Branching strategies for solving pseudo-Boolean optimization problems using ILP solvers

(Pragmatics of SAT Workshop)

Srinivas Subramanya Tamvada and Elkafi Hassini

DeGroote School of Business, McMaster University

July 4, 2021

Outline of the presentation

Preliminaries

Frequency based branching heuristic: MOHP

Heuristic for BCP based look-ahead branching

Preliminaries

What are ILPs and PBOs ?

What are ILPs and PBOs ?

- ILP: minimization problem with integer, binary, and continuous variables

What are ILPs and PBOs ?

- ILP: minimization problem with integer, binary, and continuous variables
  - Linear objective and constraints

What are ILPs and PBOs ?

- ILP: minimization problem with integer, binary, and continuous variables
  - Linear objective and constraints
- PBO: ILP with binary variables

What are ILPs and PBOs ?

- ILP: minimization problem with integer, binary, and continuous variables
  - Linear objective and constraints
- PBO: ILP with binary variables
  - Can be solved using ILP solvers such as CPLEX

Solving PBOs using ILP solvers

Solving PBOs using ILP solvers

- Advantages

Solving PBOs using ILP solvers

- Advantages
  - Deal with objective function

Solving PBOs using ILP solvers

- Advantages
  - Deal with objective function
  - Integrality gap for hard problems

Solving PBOs using ILP solvers

- Advantages
  - Deal with objective function
  - Integrality gap for hard problems
- Disadvantages

Solving PBOs using ILP solvers

- Advantages
  - Deal with objective function
  - Integrality gap for hard problems
- Disadvantages
  - Poor scaling upon distribution [Ralphs, T., Shinano, Y., Berthold, T., Koch, T.(2018)]

Solving PBOs using ILP solvers

- Advantages
  - Deal with objective function
  - Integrality gap for hard problems
- Disadvantages
  - Poor scaling upon distribution [Ralphs, T., Shinano, Y., Berthold, T., Koch, T.(2018)]
  - Dependency on pseudo-cost updates from other parts of the search tree

Motivation and Goals

Motivation and Goals

- Incorporate SAT/PBO branching heuristics into CPLEX

Motivation and Goals

- Incorporate SAT/PBO branching heuristics into CPLEX
  - Only use properties of the node being branched (compare to pseudo-Cost branching)

Motivation and Goals

- Incorporate SAT/PBO branching heuristics into CPLEX
  - Only use properties of the node being branched (compare to pseudo-Cost branching)
  - Quality must be comparable to CPLEX strong branching (best available for ILPs)

Motivation and Goals

- Incorporate SAT/PBO branching heuristics into CPLEX
  - Only use properties of the node being branched (compare to pseudo-Cost branching)
  - Quality must be comparable to CPLEX strong branching (best available for ILPs)
- But SAT/PBO branching heuristics are not designed to handle the objective function !

Motivation and Goals

- Incorporate SAT/PBO branching heuristics into CPLEX
  - Only use properties of the node being branched (compare to pseudo-Cost branching)
  - Quality must be comparable to CPLEX strong branching (best available for ILPs)
- But SAT/PBO branching heuristics are not designed to handle the objective function !
  - Modify the popular Maximal Occurance in Minimal Size heuristic (MOMS)

Motivation and Goals

- Incorporate SAT/PBO branching heuristics into CPLEX
  - Only use properties of the node being branched (compare to pseudo-Cost branching)
  - Quality must be comparable to CPLEX strong branching (best available for ILPs)
- But SAT/PBO branching heuristics are not designed to handle the objective function !
  - Modify the popular Maximal Occurance in Minimal Size heuristic (MOMS)
  - Proposed heuristics can also be used in existing PBO solvers

Use case

Use case

- Less-than-truckload freight routing

Use case

- Less-than-truckload freight routing
- Commercially important (Multi-Billon dollar industry for companies like FedEX, UPS, Canada Post)

Use case

- Less-than-truckload freight routing
- Commercially important (Multi-Billon dollar industry for companies like FedEX, UPS, Canada Post)
- Real data from an industry partner

Use case

- Less-than-truckload freight routing
- Commercially important (Multi-Billon dollar industry for companies like FedEX, UPS, Canada Post)
- Real data from an industry partner
- ILP has many constraints with 2 variables (Boolean Constraint Propagation, BCP)

Recall: 2 types of Branching

Recall: 2 types of Branching

- Estimate based

Recall: 2 types of Branching

- Estimate based
  - Frequency of occurrence in constraints or the objective coefficient (SAT/PBO)

Recall: 2 types of Branching

- Estimate based
  - Frequency of occurrence in constraints or the objective coefficient (SAT/PBO)
  - Pseudo cost estimates (ILP)

Recall: 2 types of Branching

- Estimate based
  - Frequency of occurrence in constraints or the objective coefficient (SAT/PBO)
  - Pseudo cost estimates (ILP)
- Look ahead based

Recall: 2 types of Branching

- Estimate based
  - Frequency of occurrence in constraints or the objective coefficient (SAT/PBO)
  - Pseudo cost estimates (ILP)
- Look ahead based
  - BCP for SAT/PBO

Recall: 2 types of Branching

- Estimate based
  - Frequency of occurrence in constraints or the objective coefficient (SAT/PBO)
  - Pseudo cost estimates (ILP)
- Look ahead based
  - BCP for SAT/PBO
  - Strong branching (ILP), but it is known that probing has a role to play in branching

Recall: 2 types of Branching

- Estimate based
  - Frequency of occurrence in constraints or the objective coefficient (SAT/PBO)
  - Pseudo cost estimates (ILP)
- Look ahead based
  - BCP for SAT/PBO
  - Strong branching (ILP), but it is known that probing has a role to play in branching
- First few branching decisions are vitally important ! (for both SAT and ILP) [Heule, M. J., Kullmann, O., Wieringa, S., & Biere, A. (2011, December).]

Frequency based branching heuristic: MOHP

Maximal occurrence in Highest Priority

Maximal occurrence in Highest Priority

- Recall: MOMS heuristic quickly gets rid of large volumes of infeasible space

Maximal occurrence in Highest Priority

- Recall: MOMS heuristic quickly gets rid of large volumes of infeasible space
  - High branching priority to variables in the smaller size constraints

Maximal occurrence in Highest Priority

- Recall: MOMS heuristic quickly gets rid of large volumes of infeasible space
  - High branching priority to variables in the smaller size constraints
- In the presence of an objective function, we change the priority definition that MOMS uses

Maximal occurrence in Highest Priority

- Recall: MOMS heuristic quickly gets rid of large volumes of infeasible space
  - High branching priority to variables in the smaller size constraints
- In the presence of an objective function, we change the priority definition that MOMS uses
  - quickly get rid of infeasible vertices which have very low objective function values

No Goods (PBO to SAT)

No Goods (PBO to SAT)

- We convert constraints into no-goods

No Goods (PBO to SAT)

- We convert constraints into no-goods
  - $x_1 + x_2 >= 1$ converts to the nogood $\{(x_1=0), (x_2=0)\}$ [Eén, N., & Sörensson, N. (2006).]

No Goods (PBO to SAT)

- We convert constraints into no-goods
  - $x_1 + x_2 >= 1$ converts to the nogood $\{(x_1=0), (x_2=0)\}$ [Eén, N., & Sörensson, N. (2006).]
  - A nogood is unit hypercube in n-dimensional space every vertex of which is infeasible

No Goods (PBO to SAT)

- We convert constraints into no-goods
  - $x_1 + x_2 >= 1$ converts to the nogood $\{(x_1=0), (x_2=0)\}$ [Eén, N., & Sörensson, N. (2006).]
  - A nogood is unit hypercube in n-dimensional space every vertex of which is infeasible
- Then we assign a priority to each no-good

No Goods (PBO to SAT)

- We convert constraints into no-goods
  - $x_1 + x_2 >= 1$ converts to the nogood $\{(x_1=0), (x_2=0)\}$ [Eén, N., & Sörensson, N. (2006).]
  - A nogood is unit hypercube in n-dimensional space every vertex of which is infeasible
- Then we assign a priority to each no-good
  - REMEMBER: we want to get rid of the best unconstrained vertex (if its infeasible), PLUS

No Goods (PBO to SAT)

- We convert constraints into no-goods
  - $x_1 + x_2 >= 1$ converts to the nogood $\{(x_1=0), (x_2=0)\}$ [Eén, N., & Sörensson, N. (2006).]
  - A nogood is unit hypercube in n-dimensional space every vertex of which is infeasible
- Then we assign a priority to each no-good
  - REMEMBER: we want to get rid of the best unconstrained vertex (if its infeasible), PLUS
  - also get rid of infeasible vertices in the vicinity that have low objective values

Nogood Priority

Nogood Priority

- Reference point is defined for each variable in the nogood as $= 0.5$

Nogood Priority

- Reference point is defined for each variable in the nogood as $= 0.5$
  - Can use the linear relaxed fractional value of the variable (simplex)

Nogood Priority

- Reference point is defined for each variable in the nogood as $= 0.5$
  - Can use the linear relaxed fractional value of the variable (simplex)
- The priority of a variable fixing ($\mathtt{x_i = v_i}$) is defined as the decrease in the objective function when $\mathtt{x_i}$'s value changes from the reference point to $\mathtt{v_i}$.

Nogood Priority

- Reference point is defined for each variable in the nogood as $= 0.5$
  - Can use the linear relaxed fractional value of the variable (simplex)
- The priority of a variable fixing ($\mathtt{x_i} = \mathtt{v_i}$) is defined as the decrease in the objective function when $\mathtt{x_i}$'s value changes from the reference point to $\mathtt{v_i}$.
- Each no-good is assigned a priority that is equal to the sum of the priorities of its variable fixings.

Nogood Priority Example

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:
  - Constraint$_1$: $x_1 + x_2 \geq 1$ priority $0.5 + 2*0.5 = 1.5$

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:
  - Constraint$_1$: $x_1 + x_2 >= 1$ priority $0.5 + 2*0.5 = 1.5$
  - Constraint$_2$: $x_2 + x_3 >= 1$ priority $2*0.5 + 300*0.5 = 151$

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:
  - Constraint$_1$: $x_1 + x_2$ >=1 priority $0.5 + 2*0.5 = 1.5$
  - Constraint$_2$: $x_2 + x_3$ >=1 priority $2*0.5 + 300*0.5 = 151$
  - Constraint$_3$: $x_1 + x_6$ <=1 priority $-0.5 - 6000*0.5 = -3000.5$

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:
  - Constraint$_1$: $x_1 + x_2 >= 1$ priority $0.5 + 2*0.5 = 1.5$
  - Constraint$_2$: $x_2 + x_3 >= 1$ priority $2*0.5 + 300*0.5 = 151$
  - Constraint$_3$: $x_1 + x_6 <= 1$ priority $-0.5 - 6000*0.5 = -3000.5$
  - Constraint$_4$: $x_1 + x_4 <= 1$ priority $-0.5 - 0.5 = -1$

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:
  - Constraint$_1$: $x_1 + x_2 >= 1$ priority $0.5 + 2*0.5 = 1.5$
  - Constraint$_2$: $x_2 + x_3 >= 1$ priority $2*0.5 + 300*0.5 = 151$
  - Constraint$_3$: $x_1 + x_6 <= 1$ priority $-0.5 - 6000*0.5 = -3000.5$
  - Constraint$_4$: $x_1 + x_4 <= 1$ priority $-0.5 - 0.5 = -1$
  - Constraint$_5$: $x_1 + x_5 <= 1$ priority $-0.5 - 0.5 = -1$

Nogood Priority Example

- Minimize: $x_1 + 2x_2 + 300x_3 + x_4 + x_5 + 6000x_6$
- Subject to:
  - Constraint$_1$: $x_1 + x_2 >= 1$ priority $0.5 + 2*0.5 = 1.5$
  - Constraint$_2$: $x_2 + x_3 >= 1$ priority $2*0.5 + 300*0.5 = 151$
  - Constraint$_3$: $x_1 + x_6 <= 1$ priority $-0.5 - 6000*0.5 = -3000.5$
  - Constraint$_4$: $x_1 + x_4 <= 1$ priority $-0.5 - 0.5 = -1$
  - Constraint$_5$: $x_1 + x_5 <= 1$ priority $-0.5 - 0.5 = -1$
- Branch on x_2 or x_3, tie-break won by x_2

MOHP observations

MOHP observations

- Only consider those no-goods which render the best unconstrained vertex infeasible

MOHP observations

- Only consider those no-goods which render the best unconstrained vertex infeasible
- If there aren't any such no-goods, then the best unconstrained vertex is a solution

MOHP observations

- Only consider those no-goods which render the best unconstrained vertex infeasible
- If there aren't any such no-goods, then the best unconstrained vertex is a solution
- In a best-first search, this vertex is also the optimal solution

Heuristic for BCP based look-ahead branching

Motivation

Motivation

- Look-ahead branching based on Probing (BCP)

Motivation

- Look-ahead branching based on Probing (BCP)
- Time consuming to run BCP with every variable that appears in constraints having 2 variables

Motivation

- Look-ahead branching based on Probing (BCP)
- Time consuming to run BCP with every variable that appears in constraints having 2 variables
- Heuristic for running BCP with only a few such variables

Motivation

- Look-ahead branching based on Probing (BCP)
- Time consuming to run BCP with every variable that appears in constraints having 2 variables
- Heuristic for running BCP with only a few such variables
- Example: PROP heuristic [Li and Anbulagan, 1997]

Dominated Trigger

Dominated Trigger

- A trigger is a variable and its fixing (to 0 or 1) used to initiate BCP

Dominated Trigger

- A trigger is a variable and its fixing (to 0 or 1) used to initiate BCP
- A dominated trigger is a trigger corresponding to a variable fixing that is caused by some other trigger

Dominated Trigger

- A trigger is a variable and its fixing (to 0 or 1) used to initiate BCP
- A dominated trigger is a trigger corresponding to a variable fixing that is caused by some other trigger
- An APEX trigger is a trigger that is not dominated by any other trigger

Dominated Trigger

- A trigger is a variable and its fixing (to 0 or 1) used to initiate BCP
- A dominated trigger is a trigger corresponding to a variable fixing that is caused by some other trigger
- An APEX trigger is a trigger that is not dominated by any other trigger
- Only variables from APEX triggers are considered for branching

Dominated Trigger

- A trigger is a variable and its fixing (to 0 or 1) used to initiate BCP
- A dominated trigger is a trigger corresponding to a variable fixing that is caused by some other trigger
- An APEX trigger is a trigger that is not dominated by any other trigger
- Only variables from APEX triggers are considered for branching
- Branch on variables on the periphery of the implication graph

Apex Triggers example

Apex triggers in the constraint set below are (x_4,0) and (x_6,1).
We branch on one of x_4 or x_6, using maximin criteria

$$
\begin{aligned}
c1 &: x_1 + x_2 \geqslant 1 \\
c2 &: x_3 + x_2 \leqslant 1 \\
c3 &: x_3 + x_5 \geqslant 1 \\
c4 &: x_7 + x_4 \geqslant 1 \\
c5 &: x_1 + x_7 \leqslant 1 \\
c6 &: x_5 + x_6 \leqslant 1
\end{aligned}
$$

Results

Table 1. A comparison of CPLEX strong branching with look-ahead based branching and MOHP.

PBO	Root node BCP			Results				Nodes (first hour)		Nodes (end of test)	
	Fractional	Apex	Savings	Time	Solution	Best-bound	Gap	Remaining	Processed	Remaining	Processed
				24	75863.82	73486.57	0.03	32180	36189	770272	940175
LTL routing 11th March	2	2	0	24	75334.53	74550.1	0.01	767	1837	782504	1107662
				24	75196.82	74590.76	0.01	1137	1871	584775	1098021
				24	58391.92	54768.51	0.06	15631	20580	358302	400829
LTL routing 9th March	6	6	0	24	58179.65	57230.78	0.02	4118	9580	161024	293392
				24	58049.91	57122.12	0.02	9016	14110	213597	368718
				24	None	1880.90	NA	62844	73360	1109825	1324771
hanoi5	646	565	0.12	24	None	1882.15	NA	517	522	1329967	1528334
				24	None	1880.52	NA	23770	27125	1146661	1384544
				23	-33266.98	-33270.30	0.0	1446	12319	794	76102
opm2-z10-s4+	111	89	0.2	24	-33265.0	-33270.5	0.0	2520	2525	7491	61040
				23	-33269	-33269	0.0	2613	4631	5384	48433

Conclusions and future work

Conclusions and future work

- Both heuristics comparable in performance to CPLEX strong branching

Conclusions and future work

- Both heuristics comparable in performance to CPLEX strong branching
- Can be integrated into PBO solvers

Conclusions and future work

- Both heuristics comparable in performance to CPLEX strong branching
- Can be integrated into PBO solvers
- Can even be used to solve ILPs that are not PBOs, by treating the fractional part of each integer variable as a binary variable

Thank You!

Questions and comments to tamvadss@mcmaster.ca and hassini@mcmaster.ca