Qualitative Constraints for Job Shop Scheduling

Jean-François Condotta[1], **Gérard Ligozat**[2] and **Mahmoud Saade**[1]

[1] CRIL-CNRS, University of Artois
62307 Lens Cedex, France
{condotta,saade}@cril.univ-artois.fr
[2] LIMSI-CNRS, University of Paris-Sud, 91403 Orsay, France
ligozat@limsi.fr

Abstract

This paper introduces a translation of the *job shop scheduling problem* into a qualitative constraint satisfaction problem using *INDU* and Allen relations. We show that the translation is sound and complete. We also use the notion of frozen constraints and show that it allows the user to consider only partial solutions when searching for solutions. Our work constitutes a new approach to the problem of constructing content-motivated benchworks for qualitative calculi.

Introduction

Many qualitative formalisms have been proposed and studied during the last two decades. Most of the research activity in this domain, however, has been devoted to studying the formal properties of those formalisms. A prominent aspect has been the search for tractable classes which has motivated many studies (Nebel and Bürckert 1995; Ligozat 1996), starting with the *ad hoc* study of specific formalisms, followed nowadays by the appearance of generic methods (Renz 2007). More recently, a coordinated effort has been initiated in order to develop generic software tools for solving qualitative constraint networks problems (Dylla et al. 2006; Wallgrün et al. 2006; Gantner, Westphal, and Wölfl 2008; Condotta, Ligozat, and Saade 2006b; 2006a). These tools try to integrate the theoretical results obtained during the preceding years.

Contrary to the situation for discrete constraint satisfaction problems (discrete CSPs), there are few if any accepted benchmarks for qualitative constraint networks, whether derived from real world problems or from academic sources. Most of the experiments on qualitative networks use randomly generated networks with no externally motivated structure (Ladkin and Reinefeld 1992; Bessière 1996; Van Beek and Manchak 1996; Nebel 1996).

The main contribution of this paper consists in proposing a new approach to the constitution of benchmarks for qualitative constraint networks based on translations of instances of the *job shop scheduling problem* (JSSP). Although we only consider one particular type of job scheduling problem, our translation method should be extensible to other types of the problem.

We show that for the networks obtained by this translation process, only partial solutions of the network are of relevance. Hence the notion of *frozen constraints* introduced in (Condotta, Ligozat, and Saade 2007) can be used in this context. We then prove that a property of local consistency (namely, weak path-consistency) on part of the network (namely, the non-frozen part) is a sufficient condition for detecting the consistency of the full network and, if consistency indeed holds, for obtaining a solution to the original job shop problem.

The structure of the paper is as follows. In the next section, we define the type of job shop scheduling problem we will be considering. Then, we recall some of the basics of qualitative networks and Allen's and INDU formalism. We define the translation from the JSSP to qualitative networks. We close the paper with a section devoted to the frozen constraints and a conclusion.

The *job shop scheduling problem*

Many versions of the *job shop scheduling problem* have appeared in the literature. Here we consider the version proposed by Sadeh (Sadeh 1991) for constituting a benchmark for discrete CSPs. Note that this benchmark is part of the benchmarks used in international competitions organized by the community of discrete CSPs (Van Dongen and Lecoutre 2006).

The *job shop scheduling* problem consists in assigning a set of n_{job} jobs $\{job_0, \ldots, job_{n_{job}-1}\}$ to a set of n_{res} physical resources $\{res_0, \ldots, res_{n_{res}-1}\}$, where $n_{job} > 0$ and $n_{res} > 0$.

Each job job_i consists in an ordered set of n_i operations $\{O_{i,0}, \ldots, O_{i,n_i-1}\}$, where $n_i > 0$. A job job_i has a starting time td_i and a realization time tr_i which delimit an interval during which all operations of the job have to be realized. Each operation $O_{i,j}$ has a duration $d_{i,j} > 0$. Finally, each operation has to be realized on a resource $res_{i,j} \in \{res_0, \ldots, res_{n_{res}-1}\}$, where $n_{res} > 0$.

Solving a JSSP consists in finding a schedule for its operations such that some temporal constraints are met. More specifically, a solution to the problem is a map st which associates to each operation $O_{i,j}$ an integer number $st_{i,j}$ (the starting instant of the operation) in such a way that the following conditions are met:

- $st_{i,j} + d_{i,j} \leq st_{i,j+1}$ for all $i \in \{0, \ldots, n_{job}-1\}$ and for all $j \in \{0, \ldots, n_i - 1\}$;
- either $st_{i,j} + d_{i,j} \leq st_{i',j'}$ or $st_{i',j'} + d_{i',j'} \leq st_{i,j}$ for all $i, i' \in \{0, \ldots, n_{job} - 1\}$ and for all $j \in \{0, \ldots, n_i - 1\}$ and $j' \in \{0, \ldots, n_{i'} - 1\}$, where $(i,j) \neq (i',j')$ and $res_{i,j} = res_{i',j'}$;
- $st_{i,0} \geq td_i$ and $st_{i,n_i-1} + d_{(i,n_i-1)} \leq tr_i$ for all $i \in \{0, \ldots, n_{job} - 1\}$.

More complex models of the *job shop scheduling problem* could be considered, including versions where the ordering of the operations is tree-like rather than linear. However, the model we use in this paper will suffice to give a full and precise picture of the translation techniques into qualitative constraint problems. Moreover, some existing benchmarks use precisely the model we consider here.

Allen's calculus and the *INDU* calculus

A (binary) qualitative formalism is based on a finite set $\mathcal{B} = \{b_1, \ldots, b_k\}$ of k binary relations defined on a domain D. These relations are called basic relations. They represent specific relative positions between the spatial or temporal objects represented by the elements of the domain D .

Allen's calculus (Allen 1981) is one of the best known among qualitative calculi. It is based on 13 binary relations between objects which are usually interpreted as intervals in the real line. Each basic relation corresponds to a specific configuration of the endpoints of those intervals: if the ending point of the first interval coincides with the starting point of the second, we get the $meets$ relation; if this starting point is located inside the second interval, the $overlaps$ relation holds, and so on. The full set of basic relations contains the relations $\{meets, met-by, before, after, equals, overlaps, overlapped-by, finishes, finished-by, starts, started-by, during, contains\}$.

The *INDU* calculus introduced by Pujari *et al.* (Pujari, Kumari, and Sattar 1999; 1999) is a refinement of Allen's calculus. Namely, besides the relative position of two intervals, the calculus also encodes the relative durations of those intervals: the first interval can have a shorter or a longer duration, or both can have the same duration. Accordingly, some among the basic relations of Allen's calculus split into sub-relations which are considered as basic relations in the *INDU* calculus. For instance, the relation $after$ splits into three basic relations of *INDU*: $after^=$, $after^<$ and $after^>$.

Qualitative constraint networks are used to represent some information about spatial or temporal entities. A qualitative constraint network $\mathcal{N}$ is a pair (V, C) where V is a set of variables (standing for the entities) and C a map which to each pair of variables (V_i, V_j) in V associates a subset of basic relations in $\mathcal{B}$ (which stand for the admissible relations between the entities represented by V_i and V_j). In Allen's case, for example, $C(V_i, V_j)$ could consist of the set $\{meets, after\}$, which stipulates that the interval V_i either meets or follows interval V_j. A solution of a qualitative constraint network $\mathcal{N} = (V, C)$ is a map sol which associates to each variable $V_i \in V$ a value s_i in D in such a way that for each pair $(V_i, V_j) \in V \times V$, $(s_i, s_j) \in b$ for some b in $C(V_i, V_j)$.

Translating JSSPs into qualitative constraint networks

Consider an instance JSS of the JSSP as defined in the second section. We will now define a qualitative constraint network for the *INDU* calculus, denoted by INDU(JSS), such that for each solution of JSS there is a solution of INDU(JSS) and conversely, to each solution of INDU(JSS) there will correspond a solution of JSS.

Let INDU(JSS) $= (V, C)$ be the *INDU* network defined as follows:
Time is modelled using t_{max} consecutive *unit* intervals of equal length represented by variables $T_0, T_1, \ldots, T_{t_{max}-1}$. Here t_{max} is greater than $\mathrm{Max}_{i \in \{0,\ldots,n_{job}-1\}}(tr_i)$. The starting point of T_i models the discrete instant i, its ending point the discrete instant $i+1$. The following constraints are enforced on these variables:

$$C(T_i, T_{i+1}) = \{meets^=\} \text{ for all } i \in \{0, \ldots, t_{max} - 2\}.$$

To the j^{th} operation $O_{i,j}$ of the i^{th} job is associated a variable $Op_{i,j}$ which represents the time interval during which this operation is active, as well as a set of $d_{i,j}$ variables $Op^0_{i,j}, \ldots, Op^{d_{i,j}-1}_{i,j}$ which represent the consecutive unit intervals contained in $Op_{i,j}$. Those intervals are used to constrain the duration of $Op_{i,j}$ and each one of them has to equal one of the intervals representing time. The following constraints are imposed on these intervals:

- $C(Op_{i,j}, Op^0_{i,j}) = \{started\_by^>, equals^=\}$;
- $C(Op_{i,j}, Op^{d_{i,j}-1}_{i,j}) = \{finished\_by^>, equals^=\}$;
- $C(Op^k_{i,j}, Op^{k+1}_{i,j}) = \{meets^=\}$, for all $k \in \{0, \ldots, d_{i,j} - 2\}$.

These intervals also have to be correctly positioned with respect to those which represent time. Hence we impose the following constraints:

- $C(Op^0_{i,0}, T_{td_i}) = \{equals^=, met\_by^=, after^=\}$;
- $C(Op^{d_{i,(n_i-1)}-1}_{i,n_i-1}, T_{tr_i-1}) = \{equals^=, meets^=, before^=\}$,
- $C(Op^k_{i,j}, T_l) = \{equals^=, meets^=, met\_by^=, before^=, after^=\}$ for all $l \in \{0, \ldots, T_{t_{max}-1}\}$ when $(j,k,l) \neq (0, 0, td_i)$ and $(j,k,l) \neq (n_i - 1, d_{i,(n_i-1)} - 1, tr_i - 1)$.

The operations of each job have to be realized sequentially. Hence we assert the following:

- $C(Op_{i,j}, Op_{i,j+1}) = \{before^<, before^=, before^>, meets^<, meets^=, meets^>\}$, for all $i \in \{0, \ldots, n_{job} - 1\}$ and for all $j \in \{0, \ldots, n_i - 2\}$.

As for constraints between operations belonging to different jobs, we need constraints of exclusion expressing the fact that two operations sharing the same resource have to be realized on non-overlapping intervals:

- $C(Op_{i,j}, Op_{i',j'}) = \{before^<, before^=, before^>, meets^<, meets^=, meets^>, after^<, after^=, after^>, met\_by^<, met\_by^=, met\_by^>\}$, for all $i, i' \in \{0, \ldots, n_{job} - 1\}$ and for all $j \in \{0, \ldots, n_i - 1\}$ and $j' \in \{0, \ldots, n_{i'} - 1\}$, with $(i, j) \neq (i', j')$ and $res_{i,j} = res_{i',j'}$.

Finally, if a constraint $C(V_i, V_j)$ has been defined between V_i and V_j, then the inverse of $C(V_i, V_j)$ is enforced between V_j and V_i. For all remaining pairs, the universal relation, corresponding to the set of all basic relations, holds.

Theorem 1 *The network* INDU(JSS) *is consistent if and only if the JSSP* JSS *has a solution.*

Proof.

- Let st be a solution of JSS. Using st, we build a solution sol of INDU(JSS) in the following way: the interval $sol(T_i)$ is defined as $[i, i+1]$ for all $i \in \{0, \ldots, t_{max}-1\}$. We define $sol(Op_{i,j})$as $[st_{i,j}, st_{i,j} + d_{i,j}]$ for all $i \in \{0, \ldots, n_{job}-1\}$ and for all $j \in \{0, \ldots, n_i-1\}$. Finally, each interval $sol(Opi, j^k)$ is defined as $[st_{i,j} + k, st_{i,j} + k + 1]$ for all $i \in \{0, \ldots, n_{job} - 1\}$, $j \in \{0, \ldots, n_i - 1\}$ and $k \in \{0, \ldots, d_{i,j} - 3\}$. If we examine the instantiation sol obtained in this way, we can easily check that it is a solution of INDU(JSS).
- Let now sol be a solution of INDU(JSS). Define st as the function that associates to each operation $O_{i,j}$ the integer $st_{i,j}$ defined by $sol(T_{st_{i,j}}) = sol(Op^0_{i,j})$. We can easily check that such an integer number exists and is uniquely defined considering the constraints we have defined. Moreover, an examination of the constraints shows that st is a solution of JSS.

$\dashv$

An examination of the constraint network INDU(JSS) and of the solutions we have just described in the above proof shows that the fact that the intervals used for modelling time have a uniform duration plays no important role. Hence we can construct a network in a similar fashion as before while forgetting about the relative duration part, and obtain in this way an Allen network Allen(JSS). For instance, we will have $C(Op_{i,j}, Op_{i,j+1})$ as $\{before, meets\}$ instead of $\{before^<, before^=, before^>, meets^<, meets^=, meets^>\}$, for all $i \in \{0, \ldots, n_{job} - 1\}$ and for all $j \in \{0, \ldots, n_i - 2\}$.

Frozen constraints

We now use the notion of frozen constraints introduced in (Condotta, Ligozat, and Saade 2007). Essentially, a frozen constraint is a constraint whose relation remains fixed and cannot be changed while looking for a solution or during local constraint propagation. Freezing constraints is of practical interest, as it allows to use efficient constraint propagation and search techniques, as explained in (Condotta, Ligozat, and Saade 2007). Those constraints which have been frozen by the user are constraints which do not have to be refined when looking for a solution. For example, if we consider the construction of INDU(JSS), obviously the instantiations of the constraints between variables representing the operations and time as basic relations is relevant for the search of a solution to the JSSP. Hence these constraints cannot be frozen. We will later determine, using a particular property, what constraints can be frozen when looking for a solution to the JSSP.

In what follows, we assume given a qualitative constraint network $\mathcal{N} = (V, C)$ and a set $SFrozen \subseteq V \times V$ of frozen constraints. We assume that, if $(V_i, V_j) \in SFrozen$, then $(V_j, V_i) \in SFrozen$.

Definition 1 *A $SFrozen$-scenario of $\mathcal{N} = (V, C)$ is a subnetwork (V, C') of $\mathcal{N}$ such that:*

- $C'_{ij} = C_{ij}$ *if* $(V_i, V_j) \in SFrozen$, *and* $C'_{ij} = \{A\}$, *where* $A \in C_{ij}$ *otherwise.*

One of the interests of the use of the notion of frozen scenarios is that in some cases it is possible to isolate "critical" constraints which can result in the inconsistency of the system, in particular using local consistency properties. Such a property was introduced in (Condotta, Ligozat, and Saade 2007). It is a property of weak closure under weak composition defined as follows:

Definition 2 *Let $\mathcal{N} = (V, C)$ be a network, and $SFrozen \subseteq V \times V$ a set of frozen variables. $\mathcal{N}$ is said to be $(SFrozen, \circ)$-closed if for each pair of variables $V_i, V_j \in V$ such that $(V_i, V_j) \notin SFrozen$, $C(V_i, V_j) \subseteq C(V_i, V_k) \circ C(V_k, V_j)$ for all $V_k \in V$.*

If we now consider our translation of the JSSP, and define the set $SFrozenJSS$ as $SFrozenJSS = \{C(T_i, T_j) : i, j \in \{0, \ldots, t_{max} - 1\}\} \cup \{C(Op^k_{i,j}, Op^{k'}_{i',j'}) : i, i' \in \{0, \ldots, n_{job} - 1\}, j \in \{0, \ldots, n_i - 1\}, j' \in \{0, \ldots, n_{i'} - 1\}, k \in \{0, \ldots, d_{i,j} - 1\}$ and $k' \in \{0, \ldots, d_{i',j'} - 1\}\}$ $\cup \{C(Op^k_{i,j}, Op_{i',j'}) : i, i' \in \{0, \ldots, n_{job} - 1\}, j \in \{0, \ldots, n_i - 1\}, j' \in \{0, \ldots, n_{i'} - 1\}, k \in \{0, \ldots, d_{i,j} - 1\}\}$ $\cup \{C(Op_{i,j}, Op^{k'}_{i',j'}) : i, i' \in \{0, \ldots, n_{job} - 1\}, j \in \{0, \ldots, n_i - 1\}, j' \in \{0, \ldots, n_{i'} - 1\}, k' \in \{0, \ldots, d_{i',j'} - 1\}\}$ then we can prove the following property:

Proposition 1 *Any $SFrozenJSS$-scenario of* INDU(JSS) *which is $(SFrozenJSS, \circ)$-closed is consistent.*

Conclusion

We have introduced a method for translating the job shop scheduling problem (JSSP) in terms of qualitative constraint networks for the $INDU$ and Allen calculi. This method can be generalized to other variants of the JSSP. We have also shown that the notion of frozen constraints can be used for the networks obtained in this way, and we have given a characterization of a set of constraints which can be frozen and for which a local consistency property is a sufficient condition for consistency. We are now running experiments using the software platform QAT in order to validate our approach, aiming to get state-of-the-art efficiency. Using existing benchmarks for JSSPs, our work opens a perspective for constructing content-motivated benchmarks for qualitative networks.

References

Allen, J. F. 1981. An interval-based representation of temporal knowledge. In *Proc. of the Seventh Int. Joint Conf. on Artificial Intelligence (IJCAI'81)*, 221–226.

Bessière, C. 1996. A Simple Way to Improve Path Consistency Processing in Interval Algebra Networks. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, volume 1, 375–380.

Condotta, J.-F.; Ligozat, G.; and Saade, M. 2006a. QAT : une boîte à outils dédiée aux algèbres qualitatives. In *Semaine de la Connaissance (SdC2006)*, volume 4, 149–155.

Condotta, J.-F.; Ligozat, G.; and Saade, M. 2006b. The QAT: A Qualitative Algebra Toolkit. In *Proceedings of the second IEEE International Conference on Information Technologies: from Theory to Applications (ICTTA'06), Damascus, Syria*.

Condotta, J.-F.; Ligozat, G.; and Saade, M. 2007. Eligible and frozen constraints for solving temporal qualitative constraint networks. In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming(CP'07)*.

Dylla, F.; Frommberger, L.; Wallgrün, J. O.; and Wolter, D. 2006. SparQ: A toolbox for qualitative spatial representation and reasoning. *In Proceedings of the Workshop on Qualitative Constraint Calculi: Application and Integration at KI*.

Gantner, Z.; Westphal, M.; and Wölfl, S. 2008. GQR - a fast reasoner for binary qualitative constraint calculi.

Ladkin, P. B., and Reinefeld, A. 1992. Effective solution of qualitative interval constraint problems. *Artificial Intelligence* 57(1):105–124.

Ligozat, G. 1996. A New Proof of Tractability for ORD-Horn Relations. In *Proc. of the Thirteenth Nat. Conference on Artificial Intelligence (AAAI'96)*, volume 1, 395–401.

Nebel, B., and Bürckert, H.-J. 1995. Reasoning About Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *Journal of the ACM* 42(1):43–66.

Nebel, B. 1996. Solving hard qualitative temporal reasoning problems: Evaluating the efficienty of using the ORD-Horn class. In *Proceeding of the Twelfth Conference on Artificial Intelligence (ECAI'96)*.

Pujari, A. K.; Kumari, G. V.; and Sattar, A. 1999. INDU: An interval and duration network. In *Australian Joint Conference on Artificial Intelligence*, 291–303.

Renz, J. 2007. Qualitative spatial and temporal reasoning: Efficient algorithms for everyone. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 526–531.

Sadeh, N. 1991. *Look-ahead Techniques for Micro-opportunistic Job Shop Scheduling*. Ph.D. Dissertation, Pittsburgh, PA 15213.

Van Beek, P., and Manchak, D. W. 1996. The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research* 4:1–18.

Van Dongen, M., and Lecoutre, C., eds. 2006. *Proceedings of the 3rd International Workshop on Constraint Propagation And Implementation (CPAI'2006) held with CP'2006*.

Wallgrün, J. O.; Frommberger, L.; Dylla, F.; and Wolter, D. 2006. SparQ user manual v0.6. *Technical Report 007-07/2006, Cognitive Systems - SFB/TR 8 Spatial Cognition*.