METHODS

STABLE Kick-off meeting, Fortaleza, April 2013

Content

Content

Content

Content

Content

Content

Content

Content

Outline

What we said:

	Maximum Stable Set	Weighted Stable Set	Graph Coloring	Stable Set Subgraph-weighted	k-partite Subgraph	Weighted Coloring
Branch-and-Cut	X	X	X	X	X	X
Branch-and-Price			X		X	
Lagrangean Price					X	
Constraint Programming	X	X	X	X	X	X
Metaheuristic		X		X		X
Russian Dolls	X		X			
Resolution Search	X		X			

General Overview

- No need to follow exactly what is in the proposal !

General Overview

- No need to follow exactly what is in the proposal !
- We can think in other methods as well !

General Overview

- No need to follow exactly what is in the proposal !
- We can think in other methods as well !
- We can also forget some of the couples (method/problem) !

Outline

Branch-and-Cut

- Well known technique used when the Linear Relaxation of an Integer Optimization problem is not good enough.

Branch-and-Cut

- Well known technique used when the Linear Relaxation of an Integer Optimization problem is not good enough.
- In such a case, one might whish to make it tighter by adding valid inequalities (and, whenever possible, even facets).

Branch-and-Cut

- Well known technique used when the Linear Relaxation of an Integer Optimization problem is not good enough.
- In such a case, one might whish to make it tighter by adding valid inequalities (and, whenever possible, even facets).
- Theoretical difficulty (or interest !) in finding such valid inequalities or facets !

Branch-and-Cut

- Well known technique used when the Linear Relaxation of an Integer Optimization problem is not good enough.
- In such a case, one might whish to make it tighter by adding valid inequalities (and, whenever possible, even facets).
- Theoretical difficulty (or interest !) in finding such valid inequalities or facets !
- Practical difficulty in iddentifying which cuts are not satisfied by the current solution (the Separation Problem).

Branch-and-Cut

- Well known technique used when the Linear Relaxation of an Integer Optimization problem is not good enough.
- In such a case, one might whish to make it tighter by adding valid inequalities (and, whenever possible, even facets).
- Theoretical difficulty (or interest !) in finding such valid inequalities or facets !
- Practical difficulty in iddentifying which cuts are not satisfied by the current solution (the Separation Problem).
- Many interesting theoretical and practical results !

Branch-and-Cut

- Argentinian and Brazilian teams have a great expertise on this technique !

Branch-and-Cut

- Argentinian and Brazilian teams have a great expertise on this technique !
- They found out some facets on the graph coloring problem !

Branch-and-Cut

- Argentinian and Brazilian teams have a great expertise on this technique !
- They found out some facets on the graph coloring problem !
- It is supposed to be applied on each on our problems ...

Outline

Branch-and-Price

- A Branch-and-Bound where each node is evaluated by a Column Generation (or Dantzig-Wolfe decomposition) approach !

Branch-and-Price

- A Branch-and-Bound where each node is evaluated by a Column Generation (or Dantzig-Wolfe decomposition) approach !
- It is supposed to be applied to the Graph Coloring Problem and to the subgraph k-partite of maximal weight.

Branch-and-Price

- A Branch-and-Bound where each node is evaluated by a Column Generation (or Dantzig-Wolfe decomposition) approach !
- It is supposed to be applied to the Graph Coloring Problem and to the subgraph k-partite of maximal weight.
- Brazilian team has a great experience in using Column Generation for Graph Coloring !

Branch-and-Price

- A Branch-and-Bound where each node is evaluated by a Column Generation (or Dantzig-Wolfe decomposition) approach !
- It is supposed to be applied to the Graph Coloring Problem and to the subgraph k-partite of maximal weight.
- Brazilian team has a great experience in using Column Generation for Graph Coloring !
- The subproblems to be solved are then STABLE sets related problems !

Outline

Lagrangean Relaxation:

$$
\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & \\
& Ax \leq b, \\
& Bx = d, \qquad (*) \\
& x \in X(\subset \mathrm{R}^n)
\end{aligned}
$$

Lagrangean Relaxation:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \\ & Ax \leq b, \\ & Bx = d, \qquad (*) \\ & x \in X (\subset \mathrm{R}^n) \end{aligned}$$

It is assumed that the problem is difficult to solve because of constraints (*).

Lagrangean Relaxation:

$$\min \quad f(x)$$

s.t.

$$Ax \leq b,$$

$$Bx = d, \qquad (*) \longmapsto \lambda$$

$$x \in X (\subset \mathrm{R}^n)$$

Lagrangean Relaxation:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & \\ & Ax \leq b, \\ & Bx = d, \qquad (*) \longmapsto \lambda \\ & x \in X(\subset \mathrm{R}^n) \end{array}$$

$$\begin{array}{ll} l(\lambda) = \min & f(x) + < \lambda, Bx - d > \\ \text{s.t.} & \\ & Ax \leq b, \\ & x \in X(\subset \mathrm{R}^n) \end{array}$$

Lagrangean Relaxation:

- l is a concave (but non differentiable) function

Lagrangean Relaxation:

- l is a concave (but non differentiable) function
- l is piecewise linear if f is linear

Lagrangean Relaxation:

- l is a concave (but non differentiable) function
- l is piecewise linear if f is linear
- For each value of λ, $l(\lambda)$ is a lower bound to the optimal value of the initial problem

Lagrangean Relaxation:

- l is a concave (but non differentiable) function
- l is piecewise linear if f is linear
- For each value of λ, $l(\lambda)$ is a lower bound to the optimal value of the initial problem
- Dual Problem : $\max\{l(\lambda)/\lambda \in \mathrm{R}^m\}$

Lagrangean Relaxation:

- l is a concave (but non differentiable) function
- l is piecewise linear if f is linear
- For each value of λ, $l(\lambda)$ is a lower bound to the optimal value of the initial problem
- Dual Problem : $\max\{l(\lambda)/\lambda \in \mathrm{R}^m\}$
- Lagrangean Relaxation is used since the 70's !

Lagrangean Relaxation:

- l is a concave (but non differentiable) function
- l is piecewise linear if f is linear
- For each value of λ, $l(\lambda)$ is a lower bound to the optimal value of the initial problem
- Dual Problem : $\max\{l(\lambda)/\lambda \in \mathrm{R}^m\}$
- Lagrangean Relaxation is used since the 70's !
- We planned to use Lagrangean Relaxation for the Subgraph k partite problem.

Outline

Constraint Programming:

- Issued from the Artificial Intelligence community.

Constraint Programming:

- Issued from the Artificial Intelligence community.
- To each decision variable is associated a "domain", the set of possible values for this variable.

Constraint Programming:

- Issued from the Artificial Intelligence community.
- To each decision variable is associated a "domain", the set of possible values for this variable.
- To each constraint of an Optimization Problem is associated a so-called filtering algorithm whose goal is to eliminate the values of the domains which are not possible anymore.

Constraint Programming:

- Issued from the Artificial Intelligence community.
- To each decision variable is associated a "domain", the set of possible values for this variable.
- To each constraint of an Optimization Problem is associated a so-called filtering algorithm whose goal is to eliminate the values of the domains which are not possible anymore.
- Whenever a domain has been reduced, all the filtering algorithms of the constraints linked to the corresponding variable are called (propagation phase).

Constraint Programming:

- Constraint Programming tackles optimization problems by considering each constraint independently of the others.

Constraint Programming:

- Constraint Programming tackles optimization problems by considering each constraint independently of the others.
- For this purpose, "Global Constraints" have been introduced.

Constraint Programming:

- Constraint Programming tackles optimization problems by considering each constraint independently of the others.
- For this purpose, "Global Constraints" have been introduced.
- Constraint Programming models are generally more compact and more natural.

Constraint Programming:

- Constraint Programming tackles optimization problems by considering each constraint independently of the others.
- For this purpose, "Global Constraints" have been introduced.
- Constraint Programming models are generally more compact and more natural.
- Most of the efficient methods for scheduling problems are of the CP type ...

Constraint Programming:

- Constraint Programming tackles optimization problems by considering each constraint independently of the others.
- For this purpose, "Global Constraints" have been introduced.
- Constraint Programming models are generally more compact and more natural.
- Most of the efficient methods for scheduling problems are of the CP type ...
- We are supposed to apply that to all the problems.

Outline

Metaheuristic and Heuristic methods:

- Methods able to find quickly a "good" solution ...

Metaheuristic and Heuristic methods:

- Methods able to find quickly a "good" solution ...
- but unable to prove the optimality even when the optimum has been found.

Metaheuristic and Heuristic methods:

- Methods able to find quickly a "good" solution ...
- but unable to prove the optimality even when the optimum has been found.
- In our project, Heuristic methods are of special importance: they will be used to make Column Generation and Cut Generation approaches faster (by solving the corresponding subproblems).

Metaheuristic and Heuristic methods:

- Methods able to find quickly a "good" solution ...
- but unable to prove the optimality even when the optimum has been found.
- In our project, Heuristic methods are of special importance: they will be used to make Column Generation and Cut Generation approaches faster (by solving the corresponding subproblems).
- Interesting by themselves !

Metaheuristic and Heuristic methods:

- Methods able to find quickly a "good" solution ...
- but unable to prove the optimality even when the optimum has been found.
- In our project, Heuristic methods are of special importance: they will be used to make Column Generation and Cut Generation approaches faster (by solving the corresponding subproblems).
- Interesting by themselves !
- We are supposed to design heuristic methods for the Weighted Stable Set, Subgraph-Weighted Stable set and the Weighted Coloring problem.

Metaheuristic and Heuristic methods:

- Methods able to find quickly a "good" solution ...
- but unable to prove the optimality even when the optimum has been found.
- In our project, Heuristic methods are of special importance: they will be used to make Column Generation and Cut Generation approaches faster (by solving the corresponding subproblems).
- Interesting by themselves !
- We are supposed to design heuristic methods for the Weighted Stable Set, Subgraph-Weighted Stable set and the Weighted Coloring problem.
- Chilean and French teams have a good experience on these approaches.

Outline

Russian Dolls approach:

- Introduced by Verfaillie, Schleix and Lemaitre in 1996.

Russian Dolls approach:

- Introduced by Verfaillie, Schleix and Lemaitre in 1996.
- consists in solving a sequence of subproblems such that each subproblem is identical to the previous one except that it is bigger and contains it, just like Russian dolls ...

Russian Dolls approach:

- Introduced by Verfaillie, Schleix and Lemaitre in 1996.
- consists in solving a sequence of subproblems such that each subproblem is identical to the previous one except that it is bigger and contains it, just like Russian dolls ...
- Interesting only if we can take advantage of the previous resolution for accelarating the current resolution !

Russian Dolls approach:

- Introduced by Verfaillie, Schleix and Lemaitre in 1996.
- consists in solving a sequence of subproblems such that each subproblem is identical to the previous one except that it is bigger and contains it, just like Russian dolls ...
- Interesting only if we can take advantage of the previous resolution for accelarating the current resolution !
- Most of the time, the objective function of the previous (smaller) subproblems are used as cuts

Russian Dolls approach: example for the MAximum Stable Set Problem

- Let G be a graph of n vertices in which we look for the maximum stable set.

Russian Dolls approach: example for the MAximum Stable Set Problem

- Let G be a graph of n vertices in which we look for the maximum stable set.
- Start with a subgraph of small size (50) whose vertices are 1, ..., 50 and solve it by any algorithm.

Russian Dolls approach: example for the MAximum Stable Set Problem

- Let G be a graph of n vertices in which we look for the maximum stable set.
- Start with a subgraph of small size (50) whose vertices are 1, ..., 50 and solve it by any algorithm.
- Consider now the vertice whose index is 51. We want to solve the problem with this additional vertice.

Russian Dolls approach: example for the MAximum Stable Set Problem

- Let G be a graph of n vertices in which we look for the maximum stable set.
- Start with a subgraph of small size (50) whose vertices are 1, ..., 50 and solve it by any algorithm.
- Consider now the vertice whose index is 51. We want to solve the problem with this additional vertice.
- Either the maximum stable contains 51 or it does not !

Russian Dolls approach: example for the MAximum Stable Set Problem

- Let G be a graph of n vertices in which we look for the maximum stable set.
- Start with a subgraph of small size (50) whose vertices are 1, ..., 50 and solve it by any algorithm.
- Consider now the vertice whose index is 51. We want to solve the problem with this additional vertice.
- Either the maximum stable contains 51 or it does not !
- If it does not, then this is the stable found at the previous iteration !

Russian Dolls approach: example for the MAximum Stable Set Problem

- Let G be a graph of n vertices in which we look for the maximum stable set.
- Start with a subgraph of small size (50) whose vertices are 1, ..., 50 and solve it by any algorithm.
- Consider now the vertice whose index is 51. We want to solve the problem with this additional vertice.
- Either the maximum stable contains 51 or it does not !
- If it does not, then this is the stable found at the previous iteration !
- If it contains 51, then we can eliminate all the nodes which are in relation with 51 (then we work on a small size problem) with the additional constraint that $\sum_{i=1}^{50} v_i \leq S_{50}$

Outline

Resolution Search

- Introduced by Chvatal in 1997

Resolution Search

- Introduced by Chvatal in 1997
- as an alternative to the Branch-and-Bound !

Resolution Search

- Introduced by Chvatal in 1997
- as an alternative to the Branch-and-Bound !
- First designed only for 0/1 problems, within the traditional branching scheme ($x_i = 0$ or $x_i = 1$)

Resolution Search

- Introduced by Chvatal in 1997
- as an alternative to the Branch-and-Bound !
- First designed only for 0/1 problems, within the traditional branching scheme ($x_i = 0$ or $x_i = 1$)
- Now generalized to any kind of problem and any kind of branching scheme.

Resolution Search

- The basic idea is to iteratively cover the feasible set of the problem.

Resolution Search

- The basic idea is to iteratively cover the feasible set of the problem.
- Each covered subset is recognized by a partial solution (such partial solutions are called "nogoods").

Resolution Search

- The basic idea is to iteratively cover the feasible set of the problem.
- Each covered subset is recognized by a partial solution (such partial solutions are called "nogoods").
- Hence, the method maintains a list (family) of nogoods.

Resolution Search

- The basic idea is to iteratively cover the feasible set of the problem.
- Each covered subset is recognized by a partial solution (such partial solutions are called "nogoods").
- Hence, the method maintains a list (family) of nogoods.
- At each iteration, a partial (or complete) solution is generated outside the covered subsets. From this partial solution is deduced a new nogood which is added to the family.

Resolution Search

- The basic idea is to iteratively cover the feasible set of the problem.
- Each covered subset is recognized by a partial solution (such partial solutions are called "nogoods").
- Hence, the method maintains a list (family) of nogoods.
- At each iteration, a partial (or complete) solution is generated outside the covered subsets. From this partial solution is deduced a new nogood which is added to the family.
- Two difficulties: generating the new partial solution and deducing the new nogood.

Resolution Search

- Chvatal showed that if the family keeps the "path-like" property, then generating the new partial solution is easy !

Resolution Search

- Chvatal showed that if the family keeps the "path-like" property, then generating the new partial solution is easy !
- For deducing a new nogood from this partial solution, a completion phase is achieved (just like in a greedy heuristic or in a depth-first branch-and-bound procedure)

Resolution Search

- Chvatal showed that if the family keeps the "path-like" property, then generating the new partial solution is easy !
- For deducing a new nogood from this partial solution, a completion phase is achieved (just like in a greedy heuristic or in a depth-first branch-and-bound procedure)
- From this now completed solution, a partial solution such that none of its completion can be better than the best known solution is deduced.

Resolution Search

- Chvatal showed that if the family keeps the "path-like" property, then generating the new partial solution is easy !
- For deducing a new nogood from this partial solution, a completion phase is achieved (just like in a greedy heuristic or in a depth-first branch-and-bound procedure)
- From this now completed solution, a partial solution such that none of its completion can be better than the best known solution is deduced.
- It will be the next nogood.

Resolution Search

- Only alternative to the Branch-and-Bound ...

Resolution Search

- Only alternative to the Branch-and-Bound ...
- Within the RS framework, no need to start the search by the root node : we can start from a feasible solution or a set of feasible solutions ...

Resolution Search

- Only alternative to the Branch-and-Bound ...
- Within the RS framework, no need to start the search by the root node : we can start from a feasible solution or a set of feasible solutions ...
- Many nodes are not evaluated (while they would be in a BB scheme).

Resolution Search

- Only alternative to the Branch-and-Bound ...
- Within the RS framework, no need to start the search by the root node : we can start from a feasible solution or a set of feasible solutions ...
- Many nodes are not evaluated (while they would be in a BB scheme).
- ... but up to now, no problem has been solved by RS !