Achieving Consistency with Cutting Planes

John Hooker
Carnegie Mellon University

Joint work with
Danial Davarnia & Atafeh Rajabalizadeh
Iowa State University

INFORMS 2021

A Different Perspective on IP

- The concept of **consistency** from constraint programming can provide a **new perspective** on **cutting planes**.
  - We can view cutting planes as excluding **infeasible partial assignments** rather than fractional LP solutions.
    - A partial assignment assigns integer values to only **some** of the variables.

A Different Perspective on IP

- The concept of **consistency** from constraint programming can provide a **new perspective** on **cutting planes**.
  - We can view cutting planes as excluding **infeasible partial assignments** rather than fractional LP solutions.
    - A partial assignment assigns integer values to only **some** of the variables.
  - Cutting planes can reduce backtracking even when there **are no bounds from an LP relaxation**.
    - This could have **computational** implications.
    - …and provide additional insight into IP.

Consistency

- **Consistency** is a core concept of constraint programming.
  - Roughly speaking,

Constraint set is **consistent**	=	**Partial assignments** that violate no single constraint are **feasible** (are part of some feasible solution)

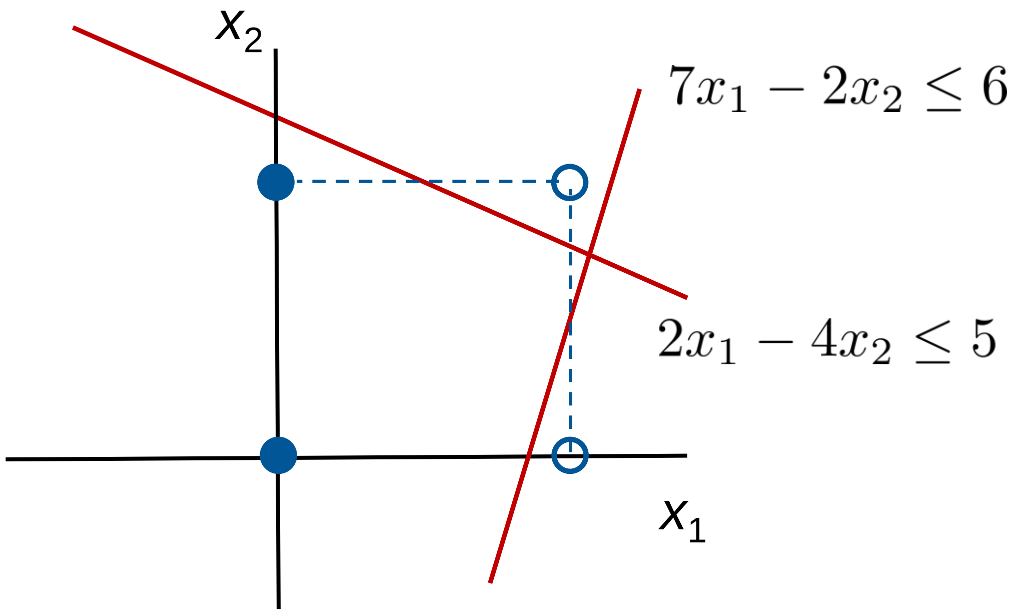
  - Consistency $\Rightarrow$ **no backtracking**
    - A **node** in a branching tree corresponds to a **partial assignment**.
    - If it **violates no constraint**, we can proceed to a **feasible solution** without backtracking.

Consistency

The constraint set

$$2x_1 + 4x_2 \le 5$$
$$7x_1 - 2x_2 \le 6$$
$$x_1, x_2 \in \{0, 1\}$$

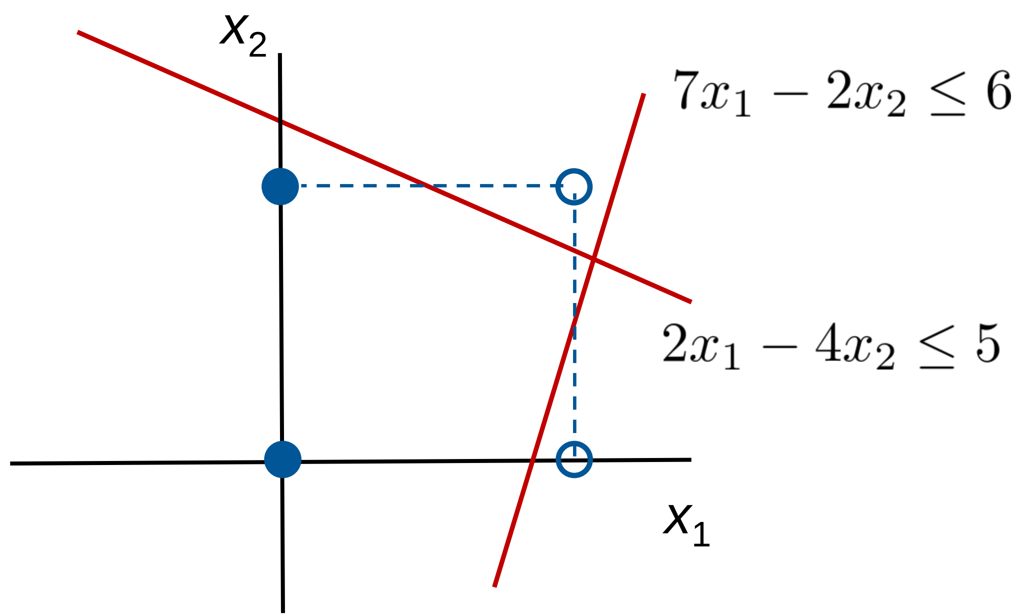
is **not consistent** because the partial assignment $x_1 = 1$ violates no single constraint* but is infeasible.



Consistency is a **much stronger** condition on a constraint set than feasibility.

*A partial assignment must fix all variables in a constraint to violate it

Consistency

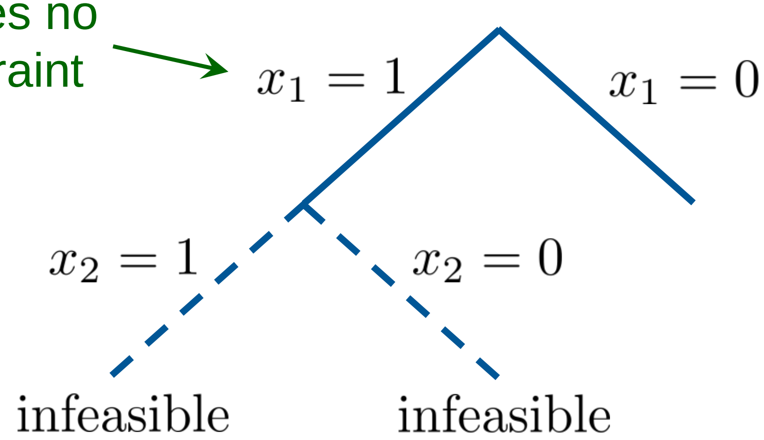


The constraint set

$$2x_1 + 4x_2 \leq 5$$
$$7x_1 - 2x_2 \leq 6$$
$$x_1, x_2 \in \{0, 1\}$$

is **not consistent**

Violates no constraint



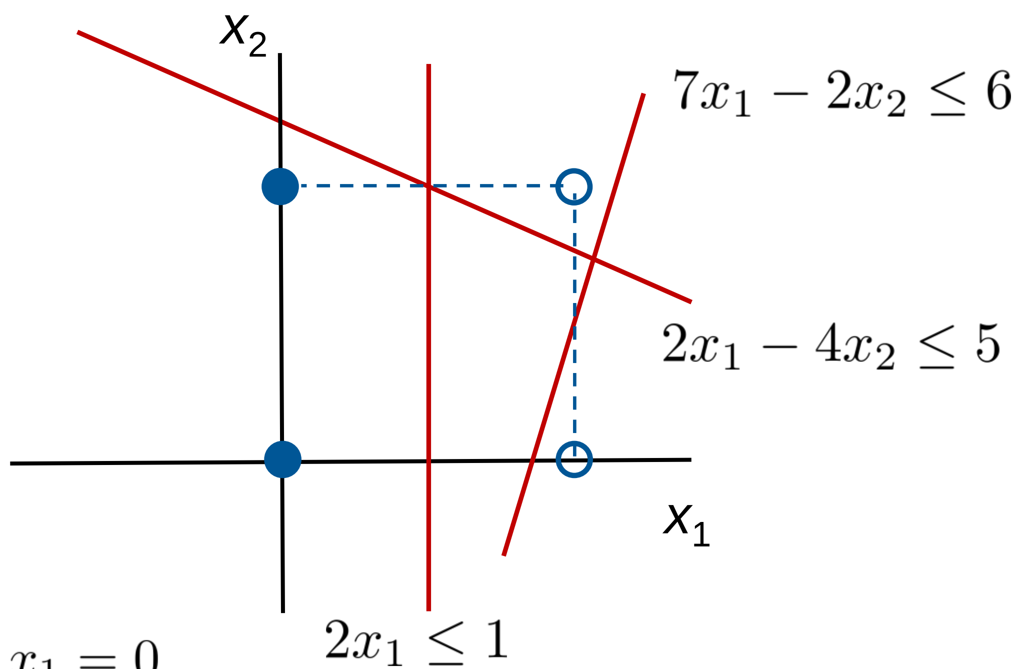
Backtracking can result even with 1-step lookahead (forward checking).

Consistency

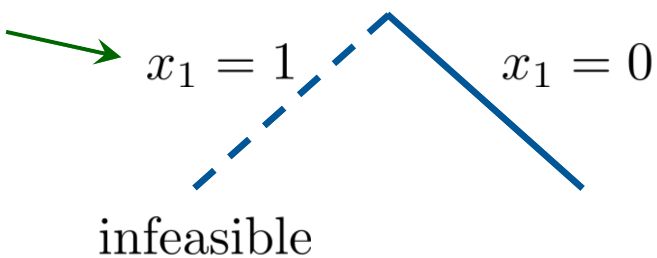
The constraint set

$$2x_1 + 4x_2 \leq 5$$
$$7x_1 - 2x_2 \leq 6$$
$$\boxed{2x_1 \leq 1}$$
$$x_1, x_2 \in \{0, 1\}$$

is consistent



Violates a constraint



No backtracking
with forward checking

Don't take the $x_1 = 1$ branch

Consistency

- Full consistency is very hard to achieve, but...
  - Various forms of **partial consistency** can reduce backtracking.
  - Especially **domain consistency**.
    - This is the workhorse of constraint programming,
    - …analogous to cutting planes in IP.

Consistency

- The concept of consistency **never developed** in the optimization literature.
  - Even though it is **closely related** to the amount of backtracking…
  - …and **cutting planes** can reduce backtracking by achieving a **greater degree of consistency**
    - …as well as by **tightening a relaxation**.

Consistency

- Goal: Explore the role of consistency in IP.
  - Understand connection between **cutting planes** and consistency.
  - Develop **LP consistency –** a form of consistency **suitable for IP**.
  - Use partial LP consistency to **reduce backtracking**.
  - **Bridge** the two thought systems (CP and IP).

Consistency and Relaxation

- Consistency allows us to **check** whether a partial assignment is feasible…
  - By checking whether it is feasible in a **relaxation** of the constraint set.
    - …a relaxation that makes this **easy to check**.
  - The relaxation consists of constraints that **contain only the variables in the partial assignment**.

Consistency and Relaxation

- Consistency allows us to **check** whether a partial assignment is feasible…
  - By checking whether it is feasible in a **relaxation** of the constraint set.
    - …a relaxation that makes this **easy to check**.
  - The relaxation consists of constraints that **contain only the variables in the partial assignment**.

We can check if $x_1 = 1$ is feasible in the **consistent** constraint set

$$
\begin{aligned}
&2x_1 + 4x_2 \le 5 \\
&7x_1 - 2x_2 \le 6 \\
&2x_1 \le 1 \\
&x_1, x_2 \in \{0, 1\}
\end{aligned}
$$

Consistency and Relaxation

- Consistency allows us to **check** whether a partial assignment is feasible…
  - By checking whether it is feasible in a **relaxation** of the constraint set.
    - …a relaxation that makes this **easy to check**.
  - The relaxation consists of constraints that **contain only the variables in the partial assignment**.

We can check if $x_1 = 1$ is feasible in the **consistent** constraint set

$$
\begin{aligned}
&2x_1 + 4x_2 \leq 5 \\
&7x_1 - 2x_2 \leq 6 \\
&2x_1 \leq 1 \\
&x_1, x_2 \in \{0, 1\}
\end{aligned}
$$

By checking whether it is feasible in the **relaxation**

$$
\begin{aligned}
&2x_1 \leq 1 \\
&x_1, x_2 \in \{0, 1\}
\end{aligned}
$$

Consistency and Relaxation

- Consistency allows us to **check** whether a partial assignment is feasible…
  - By checking whether it is feasible in a **relaxation** of the constraint set.
    - …a relaxation that makes this **easy to check**.
  - The relaxation consists of constraints that **contain only the variables in the partial assignment**.

We can check if $x_1 = 1$ is feasible in the **consistent** constraint set

$$2x_1 + 4x_2 \leq 5$$
$$7x_1 - 2x_2 \leq 6$$
$$2x_1 \leq 1$$
$$x_1, x_2 \in \{0, 1\}$$

By checking whether it is feasible in the **relaxation**

$$2x_1 \leq 1$$
$$x_1, x_2 \in \{0, 1\}$$

It is obviously **infeasible.**

LP-Consistency

- We want to do the same for IP using the **LP relaxation**

An IP constraint set is **LP-consistent** if any integer partial assignment feasible in its LP relaxation is feasible in the IP.

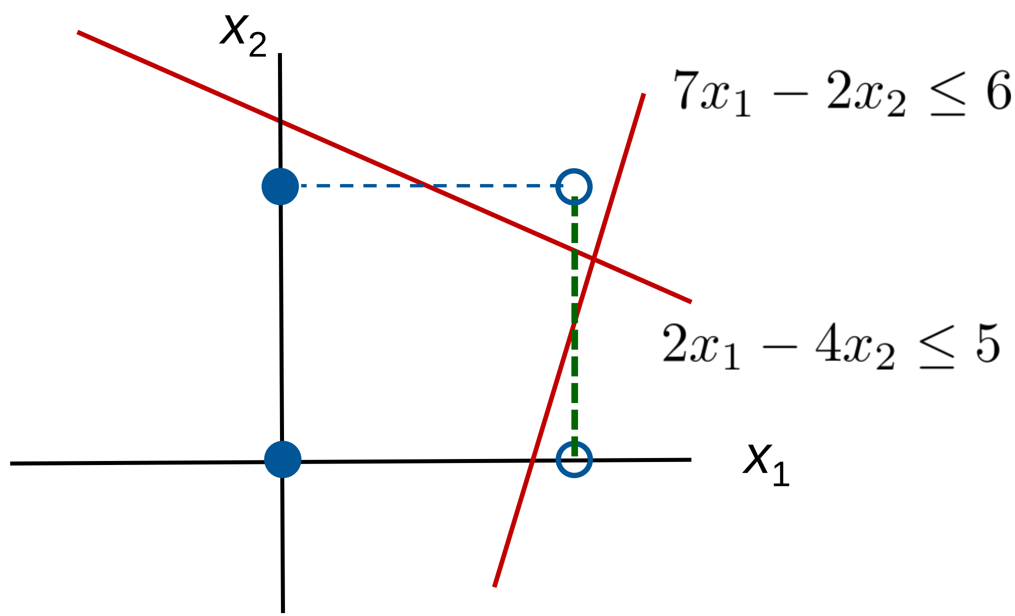
- Given LP-consistency, we can **avoid backtracking by solving LPs**
  - Check whether the partial assignment at a node is feasible in the **LP relaxation**.
  - This is **easy** – just solve the LP that results from adding the partial assignment to the constraint set.

LP-Consistency

The constraint set

$$2x_1 + 4x_2 \leq 5$$
$$7x_1 - 2x_2 \leq 6$$
$$x_1, x_2 \in \{0, 1\}$$

is **not LP-consistent** because the partial assignment $x_1 = 1$ is feasible in the LP relaxation but is infeasible in the IP.



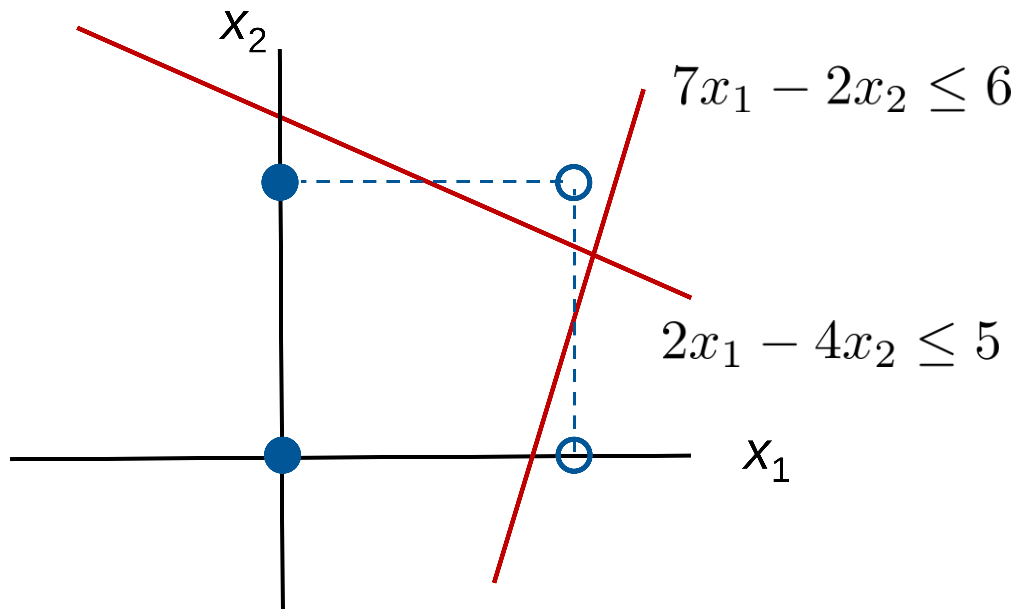
LP-consistency is a **much stronger** condition on a constraint set than feasibility of the LP relaxation.

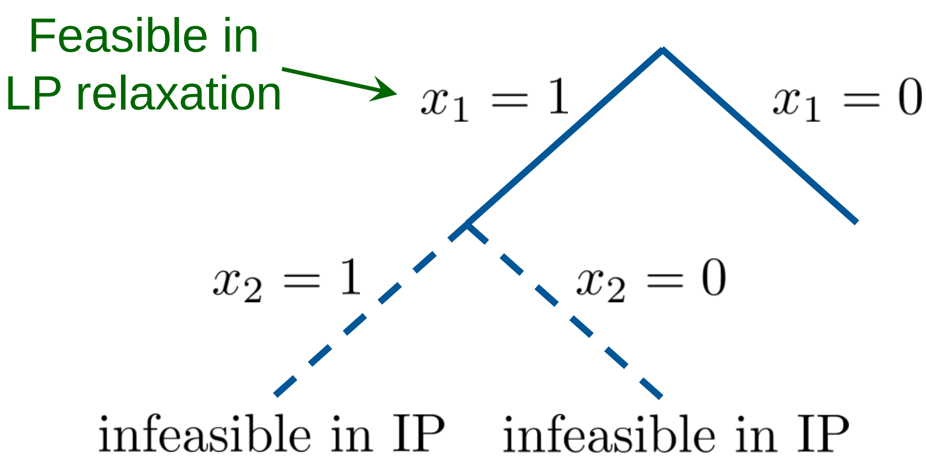
LP-Consistency

The constraint set

$$2x_1 + 4x_2 \leq 5$$
$$7x_1 - 2x_2 \leq 6$$
$$x_1, x_2 \in \{0, 1\}$$

is **not LP-consistent**





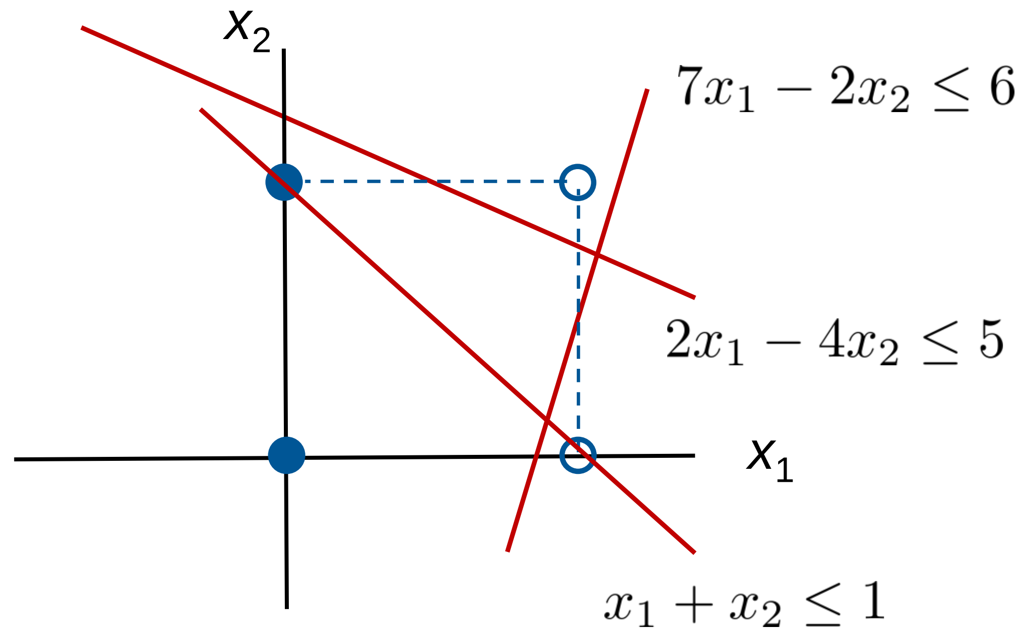
Backtracking can result even with 1-step lookahead (forward checking).

LP-Consistency

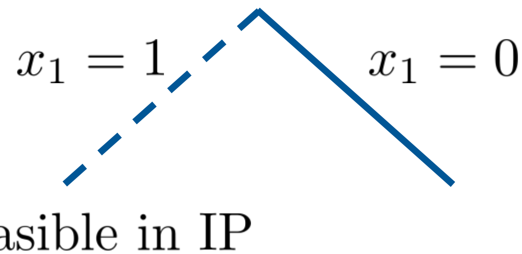
The constraint set

$$2x_1 + 4x_2 \le 5$$
$$7x_1 - 2x_2 \le 6$$
$$x_1 + x_2 \le 1$$
$$x_1, x_2 \in \{0, 1\}$$

is LP-consistent



Infeasible in LP relaxation



No backtracking
with forward checking

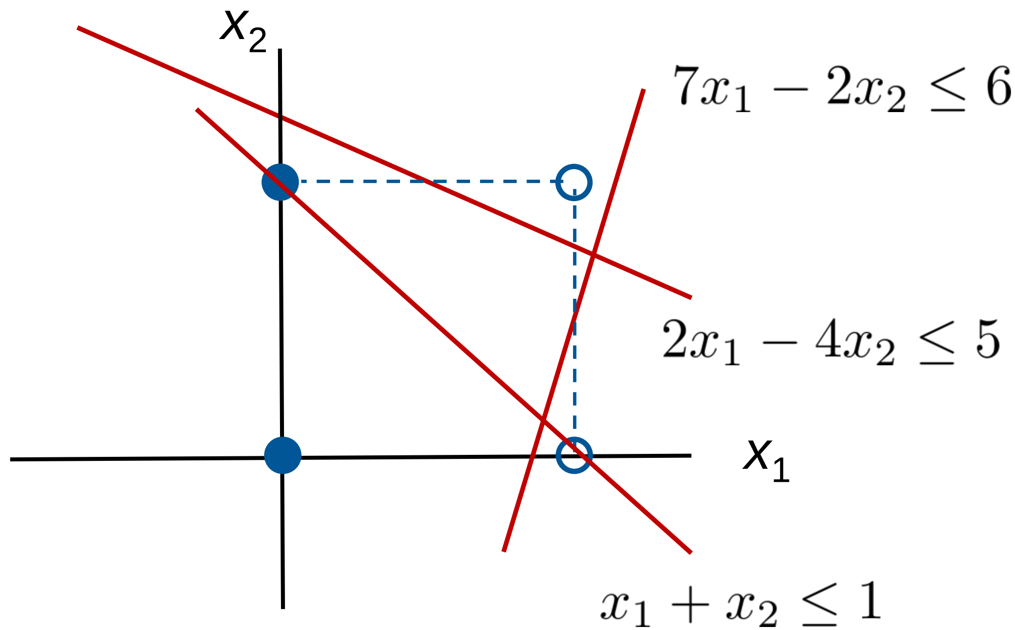
Don't take the $x_1 = 1$ branch

LP-Consistency

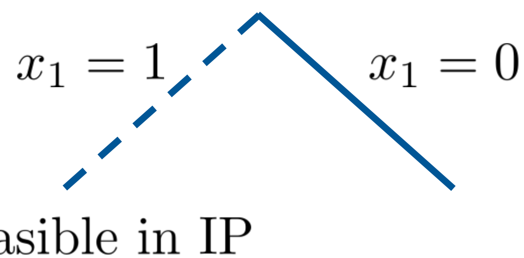
The constraint set

$$2x_1 + 4x_2 \le 5$$
$$7x_1 - 2x_2 \le 6$$
$$\boxed{x_1 + x_2 \le 1}$$
$$x_1, x_2 \in \{0, 1\}$$

is LP-consistent



Infeasible in LP relaxation



Note that we **do not use bounds from the LP relaxation.**
We only use the LP to identify infeasible partial assignments.

No backtracking
with forward checking

Don't take the $x_1 = 1$ branch

LP-Consistency and C-G Cuts

- Can **cutting planes** achieve LP-consistency?
  - Certain **Chvátal-Gomory** cuts can achieve LP-consistency.
  - For this, we need the concept of a **clausal inequality**.
    - It is a 0-1 inequality that expresses a **logical clause**.

Logical clause	Clausal inequality
$\neg x_1 \vee \neg x_2$	$x_1 + x_2 \leq 1$
$\neg x_1 \vee x_2$	$x_1 - x_2 \leq 0$
$x_1 \vee x_2$	$-x_1 - x_2 \leq -1$
$\neg x_1$	$x_1 \leq 0$

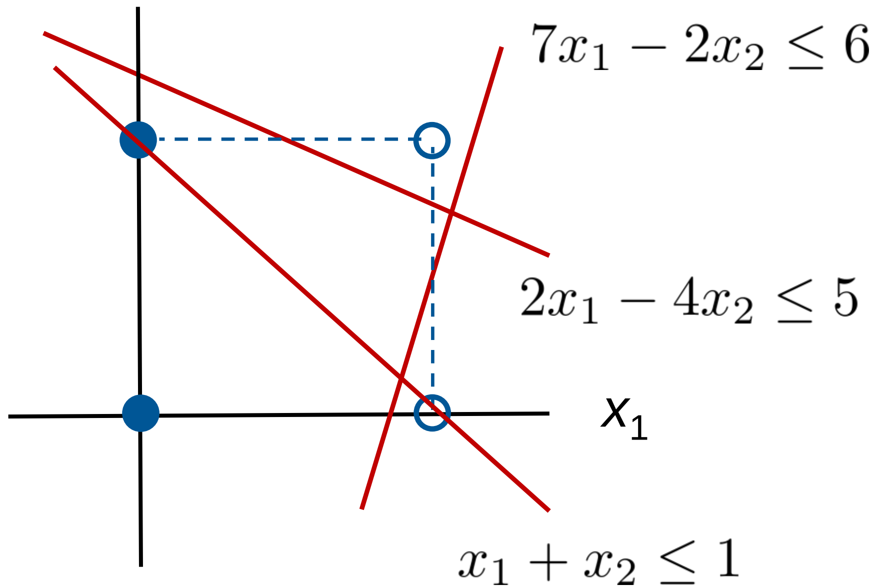
LP-Consistency and C-G Cuts

Theorem. A 0-1 constraint set is **LP-consistent** if and only if any **implied clausal inequality** is a **rank 1 C-G cut**.

The **LP-consistent** constraint set

$$2x_1 + 4x_2 \leq 5$$
$$7x_1 - 2x_2 \leq 6$$
$$x_1 + x_2 \leq 1$$
$$x_1, x_2 \in \{0, 1\}$$

implies the clausal inequality $x_1 \leq 0$
which **is** a rank 1 C-G cut…



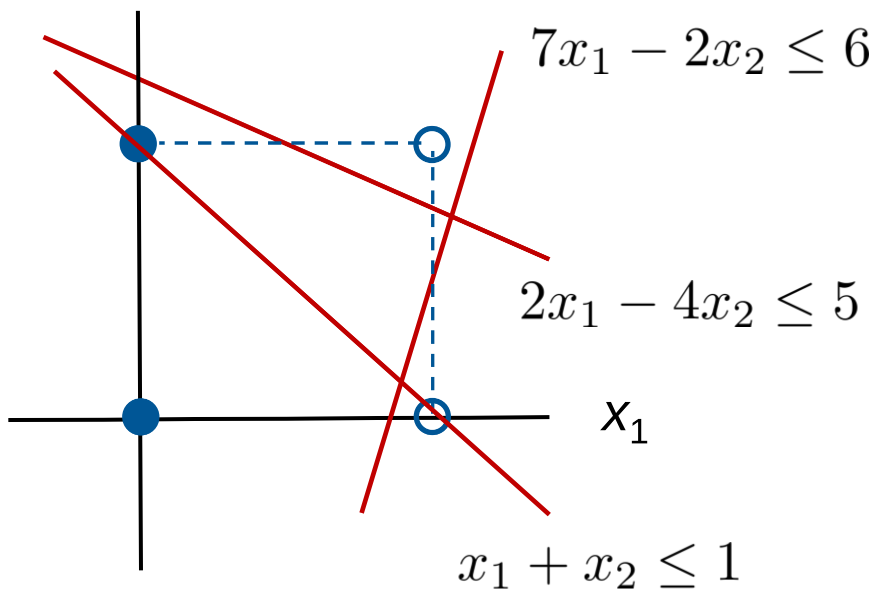
LP-Consistency and C-G Cuts

Theorem. A 0-1 constraint set is **LP-consistent** if and only if any **implied clausal inequality** is a **rank 1 C-G cut**.

…as shown by linear combination and rounding:

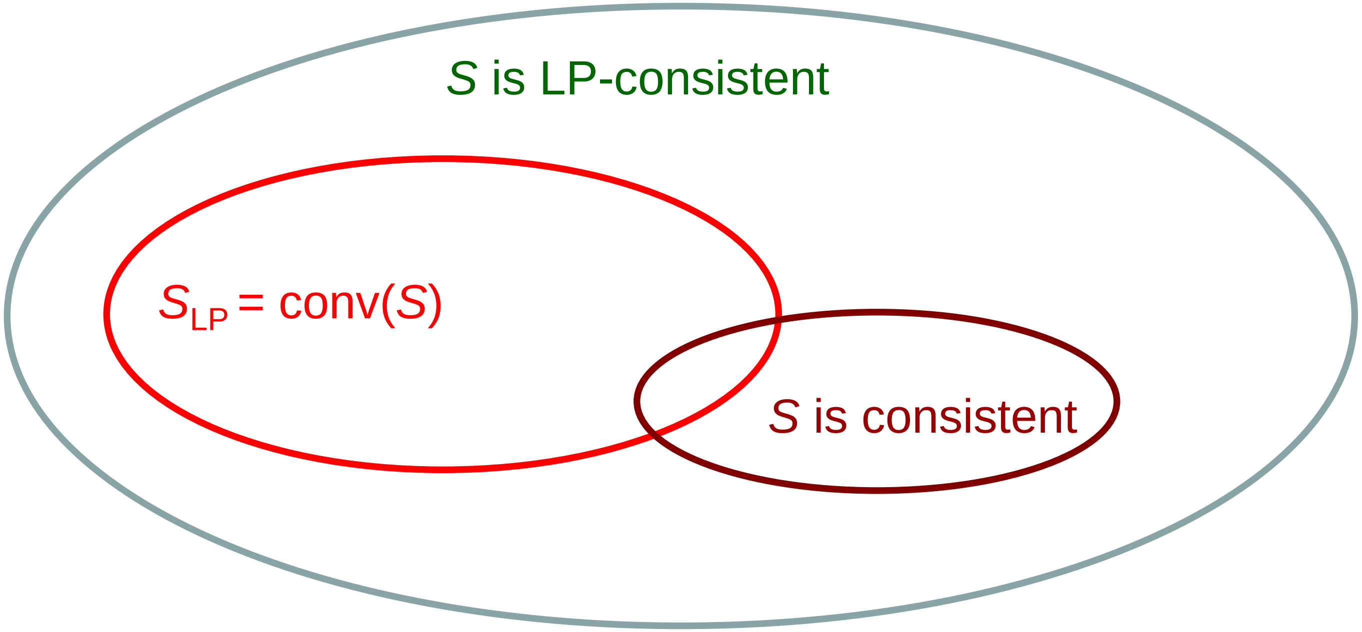
$2x_1 + 4x_2 \leq 5$	(0)
$7x_1 - 2x_2 \leq 6$	$(1/9)$
$x_1 + x_2 \leq 1$	$(2/9)$

$$x_1 \leq 8/9 \Rightarrow x_1 \leq 0$$



Consistency and the Convex Hull

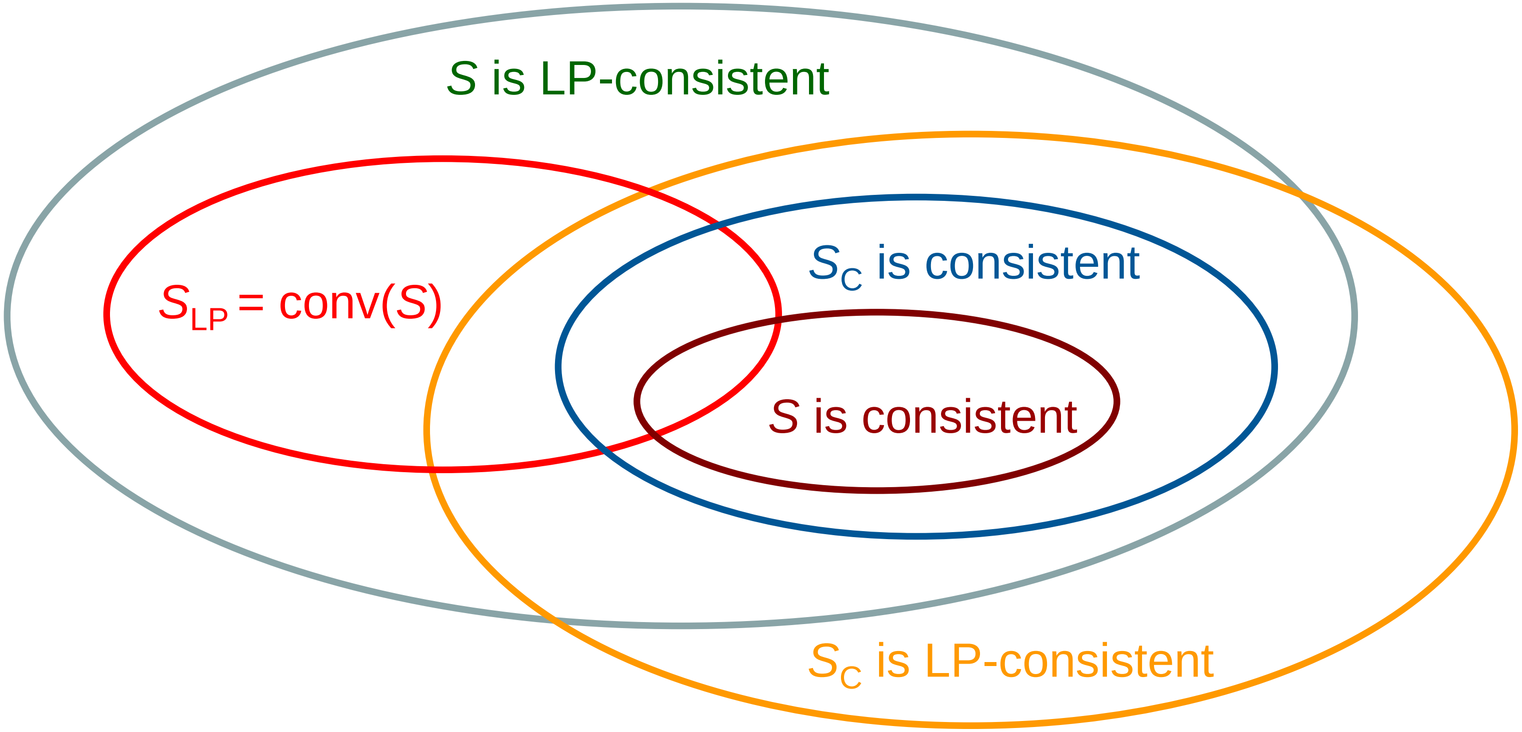
S = 0-1 constraint set $\qquad S_{\mathrm{LP}}$ = LP relaxation of S



Consistency and the Convex Hull

S = 0-1 constraint set $\qquad$ S_{LP} = LP relaxation of S

$$S_{\mathrm{C}} = \left\{ \begin{array}{c} \text{clausal inequalities implied by} \\ \text{individual constraints in } S \end{array} \right\}$$



Partial LP-consistency

- Full LP-consistency is hard to achieve.
  - In principle, can achieve it by generating **all rank 1 clausal C-G inequalities** (from the Theorem).
    - This is not practical.
  - We define a form of **partial** LP-consistency.
    - Analogous to *k*-consistency in constraint programming.

Partial LP-consistency

- Full LP-consistency is hard to achieve.
  - In principle, can achieve it by generating **all rank 1 clausal C-G inequalities** (from the Theorem).
    - This is not practical.
  - We define a form of **partial** LP-consistency.
    - Analogous to *k*-consistency in constraint programming.

A 0-1 constraint set is **rank *r* LP-consistent over variable set *J***…
if any partial assignment to variables in ***J*** that is feasible in the LP relaxation…
can be **extended** to ***r*** additional variables and still be feasible in the LP.

Partial LP-consistency

- Rank *r* LP-consistency reduces backtracking.
  - Roughly speaking, one can descend *r* more levels into the search tree without having to backtrack.

Partial LP-consistency

- Rank r LP-consistency reduces backtracking.
  - Roughly speaking, one can descend r more levels into the search tree without having to backtrack.
- We can achieve rank r LP-consistency over J with a restricted form of RLT.*

Theorem. Rank r LP-consistency can be achieved by a rank r' RLT algorithm** for a computable value of r', where r' may be substantially less than r.

*Reformulation and linearization technique.

**The RLT algorithm lifts into r' additional dimensions.

Partial LP-consistency

Let $S = \{Ax \le b,\ x \in \{0,1\}^n\}$ be a 0–1 constraint set. Apply RLT to S for a given $K \subset N \setminus J$ by generating the nonlinear system

$$(Ax - b) \prod_{j \in J_1} x_j \prod_{j \in J \setminus J_1} (1 - x_j) \le 0, \quad \text{all } J_1 \subseteq J$$

Linearize this system and project it onto J to obtain $\mathcal{R}_K(S_{\text{LP}})|_J$. Let $\mathcal{R}(S_{\text{LP}})|_J$ be the union of $\mathcal{R}_K(S_{\text{LP}})|_J$ over all K with $|K| = r'$, and add the inequalities in $\mathcal{R}(S_{\text{LP}})|_J$ to S to obtain $\hat{S}$.

Theorem. Define

$$r = \min_{K \subseteq N \setminus J} \left\{ |K| \;\middle|\; S_{\text{LP}} \cup \{x_{J \cup K} = v_{J \cup K}\} \text{ is infeasible for all } v_K \in \{0,1\}^{|K|} \right\}$$

with minimizer $K_{\min}$. Let K^* consist of the elements k of $K_{\min}$ such that $S_{\text{LP}} \cup \{x_{J \cup \{k\}} = v_{J \cup \{k\}}\}$ is infeasible for exactly one 0–1 value assignment v_k. Then $\hat{S}$ is rank r LP-consistent over J if we set

$$r' = \max\{r - |K^*|, 1\}$$

Partial LP-consistency

Consider the constraint set S :

$$
\begin{aligned}
2x_1 + 2x_2 \qquad\qquad\qquad\quad &\leq 3 \\
2x_1 \qquad\quad + 2x_3 \qquad\quad &\leq 3 \\
2x_1 - 2x_2 - 2x_3 - 2x_4 &\leq 1 \\
2x_1 - 2x_2 - 2x_3 + 2x_4 &\leq 3 \\
x_j \in \{0, 1\},\ \text{all } j &
\end{aligned}
$$

$x_1 = 1$ is feasible in S_{LP} but not in S.

Setting $x_1 = 1$ results in backtracking.

Partial LP-consistency

Consider the constraint set S :

$$\begin{aligned}
&2x_1 + 2x_2 &&\le 3\\
&2x_1 \qquad\quad + 2x_3 &&\le 3\\
&2x_1 - 2x_2 - 2x_3 - 2x_4 &&\le 1\\
&2x_1 - 2x_2 - 2x_3 + 2x_4 &&\le 3\\
&x_j \in \{0,1\},\ \text{all } j
\end{aligned}$$

$x_1 = 1$ is feasible in S_{LP} but not in S.

Setting $x_1 = 1$ results in backtracking.

Here ***r* = 3** and ***r′* = 1**. We apply RLT with $J = \{1\}$ and $r' = 1$ and thereby achieve **rank 3 LP-consistency** over {1}.

This means we can move **3 levels deeper** into the tree without backtracking, by applying only a **rank 1** RLT algorithm.

Partial LP-consistency

Consider the constraint set S :

$$
\begin{aligned}
&2x_1 + 2x_2 && \leq 3 \\
&2x_1 \qquad\quad + 2x_3 && \leq 3 \\
&2x_1 - 2x_2 - 2x_3 - 2x_4 && \leq 1 \\
&2x_1 - 2x_2 - 2x_3 + 2x_4 && \leq 3 \\
&x_j \in \{0,1\}, \text{ all } j
\end{aligned}
$$

$x_1 = 1$ is feasible in S_{LP} but not in S.
Setting $x_1 = 1$ results in backtracking.

Here $\boldsymbol{r = 3}$ and $\boldsymbol{r' = 1}$. We apply RLT with $J = \{1\}$ and $r' = 1$ and thereby achieve **rank 3 LP-consistency** over $\{1\}$.

This means we can move **3 levels deeper** into the tree without backtracking, by applying only a **rank 1** RLT algorithm.

Since there are 4 variables, we can now solve the problem **without backtracking** by checking which branches are feasible in the LP relaxation.

Consistency Cuts

- There is no need to use **all** the inequalities generated by RLT.
  - At each node of the search tree, we use a **cut generating LP** to identify **one RLT inequality** that makes the LP relaxation at the current node infeasible.
    - If such an inequality exists, of course.
  - We call this inequality a **consistency cut**.

Experiments

- At this stage, no attempt to incorporate consistency cuts into a state-of-the-art solver.
- Only a preliminary comparison of **consistency** RLT cuts with **separating** RLT cuts.
  - Use rank 1 RLT only.
  - No other cutting planes, for direct comparison.
    - Solve with CPLEX 12.8
    - Fixed branching order, no presolve.
  - Random and MIPLIB instances
    - Small, dense random instances.
    - MIPLIB instances hard enough for meaningful comparison, easy enough for manageable search tree.

Experiments

- At this stage, no attempt to incorporate consistency cuts into a state-of-the-art solver.
- Only a preliminary comparison of **consistency** RLT cuts with **separating** RLT cuts.
  - Use rank 1 RLT only.
  - No other cutting planes, for direct comparison.
    - Solve with CPLEX 12.8
    - Fixed branching order, no presolve.
  - Random and MIPLIB instances
    - Small, dense random instances.
    - MIPLIB instances hard enough for meaningful comparison, easy enough for manageable search tree.
  - Bounds on objective function.
    - Since consistency cuts detect infeasibility.

Experiments

Random instances

Separating RLT cuts vs. consistency cuts

Each number is an average over 5 instances

Rows	Cols	Nodes		Time (sec)	
		Sep RLT	Consis	Sep RLT	Consis
30	30	2824	**299**	579	**202**
35	35	4136	**408**	1550	**522**
45	45	23058	**7768**	16993	**10276**
50	40	16981	**1198**	11672	**2822**
60	50	*	**47936**	*	**151401**

*Memory exceeded in 4 of 5 instances

Experiments

MIPLIB instances

Separating RLT cuts vs. consistency cuts

Instance	Rows	Cols	Nodes		Time (sec)	
			Sep RLT	Consis	Sep RLT	Consis
p0040	23	40	50	**30**	**27**	31
stein15inf	37	15	75	**20**	3	**2**
bm23	20	27	178	**38**	19	**14**
sentoy	30	60	258	**29**	152	**80**
pipex	41	48	762	**547**	**1362**	1415
p0201	133	201	847	**533**	519	**514**
f2gap40400	40	400	861	**780**	662	**304**
stein27	118	27	4099	**3900**	2242	**1715**
p0033	15	33	22581	**321**	4761	**180**
enigma	42	100	40218	**27960**	423	**118**
mod008inf	7	319	57495	**65**	35656	**684**
lseu	28	89	247795	**234450**	4196	**3096**

Research Issues

- Are there **other general-purpose schemes** for achieving LP consistency with cutting planes?
  - Or perhaps other types of consistency.
- To what extent do cutting planes for **particular problem classes** achieve consistency?
  - Clique cuts, covers, TSP cuts, etc.
- Can LP consistency yield **new approaches** to solving particular problem classes?
  - Using new families of specialized consistency cuts.

References

- D. Davarnia and J. N. Hooker, Consistency for 0-1 programming, in L.-M. Rousseau and K. Stergiou, eds., *CPAIOR 2019 Proceedings*, 225-240.
- D. Davarnia, A. Rajabalizadeh, and J. N. Hooker, Achieving consistency with cutting planes, 2021, submitted.