Models of Robustness in Scheduling and Temporal Planning Based on Optimisation with Dynamic Controllability Constraints

Jing Cui and **Patrik Haslum (supervisor)**
ANU & DATA61
cui.jing|patrik.haslum@anu.edu.au

Introduction

Robustness is vital in some scheduling and temporal planning problems, such as evacuation planning (Even, Pillac, and Hentenryck 2014). However, it is hard to measure robustness directly, so some metrics measuring relative features have been introduced (e.g., Cesta, Oddi, and Smith 1998; Aloulou and Portmann 2003; Wilson et al. 2014), as well as algorithms for finding high-robustness schedules (e.g., Aloulou and Portmann 2003; Policella et al. 2009; Banerjee and Haslum 2011). The goal of my PhD thesis is to explore and compare existing measures that measure robustness or other features related to robustness, try to provide new robustness models and try to get more robust schedules or plans.

Robustness Measures

Among different robustness measures, we list two that are used in the following experiments.

Flexibility A partial-order schedule (POS) consists of a set of time constraints between activities such that any realisation that meets these constraints is also resource feasible. Flexibility (Aloulou and Portmann 2003) counts the number of pairs of activities that do not have any explicit or implicit precedence relations in a POS. The definition of $flex$ is

$$flex = \frac{|\{(a_i, a_j)|a_i \nprec a_j \wedge a_j \nprec a_i\}|}{n(n-1)} \tag{1}$$

where the precedence constraints between activities include both explicit and implicit relations.

Fluidity In order to take slacks into account when measuring robustness, Cesta, Oddi, and Smith (1998) introduced fluidity. It represents the ability to absorb temporal deviation. It is defined as

$$fldt = \sum_{i=1}^{n} \sum_{j=1 \wedge j \neq i}^{n} \frac{slack(a_i, a_j)}{H \times n \times (n-1)} \times 100 \tag{2}$$

where H is a fair bound which is large enough to allow all activities to be executed, and $slack(a_i, a_j)$ is the width of the allowed distance interval between two activities.

Dynamic Controllability of Simple Temporal Problem with Uncertainty

The simple (non-disjunctive) temporal problem with uncertainty, or STPU (Vidal and Fargier 1999), is a widely used model for representing schedules or temporal plans that have both uncertainty about the timing of some events (for example, the time needed to complete an activity) and flexibility for the executing agent to choose the timing of other events (for example, the time to start an activity).

Formally, an STPU consists of a set of nodes $X = X_E \cup X_U$, representing executable (X_E) and uncontrollable (X_U) time points, and a set of links $E = R \cup C$, called *requirement* and *contingent* links. Each link e_{ij} has a lower bound L_{ij} and upper bound U_{ij}, representing the constraints $L_{ij} \leq t_j - t_i \leq U_{ij}$.

Because of the uncertainty in STPU, 3-level controllability is introduced (Vidal and Fargier 1999), among which **dynamic controllability** is most interesting. An STPU is dynamically controllable when there is a dynamic strategy that schedule current controllable timepoints according to observations of the past contingent links. Different dynamic controllability checking algorithm are introduced (Morris, Muscettola, and Vidal 2001; Shah et al. 2007; Nilsson, Kvarnström, and Doherty 2013; Combi, Hunsberger, and Posenato 2014; Morris 2014) .

Besides checking dynamic controllability, optimising STPU with constraints representing dynamic controllability enables different applications, such as introducing robustness measures. Those robustness metrics can answer the question what the worst, best or average case is, under which the schedule or temporal plan with uncertainty is still dynamically controllable.

Controllable Conditional Temporal Problem with Uncertainty

The concept of CCTPU extends from Conditional Temporal Problem (CTP) (Tsamardino, Vidal, and Pollack 2003) and STPU. CTP is an extension of temporal constraint-satisfaction problem by adding observation nodes and labels to all non-observation nodes in the network. The label of a node in CTP represents the situations in which the node will be executed. In (Yu, Fang, and Williams 2014), a relaxation method is introduced to solve over-constrained

CCTPUs and achieve dynamically controllable solutions by implementing dynamic controllability checking algorithms (Morris, Muscettola, and Vidal 2001; Morris 2014) to find conflicts. Thus the solutions consist of a set of static choices and dynamically controllable bounds of the network associating with the choices. However, the choices are still made before execution which can be postponed to achieve a more flexible and dynamic strategy.

In our work, we attempt to define dynamic controllability of CCTPU. However, due to the difficulty to project discrete decisions to continuous timeline, we introduce conservative assumptions. Based on these assumptions, we introduce dynamic controllability of CCTPU and an approach to check it. Additionally, we are trying to formulate optimisation model with constraints of dynamic controllability of CCTPU, which may provide robustness measures measuring robustness of both temporal scheduling and discrete options.

Optimising STPU with Dynamic Controllability

The problem of optimising time bounds under dynamic controllability was previously considered by Wah and Xin (2004), who formulated a non-linear constraint optimisation model. In fact, dynamic controllability is a disjunctive linear constraint, and using this insight we consider several alternative ways of dealing with it, including a conflict-driven search (Yu, Fang, and Williams 2014), a formulation as a mixed-integer linear program with 0/1 variables, and the non-linear encoding proposed by Wah and Xin. (This work has been published in ICAPS 2015.)

Optimisation Model

The general form of the optimisation problem can be stated as follows: We are given the structure of an STPU, that is, the set of time points $X = X_E \cup X_U$ and links $E = R \cup C$, but not the upper and lower bounds on (all) links, and an objective function. The problem is then to set those bounds so as to optimise the objective function value:

$$\begin{array}{ll} \text{opt} & f_{obj}(l_{ij}, u_{ij} \mid e_{ij} \in E) \\ \text{s.t.} & L_{ij} \leq l_{ij} \leq u_{ij} \leq U_{ij} \\ & N(l_{ij}, u_{ij} \mid e_{ij} \in E) \text{ is dynamically controllable} \\ & \text{application-specific side constraints} \end{array}$$

The decision variables, l_{ij} and u_{ij}, represent the lower and upper bounds on link e_{ij}. Thus, a satisfying assignment defines an STPU, $N(l_{ij}, u_{ij} \mid e_{ij} \in E)$, and this STPU must be dynamically controllable.

Dynamic Controllability Constraints

The formulation of dynamic controllability is set of disjunctive linear constraints. It consists of shortest path constraints, precede constraints, and wait constraints. The formulation follows the dynamic controllability reductions rules in (Morris, Muscettola, and Vidal 2001), which considers reductions within a triangle consisting a contingent link. In a simple formulation, we can consider all triangles, however, in a reduced formulation, some constraints can be ignored since they are implied by other constraints (Wah and Xin 2004).

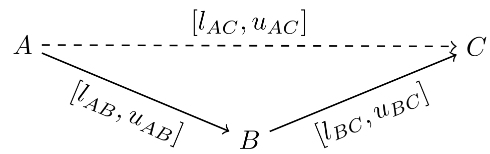


Figure 1: An STPU triangle. The A–C link is contingent.

In the dynamic controllability reductions rules, for every triangle as figure 1 with one contingent link, the schedule of timepoint B can be classified into precede, follow and unorder cases according to the bounds of BC. Because of the space limitation, We only illustrates constraints of precede case as an example here. If $L_{BC} \geq 0$, then $l_{BC} \geq 0$ and the triangle will be in the precede case. The following constraints must hold:

$$\begin{array}{lcl} u_{AB} & \leq & l_{AC} - l_{BC} \\ l_{AB} & \geq & u_{AC} - u_{BC} \end{array} \quad (3)$$

If the loose bounds are in unorder case, $L_{BC} < 0$ and $U_{BC} \geq 0$, the triangle can be in any case, depending on the values given to l_{BC} and u_{BC}. The precede constraint then becomes disjunctive:

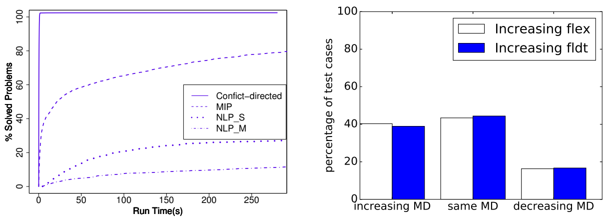
$$(l_{BC} < 0) \quad \vee \quad \left(\begin{array}{lcl} u_{AB} & \leq & l_{AC} - l_{BC} \\ l_{AB} & \geq & u_{AC} - u_{BC} \end{array} \right) \quad (4)$$

Other reduction rules can be formulated as disjunctive linear constraints in the same way. This disjunctive linear constraints model can be encoded into Mixed Integer Programming (MIP) or Non-Linear Programming (NLP) models, which can be solved by existing solvers.

Robustness with Non-Probabilistic Uncertainty

In abstract terms we may define robustness as the greatest level of disturbance (deviation from expected outcomes) at which the schedule is still successfully executed. Here, we exemplify by assuming (1) that the possible disturbances are deviations in the time taken to execute an activity from its normal duration, and (2) a partial-order schedule with a dynamic execution strategy.

In the deterministic case, where the duration of each activity i is a constant d_i, the POS can be represented as an STN with time points t_{s_i} and t_{e_i} for the start and end, respectively, of each activity. Assuming the duration of each activity can vary within some bounds, $[l_{s_i,e_i}, u_{s_i,e_i}]$, the schedule can be modelled as an STPU where the link $e_{s_i e_i}$ from each activity's start to its end is contingent, while remaining time constraints are requirement links. Thus, given a POS we can ask, what is the maximum deviation (i.e., width of the contingent bound) on any activity at which the STPU is dynamically controllable. This defines our measure of robustness.



(a) (b)

Figure 2: Result

To compute it, we solve the following problem:

$$\begin{array}{lll} \max & \Delta & \\ \text{s.t.} & l_{s_i,e_i} = d_i - \delta_i \geq 0 & \forall i \\ & u_{s_i,e_i} = d_i + \delta_i & \forall i \\ & 0 \leq \Delta \leq \delta_i & \forall i \\ & \text{POS constraints (requirement links)} & \\ & \text{dynamic controllability} & \end{array}$$

We can also define a one-sided variant of this robustness metric, accounting for delays only, by fixing $l_{s_i,e_i} = d_i$ (i.e., adding deviations only to the upper bound).

Result We compared solvers on the one-sided (maximum delay) variant of the problem. As test cases, we use 3400 partial-order schedules for RCPSP/max problems (e.g., Kolisch and Padman 2001) with 10–18 jobs[1] The schedules are generated by a scheduler that optimises a measure of POS flexibility (Banerjee and Haslum 2011). The number of (given) requirement links varies from 50 to 300.

The adapted CDRU algorithm is very effective for this problem, and the relative runtimes of the MIP and non-linear solvers, as shown in Figure 2 (a), which are runtime distributions for three different solvers (conflict-directed relaxation (CDRU), the MIP model solved with Gurobi and the non-linear model solved with SNOPT) on schedule robustness (maximum delay) problems.

We also compare this robustness measure with other two metrics: flexibility and fluidity. The scheduler can generate several POS with increasing flexibility (or fluidity) for one problem. However, in some problems, the solution has the best flexibility does not has the best maximum delay. Detailed result is in Figure 2 (b), around 16% pairs of POS generated from the same problem with increasing flexibility (or fluidity) have decreasing MD, which means the POS with higher flexibility (or fluidity) in those problems are easier to fail when uncontrollable events increasing delays evenly.

CCTPU with Dynamic Controllability

The Controllable Conditional Temporal Problem with Uncertainty (CCTPU) extends the STPU with controllable discrete choices. It was introduced by Yu, Fang, and Williams (2014). We adopt their definition, but omit the reward and cost functions since we consider feasibility only.

[1]Set J10 from PSPLIB (`http://www.om-db.wi.tum.de/psplib/`), plus additional problems generated to have more max time lag constraints.

Definition 1. A **Controllable Conditional Temporal Problem with Uncertainty (CCTPU)** is a 5-tuple $< V, E, C, D, \ell_E >$, where

- V is the set of time points, where $V = VC \cup VU$ and $VU = V \setminus VC$. VC is the set of controllable time points, VU is the set of uncontrollable time points which can be observed,
- E is the set of constraints of form $l_{ij} \leq v_j - v_i \leq u_{ij}$, where $E = EC \cup EU$ and $EU = E \setminus EC$. EC is the set of controllable constraints between pairs of time points, EU is a set of uncontrollable constraints, denoted as contingent links, the exact duration of eu_{ij} is not controllable but within the range $[l_{ij}, u_{ij}]$,
- C is a set of controllable discrete variables,
- $D(c)$ is the domain of variable $c \in C$,
- ℓ_E is a mapping that attaches to each link in E a (possibly empty) conjunction of assignments of values to variables in C.

Definition 2. A CCTPU is **dynamically controllable** if there is a viable execution strategy $\langle DT, ES \rangle$ such that for any two projections p_1 and p_2, $ES(p_1)\{\prec t\} = ES(p_2)\{\prec t\} \Rightarrow ES(p_1)(x) = ES(p_2)(x)$, $t = ES(p_1)(x)$ for each controllable time point x and $ES(p_1)\{\prec DT(c)\} = ES(p_2)\{\prec DT(c)\} \Rightarrow ES(p_1)(c) = ES(p_2)(c)$ for each each discrete variable c and its decision timepoint $DT(c)$.

Our assumptions are: (1) $A(c)$ is only made once at $DT(c)$ and (2) $DT(c)$ associates to a node no later than any links can be activated by $A(c)$.

Central to our algorithm for finding a dynamic execution strategy is the notion of the "envelope" of a partial assignment.

Definition 3. Given a partial assignment to a subset of discrete variables, $C_{Ass} \subseteq C$, the **dynamically controllable envelope** of an unassigned variable, $c \in (C - C_{Ass})$, is the set of prehistories of c for which there exists a viable dynamic execution strategy.

Approach

The structure of dynamic controllability checking for CCTPU is a tree search as figure 3. Each leaf of the search tree is the STPU obtained from a full assignments to discrete variables and other nodes are CCTPUs with partial assignments. Besides the root which is the original CCTPU, each node has one parent node that eliminates the assignment to the "latest" variable. The chronological order of variables is given in the next subsection.

From the root, the algorithm branches by assigning variables in chronological order and traverses the tree depth-first. When arriving at a leaf, it extracts the conflict resolution constraints that must be satisfied to make the leaf dynamically controllable and records those as the DC envelope. When more than one child branch of a node have been explored, their DC envelopes are combined and recorded as the envelope of current node. The CCTPU is dynamically controllable when a dynamically controllable node is found. The detailed algorithm is in our incoming ICAPS paper.

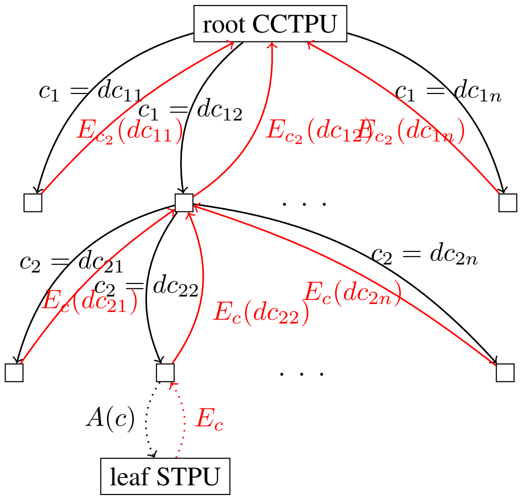


Figure 3: Expanding & Combining Tree of DC Checking for CCTPU

Future Work

In the future work, we will try to remove the current assumptions, which could allow a more dynamic strategy. Another possible extension is to see how much improvement can be made in solving optimisation problems of CCTPU when considering making choices dynamically. With the optimisation model, we can provide other robustness measures that can measure both temporal robustness and dynamic options. Additionally, some other constraints such as resource constraints in scheduling problem can be represented as alternations of time constraints (Banerjee and Haslum 2011). Therefore, more robust schedules can be achieved by applying those optimising robust measures on those schedulers.

References

Aloulou, M. A., and Portmann, M.-C. 2003. An efficient proactive reactive scheduling approach to hedge against shop floor disturbances. In *Proc. 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)*, 337–362.

Banerjee, D., and Haslum, P. 2011. Partial-order support-link scheduling. In *Proc. 21st International Conference on Automated Planning and Scheduling (ICAPS)*, 307–310.

Cesta, A.; Oddi, A.; and Smith, S. F. 1998. Profile-based algorithms to solve multiple capacitated metric scheduling problems. In *Proc. 4th International Conference on Artificial Intelligence Planning and Scheduling (AIPS)*, 214–223.

Combi, C.; Hunsberger, L.; and Posenato, R. 2014. An algorithm for checking the dynamic controllability of a conditional simple temporal network with uncertainty - revisited. In *Proc. 5th International Conference on Agents and Artificial Intelligence (ICAART)*, 314–331.

Even, C.; Pillac, V.; and Hentenryck, P. V. 2014. Nicta evacuation planner: Actionable evacuation plans with contraflows. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, ECAI'14, 1143–1148. Amsterdam, The Netherlands, The Netherlands: IOS Press.

Kolisch, R., and Padman, R. 2001. An integrated survey of project scheduling. *OMEGA International Journal of Management Science* 29(3):249–272.

Morris, P.; Muscettola, N.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *Proc. 17th International Conference on Artificial Intelligence (IJCAI)*, 494–499.

Morris, P. 2014. Dynamic controllability and dispatchability relationships. In *Proc. 11th Integration of AI and OR Techniques in Constraint Programming (CPAIOR)*, 464–479.

Nilsson, M.; Kvarnström, J.; and Doherty, P. 2013. Incremental dynamic controllability revisited. In *Proceedings of the 13th International Conference on International Conference on Automated Planning and Scheduling (ICAPS)*, 337–341.

Policella, N.; Cesta, A.; Oddi, A.; and Smith, S. 2009. Solve-and-robustify. *Journal of Scheduling* 12:299–314.

Shah, J. A.; Stedl, J.; Williams, B. C.; and Robertson, P. 2007. A fast incremental algorithm for maintaining dispatchability of partially controllable plans. In *Proceedings of the Seventeenth International Conference on International Conference on Automated Planning and Scheduling (ICAPS)*, 296–303.

Tsamardino, I.; Vidal, T.; and Pollack, M. E. 2003. CTP: A new constraint-based formalism for conditional, temporal planning. *Constraints* 8(4):365–388.

Vidal, T., and Fargier, H. 1999. Handling contingency in temporal constraint networks: From consistency to controllabilities. *Journal of Experimental and Theoretical AI* 11(1):23–45.

Wah, B. W., and Xin, D. 2004. Optimization of bounds in temporal flexible planning with dynamic controllability. In *Proc. 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 40–48.

Wilson, M.; Klos, T.; Witteveen, C.; and Huisman, B. 2014. Flexibility and decoupling in simple temporal networks. *Artificial Intelligence* 214:26–44.

Yu, P.; Fang, C.; and Williams, B. C. 2014. Resolving uncontrollable conditional temporal problems using continuous relaxations. In *Proc. 24th International Conference on Automated Planning and Scheduling (ICAPS)*, 341–349.