Integrated production and material handling scheduling using mathematical programming and constraint programming

Ghada El Khayat, André Langevin and Diane Riopel
École Polytechnique de Montréal and Gerad, Canada
C.P. 6079, Succ. Centre-ville Montréal (Québec) H3C 3A7
ghada.el-khayat@polymtl.ca, andre.langevin@polymtl.ca, diane.riopel@polymtl.ca

Abstract

In this article, we optimally solve an integrated production and material handling scheduling problem. Traditionally, scheduling problems consider machines as the only constraining resource. This is however no longer true as material handling equipment are becoming more and more valuable resources requiring important investments. Their operations should be optimized and above all synchronized with machine operations. In the problem, addressed in this paper, a job-shop context is considered. Machines and material handling equipment (AGVs) are considered constraining resources. The shortest path between two machines is used for a material handling operation or an empty travel. A mathematical programming model and a constraint programming model are presented for the problem and solved optimally on test problems from the literature and larger instances that we generated. A commercial software (ILOG OPLStudio) was used for modeling, testing and integrating both models in a decision support system. The performance of the two methods is comparable when using the data from the literature. However, for larger instances the performance of the constraint programming model was superior to the mathematical programming model.
Keywords: Scheduling, Job-shop, Material handling, Mixed integer programming, Constraint programming.

1 Introduction

Real-life scheduling problems are very complex, not only because of their combinatorial nature, but also because of constraints dictated by different production environments. These constraints are sometimes very specific and other times very general. In manufacturing, these include material handling constraints. Unfortunately, classical scheduling problems formulations neglect this reality. Researchers working on material handling problems address the material handling scheduling issue, often independently of the machine-scheduling problem. They typically propose, for a scheduling environment, the resolution of two independent eventually easier problems: machine scheduling and material handling scheduling. This simplification results in sub-optimal solutions for the real scheduling problem featuring resource constraints for both machines and material handling equipment. Recently, researchers who studied the problem of simultaneous scheduling of machines and material handling emphasized this fact. Contributions in this area presented heuristic solution approaches. Moreover, test problems represented very special cases in manufacturing where travel times for material handling equipment are comparable to processing times on machines if not longer. This hypothesis, although it justifies in part the simultaneous scheduling approach, is very restrictive. In this paper, we consider material handling times that are shorter than the processing times, which corresponds to real manufacturing settings.

Several researchers proposed frameworks for the representation of production systems: Hax and Meal (1975), Stecke (1984) and Pinedo (1995) to name some. Several others proposed integration schemes for different levels of decision making as Lasserre (1992) and Wein and Chevalier (1992). This paper presents an integration, at the same decision making level, among resources influencing a production schedule: machines and AGVs. From a practical standpoint, the two solution methodologies proposed in this work can be integrated in a decision support system for industrial organizations. OPLStudio supports both methodologies and integration can be envisaged through script files for special applications. This paper is organized as follows. Section 2 presents a literature revue. Section 3 presents the problem and a description of the operational system considered. The models: mathematical programming and constraint programming are presented in section 4. Section 5 presents the experimentation and the conclusion follows.

2 Literature review

The first contributions in integrated scheduling are extensions of the 2 machine flow-shop problem solved optimally by Johnson (1954). Researchers tried to solve the same problem considering material handling times, equipment availability and capacity constraints for buffers. Among others, Maggu et al. (1981), Stern and Vitner (1990), Panwalker (1991) and Levner et al. (1995) studied the problem for the minimization of the makespan. A larger size flow-shop problem was studied by Raman et al. (1986) in an FMS setting. Handling times were not sequence dependent.

In a job-shop context, Bilge and Ulusoy (1995) consider an integrated production and material handling scheduling problem with the objective of minimizing the makespan in an FMS. Unlike in Raman et al. (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 et al. (1997) study the same problem and propose a genetic algorithm for solving 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 deadhead time. There are no buffer considerations.

Lee and DiCesare (1994) 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 till 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.

Sabuncuoglu and Karabuk (1998) 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.

Smith et al. (1999) 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 first.

A 2-stage assembly production system composed of 4 machine cells is studied by Anwar and Nagi (1998). 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 (1997), study the same assembly shop. They propose a heuristic methodology that accounts for conflicts. No numerical results are reported.

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 (1989) studied scheduling rules for machines and AGVs in an FMS. In 1992, they studied integrated scheduling of production and material handling for an FMS with a job-shop production environment. 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. In 1992, Sabuncuoglu and Hommertzheim propose an on-line algorithm with a better vision compared to the scheduling rules. The algorithm considers more than one operation at a time. In 1993, the authors consider the experimental investigation of scheduling rules for a wide variety of objective functions. In 1998, they consider this investigation for the case of machine break-down. Jawahar et al. (1998) also study the dynamic problem. They present a heuristic that uses dispatching rules, accounting for conflicts, for AGVs.

Previous contributions consider inter-machine distances to be the most influencing for the material handling activities. Lee and Maneesavet (1999), 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 an SMF.
As presented, although different complementary aspects are integrated to the machine scheduling problem, solution methodologies are all heuristic. We optimally solve the integrated scheduling

problem, in a job-shop setting, for the first time. Methodologies we use: mathematical programming and constraint programming are also used for the first time to solve the problem.

3 Problem statement

The problem reads : given the shop layout 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 to minimize the makespan.

A job is composed of a 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 2 machines.

The operational system considered is a job-shop environment, with machines and AGVs referred to as workcentres. 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 follow the shortest path 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 the deadhead depends on the assignment sequence of the material handling operations to AGVs. All data are deterministic.

Five layouts are considered in generating the test problems. They are presented in Figure 1. The first 4 layouts were presented by Bilge and Ulusoy (1995). Each has 4 machines and one load/unload station. We propose a fifth layout that includes 8 machines and a load/unload station.

4 Models

Mathematical programming formulations proposed in the literature for job-shop scheduling problems are mixed integer and sometimes non-linear for disjunctive constraints. For large size problems, these are quite difficult to solve. Constraint programming is a rather new research area that has proven effective for scheduling applications (Baptiste et al., 2001). In section 4.1, we present a mixed-integer model for the integrated scheduling job shop. In section 4.2, we present a constraint programming model for the problem.

4.1 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 connectivity constraints ensuring that empty travel times are considered. 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. Conflicts are not considered.

4.1.1 Notation

O	Set of operations ($j \in O$), every operation belongs to a specific job
$O^M \subset O$	Set of material handling operations, every operation belongs to a specific job
$O^L \subset O$	Set of last operations of all jobs
$O^F \subset O$	Set of first operations of all jobs
W	Set of all workstations: machines and AGVs $w \in W$
$C \subset W$	Set of AGVs
$M \subset W$	Set of machines
O_w	Set of operations soliciting workstation w
$n(j)$	Following operation of operation *j*
S	Start of the time horizon
t_j	Processing time of operation *j*
k_{ij}	Empty travel time from the destination machine of material handling operation *i* to the source machine of material handling operation *j*
H	A big value
Ψ_{ij}	Variable having the value 1 if operation *i* precedes operation *j* on their requested resource, 0 otherwise
Φ_{jw}	Variable having the value 1 if the material handling operation *j* is assigned to AGV *w*, 0 otherwise
S_j	Variable indicating the start time of operation *j*
C_{max}	Variable indicating the end time of the schedule (makespan)

4.1.2 Mathematical programming model

$\text{Min } C_{max}$		(1)
$C_{max} \geq S_j + t_j$	$\forall j \in O^L$	(2)
$S_{n(j)} \geq S_j + t_j$	$\forall j \in O \backslash O^L$	(3)
$\Psi_{ij} + \Psi_{ji} = 1$	$\forall i,j \in O_w \mid i \neq j, \forall w$	(4)
$S_j \geq (S_i + t_i) + (\Psi_{ij} - 1)\, H$	$\forall w \in M, \forall i,j \in O_w \mid i \neq j$	(5)
$S_j \geq S_i + t_i + k_{ij} - (3 - \Psi_{ij} - \Phi_{iw} - \Phi_{jw})\, H$	$\forall w \in C, \forall i,j \in O_w \mid i \neq j$	(6)
$S_j >= S$	$\forall j \in O^F$	(7)
$\sum_w \Phi_{jw} = 1$	$\forall w \in C, \forall j \in O_w$	(8)
$\Psi_{ij} \in \{0,1\}$		(9)
$\Phi_{jw} \in \{0,1\}$		(10)
$S_j \geq 0$		(11)

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 are respected. Constraints (4) determine the order in which a machine or an AGV is visited, operation *i* before *j* or inversely *j* before *i*. Constraints (5) and (6) model the disjunctive character of the resources. For two operations that follow on a resource, the second cannot start before the first is accomplished. Constraints (6) also account for connectivity of the AGVs routes. They consider empty travel time when two material handling operations that follow on an AGV. Of course, H must be chosen as small as possible to get a better formulation. The first operations start after or at the beginning of the schedule horizon according to constraints (7). Constraints (8) state that a material handling operation is assigned to one AGV. Constraints (9-10,11) define binary and continuous variables respectively. The model was solved using OPLStudio version 3.6.

4.2 Constraint programming formulation

The constraint programming model is formulated using the same commercial software as the mathematical programming model (OPLStudio). The model uses software functions that enable solving by the solver (Scheduler), part of OPLStudio. This solver is specially designed for scheduling problems. It uses constraint programming algorithms. 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 resource types in OPLStudio (Van Hentenryck, 1999). For an introduction to constraint programming the reader is referred to Marriott and Stucky (1998). For a presentation of logic-based methods for optimization, we refer the reader to Hooker (2000).

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 standpoint. We can use one or the other. The instruction **ActivityHasSelectedRessource** 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.

4.2.1 Notation

In this section we present the notation for the constraint programming model. We present the declarative names as in the model formulated with OPLStudio. Next, we present the model using usual logical constraints. Furthermore, functions of the software used in modeling are stated in their names and described below and correspondence with the model constraints is indicated.

Machines	The machines
Vehicles	The AGVs
Tasks	Set of operations of machines and material handling
Tasksmach $\subset$ Tasks	Machine operation
Tasksman $\subset$ Tasks	Material handling operation
setOfPrecedences (t,q)| t et q $\in$ Tasks	Pairs of tasks with a precedence relation (t precedes q).
d[t]	Processing time of machine or material handling operation t
Resourcem[t]	Machine solicited by operation t
k[t,q]	Empty travel time between material handling tasks t and q assigned to the same AGV
S[t]	Variable (type : integer) indicating the start time of operation t
Hosts[t]	Variable (type: AGV)indicating the assignment of a material handling operation to an AGV
S[makespan]	End time of the schedule (makespan)

4.2.2 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 are not operations to accomplish. They arise as material handling operations are assigned to AGVs. They are accounted for by a global constraint that formulates the AGVs disjunction.

Minimize S[makespan]		(1)
$S[q] \geq S[t] + d[t]$	$\forall$ (t,q) $\in$ setOfPrecedences	(2)
$S[makespan] \geq S[t] + d[t]$	$\forall$ t$\in$ Tasks	(3)
$S[q] \geq S[t] + d[t] \lor S[t] \geq S[q] + d[q]$	$\forall$m|resourcem[t]=resourcem[q]=m	(4)
$Hosts[t] = Hosts[q] \Rightarrow S[q] \geq S[t] + d[t] + k[t,q] \lor S[t] \geq S[q] + d[q] + k[q,t]$	$\forall$ t,q $\in$ Tasksman	(5)

Table 2. Complete model in constraint programming

In (1) we minimize the makespan. Precedence relations between operations are modeled by constraints (2) and (3). Most of constraints (3) are redundant (except for the last operations) as they are already included in constraints (2). However, this affects the solution time positively. Disjunction constraints for machines are presented by constraints (4) 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 resource requirements are met. Constraints (5) account for AGVs disjunction and deadheads. It imposes that if 2 material handling operations follow on an AGV, empty travel time should be calculated before the second material handling operation starts. In the OPLStudio model, AGVs are declared **alternative** resources, as defined earlier. This declaration is also used in conjunction with a **requires** constraint to respect resource needs. Corresponding to constraints (5), we formulate a global constraint using **ActivityHasSelectedResource** instruction, that when holds for the same AGV implies the respect of the deadhead time. 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 (1989).

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. To our knowledge, this problem has not been formulated nor solved using the constraint programming technique. However, production scheduling problems (Jain and Grossmann, 2001) as well as vehicle routing problems (Pesant et al., 1996) were solved using the constraint programming technique and results were promising.

Complementary to the problem formulation, a good search procedure should be designed. Several procedures are available in OPLStudio. In our problem, we use **setTimes** that assigns starting times to activities. This procedure is efficient with discrete resources and when activity duration is fixed. We also use **assignAlternatives** procedure that proposes assignments of operations to AGVs or inversely AGVs to operations. It is a non deterministic instruction. Best results were obtained with a dichotomic search combined with Interleaved Depth-First Search (IDFS). This search strategy simulates a parallel depth-first search exploration of a search space on a sequential machine. The motivation is to avoid losing time due to early bad choices in the search.

5 Experimentation

The only complete test problems available in the literature are presented by Bilge and Ulusoy (1995). The test problems consist of 10 job sets with different routings (job-shop). Using these 10 sets with the first four layouts presented in Figure 1, we generated the first 40 test problems. Average processing time on machines is of 9 time-units. Average material handling or empty travel time is of 3.5 time units.

In those problems, processing times on machines and material handling times are comparable. Sometimes, material handling times were longer. In many industrial settings, material handling times are much smaller than processing times on machines.

In the beginning, these literature test problems were used without any modification. This constitutes our first set of problems presented in Table 3. Later, modifications were introduced. These include:
Adjusting material handling time: to become realistic and inferior to machining times. 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 to obtain the second set of problems presented in Table 4. A third AGV was added to the fleet for this second set of test problems to obtain the third set presented in Table 5.

A larger shop with a bidirectional network: the number of machines was doubled as well as the number of jobs in the shop to reach up to 16 jobs. Layout no. 4 was modified adding 4 more machines to generate layout no. 5. The Bilge and Ulusoy (1995) problems consider a unidirectional network. Consequently travel time between machines x and y is not the same as that between y and x. This hypothesis is of less importance as routing conflicts are not considered anyways. We consider a bidirectional network with the fifth layout to generate the fourth set of test problems presented in Table 7. Detailed distance matrixes, job routes and the processing times are presented in El Khayat (2003).

Tests were performed on a Pentium 4, 2.53GHz personal computer using OPLStudio version 3.6 that incorporates Cplex version 8. Result tables are presented hereafter. Table 2 shows results for literature data. All results are optimal. Only two test problems experienced long solving time (3700 and 1549 seconds). A third was solved in 66 seconds. All the rest of the test problems were solved in less than 27 seconds.

Problems code
MP/ CP-nb.1-nb.2 or MP/ CP-MA- nb.1-nb.2

MP / CP	Mathematical/ Constraint programming
MA	Material handling time adjusted
nb. 1	Number of the job set
nb.2	Number of the layout

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Nodes	Iterations
MP-1-1	21	4	2	551	339	0,11	72	344	1187
MP-1-2	21	4	2	551	339	0,03	72	5	71
MP-1-3	21	4	2	551	339	0,03	72	20	143
MP-1-4	21	4	2	551	339	0,03	70	13	112
MP-2-1	24	4	2	701	433	0,80	80	2328	61515
MP-2-2	24	4	2	701	433	0,11	72	99	713
MP-2-3	24	4	2	701	433	0,16	78	58	551
MP-2-4	24	4	2	701	433	2,02	82	6913	19210
MP-3-1	27	4	2	811	522	0,19	82	120	584
MP-3-2	27	4	2	811	522	0,13	74	22	323
MP-3-3	27	4	2	811	522	0,09	78	11	189
MP-3-4	27	4	2	811	522	0,44	82	833	1779
MP-4-1	33	4	2	1259	968	26,47	84	47329	155093
MP-4-2	33	4	2	1259	968	66,97	74	116263	413777
MP-4-3	33	4	2	1259	968	6,09	74	10395	34593
MP-4-4	33	4	2	1259	968	17,72	83	32831	95476
MP-5-1	21	4	2	551	338	0,13	59	306	797
MP-5-2	21	4	2	551	338	0,06	58	119	303
MP-5-3	21	4	2	551	338	0,11	56	22	157
MP-5-4	21	4	2	551	338	0,11	59	112	376
MP-6-1	30	4	2	1055	736	0,72	102	713	4391
MP-6-2	30	4	2	1055	736	0,61	94	532	3125

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Nodes	Iterations
MP-6-3	30	4	2	1055	736	0,61	98	476	3238
MP-6-4	30	4	2	1055	736	11,42	101	25344	71982
MP-7-1	31	4	2	1055	643	3700	79	967748	6142831
MP-7-2	31	4	2	1055	643	0,52	66	178	1223
MP-7-3	31	4	2	1055	643	12,02	69	14846	50725
MP-7-4	31	4	2	1055	643	1549	83	3917591	12782333
MP-8-1	34	4	2	1331	998	13,14	153	6777	45247
MP-8-2	34	4	2	1331	998	2,77	145	1062	8841
MP-8-3	34	4	2	1331	998	5,95	149	3129	17896
MP-8-4	34	4	2	1331	998	5,08	155	4035	21606
MP-9-1	29	4	2	991	718	1,55	97	1088	4204
MP-9-2	29	4	2	991	718	0,70	91	221	1304
MP-9-3	29	4	2	991	718	0,72	93	232	1492
MP-9-4	29	4	2	991	718	15,94	97	17945	54180
MP-10-1	36	4	2	1481	1115	7,22	127	4385	18321
MP-10-2	36	4	2	1481	1115	3,78	123	3938	11609
MP-10-3	36	4	2	1481	1115	1,28	126	720	4159
MP-10-4	36	4	2	1481	1115	24,47	130	29867	68709

Table 3. Results for the literature data

Three strategies were used to reduce running time. The effect was remarkable especially for problems 7-1 and 7-4 that needed several days before proving optimality. *Heuristic upper bound:* if needed, the program is run and then stopped after the first solution found (in less than 30 seconds) which is then used as an upper bound. *Constraints 7* affect significantly the solution time. In the problems, time horizons starts at zero. However, this redundant constraint acts as an effective cut and reduces the solving time. This constraint also reflects a practical aspect that is starting the scheduling horizon at different points of time for some jobs or some resources. *A valid constraint* was also introduced to impose that at least one material handling operation is assigned to every AGV in the fleet. Test problems with shorter material handling times, where the AGVs are less demanded are easier to solve. All test problems are solved in less than 20 seconds.

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Nodes	Iterations
MP-MA-1-1	21	4	2	551	336	0,06	63	20	123
MP-MA-1-2	21	4	2	551	336	0,02	61	3	72
MP-MA-1-3	21	4	2	551	336	0,13	65	8	93
MP-MA-1-4	21	4	2	551	336	0,14	65	8	93
MP-MA-2-1	24	4	2	701	431	0,22	77	48	365
MP-MA-2-2	24	4	2	701	431	0,14	74	38	400
MP-MA-2-3	24	4	2	701	431	0,11	77	36	359
MP-MA-2-4	24	4	2	701	431	0,11	76	20	316
MP-MA-3-1	27	4	2	811	520	0,22	75	45	323
MP-MA-3-2	27	4	2	811	520	0,20	72	41	461
MP-MA-3-3	27	4	2	811	520	0,25	74	50	409
MP-MA-3-4	27	4	2	811	520	0,14	73	33	264
MP-MA-4-1	33	4	2	1259	966	9,58	61	13257	30758
MP-MA-4-2	33	4	2	1259	966	0,34	57	50	307
MP-MA-4-3	33	4	2	1259	966	0,31	58	48	312

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Nodes	Iterations
MP-MA-4-4	33	4	2	1259	966	0,31	61	51	385
MP-MA-5-1	21	4	2	551	336	0,16	52	10	122
MP-MA-5-2	21	4	2	551	336	0,09	53	23	127
MP-MA-5-3	21	4	2	551	336	0,03	52	10	95
MP-MA-5-4	21	4	2	551	336	0,03	51	0	94
MP-MA-6-1	30	4	2	1055	734	0,44	95	209	1316
MP-MA-6-2	30	4	2	1055	734	0,44	91	195	1108
MP-MA-6-3	30	4	2	1055	734	0,42	92	165	1060
MP-MA-6-4	30	4	2	1055	734	0,33	93	55	516
MP-MA-7-1	31	4	2	1055	641	0,28	66	60	534
MP-MA-7-2	31	4	2	1055	641	0,33	66	133	693
MP-MA-7-3	31	4	2	1055	641	0,33	66	89	772
MP-MA-7-4	31	4	2	1055	641	0,70	67	775	3014
MP-MA-8-1	34	4	2	1331	996	4,38	147	4118	17447
MP-MA-8-2	34	4	2	1331	996	19,08	143	20740	79123
MP-MA-8-3	34	4	2	1331	996	8,83	145	9561	36738
MP-MA-8-4	34	4	2	1331	996	4,69	148	4673	20022
MP-MA-9-1	29	4	2	991	716	0,38	88	127	757
MP-MA-9-2	29	4	2	991	716	0,38	88	154	914
MP-MA-9-3	29	4	2	991	716	0,31	88	83	713
MP-MA-9-4	29	4	2	991	716	0,36	87	106	809
MP-MA-10-1	36	4	2	1481	1113	1,33	121	900	4009
MP-MA-10-2	36	4	2	1481	1113	0,88	119	432	2002
MP-MA-10-3	36	4	2	1481	1113	1,30	121	857	3933
MP-MA-10-4	36	4	2	1481	1113	0,95	120	422	2235

Table 4. Results for the literature data with adjusted material handling times

The resource constraints affect significantly the solution time as shown by the results when increasing the number of AGVs from 2 to 3.

Data	Operations	Machines	AGVs	Variables	Contstraints	Time (sec)	Objective	Nodes	Iterations
MP-MA-1-1	21	4	3	573	448	0,06	63	11	127
MP-MA-1-2	21	4	3	573	448	0,14	61	7	97
MP-MA-1-3	21	4	3	573	448	0,17	65	11	116
MP-MA-1-4	21	4	3	573	448	0,05	65	11	145
MP-MA-2-1	24	4	3	726	575	0,14	77	98	491
MP-MA-2-2	24	4	3	726	575	0,17	74	60	399
MP-MA-2-3	24	4	3	726	575	0,28	77	95	514
MP-MA-2-4	24	4	3	726	575	0,31	76	101	572
MP-MA-3-1	27	4	3	838	700	0,31	75	55	345
MP-MA-3-2	27	4	3	838	700	0,25	72	43	299
MP-MA-3-3	27	4	3	838	700	0,27	74	49	339
MP-MA-3-4	27	4	3	838	700	0,28	73	68	366
MP-MA-4-1	33	4	3	1293	1330	0,44	57	60	309
MP-MA-4-2	33	4	3	1293	1330	0,56	54	120	464
MP-MA-4-3	33	4	3	1293	1330	0,52	55	110	460

Data	Operations	Machines	AGVs	Variables	Contstraints	Time (sec)	Objective	Nodes	Iterations
MP-MA-4-4	33	4	3	1293	1330	0,61	57	197	729
MP-MA-5-1	21	4	3	573	448	0,08	52	11	117
MP-MA-5-2	21	4	3	573	448	0,05	53	10	119
MP-MA-5-3	21	4	3	573	448	0,16	52	8	111
MP-MA-5-4	21	4	3	573	448	0,05	51	20	145
MP-MA-6-1	30	4	3	1086	998	0,77	95	246	1839
MP-MA-6-2	30	4	3	1086	998	0,75	91	269	2373
MP-MA-6-3	30	4	3	1086	998	0,76	92	291	2070
MP-MA-6-4	30	4	3	1086	998	0,67	93	236	1402
MP-MA-7-1	31	4	3	1086	861	0,50	66	139	741
MP-MA-7-2	31	4	3	1086	861	0,41	66	59	394
MP-MA-7-3	31	4	3	1086	861	0,41	66	81	484
MP-MA-7-4	31	4	3	1086	842	0,94	67	421	2920
MP-MA-8-1	34	4	3	1366	1360	7,80	147	6093	20186
MP-MA-8-2	34	4	3	1366	1360	16,95	143	11687	59454
MP-MA-8-3	34	4	3	1366	1360	5,63	145	3390	17517
MP-MA-8-4	34	4	3	1366	1360	2,88	148	1741	7287
MP-MA-9-1	29	4	3	1021	980	0,42	88	133	750
MP-MA-9-2	29	4	3	1021	980	41,03	88	62202	67998
MP-MA-9-3	29	4	3	1021	980	0,38	88	110	749
MP-MA-9-4	29	4	3	1021	980	0,44	87	127	807
MP-MA-10-1	36	4	3	1518	1533	1,28	121	502	2833
MP-MA-10-2	36	4	3	1518	1533	71,06	119	74705	94860
MP-MA-10-3	36	4	3	1518	1533	3,81	121	2098	5600
MP-MA-10-4	36	4	3	1518	1533	1,73	120	800	4189

Table 5. Results for data with adjusted material handling times and 3 AGVs in shop

According to these results, we have noticed that the problems relatively difficult to solve take less time if we introduce a third AGV in the system. This is the case for problem PM-4-1 and PM-8-2. On the other hand, some problems easy to solve with 2 AGVs in the system take more time to solve if we introduce a third AGV. This is the case for problem PM-9-2 where solving time passes from 0,38 seconds to 41,03 seconds. For problem PM-10-2 solving time passes from 0,88 seconds to 71,06 seconds. Solving the problem with 2 and 3 AGVs gives important information on whether or not the AGV fleet in undersized, or if the resource constraints for AGVs are very tight. Instances of 8 machines/16 jobs are very difficult to solve and they need several hours and sometimes days before proof of optimality. However, the optimal solution is obtained in a few minutes. To test the constraint programming model, the same test problems were used. Results are encouraging, especially for bigger instances. The optimal solution is obtained in a few seconds. Results are presented below:

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Failures	Choice Points
CP-1-1	21	4	2	154	170	0,05	72	65	176
CP-1-2	21	4	2	154	170	0,00	72	4	69
CP-1-3	21	4	2	154	170	0,00	72	4	63
CP-1-4	21	4	2	154	170	0,05	70	32	188
CP-2-1	24	4	2	175	210	0,38	80	188	1372

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Failures	Choice Points
CP-2-2	24	4	2	175	210	0,05	72	4	106
CP-2-3	24	4	2	175	210	0,06	78	21	405
CP-2-4	24	4	2	175	210	0,72	82	812	1275
CP-3-1	27	4	2	189	252	0,27	82	119	868
CP-3-2	27	4	2	189	252	0,05	74	9	195
CP-3-3	27	4	2	189	252	0,14	78	6	110
CP-3-4	27	4	2	189	252	0,36	82	204	1158
CP-4-1	33	4	2	238	458	37,98	84	19246	36323
CP-4-2	33	4	2	238	458	9,70	74	4993	7763
CP-4-3	33	4	2	238	458	1,24	74	647	1622
CP-4-4	33	4	2	238	458	7,56	83	4103	11043
CP-5-1	21	4	2	154	170	0,06	59	38	132
CP-5-2	21	4	2	154	170	0,02	58	4	57
CP-5-3	21	4	2	154	170	0,02	56	10	77
CP-5-4	21	4	2	154	170	0,03	59	50	142
CP-6-1	30	4	2	217	348	0,25	102	55	442
CP-6-2	30	4	2	217	348	0,05	94	8	53
CP-6-3	30	4	2	217	348	0,05	98	10	135
CP-6-4	30	4	2	217	348	0,97	101	675	842
CP-7-1	31	4	2	217	302	294	79	733628	733648
CP-7-2	31	4	2	217	302	0,39	66	223	700
CP-7-3	31	4	2	217	302	0,23	69	42	690
CP-7-4	31	4	2	217	302	1816	83	4942737	4942747
CP-8-1	34	4	2	245	460	0,33	153	36	1135
CP-8-2	34	4	2	245	460	0,14	145	5	75
CP-8-3	34	4	2	245	460	0,06	149	7	75
CP-8-4	34	4	2	245	460	0,38	155	1502	1519
CP-9-1	29	4	2	210	346	0,22	97	37	379
CP-9-2	29	4	2	210	346	0,08	91	9	114
CP-9-3	29	4	2	210	346	0,09	93	14	120
CP-9-4	29	4	2	210	346	0,27	97	105	182
CP-10-1	36	4	2	259	522	0,44	127	99	477
CP-10-2	36	4	2	259	522	0,33	123	64	417
CP-10-3	36	4	2	259	522	0,30	126	39	446
CP-10-4	36	4	2	259	522	0,67	130	203	1067

Table 6. Results of the constraint programming model for the literature data

These first results presents a slightly better performance compared to the results obtained with the mixed integer programming model. The 8 machines/16 jobs instances are solved optimally in a maximum of 60 seconds unlike the mixed-integer model. These problems have more variables and more constraints but results are very good. Problems with adjusted (shorter) material handling times with 2 and 3 AGVs also presented an excellent performance. Instances were solved in few seconds.

We note that it is difficult to define limits for the models in terms of maximal number of variables and of constraints. Solving times change depending on data. It is a combination of several factors that determines the difficulty of the problem. In trail to test the limits, the (mt10) 10x10 problem of Fisher and Thompson, available in the electronic OR-library, was solved. Data corresponding to the material handling system was added and a 10 machine layout that represents an extension of layout 4 was considered. The constraint programming model presented excellent results. The optimal solution was obtained in 30 minutes and the proof of optimality was obtained in two hours. The mixed integer model ran for one week without finding the optimal solution. Clearly, this benchmark problem is difficult to solve whether we consider material handling constraints or not. However, adding the material handling constraints to the model did not deteriorate the performance in the constraint programming case. This resulted in obtaining a feasible schedule for a shop.

Data	Operations	Machines	AGVs	Variables	Constraints	Time (sec)	Objective	Failures	Choice Points
CP-11-5	42	8	2	301	596	0,09	76	220	243
CP-12-5	48	8	2	343	744	0,17	138	108	127
CP-13-5	52	8	2	371	904	1,64	94	4939	4983
CP-14-5	62	8	2	441	1450	60	131	21055	21090
CP-15-5	42	8	2	301	596	0,16	84	10	484
CP-16-5	54	8	2	385	1410	1,70	131	864	905
CP-17-5	60	8	2	427	1088	0,13	132	83	105
CP-18-5	68	8	2	483	1704	0,86	205	97	125
CP-19-5	64	8	2	455	2261	11,13	114	3979	4013
CP-20-5	60	8	2	427	1272	3.63	134	7566	7601

Table 7. Results of the constraint programming model for the 8-machines shop

6 Conclusion

Constraint programming is a technique at crossroads of computer science, artificial intelligence and operations research. Using this technique in solving the integrated scheduling problem was a successful and promising experience. However, using a commercial solver for which we don't know much about the underlying algorithms makes the performance unpredictable. This difficulty was confirmed by Marriott and Stuckey (1998). Choosing the appropriate model becomes an empirical exercise. For a problem, performance might vary according to the software used.

On the practical side, the developments presented in this paper represent a dual tool to practitioners, with which the problem is solved to optimality using one technique or the other. Mathematical programming and constraint programming are not necessarily appropriate for the same type of problem. Scheduling problems present another difficulty. For the same number of operations we may experience different levels of difficulty when using either of the techniques. This is, in part, because of precedence constraints that impose temporal constraints on start time variables. These differ from a test problem to another. On the other hand, processing time on resources combined with disjunction constraints determines how critical a resource is. In our tests, we noticed that for some instances constraint programming had a better performance compared to mathematical programming, and for other instances, it was the inverse. Development of an efficient tool based on a script integrating the two methods is undergoing.

Material handling resource constraints influence the solving time for some instances. When we increase the number of AGVs, solution time rapidly decreases. This is the case for the two solution methods used. Sometimes, adding a resource does not change the value of the objective. However, in other instances, the value of the objective decreases. If this happens repeatedly when solving most of the problem instances for a certain operational system, it justifies the addition of a material handling equipment. Hence, the scheduling tool can be used to handle design issues. The speed of solution 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 include testing larger instances, conflict avoidance and limited buffer capacities in a job-shop setting. Also, we think that bottleneck resources is an interesting concept on which Adams et al. (1988) based their important development of the Shifting Bottleneck procedure. Using their heuristic, they optimally solved the (10*10) test problem presented by Muth and Thompson (1963) that remained unsolved for more than 20 years. Their method was generalized by Ramudhin and Marier (1996) for solving different classes of scheduling problems. We think it is important to develop methods to characterize scheduling problems in terms of resource criticality in a formal fashion. This should help choosing the appropriate solution approach.

References

ADAMS, J., BALAS, E. and ZAWAK, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science,* 34/3, 391-401.

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.

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 6th annual Industrial Engineering Research Conference, May 17-18. Miami, FL, USA.

BAPTISTE, P. LE PAPE, C. and NUIJTEN, W. (2001). *Constraint-Based Scheduling - Applying Constraint Programming to Scheduling Problems.* Kluwer Academic Publishers.

BEASLEY, J. E., OR-Library test problems, http://mscmga.ms.ic.ac.uk/info.html

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.

EL KHAYAT, G. (2003). Ordonnancement intégré de la production et de la manutention. Ph.D. Dissertation, École Polytechnique de Montréal, Canada.

HAX, A. C. and Meal, H. (1975). Hierarchical Integration of production planning and scheduling. In *Logistics* (M. A. GEISLER, Ed.), 53-69. North Holland, Amsterdam.

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

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.

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.

JOHNSON, S.M. (1954). Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1, 61-67.

MARRIOT, K. and STUCKEY, P.J. (1998). *Programming with constraints: an introduction.* The MIT Press.

MUTH, J.F., and THOMPSON, J. L. (1963). *Industrial Scheduling.* Prentice Hall, Englewood Cliffs, N. J.

KUTANOGLU, E. and SABUNCUOGLU, I. (1999). Analysis of heuristics in a dynamic job-shop with weighted tardiness objectives. *International Journal of Production Research*, 37/1, 165-187.

LASSERRE, J. B. (1992). An integrated model for job-shop planning and scheduling, *Management Science*, 38/8, 1201-1211.

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.

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.

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.

LEVNER, E., KOGAN, K. and LEVIN, I. (1995). Scheduling a two-machine robotic cell: a solvable case. *Annals of Operations Research*, 57, 217-232.

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

PANWALKER, S. (1991). Scheduling of a two-machine flowshop with travel time between machines. *Journal of the Operational Research Society*, 42, 609-613.

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.

PINEDO, M. (1995). *Scheduling : Theory , algorithms and systems,* Prentice Hall, Englewood Cliffs, New Jersey.
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.
RAMUDHIN, A. and MARIER, P. (1996). The generalized shifting bottleneck procedure. *European Journal of Operational Research*, 93/1, 34-48.
SABUNCUOGLU, I. (1998). A study of scheduling rules of flexible manufacturing systems: a simulation approach. *International Journal of Production Research*, 36/2, 527-546.
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.
SABUNCUOGLU, I. and HOMMERTZHEIM, D. (1992 a). Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion. *International Journal of Production Research*, 30/7, 1617-1635.
SABUNCUOGLU, I. and HOMMERTZHEIM, D.L. (1992 b). Dynamic Dispatching Algorithm for Scheduling Machines and Automated Guided Vehicles in a Flexible Manufacturing System. *International Journal of Production Research*, 30/ 5, 1059-1079.
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.
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.
SMITH, J., PETERS, B. and SRINIVASAN, A. (1999). Job shop scheduling considering material handling. *International Journal of Production Research*. 37/7, 1541-1560.
STECKE, K. (1984). Design, planning, scheduling and control problems of flexible manufacturing systems. *Proceedings First ORSA /TIMS*, *Conference on Flexible Manufacturing Systems*, Ann Arbor, Michigan, USA.
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.
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.
VAN HENTENRYCK, P. (1989). *Constraint Satisfaction in Logic Programming.* The MIT Press.
VAN HENTENRYCK, P. (1999). *The OPL Optimization Programming Language.* The MIT Press.
WEIN, L. M. and CHEVALIER, P.B. (1992). A broader view of the job-shop scheduling problem. *Management Science*, 38/7, 1018-1033.

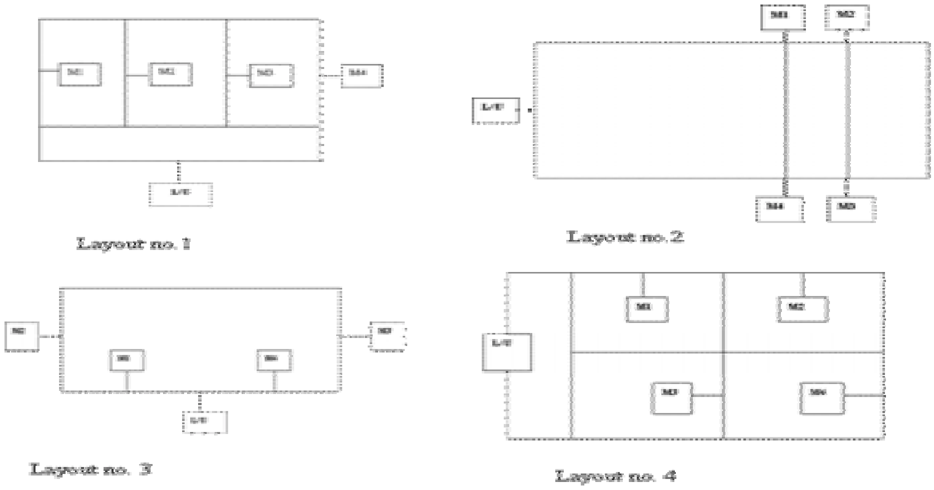

Figure 1. The layouts