A PROBABILISTIC FRAMEWORK FOR RESOURCE-CONSTRAINED MULTI-AGENT PLANNING*

Nicola Muscettola
Dipartimento di Elettronica
Politecnico di Milano
20133 Milano, ITALY

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA

ABSTRACT

In this paper, we consider the problem of temporally coordinating the resource demands of a set of independent agents. We assume that resources are unreliable, making it necessary to retain imprecision in the execution times assigned to specific agent operations. To this end, a probabilistic model of resource allocation is developed for use in estimating the consequences of execution intervals (representing sets of possible resource allocation decisions). This leads to a probabilistic representation of requests for resource usage for which resource congestion constraints can be defined. We consider two applications of the framework: prediction of bottleneck resources and time bound scheduling.

I COORDINATING RESOURCE USAGE IN MULTI-AGENT PLANNING

Multi-agent planning involves coordination of the operations of a set of independent agents so as to achieve the goals of each individual agent. Of central importance here is consideration of the possible interactions among agents, which can lead to situations of mutual advantage (in the case of cooperative agents) or interference (if agents have conflicting objectives and do not cooperate). Research in multi-agent planning [4, 8] typically assumes that the goals being pursued by agents relate strictly to the achievement of a certain state. Emphasis is placed on determining "how" this state can be achieved in the presence of other agents. In time-constrained problem domains, however, the goals of agents often relate more to "when" a specific state can be achieved than "how". The problem of factory scheduling, which has been the context of our work, is perhaps an extreme example of this situation. Parts must be produced for multiple orders (agents) to meet imposed deadlines, minimize production time and satisfy other factory objectives, and the primary source of interactions among agents is the need to share resources (e.g. machines, operators, tools). Interactions among agents in this domain very rarely prevent achievement of each agent's desired state (i.e. the finished parts) but can significantly affect the circumstances under which this desired state is achieved (e.g. the finish date, the duration of the production process, etc J. Dealing with these types of interactions is the subject of this paper.

Assuming a cooperative planning framework, there is much to be gained by anticipating situations of resource contention and attempting to temporally coordinate requests for these resources. Analyses of projected resource demands can guide individual agents in planning to achieve their goals (e.g. specific resources to avoid when possible). Advance coordination of demands for heavily utilized resources can maximize the common utility of agent's plans (e.g. make the most efficient use of shared resources) [10]. An important issue, of course, is the level of detail at which "deals" among agents concerning future resource usage should be made. While we might expect that precise temporal coordination of requests for each shared resource would lead to the best possible plans, such plans ignore the dynamics of the planning environment. Resources are characteristically unreliable (e.g. susceptible to unanticipated periods of unavailability, sometimes fail to produce the desired effect, etc.), and consequently, advance commitment to a detailed course of action and timetable is of little use. Agents should seek less precise agreements concerning resource access that can usefully guide plan refinement as the external environment allows and requires it.

Less detailed agreements among agents require more abstract problem descriptions. To this end, we can formulate problems in which agent operations require capacity on aggregate resources (representing functional groupings of individual resources) over their expected durations, and we will assume in this paper that both the amount of capacity required by a given operation and its duration are fixed approximations of more detailed suboperation characteristics. We also assume that resource capacity is required throughout an operation's duration. As with the identity of the resource that will be used to perform a given operation, we would like to remain imprecise with respect to operation execution times. This, however, presents a difficult problem. Classical deterministic scheduling methods [1] must make specific allocation decisions in order to take resource capacity constraints into account This defeats much of the purpose of abstracting in the first place since the resulting plans will designate a single point in the temporal dimension of the abstract solution space. We might consider generalizing these temporal constraints to delineate sets of possible allocation decisions after the fact. However, given the high degree of interaction between allocation decisions, it is difficult to imagine how this could be accomplished in any meaningful fashion. We might also consider reasoning with a coarser granularity of time at the aggregate level (e.g. time steps of hours instead of minutes), but this also introduces temporal imprecision in a fairly arbitrary fashion.

In this paper, we present an approach to reasoning with temporally imprecise requests for resource capacity (representing sets of possible allocation decisions) that provides a oasis for producing more meaningful aggregate plans. This is accomplished by adopting a probabilistic view of resource allocation and injecting randomness into the decision-making process. At the same time, we can describe characteristics of the stochastic allocation process that enable the definition of consistency constraints analogous to those that would result from a deterministic model. Furthermore, we can bias the stochastic process to reflect the strategies and preferences of the actual deterministic allocator (i.e. the generator of final decisions for actual execution).

II PROBABILISTIC RESOURCE ALLOCATION

As stated above, we propose the use of a probabilistic model of resource allocation as a means of reasoning about sets of possible allocation decisions. We are interested in a

*This research was supported in part by the Air Force Office of Scientific Research under contract F49620-82-K-0017 and the Robotics Institute, and was performed while Nicola Muscettola was a visiting scholar in the Intelligent Systems Laboratory of the Robotics Institute.

mechanism for evaluating an abstract, temporally underconstrained set of plans with respect to its expected requests for resource capacity. Our approach is to develop a random model of the actual resource allocation process, ana to use the probabilistic characteristics of randomly generated allocation decisions as the basis for plan evaluation. In modeling resource allocation as a stochastic process, it is crucial tnat the random allocator take into account not only the restrictions specified in the abstract plan, but also the preferences and strategies of the actual deterministic allocator. Thus, we will conceptualize the allocation process as consisting of a *random start time generator* (RSTG) that stochastically selects a specific allocation for each operation in accordance with this larger set of constraints, which we will call the RSTG-constraints. We will refer to the set of allocation decisions produced by RSTG as the RSTG-allocations.

A. Constraining the Random Allocation Mechanism

We distinguish three different types of RSTG-constraints:

1. *temporal constraints* - These constraints refer to the earliest start time and latest end time restrictions associated with each operation op_r Since we assume that each op_i has a fixed duration, designated henceforth as *bopp* these constraints define the set of the possible start times for each op_p $STI(op_t)$ (Start Times Interval), an interval of time bounded by *est(op),* the operation's earliest start time, and *lst(op),* its latest start time** The temporal constraints, and consequently the ST/s, associated with operations belonging to the sameplan must be mutually consistent. Consider Figure 1, where $STI(op_1)$ is depicted in the context of a two step plan. In this case, $est(op_2)$ must not precede r, in time, since $STI(op_2)$ would otherwise contain allocation possibilities that would introduce a conflict Similarly, $lst(op)_2$ must not precede t_2 in time. These consistency conditions are precisely those that are maintained in many existing temporal planners (e.g. [11]). *STI(op)* must also be consistent with currently imposed resource capacity limitations. Such consistency conditions are discussed in Section IV.

2. *allocation strategy* - This constraint dictates the sequence in which allocation decisions will be generated. It specifies a partial ordering of all the operations constituting the planning problem. For example, a plan by plan in priority order strategy dictates that all operations of a given agent's plan will be considered before those of any agents with lower priority. Operations belonging to the same plan are ordered according to a given plan scheduling strategy (e.g. forward from the first operation).

3. *preference constraints* - These constraints define

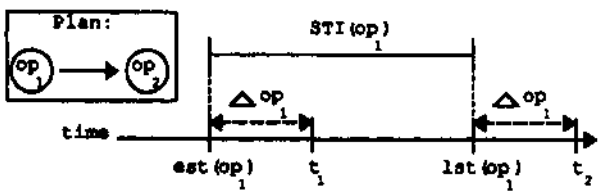

Figure 1: A two operation plan with $STI(op_1)$ fixed

To simplify the presentation below, we assume *STI(op)* to be a connected interval. The framework can be easily generalized to accommodate a discontinuous set of possible operation start times.

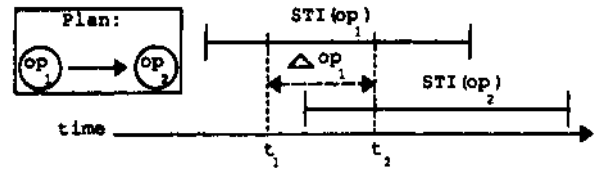

Figure 2: A two operation plan with op_1 starting at time t_1

objectives and concerns that influence actual allocation decisions. As has been pointed out in previous research [2, 3], we can model specific allocation preferences as real functions over time that estimate the relative desirability of various allocation decisions. For example, if an operation has a fixed due date with a delay penalty, we can express the tardiness constraint as a function that decreases after the due date at a rate proportional to the marginal tardiness loss. Constraints on finishing early (e.g. factory holding costs) can be expressed similarly, with the utility function increasing until the due date is reached in this case. The formulation of other allocation preferences depends not only on problem characteristics (as do the above examples) but also on prior allocation decisions. A desire to minimize idle time between operations, for example, can be expressed with a utility function that decreases from the current earliest start time of a given operation onward. However, this constraint is meaningful only in the case of an *op.* that is preceded in the allocation strategy by one or more upstream operations in *op's* plan, and its exact structure can be defined only when actual allocation times have been determined for these preceding operations. Nonetheless, all active constraints are known when making a specific allocation decision, and we can therefore obtain their combined "level of satisfaction".

B. Modeling the Behavior of the Random Allocation Process

We can derive, for each *op* the probability of it starting around a certain instant of time in *STI(op),* expressed by the density $P_{ft}(op_r t)$. This is of central importance to our approach, as we will see in Section III when we consider the representation of a resource's available capacity. Here, we consider our model of the behavior of RSTG and the derivation of $P_{st}(op_i, t)$.

As allocation decisions are generated by RSTG, it is quite possible that $STI(op_t)$ for a given op_i will become further constrained. In Figure 2, for example, the decision to schedule op_1 at r_1 constrains RGST to start op_2 after t^2 We will refer to actual set of possible allocation times for *op.* at the point it is considered by RSTG as *ASTI(pp)* (Actual Start Times Interval). Since *ASTI(op)* is a function of random decisions, its limits are themselves random variables.

RSTG will select a start time in *ASTI(op)* according to a *choice rule,* a density of probability that defines, for every point in $ASTI(op)_{\%}$ the likelihood of the start time falling in its neighborhood. The choice rule is constructed to reflect the combined value of all active preference constraints according to the following criterion:

> *Choice Rule Criterion: Given two possible start times t_1 and t_2 if the value of the combined constraints at t_1 is c times that in t_2 the likelihood of selecting t_1 will be c times that of t_2*

If *ASTI(op)* is the interval $t_{min} \le t \le t_{max}$, we indicate the choice rule for op_i with $P_{rule}(op_i, t / t_{min}, t_{max})$. If v(r) is the value in *t* of the combined active constraints, the *Choice Rule Criterion* becomes

$$\frac{P_{rule}(op_i, t_1 / t_{min}, t_{max})}{P_{rule}(op_i, t_2 / t_{min}, t_{max})} = \frac{\nu(t_1)}{\nu(t_2)}$$

This completely determines the *choice rule* and together with the normalization condition for a density of probability we have:

$$\int^{\tau \in ASTI(op_i)} P_{rule}(op_i, \tau / t_{min}, t_{max})\, d\tau = 1$$

We are now ready to determine $P_{st}(op_i,t)$ over *STI(op).* First some notation. We will assume that the operations in a given plan are indexed sequentially from the first operation to the last. Further, we will indicate with *PREC(op)* (resp. *FOLL{op))* the set of operations belonging to the same plan as op_i that both precede (resp. follow) it in the allocation strategy and precede (resp. follow) it in the plan.

We distinguish three cases:

1. *Both PREC{op) andFOLL(op) not empty.*

   ASTI(op) is therefore determined by the start times generated for op_r the operation with maximum index in *PREC{op)* and op_r, the operation with minimum index in *FOLL{op).* If these are respectively $t_1$and t_2 the left and right limit of *ASTI{op)* are given respectively by:

$$ASTI_{min}(op_i, op_l, t_1) = t_1 + \Delta op_{l,i}$$
$$ASTI_{max}(op_i, op_r, t_2) = t_2 - \Delta op_{i,r}$$

   where $\Delta op_{n,m} = \sum_{k=n}^{m-1} \Delta op_k$. We indicate with $P_{st}(op_i;t_1p_i;t_2)$ the conjunct density of probability of the start times of op_l and op_r to be respectively t_l and t_T We have:

$$P_{st}(op_i, t) = \int_{est(op_i)}^{t-\Delta op_{i,l}} d\tau_1 \int_{t+\Delta op_{i,r}}^{lst(op_r)} P_{rule}(op_i, t / ASTI_{min}(op_i, op_l, \tau_1), ASTI_{max}(op_i, op_r, \tau_2)) \times P_{st}(op_l, \tau_1, op_r, \tau_2)\, d\tau_2$$

2. *Only one ofPREC(op) or FOLL(op) empty.*

   Suppose *PREC(op)* is empty. Then *ASTI(op)* is always limited to the left by *est(op).* We have:

$$P_{st}(op_i, t) = \int_{t+\Delta op_{i,r}}^{lst(op_r)} P_{rule}(op_i, t / est(op_i), ASTI_{max}(op_i, op_r, \tau)) \times P_{st}(op_r, \tau)\, d\tau$$

   A similar expression holds if *FOLL(op)* is empty.

3. *Both PREC(op) and FOLL(p) empty.*

   We have trivially:

$$P_{st}(op_i, t) = P_{rule}(op_i, t / est(op_i), lst(op_i))$$

Notable special cases follow from assumptions concerning the allocation strategy that are typical in scheduling/planning work. Assume that the first and last operations in a given plan are indexed m and *n* respectively. If allocation decisions for the operations in the plan are generated sequentially starting from the first operation and moving forward, we have:

$$P_{st}(op_m, t) = P_{rule}(op_m, t / est(op_m), lst(op_m))$$

$$P_{st}(op_i; t) = \int_{est(op_{i-1})}^{t-\Delta op_{i-1}} P_{rule}(op_i, t / ASTI_{min}(op_i, op_{i-1}, \tau), lst(op_i)) \times P_{st}(op_{i-1}, \tau)\, d\tau$$

Similar expressions hold if allocation decisions are made sequentially starting from the last operation and moving backward. Moreover, these forward and backward formulas can be directly extended to strategies in which the operations in a plan are scheduled outward from some initial anchor (e.g. a bottleneck operation).

III PROBABILISTIC REPRESENTATION OF CAPACITY REQUESTS

We can now represent the consequences of RSTG-allocations with respect to a resource's available capacity. From the resource's point of view, the capacity requested by an RSTG-allocation is a discrete stochastic process, *R(opp).* At any instant of time / the process can assume a value n the units of capacity needed for the execution of op_r or a value 0, if the operation is not allocated over *t.* The instantaneous probabilities of these values are respectively:

$$P\,(R(op_i, t) = n) = \sum_{\tau = t - \Delta op_i}^{t} P_{st}(op_i, \tau)$$
$$P\,(R(op_i, t) = 0) = 1 - P\,(R(op_i, t) = n)$$

The total request for capacity on a resource ,res_k **at** time t, $R\{res_{k,t)}$s a discrete stochastic process with value

$$R(res_k, t) = \sum_{op_i \in \Omega} R(op_i, t)$$

where *Q.* is the set of operations allocated on the resource.

From the description of RSTG in Section II, we see that distinct RSTG-allocations on the same resource are statistically independent. RSTG can indeed consider the effects of prior RSTG-allocations in generating a particular allocation, but only indirectly through the currently posted RSTG-constraints. Given this statistical independence, it follows that if the number of allocations that overlap at time *t* is sufficiently high, we can approximate the first order distribution of $R(res_{k,})$ with a Gaussian random variable, with mpun and variance crivpn bv

$$E[R(res_k, t)] = \sum_{op_j \in \Omega} E[R(op_j, t)]$$
$$\sigma^2(res_k, t) = \sum_{op_j \in \Omega} \sigma^2(op_j, t)$$

We now need a way to compare the requests of capacity with that actually available, to detect or avoid possible situations of overflow. Notice that if we wanted to prohibit RSTG from generating overflow situations, we should post RSTG constraints that would imply, once implemented, a very low load on the resources. However, since RSTG is not intended to generate actual executable allocations but rather to evaluate constraints, we will adopt a less strict criterion that generalizes from deterministic methods for detecting capacity violations. We say that a resource is *congested* at time *t* if the probability of a high instantaneous request of capacity in *t* is greater than a fixed threshold . If $C(res_{k)}$ is the actual capacity of res_k and *T* is the congestion threshold, we can express the congestion constraint with:

$$1 - \Phi\left(\frac{C(res_k) - E[R(res_k, t)]}{\sigma(res_{,:}t)}\right) < T$$

where $\Phi(.)$ denotes the cumulative distribution for a standard normal random variable.

We note in passing that in addition to detecting overflow situations, we can use this constraint to deduce the maximum instantaneous probability of a given request for capacity that does *not* induce an overflow, assuming constraints over the allocation of other operations have Been fixed. This is discussed further in [6].

IV APPLYING THE PROBABILISTIC FRAMEWORK

Depending on what kind of RSTG-constraints are specified and how they are posted, the probabilistic framework can have different uses. We describe two applications below.

A. Bottleneck Detection

The opportunistic job-shop scheduling methodology described in [9,7] obtains notable advantages by first making allocation decisions at critical (i.e. bottleneck) resources. Since bottlenecks can vary depending on the characteristics of the scheduling problem, this methodology requires a focusing mechanism that identifies potential bottlenecks from a specification of the current problem.

Conceptually, we can define a probabilistic bottleneck detection mechanism as operating in two steps:

1. Assuming no interactions among agents (i.e. infinite resource capacity), RSTG is used to produce a schedule for each agent in accordance with its personal constraints (e.g. due date).

2. The consequences of these uncoordinated decisions are assessed with respect to available resource capacity. A resource will be considered a bottleneck during time intervals in which the congestion constraint of Section III is violated.

Correspondent to this definition, the RSTG-constraints posted for bottleneck detection will satisfy the following conditions:

- Since the goal is to analyze the effect of the initial temporal constraints of the problem, *STI(op)* for a given op_i will be the largest interval compatible with the release and due dates of the plan containing op_r

- Independence among agents implies that the allocation strategy will not impose ordering relations between operations belonging to different plans. Operations belonging to the same plan might be ordered according to some internal strategy (e.g. forward scheduling).

- For analogous reasons, only preference constraints related to temporal aspects of plans (e.g. work-in-process time) will be posted.

To see the advantage of this probabilistic mechanism, consider a deterministic counterpart. In this case, bottleneck analysis must be based on one potential shop schedule (e.g. the best possible schedule for each agent), and, consequently, the analysis is subject to idiosyncratic results (e.g. detected resource congestion might, in fact, disappear if one or more agent schedules where slightly altered). The probabilistic mechanism avoids this problem by considering the consequences of a set of possible solutions.

B. Time Bound Scheduling

Another application is time bound scheduling, the establishment of temporal constraints on each operation that refine initial problem constraints (i.e. release and due dates) without committing to specific allocation times to identify a solution sub-space from which the final executable scheaule will be generated. In this case, we assume a scheduling mechanism that iteratively adds (posts) RSTG-constraints (e.g. fixes an £77) for a previously unconsidered operation on each iteration. RSTG is used at each step to estimate the consequences of the constraints posted thus far, enabling the scheduler to avoid situations of resource congestion in its future decisions.

There are two ways in which knowledge of resource congestion can be included in the RSTG-constraints:

- In establishing a new $STI(op)_t$ time intervals in which resource congestion is already over the congestion threshold can be excluded. This consistency constraint is the analog of a resource unavailability constraint in deterministic representations of available capacity (e.g. [5]).

- Preference constraints can be introduced to weigh alternative allocations in *ASTI(pp),* using a utility function inversely proportional to the level of congestion of the resource.

V FINAL REMARKS

We have presented a framework for evaluation and generation of abstract plans in resource-constrained problem aomains. Our current research centers on the development of a computationally inexpensive implementation in the context of bottleneck detection. in particular, we feel that both P_{ngg} and P_{st} can be reasonably approximated with piece-wise constant functions over time.

REFERENCES

[I] Baker, K.R.. *Introduction to Sequencing and Scheduling.* John Wiley and Sons, New York, 1974.

[2] Fox, M.S. *Constraint-Directed Search: A Case Study of Job Shop Scheduling.* Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, 1983.

[3] Fox, M.S. and S.F. Smith. "ISIS: A Knowledge-Based System for Factory Scheduling". *Expert Systems* 7, 1 (July 1984), 25-49.

[4] Konolidge, K. and N.J. Nilsson. Multi-Agent Planning Systems. Proc. AAAI-80, August, 1980, pp. 138-144.

[5] LePape, C. and S.F. Smith. Management of Temporal Constraints for Factory Scheduling. Proc. AFCET Conf. on Temporal Aspects in Information Systems, Paris, May, 1987.

[6] Muscettola, N. Time Bound Scheduling: A Probabilistic Approach. The Robotics Institute, Carnegie Mellon University, in preparation.

[7] Ow, P.S. and S.F. Smith. Viewing Scheduling as an Opportunistic Problem Solving Process. In *Annals of Operations Research: Approaches to Intelligent Decision Support,* R.G. Jeroslow, Ed., Baltzer Scientific Pub., 1987.

[8] Rosenschein, J.S. and M.R. Genesereth. Deals Among Rational Agents. Proc. UCAI-85, Los Angeles, CA, August, 1985, pp. 91-99.

[9] Smith, S.F., P.S. Ow, C. LePape, B. McLaren, and N. Muscettola. Integrating Multiple Scheduling Perspectives to Generate Detailed Production Plans. Proc. SME Conf. on AI in Manufacturing, Long Beach, CA, September, 1986.

[10] Smith, S.F. and P.S. Ow. The Use of Multiple Problem Decompositions in Time-Constrained Planning Tasks. Proc. DCAI-85, Los Angeles, CA, August, 1985, pp. 1013-1015.

[II] Vere, S.A. "Planning in Time: Windows and Durations for Activities and Goals". *IEEE Trans, on Pattern Analysis & Machine Intelligence PAMI-5,*3 (May 1983), 246-267.