CONSTRAINED FORMULATION OF TEMPORAL FLEXIBLE PLANNING WITH DYNAMIC CONTROLLABILITY

Benjamin W. Wah and Dong Xin

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
http://manip.crhc.uiuc.edu

International Conference on Tools for Artificial Intelligence, 2004

11/15/2004

Outline

- Introduction
- Dynamic Controllability
- Constrained Formulation
- Reduced Formulation
- Experimental Results

Simple Temporal Network under Uncertainty

Temporal flexible planning problem

- Nondeterministic events (from nature) that govern actions
- Planner develops plans to respond to contingent events and satisfies requirements
- Example: Planner in Mars rover must replan when rover encounters obstacle

Simple temporal network with uncertainty

- Contingent link: causal processes of uncertain duration and controlled by nature
- Requirement link: processes controlled by planner

A STNU is a 5-tuple $\Gamma =< V, E, L, U, C >$

- V: set of nodes
- E: set of links
- $L : E \rightarrow \mathcal{R} \cup \{-\infty\}$, the lower bounds
- $U : E \rightarrow \mathcal{R} \cup \{+\infty\}$, the upper bounds
- C: subset of contingent links; $E \backslash C$: subset of requirement links

Controllability

STNU is controllable if

- Plan exists that satisfy requirements in all situations involving contingent events

Strong Controllability

- Actions schedulable under all possible times of contingent events

Weak Controllability

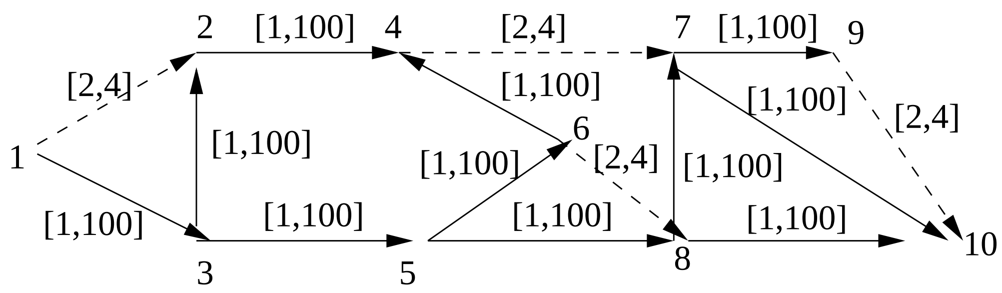
- Actions schedulable under all possible times of contingent events if those times were specified *a priori*

Dynamic Controllability

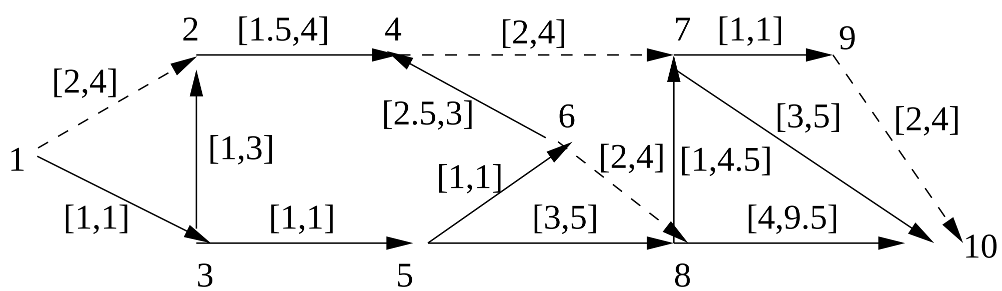
- Remaining actions in STNU schedulable under all possible future times of contingent events when all past contingent events are known

Strong controllability $\Longrightarrow$ Dynamic controllability $\Longrightarrow$ Weak controllability

Example: Dynamically Controllable STNU



a) Original dynamically controllable STNU with loose bounds



b) Dynamically controllable STNU with optimized bounds

Problem Statement

Given

- STNU: $\Gamma =< V, E, L, U, C >$ where $E = \{e_i\}$
- Cost function: $f : l(e_1), u(e_1), \ldots, l(e_n), u(e_n) \rightarrow \mathcal{R}$
- Desired bounds $l(e) : e \in E \rightarrow [L(e), U(e)]$ and $u(e) : e \in E \rightarrow [L(e), U(e)]$

Problem P_{stnu}:

$$\begin{array}{rll} \min & f(l(e_1), u(e_1), \ldots, l(e_n), u(e_n)) & E = \{e_i\} \\ \text{subject to} & L(e) \le l(e) \le u(e) \le U(e), & e \in E \backslash C \\ & l(e) = L(e), u(e) = U(e), & e \in C \\ \text{and} & \Gamma' =< V, E, l, u, C > \text{ is dynamically controllable} & \end{array}$$

Example: Minimize the time interval a camera is turned on on board a satellite

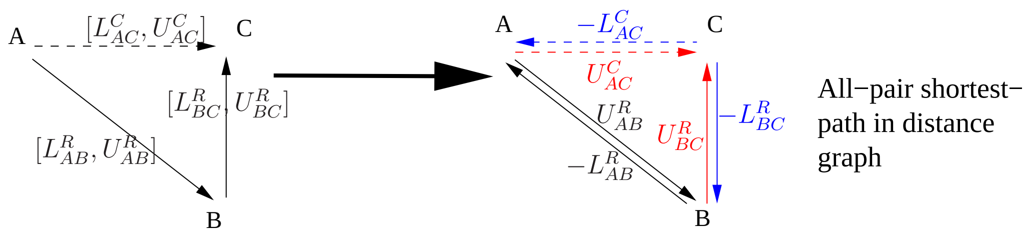
Problem P_{stnu} is NP-hard

- Construct P_{stnu} from $G = (V, E)$:
  - Introduce source node s
  - For each node $v_i \in V$, construct link (s, v_i) in STNU with unary part (positive integer times)
  - For each edge $(v_i, v_j) \in E$, construct link (v_i, v_j) in STNU with binary part (positive/negative integer times)
  - Cost function is the sum of all cost function of mutual exclusion parts
- Solution of P_{stnu} maps to solution of the 3-coloring problem
  - Cost function has minimum value 0
  - Each mutual exclusion part is dynamically controllable
  - Entire STNU is consistent

Outline

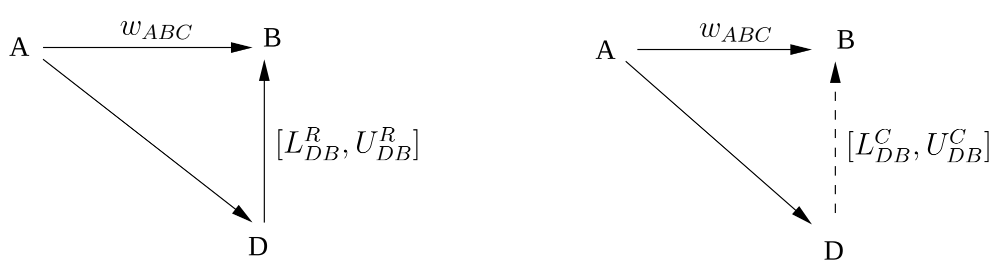
- Introduction
- Dynamic Controllability
- Constrained Formulation
- Reduced Formulation
- Experimental Results

Local Dynamic Controllability



- Precede case: $L^R_{BC} \geq 0$ and $U^R_{BC} > 0$
  - B must occur before or simultaneously with C because the information on the occurrence of C is not available to B when B is scheduled
  - AB must be tightened to $[U^C_{AC} - U^R_{BC}, L^C_{AC} - L^R_{BC}]$
- Unordered case: $L^R_{BC} < 0$ and $U^R_{BC} \geq 0$
  - B can occur before or after C has occurred
  - Triangle wait on AB for AC: $< C, w_{ABC} >$ where $w_{ABC} = U^C_{AC} - U^R_{BC}$
    - At any time before w_{ABC}, B cannot occur until C has occurred
    - At any time after w_{ABC}, B can occur independent of C

Global Dynamic Controllability



Propagating wait information throughout a network by regression

- Regression wait: regressing $< C, w_{ABC} >$ for a wait from AB to AD
  - DB is requirement link: wait regressed is $< C, w_{ABC} - U^R_{DB} >$
  - DB is contingent link and $w_{ABC} \geq 0$: wait regressed is $< C, w_{ABC} - L^C_{DB} >$
- $< C, w_{ABD} >$ on AD will be the maximum of its regression and triangle waits

STNU is dynamically controllable iff $U^R_r \geq L^R_r$ for every requirement link, and $[L^C_c, U^R_c]$ has not been tightened for any contingent link [Morris, et al., 2001]

Outline

- Introduction
- Dynamic Controllability
- Constrained Formulation
- Reduced Formulation
- Experimental Results

Naive Formulation

- Generate precede constraints and wait-bound constraints (triangle wait, regression wait, wait-bound constraint) according to each contingent link
- Consider every link as a requirement link and formulate constraints to ensure every bound to be the shortest path in the corresponding distance graph
- Conditions for different cases are incorporated as constraints

Complexity for an N-node STNU with C contingent links

Type of Constraints	#Const.	Type of Var.	#Variables
Shortest-Path	$\mathcal{O}(N^3)$	Bound	$\mathcal{O}(N^2)$
Precede	$\mathcal{O}(CN)$	Wait	$\mathcal{O}(CN)$
Triangular-Wait	$\mathcal{O}(CN)$	Auxiliary	$\mathcal{O}(C^2 + CN)$
Regression-Wait	$\mathcal{O}(CN^2)$		
Wait-Bound	$\mathcal{O}(CN)$		

Outline

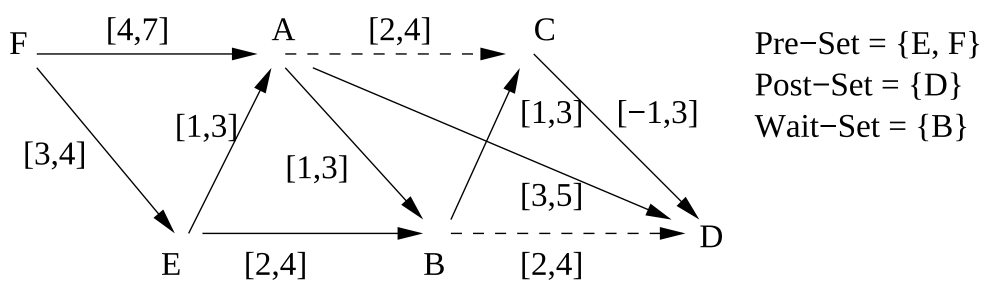
- Introduction
- Dynamic Controllability
- Constrained Formulation
- Reduced Formulation
- Experimental Results

Reduced Formulation

- Examine temporal order among nodes and links
- Analyze relationship among constraints to avoid generating implied constraints
  - Some wait-bound constraints are implied by shortest-path, precede and other wait-bound constraints
  - Some precede constraints are implied by shortest-path and other precede constraints
  - Some shortest-path constraints are implied by other shortest-path constraints
- Reductions do not introduce new constraints
- Order of reduction
  - Find wait-bound constraints not implied by others
  - Find precede constraints not implied by others
  - Formulate shortest-path constraints in the updated network

Reductions on Wait-Bound Constraints

- Two important facts (if the shortest-path constraints are satisfied)
  - The triangle wait satisfies the wait-bound constraints
  - The regression of a triangle wait through a requirement link will not raise the target triangle wait
- Given contingent link AC, classify every node B into:
  - Pre Set: $\{B \text{ such that } U_{AB} \leq L^C_{AC}\}$
  - Post Set: $\{B \text{ such that } L_{AB} \geq L^C_{AC}\}$
  - Wait Set: Otherwise.



Reductions on Wait-Bound Constraints: Post Set

- Post Migration: move E and F from Post Set to Wait Set
  - B is a finishing point of a contingent link EF
  - P is node C or any starting point of a contingent link in Wait Set where $L_{PF} < 0$
- If the shortest-path and precede constraints are satisfied, we can exclude the Post Set in wait-bound constraints
  - Triangle wait in Post Set satisfies bound constraints (by definition of Post Set)
  - Regression of triangle waits in Post Set will not raise other waits
  - If waits in Wait Set satisfy the bound constraints, so does their regression to Post Set
  - The regression of waits in Pre Set will not raise the waits in Post Set

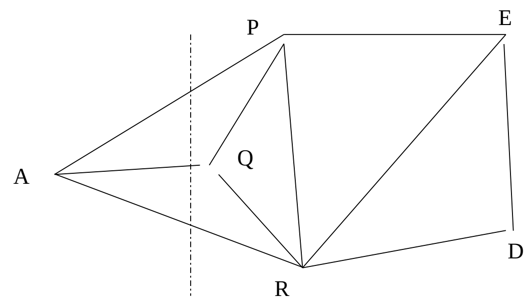
Reductions on Wait-Bound Constraints: Pre Set

- Pre Migration: move E and F from Pre Set to Wait Set
  - F is the finish point of contingent link EF where $U_{AF} \leq L_{AC}$
- Guard Migration: move F from Pre Set to Wait Set
  - For any path starting from C, the first node in Pre Set is F
- If shortest-path and precede constraints are satisfied, we can exclude Pre Set in wait-bound constraints
  - Triangles wait in Pre Set satisfies wait-bound constraints (by definition and guard migration)
  - Regression wait out of Pre Set will not raise the wait in Pre Set (by guard migration)
  - Regression of wait in Pre Set will not raise other waits (by pre migration)

Reductions on Precede Constraints

- Given contingent link AC, classify every node B into:
  - Post Set: $\{B \text{ such that } L_{CB} \geq 0\}$
  - Pre Set: $\{B \text{ such that } U_{CB} < 0\}$
  - Unordered Set: Otherwise.
- Reduction (similar with Pre Set of wait)
  - Post set: no constraints
  - Pre Set: Move guard nodes to unordered set
  - Unordered Set: Formulate constraints

Reductions on Shortest-Path Constraints



- For any node A, nodes adjacent to A constitute the Adjacent Set
- Formulate shortest-path constraints in triangles formed by A and any two nodes in its adjacent set
- Remove A from network, and consider the remaining as a new network
- Recursively decompose nodes

Outline

- Introduction
- Dynamic Controllability
- Constrained Formulation
- Reduced Formulation
- Experimental Results

Experimental Results

Complexity of test networks generated and their solution times of SNOPT in seconds on the NEOS Server

STNU	Topology				Naive Formulation				Reduced Formulation			
Nodes	Layers	Height	Links	Ctg.	Var.	Const.	NL	Time	Var.	Const.	NL	Time
41	20	2	60	12	2099	85630	51	-	276	1438	51	1.194
41	10	4	70	8	1971	79642	102	-	502	3972	102	16.066
41	8	5	72	6	1923	77416	114	-	574	5224	114	19.993
81	40	2	120	25	8436	681622	106	-	609	3418	106	3.110
81	20	4	140	17	7865	631993	215	-	1081	8810	215	36.237
81	16	5	144	14	7672	615368	243	-	1265	12088	243	128.255
121	60	2	180	36	18826	2274089	156	-	881	4755	156	8.212
121	30	4	210	26	17727	2133019	334	-	1698	14317	334	100.734
121	24	5	216	22	17219	2069761	369	-	1977	19266	369	251.766
161	80	2	240	52	33992	5465487	224	-	1216	6643	224	13.276
161	40	4	280	33	31121	4989835	417	-	2219	18608	417	154.296
161	32	5	288	28	30380	4867119	476	-	2632	25812	476	486.370
201	100	2	300	63	52772	10596461	273	-	1508	8223	273	19.260
201	50	4	350	42	48783	9772733	537	-	2814	23622	537	221.225
201	40	5	360	37	47660	9540734	617	-	3319	32478	617	837.102

Conclusions

- Constrained formulation for finding the bounds on requirement links of STNU
- Optimization problem is NP hard
- Naive formulations are prohibitively large
- Reduced formulations can be solved in reasonable time