PROXIMITY SEARCH

DUSI MICHELE (717462)

SALVALAI MATTEO (715827)

VENTURINI PIETRO (715166)

Optimization Algorithms – A.A. 2019/2020

MODEL OF THE PROBLEM

Objective Function $\min f(x)$

Constraints

$$g(x) \leq 0$$

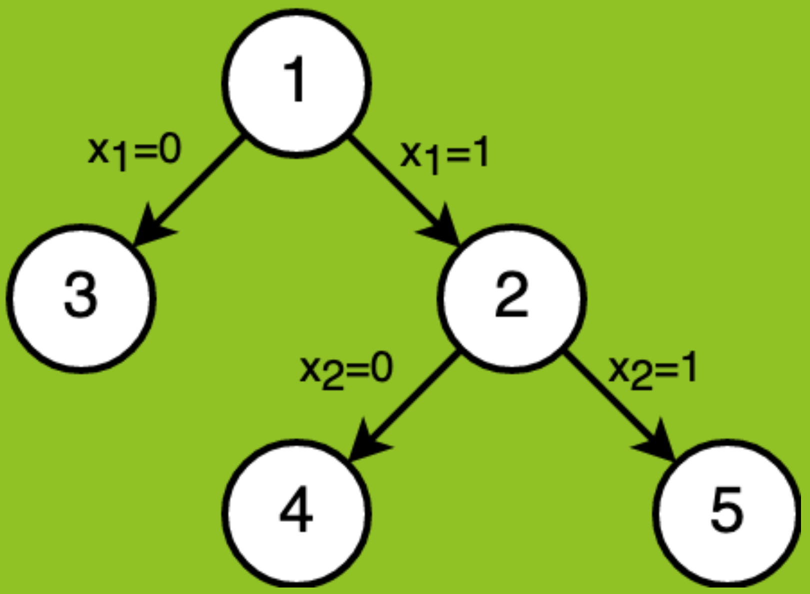
$$x_j \in \{0,1\}\ \forall j \in J$$

$f: \mathbb{R}^n \to \mathbb{R}$ $\quad$ $g: \mathbb{R}^n \to \mathbb{R}^m$ $\quad$ $J \subseteq N \coloneqq \{1 \dots n\}$

TYPICAL APPROACH

STRATEGY USED: Best Bound First

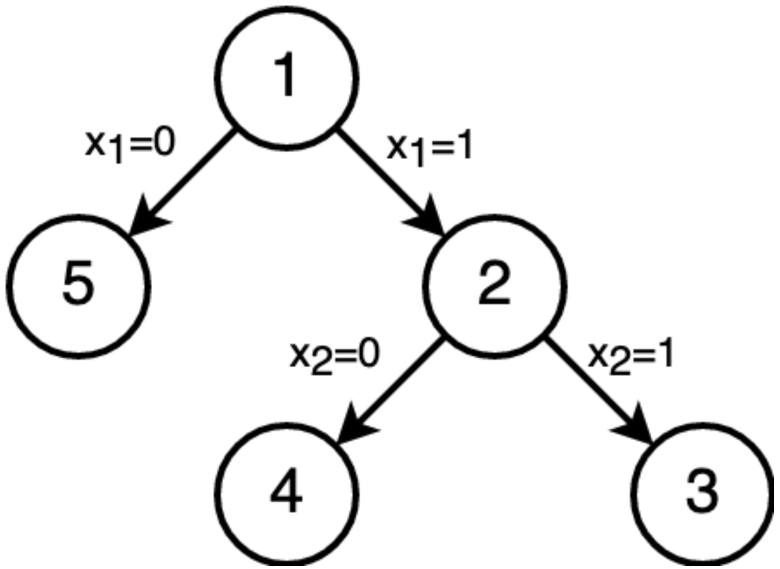
DRAWBACK: search tends to be initially trapped in the upper part of the tree, where lower bounds are smaller, which is counterproductive in terms of probability of updating the incumbent.



PROXIMITY SEARCH
APPROACH

MIXED STRATEGY:

1. Selecting the next node to elaborate (e.g. using best-bound first)
2. Making a sequence of diving branching steps by visiting the nodes at increasing distance from the root.



STEP 1

Finding an initial feasible solution $\tilde{x}$ by:

- An ad-hoc heuristic
- Running a black-box MIP solver

STEP 2

Adding an explicit cutoff constraint $(\theta > 0)$

$$f(x) \leq f(\tilde{x}) - \theta$$

STEP 3

Replacing the objective function by a proximity one

(e.g. the Hamming distance):

$$\Delta(x, \tilde{x}) := \sum_{j \in J : \tilde{x}_j = 0} x_j + \sum_{j \in J : \tilde{x}_j = 1} (1 - x_j)$$

Example

$$\tilde{x} = (101001)$$
$$x = (011101)$$
$$\Delta(x, \tilde{x}) = 3$$

STEP 4

Run the black-box MIP solver to hopefully find a new incumbent x^* with $f(x^*) \leq f(\tilde{x}) - \theta$

STEP 5

The new solution x^* is possibly improved by solving a convex problem where all the binary variables have been fixed to their value in x^*

STEP 6

Recenter $\Delta(x, \cdot)$ by setting $\tilde{x} := x^*$, and/or update θ

THE ALGORITHM

`Proximity Search`:

1. let $\tilde{x}$ be the initial heuristic feasible solution to refine;
   repeat
2. explicitly add the *cutoff constraint* $f(x) \leq f(\tilde{x}) - \theta$ to the MIP model;
3. replace $f(x)$ by the "proximity" objective function $\Delta(x, \tilde{x})$;
4. run the MIP solver on the new model until a termination condition is reached, and let x^* be the best feasible solution found (x^* empty if none);
   if x^* is nonempty and $J \subset N$ **then**
5. refine x^* by solving the convex program
   $$x^* := \text{argmin}\{f(x) : g(x) \leq 0,\ x_j = x_j^*\ \forall j \in J\}$$
   end
6. recenter $\Delta(x, \cdot)$ by setting $\tilde{x} := x^*$, and/or update θ
   until *an overall termination condition is reached*;

Proximity Search **Variants**

Variants

Possible **alternative implementations** of the basic algorithm:

Proximity Search without recentering

Proximity objective function never changes

Proximity Search with recentering

Proximity objective function changes when a new incumbent is found

Proximity Search with an incumbent

Each new incumbent triggers the MIP solver internal refinement heuristic

Proximity Search **without recentering** (proxy_norec)

Features

- Each new incumbent $\bar{x}$ is declared unfeasible
- New cutoff constraint:
  $f(x) \leq f(\bar{x}) - \theta$

Drawbacks

The MIP incumbent is never updated explicitly →

- *No propagation scheme*
- *No variable-fixing scheme*
- *No refinement heuristic*

Proximity Search without recentering: algorithm

Proximity Search, no recentering (proxy_norec):

1. let $\tilde{x}$ be the initial heuristic feasible solution to refine;
2. explicitly add the *cutoff constraint* $f(x) \leq f(\tilde{x}) - \theta$ to the MIP model;
3. replace $f(x)$ by the "proximity" objective function $\Delta(x, \tilde{x})$;
4. run the MIP solver on the new model until a termination condition is reached or no more feasible solutions are found;

Callback (*when a new incumbent is discovered*):

5. Let $\bar{x}$ be the discovered incumbent, record $\bar{x}$;
6. Introduce the *new cutoff constraint* $f(x) \leq f(\bar{x}) - \theta$;
7. *refinement of the current incumbent;*

Proximity Search with recentering (proxy_rec)

Features

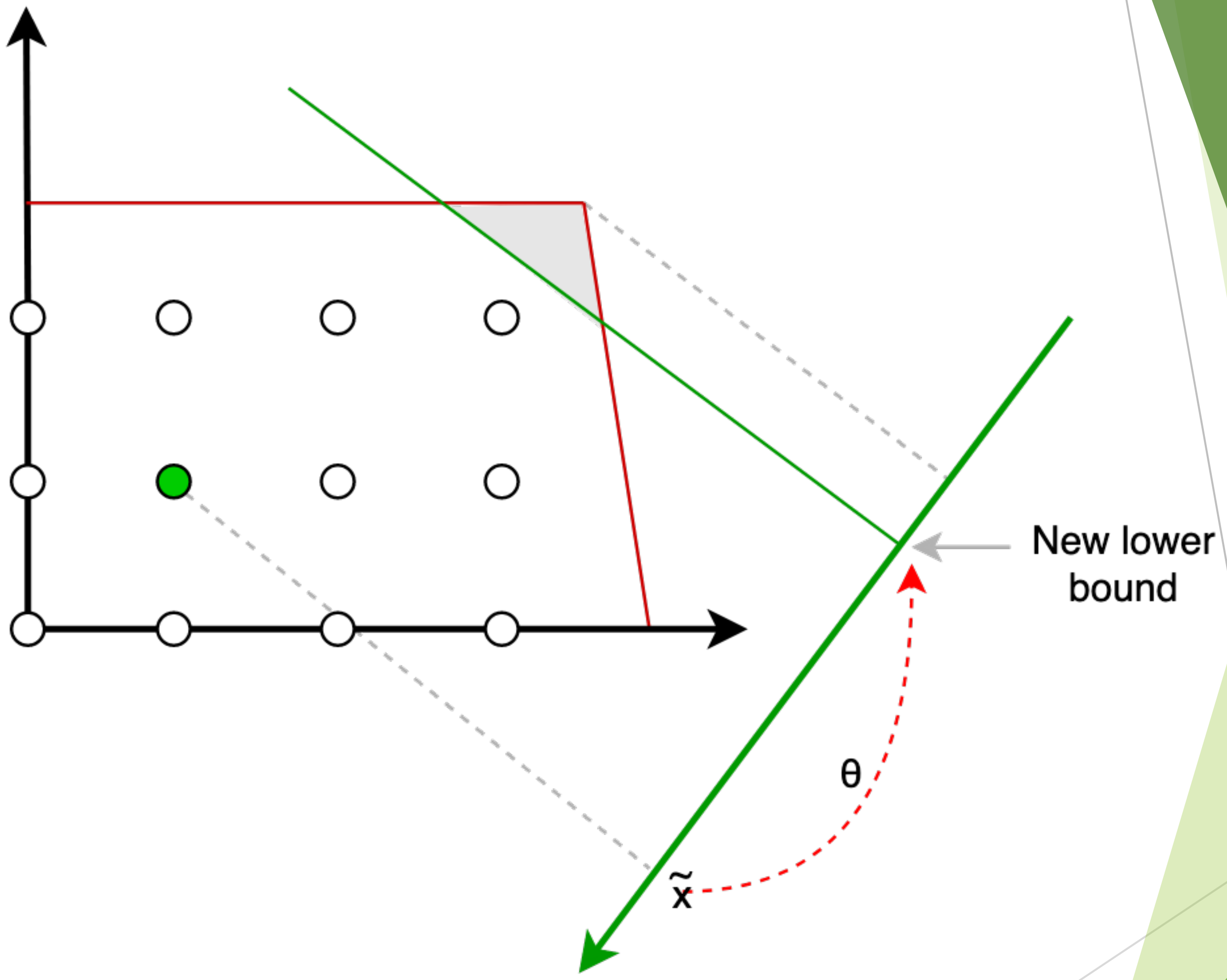
- Each new incumbent is the center of the proximity objective function $\Delta(x, \cdot)$
- MIP solver as a Black Box

Drawbacks

The MIP incumbent is never updated explicitly →

- No propagation scheme
- No variable-fixing scheme
- No refinement heuristic

Proximity Search with recentering: **property**



θ-optimality of the solution

Proximity Search with recentering: algorithm

`Proximity Search, with recentering (proxy_rec):`

1. let $\tilde{x}$ be the initial heuristic feasible solution to refine;

   repeat

2. explicitly add the *cutoff constraint* $f(x) \leq f(\tilde{x}) - \theta$ to the MIP model;
3. replace $f(x)$ by the "proximity" objective function $\Delta(x, \tilde{x})$;
4. run the MIP solver on the new model until the first feasible solution is found and let it be x^* (x^* empty if none);
5. *refinement of the current incumbent* x^* *;*
6. recenter $\Delta(x, \cdot)$ by setting $\tilde{x} := x^*$;

   until *an overall termination condition is reached;*

Proximity Search with an incumbent (proxy_incum)

Features

- New proximity objective function:

$$\Delta(x, \cdot) + M z \qquad M > 0$$

- Soft cutoff constraints:

$$f(x) \leq f(\tilde{x}) - \theta + z \qquad z \geq 0$$

Drawbacks

A too aggressive (large) θ would make this approach useless:

$$\theta = f(\tilde{x}) \longrightarrow z = f(x)$$

Proximity Search with an incumbent: algorithm

Proximity Search, with an incumbent (proxy_incum):

1. *let $\tilde{x}$ be the initial heuristic feasible solution to refine;*

   repeat

2. *explicitly add the soft cutoff constraint $f(x) \leq f(\tilde{x}) - \theta + z$ to the MIP model;*
3. *replace $f(x)$ by the "proximity" objective function $\Delta(x, \tilde{x}) + M z$ ($M > 0$);*
4. *run the MIP solver on the new model until the first feasible solution is found and let it be x^* (x^* empty if none);*
5. *refinement of the current incumbent x^* ;*
6. *recenter $\Delta(x, \cdot)$ by setting $\tilde{x} := x^*$;*

   ***until** an overall termination condition is reached;*

Trivial example (1)

Let's consider the Proximity Search with recentering applicated on a trivial MIP problem:

$$\min 5x_1 + 3x_2 + 4x_3 + 2x_4$$
$$x_1 + x_2 \geq 1$$
$$x_4 \geq 7x_1$$
$$x_4 \geq 8x_2$$
$$x_i \in \{0;1\} \quad i=1,2,3$$
$$x_4 \geq 0$$

Suppose to have the initial (feasible) solution $\tilde{x}$ = (1,1,1,8)
Set θ = 2
Use the Hamming Distance as "proximity" objective function:

$$\Delta(x, \tilde{x}) = 3 - x_1 - x_2 - x_3$$

The objective function in $\tilde{x}$ has value: $f(\tilde{x})$ = 28

Trivial example (2)

The model after Step 1 and 2 of the algorithm:

$$\min \boxed{3 - x_1 - x_2 - x_3}$$ Hamming Distance

$$\boxed{5x_1 + 3x_2 + 4x_3 + 2x_4 \leq 28 - 2}$$ Cutoff constraint

$$x_1 + x_2 \geq 1$$

$$x_4 \geq 7x_1$$

$$x_4 \geq 8x_2$$

$$x_i \in \{0;1\} \quad i=1,2,3$$

$$x_4 \geq 0$$

$\Delta(x, \tilde{x}) = 1$: (0, 1, 1, 8) -> $f(x^*) = 23 \leq 26$ -> feasible solution
(1, 0, 1, 8)
(1, 1, 0, 8)

x^*= (0, 1, 1, 8) is the new incumbent, it will be the new center $\tilde{x}$

Trivial example (3)

The model, recentered, after Step 1 and 2 of the algorithm:

$$\min \boxed{x_1 + 2 - x_2 - x_3}$$ Recentered Hamming Distance

$$\boxed{5x_1 + 3x_2 + 4x_3 + 2x_4 \leq 26}$$ Previous cutoff constraint

$$\boxed{5x_1 + 3x_2 + 4x_3 + 2x_4 \leq 23 - 2}$$ New cutoff constraint

$$x_1 + x_2 \geq 1$$

$$x_4 \geq 7x_1$$

$$x_4 \geq 8x_2$$

$$x_i \in \{0;1\} \quad i=1,2,3$$

$$x_4 \geq 0$$

Performances and **comparisons**

Related approaches in the literature

The **ideas** from which the Proximity Search takes its success are not new in the literature:

Replacing constraints with penalizations

Augmented Lagrangian (Hestenes, Powell, 1969)

Replacing the objective function with a distance-related function

Parametric Branch and Bound Scheme (Glover, 1978)

Feasability Pump (Fischetti, Glover, Lodi, 2005 - in the interpretation of Boland, 2012)

Using a cutoff constraint

Parametric Tabu Search (Glover, 2006)

Related approaches in the literature

The **ideas** from which the Proximity Search takes its success are not new in the literature:

Solving subproblems with fixed variables

Relaxation Induced Neighborhood Search
(Danna, Rothberg, Le Pape, 2005)

Inserting a distance constraint

Local Branching
(Fischetti, Lodi, 2003)

So, is Proximity Search a good choice?

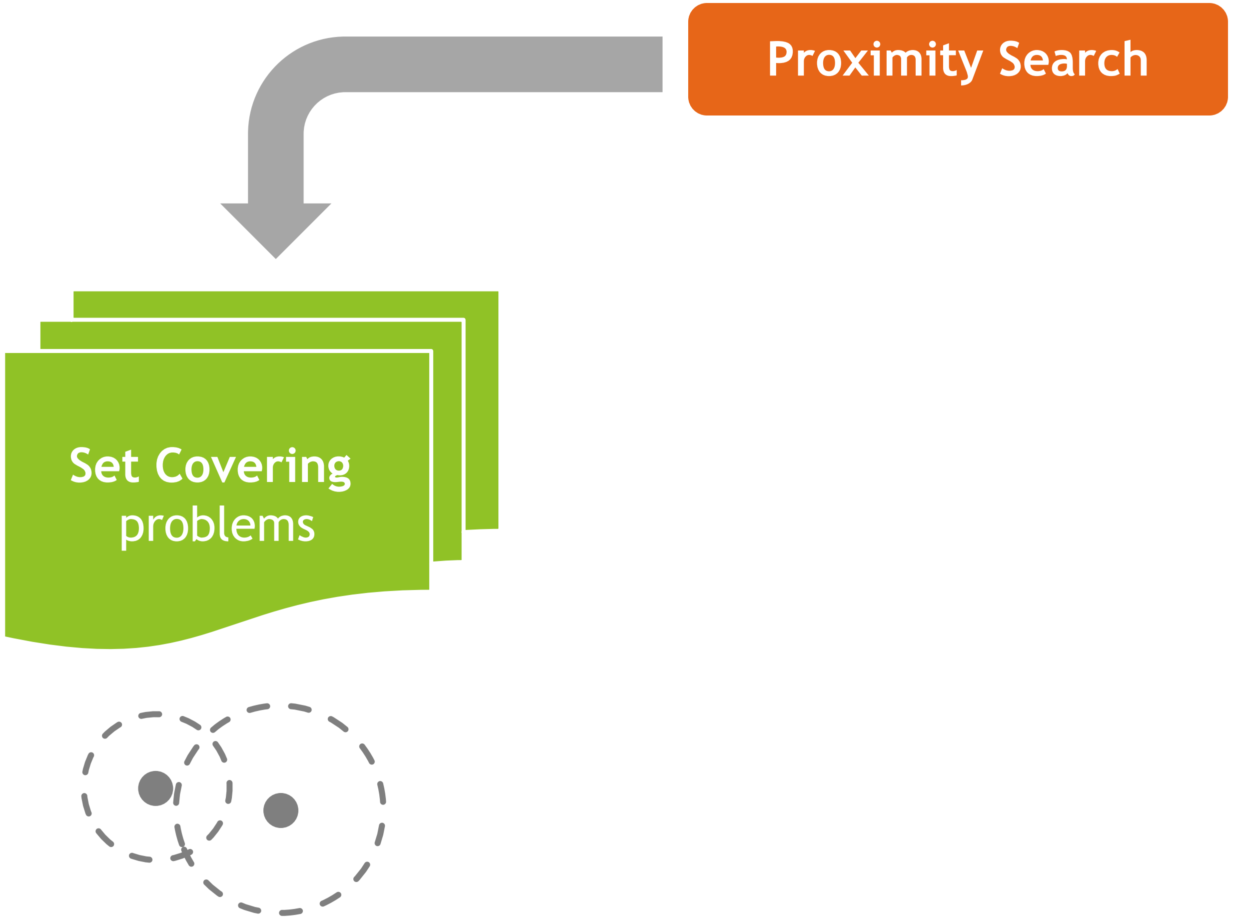
Computational **Results**

In the original article, Proximity Search has been **tested** on three different classes of problems:

Proximity Search

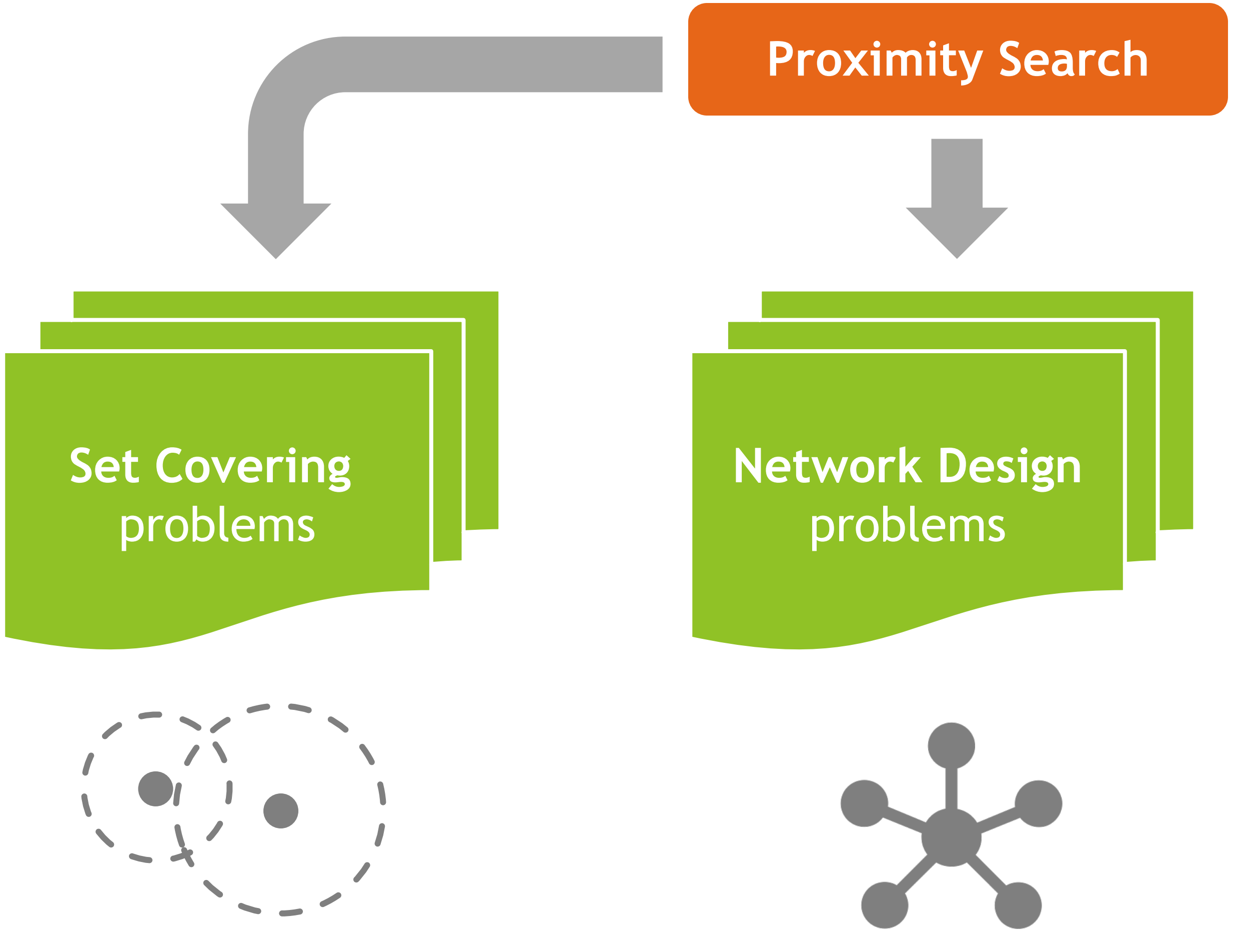
Computational **Results**

In the original article, Proximity Search has been **tested** on three different classes of problems:



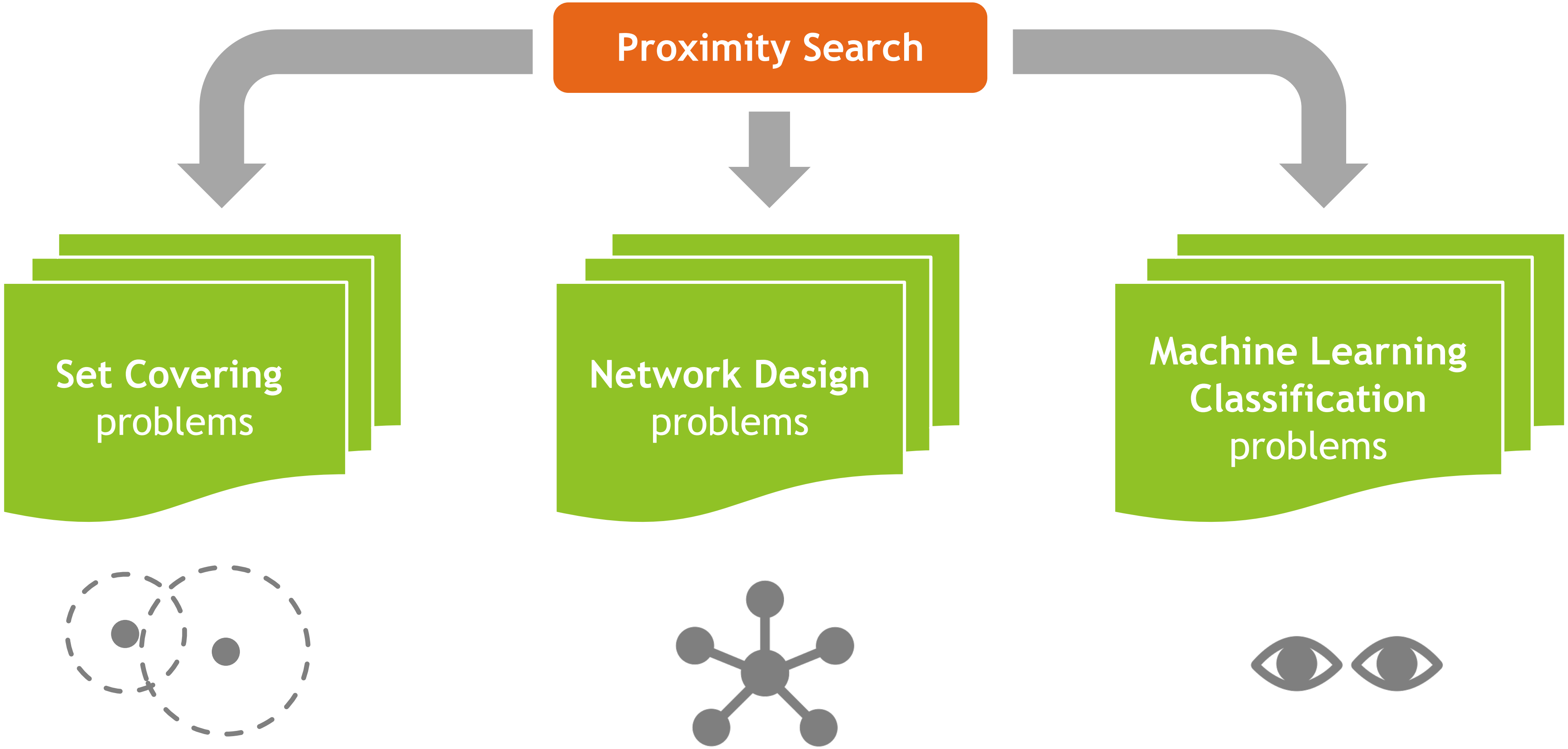
Computational **Results**

In the original article, Proximity Search has been **tested** on three different classes of problems:



Computational **Results**

In the original article, Proximity Search has been **tested** on three different classes of problems:



Set Covering **Performances**

algorithms	time limit (s)							
	5	10	30	60	120	300	600	1,200
Set covering instances								
proxy_norec	0.132	0.215	0.452	0.703	1.090	1.886	2.851	4.247
cplex_def	0.178	0.310	0.698	1.121	1.753	2.880	4.108	5.775
cplex_heu	0.174	0.305	0.703	1.113	1.671	2.697	3.774	5.086
cplex_no_cuts	0.176	0.305	0.694	1.138	1.760	2.865	3.949	5.301
cplex_gui_div	0.175	0.297	0.651	1.031	1.594	2.605	3.565	4.750
proxy_incum	0.124	0.195	0.374	0.550	0.797	1.232	1.600	1.978
proxy_rec	0.122	0.198	0.400	0.599	0.858	1.335	1.749	2.182
locBra_orig	0.170	0.278	0.551	0.803	1.122	1.722	2.304	2.900
locBra_aggr	0.121	0.192	0.376	0.561	0.773	1.157	1.533	1.974
cplex_polish	0.181	0.298	0.596	0.876	1.251	1.895	2.498	3.252

Comparison metric: geometric mean of **primal integral** (the lower, the better)

Set Covering **Performances**

	time limit (s)							
	5	10	30	60	120	300	600	1,200
Set covering instances								
proxy_norec	0.132	0.215	0.452	0.703	1.090	1.886	2.851	4.247
cplex_def	0.178	0.310	0.698	1.121	1.753	2.880	4.108	5.775
cplex_heu	0.174	0.305	0.703	1.113	1.671	2.697	3.774	5.086
cplex_no_cuts	0.176	0.305	0.694	1.138	1.760	2.865	3.949	5.301
cplex_gui_div	0.175	0.297	0.651	1.031	1.594	2.605	3.565	4.750
proxy_incum	0.124	0.195	0.374	0.550	0.797	1.232	1.600	1.978
proxy_rec	0.122	0.198	0.400	0.599	0.858	1.335	1.749	2.182
locBra_orig	0.170	0.278	0.551	0.803	1.122	1.722	2.304	2.900
locBra_aggr	0.121	0.192	0.376	0.561	0.773	1.157	1.533	1.974
cplex_polish	0.181	0.298	0.596	0.876	1.251	1.895	2.498	3.252

algorithms

Proximity Search
(three variants)

Comparison metric: geometric mean of **primal integral** (the lower, the better)

Set Covering **Performances**

algorithms	time limit (s)							
	5	10	30	60	120	300	600	1,200
Set covering instances								
proxy_norec	0.132	0.215	0.452	0.703	1.090	1.886	2.851	4.247
cplex_def	0.178	0.310	0.698	1.121	1.753	2.880	4.108	5.775
cplex_heu	0.174	0.305	0.703	1.113	1.671	2.697	3.774	5.086
cplex_no_cuts	0.176	0.305	0.694	1.138	1.760	2.865	3.949	5.301
cplex_gui_div	0.175	0.297	0.651	1.031	1.594	2.605	3.565	4.750
proxy_incum	0.124	0.195	0.374	0.550	0.797	1.232	1.600	1.978
proxy_rec	0.122	0.198	0.400	0.599	0.858	1.335	1.749	2.182
locBra_orig	0.170	0.278	0.551	0.803	1.122	1.722	2.304	2.900
locBra_aggr	0.121	0.192	0.376	0.561	0.773	1.157	1.533	1.974
cplex_polish	0.181	0.298	0.596	0.876	1.251	1.895	2.498	3.252

Proximity Search
(three variants)

Local Branching
(aggressive variant)

Comparison metric: geometric mean of **primal integral** (the lower, the better)

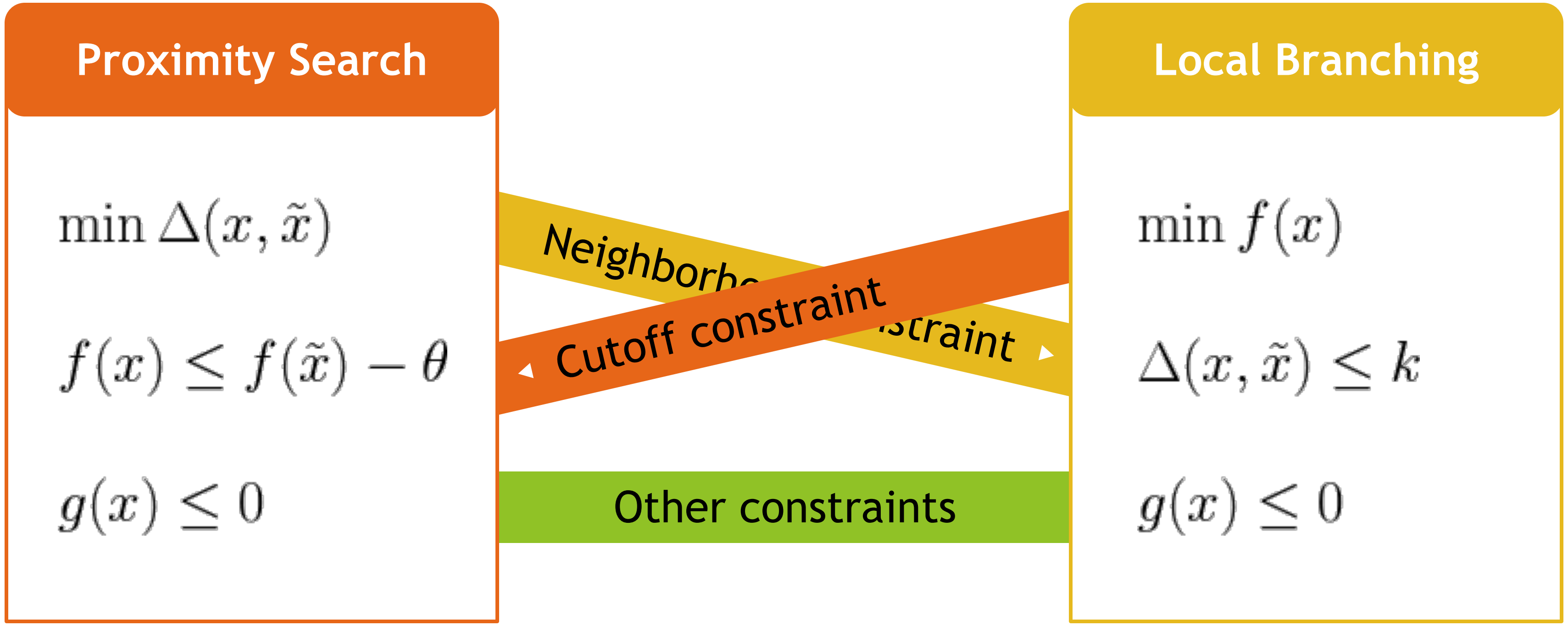
Comparison with **Local Branching**

Proximity Search		Local Branching
$\min \Delta(x, \tilde{x})$	Objective Function	$\min f(x)$
$f(x) \leq f(\tilde{x}) - \theta$		$\Delta(x, \tilde{x}) \leq k$
$g(x) \leq 0$	Other constraints	$g(x) \leq 0$

Comparison with **Local Branching**

Proximity Search		Local Branching
$\min \Delta(x, \tilde{x})$	Objective Function	$\min f(x)$
$f(x) \leq f(\tilde{x}) - \theta$	Cutoff constraint / Neighborhood constraint	$\Delta(x, \tilde{x}) \leq k$
$g(x) \leq 0$	Other constraints	$g(x) \leq 0$

Comparison with **Local Branching**



Proximity Search

$$\min \Delta(x, \tilde{x})$$

$$f(x) \leq f(\tilde{x}) - \theta$$

$$g(x) \leq 0$$

Local Branching

$$\min f(x)$$

$$\Delta(x, \tilde{x}) \leq k$$

$$g(x) \leq 0$$

The cutoff constraint **does not exclude** any **improving solutions** (for small values of θ).

For problems where a **large improvement** is possible **for a small neighborhood**, Local Branching is faster.

On the **primal nature** of Proximity Search

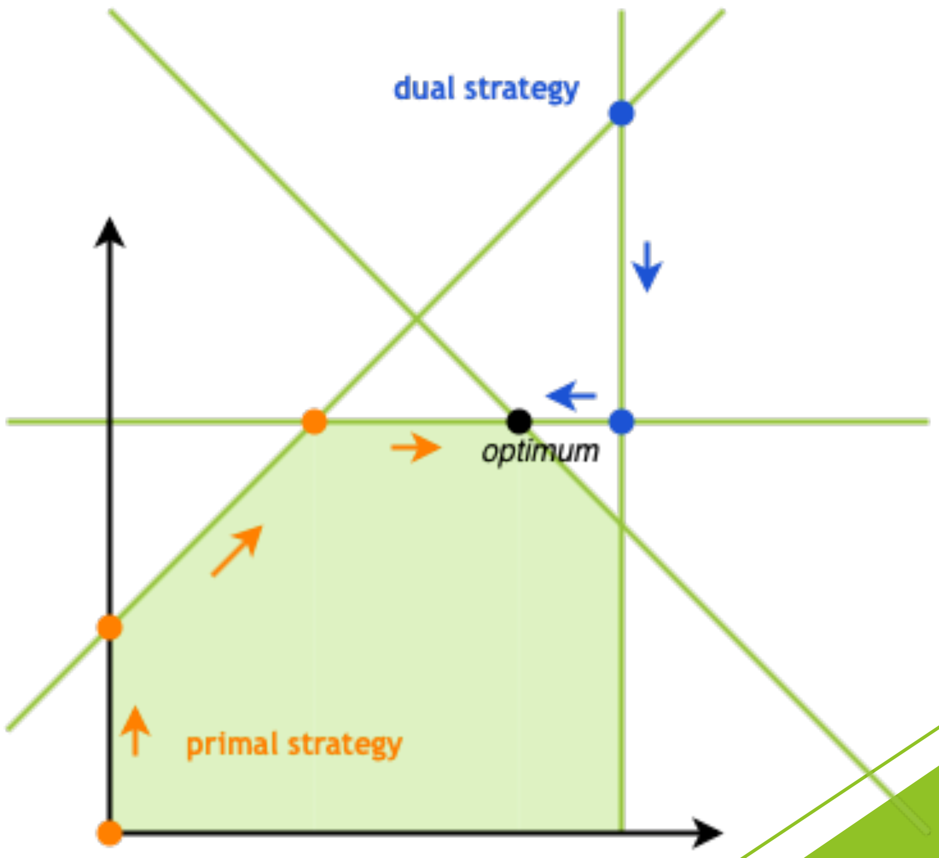
We can say that Proximity Search has a **primal nature**:

Primal Methods

Produce a sequence of improved (**feasible**) solutions that eventually leads to an optimal one.

Dual Methods

Eventually reach the optimal feasible solution through a sequence of more-than-optimal (**infeasible**) solutions.



On the primal nature of Proximity Search

We can say that Proximity Search has a **primal nature:**

Primal Methods	Dual Methods

On the **primal nature** of Proximity Search

We can say that Proximity Search has a **primal nature**:

Primal Methods

Much more satisfactory in terms of "behavior as a **heuristic**".

Dual Methods

A dual method stopped before the optimum is found, gives **no guarantee** to produce a feasible solution.

On the **primal nature** of Proximity Search

We can say that Proximity Search has a **primal nature**:

Primal Methods

Much more satisfactory in terms of "behavior as a **heuristic**".

Proximity Search can be trapped in a long series of **small improvements.**

Dual Methods

A dual method stopped before the optimum is found, gives **no guarantee** to produce a feasible solution.

A more aggressive dual policy can produce less frequent but **much larger improvements**.

Bibliography

- Boland, Eberhard, Engineer, Tsoukalas: *A new approach to the feasibility pump in mixed integer programming*, 2012.
- Danna, Rothberg, Le Pape: *Exploring relaxation induced neighborhoods to improve MIP solutions*, 2005.
- Fischetti, Monaci: *Proximity search for 0-1 mixed-integer convex programming*, 2014.
- Fischetti, Lodi: *Local Branching*, 2003.
- Fischetti, Glover, Lodi: *The Feasability Pump*, 2005.
- Glover: *Parametric branch and bound*, 1978.
- Glover: *Parametric tabu search for mixed integer programs*, 2006.
- Hestenes: *Multiplier and gradient methods*, 1969.
- Rothberg: *An evolutionary algorithm for polishing mixed integer programming solutions*, 2007.

Presentation adapted from:

Fischetti, M., Monaci, M.,

Proximity search for 0-1 mixed-integer convex programming,

(2014)