Mixing Planning and Scheduling to Model Complex Process Environments

Roman Barták[*]

Charles University, Prague, Czech Republic
bartak@kti.mff.cuni.cz
http://kti.mff.cuni.cz/~bartak

1 Introduction

Planning and scheduling attract a high attention of computer science community because of their real-life applicability and challenging complexity. However, despite of their similar character, planning and scheduling problems are usually handled separately using different methods and technologies. *Planning* deals with finding plans to achieve some goal, i.e., finding a sequence of actions that will transfer the initial world into one in which the goal description is true. Moreover, the feasible sequences of actions must respect constraints describing the limitations of the world. Opposite to planning, *scheduling* deals with the exact allocation of resources to activities over time respecting precedence, duration, capacity, and incompatibility constraints.

Such strict decomposition of the task into planning and scheduling parts is not desirable in some applications where activities must be introduced during scheduling. We describe several such problems from the area of complex process environments in the next section. Moreover, in industry, the border between planning and scheduling task is shifted a bit and it becomes a little bit fuzzy. Both tasks include generation and allocation of activities here; the resolution of the plan or schedule is the main difference between planning and scheduling in industry. The similarity of both tasks as well as some real-life problems bring us the idea of mixing both planning and scheduling within single system.

In the paper we describe the mixed planning and scheduling framework and we show how this approach can help in solving some real-life problems. We also overview some typical constraint models used to solve scheduling problems and we compare their applicability to solve mixed planning and scheduling tasks in complex process environments.

2 Problem Area

The problem area that we deal with can be characterised as a complex process environment where a lot of complicated real-life constraints bind the problem variables. Typical examples of such environments can be found in plastic, petrochemical, chemical or pharmaceutical industries. The task is to prepare a schedule for fixed period of time maximising the profit.

We deal with heterogeneous environments where several *different resources* like producers, movers, stores, workers and tools may appear. It is possible to use *alternative* resources to process the same activity and single resource can handle several activities in parallel (during batch production). In such a case, capacity and compatibility constraints must be considered to capture which activities and in what quantity may be processed together. Moreover the sequence of activities in single resource must follow special *transition patterns* describing feasible consecutive activities of given activity. Expensive *set-ups* included between consecutive activities or special activities like re-heating (Pegman, 1998) must also

[*] Supported by the Grant Agency of the Czech Republic under the contract no. 201/99/D057 and by InSol Ltd.

be assumed. During set-ups low-quality products, called *by-products*, may be produced that may be used as a raw material in further production to decrease the production cost. Thus, we must assume *re-cycling* of material in the factory as well as *cycling* of items to change their features. Figure 1 shows a typical complex-process environment.

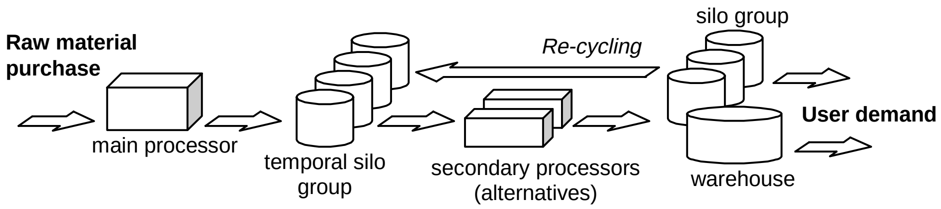

Figure 1 (example of complex process environment)

In many complex process environments, the production is not driven by custom orders only but it is necessary to schedule production for store according to the factory patterns and marketing forecast. Such production is called non-ordered production.

3 Towards Mixed Planning and Scheduling

In most current APS (Advanced Planning and Scheduling) systems the planning and scheduling components are separated and the communication between the components is limited. Such decomposition seems natural because the conventional planning and scheduling deal with a bit different tasks and different methods are used to solve the tasks. There also exist trends requiring even more decomposition of resource scheduling and planning (Srivastava & Kambhampati, 1999). Nevertheless, we argue for mixing both tasks within single system and, in next paragraphs, we give several examples where the mixed approach surpasses the conventional approach.

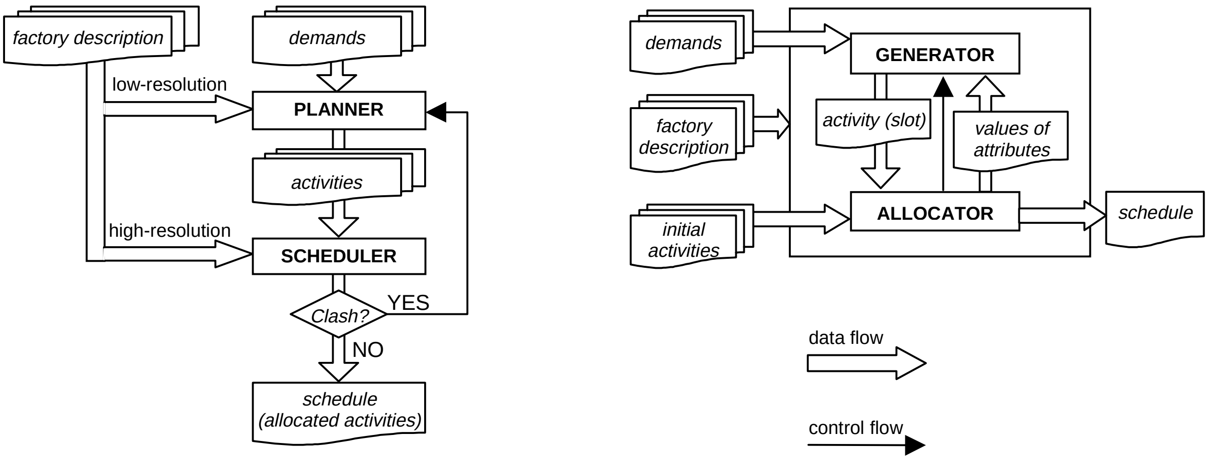

Figure 2 (conventional – left vs. mixed – right approach)

The interface between separated planning and scheduling modules is well defined; it is the set of activities that is passed from the planner to a scheduler. Thus the scheduler knows all the activities in advance and it just allocates them to resources. The first problem appears if the allocation fails due to clash between planned activities (in general such conflicts cannot be avoided). In such a case we must backtrack to the planner to find another plan. Another solution to prevent clashes is to postpone planning decisions to scheduling phase, i.e., to allow introduction of new activities during scheduling. In some situations, generating of activities during scheduling is necessary to solve the problem, typically when the appearance of the activity depends on allocation of other activities. Introduction of set-up activities is a nice example of such behaviour. Another example is consumption of by-products (until we know that the by-product is produced we cannot introduce activities for its consumption). Finally, if the activities are introduced during scheduling they may fill the gaps in production and, thus, we may exploit the resources better and produce more profitable schedules (this is non-ordered production).

The above paragraph gives evidence that we need more tighten co-operation between planner and scheduler. What we propose is to pass activities from the planner to the scheduler as soon as they are generated. Then, the planner may use information from the scheduler about (partial) allocation of activities when deciding about next activities. Figure 2 compares the conventional approach with our mixed approach.

4 Constraint Models

Currently, many scheduling problems are solved using constraint satisfaction technology, which expects a static formulation of the problem, i.e., all the variables and constraints are known beforehand. However, this [static] formulation is not suitable for planning because of high variability of plans and impossibility to predict which activities will be used in which combinations (Nareyek, 2000). The same dynamic character can be identified in the proposed mixed planning and scheduling (MPS) approach.

To formalise which constraints will be dynamic during scheduling, we classify the constraints into three groups: resource, transition and dependency constraints. *Resource constraints* describe limitations of given resource at given time (capacity, compatibility etc.), *transition constraints* describe behaviour of single resource in time (e.g., feasible transitions between activities) and, finally, *dependency constraints* specify relations between resources (like supplier-consumer relation). Typically, in problems where MPS is required the constraints of all above types appear. Depending on the constraint model chosen to describe the problem, the constraints in particular group will be dynamic, i.e., introduced during scheduling. We identified three basic constraint models used in scheduling applications.

Time-line model uses discrete time intervals called time slices or time slots. It expects the behaviour of the resource to be homogenous within the time slice; i.e. activities may change at the border between slices only. Thus, the duration of the slice is computed as the greatest common divisor of activities' duration (due to efficiency issues we prefer smaller number of slices, therefore GCD). The behaviour of the resource is described by a chunk of variables assigned to each slice. Then the resource constraints bind variables from single slice, the transition constraints bind variables from consecutive slices and dependencies bind variables from slices of different resource. Note that we allocate the activity to the slice simply by choosing the value for activity variable in the slice and that the activity may spread over several slices. Consequently, we need not any special activity generator and mixed planning and scheduling is integral part of this model. The only problem here is posting dependency constraints because until we know the activity in the slice, we cannot decide which slices are suppliers and consumers respectively (the processed items are unknown). We may either introduce these constraints dynamically during scheduling or we may equip static constraints by triggers that fire the constraint if particular situation occurs. The choice depends on complexity of triggers and required propagation through dependencies.

Task-centric model is a model based on activities that are grouped into tasks. Task is a sequence (or tree etc.) of activities connected by dependency constraints, usually priority constraints are used only. In this model resource and transition constraints should be introduced dynamically because of too complicated triggers for the static representation. This model is common in most conventional scheduling problems because they do not include transition constraints at all and the resource constraints are very simple. However, in complex process environments both resource and transition constraints play important role and, thus this model is less suitable for them. To allow MPS we should use activity slots instead of activities here; these slots are filled during scheduling by real activities. This allows us to model alternative chains of activities in the task but it still does not solve the problem with introduction of set-ups, activities for processing of by-products and for non-ordered production and (re-)cycling

Resource-centric model is orthogonal to the task-centric model; the activities are grouped per resource here. This simplifies expressing the resource and transition constraints but complicates the dependencies that must be posted dynamically. In MPS we use sequence of slots per resource and these slots are filled by activities during scheduling. Opposite to time slices these slots are not fixed in time and the allocation is part of scheduling (also activity is in single slot only). Similarly to the time-line model, there is no problem with set-ups, (re-)cycling and processing of by-products or non-ordered production. Thus, this model is better for complex process environments than a more common task-centric model.

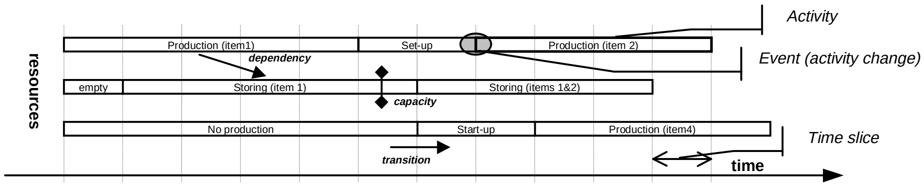

Figure 3 (Constraint models and constraint classification)

5 Conclusions

In the paper we propose a mixed planning and scheduling framework or, in other words, we show how to extend a conventional scheduler by some planning capabilities. We analyse three constraint models for scheduling and we sketch how these models can be modified to allow mixed planning and scheduling. We also describe the capabilities of these models when applied to complex-process environments.

Because the time-line model is too big for large-scale problems and the capabilities of task-centric model are limited, we chose the resource-centric model as the best model for complex process environments. First experiments in VisOpt project (VisOpt, 1999) confirm the modelling capabilities but they also showed that it is necessary to find balance between dynamics of constraints and constraint propagation.

The research was motivated by real-life problems in complex process environments but we believe that the results are applicable in other [non-production] areas as well.

6 References

Barták, R. (1999): On the Boundary of Planning and Scheduling: A Study, in Proceedings of PLANSIG99 Workshop, Manchester, UK

Nareyek, A. (2000): AI Planning in a Constraint Programming Framework, in Proceedings of the Third International Workshop on Communication-Based Systems (CBS-2000)

Pegman, M. (1998): Short Term Liquid Metal Scheduling, in: Proceedings of PAPPACT98 Conference, London

Srivastava, B., Kambhampati, S. (1999): Scaling up Planning by teasing out Resource Scheduling, Tech. Report ASU CSE TR 99-005, Arizona State University

VisOpt System web site (1999), http://www.visopt.com