An Improved Hybrid Model for the Generic Hoist Scheduling Problem

Daniel Riera (daniel.riera@uab.es)
Universitat Autònoma de Barcelona, Spain

Neil Yorke-Smith (nys@icparc.ic.ac.uk)
IC–Parc, Imperial College, London, United Kingdom

Abstract. The generic hoist scheduling problem is NP-hard and arises from automated manufacturing lines. In recent work using the constraint logic programming (CLP) formalism, a unified model has been developed with the problem description and solution method separated. We provide an improved model and new preprocessing stages where, as before, solutions and proof of optimality are provided by a hybrid CLP–MIP algorithm. The new algorithm is more scalable and robust. We give empirical results for a range of problem classes on benchmark problems from several sources.

Keywords: hoist scheduling, hybrid methods, modelling, benchmarks

1. Introduction

Hoist scheduling is an abstraction of a common industrial problem. Computer-controlled *hoists* (or transport robots) are used in PCB electroplating and other sectors to move material through some fixed sequence of operations. The importance of optimising the hoist movements is that the same procedure is performed continuously for many weeks, and a change in the production run may require several weeks of downtime (Shapiro and Nuttle, 1988).

The first solutions to the hoist scheduling problem used mathematical programming (Phillips and Unger, 1976). Later, artificial intelligence techniques in the forms of local search and constraint logic programming (CLP) were applied (Baptiste et al., 1994; Lam, 1997). More recently, a hybrid technique that combines MIP and CLP has been developed (Rodošek and Wallace, 1998). In this paper, we elaborate and generalise the hybrid approach, presenting a revised CLP model and new preprocessing stages.

Below we introduce and classify the hoist scheduling problem (HSP) in more detail, and, in Section 2, present the new model. Section 3 discusses the solvers used and the hybridization between them, and Section 4 gives our results for a number of benchmark problems. Finally, we draw some conclusions in Section 5.

Published in Annals of Operations Research **115** (2002). Some minor oversights have been corrected in this version.

1.1. Hoist Scheduling

The simplest case of the hoist scheduling problem is the *basic problem* introduced by Phillips and Unger (1976). A single computer-controlled hoist operates on a single track above a sequential line of tanks. A large number of identical jobs are placed at the initial stage of the line. Each job is to be processed through the tanks and placed at the far end.

Hoist scheduling is distinguished from classical scheduling problems, such as flowshop or jobshop, in that, first, no waiting of jobs during an operation is permissible, second, travel times of an operation are not negligible, and third, there is no intermediate storage between tanks. In addition, to reflect the industrial situation, the prescribed processing times in the tanks are bounded in a time window but need not be fixed.

Jobs are assumed to be identical in the basic problem. If the jobs differ or fall into multiple types, the nature of the HSP changes and the algorithms are quite different. The same occurs if the arrival of jobs is not known in advance (the *dynamic* hoist scheduling problem (Lamothe et al., 1994)).

In the *cyclic* hoist scheduling problem (CHSP), the same sequence of operations is repeated. One complete sequence is a *cycle*, and the length of time required for one cycle is the *period* (or *cycle time* or *makespan*). In a *single part* problem, one job enters and one job leaves the system in every cycle. In a *multi-part* or *r-part* system, r jobs enter and leave in every cycle (Kats et al., 1999). Generalising the CHSP, the *n-periodic* hoist scheduling problem (Levner et al., 1995) has a cycle composed of n smaller repeated sequences. The most common case in industry is the cyclic hoist scheduling problem with homogeneous jobs (Shapiro and Nuttle, 1988).

The basic problem does not account for all of industrial practice. The most notable extension is to multiple hoists, which may share one track or have one track each. If the number of hoists is greater than the number of tracks, collisions must be avoided, which necessitates *hoist assignment* (how hoists are assigned to tanks). Other variations from the basic problem include a single load/unload stage rather than one stage at each end, and *multi-function* tanks, those visited by a job more than once. Either of these possibilities entails bidirectional hoist movement while carrying a job. Some tanks may be duplicated or have a capacity greater than one, in order to reduce a bottleneck in the line.

Varnier et al. (1997) give a partial survey of the literature on the forms of the HSP. To further aid in the classification, we propose a notation for the problem classes similar to that used in queueing theory. Denote a class of the HSP as $\mathsf{F/H/T/A/r}$ to indicate:

– F: zero or more of the flags:

  - C: cyclic
  - D: dynamic
  - H: heterogeneous jobs
  - N: non-sequential
- H: the number of hoists, 1 (single) or M (multiple)
- T: the number of tracks, 1 (single) or M (multiple)
- A: hoist assignment[1], one of:
  - D (determined, i.e. partitioned)
  - C (collision-based)
  - O (optimal)
- r: the number of parts, 1 or an integer r

Terms may be omitted where implied or irrelevant: for instance, we consider only single-part problems so will omit the r term. Phillips and Unger's problem PU12, introduced below, is classified as C/1/1, for example, whereas the most general problem considered by Rodošek and Wallace (1998) is C/M/M/C. The notation is easily extensible.

1.2. Previous Work

1.2.1. *MIP Approaches*

Phillips and Unger (1976) introduced the HSP and provided the now-famous benchmark problem, PU12. They used a MIP model to find the minimum cycle time for this real-world twelve-tank problem[2].

Shapiro and Nuttle (1988) introduced a revised branch-and-bound procedure and used MIP on different sub-problems to bound the search space. Levner et al. (1995) improved the upper bound for the period.

Lei and Wang (1991) solved the HSP for two hoists on the same track. They introduced a new heuristic algorithm which partitions the line of tanks into two contiguous sets and assigns to each hoist one set. Lei et al. (1993) gave an improved heuristic, where the hoists must be scheduled to avoid traffic collisions (no partitioning is used). They were not able to guarantee the optimal solution.

There have been a number of works presenting progressively better or broader MIP models (Armstrong et al., 1992; Chen et al., 1998; Yih et al., 1993; Leung et al., 1997) and branch-and-bound procedures (Lei and Wang, 1994; Ng and Leung, 1997). Bidirectional multiple hoists were considered first by Manier (1994).

1.2.2. *CP Approaches*

The first approach to exploit constraint programming was by Baptiste et al. (1992). They demonstrated that the versatility of CLP allowed the rapid development of computational models for different classes of hoist scheduling problems. Their results showed that CLP with a linear solver is more effective than constraint propagation over finite domains. They were able to produce an optimal schedule for the problem PU12, with a revised model, in less than one minute.

Varnier and others (Varnier et al., 1997; Manier et al., 2000) extended this work to model multiple hoists and non-sequential treatment, including multi-function and duplicated tanks but not higher capacity tanks. They resolved the hoist assignment in only a restricted form using heuristics.

Cheng and Smith (1996) considered a multi-product single-hoist HSP where each job may require treatment in a subset of the tanks (tank skipping). Mak et al. (1998) used constraint satisfaction to solve the single-hoist CHSP with multi-function and duplicated tanks and a single load/unload stage.

1.2.3. *Hybridization and Hybrid Approaches*

Tsang et al. (1999) document a range of difficult combinatorial problems solved by exploiting integer programming (IP) and constraint programming (CP) together in hybrid approaches. Hooker et al. (2000) provide a generic scheme for hybridization between optimisation and constraint satisfaction methods, emphasising the complementary strengths of the two methods respectively in search and relaxation and in inference and strengthening. Several researchers report on problems not solvable in reasonable time by either IP or CP alone, but which fall to a combined approach (Baptiste et al., 1998).

A hybrid model was first applied to the HSP by Rodošek and Wallace (1998). Their model was resolved using propagation and search in the CLP platform ECL^{i}PS^{e} and linear solving in an external package. The advantage of hybridization for the HSP has been to extend the class of problems that can be handled by a single model, to allow a single solver algorithm to work across these classes, and to provide proof of optimality in cases which are beyond traditional methods.

Rodošek and Wallace (1998) consider a generic class of CHSP with single or multiple hoists, tracks, and tank capacities. In their hybrid, every constraint is passed to a CLP solver and a MIP solver. They introduced a harder thirteen-tank variant, PU13, of the Phillips and Unger problem, in which jobs are moved from the load stage to the first tank by a separate mechanism. To be able to compare results, we will consider PU13 rather than PU12.

Other algorithmic approaches to the problem were partially surveyed in Hall et al. (1997). For example, petri-nets (Denat et al., 2000), genetic algorithms for the single-hoist instance (Lim, 1997), and local search, notably simulated annealing (Lam, 1997). With the exception of certain restricted or simplified cases, all the hoist scheduling problems introduced above have been shown to be NP-hard (Hanen, 1994).

This paper contributes a more scalable and robust hybrid model, improving the work of Rodošek and Wallace (1998). We provide results analysing the performance of the three main approaches to the HSP on known benchmark problems, and point out future enhancements for the hybridization.

2. The Model

The expressiveness of CLP allows the easy modelling of both the linear and disjunctive constraints in the HSP. We exploit the automatic translation of Rodošek et al. (1997) to produce from the declarative CLP model one suitable for a MIP solver.

2.1. Variables and Notation

The following are all integers, with the decision variables in bold face:

N	number of tanks
R	number of tracks
H	number of hoists
J	number of simultaneous jobs
T	number of treatments
$S(i)$	tank for i^{th} treatment
$C(j)$	capacity of tank j
$m(j)$	minimum treatment time in tank j
$M(j)$	maximum treatment time[3] in tank j
$E(j,k)$	time to move empty hoist from tank j to tank k
$F(j,k)$	time to move full hoist from tank j to tank k
$T(i)$	actual time of i^{th} treatment
$\mathbf{R(i)}$	removal time upon completion of i^{th} treatment
$\mathbf{B(i)}$	number of the hoist that performs the i^{th} transfer operation
$\mathbf{P}$	cycle period

The hoists and tanks are numbered from left to right; the load and unload tanks are considered to be tanks 0 and $N+1$ respectively[4]. We take the i^{th} transfer operation as being from tank $S(i)$ to tank $S(i+1)$;

hence $T \geq N+1$ if every tank is used (strict inequality is possible in the case of multi-function tanks).

For all tanks, $E(j,k) = E(k,j)$. For non-duplicated tanks, $E(j,j) = 0$, while for duplicated tanks, following Shapiro and Nuttle (1988), $E(j,j) \neq 0$ and we take $E(j,k) = \max_l(E^{(l)}(j,k))$, where $E^{(l)}(j,\cdot)$ denotes the l^{th} duplicated tank at position j.

For those variables which relate to tanks by absolute index rather than by treatment sequence, we write C_i for $C(S(i))$, F_i for $F(S(i), S(i+1))$, etc. For the other variables, $R_i \equiv R(i)$, etc. We will consider later only sequential treatments, that is when $T = N+1$ and $S(i) \equiv i$.

In an industrial setting, the time to perform a transfer operation is greater than the time to move the hoist between two tanks: we must raise the job, allow it to drip off above the tank, move, stabilise and lower. Without loss of generality, we include all this in the full times.

Our model differs from previous CLP approaches in that we optimise using the removal times. Previous authors used both entry and removal times (Rodošek and Wallace, 1998), entry and treatment times (Varnier et al., 1997) or treatment and removal times (Mak et al., 1998). A single variable per treatment yields a smaller search space and simplified constraints; the treatment times can be derived by:

$$T_i = R_i - (R_{i-1} + F_{i-1}) \tag{1}$$

Given a single-part cycle, we suppose the system is in steady state; thus exactly one job enters and one job leaves the system in every cycle. We seek values for the unknowns $R(i)$, $B(i)$, and P; the period will be integral since the data is integral.

2.2. Constraints

The constraints for a simple single-part cyclic HSP with time windows and tank capacities fall into four categories: the treatment sequence, the 1-part cycle, the tanks, and the hoists.

First, the cyclic structure and tank capacities are linked. If a tank is full to its capacity, the first job that entered (we assume first-in, first-out) must be removed before the arrival of the next job. Hence, considering tank i with tank capacity $C(i)$, a job must be removed before the arrival of the job which is $C(i)$ cycles behind it in the production sequence. We derive:

$$\begin{aligned} R_1 &< P \cdot C_1 \\ R_i &< P \cdot C_i + R_{i-1} + F_{i-1} \end{aligned} \tag{2}$$

In the case of simple capacities, the same holds with $C_i \equiv 1$.

The cyclic structure leads to a second constraint, since in every cycle all T operations must be performed once. It follows that the process time for a job cannot be greater than the product of the number of jobs and the period:

$$R_T + F_T \leq J \cdot P \tag{3}$$

Third, the treatment time for every tank must be bounded by the given data. Using (1), we have:

$$m_i \leq R_i - (R_{i-1} + F_{i-1}) \leq M_i \tag{4}$$

Finally, we have constraints on the hoists. Since each hoist can perform only one action at a time, we must avoid: removal of jobs from multiple tanks at once, removal of a job from a tank while the hoist is transporting another job, and the movement of the hoist faster or slower than the specified translation times.

Assuming a 1-part system, more subtly, a resource clash on a hoist will occur if we ask it to perform a task on a job at time τ, and a task is being performed on another job at time $\tau + P$. Here, the clash will be between the first task on one job and the second task on the *following* job. Generalising, we must rule out two tasks for any pair of times τ and $\tau + kP$, where $0 < k < J$.

It follows that at the centre of the HSP model is a three-variable disjunctive constraint: exactly one of

$$R_i + F_i + E_{i+1,j} \leq R_j + k \cdot P \tag{5}$$

or

$$R_j + F_j + E_{j+1,i} + k \cdot P \leq R_i \tag{6}$$

must hold, for all $k = 1, \ldots, J-1$; $i, j = 1, \ldots, T$; $j < i$.

2.3. Boolean Variables and Preprocessing

We use a boolean variable to denote, given any two tanks $i > j$, whether the hoist either goes first to i (when (5) applies) or to j (when (6) applies). For each pair of tanks $i > j$ and distance k between jobs in the production sequence, let $B_{i,j,k}$ be a boolean variable such that (5) holds iff $B_{i,j,k} = 1$. Then constraint (5) becomes (with (6) similarly):

$$R_i + F_i + E_{i+1,j} \leq R_j + kP + (1 - B_{i,j,k}) \cdot \Omega \tag{7}$$

where Ω is an integer that dominates the expression (a 'big-M' term).

Let us now assume sequential treatment. It is possible to eliminate certain contradictory situations in advance by a preprocessing step, initialising some of the $B_{i,j,k}$. Consider a job m in tank i and a job $m+k$, $1 \leq k < J$, in tank $j = i-1$. With unit tank capacities it is trivially not possible to move job $m+k$ before m; that is, the hoist must visit i before j, or equivalently, $B_{i,j,k} = 1$.

More generally, for $2 \leq i \leq T$ and $1 \leq j, k, m < T$,

$$(j = i - m \ \wedge \ k \geq m) \Rightarrow B_{i,j,k} = 1 \tag{8}$$

The reduction in boolean variables reduces the search space and the number of active constraints, and sharpens the relaxation bounds. With four-job PU13, for example, the reduction is from 234 to 98 variables.

2.4. Further HSP Classes

2.4.1. *Multiple Hoists, Multiple Tracks*

The same model applies for C/M/M as C/1/1 but with some constraints removed. We assume that each treatment tank $S(i)$ is assigned to one hoist $B(S(i))$ which will handle the tank for the i^{th} transfer. Since each hoist has a dedicated track, collisions are not possible. Hence, if two tanks are handled by different hoists, neither (5) nor (6) applies.

As before, we explicitly unfold the disjunction with an auxiliary boolean, in order to be able to perform some preprocessing. For each pair of tanks $i > j$, introduce $C_{i,j}$ such that the disjunction applies only if both tanks are handled by the same hoist:

$$B_i \neq B_j \Leftrightarrow C_{i,j} = 1 \tag{9}$$

Hence, (5) becomes:

$$R_i + F_i + E_{i+1,j} \leq R_j + kP + (1 - B_{i,j,k} + C_{i,j}) \cdot \Omega \tag{10}$$

Without loss of generality, we assign $B(1) = 1$ to remove symmetries. It is not possible to similarly set $R_1 = m_1$; this may prevent the optimal period from being found.

2.4.2. *Hoist Assignment*

Before we can present the model for C/M/1/x, we must discuss the problem of hoist assignment. Varnier et al. (1997) describe the three possibilities in the literature. In order of generality:

1. Disjoint zones (Lei and Wang, 1991). Each hoist is assigned a contiguous set (or *zone*) of tanks, and moves only within this set. Sets are disjoint between hoists except for boundary tanks.

2. Collision-based (Hanen and Munier, 1994; Lei and Wang, 1994; Manier, 1994). The sets for each hoist need not be disjoint, nor even contiguous. The hoists must be managed to avoid collisions.

3. Optimal (Kats and Levner, 1997; Levner et al., 1995). Rather than fixing the number of hoists then assigning tanks, find the minimal number of hoists that can achieve a given period, or the minimal period.

The third class, C/M/1/O, is a very difficult problem. It has been solved by non-CP methods only when other restrictions have been placed on the HSP (Kats and Levner, 1997; Lei et al., 1993), such as fixed processing times rather than time windows. With CP, we can find the number of hoists that minimises P, by labelling H upwards from the lowest value in its domain, until the period found with $H+1$ hoists is the same as that with H hoists.

Varnier et al. (1997) consider the second class, C/M/1/C, but with some restrictions. They assume, reasonably, that overlap tanks are accessed only by adjacent hoists, and, more restrictively, that each hoist has at least one tank that it alone accesses. Their solution uses heuristics and is not guaranteed to be optimal. Below, we make the first assumption but not the second, thus giving the optimal collision-based hoist assignment.

2.4.3. *Multiple Hoists, Single Track*
We assume again sequential treatment. First, the additions to the basic model for C/M/1/D. Every hoist has a set of tanks that it handles, and the intersection with the other zones is null except for boundary tanks. To achieve this, a system of constraints related with the $B(i)$ variables is added:

$$B(i-1)+1 \geq B(i) \geq B(i-1) \qquad (i=2,\ldots,T-1) \tag{11}$$

The domains of the $B(i)$ can be reduced by eliminating unreachable tanks (since hoists may not pass each other). Observe that the first tank can be reached by the first hoist alone, the second tank by the first and the second hoists only, etc. By symmetry, $B(1) \in \{1\}, B(2) \in \{1,2\}, \ldots, B(N-1) \in \{H-1,H\}, B(N) \in \{H\}$. Further, as in C/M/M, we need no constraint for those tanks handled by different hoists, but now the relation for the $C_{i,j}$ is linear: $C_{i,j} = B_i - B_j$.

Second, collision-based hoist assignment, C/M/1/C. We must consider three new possibilities:

– $B(i) = B(j)$: tanks i and j are handled by the same hoist; the disjunctive constraint remains unchanged.
– $B(i) > B(j)$: the disjunctive constraint is unnecessary, because tanks i and j are handled by hoists which will never meet.

– $B(i) < B(j)$: the disjunctive constraint must be modified to ensure the first hoist to move retires in time for the second.

Introducing additional boolean variables $D_{i,j}$, we obtain:

$$B_i > B_j \Leftrightarrow C_{i,j} = 1$$
$$B_i < B_j \Leftrightarrow D_{i,j} = 1 \tag{12}$$

and constraint (5) now becomes, for $i > j$ and $\delta_1 = E_{i+1,j-1} - E_{i+1,j}$:

$$R_i + F_i + E_{i+1,j} \leq R_j + kP + ((1 - B_{i,j,k}) + C_{ij}) \cdot \Omega - \delta_1 \cdot D_{i,j} \tag{13}$$

2.4.4. *Duplicated and Multi-function Tanks*

Duplicated tanks are often used in industry to remove a bottleneck on the line: for instance for a drying stage that takes 10 times longer than any other stage. We have instead considered higher capacity tanks.

Multi-function tanks are those that are visited more than once by the same job. From such a non-monotonic treatment sequence, it follows that jobs will have to be transported both right and left, which adds considerably to the complexity of the problem, particularly in hoist assignment. Combining load and unload stages also introduces bidirectional hoist movement; Mak et al. (1998) examine this case and show a small change in the constraint model they give is sufficient.

3. The Solvers

To find solutions to the constraint model given in the previous sections, we implemented solvers for the three main approaches to the HSP — pure CP, pure MIP and hybrid — using the Prolog-based ECL^{i}PS^{e} platform (IC–Parc, 2001). ECL^{i}PS^{e} implements a number of CLP schemes and provides an interface to internal and commercial solvers. For the constraint propagation, we used the `fd` finite domain solver built-in to ECL^{i}PS^{e}. For the MIP solving, we used the CPLEX package.

Initial bounds for the unknowns, the period P and removal times $R(i)$, are crucial to the MIP solver (constraint propagation will infer the bounds in the CP solver). For the period, the bounds are:

$$\left\lceil \frac{\sum_{k=1}^{T} m_k + \sum_{k=1}^{T} F_k}{N} \right\rceil \leq P \tag{14}$$

and

$$P \leq \sum_{k=1}^{T} (m_k + F_k) \tag{15}$$

The second inequality (15) is obtained by supposing we process one job at a time, but is not the tightest known; Levner et al. (1995) give an algorithm based on a merge-sort of interval sets. For PU13, for example, the bound is 933 rather than 1472 (Mak et al., 1998). However, since the CP search begins from the lower bound of the period, the upper bound is of little relevance to the CP solver.

For the removal times, the bounds are:

$$m(1) \leq R(i) \leq \sum_{k=1}^{T} M_k + \sum_{k=1}^{T-1} F_k \qquad (i = 1, \ldots, T) \tag{16}$$

The final sum is only to $T-1$ because the time to move the job from the last treatment tank to the unload station is irrelevant to the removal times.

3.1. CP Solver

The constraints, those described in Section 2, are considered only by the `fd` solver; propagation is performed over finite domains. Heuristics can be applied easily in CP to decisions such as the order of labelling of the variables and the order of value selection. We evaluated various possibilities before settling on that which performed best overall.

First we label the period, P, starting from the lowest value in its domain. If constraint propagation yields a consistent situation from the candidate value of P, we attempt to label the other variables. By labelling P from the lowest value, the first solution found will be the optimal solution; no separate proof of optimality is necessary.

After P, we label the assignment of hoists to transfer operations (in multi-hoist problems), then the removal times, $R(i)$, and finally the auxiliary booleans. This order gives maximum constraint propagation and search tree pruning. We used no in-search symmetry breaking.

For the problem C/M/1/D, we applied a redundant constraint: at most $T-H+1$ treatments can be handled by a single hoist (since each hoist must perform at least one treatment not to be redundant). This constraint furthers propagation a little, and is also applied to the CP component of the hybrid solver.

3.2. MIP Solver

The constraints are passed to the MIP solver via the `eplex` interface of ECL^{i}PS^e. Search is performed within the MIP package, by the default linear solving algorithm chosen by the CPLEX heuristics.

MIP performance is very dependent on the number of variables and on the bounds for the objective. The preprocessing steps of boolean variable reduction, symmetry removal and hoist assignment domain reduction give a marked improvement, examined in Section 4. The optimal period and whole solution is returned to ECL^{i}PS^e.

For the problem C/M/1/C, we performed a two-stage solution. The first solves the same problem but with partitioned hoist assignment. This is much simpler than collision-based assignment but — since the period for the latter, being less constrained, cannot be larger than the former — this first step provides a greatly improved upper bound on the period. With this upper bound, we then solve the full problem. This technique was also used for the hybrid solver.

3.3. Hybrid Solver

The constraints here are considered by both solvers, with search control handled by ECL^{i}PS^e. Information obtained by one solver is immediately available to the other via shared bounds and variable domains.

We first perform constraint propagation on finite domains, potentially yielding new upper and lower bounds on the variables. Second, the LP solver is invoked on the continuous relaxation (integer conditions omitted). When the relaxed problem is solved, control returns to ECL^{i}PS^e with the lower bound on P improved.

Third, we label the period P, as in Section 3.1. If a candidate value is acceptable after propagation, we attempt to label the other variables (hoist assignment and then removal times) by domain splitting. Should this succeed, we have the optimal solution; should it fail, we backtrack to label P further.

In the previous hybrid method (Rodošek and Wallace, 1998), the LP solver was invoked on the relaxed problem each time the bounds on P changed. Our experimentation showed that the information gained by subsequent invocations did not outweigh the cost overhead of the LP solver runs. While the simpler hybrid performs better overall, the argument can be made for a greater LP involvement in some cases.

Indeed, Rodošek and Wallace show that both finite domain and LP failures are necessary to prune the search space, and that there is a certain amount of orthogonality between the two methods. The hybrid algorithm gains from the LP solver global consistency of the continuous relaxation, an improved lower bound on the period, and (although we do not make use of them) suggested values for the other variables.

Most important to the hybrid is the quality of the bound provided by the relaxation. We discuss in Section 5 the formulation of the hybrid, co-operation between the solvers, and quality of the relaxed bound.

Table I. Hoist scheduling problems from Rodošek and Wallace

RW1	PU13 with four jobs
RW2	RW1 with tank capacity two
RW3	RW1 with two hoists and one track, collision based
RW4	RW1 with two hoists and two tracks

Table II. Results for the instance PU13

Parameters			Solution		Time (secs)		
hoists	tracks	capacity	jobs	period	cp	mip	hybrid
1	1	1	4	521	1.95	2.61	4.90
1	1	2	4	521	27.20	2.88	69.96
2	2	1	4	379	6.45	101.72	11.36
2	2	1	8	219	*	*	352.25
3	3	1	11	155	*	8550	311.91
4	4	1	11	151	7207	13.88	11.19
2	1	1	4	395	1192	7.65	89.64
2	1	1	7	251	*	620.83	3601
3	1	1	10	196	*	15910	*
4	1	1	11	152	9092	2612	645.76

4. Results and Analysis

4.1. Empirical Results

We give the results of the three solvers, pure CP, pure MIP and hybrid, for a range of data sets and problem parameters. The results were obtained on a 450MHz Pentium II with 384MB memory, using $\text{ECL}^{i}\text{PS}^{e}$ version 5.1. The times given below are in seconds to find the optimum and prove optimality. CPLEX version 7.0 was the LP solver used, with default settings and propagation.

The first results are those for variants of the PU13 problem. Table I shows the four benchmark problems considered in Rodošek and Wallace (1998), and in our results, given in Table II, these correspond to the lines with four jobs. The other lines in Table II give the number of jobs for which the minimal period is first achieved; * denotes more than 18,000 seconds (five hours) of CPU time.

Figure 1 shows our results for PU13 with four hoists and four tracks. An extended timeout of 40,000 seconds was used. The hardest instances

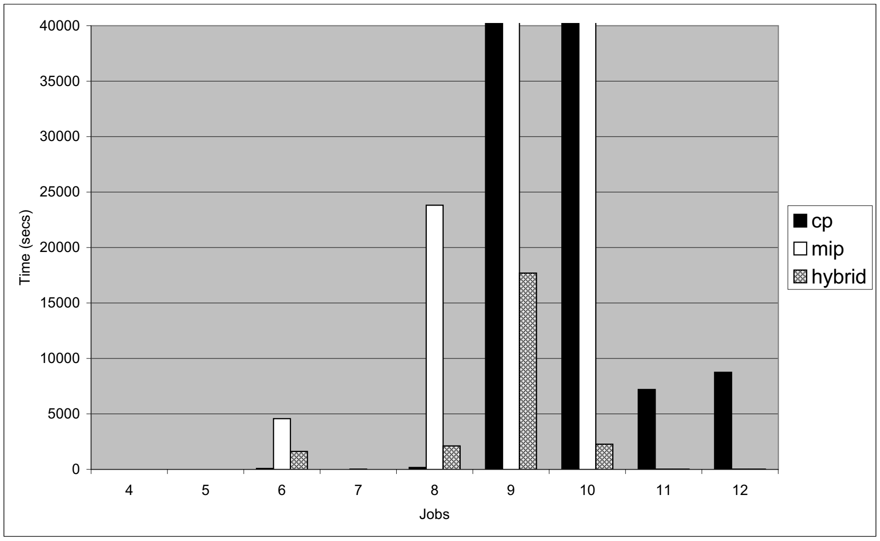

Figure 1. Solvers compared: PU13 with four hoists and four tracks

are with 9 and 10 jobs, when only hybrid finds a solution. It appears that 7 jobs is easier than 6 or 8, but the general trend is that the problem moves from easy to hard to easier again.

Our results for optimal period agree with previous work where the problems have been considered before. To find the number of jobs for which the period is minimal has not been previously done using a hybrid method, and to our knowledge not been done by any method for C/M/1/C. Three tracks, one hoist appears the hardest class of those we examined and, for large numbers of jobs, all of the solvers struggle.

In the question of hoist assignment, partitioning the line simplifies the situation greatly. We found that every instance of C/M/1/D could be consistently solved by MIP, for example, in less than one minute, and often in a few seconds with any solver. C/M/1/C is a much harder problem but gives correspondingly lower cycle times — for instance, with three hoists and ten jobs, 196 compared to 217.

We examined our results in light of those reported by Rodošek and Wallace. A closer consideration of their constraint model reveals a flaw: the following illegal possibility is permitted. A hoist carrying a job j descends to a tank containing some other job k, releases j and picks up k in the same instant. Clearly this is not possible unless there is some intermediate storage. This possibility cannot occur in our model because the inequality in (2) is strict.[5]

We produced benchmarks with the three solvers on other problems. Firstly, we used three data sets appearing in the literature: SZS5, LKS9

Table III. 100 randomly-generated instances of RW4

	literature	hybrid
minimal time	167	0.13
maximal time	1146	34.18
mean time	314	8.71

and DEGEM. None of these presented any hard problem instances for the HSP classes we consider. Were, for instance, the multi-part HSP to be considered, then these production lines with fewer tanks may become interesting. Details and timings obtained are given in the Appendix.

Secondly, we introduced a new problem, RYS16, with sixteen tanks, and tested the solver performance. The size of the new problem gave many hard instances. The results, broadly, are in accordance with those for PU13. For simpler instances, there is little to choose between the solvers. For harder instances, the hybrid is often at an advantage. For example, with four hoists and tracks, four jobs is solved quickly by all solvers but five jobs proves difficult for MIP. We do not give detailed timings here.

Thirdly, we used a randomly generated problem following Lei and Wang (1991). Minimal and maximal processing times are generated for each tank by $m_{rand}(i) = m(i) - 10 + 20r_1$ and $M_{rand}(i) = M(i) - 10 + 20r_2$, and the full travel times are $F_{rand}(i, i+1) = E(i, i+1) + 15 + 10r_3$, where r_1, r_2, r_3 are U$[0, 1]$ random variables.

Table III compares the results given in Rodošek and Wallace (1998) with our hybrid solver, for 100 random instances with two hoists, two tracks and four jobs. The times given are to prove optimality; the literature results were obtained on a Sun Sparc/20. The results indicate the robustness of the hybrid approach.

For the problems C/1/1 and C/M/M, our solver proved to be approximately 2–5 times faster than that of Rodošek and Wallace, supposing identical hardware. For instance, to prove optimality with four jobs, two hoists and two tracks took 11.4s compared to 289s, and four jobs, two hoists and one track took 90s compared to 2105s. This may be attributed to the simplified constraint model and the improved hybridization.

4.2. Analysis

There are two key factors in the HSP: the combinatorial complexity and the constrainedness. We consider an instance of the HSP to be *easy* if the number of jobs is at most twice the number of hoists; otherwise, we consider the instance to be *hard*.

For easy problems, the fastest solutions are found using pure CP, although the difference with the other solvers is not notable. Using `fd` only, most of the search tree is pruned. For hard problems, the fastest solutions are usually found using the hybrid solver: with occasional exceptions, the hybridization avoids the prohibitive CPU time the other solvers may require.

Multiple-capacity tanks are handled far better by MIP than CP or hybrid. However, we found no interesting instances among the data sets where the extra capacity improved the solution, most likely because the limiting resource in the long PU13 production line is not the tank capacities but the hoists.

Secondly, there is a distinction between *highly constrained* and *weakly constrained* problems, by which we mean constrainedness in the classical CP sense, not in terms of feasible set. C/M/1 is more constrained, for example, than C/M/M.

We observe that highly constrained problems are solved quickly by CP (if they are not hard problems). A large number of constraints ensures rapid `fd` propagation, which is the main element of the CP solver. In contrast, weakly constrained problems are solved more quickly by MIP, because its effort is dominated by the size of the branch-and-bound search tree, which depends on the number of integer variables rather than the number of constraints. The hybrid method, therefore, gives at least reasonable performance provided the orthogonality of the hybridization is effective.

5. Conclusion

In this work, we considered a revised hybrid approach to the hoist scheduling problem. The new model is more robust and scalable than the old, performing well across a range of problem classes and benchmark data. We used new preprocessing steps, in the case of sequential treatment, to reduce the search space, and we suggested a novel notation for the classes of the HSP.

5.1. Hybridization and the Hoist Scheduling Problem

In some instances of the HSP, the pure CP solver works well; in others, the pure MIP solver. Neither is able to perform consistently across the range of problems we examined. Our MIP solver provides better performance than our CP solver for weakly constrained hoist problems,

		Difficulty: Easy	Difficulty: Hard
Constrainedness	Weak	MIP	Hybrid/MIP
	Strong	CP	Hybrid

Figure 2. HSP solvers compared: conclusion

and overall, the results showed it handles more problem classes in reasonable time. Figure 2 summarises our comparison of the solvers. We have seen that CLP is easily adapted to the different HSP classes, with changes in the constraints only, and has a search procedure that is straight-forward to describe and easy to modify.

The performance of the hybrid approach can be attributed to two factors. First, the early detection of different failures by the component solvers (for which CP and linear solvers are orthogonal), and second, the complementary strength of LP in guiding search towards the global optimum and of CP in handling disjunctions. Thus our experience in the HSP is in line with the theory given by Hooker et al. (2000).

In the previous hybrid formulation, the linear solver was invoked at each node in the CP search. We found that, overall, the cost of these invocations outweighed their usefulness, and that a simpler hybrid with single LP invocation gave better performance.

In sequencing problems, to which HSP is related, it is common for boolean formulations to have weak relaxations. If it were possible to find simple cutting inequalities, some of the auxiliary boolean variables we use could be omitted — an approach that has worked in other applications of hybrid methods. The correspondingly smaller relaxation then could be solved at more nodes in the search for both hybrid and MIP approaches.[6]

We have seen that hybridization retains the modelling advantage of CP while leading to a more robust solver. For larger problems, in particular, the hybrid solver can often find solutions and prove optimality when neither our CP or MIP solvers can do so. In almost every case, the hybrid solver is better than the previous hybrid approach, and our results overall are competitive with the best of those obtained so far by any approach in the literature.

5.2. Further Work

Three areas appear promising for future work on hybrid HSP. First, improving the existing hybrid solver. It may be appropriate to invoke the LP solver subsequently once `fd` has progressed by some degree. It is the case that more symmetries could be removed than at present.

In the class C/M/1, the problem can be decomposed into two parts: hoist-tank assignment as a master problem and hoist scheduling as a subproblem. When the subproblems are independent, a tighter form of hybridization than we have used, notably Benders decomposition (Eremin and Wallace, 2001), may give better results; this independence is so for C/M/1/D but not for collision-based assignment.

Our combined use of explicit and automated constraint linearization, choice of problem variables and presolving of the partitioned problem have together yielded an efficient MIP model. Even so, there is room for improvement. For instance, using the better upper bound of Levner et al. (1995) for C/M/1/D should yield the solution more quickly (and would give some aid to the hybrid, too). In addition, the automated translator of Rodošek et al. (1997) has been recently extended (Ottosson and Thorsteinsson, 2000); this may allow us to exploit more of the modelling features of CLP and to tighten the linear relaxation.

Second, the model given in Section 2 holds for multi-function tanks and non-sequential treatment, although as noted, some of the preprocessing steps do not (it would be necessary to relax expressions in which $S(i)$ occurs as a subscript). Constraint-based solutions to the HSP with these extensions have already been demonstrated (Mak et al., 1998; Varnier et al., 1997), but a hybrid solver not yet applied.

Other possible extensions include a single load/unload stage (which would be straight-forward), multi-part cycles (challenging, but permits better solutions) and C/M/1/O optimal hoist allocation (very challenging, perhaps heuristics would be necessary).

Third, there appears to have been no work to date on the HSP with H hoists and R tracks where $1 < R < H$ and $H \geq 3$. This increases the complexity, with the new problem of how to assign hoists to tracks as well as to transfer operations. To refine and apply the CLP–MIP hybrid to such problems seems a natural step.

Appendix

As discussed in Section 1.2, PU13 is a variant of the classic benchmark problem PU12 in which the tanks are thought of as arranged in a circle. Specifically, an extra tank is added at the front of the line, with minimal treatment time 120 and maximal treatment time unbounded. Jobs are automatically entered into this first tank, and travel times from it to the others are as from the load stage to the tanks in PU12.

We considered three other previously known data sets:

Table IV. Results for the real-world problem DEGEM

Parameters			Solution		Time (secs)		
hoists	tracks	capacity	jobs	period	cp	mip	hybrid
1	1	1	1	693	0.01	0.01	0.01
1	1	1	2	347	0.02	0.03	0.06
1	1	1	3	250	0.05	0.05	0.09
1	1	1	4	250	0.08	0.07	0.12
2	2	1	2	347	0.04	0.19	0.15
2	2	1	3	250	0.07	0.23	0.19
2	2	1	4	250	0.15	0.30	0.23
3	3	1	3	250	0.07	0.22	0.19
3	3	1	4	250	0.15	0.18	0.22
2	1	1	2	347	0.22	0.03	0.06
2	1	1	3	250	0.04	0.05	0.09
2	1	1	4	250	0.08	0.07	0.12
3	1	1	3	250	0.05	0.05	0.09
3	1	1	4	250	0.08	0.07	0.12

– SZS5: five tank test problem from Song et al. (1993)
– LKS9: nine tank problem from Levner et al. (1996)
– DEGEM: seven tank real-world process from Kats et al. (1999)

We did not use the ten tank benchmark problem 'SSZ' quoted in Kats et al. (1999) because it contained non-integer travel times. As discussed in Section 4, none of these data sets were challenging. For instance, Table IV gives the results for the problem DEGEM.

RYS16 is a sixteen tank problem of our construction, based loosely on PU13. The data for the problem is as follows, with the empty travel times given in Table V:

$$m(i) = [0, 120, 150, 100, 120, 90, 200, 25, 60, 0, 60, 45, 130, 120, 90, 30, 30]$$
$$M(i) = [0, \infty, 200, 120, 195, 125, 200, 40, 120, \infty, 120, 75, \infty, \infty, 120, 60, 60]$$
$$F(i, i+1) = [0, 31, 22, 22, 20, 25, 10, 23, 22, 50, 22, 22, 46, 27, 22, 30, 30]$$

Acknowledgements

We are grateful to Mark Wallace for many wise insights, and thank Andrew Eremin and the anonymous reviewers for their suggestions. The first author is supported by CICYT grant TAP98-0364.

Table V. Empty travel times in RYS16

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	11	14	16	14	19	10	24	26	29	6	8	10	11	4	5	5	0
1	11	0	2	5	2	8	12	13	15	17	10	3	1	11	6	4	7	6
2	14	2	0	2	0	5	8	10	13	15	12	6	3	14	8	5	9	10
3	16	5	2	0	2	3	5	8	10	13	15	8	6	16	10	6	7	3
4	14	2	0	2	0	5	9	10	13	15	12	6	3	14	12	7	9	14
5	19	8	5	3	5	0	3	5	7	10	18	11	9	19	14	8	11	10
6	10	12	8	5	9	3	0	2	5	7	20	14	11	22	16	16	13	10
7	24	13	10	8	10	5	2	0	2	5	23	16	14	24	18	17	11	9
8	26	15	13	10	13	7	5	2	0	2	25	19	16	26	20	18	9	8
9	29	17	15	13	15	10	7	5	2	0	0	21	19	29	22	19	11	7
10	6	10	12	15	12	18	20	23	25	0	0	7	9	6	24	20	13	7
11	8	3	6	8	6	11	14	16	19	21	7	0	2	8	26	26	15	11
12	10	1	3	6	3	9	11	14	16	19	9	2	0	10	28	27	13	20
13	11	11	14	16	14	19	22	24	26	29	6	8	10	0	26	28	12	14
14	4	6	8	10	12	14	16	18	20	22	24	26	28	26	0	28	12	14
15	5	4	5	6	7	8	16	17	18	19	20	26	27	28	28	0	12	14
16	5	7	9	7	9	11	13	11	9	11	13	15	13	12	12	12	0	14
17	0	6	10	3	14	10	10	9	8	7	7	11	20	14	14	14	14	0

Notes

[1] In the case of multiple hoists on a single track.

[2] In fact, the solution they published is not optimal. Phillips and Unger found the optimum for three simultaneous components (jobs), but a solution of lower period exists with four jobs.

[3] It is permissible for this value to be unbounded.

[4] It may be that the two physically coincide, that is, a single load/unload stage.

[5] A further consequence of this is that the time for the hoist to return to the load stage, having deposited a finished job at the unload stage, cannot be zero. For PU13, which would otherwise permit this possibility, the period could therefore be one second shorter in certain cases, namely 11 jobs with 3 or 4 hoists and tracks.

[6] We are grateful to an anonymous reviewer for this observation.

References

Armstrong, R., L. Lei, and S. Gu: 1992, 'A Bounding Scheme for Deriving the Minimal Cycle Time of a Single-Transporter N-Stage Process with Time-Window Constraints'. GSM Working Paper 92-07, Rugters University.

Baptiste, P., Y. Caseau, T. Kökény, C. Le Pape, and R. Rodošek: 1998, 'Creating and Evaluating Hybrid Algorithms for Inventory Management Problems'. *Journées Nationales sur la Résolution Pratique de Problèmes NP-Complets*.

Baptiste, P., B. Legeard, M. Manier, and C. Varnier: 1994, 'A Scheduling Problem Optimisation Solved with Constraint Logic Programming'. In: *Proc. of the Intl. Conf. on Parallel Architectures and Compilation Techniques*. pp. 47–66.

Baptiste, P., B. Legeard, and C. Varnier: 1992, 'Hoist Scheduling Problem: An Approach Based on Constraints Logic Programming'. In: *Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation (ICRA'92)*. pp. 1139–1144.

Chen, C., C. Chu, and J.-M. Proth: 1998, 'Cyclic Scheduling of a Hoist with Time Window Constraints'. *IEEE Trans. on Robotic Automation* **14**, 144–152.

Cheng, C. and S. Smith: 1996, 'A Constraint Satisfaction Approach to Makespan Scheduling'. In: *Proc. of the 4th Intl. Conf. on Artificial Intelligence Planning Systems (AIPS'96)*. pp. 45–54.

Denat, J.-P., S. Collart Dutilleul, and S. Marteau: 2000, 'A Static Robust Control of an Electroplating Line with a Periodic Output Objective'. In: *2nd Intl. Conf. on Management and Control of Production and Logistics (MCPL'00)*.

Eremin, A. and M. Wallace: 2001, 'Hybrid Benders Decomposition Algorithms in Constraint Logic Programming'. In: *Proc. of the 7th Intl. Conf. on Principles and Practices of Constraint Programming (CP'01)*. Cyprus, pp. 1–15.

Hall, N., H. Kamoun, and C. Sriskandarajah: 1997, 'Scheduling in Robotic Cells: Classification, Two and Three Machine Cells'. *Operational Research* **45**, 421–439.

Hanen, C.: 1994, 'Study of a NP-Hard Cyclic Scheduling Problem: The Recurrent Job-Shop'. *Euro. J. of Operations Research* **72**, 82–101.

Hanen, C. and A. Munier: 1994, 'Pilotage periodique d'une ligne de galvanoplastie à plusieurs robots'. Research report, LITP.

Hooker, J., G. Ottosson, E. Thorsteinsson, and H.-J. Kim: 2000, 'A Scheme for Unifying Optimization and Constraint Satisfaction Methods'. *Knowledge Engineering Review, Special issue on AI/OR* **15**(1), 11–30.

IC–Parc: 2001, 'ECL^{i}PS^e User Manual Version 5.1'. `www.icparc.ic.ac.uk`.

Kats, V. and E. Levner: 1997, 'Minimizing the Number of Robots to Meet a Given Cyclic Structure'. *Annals of Operations Research* **69**, 209–226.

Kats, V., E. Levner, and L. Meyzin: 1999, 'Multiple-Part Cyclic Hoist Scheduling Using a Sieve Method'. *IEEE Trans. on Robotics and Automation* **15**(4), 704–713.

Lam, K.: 1997, 'A Heuristic Method for Multiple Hoist Scheduling Problems by using Simulated Annealing and Local Search'. Working Paper, MS Dept., City University of Hong Kong.

Lamothe, J., M. Correge, and J. Delmas: 1994, 'Hoist Scheduling Problem in a Real Time Context'. In: *11ième conférence internationale sur l'analyse et l'optimisation des systèmes*.

Lei, L., R. Armstrong, and S. Gu: 1993, 'Minimizing the Fleet Size with Dependent Time-Window and Single-Track Constraints'. *Operations Research Letters* **14**, 91–98.

Lei, L. and T. Wang: 1991, 'The Minimum Common-Cycle Algorithm for Cycle Scheduling of Two Material Handling Hoists with Time Window Constraints'. *Management Science* **37**(12), 1629–1639.

Lei, L. and T. Wang: 1994, 'Determining Optimal Cyclic Hoist Schedules in a Single Hoist Electroplating Line'. *IEE Transactions* **26**(2), 25–33.

Leung, J., X. Yang, R. Mak, and K. Lam: 1997, 'Hoist Scheduling for a Circuit Board Manufacturing Line — A Mixed Integer Programming Approach'. In: *16th Intl. Symp. on Mathematical Programming*.

Levner, E., V. Kats, and V. Levit: 1995, 'An Improved Algorithm for a Cyclic Robot Scheduling Problem'. In: E. Levner (ed.): *Intelligent Scheduling of Robots and Flexible Manufacturing Systems*. CTEH, pp. 129–141.

Levner, E., V. Kats, and C. Sriskandarajah: 1996, 'A Geometric Algorithm for Finding Two-Unit Cyclic Schedules in a No-Wait Robotic Flowshop'. In: E. Levner (ed.): *Intelligent Scheduling of Robots and Flexible Manufacturing Systems II.* CTEH, pp. 101–112.

Lim, J.-M.: 1997, 'A Genetic Algorithm for a Single Hoist Scheduling in the Printed Circuit Board Electroplating Line'. *Computers in Ind. Eng.* **33**(3–4), 789–792.

Mak, R., K. Lam, and S. M. Gupta: 1998, 'A Practical Algorithm for Cyclic Hoist Scheduling in a PCB Manufacturing Facility'. *J. of Electronics Manufacturing* **8**(3/4), 193–207.

Manier, M.: 1994, 'Contribution à l'ordonnanement cyclique du système de manutention d'une ligne de galvanosplastie'. Ph.D. thesis, Franche-Comté University.

Manier, M., C. Varnier, and P. Baptiste: 2000, 'Constraint Based Model for Cyclic Multi-Hoist Scheduling Problem'. *Production Planning and Control* **11**(3), 244–257.

Ng, W. and J. Leung: 1997, 'Determining the Optimal Move Times for a Given Cyclic Schedule of a Material Handling Hoist'. *Computers in Ind. Eng.* **32**(3), 595–606.

Ottosson, G. and E. Thorsteinsson: 2000, 'Linear Relaxations and Reduced-Cost Based Propagation of Continuous Variable Subscripts'. In: *Proc. of the 2nd Intl. Workshop CP-AI-OR'00.* pp. 129–138.

Phillips, L. and P. Unger: 1976, 'Mathematical Programming Solution of a Hoist Scheduling Problem'. *AIIE Transactions* **8**(2), 219–255.

Rodošek, R. and M. Wallace: 1998, 'A Generic Model and Hybrid Algorithm for Hoist Scheduling Problems'. In: *Proc. of the 4th Intl. Conf. on Principles and Practice of Constraint Programming (CP'98).* pp. 385–399.

Rodošek, R., M. Wallace, and M. Hajian: 1997, 'A New Approach to Integrating Mixed Integer Programming with Constraint Logic Programming'. *Annals of Operational Research* **86**, 63–87.

Shapiro, G. and H. Nuttle: 1988, 'Hoist Scheduling for a PCB Electroplating Facility'. *IIE Transactions* **20**(2), 157–167.

Song, W., Z.-B. Zabinsky, and R.-L. Storch: 1993, 'An Algorithm for Scheduling a Chemical Processing Tank Time'. *Production Planning and Control* **4**(4), 323–332.

Tsang, E., C. Voudouris, J. Ford, and P. Mills: 1999, 'Operations Research Meets Constraint Programming: Some Achievements So Far'. In: *Proc. of the 15th National Conference of the Australian Society for Operations Research (ASOR'99).* pp. 872–883.

Varnier, C., A. Bachelu, and P. Baptiste: 1997, 'Resolution of the Cyclic Multi-Hoists Scheduling Problem with Overlapping Partitions'. *INFOR, Special Issue on Intelligent Scheduling of Robots* **35**(4), 309–324.

Yih, Y., T. Liang, and H. Moskowitz: 1993, 'Robot Scheduling in a Circuit Board Production Line, an OR/ANN Approach'. *IIE Transactions* **25**(2), 26–33.