Planning in Supply Chain Optimization Problem

N.H. Mohamed Radzi, Maria Fox and Derek Long
Department of Computer and Information Sciences
University of Strathclyde, UK

Abstract

The SCO planning problem is a tightly coupled planning and scheduling problem. We have identified some important features underlying this problem including the coordination between actions, maintaining temporal and numerical constraints and the optimization metric. These features have been modeled separately and experimented with the state-of-the-art planners, Crikey, Lpg-td and $SgPlan_5$. However, none of these planners are able to handle all features successfully. This indicates a new planning technology is required to solve the SCO planning problem. We intend to adopt Crikey as a basis of the new technology due to the capability of solving the tightly coupled planning and scheduling problem.

Introduction

The Supply Chain Optimization (SCO) covers decision making at every level and stage of a system that produces products for a customer. The foremost important issues include the decisions about the quantities of products to be produced, scheduling the production and delivery whilst minimizing utilization of resources by the system within a certain planning period. All these decisions require reasoning and planning: understanding the factors that are relevant to the decisions and evaluation of the combinatorial of the problem. This means the planning process in SCO is not only deciding which action should be chosen to reach the goal state based on the logical constraints but also what is the consequence of selecting the action to the given optimization function. Due to these features the planning problems in SCO are different from the standard planning problem. The SCO planning domains are richer in temporal structure than most temporal domains in standard planning.

Temporal domains were introduced in the third International Planning Competition (IPC3) along with the temporal planning language PDDL2.1 (Long & Fox 2003)(Fox & Long 2003). The durative action is introduced as a new feature in the language. This feature allows actions in domains to be allocated a unit of time specifying time taken to complete said action. (Weld 1994). Furthermore, the quality of the plan is also measured by the overall length or duration of the plan generated. The temporal features in the language were later extended by the introduction of timed initial literals in PDDL2.2 (Edelkamp & Hoffman 2003). This is the language used in IPC4. Timed initial literals provide a syntactically simple way of expressing the exogenous events that are both deterministic and unconditional (Cresswell & Coddington 2003). Another way to express exogenous events was then introduced in PDDL3.0 by using hard and soft constraints (Gerevini & Long 2006). Hard and soft constraints express that certain facts must be, or are preferred to be, true at a certain time point as benchmarked in IPC5 (Dimopoulos *et al.* 2006).

Temporal domains in IPC3 require certain facts to be true at the end of the planning period. Although domains with deadlines or exogenous events are modeled in IPC4 and 5, none of these domains require actions overlap in time. In contrast, SCO domains require some collections of facts to be true not only at a particular final state but also throughout the trajectory. For example, some quantities of a product may be required to be in production throughout the planning period. Add to that, the SCO problems also require that actions to be executed concurrently during the planning process. For instance, there are exogenous events such as order deadlines that have to be met. We have to maintain these deadlines and concurrently execute other production activities. Moreover, there might be some threshold values that have to be maintained over the planning period.

As well as temporal structure, SCO domains are also rich with numerical structure. The domains with numerical structure were also introduced in IPC3. But most of the competition domains in the IPCs mainly deal with the consumption of resources and cost. In the SCO problems, numerical facts and constraints are used to model beyond the consumption of resources and cost. The numerical facts and constraints are also used to model the multiple actions: actions that have equivalent chances of being selected but the difference between them lies in the cost associated with performing them. In sum, SCO problems are very complex planning problems where temporal and numerical con-

straints enforced over time must be met as well as the logical constraints.

From another point of view, SCO planning problem is different from standard planning problems in terms of the way plans are constructed. The standard or classical planning problems concentrates on a process to find *what* actions should be carried out in a constructed plan by reasoning about the consequence of acting in order to choose among a set of possible courses of action (Dean & Kambhampati 1997). The number of actions required in a plan is usually unknown. The temporal planning problem is basically a combination of classical planning and scheduling. In the pure scheduling process, the actions are usually known and the choice of actions is limited compared to planning (Smith, Frank, & Jonsson 2000). The scheduling process concentrates on figuring out *when* and *with what* resources to carry out so as satisfy various types of constraint on the order in which the actions need to be performed (Dean & Kambhampati 1997). Therefore in temporal planning, the process of constructing a plan combines the decisions on *what* actions should be applied, with *when* it should be applied and *with what* resources (Halsey 2004).

The SCO problem however, is an example of a combinatorial problem that has string planning, scheduling and constraint reasoning components. Besides *what* and *when* choices it also contain choices about *how* to act. One way to introduce a *how* choice is to differentiate actions for achieving the same effect by numerical values such as duration or resource consumption. The *what* choices concern what resources are required for an action to be applied and the *when* choices concern how the action should be scheduled in to the rest of the plan. A very good example of the problem is the following: a manufacturer receives several orders from customers that consist of producing various quantities of several different items. These orders should be delivered within specified deadlines. The manufacturer has to schedule the production of each item. Due to the capacity limitations of the producer, the manufacturer has to decide which items should be produced using his own facilities and which items should be produced using other production options that are available. No matter how, the deadlines have to be met and the overall production cost should be minimized. In this case, the solution is not as simple as performing a sequence of actions but could involve executing many actions concurrently.

We have discovered that, although there are a number of planners in the literature that are capable of handling the individual features of PDDL2.1 and PDDL3, there are no planners currently available that can reliably solve non-trivial problems.

The reminder of this paper is structured as follows. First we present a description of a simple domain within the class of problems. We have encoded the domain and applied several state-of-the-art planners to it. The outcomes of the experiment are discussed in the following section but, in brief, the best performing planners in IPC4 and IPC5 are unable to solve the problems we set. Clearly, SCO problems encompass a huge variety and would in general be beyond the reach of any automated planner. Therefore, this discussion is followed by the definition of a subclass of problems that we intend to focus on in our work. We will develop a planner (by enhancing an existing planning system) that is capable of solving this subclass of problems. Later, we briefly describe our future work including the planner that we intend to enhance.

Domain Definition

A simple example of production planning problem in the supply chain is illustrated in Figure 1. The process starts with receiving the customers orders. Each order has a different combination of products and also different delivery deadlines. The process is then followed by selecting the production types of each product. In our example, each production type has a different processing time and cost: *normal-time*, *over-time* and *outsource*. The outsource action furthermore can be performed by several suppliers where each supplier is associated with a different lead time and cost. The domain demonstrates the properties discussed in the above section. The choices of production action represent the multiple choice of actions for achieving the same task. These actions can be executed simultaneously as well in parallel with other activities. The probability of the action being selected is dependent on the objective function. Any plan produced by the planner should minimize the overall cost and time taken to produce all items as well as meeting the specified deadlines. For example, item_1 can be produced either by the *normal-time* action or the *outsource* action but, choosing the *normal-time* action might cause a delay in the product delivery so that it is better to choose the *outsource* action. In an efficient plan we might be producing item_2 while we are also producing item_1. This domain has been encoded and presented to some of state-of-the-art planners.

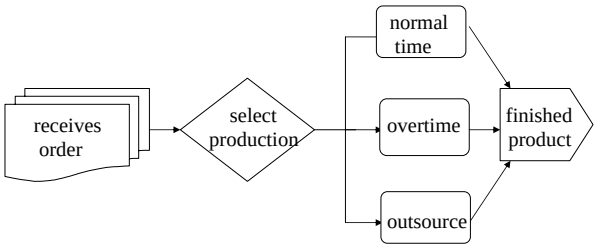

Figure 1: A Simple production process

State-of-the-Art Planners

We chose three different types of temporal planners for our experiments. All of these planners are claimed to be able to handle the temporal features of PDDL2.1 and also features to express deadline such as time windows and hard constraint. The planners are as follows:

SGPlan$_5$ is a temporal planner that received a prize for overall best performance in IPC5. The planner works with PDDL3.0, which features timed initial literals, hard constraints and preferences. It generates a plan by partitioning the planning problem into subproblems and finds a feasible plan for each sub-goal. The multi-value domain formulation (MDF) is used as a heuristic technique in the planner for resolving goal preferences and trajectory and temporal constraints (Chih-Wei *et al.* 2006).

The **LPG-td** planner is an extension of LPG (Gerevini & Serina 2002) that can handle features in PDDL2.1 and most features in PDDL2.2, including timed initial literals and derived predicates. The timed initial literals represent facts that become true or false at certain time points, independently of the actions in the plan. This feature can be used to model a range of temporal constraints including deadlines. LPG-td is an incremental planner that generates a plan in the domain involving maximization or minimization of complex plan metrics. An incremental process improves the plan by using the first generated plan to initialize a new search for a second plan of better quality and so on. It can be stopped at any time to give the best plan computed so far (Gerevini, Saetti, & Serina 2004).

CRIKEY is a temporal planner that solves a tightly coupled type of planning and scheduling problem. This planner supports PDDL2.1. It has implemented the envelope and content concept in order to handle the communication between the planning and scheduling. Content actions are executed within envelope actions. Therefore the minimum length of time for the content actions must be less than or equal to the maximum total length of time for the envelope actions (Halsey, Long, & Fox 2004). The envelope and content concepts were introduced to allow Crikey to solve problems in which actions must be executed in parallel in order to meet temporal and resource constraints.

Experimental Results

The aim of the experiments is to investigate the capability of each planner to cope with the following features:

(1) temporal constraints that require facts to be maintained over time; (2) optimization metrics including temporal and numerical optimization; (3) coordination and concurrent actions.

The domain described in the previous section was encoded using PDDL. There were six actions modelled in the domain including STACK_ORDER, CHOOSE_BRAND, OVERTIME, NORMAL_TIME, OUTSOURCE and SHIP_ON_TIME. Since different planners can work with different versions of PDDL, we have exploited PDDL2.1 features to represent domains presented to Crikey, PDDL2.2 for domains presented to Lpg-td and PDDL3.0 for domains presented to SgPlan$_5$. We have had to use different syntax to express the deadlines: timed initial literals for Lpg-td and hard constraints for SgPlan$_5$. No specific syntax is given in PDDL2.1 for expressing deadlines, but it is possible to encode them using envelope actions and clips (Fox, Long, & Halsey 2004; Cresswell & Coddington 2003). In the first experiment we have encoded only a single deadline. The encoded problem has been presented three times to each planner, each time with a different set-up. The problem instances are described in Table 1. For example in the first instance, the duration for actions NORMAL_TIME and OVERTIME are 7 and 8 unit time respectively. The OUTSOURCE action can be performed through either by $supplier_1$ or $supplier_2$ with the duration are 5 and 6 unit time respectively. The planners are expected to perform one of these actions in order to accommodate the deadlines. The duration of other actions defined in the domain is 1 unit time. Table 2 describes the deadlines and the plan duration given by each planner (if any) together with the action selected in the plan.

prob	normal-time	overtime	supplier$_1$	supplier$_2$
1	7	8	5	6
2	7	5	7	6
3	5	8	7	9

Table 1: problem instances set-up

prob	d	Crikey	SgPlan	Lpg-td
1	8.05	7.05 supplier$_1$	8.004 supplier$_1$	8.000 supplier$_1$
2	8.05	8.05 supplier$_2$	no solution	8.00 overtime
3	8.05	7.05 normal-time	no solution	8.00 normal-time

d: deadline

Table 2: maintaining time constraints by each planners

As we can see in Table 2, Crikey and Lpg-td planners perform very well in maintaining the temporal constraint. Both planners managed to obtain a plan with the most appropriate actions so that the completion time is within the deadline. But, SgPlan$_5$ only generates a plan for the first instance. There are no solutions for the second and third instances.

Later, the second experiments were carried out to see whether these planners can reason about the optimization metric, for example, minimize the makespan. For this purpose, the deadlines were excluded from the domains and then replaced with the minimization metric of *total-time*. The same encoded problems were applied to all planners. The description of the problem instances were remained the same as in the Table 1. Table 3 exhibits the completion time of the plan generated by each planner. We can see from this Table, Lpg-td was capable of minimizing the makespan compared to both SgPlan$_5$ and Crikey. SgPlan$_5$ as described in the Table 3 always choose the same action no matter the changes made in the duration of the actions

in the domain. Therefore in experiment 1, $SgPlan_5$ unable to produce any plan for $instance_2$ and $instance_3$ since the set up time for the particular action has violated the deadlines. Crikey also performs similar to $SgPlan_5$ in this experiment. As depicted in Table 3, NORMAL_TIME action is chosen regardless the duration set up for the action in each instance. The result from experiment 1 also indicates that Crikey will only maintain the temporal constraint by finding a feasible solution but not an optimal solution. Refer to Table 2 for $instance_2$. Although plan duration given by Crikey meeting the deadlines, but the duration is slightly bigger than plan duration generated by Lpg-td.

prob	Crikey	SgPlan	Lpg-td
1	9.004 normal-time	8.004 $supplier_1$	8.000 $supplier_1$
2	9.004 normal-time	10.004 $supplier_1$	8.000 overtime
3	7.004 normal-time	10.004 $supplier_1$	8.000 normal-time

Table 3: optimization metric: minimizing makespan

Besides minimizing the makespan, some experiments to investigate whether these planners can reason about numerical values by giving a plan that minimizes the total cost incurred due to action selection were carried out. The problems used in experiment 2 were applied in this experiment. But, the optimization metric was changed to minimize *total-cost* and the same durations were set to each action. The number of instances were also increased to ten, each instance has been set up with a different cost. The metric value of the plan is given in the generated plan for the plan produced by $SgPlan_5$ or Lpg-td. But for Crikey the metric value can be identified through the action selected in the plan. Table 4 shows the metric values or total cost obtained from the plan generated by each planner. Lpg-td produced plans that minimized the total cost for every instances. In some instances, either $SgPlan_5$ or Crikey also able to produce the optimized plans. The optimized plans were obtained due to the cost of the actions that are considered to be selected have the smallest cost compared to other actions in the problem. This is definitely not because of the capability of the planner to reason on the numerical values. The domains and problems involved in the experiment can be accessed at http://www.cis.strath.ac.uk/∼nor.

As mentioned in the previous section, the SCO contains choices about how to act. The choices of how to act affect the quality of solution as well as satisfiability of the schedule. We cannot simply perform the action selection first and later schedule the actions according to their temporal and numerical information. This means the planning and scheduling tasks are tightly coupled and cannot be performed separately. This situation requires coordination between actions and execution of the concurrent actions. Coordination is where the actions can happen together and interact with another. Meanwhile concurrency means more than one action happen simultaneously but they are not to interfere with each other (Halsey 2004). For example see Figure 2. There are three deadlines, denoted by x_1, x_2 and x_3. The x_2 and x_3 happen at the same time point. These deadlines x_1, x_2 and x_3 require actions (a_1,a_2,a_3), (a_1,a_4,a_5) and (a_1,a_4,a_6) respectively. Either some parts or all parts of the actions' durations are overlapped in time or executed in parallel. The actions a_2 and a_3 must interact with action a_1. These actions must execute during the life time of action a_1. But there is no interaction between a_2 and a_3. The actions are required to execute simultaneously in order to achieve the deadline. The actions a_4, a_5 and a_6 are also examples of coordination where action a_5 and a_6 are executed in some portion of the life time of a_4. Furthermore, the a_5 and a_6 actions demonstrate the choice of how to act. In achieving deadlines x_2 and x_3, either a_5 or a_6 has to be executed following a_4. Another clear example of a domain in which some actions must happen in parallel, which has been investigated in the previous literature, is the Match Domain (Halsey, Long, & Fox 2004). However, the choices on how to act is not demonstrated in this domain.

problem	Crikey	SgPlan	Lpg-td
1	9.00	11.00	3.20
2	7.00	4.00	3.20
3	9.00	11.00	4.20
4	12.00	11.00	7.20
5	12.00	5.00	5.20
6	4.00	5.00	4.20
7	7.00	12.00	7.20
8	20.00	15.00	13.20
9	20.00	9.00	9.20
10	4.00	9.00	4.20

Table 4: optimization metric: minimizing total-cost

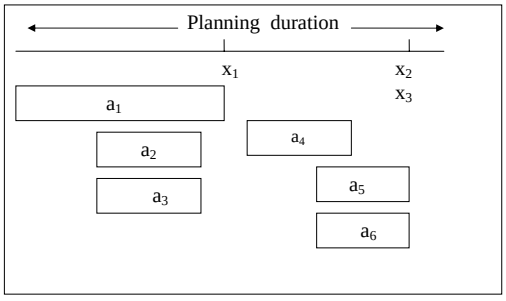

Figure 2: concurrent actions

Due to the importance of the above features in the SCO domain, we also investigate the capability of these planners to support these requirements. The coordina-

tion and concurrent features were indirectly performed in the temporal constraint problem to which Crikey was applied in experiment 1. In Crikey, actions are either *wrappers* or *contents* with wrappers containing contents and contents being completely contained within wrappers. Some content actions are also wrappers for other actions. In experiment 1, we have encoded deadlines as the wrapper actions. The other six actions decribed in the beginning of this section were the content actions. These content actions have to start after the wrapper start and end before the wrapper end. In other words, the wrapper and the content actions are performed in parallel. The wrapper and the content actions can be illustrated as actions (a_1,a_2,a_3) in Figure 2. Lpg-td and $SgPlan_5$ were then applied to the same domain, since both planners are capable of handing all PDDL2.1 features. Unfortunately, neither planner can solve the problem. As in Table 5, besides SCO domain, two other domains including the match domain were also tried. The driverlog-shift domain (Halsey 2004) is an extension of the driverlog domain used in IPC3. In this extended domain, the driver can only work for a certain amount of time or in a shift. The shift action is modelled as an envelope action. Therefore, driving and walking actions must be fitted into the shift action. $SgPlan_5$ produced a plan for this domain. But, in the plan, the walking and driving actions are performed after the shift action finished. In other words, they are performed in a sequence. $SgPlan_5$ and Lpg-td are able to perform concurrent actions, provided that the actions do not interfere with each other. This is as a result of both planners generating the temporal planning problem by finding out the sequential solution first and rescheduling them using temporal information.

domain	Crikey	SgPlan	Lpg-td
SCO	plan obtained	no solution	no solution
match	plan obtained	no solution	no solution
driverlog-shift	plan obtained	plan obtained	no solution

Table 5: domain with concurrent actions

Moreover, domains that are encoded with coordination or concurrent actions will have plans that shorten the makespan. Refer to Table 2. Although Crikey and Lpg-td choose the same action, the plan duration generated by each of the planner is different. The plan produced by Crikey has a shorter duration than the plan generated by Lpg-td.

The overall performance based on the criteria outlined or properties underlying in the SCO problems in the experiment are summarized in Table 6. Crikey is very good at maintaining constraints and coordination of tasks but very poor at metric optimization. Nevertheless for this problem, Crikey is still able to produce a feasible plan. Lpg-td, although it has a very good performance both in maintaining constraints and optimization, cannot perform coordination of actions. When this is required no plan can be produced at all. Although, $SgPlan_5$ can handle temporal constraints as benchmarked in IPC5, the domains involved do not include choices about how to act. An example arises in the truck domain. This domain only encodes what action should be carried out in order to meet the temporal constraints. $SgPlan_5$ seems unable to reason with choices about how to act. Therefore for some instances in experiment conducted, $SgPlan_5$ did not produce any plan. Unlike Lpg-td, $SgPlan_5$ is sometimes able to produce a plan for a concurrent domain but the execution of actions in the plan are performed in a sequenced manner.

planner	time constraint	optimization metric	coordination
Crikey	very good	poor	very good
Lpg-td	very good	very good	cannot performed
SgPlan	poor	poor	cannot performed

Table 6: overall performance of planners

Subclass of SCO problem

As discussed in the beginning of the paper, SCO is a hard combinatorial problem that requires not only reasoning about the logical relations between actions but also has to examine the temporal and numeric relations between actions. Since it is very hard to solve the overall problem features, only the subclass of this problem will be focused on in this research. The properties of the subclass problem are identified as follows. The very important properties are maintaining temporal and numerical constraints. The second feature is the optimization metric in term of numerical values. All these properties require coordination between actions as well as actions to be performed concurrently in the generated plan. Since planning problems have a strong scheduling element, we will have a selection of alternative actions (planning) within the large selection of actions described in the domain. This situation exhibits the how choices action in the domain. Within the alternative actions, there is also a selection of possible resources, giving rise to a scheduling problem. All these actions are weighted by numerical values representing their costs. At this stage we are not interested in optimization in term of temporal metrics.

Conclusion

This paper discusses the features of SCO planning problems and investigates the performance of state-of-the-art planners on domains with these features. We have

run the experiments on the individual features separately. The planners are expected to handle some of the features, such as minimization of *total-cost* or *total-time* metric as well as satisfying the hard constraints. However, as we can see, none of the state of the art planners we tried were able to successfully handle all the features. Therefore, experiments conducted to date have identified several improvements in the planning technology that are required in order to solve the SCO type of domain.

Future Work

In the near future we will develop a subclass of the SCO problem that combines all the features together. The more complex optimization metric will be included in the problem since the numerical features considered in the experiment so far are very simple. As numerical constraints are identified as one of the properties of the SCO subclass, the numerical constraints will also be included in the domain. The domain will be used to test a variety of planners. We plan to adopt Crikey as the basis of the new technology that we intend to develop. Crikey is chosen due to its ability to cleanly manage the tightly coupled interaction between planning and scheduling as well as other features such as duration inequalities and interesting metric optimisation.

References

Chih-Wei; Wah, B.; Ruoyun; and Chen, Y. 2006. Handling soft constraints and goal preferences in SGPLAN. In *ICAPS Workshop on Preferences and Soft Constraints in Planning*. ICAP.

Cresswell, S., and Coddington, A. 2003. Planning with timed literals and deadlines. In Porteous, J., ed., *Proceedings of the 22nd Workshop of the UK Planning and Scheduling Special Interest Group*, 22–35. University of Strathclyde. ISSN 1368-5708.

Dean, T., and Kambhampati, S. 1997. Planning and scheduling. In *The Computer Science and Engineering Handbook 1997*. CRC Press. 614–636.

Dimopoulos, Y.; Gerevini, A.; Haslum, P.; and Saetti, A. 2006. The benchmark domains of the deterministic part of IPC-5. In *Booklet of the 2006 Planning Competition, ICAPS'06*.

Edelkamp, S., and Hoffman, J. 2003. PDDL2.2: The language for the classical part of the 4th international planning competition.

Fox, M., and Long, D. 2003. An extension of PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20:61–124.

Fox, M.; Long, D.; and Halsey, K. 2004. An investigation into the expressive power of PDDL2.1. In *Proceedings of ECAI'04*.

Gerevini, A., and Long, D. 2006. Plan constraints and preferences in PDDL3. In *ICAPS Workshop on Preferences and Soft Constraints in Planning*. ICAPS.

Gerevini, A., and Serina, I. 2002. LPG: A planner based on local search for planning graphs. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling*.

Gerevini, A.; Saetti, A.; and Serina, I. 2004. Planning with numerical expressions in LPG. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04)*. IOS-Press, Valencia, Spain.

Halsey, K.; Long, D.; and Fox, M. 2004. CRIKEY - a planner looking at the integration of scheduling and planning. In *Proceedings of the Workshop on Integration Scheduling Into Planning at 13th International Conference on Automated Planning and Scheduling (ICAPS'03)*, 46–52.

Halsey, K. 2004. *CRIKEY! Its Co-ordination in Temporal Planning: Minimising Essential Planner–Scheduler Communication in Temporal Planning*. Ph.D. Dissertation, Ph. D. Dissertation, University of Durham.

Long, D., and Fox, M. 2003. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research* 20:1–59.

Smith, D. E.; Frank, J.; and Jonsson, A. 2000. Bridging the gap between planning and scheduling. *The Knowledge Engineering Review* 15:47–83.

Weld, D. S. 1994. An introduction to least commitment planning. *AI Magazine* 4.