Constraint monitoring in TOSCA *

Howard Beck
Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

Introduction

The Job-Shop Scheduling Problem (JSSP) deals with the allocation of resources over time to factory operations. Allocations are subject to various constraints (*e.g.*, production precedence relationships, factory capacity constraints, and limits on the allowable number of machine setups) which must be satisfied for a schedule to be valid.

The identification of constraint violations and the monitoring of constraint threats plays a vital role in schedule generation both in terms of (i) directing the scheduling process and (ii) informing scheduling decisions. This paper describes a general mechanism for *identifying constraint violations* and *monitoring threats to the satisfaction of constraints* throughout schedule generation.

Identifying constraint violation To achieve a valid result in which all constraints are satisfied, a scheduler must be capable of distinguishing between valid and invalid solutions. This involves, at minimum, being able to identify constraint violations in fully-generated schedules. Clearly, if the scheduler is *only* able to identify constraint violations in fully-generated schedules, backtracking can only be introduced after considerable computational effort has already been expended. To avoid wasted effort, the scheduler should be capable of identifying *failed states* (*i.e.*, states from which it will be impossible to *achieve* a valid solution) during the process of generating the schedule. The earlier that failed states can be identified, the less unnecessary work need be done.

Monitoring of threats to constraints Given a particular factory capacity, constraint violations may be identified from the specification of the factory problem itself and could lead to a respecification of the problem. Alternatively, constraint violations may be (inadvertently) introduced by decisions taken by the scheduler. To avoid taking such decisions, potential threats to constraint violations may be tracked by a lookahead analysis (*e.g.*, [Liu88, Sad91]). Potential constraint violations occur where the magnitude of the estimated demand is close to the available capacity. Monitoring constraint threats may be used to *direct the scheduling process to the most critical constraints* and *inform the decision making process.*

*This research is supported by Hitachi Ltd.

Constraint Monitoring

Methods of constraint monitoring assuming distributions of operation demand

The monitoring of temporal-capacity constraints has been a central aspect of a number of scheduling systems (*e.g.*, [Liu88, Sad91, Ber91]). Each of these systems has been concerned with estimating demand on resources over time to allow comparisons with available capacity to be made.

Although there are important differences between the methods adopted for monitoring temporal-capacity constraints, the general approach adopted for estimating demand is based on assumptions as to the demand each operation imposes on a resource. In the case of RESS-II [Liu88], operation demand is assumed to be split equally across the valid timewindow of the operation. In the case of MICRO-BOSS [Sad91], operation demand is assumed to be split across the valid timewindow of the operation on essentially the inverse proportion of the cost associated with different start times.

Temporal-capacity analysis provides strategic information to the scheduler by highlighting critical resource time periods. This information can then be used during schedule generation to choose which particular resource time period to address next, to choose which operation to allocate and when to allocate the operation to effectively redistribute estimated resource demand.

Limitations of making assumptions about distributions of operation demand

It is in undertaking an analysis based on *splitting operation demand into a number of separate time periods* that limitations are introduced in that:

1. the estimated demand for resource over time introduces uncertainties associated with assumptions made regarding operation demand over time
2. contiguous time periods are not recognised as being contiguous

For schedulers undertaking an analysis of temporal-capacity constraints based on splitting operation demand over time, capacity bottlenecks indicate regions of high resource contention. As a result of the uncertainties introduced by the assumptions made regarding estimated operation demand, it is not possible to tell, even where the estimated demand is greater than available capacity, whether a capacity constraint has been violated or not. This is illustrated in the next section.

Constraint monitoring in TOSCA

TOSCA analyses temporal-capacity and setup-capacity constraints throughout the factory capacity hierarchy across multiple time periods. Operation demand is represented down to the granularity where the operation must legally occur, *i.e.*, the full operation demand is associated with the legal timewindow of the operation. The operation demand is not subdivided over the duration of its legal timewindow, avoiding the need to assign probabilities to the possible start times of each operation. Normally the operation timewindow is set by the release date and due date of the job and the intra-lot temporal relationships. Aggregated demand can be checked against available capacity both before and during schedule generation.

An example

To distinguish the TOSCA approach, a small example is considered using, in the first case, a method based on assumptions as to the distribution of operation demand and, in the second case, the method adopted in TOSCA which avoids such assumptions. The example involves the allocation of three operations to a single resource which is available for 7 hours per day. For the purpose of capacity analysis, the schedule timeline is split into periods of 1 day duration.

Demand:

Operation	Duration (Hrs)	Earliest Start (Day)	Latest End (Day)
op1	18 hrs	1	4
op2	3 hrs	2	5
op3	12 hrs	2	3

Capacity:
7 hours per day

Figure 1: Single resource example

Method 1: Constraint monitoring assuming distributions of operation demand

Constraint monitoring typically involves:

- maintaining an up-to-date representation of the legal timewindow of each operation throughout schedule generation
- splitting the timeline into discrete periods for the purpose of analysis
- for each operation, making assumptions about the likelihood of start times across its legal timewindow
- for each operation, calculating an expected operation demand across its legal timewindow
- aggregating demand for individual resources and comparing it against available capacity

Resource bottleneck periods (*i.e.*, periods where demand is high relative to available capacity) indicate potential threats to capacity constraints and are typically used to direct the scheduler to the most critical parts of the remaining schedule.

Methods which split operation demand across the operation timewindow assume that each operation exerts a demand across *each* of the discrete time periods under consideration that fall within the operation's timewindow. For instance op1 exerts a demand in periods day1, day2, day3 and day4. Every operation which could *possibly* be active over a particular time period contributes to the overall aggregate demand over that time period. In this example, the three operations (op1, op2, op3) all contribute to the estimated resource demand in day2.

Bottlenecks where *estimated* demand exceeds available capacity cannot be used for the purpose of detecting constraint violations. Where estimated demand exceeds available capacity, it may or may not be possible to redistribute demand away from the bottleneck and so avoid a constraint violation.

Figure 2 indicates a distribution of operation demand based on an assumed uniform probability distribution of start times. Figure 3 shows the aggregation of the demand of these operations, with the horizontal dashed line indicating the available capacity. The vertical dashed lines indicate the granularity of capacity analysis.

Method 2: Constraint monitoring without assuming distributions of operation demand

In TOSCA, the demand of an operation is associated with its temporal constraints (*i.e.*, its legal timewindow), *without assuming any subdivision of that demand across the timewindow.* An operation's demand is associated with a *single* time period. For instance, op2

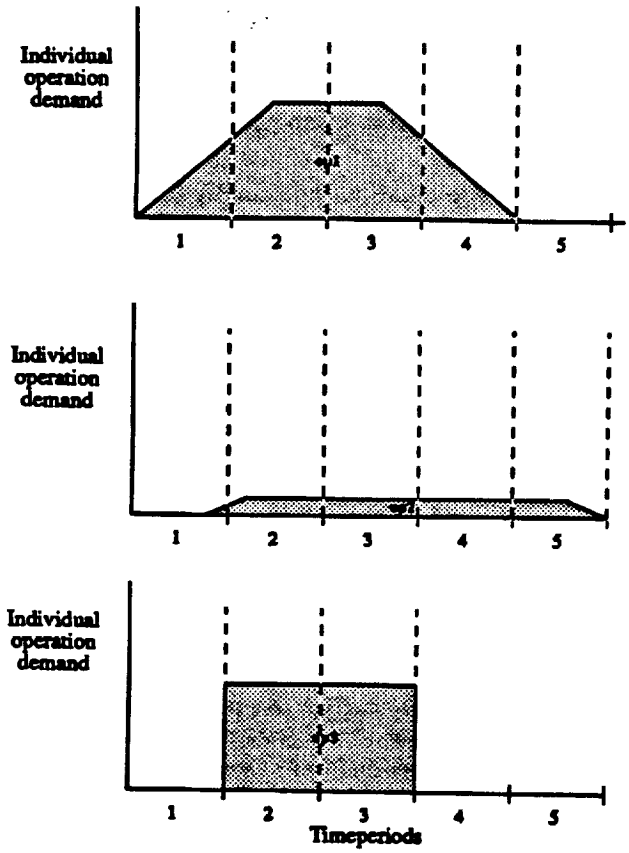

Figure 2: Individual operation demand assuming a uniform operation start time distribution

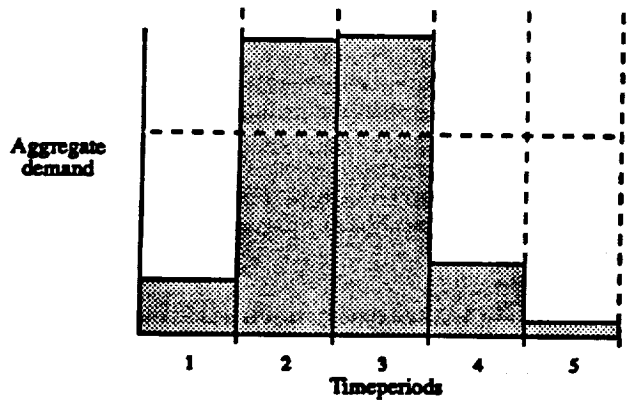

Figure 3: Estimated aggregate demand assuming a uniform operation start time distribution

exerts a demand of 3 hours over the period [2, 5], no assumptions being made regarding the probabilistic distribution of that demand within that period.

Only operations which are *necessarily* active, given that their temporal constraints are to be satisfied, contribute to the aggregate demand over the time period. That is, demand arises from only those operations whose legal timewindow are subperiods of the period under consideration. For instance, only the demand of op1 and op3 are associated with the time period [1,4]; the demand of op2 is not included.

Figure 4 shows the demand over time associated with the individual operations. op1 has a demand of 18 hours associated with the period [1, 4], op2 has a demand of 3 hours associated with the period [2, 5]; and op3 has a demand of 12 hours associated with the period [2, 3].

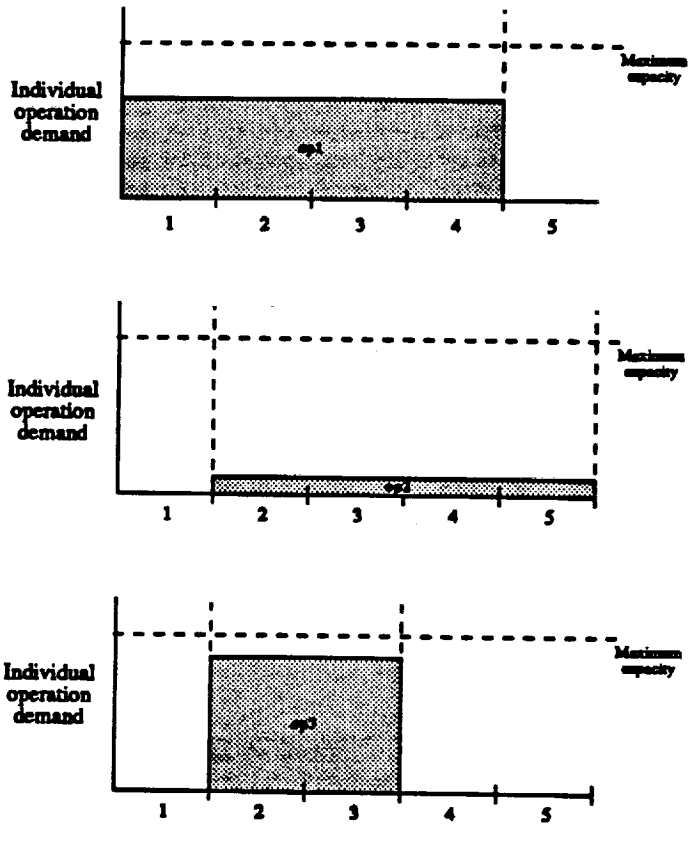

Figure 4: Individual operation demand not assuming an operation start time distribution

In estimating resource demand, temporally overlapping operations are aggregated. The operations op1 and op2 together ({op1, op2}) have a demand of 21 hours over the period [1, 5], {op1, op3} have a demand of 30 hours over the period [1, 4], {op2, op3} have

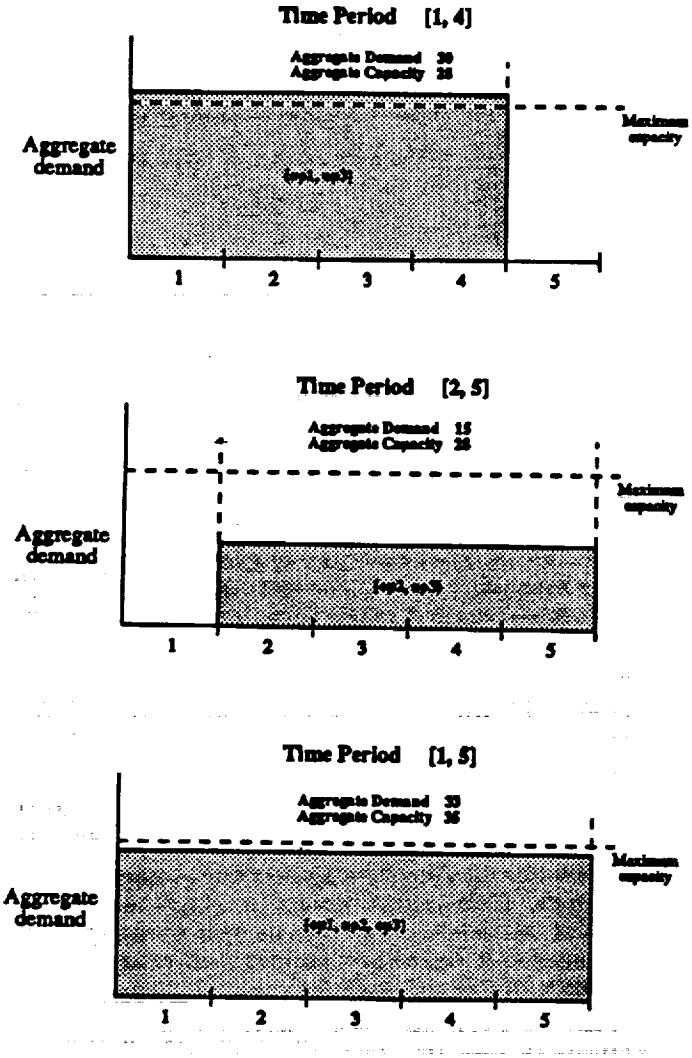

Figure 5: Aggregate demand not assuming operation start time distribution

a demand of 15 hours over the period [2, 5] and all three operations together have a demand of 33 hours over the period [1, 5]. Where multiple sets of operations are associated with a time period, the demand is that of the *maximal set* of operations. This means that the demand on the period [1, 5] is 33 hours, the demand associated with {op1, op2, op3} rather than {op1, op2}.

The demand associated with any time period can be directly compared with the available capacity — in this example, 7 hours per day — to find constraint violations and threats. A capacity constraint violation is indicated by the demand of {op1, op3}, its demand being greater than the maximum available capacity over the period [1, 4]. Figure 9 shows the demand associated with the maximal sets of operations associated with the periods [1, 4], [2, 3], and [1, 5].

In that each timeline period is associated with a set of necessary operations - assuming that the operation timewindow constraint holds - the operations implicated in a constraint violation can be readily identified. This can be used to inform constraint relaxations. In this example, the timewindow and duration constraints of op1 and op3 introduce a constraint violation. One of *their* constraints will need to be relaxed to avoid this constraint violation. Altering the constraints of op2, another operation active over this period, will not avoid the violation of the capacity constraint in the period [1, 4].

Scheduling in TOSCA involves the *iterative refinement* of the timewindow of each of the operations. Each decision to restrict the timewindow of an operation has the effect of redistributing resource demand. Before scheduling begins, op1 has a demand associated with the period [1, 4]. In deciding, for example, to restrict the timewindow of op1 to end by the third day at the latest, the operation demand becomes associated with the period [1, 3]. The effect of these decisions is monitored using *habographs*.

Constraint monitoring using habographs Habographs (Hierarchical Abstraction for Balancing Objectives) are two-dimensional datastructures used within TOSCA to represent and monitor temporal-capacity constraints. Habograph coordinates are given as start-end pairs and refer to cells representing a time period at a resource. Each operation's earliest start time is plotted on the y axis and its latest end time is shown on the x axis. Since it does not make any sense to have an earliest start time which is later than a latest end time all of the cells above the leading diagonal are always empty. The units of the axes are problem-dependent.

In referring to habographs it is important to be clear about the use of a couple of terms with respect to information held at a habograph cell: *local* and *aggregate*. A cell refers to a time period at a resource. Information about a resource time period may or may not include information about its sub-period.

Figures 7 and 9 present an illustration of local and aggregate demand in habographs on the example described above.

Cell	Local operations	Local Demand
[1, 4]	{op1}	18
[2, 5]	{op2}	3
[2, 3]	{op3}	12

Figure 6: Local demand

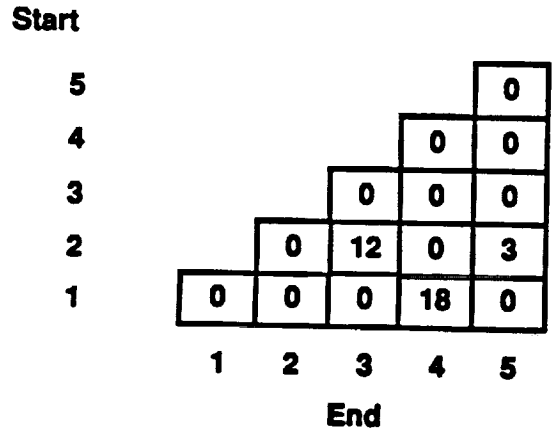

Figure 7: Habograph showing local demand

Figure 7 indicates the local operations over the periods: [1, 4], [2, 5] and [2, 3]. op1 is local to [1, 4], op2 is local to [2, 5] and op3 is local to [2, 3].

Cell	Aggregate operations	Aggregate Demand
[1, 4]	{op1,op3}	30
[2, 5]	{op2,op3}	15
[2, 3]	{op3}	12
[1, 5]	{op1,op2,op3}	33

Figure 8: Aggregate demand

Figure 9 indicates the aggregate set of operations over three time periods. The aggregate set of operations includes all the operations which must be processed in a particular period. In the period [1, 4], two operations must be processed, these being: op1, which must occur between [1, 4] (*i.e.*, day1 through day4), and op3, which must occur in the subperiod [2, 3] (*i.e.*, day2 through day3).

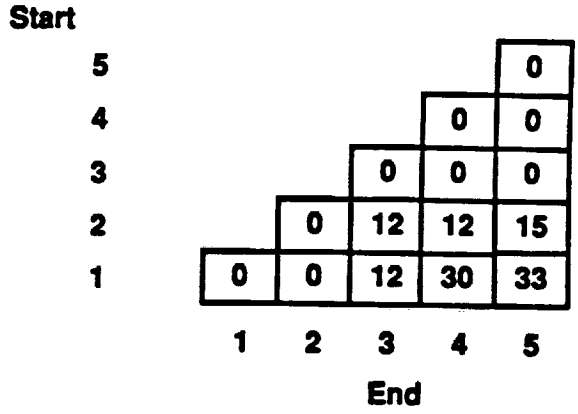

Figure 9: Habograph showing aggregate demand

The contents of habograph cells Each cell within a habograph has a representation of number of objects. The main object within each cell is a list of the operations which are local to that cell. Each of these operations exerts a demand for capacity at that cell and the sum of the demand exerted by all the cell's local operations is stored as the cell's *local demand*. Each cell also has an *aggregate demand* figure, a number calculated by summing all the local demands in all of the cells that are above and to the left of the current cell.

In addition to the demand associated with a set of operations, information is also held as to the capacity available over the time period represented by the cell. As with demand, capacity information is represented by a local and an aggregate figure. *Local capacity* is represented only over the *leading diagonal* of the the habograph. In the example under consideration, the capacity of 7 hours per day is represented along the leading diagonal with zero's everywhere else, as is shown in Figure 10. *Aggregate capacity*, shown in Figure 11, is calculated in the same manner as the aggregate demand, described above, except summing the local capacity figures rather than the local demand.

Finally the cell also has a representation for *demand pressure* (Figure 12). This is simply the ratio of the aggregate demand at that cell, divided by the aggregate capacity of that cell. Where the demand pressure is greater than one, a constraint violation is indicated. Where the demand pressure is close to but less than one, a constraint threat is indicated. In this example, a constraint violation is indicated over the period [1, 4].

Conclusion

Most current approaches to capacity constraint monitoring involve assumptions regarding the probabilistic

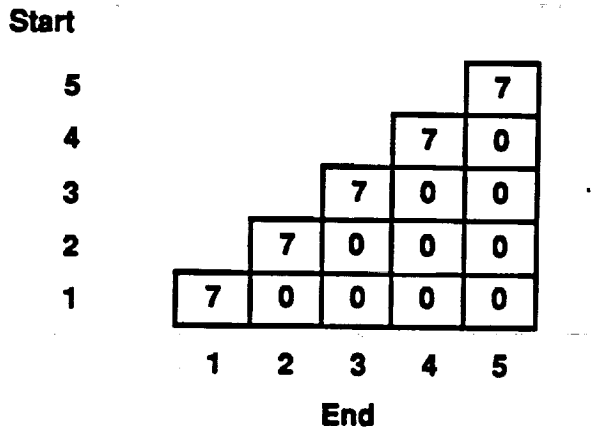

Figure 10: Habograph showing local capacity

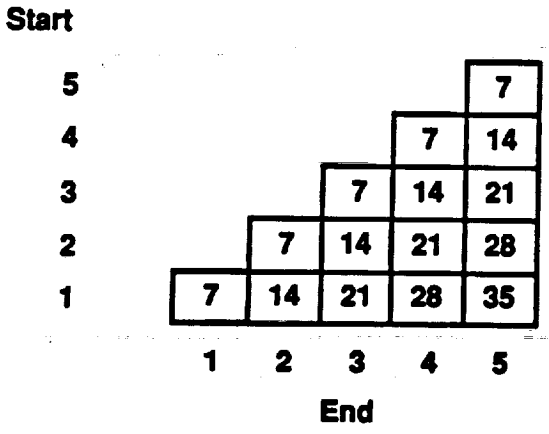

Figure 11: Habograph showing aggregate capacity

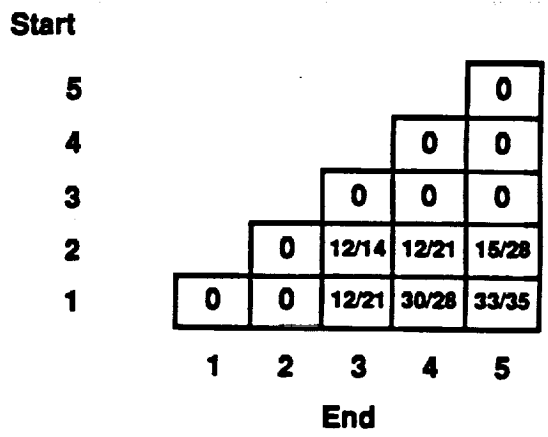

Figure 12: Habograph showing demand pressure

distribution of operation start times. Such approaches indicate resource bottleneck periods (*i.e.*, periods of potential constraint threat) but are unable to identify constraint violations.

This paper describes *habographs*, a novel datastructure, used for capacity constraint monitoring in TOSCA. The approach avoids assumptions regarding the probabilistic distribution of operation start times and has the advantage of enabling the identification of resource bottleneck periods which necessarily involve a constraint violation.

Habographs are currently being investigated within the TOSCA project as a unifying representation to support resource allocation, temporal allocation and setup management.

References

[Ber91] Pauline Berry. *A Predictive Model for Satisfying Conflicting Objectives in Scheduling Problems.* PhD Thesis, University of Strathclyde, Glasgow, 1991.

[Liu88] Bing Liu. Scheduling via reinforcement. *Journal of AI in Engineering*, 3(2), 1988.

[Sad91] Norman Sadeh. *Look-ahead Techniques for Micro-oportunistic job shop scheduling.* PhD Thesis, School of Computer Science, Carnegie Mellon University, 1991. (CMU-CS-91-102).