The Minimum Spanning Forest with Balance Constraints Problem (MSFBC)

Ian Herzterg, **Marcus Poggi**, Thibault Vidal
Departamento de Informática, PUC-Rio, Brazil
{iherzterg, poggi, vidalt} @inf.puc-rio.br

July 15, 2015

Minimum Spanning Forest with Balance Constraints

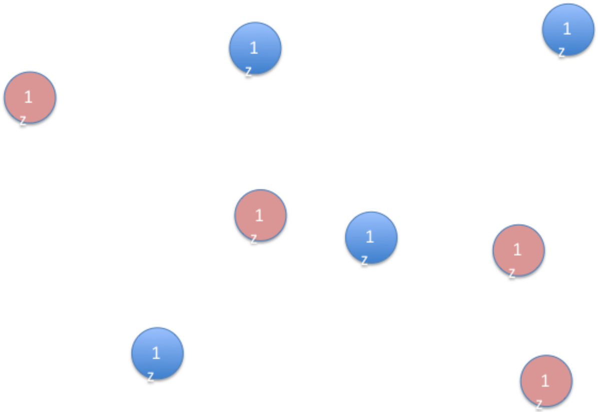
- Instance:
  - a vertex set R integer weights p_v, $v \in R$;
  - nonnegative costs between all pairs of vertices and from all vertices to terminal 0(border), let $V = R \cup \{0\}$;
- Find a minimum cost edge set F such that all connected components are:
  - acyclic;
  - either balanced ($\sum_{v \in comp} p_v = 0$) or connected to the terminal.

Minimum Spanning Forest with Balance Constraints

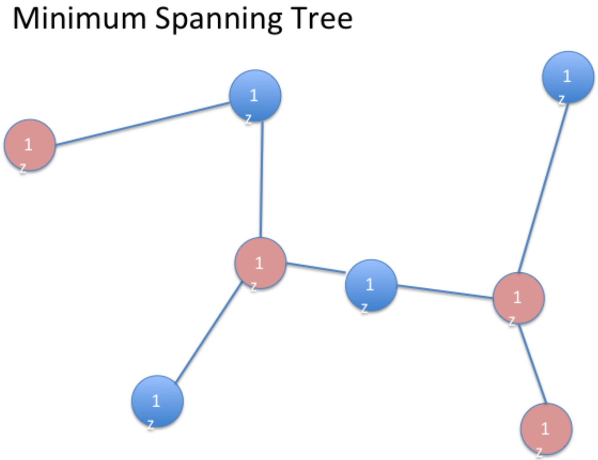
- Instance:
  - a vertex set R integer weights p_v, $v \in R$;
  - nonnegative costs between all pairs of vertices and from all vertices to terminal 0(border), let $V = R \cup \{0\}$;
- Find a minimum cost edge set F such that all connected components are:
  - acyclic;
  - either balanced ($\sum_{v \in comp} p_v = 0$) or connected to the terminal.

Minimum Spanning Forest with Balance Constraints

Given:

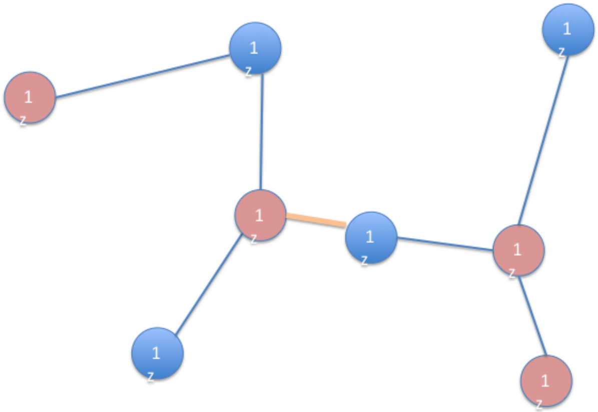


Minimum Spanning Forest with Balance Constraints

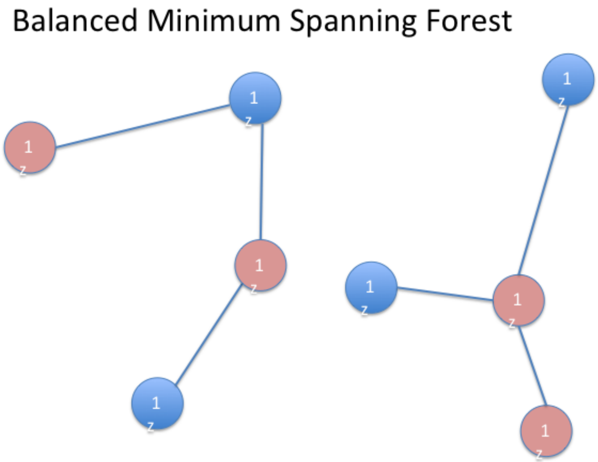


Minimum Spanning Forest with Balance Constraints

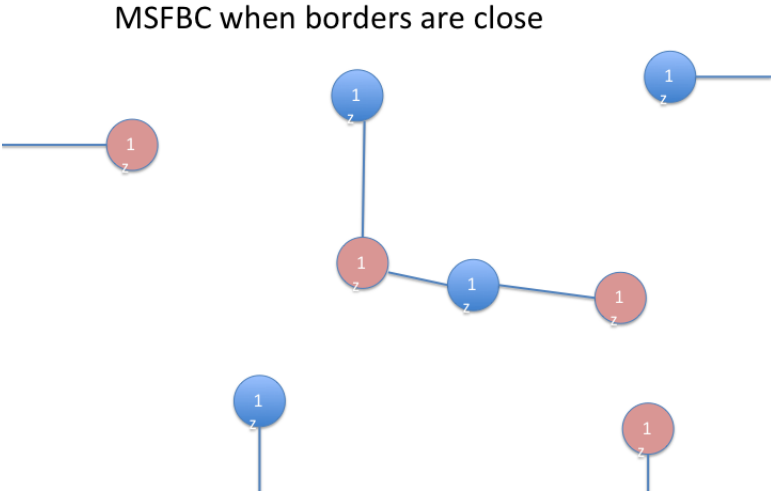
MST → MSFBC



Minimum Spanning Forest with Balance Constraints



Minimum Spanning Forest with Balance Constraints



Minimum Spanning Forest with Balance Constraints

MSFBC: Instance

+1	-1	-1	+1	+1
3	8	12	9	13
	17	5	16	6
		7	21	9
			2	11
				12

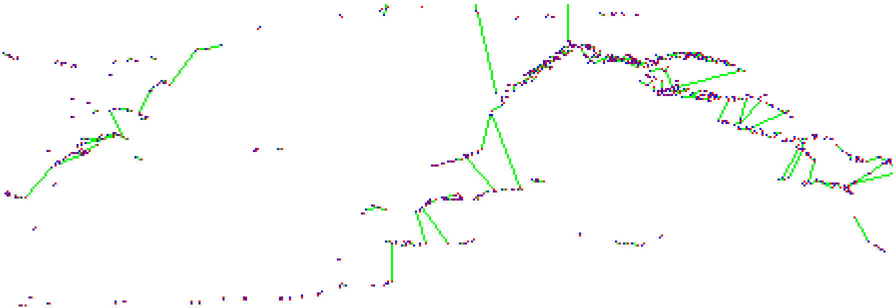


Figure: *Long - Matching*

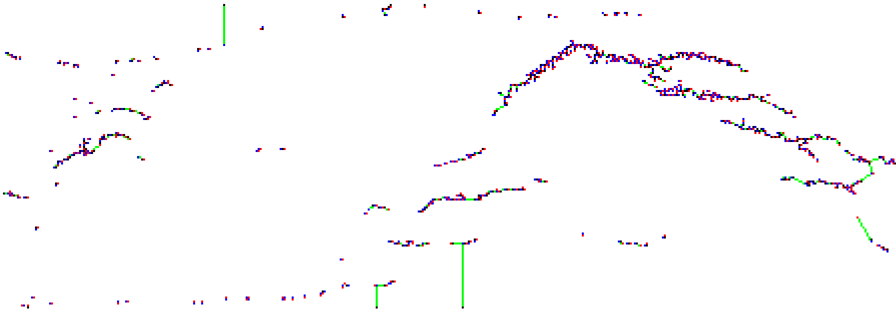


Figure: *Long - MSFBC*

Theorem

MSFBC is NP-Hard

- Steiner Problem in Graphs (SPG): $G = (V, E)$, $T \subset V$, $w_e\ e \in E$, find minimum weight edge set E' connecting all vertices in T.

Reduction:

- Let R be V where all non-terminal vertices are doubled $|R| = 2.|V| - |T|$
- Choose a vertex $r \in T$ and assign weights p_v as follows:
  - $p_v = -1, \forall\ v \in T \setminus \{r\}$
  - $p_r = |T| - 1$
  - For each pair v, w of doubled non-terminal vertex assign $p_v = +1$, $p_w = -1$
- Connect all pairs of non-terminal vertices by a zero cost edge, and each of them to the other vertices as before (both when also non-terminal vertex).

Theorem

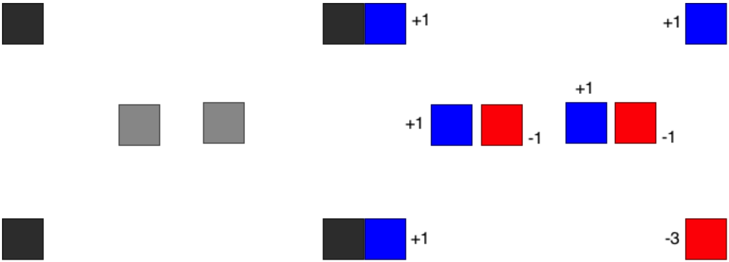
MSFBC is NP-Hard

- Steiner Problem in Graphs (SPG): $G = (V, E)$, $T \subset V$, w_e $e \in E$, find minimum weight edge set E' connecting all vertices in T.

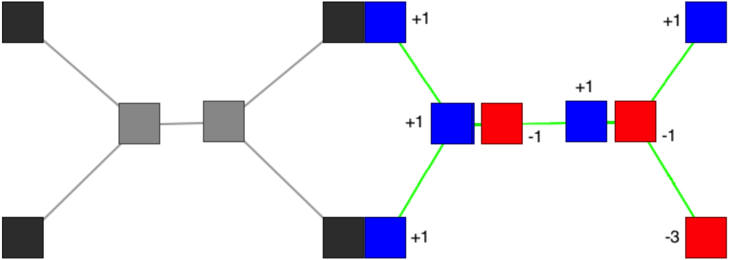
Reduction:

- Let R be V where all non-terminal vertices are doubled $|R| = 2.|V| - |T|$
- Choose a vertex $r \in T$ and assign weights p_v as follows:
  - $p_v = -1$, $\forall\ v \in T \setminus \{r\}$
  - $p_r = |T| - 1$
  - For each pair v, w of doubled non-terminal vertex assign $p_v = +1$, $p_w = -1$
- Connect all pairs of non-terminal vertices by a zero cost edge, and each of them to the other vertices as before (both when also non-terminal vertex).

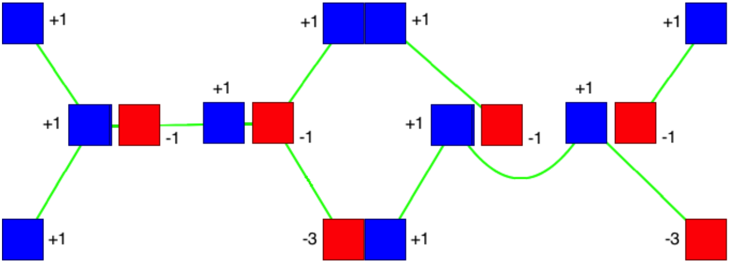
Steiner Problem in Graphs to MSFBC



Optimal Solutions SPG and MSFBC



Alternative Optimal Solutions MSFBC (Decomposition for SPG ?)



- Undirected formulation for the MSFBC:

$$\min \sum_{e \in E} d_e x_e \quad s.t. \tag{1}$$

$$\sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subset V, \quad s.t. \quad \sum_{v \in S} p_v \neq 0 \tag{2}$$

$$x_e \in \{0, 1\}, \forall e \in E, \tag{3}$$

- Directed formulation for the MSFBC:

$$\min \sum_{e \in E} d_e x_e \quad \boldsymbol{s.t.} \tag{4}$$

$$\sum_{a \in \delta^+(S)} x_a \geq 1, \quad \forall S \subset R, \quad s.t. \quad \sum_{v \in S} p_v > 0 \tag{5}$$

$$\sum_{a \in \delta^-(S)} x_a \geq 1, \quad \forall S \subset R, \quad s.t. \quad \sum_{v \in S} p_v < 0 \tag{6}$$

$$x_a + x_{a'} \leq 1, \quad \forall a = (i,j), a' = (j,i) \in E_R \tag{7}$$

$$x_a \in \{0,1\}, \forall a \in E \tag{8}$$

- Set Partitioning formulation (SPF) for the MSFBC:
  - Let J be the set of all subsets R_j of R
  - c_j is the cost of the MST connecting subset R_j, plus the cheapest connection to the terminal if R_j is not balanced

$$\min \sum_{j \in J} c_j x_j \quad \boldsymbol{s.t.} \tag{9}$$

$$\sum_{j \in J} a_{vj} x_j = 1, \quad \forall v \in R \tag{10}$$

$$x_j \in \{0,1\}, \forall j \in J, \tag{11}$$

$$a_{vj} = \begin{cases} 1 & \text{if } v \in S_j \\ 0 & \text{if } v \notin S_j \end{cases} \tag{12}$$

- Dual Heuristics: Dual Ascent over Directed Formulation
  - Dual Ascent: Selects connected components until they become balanced
  - Selection: Greedy and Random
  - Reverse delete step to generate primal solution

Dual Ascent method - Dual heuristic

input : A dual feasible solution π
output: A maximal dual feasible solution π'

Initialization: *Build* G_π = *(V, E) from the saturated arcs in* π
$\pi' \leftarrow \pi$;
while *exists a violated cut* $W \in G_\pi$ **do**
 W ← *selectViolatedCut()*;
 if $\sum_{v \in W} p_v > 0$ **then**
  Augment π'_W until at least one arc in δ^-(W) becomes saturated;
 end
 else if $\sum_{v \in W} p_v < 0$ **then**
  Augment π'_W until at least one arc in δ^+(W) becomes saturated;
 end
 Add the newly saturated arcs in G_π;
end
return π';

- Primal Heuristics - Iterated Local Search (ILS) (enumerative neighbourhoods)
  - Executed 10 times for each instance, with 25 shaking operations each.
  - simple **local improvement** procedures (swap, relocate, break, merge, break1-insert1) for each pair of trees T_1 and T_2, in random order.
  - Hybrid set covering formulation (under development).

ILS - Primal Heuristic

input : The graph $\overline{G} = (\overline{V}, \overline{E_1})$
output: A set of balanced trees S

Initialization: Generate an initial solution I by computing the minimum spanning tree for the graph $\overline{G}$ and disconnecting the set edges whose costs exceeds the average edge cost in the tree. For every unbalanced tree, connect it to its closest border point.
$It_{shake} \leftarrow 0$;
$S \leftarrow I$;
$S* \leftarrow S$;
while $It_{shake} < It_{MAX}$ **do**
  $S \leftarrow$ LocalSearch(S);
  if $c(S) < c(S*)$ **then**
    $S* \leftarrow S$;
    $It_{shake} \leftarrow 0$;
  end
  foreach It_{SC} *iterations* **do** $S \leftarrow$ SetCovering(S) **if** It_s *successive iterations without improvement* **then**
    $S \leftarrow$ Shake(S) or $S \leftarrow$ Shake($S*$) with 50% chance each;
    $It_{shake}{++}$;
    $It_s \leftarrow 0$;
  end
end
return $S*$;

- Branch-and-Cut
  - Directed formulation
  - Uses primal bounds and dual bound to fix arcs by reduced cost
- Linear Relaxation
  - Accelerates cut separation by testing whether connect components considering all edges with current $\overline{x}_a$ positive are balanced
    - Keeps current graph and runs DFS
  - Solves Max-Cut for all pairs

- Branch-and-Cut
  - Directed formulation
  - Uses primal bounds and dual bound to fix arcs by reduced cost
- Linear Relaxation
  - Accelerates cut separation by testing whether connect components considering all edges with current $\overline{x}_a$ positive are balanced
    - Keeps current graph and runs DFS
  - Solves Max-Cut for all pairs

- Branch-and-Cut
  - Directed formulation
  - Uses primal bounds and dual bound to fix arcs by reduced cost
- Linear Relaxation
  - Accelerates cut separation by testing whether connect components considering all edges with current $\overline{x}_a$ positive are balanced
    - Keeps current graph and runs DFS
  - Solves Max-Cut for all pairs

input : The undirected graph $\overline{G_1} = (\overline{V}, \overline{E_1})$ and the directed graph $\overline{G_2} = (\overline{V}, \overline{E_2})$, both described by the solution set **x**
output: The set of cuts **S**

Initialization: *For each node v in* $\overline{V}$*, assign visited[v]* $\leftarrow$ *false. Let C be the list of connected components and S the list of cuts.*

```
foreach node v in V̄ do
    if !visited[v] then
        Run a depth-first-search on graph Ḡ1, starting from node
        v. Add the discovered connected component in C.
    end
end
foreach connected component c_k ∈ C do
    foreach pair of nodes (i,j) ∈ c_k do
        {s', MaxFlow} ← minCutMaxFlow(i,j,Ḡ2);
        if s' is unbalanced, MaxFlow < 1 and s' ∉ S then
            S ← S + s'
        end
    end
    foreach node i ∈ c_k and a single node j ∉ c_k do
        s' ← minCutMaxFlow(i,j,Ḡ2);
        if s' is unbalanced, MaxFlow < 1 and s' ∉ S then S ← S
        + s'
    end
    if s(c_k) is unbalanced and s(c_k) ∉ S then S ← S + s(c_k)
end
return S;
```

- Solving Set Partitioning's Linear Relaxation
  - Column Generation subproblem:
    - Minimum Tree with Profits on Vertices: too hard
  - Relaxation for the SPF:
    - Based on Uchoa et al.(2008)'s $q - arbor$
    - Find a minimum cost connected component with limited number of vertices and vertex degree ($q - arbor$)

- Solving SPF's relaxed Linear Relaxation
  - Let $T(v, a, q)$ be the minimum cost of a $q - arbor$ with root in vertex v where v has *arity* d and the total $q - arbor$ has q vertices.
  - Let $inCost(v, a, q)$ be the cost without connecting the to the terminal
  - Let $c2terminal(v, a, q)$ be the cost to connect to the terminal.
  - Finally let $p(v, a, q)$ be the balance of $q - arbor$ (a, v, q).

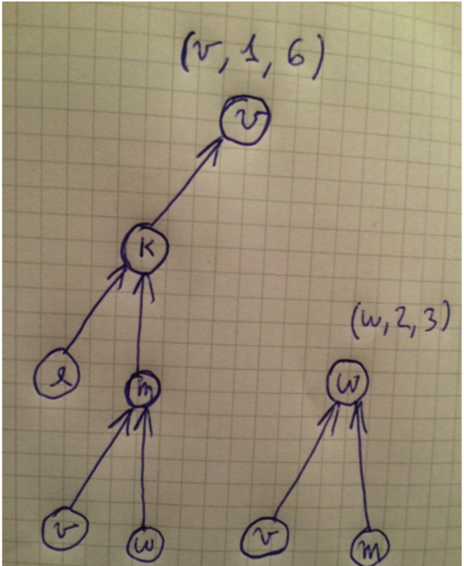
- Solving SPF's relaxed Linear Relaxation
  - The dynamic programming recursion follows:

$$T(v,a,q) = \min_{w,ar,q2}\{inCost(v,a-1,q1)+inCost(w,ar,q2)+d_{vw}+\Delta\}$$

where

- the minimum is taken over, $w \neq v$, all ar and with $q1, q2$ such that $q1 + q2 = q$;
- Δ is equal to zero, when $p(v,a-1,q1) + p(w,ar,q2) == 0$ and to the minimum of $c2terminal(v,a,q1)$ and $c2terminal(w,ar,q2)$, otherwise.

$q-arbor$ example



Instances

- MSFBC - How good are we ?
    - Preliminary Tests on Random Instances
        - Equal number of positive and negative residues
        - Report on instances with up to 512 pairs of vertices

PrimaL Heuristic (ILS)

Instance	|V|	|E|	Best Sol. (out	Avg Sol. (out	GAP (%)	T (s, per run)
PUC_p_48_4	96	9312	1436.65	1476.02	2.67	209.85
PUC_p_48_4	96	9312	1680.83	1724.54	2.53	196.93
PUC_p_48_4	96	9312	1555.64	1600.97	2.83	202.94
PUC_p_48_4	96	9312	1564.42	1602.36	2.37	197.23
PUC_p_48_4	96	9312	1498.14	1551.57	3.44	221.07
PUC_p_64_6	128	16512	2602.29	2649.67	1.79	433.66
PUC_p_64_6	128	16512	1567.21	2365.56	33.75	450.80
PUC_p_64_6	128	16512	2576.42	2642.64	2.51	407.71
PUC_p_64_6	128	16512	2461.56	2519.28	2.29	435.59
PUC_p_64_6	128	16512	1618.22	2561.53	36.83	455.26
PUC_p_128_	256	65792	7780.95	8031.96	3.13	888.87
PUC_p_128_	256	65792	8499.00	8631.70	1.54	885.38
PUC_p_128_	256	65792	7824.09	8053.78	2.85	860.46
PUC_p_128_	256	65792	8046.53	8261.84	2.61	864.26
PUC_p_128_	256	65792	8038.89	8225.87	2.27	800.27
PUC_p_256_	512	262656	26140.37	26191.58	0.20	1315.52
PUC_p_256_	512	262656	23978.25	25800.13	7.06	1423.43
PUC_p_256_	512	262656	25420.15	25620.36	0.78	1408.63
PUC_p_256_	512	262656	24726.65	25999.47	4.90	1546.93
PUC_p_256_	512	262656	23672.33	25248.97	6.24	1304.52
PUC_p_512_	1024	1048576	69526.20	72111.25	3.58	2342.23
PUC_p_512_	1024	1048576	65418.52	68442.63	4.42	2011.24
PUC_p_512_	1024	1048576	65652.68	66592.54	1.41	2245.45
PUC_p_512_	1024	1048576	66275.43	69823.12	5.08	2111.44
PUC_p_512_	1024	1048576	67077.52	68022.14	1.39	2564.57

Hybrid PrimaL Heuristic (SCF)

Instance	V	E	Best sol.	Avg sol	Best SC	Avg SC	Time per Run (s)
PUC_p_4_4_	8	72	24.70	24.90	24.70	24.70	0.01
PUC_p_6_6_	12	156	54.54	54.54	54.54	54.54	0.03
PUC_p_8_8_	16	272	79.38	79.38	79.38	80.72	0.08
PUC_p_10_1	20	420	128.16	128.31	128.16	128.26	0.11
PUC_p_12_1	24	600	183.24	187.03	183.24	192.28	0.17
PUC_p_14_1	28	812	203.86	210.47	203.86	226.58	0.36
PUC_p_16_1	32	1056	259.67	260.65	259.67	262.09	0.78
PUC_p_18_1	36	1332	297.53	297.85	297.72	299.43	0.76
PUC_p_20_2	40	1640	347.49	354.32	349.15	360.85	1.32
PUC_p_22_2	44	1980	462.14	469.39	477.60	485.51	2.66
PUC_p_24_2	48	2352	473.14	477.86	467.20	474.19	2.68
PUC_p_26_2	52	2756	582.54	595.55	592.94	597.61	3.67
PUC_p_28_2	56	3192	640.93	660.13	642.15	649.54	7.17
PUC_p_30_3	60	3660	651.59	670.85	654.79	675.77	4.53
PUC_p_32_3	64	4160	819.67	836.87	846.23	862.71	11.18
PUC_p_40_4	80	6480	1175.03	1240.54	1105.99	1108.74	6.58
PUC_p_48_4	96	9312	1436.65	1476.02	1416.58	1445.14	41.69
PUC_p_64_6	128	16512	2602.29	2649.67	2542.01	2613.35	77.86

Branch-and-Cut (Directed)

Instance	|V|	|E|	R	P*	Root LP	LB	UB	GAP	N	D	T(s)	FS(%)	PtPS(%)
PUC_p_26_26_1	52	2756	225	582.54	555.04	574.00	574.00	0	509	19	8.08	0.69	31.38
PUC_p_26_26_2	52	2756	59	516.55	515.26	515.26	515.26	0	1	0	0.03	0.23	13.95
PUC_p_26_26_3	52	2756	473	594.35	591.54	594.35	594.35	0	11	3	0.11	0.83	35.25
PUC_p_26_26_4	52	2756	868	472.59	460.84	472.59	472.59	0	23	4	0.49	0.58	26.45
PUC_p_26_26_5	52	2756	314	585.80	583.15	585.80	585.80	0	7	3	0.11	0.71	54.72
PUC_p_28_28_1	56	3192	637	640.93	625.61	640.93	640.93	0	9	3	0.20	0.36	42.70
PUC_p_28_28_2	56	3192	2156	653.34	618.71	631.74	631.74	0	3351	33	152.31	0.34	48.59
PUC_p_28_28_3	56	3192	446	594.36	592.02	594.23	594.23	0	13	5	0.24	0.78	60.18
PUC_p_28_28_4	56	3192	726	580.51	565.03	570.74	570.74	0	19	5	0.29	0.70	26.20
PUC_p_28_28_5	56	3192	62	556.62	544.91	544.91	544.91	0	1	0	0.04	0.16	8.00
PUC_p_30_30_1	60	3660	298	651.59	630.13	639.40	639.40	0	219	16	4.97	0.71	45.59
PUC_p_30_30_2	60	3660	915	722.47	705.39	722.47	722.47	0	345	14	10.81	0.40	62.84
PUC_p_30_30_3	60	3660	588	753.17	729.39	732.90	732.90	0	7	2	0.15	0.67	47.42
PUC_p_30_30_4	60	3660	2004	668.18	660.78	668.18	668.18	0	111	9	3.87	0.33	22.47
PUC_p_30_30_5	60	3660	1213	725.86	693.54	706.65	706.65	0	83	11	2.05	0.40	20.80
PUC_p_32_32_1	64	4160	1743	819.67	769.78	788.83	788.83	0	2403	23	153.35	0.22	28.96
PUC_p_32_32_2	64	4160	333	748.66	746.04	748.66	748.66	0	7	3	0.19	0.51	72.19
PUC_p_32_32_3	64	4160	407	772.59	754.85	754.85	754.85	0	3	1	0.06	0.70	63.29
PUC_p_32_32_4	64	4160	1807	810.72	786.56	805.77	805.77	0	8689	27	476.69	0.38	53.61
PUC_p_32_32_5	64	4160	990	819.68	773.01	789.85	789.85	0	519	22	14.42	0.60	58.37
PUC_p_40_40_1	80	6480	2561	1175.03	1055.60	1076.83	1076.83	0	159	17	9.72	0.37	35.37
PUC_p_40_40_2	80	6480	3785	1149.24	1044.58	1075.81	1075.81	0	38033	37	3841.80	0.44	49.03
PUC_p_40_40_3	80	6480	5089	1179.58	1024.10	1048.91	1048.91	0	11647	33	1129.92	0.13	9.35
PUC_p_40_40_4	80	6480	4436	1202.98	1100.40	1114.46	1114.46	0	4075	28	291.18	0.19	28.20
PUC_p_40_40_5	80	6480	972	1244.58	1102.94	1122.35	1122.35	0	255	13	15.21	0.53	71.08

Branch-and-Cut (Directed)

Instance	|V|	|E|	R	P*	Root LP	LB	UB	GAP	Nodes	Depth	T(s)	FS(%)	PtPS(%)
PUC_p_48_48_1	96	9312	913	1436.65	1338.59	1348.94	1348.94	0.00	2139	28	110.45	0.52	64.55
PUC_p_48_48_2	96	9312	8460	1680.83	1558.17	1677.91	1680.83	2.91	26356	63	6000.00	0.19	60.49
PUC_p_48_48_3	96	9312	6988	1555.64	1467.52	1532.53	1555.64	23.11	35299	47	6000.00	0.23	45.32
PUC_p_48_48_4	96	9312	7043	1564.42	1450.21	1487.52	1564.42	76.90	32970	53	6000.00	0.17	35.55
PUC_p_48_48_5	96	9312	801	1498.14	1420.79	1422.95	1422.95	0.00	5	2	0.18	0.49	64.95
PUC_p_64_64_1	128	16512	13608	2602.29	2356.62	2548.34	2602.29	53.94	12641	70	6000.00	0.13	25.81
PUC_p_64_64_2	128	16512	1634	2201.51	2134.44	2166.77	2166.77	0.00	4311	26	657.49	0.37	65.69
PUC_p_64_64_3	128	16512	6756	2576.42	2340.01	2460.77	2536.14	75.37	26396	50	6000.00	0.31	43.09
PUC_p_64_64_4	128	16512	13690	2546.46	2183.65	2367.67	2461.56	93.89	12463	66	6000.00	0.12	29.23
PUC_p_64_64_5	128	16512	1197	2311.76	2258.86	2269.26	2269.26	0.00	283	17	52.68	0.26	62.09
PUC_p_128_128_1	256	65792	65474	7780.95	6483.48	7287.54	7643.75	356.21	896	136	6000.00	0.03	53.20
PUC_p_128_128_2	256	65792	65327	8499.00	6867.98	7857.44	8008.76	151.32	1036	164	6000.00	0.05	43.10
PUC_p_128_128_3	256	65792	65792	7824.09	6437.93	7442.55	7824.09	381.54	958	159	6000.00	0.05	30.39
PUC_p_128_128_4	256	65792	61682	8046.53	6643.14	7411.02	7530.24	119.22	1310	139	6000.00	0.04	13.90
PUC_p_128_128_5	256	65792	61820	8038.89	6660.02	7435.86	8038.89	603.03	1085	130	6000.00	0.05	29.57
PUC_p_256_256_1	512	262656	262656	22777.65	19305.00	19760.60	22777.65	3017.04	59	58	6000.00	0.01	13.00
PUC_p_256_256_2	512	262656	262656	23427.66	19383.70	19742.80	23427.66	3684.85	58	57	6000.00	0.01	31.36
PUC_p_256_256_3	512	262656	262656	22562.60	18741.29	19231.87	22562.60	3330.73	59	58	6000.00	0.01	29.41
PUC_p_256_256_4	512	262656	262656	23436.48	19100.23	19649.88	23436.48	3786.59	65	64	6000.00	0.01	8.21
PUC_p_256_256_5	512	262656	262656	23087.67	19179.96	19793.05	23087.67	3294.61	65	64	6000.00	0.01	13.22
PUC_p_512_512_1	1024	1049600	1049600	69526.20	55306.06	55340.47	69526.20	14185.73	4	3	6000.00	0.00	12.65
PUC_p_512_512_2	1024	1049600	1049600	65418.52	54360.60	54543.15	65418.52	10875.37	3	2	6000.00	0.00	6.63
PUC_p_512_512_3	1024	1049600	1049600	65652.68	54354.11	54387.78	65652.68	11264.90	3	2	6000.00	0.00	28.73
PUC_p_512_512_4	1024	1049600	1049600	66275.43	53175.93	53177.49	66275.43	13097.94	4	3	6000.00	0.00	12.71
PUC_p_512_512_5	1024	1049600	1049600	67077.52	54478.71	54513.88	67077.52	12563.64	4	3	6000.00	0.00	17.06

Signal Processing: 2D Phase Unwrapping

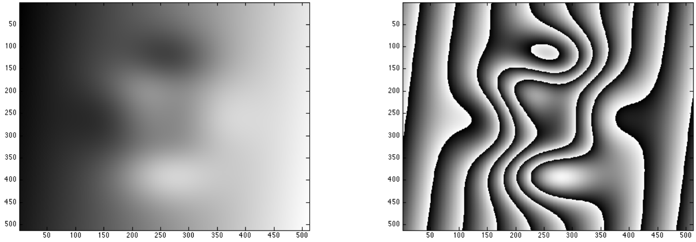


Figure: *Wrapping effect on a 2D phase image.*
(a) Absolute phase image
(b) Wrapped phase image.

Signal Processing: 2D Phase Unwrapping

MSFBC as a new model for 2D Phase Unwrapping

- Does a solution with a smaller value for the MSFBC implies a better unwrapping ?

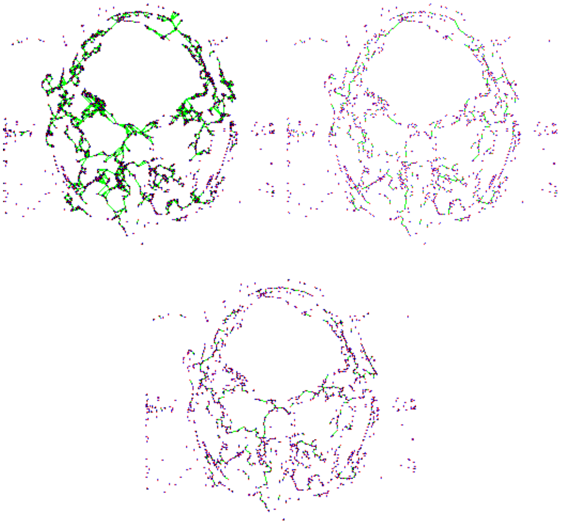


Figure: *Head. (a) Goldstein (b) Matching (c) MSFBC*

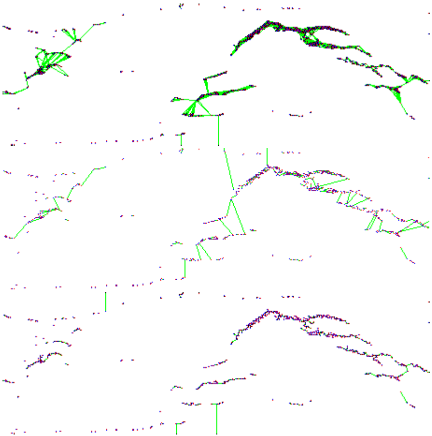


Figure: *Long (a) Goldstein (b) Matching (c) MSFBC*

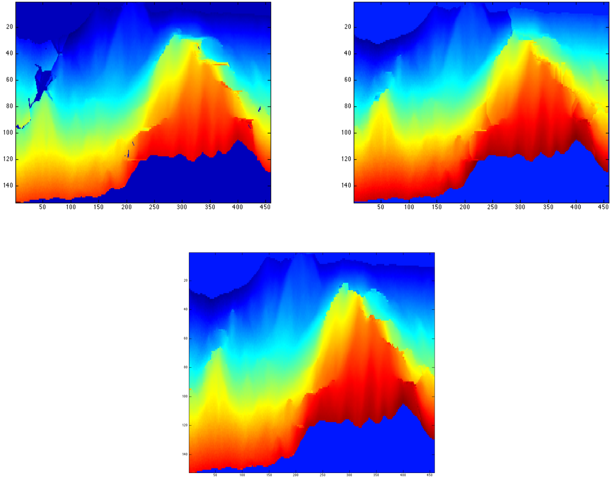


Figure: *Long (a) Goldstein* *(b) Matching* *(c) MSFBC*

Future Work

- Benchmark with tailored instances
- Gaps appear to be way larger than for Steiner: polyhedral study
- Column Generation formulation:
  - How good would be SPF bounds for transformed Steiner instances?
  - What would be an $ng - route$ equivalent SPF relaxation?
  - Would Subset Row Cuts be effective?
  - Limited memory SRC's

Future Work

- Benchmark with tailored instances
- Gaps appear to be way larger than for Steiner: polyhedral study
- Column Generation formulation:
  - How good would be SPF bounds for transformed Steiner instances?
  - What would be an $ng - route$ equivalent SPF relaxation?
  - Would Subset Row Cuts be effective?
  - Limited memory SRC's

Future Work

- Benchmark with tailored instances
- Gaps appear to be way larger than for Steiner: polyhedral study
- Column Generation formulation:
  - How good would be SPF bounds for transformed Steiner instances?
  - What would be an $ng - route$ equivalent SPF relaxation?
  - Would Subset Row Cuts be effective?
  - Limited memory SRC's

Future Work

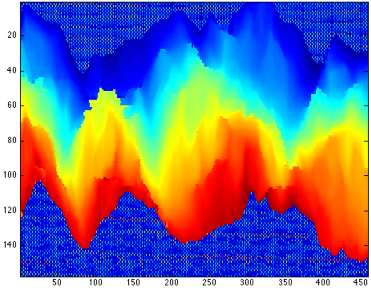
- Benchmark with tailored instances
- Gaps appear to be way larger than for Steiner: polyhedral study
- Column Generation formulation:
  - How good would be SPF bounds for transformed Steiner instances?
  - What would be an $ng - route$ equivalent SPF relaxation?
  - Would Subset Row Cuts be effective?
  - Limited memory SRC's

Future Work

- Benchmark with tailored instances
- Gaps appear to be way larger than for Steiner: polyhedral study
- Column Generation formulation:
  - How good would be SPF bounds for transformed Steiner instances?
  - What would be an $ng - route$ equivalent SPF relaxation?
  - Would Subset Row Cuts be effective?
  - Limited memory SRC's

Future Work

- Benchmark with tailored instances
- Gaps appear to be way larger than for Steiner: polyhedral study
- Column Generation formulation:
  - How good would be SPF bounds for transformed Steiner instances?
  - What would be an $ng - route$ equivalent SPF relaxation?
  - Would Subset Row Cuts be effective?
  - Limited memory SRC's



Thanks!