scheduling problems including a comparable number of constraints and variables and solving was very difficult. In other terms, consideration of constraints relative to material handling system does not necessarily complicate resolution since they reduce the solutions space. Criticality of resources and the alternative character of some of them influence the solving time.

Problems code
CP-CR-nb.1-nb.2

CP	Constraint programming
CR	Choice of Routes
nb. 1	Number of the job set
nb.2	Number of the layout

In the table, choice points refer to different stages in the search for a solution. This corresponds to nodes in a branching tree for a mixed-integer problem. Failures are choice points that has been explored during the search and that correspond to infeasible or lower quality solutions compared to the current solution.

The solution times given for problems CP-CR-6-1, CP-CR-6-2, CP-CR-8-1, CP-CR-8-2, CP-CR-8-3, CP-CR-8-4, CP-CR-9-3, CP-CR-9-4 correspond to the time necessary to find the optimal solution without proof of optimality. Obtaining optimality proof was very long. Optimal solution was not obtained for problems CP-CR-4-2, CP-CR-4-3, CP-CR-4-4, CP-CR-10-1, CP-CR-10-2, CP-CR-10-3, CP-CR-10-4.

Data	nb operations	nb machines	nb chariots	Variables	Constraints	Time (sec)	Objective	Failures	Choice Points
CP-CR-1-1	21	4	2	971	5569	15	63	17638	18815
CP-CR-1-2	21	4	2	971	6778	26	61	24500	26176
CP-CR-1-3	21	4	2	971	5569	10	66	11324	12531
CP-CR-1-4	21	4	2	960	6773	14	62	19048	20522
CP-CR-2-1	24	4	2	1189	8829	21	74	24679	26329
CP-CR-2-2	24	4	2	1189	11284	27	71	10261	12966
CP-CR-2-3	24	4	2	1189	8813	28	73	26710	28307
CP-CR-2-4	24	4	2	1189	10363	83	73	32038	42846
CP-CR-3-1	27	4	2	1419	11258	55	75	57208	59316
CP-CR-3-2	27	4	2	1419	14207	30	72	12	12027
CP-CR-3-3	27	4	2	1419	13641	18	74	9	7069
CP-CR-3-4	27	4	2	1419	12763	27	74	9	14060
CP-CR-4-1	33	4	2	2353	29655	1739	61	118529	123133
CP-CR-4-2	33	4	2	2353	40484	250*	56**	8	19504
CP-CR-4-3	33	4	2	2353	37843	15524*	58**	156715	164530
CP-CR-4-4	33	4	2	2353	–	2711*	66**	–	–
CP-CR-5-1	21	4	2	971	5605	2,13	52	48	2938
CP-CR-5-2	21	4	2	971	6911	3,9	52	48	4176
CP-CR-5-3	21	4	2	971	6697	4,7	53	27	5979
CP-CR-5-4	21	4	2	971	6359	2,3	50	6	2953
CP-CR-6-1	30	4	2	1801	18780	78*	95	231	19638
CP-CR-6-2	30	4	2	1801	26912	400*	91	890	29844
CP-CR-6-3	30	4	2	1801	26184	170	94	206	13486
CP-CR-6-4	30	4	2	1801	24240	143	93	201	12074
CP-CR-7-1	31	4	2	1691	16383	30	66	41	11879
CP-CR-7-2	31	4	2	1691	21345	109	66	14	15965
CP-CR-7-3	31	4	2	1691	20967	120	68	42	16651
CP-CR-7-4	31	4	2	1961	19706	92	66	33	15084
CP-CR-8-1	34	4	2	2363	68344	500*	147	17	—-
CP-CR-8-2	34	4	2	2363	37112	600*	143	468	38339
CP-CR-8-3	34	4	2	2363	37664	500*	149	54	32746
CP-CR-8-4	34	4	2	2363	33124	650*	148	555	37091
CP-CR-9-1	29	4	2	1935	18661	60	88	138	15451
CP-CR-9-2	29	4	2	1935	25027	235	84	13	28741
CP-CR-9-3	29	4	2	1935	24664	90*	89	137	11911
CP-CR-9-4	29	4	2	1935	23188	125*	87	340	11712
CP-CR-10-1	36	4	2	2674	39961	194*	128**	17	7923
CP-CR-10-2	36	4	2	2674	54631	2010*	138**	61095	71982
CP-CR-10-3	36	4	2	2674	52047	2287*	131**	80649	91292
CP-CR-10-4	36	4	2	2674	46970	2480*	122**	94374	103293

* Time to find the solution without proof of optimality.
** Objective values not proven optimal.

Table 4: Results for the constraint programming formulation

All problem instances except PM/CP-CR-4-3 and PM/CP-CR-4-4 were optimally solved. However a rapid feasible solution is obtained for both instances. We also noticed, in the tests, that for some instances constraint programming had a better performance compared to mathematical programming. For other instances, the inverse was true.

We note that it is difficult to find the limits of the models in terms of maximal number of variables and of constraints. The solving times change depending on data. It is a combination of several factors that determines the difficulty of the problem. However, to our knowledge, it is the first time in literature that the integrated scheduling problem is optimally solved with both methodologies with conflicts accounted for.

5 Conclusion

We have presented in this article two models for the integrated production and material handling scheduling problem with routing conflicts considerations. Only for two of the test problems, we did not obtain optimal solutions. On the practical side, the developments presented in this paper represent a dual tool to practitioners for solving the problem. In the tests, the relative performance of the mathematical programming and the constraint programming models vary according to the problem instance. For the same number of operations, we may experience different levels of difficulty when using either of the techniques. Precedence constraints have an important impact on the solution time. On the other hand, processing time on resources combined with disjunction constraints determines how critical a resource is.

Development of the constraint programming model was an interesting empirical exercise. In spite of the overall better performance of the mixed-integer program, there are merits in the constraint programming formulation. It allows formulation of direct successors in an elegant and concise manner, to the opposite of the mixed-integer formulation.

Compared to results obtained in [14], this model allowed a better utilization of resources even when conflicts are accounted for. Considering a bidirectional network, in this article, contrary to [14], resulted in shorter schedules for some problem instances.

Material handling resource constraints influence the solving time for some instances. The scheduling tools can be used to test scenarios when more or less AGVs or routes are in the system. Sometimes, adding a resource does change the value of the makespan. If this happens repeatedly when solving most of the problem instances for a certain operational system, it justifies the addition of more resources. Hence, the scheduling tool can be used to handle design issues. The solution time allows the evaluation of different scenarios to determine resource requirements. Tight resource constraints for machines might also be problematic. The two proposed models can be easily modified to account for alternative machines.

Future research includes development of more complete formulations for the problem and developing search strategies especially for the constraint programming model. Other job-shop scheduling problems featuring resource constraints for buffers, for route segments and for intersections are also part of future work.

Annex A

2 shortest-path distance matrix between machines

Bidirectional arcs and adjusted handling times

	M1	M2	M3	M4
M1	0	r1,r2	r3,r4	r5,r6
M2	r1,r2	0	r7,r8	r9,r10
M3	r3,r4	r7,r8	0	r11,r12
M4	r5,r6	r9,r10	r11,r12	0

Length of shortest paths

Layout 1	
r1	3
r2	3
r3	4
r4	4
r5	5
r6	5
r7	3
r8	3
r9	4
r10	4
r11	3
r12	3

Layout 2	
r1	1
r2	3
r3	2
r4	2
r5	1
r6	3
r7	3
r8	1
r9	2
r10	2
r11	3
r12	1

Layout 3	
r1	2
r2	10
r3	4
r4	8
r5	2
r6	10
r7	6
r8	6
r9	4
r10	8
r11	2
r12	10

Layout 4	
r1	2
r2	6
r3	2
r4	7
r5	5
r6	7
r7	7
r8	3
r9	3
r10	7
r11	4
r12	6

Job routes

Set no. 1

Job (1) : M1(8) ;M2(16);M4(12)

Job (2) : M1(20) ;M3(10);M2(18)

Job (3) : M3(12) ;M4(8);M1(15)

Job (4) : M4(14) ;M2(18)

Job (5) : M3(10) ;M1(15)

Set no. 2

Job (1) : M1(10) ;M4(8)

Job (2) : M2(10) ;M4(18)

Job (3) : M1(10) ;M3(20)

Job (4) : M2(10) ;M3(15);M4 (12)

Job (5) : M1(10) ;M2(15);M4(12)

Job (6) : M1(10) ;M2(15);M3(12)

Set no. 3

Job (1) : M1(16) ;M3(15)

Job (2) : M2(18) ;M4(15)

Job (3) : M1(20) ;M2(10)

Job (4) : M3(15) ;M4(10)

Job (5) : M1(8) ;M2(10);M3(15);M4(17)

Job (6) : M2(10) ;M3(15);M4(8);M1(15)

Set no. 4

Job (1) : M4(11) ;M1(10);M2(7)

Job (2) : M3(12) ;M2(10);M4(8)

Job (3) : M2(7) ;M3(10);M1(9);M3(8)

Job (4) : M2(7) ;M4(8);M1(12);M2(6)

Job (5) : M1(9) ;M2(7);M4(8);M2(10);M3(8)

Set no. 5

Job (1) : M1(6) ;M2(12);M4(9)

Job (2) : M1(18) ;M3(6);M2(15)

Job (3) : M3(9) ;M4(3);M1(12)

Job (4) : M4(6) ;M2(15)

Job (5) : M3(3) ;M1(9)

Set no. 6

Job (1) : M1(9) ;M2(11);M4(7)

Job (2) : M1(19) ;M2(20);M4(13)

Job (3) : M2(14) ;M3(20);M4(9)

Job (4) : M2(14) ;M3(20);M4(9)

Job (5) : M1(11) ;M3(16);M4(8)

Job (6) : M1(10) ;M3(12);M4(10)

Set no. 7

Job (1) : M1(6) ;M4(6)

Job (2) : M2(11) ;M4(9)

Job (3) : M2(9) ;M4(7)

Job (4) : M3(16) ;M4(7)

Job (5) : M1(9) ;M3 (18)

Job (6) : M2(13) ;M3(19);M4(6)

Job (7) : M1(10) ;M2(9);M3(13)

Job (8) : M1(11) ;M2(9);M4(8)

Set no. 8

Job (1) : M2(12) ;M3(21);M4(11)

Job (2) : M2(12) ;M3(21);M4(11)

Job (3) : M2(12) ;M3(21);M4(11)

Job (4) : M2(12) ;M3(21);M4(11)

Job (5) : M1(10) ;M2(14);M3(18);M4(9)

Job (6) : M1(10) ;M2(14);M3(18);M4(9)

Set no. 9

Job (1) : M3(9) ;M1(12);M2(9);M4(6)

Job (2) : M3(16) ;M2(11);M4(9)

Job (3) : M1(21) ;M2(18);M4(7)

Job (4) : M2(20) ;M3(22);M4(11)

Job (5) : M3(14) ;M1(16);M2(13);M4(9)

Set no. 10

Job (1) : M1(11) ;M3(19);M2(16);M4(13)

Job (2) : M2(21) ;M3(16);M4(14)

Job (3) : M3(8) ;M2(10);M1(14);M4(9)

Job (4) : M2(13) ;M3(20);M4(10)

Job (5) : M1(9) ;M3(16);M4(18)

Job (6) : M2(19) ;M1(21);M3(11);M4(15)

References

1. JOHNSON, S.M. (1954). Optimal two and three stage production schedules with setup times included. Naval Research Logistics Quarterly, 1, 61-67.
2. MAGGU, P. L., DAS, G. and KUMAR, R. (1981). On equivalent-job for job-block in 2xn sequencing problem with transportation-times. Journal of the Operations Research Society of Japan, 24, 136-146
3. STERN, H. I., and VITNER, G. (1990). Scheduling parts in a combined production-transportation work cell. Journal of the Operational Research Society, 41/7, 625-632.
4. PANWALKER, S. (1991). Scheduling of a two-machine flowshop with travel time between machines. Journal of the Operational Research Society, 42, 609-613.
5. LEVNER, E., KOGAN, K. and LEVIN, I. (1995). Scheduling a two-machine robotic cell: a solvable case. Annals of Operations Research, 57, 217-232.
6. RAMAN, N., TALBOT, F. B. and RACHAMADUGU, R. V. (1986). Simultaneous scheduling of machines and material handling devices in automated manufacturing. . In K. E. Stecke and R. Suri (eds), Proceedings of the second ORSA/TIMS conference on flexible manufacturing Systems, 321-332. Elsevier Science Publishers B.V., Amsterdam.
7. LEE, D.Y. and DiCESARE, F. (1994). Integrated scheduling of flexible manufacturing systems employing automated guided vehicles," IEEE Transactions on Industrial Electronics, 41/6, 602-610.
8. BILGE, U. and ULUSOY, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS. Operations Research, 43/6, 1058-1070.
9. ULUSOY, G., FUNDA, S.-S. and BILGE, U. (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Computers and Operations Research, 24/4, 335-351.
10. SABUNCUOGLU, I. and KARABUK, S. (1998). Beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. IIE Transactions, 30/2, 179-191.
11. ANWAR, M. F. and NAGI, R. (1998). Integrated Scheduling of material handling and manufacturing activities for just in time production of complex assemblies. International Journal of Production Research, 36/3, 653-681.
12. ANWAR, M. F. and NAGI, R. (1997). Integrated conflict free routing of AGVs and workcenter scheduling in a just in time production environment. Proceedings of the 1997 6^{th} annual Industrial Engineering Research Conference, May 17-18. Miami, FL, USA.

13. SMITH, J., PETERS, B. and SRINIVASAN, A. (1999). Job shop scheduling considering material handling. International Journal of Production Research. 37/7, 1541-1560.
14. EL KHAYAT, G., LANGEVIN, A. and RIOPEL, D. (2003). Integrated production and material handling scheduling using mathematical programming and constraint programming. Proceedings CPAIOR'03, Montreal, Canada, 8-10 May, 276-290.
15. LEE, J. and MANEESAVET, R. (1999). Dispatching rail-guided vehicles and scheduling jobs in a flexible manufacturing system. International Journal of Production Research, 37/1, 111-123.
16. SABUNCUOGLU, I. and HOMMERTZHEIM, D. (1989a). An Investigation of machine and AGV scheduling rules in an FMS. In K. E. Stecke and R. Suri (eds), Proceedings of the third ORSA/TIMS conference on flexible manufacturing Systems, 261-266. Elsevier Science Publishers.
17. SABUNCUOGLU, I. and HOMMERTZHEIM, D. (1992a). Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion. International Journal of Production Research, 30/7, 1617-1635.
18. SABUNCUOGLU, I. and HOMMERTZHEIM, D.L. (1992b). Dynamic Dispatching Algorithm for Scheduling Machines and Automated Guided Vehicles in a Flexible Manufacturing System. International Journal of Production Research, 30/ 5, 1059-1079.
19. SABUNCUOGLU, I., and HOMMERTZHEIM, D.L. (1993). Experimental Investigation of an FMS due-date scheduling problem: Evaluation of machine and AGV scheduling rules. The International Journal of Flexible Manufacturing Systems, 5, 301-323.
20. SABUNCUOGLU, I. (1998). A study of scheduling rules of flexible manufacturing systems: a simulation approach. International Journal of Production Research, 36/2, 527-546.
21. JAWAHAR, N., ARAVINDAN, P., PONNAMBALAM, S. G., SURESH, R. K. (1998). AGV schedule integrated with production in flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 14/6, 428-440.
22. JAIN, V. and GROSSMANN, I.E. (2001). Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems. INFORMS Journal of Computing. 13/4.
23. PESANT, G., GENDREAU, M., POTVIN, J-Y. and ROUSSEAU J-M. (1998). Exact constraint logic programming algorithm for the traveling salesman problem with time windows. Transportation Science, 32/1, 12-28.
24. VAN HENTENRYCK, P. (1999). The OPL Optimization Programming Language. The MIT Press.
25. MARRIOT, K. and STUCKEY, P.J. (1998). Programming with constraints: an introduction. The MIT Press.

26. HOOKER, J. (2000). Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction. John Wiley & Sons.

27. VAN HENTENRYCK, P. (1989). Constraint Satisfaction in Logic Programming. The MIT Press.

28. YEN, J. Y. (1971). Finding the k-shortest loopless paths in a network. Management Science, 17, 712-715.