Integrating Optimization and Constraint Satisfaction

John Hooker
Carnegie Mellon University
IJCAI-99, Stockholm
Some work joint with...

Ignacio Grossmann, Carnegie Mellon University
Hak-Jin Kim, Carnegie Mellon University
María Auxilio Osorio, Universidad Autónoma de Puebla
Greger Ottoisson, Uppsala Universitet
Erlendur Porsteinsson, Carnegie Mellon University
Integration of optimization and constraint satisfaction

- Optimization and constraint satisfaction have complementary strengths.
- There is much interest in combining them, but their different origins have impeded unification.
  - Optimization -- operations research, mathematics
  - Constraint satisfaction -- artificial intelligence, computer science
- This barrier is now being overcome, but there is no generally accepted scheme for unification.
Complementary strengths

• Problem types.
  • Optimization excels at loosely-constrained problems in which finding the best solution is the primary task.
  • Constraint satisfaction is more effective on tightly-constrained problems in which finding a feasible solution is paramount.
Complementary strengths

• Exploiting structure

  • Optimization relies on deep analysis of the mathematical structure of specific classes of problems, particularly polyhedral analysis, which yields strong cutting planes.

  • Constraint satisfaction identifies subsets of problem constraints that have special structure (e.g., all-different, cumulative) and apply tailor-made domain-reduction algorithms.
Complementary strengths

• Relaxation and inference.

• Optimization creates strong relaxations with cutting planes, Lagrangean relaxation, etc. These provide bounds on the optimal value.

• Constraint satisfaction exploits the power of inference, especially in domain reduction algorithms. This reduces the search space.
Complementary strengths

• Modeling style

  • Optimization uses declarative models that can be solved with a variety of algorithms. But the language is highly restricted (e.g., inequality constraints).

  • Constraint satisfaction models are formulated in a procedural or quasi-procedural manner that gives the user more opportunity to direct the solution algorithm. But the model is tied to the solution method.
Procedural vs. declarative

• The issue of procedural vs. declarative modeling is orthogonal to the issue of how solution methods can be combined.

  • Constraint (logic) programming generally implements constraint satisfaction techniques in a quasi-procedural manner. But these techniques can equally well be applied to a declarative model, and the flexible modeling language of C(L)P can be used declaratively (OPL moves in this direction).

  • Optimization methods are generally applied to declarative models, but they can be implemented in a high-level programming language as well.
Constraint programming vs. constraint satisfaction

- Constraint programming is sometimes identified with different techniques than constraint satisfaction.
  - For example, domain reduction (arc or hyperarc consistency) rather than minimum-width search orders, adaptive consistency, etc.
- Here, constraint programming is viewed as a modeling approach. Constraint satisfaction techniques include all solution techniques implemented by CP or CLP.
Scheme for unifying optimization and constraint satisfaction methods

• Both optimization and constraint satisfaction rely on two fundamental dualities.

• Search/inference
  • Search = branching, local search
  • Inference = constraint propagation, cutting planes

• Strengthening/relaxation
  • Strengthening = fix variables or restrict domains
  • Relaxation = weaken constraints, bound objective function
Scheme for unifying optimization and constraint satisfaction methods

• Rather than use optimization or constraint satisfaction methods exclusively, focus on how these two dualities can be exploited in a given problem.

• The resulting algorithm is likely to contain elements from both optimization and constraint satisfaction, and perhaps new methods that belong to neither.

• In particular, optimization can benefit from inference methods of constraint satisfaction.

• Constraint satisfaction can benefit from relaxation methods of optimization.
Outline

• A motivating example
  Constraint satisfaction approach
  Integer programming approach
  Combined approach

• The search/inference duality
  Complementary solution methods
  A formal duality

• The strengthening/relaxation duality
  Complementary solution methods
  Relaxations for global constraints
  Formal relaxation duality

• Relaxation duality
  An integer programming example
  Continuous relaxation
  Discrete relaxation
  dependency graph, nonserial
dynamic programming
  Relaxation duality
  Lagrangean & surrogate duals
  Discrete relax + Lagrangean dual
  Discrete relaxation dual
  Discrete + Lagrangean duals
  Summary of relaxations

• Research agenda
A motivating example

\[ \begin{align*}
\text{min} & \quad 4x_1 + 3x_2 + 5x_3 \\
\text{subject to} & \quad 4x_1 + 2x_2 + 4x_3 \geq 17 \\
& \quad \text{all-different}\{x_1, x_2, x_3\} \\
& \quad x_j \in \{1, \ldots, 5\}
\end{align*} \]

Formulate and solve 3 ways:

* a constraint satisfaction problem
* an integer programming problem
* a combined approach
Solve as a constraint satisfaction problem

\[ 4x_1 + 3x_2 + 5x_3 \leq z \]
\[ 4x_1 + 2x_2 + 4x_3 \geq 17 \]
all - different\{\(x_1, x_2, x_3\)\}
\[ x_j \in \{1, \ldots, 5\} \]

Start with \( z = \infty \).
Will decrease as feasible solutions are found.
Domain reduction used

• Bounds propagation on
  
  \[ 4x_1 + 3x_2 + 5x_3 \leq z \]
  
  \[ 4x_1 + 2x_2 + 4x_3 \geq 17 \]

• Maintain hyperarc consistency on
  
  \[ \text{all} - \text{different}\{x_1, x_2, x_3\} \]

• Cycle through domain reductions until a fixed point is obtained
1. $z = \infty$

Domain of $x_1$

2. $z = \infty$

Domain of $x_1$

3. $z = \infty$

Domain of $x_2$

4. $z = \infty$

Domain of $x_3$

5. $z = 25$

Domain of $x_1$

6. $z = 25$

Domain of $x_2$

7. $z = 23$

Domain of $x_1$

8. $z = 23$

Domain of $x_2$

9. $z = 22$

Domain of $x_2$
Solve as an integer programming problem

\[
\begin{align*}
\text{min} & \quad 4x_1 + 3x_2 + 5x_3 \\
\text{subject to} & \quad 4x_1 + 2x_2 + 4x_3 \geq 17 \\
& \quad x_j \leq (x_k - 1) + 5(1 - y_{jk}), \quad \text{all } j, k \text{ with } j < k \\
& \quad x_k \leq (x_j - 1) + 5y_{jk}, \quad \text{all } j, k \text{ with } j < k \\
& \quad 1 \leq x_j \leq 5, \quad x_j \text{ integer}, \quad j = 1, 2, 3 \\
& \quad y_{jk} \in \{0, 1\}, \quad \text{all } j, k \text{ with } j < k \\
\end{align*}
\]

\[x_j < x_k \text{ if } y_{jk} = 1\]

Big-M constraints
Linear relaxation

Use a linear programming algorithm to solve a continuous relaxation of the problem at each node of the search tree to obtain a lower bound on the optimal value of the problem at that node.

\[
\begin{align*}
\text{min} & \quad 4x_1 + 3x_2 + 5x_3 \\
\text{subject to} & \quad 4x_1 + 2x_2 + 4x_3 \geq 17 \\
& \quad x_j \leq (x_k - 1) + 5(1 - y_{jk}), \quad \text{all } j, k \text{ with } j < k \\
& \quad x_k \leq (x_j - 1) + 5y_{jk}, \quad \text{all } j, k \text{ with } j < k \\
& \quad 1 \leq x_j \leq 5, \quad j = 1, 2, 3 \\
& \quad 0 \leq y_{jk} \leq 1, \quad \text{all } j, k \text{ with } j < k \\
\end{align*}
\]

\} Relax integrality
Alternate model

The following model has a better relaxation and would be used for this problem in practice. The big-$M$ construction is used here to illustrate a popular and general technique.

\[
\begin{align*}
\text{min} & \quad 4x_1 + 3x_2 + 5x_3 \\
\text{subject to} & \quad 4x_1 + 2x_2 + 4x_3 \geq 17 \\
& \quad x_i = \sum_{j=1}^{5} jy_{ij}, \quad i = 1, 2, 3 \\
& \quad \sum_{j=1}^{5} y_{ij} = 1, \quad i = 1, 2, 3 \\
& \quad y_{jk} \in \{0, 1\}, \quad \text{all } j, k
\end{align*}
\]
Cutting planes

Infer the cutting planes
\[ x_1 + x_2 + x_3 \geq 5 \]
\[ 2x_1 + x_2 + 2x_3 \geq 9 \]
From the inequality
\[ 4x_1 + 2x_2 + 4x_3 \geq 17 \]

The cutting plane is implied by the inequality but strengthens the continuous relaxation

(One could also use the all-different constraint to obtain the stronger cutting plane \[ x_1 + x_2 + x_3 \geq 6 \] )
Branch and bound

The *incumbent solution* is the best feasible solution found so far.

At each node of the branching tree:

- If \( \text{Optimal value of relaxation} \geq \text{Value of incumbent solution} \),

  There is no need to branch further.

- No feasible solution in that subtree can be better than the incumbent solution.
Combined approach

• Use continuous relaxation with cutting planes but without big-M constraints
  • Because relaxation is distinguished from model, both are succinct.

• Use bounds propagation on cutting planes as well as original inequality constraints.

• Maintain hyperarc consistency for all-different.

• Branch on nonintegral variable when possible; otherwise branch by splitting domain.
1. $z = \infty$

2. $z = \infty$

3. $z = \infty$

4. $z = \infty$

5. $z = 22$

6. $z = 22$

7. $z = 22$

$x_1 \leq 2$  

$x_2 \geq 2$  

$x_3 \geq 2$  

$x_1 \geq 3$  

$x_3 \leq 1$

$x = (2, 3, 1)$  

Optimal

$x = (2.5, 2, 1)$  

value = 21

$x = (3, 1, 1)$  

value = 20

$x = (2, 2, 1.5)$  

value = 21.5

infeasible

infeasible

infeasible

infeasible
# Summary of dualities

<table>
<thead>
<tr>
<th>Constraint satisfaction</th>
<th>Search</th>
<th>Inference</th>
<th>Strengthening</th>
<th>Relaxation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Domain splitting</td>
<td>Bounds prop. on ineq. cons.</td>
<td>Domain splitting</td>
<td>Reduced domains</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Domain red. on all-diff</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integer programming</td>
<td>Branching on fractions</td>
<td>Cutting planes</td>
<td>Branching on fractions</td>
<td>Continuous relaxation of IP model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined</td>
<td>Both of the above</td>
<td>Bounds prop. on ineq. cons. &amp; cutting planes</td>
<td>Both of the above</td>
<td>Reduced domains Continuous relaxation of part of IP model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Domain red. on all-diff</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

• A motivating example
  Constraint satisfaction approach
  Integer programming approach
  Combined approach

• The search/inference duality
  Complementary solution methods
  A formal duality

• The strengthening/relaxation duality
  Complementary solution methods
  Relaxations for global constraints
  Formal relaxation duality

Relaxation duality
  An integer programming example
  Continuous relaxation
  Discrete relaxation
  dependency graph, nonserial
dynamic programming
  Relaxation duality
  Lagrangean & surrogate duals
  Discrete relax + Lagrangean dual
  Discrete relaxation dual
  Discrete + Lagrangean duals
  Summary of relaxations

Research agenda
The search/inference duality

• Two interpretations:
  • Complementary solution methods that can work together.
  • A formal mathematical duality that can lead to new methods.
The search/inference duality

• Complementary solution methods:
  • Search alone may find a good solution early, but it must examine many other solutions to determine that it is good.
  • Inference can rule out families of inferior solutions, but this is not the same as finding a good solution.
  • Working together, search & inference can find and verify good solutions more quickly.
The search/inference duality

- A formal duality:
  - Search and inference are related by a formal optimization duality.
  - Linear programming duality is a special case.
  - This provides a general method for sensitivity analysis.
  - It also provides a general form of Benders decomposition, which is closely related to the use of nogoods.
Outline

• A motivating example
  Constraint satisfaction approach
  Integer programming approach
  Combined approach

• The search/inference duality
  Complementary solution methods
  A formal duality

• The strengthening/relaxation duality
  Complementary solution methods
  Relaxations for global constraints
  Formal relaxation duality

• Relaxation duality
  An integer programming example
  Continuous relaxation
  Discrete relaxation
  *dependency graph, nonserial dynamic programming*
  Relaxation duality
  Lagrangean & surrogate duals
  Discrete relax + Lagrangean dual
  Discrete relaxation dual
  Discrete + Lagrangean duals
  Summary of relaxations

• Research agenda
The strengthening/relaxation duality

• Three interpretations:
  • Complementary solution methods that can work together.
  • Creation of relaxations as well as domain reduction algorithms to exploit structure of subsets of constraints (e.g., element constraints).
  • A formal mathematical duality that can lead to new relaxations, particularly for constraint satisfaction models.
The strengthening/relaxation duality

- Complementary solution methods
  - Branch-and-bound solves relaxations of strengthenings. Branching creates strengthenings, and one solves a relaxation of each to obtain bounds.
  - There are other ways strengthening and relaxation can relate. One can solve strengthenings of a relaxation. Branching creates strengthenings of an initial relaxation.
Relaxations of strengthenings vs. strengthenings of a relaxation

They are the same in integer programming because the strengthening and relaxation functions commute:

Relaxations of strengthenings vs. strengthenings of a relaxation
Relaxations of strengthenings vs. strengthenings of a relaxation

They are not the same in general; for example, when solving a propositional satisfiability problem with the help of Horn relaxation. Diagram does not commute.
Example...
The strengthening/relaxation duality

- Another interpretation: create relaxations as well as domain reduction algorithms for specially structured “global” constraints.
  - This will be illustrated with the element constraint, which can be used to implement variable subscripts (indices).
The constraint $x_y \leq \beta$ where $x_j \in D_{x_j}$, $y \in D_y$ are discrete variables, can be implemented:

$$z \leq \beta$$

$\text{element}(y,(x_1,\ldots,x_n),z)$

Here, $\text{element}$ is processed with a discrete domain reduction algorithm that maintains hyperarc consistency.

\[
D_z \leftarrow D_z \cap \bigcup_{j \in D_y} D_{x_j} \\
D_y \leftarrow D_y \cap \{j \mid D_x \cap D_{x_j} \neq \emptyset\} \\
D_{x_j} \leftarrow \begin{cases} D_z & \text{if } D_y = \{j\} \\ D_{x_j} & \text{otherwise} \end{cases}
\]
Example... \text{element}(y,(x_1,x_2,x_3,x_4),z)

The initial domains are:

\[
\begin{align*}
D_z &= \{20,30,60,80,90\} \\
D_y &= \{1,3,4\} \\
D_{x_1} &= \{10,50\} \\
D_{x_2} &= \{10,20\} \\
D_{x_3} &= \{40,50,80,90\} \\
D_{x_4} &= \{40,50,70\}
\end{align*}
\]

The reduced domains are:

\[
\begin{align*}
D_z &= \{80,90\} \\
D_y &= \{3\} \\
D_{x_1} &= \{10,50\} \\
D_{x_2} &= \{10,20\} \\
D_{x_3} &= \{80,90\} \\
D_{x_4} &= \{40,50,70\}
\end{align*}
\]
Continuous variable with variable index

The constraint $x_y \leq \beta$ where each $x_j \ (0 \leq x_j \leq m_j)$ is a continuous variable, can be implemented:

$$z \leq \beta$$

$\text{element}(y, (x_1, \ldots, x_n), z)$

Here, $\text{element}$ generates a continuous relaxation that is added to the linear programming subproblem:

$$\sum_{i \in D_y} \frac{x_i}{m_i} - \left( \sum_{i \in D_y} \frac{1}{m_i} \right) z \geq -|D_y| + 1$$

$$- \sum_{i \in D_y} \frac{x_i}{m_i} + \left( \sum_{i \in D_y} \frac{1}{m_i} \right) z \geq -|D_y| + 1$$
Example... element(y, (x₁, x₂), z)
0 ≤ x₁ ≤ 4
0 ≤ x₂ ≤ 5

The relaxation is: 5x₁ + 4x₂ − 9z ≤ 20
5x₁ + 4x₂ − 9z ≥ −20
0 ≤ x₁ ≤ 4
0 ≤ x₂ ≤ 4
The strengthening/relaxation duality

• Can be interpreted as a formal relaxation duality.
  • Linear programming duality, Lagrangean duality, surrogate duality are special cases.
  • These classical dualities apply only to numeric equality and inequality constraints.
  • General relaxation duality can be used to create new relaxations for other constraints.
  • One approach is to use the concept of induced width of a dependency graph, along with nonserial dynamic programming.
Outline

• **A motivating example**
  - Constraint satisfaction approach
  - Integer programming approach
  - Combined approach

• **The search/inference duality**
  - Complementary solution methods
  - A formal duality

• **The strengthening/relaxation duality**
  - Complementary solution methods
  - Relaxations for global constraints
  - Formal relaxation duality

• **Relaxation duality**
  - An integer programming example
  - Continuous relaxation
  - Discrete relaxation
  - *dependency graph, nonserial dynamic programming*
  - Relaxation duality
  - Lagrangean & surrogate duals
  - Discrete relax + Lagrangean dual
  - Discrete relaxation dual
  - Discrete + Lagrangean duals
  - Summary of relaxations

• **Research agenda**
Integer programming example

\[
\begin{align*}
\text{min} & \quad 35x_1 + 20x_2 + 15x_3 + 40x_4 + 30x_5 \\
\text{subject to} & \quad 2x_1 + 3x_2 + x_3 \geq 4 \\
& \quad x_2 + 4x_3 + 3x_4 \geq 3 \\
& \quad 2x_3 + 3x_4 + x_5 \geq 4 \\
& \quad 3x_1 + 4x_3 + 5x_5 \geq 5 \\
& \quad x_j \in \{0,1\}, \quad \text{all } j
\end{align*}
\]

Optimal value = 105
Continuous relaxation

\[
\begin{align*}
\text{min} & \quad 35x_1 + 20x_2 + 15x_3 + 40x_4 + 30x_5 \\
\text{subject to} & \quad 2x_1 + 3x_2 + x_3 \geq 4 \\
& \quad x_2 + 4x_3 + 3x_4 \geq 3 \\
& \quad 2x_3 + 3x_4 + x_5 \geq 4 \\
& \quad 3x_1 + 4x_3 + 5x_5 \geq 5 \\
& \quad 0 \leq x_j \leq 1, \quad \text{all } j
\end{align*}
\]

Optimal value = \( 43\frac{1}{3} \)
Dependency graph

\begin{align*}
2x_1 + 3x_2 - x_3 & \geq 4 \\
x_2 + 4x_3 + 3x_4 & \geq 3 \\
2x_3 + 3x_4 + x_5 & \geq 4 \\
3x_1 + 4x_3 + 5x_5 & \geq 5
\end{align*}
Induced width $= 3$
Discrete relaxation

To create a discrete relaxation:

• Thin out the dependency graph so that it has a smaller induced width.

• Use projection to remove variable couplings that correspond to deleted arcs.

• Solve resulting problem by nonserial dynamic programming, whose complexity varies exponentially with induced width.

• The idea of nonserial dynamic programming has surfaced in several contexts: Markov trees, solution of Bayesian networks, etc.
Reduce induced width to 2

Delete arc

Project onto $x_2, x_3$

Project onto $x_1, x_3$

$2x_1 + 3x_2 + x_3 \geq 4$

$3x_2 + x_3 \geq 1$

$2x_1 + x_3 \geq 1$

This removes coupling between $x_1, x_2$ in constraint 1
Discrete relaxation

\[
\begin{align*}
\text{min} & \quad 35x_1 + 20x_2 + 15x_3 + 40x_4 + 30x_5 \\
\text{subject to} & \quad 3x_2 + x_3 \geq 2 \\
& \quad 2x_1 + x_3 \geq 1 \\
& \quad x_2 + 4x_3 + 3x_4 \geq 3 \\
& \quad 2x_3 + 3x_4 + x_5 \geq 4 \\
& \quad 3x_1 + 4x_3 + 5x_5 \geq 5 \\
& \quad x_j \in \{0,1\}, \quad \text{all } j
\end{align*}
\]

Optimal value = 105 (same as original problem)
Nonserial dynamic programming

\[ g_1(x_3, x_5) = \min \left\{ 35x_1 + 15x_3 + 30x_5 + M(1 - 2x_1 - x_3)^+ \right\} \]

\[ g_2(x_3, x_4) = \min \left\{ 20x_2 + 40x_4 + M(2 - 3x_2 - x_3)^+ \right\} \]
NSDP, continued

\[ g_3(x_4, x_5) = \min_{x_3} \{ g_1(x_3, x_5) + g_2(x_3, x_4) + M(4 - 2x_3 - 3x_4 - x_5)^+ \} \]

solution = \( \min_{x_4, x_5} g_3(x_4, x_5) = 105 \)
Optimal value = 90

\[ \{0, 1\} \ni x \]

\[ 1 \geq x \zeta + x \zeta \]

\[ 2 \geq x \zeta + x \zeta \]

\[ 1 \geq x + x \zeta \]

\[ 2 \geq x + x \zeta \]

Subject to

\[ \min \]

\[ \xi \zeta + \zeta x + \zeta x \xi + \zeta x \xi \]

Reduce induced width to 1
Continuous & discrete relaxations

\[ 43 \frac{1}{3} < 90 < 105 \]

Value of continuous relaxation

Value of discrete relaxation

Optimal value
Parameterized relaxation

\[
\begin{align*}
\min & \quad f(x) \\
\text{subject to} & \quad x \in S
\end{align*}
\]
\} \text{ Generic optimization problem}

\[
\begin{align*}
\theta(\lambda) &= \min & \quad f(x, \lambda) \\
\text{subject to} & \quad x \in S(\lambda)
\end{align*}
\]
\} \text{ Parameterized relaxation}

\[
\max\{\theta(\lambda)\} \quad \text{Relaxation dual}
\]

General conditions for a relaxation:

\[
\begin{align*}
f(x, \lambda) &\leq f(x), \quad \text{all } x \in S, \lambda \in \Lambda \\
S(\lambda) &\supseteq S, \quad \text{all } \lambda \in \Lambda
\end{align*}
\]
Lagrangean relaxation

$$\min f(x)$$
subject to $$g_i(x) \leq 0, \quad i \in I$$  \quad \{$$\text{Optimization problem}$$\}
$$x \in S$$

$$\theta(\lambda) = \min f(x) + \sum_{i \in I} g_i(x)$$
subject to $$x \in S$$

$$\max_{\lambda \geq 0} \{\theta(\lambda)\}$$  \quad \{$$\text{Lagrangean dual}$$\}
Surrogate dual

\[
\begin{align*}
\min & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad i \in I \\
& \quad x \in S
\end{align*}
\]  \quad \text{Optimization problem}

\[
\theta(\lambda) = \min \quad f(x) \\
\text{subject to} & \quad \sum_{i \in I} \lambda_i g_i(x) \leq 0 \\
& \quad x \in S
\]  \quad \text{Surrogate relaxation}

\[
\max_{\lambda \geq 0} \{ \theta(\lambda) \} \quad \text{Surrogate dual}
\]
Combine discrete relaxation with Lagrangean duality

\[
\begin{align*}
\min & \quad 35x_1 + 20x_2 + 15x_3 + 40x_4 + 30x_5 \\
& \quad + \lambda_1 (4 - 2x_1 - 3x_2 - x_3) \\
& \quad + \lambda_2 (3 - x_2 - 4x_3 - 3x_4) \\
& \quad + \lambda_3 (4 - 2x_3 - 3x_4 - x_5) \\
& \quad + \lambda_4 (5 - 3x_1 - 4x_3 - 5x_5) \\
\text{subject to} & \quad 3x_2 \geq 1 \\
& \quad 3x_4 + x_5 \geq 2 \\
& \quad 2x_3 + x_5 \geq 1 \\
& \quad 4x_3 + 5x_5 \geq 2 \\
& \quad 3x_1 + 5x_5 \geq 1 \\
& \quad x_j \in \{0,1\}, \; \text{all } j
\end{align*}
\]
Solve by subgradient optimization

<table>
<thead>
<tr>
<th>Step</th>
<th>$\lambda$</th>
<th>Value</th>
<th>Subgradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0</td>
<td>90</td>
<td>1 -1 -4 0</td>
</tr>
<tr>
<td>6</td>
<td>6 0 0 0</td>
<td>96</td>
<td>1 -1 -4 0</td>
</tr>
<tr>
<td>3</td>
<td>9 0 0 0</td>
<td>92</td>
<td>-2 -5 -1 0</td>
</tr>
<tr>
<td>2</td>
<td>5 0 0 2</td>
<td>95</td>
<td>1 -1 -4 0</td>
</tr>
<tr>
<td>1.5</td>
<td>6.5 0 0 2</td>
<td>96.5</td>
<td>1 -1 -4 0</td>
</tr>
<tr>
<td>1.2</td>
<td>7.7 0 0 2</td>
<td>96.6</td>
<td>-2 -5 -1 0</td>
</tr>
<tr>
<td>1</td>
<td>5.7 0 0 3</td>
<td>95.7</td>
<td>1 -1 -4 0</td>
</tr>
<tr>
<td>.86</td>
<td>6.56 0 0 3</td>
<td>96</td>
<td>0 -5 -6 -3</td>
</tr>
<tr>
<td>.75</td>
<td>6.56 0 0 .75</td>
<td>96.56</td>
<td>1 -1 -4 0</td>
</tr>
<tr>
<td>.67</td>
<td>7.23 0 0 .75</td>
<td>96.29</td>
<td>-2 -5 -1 1</td>
</tr>
<tr>
<td>.6</td>
<td>7.23 0 0 1.35</td>
<td>96.89</td>
<td>-2 -5 -1 1</td>
</tr>
</tbody>
</table>

Use NSDP at each iteration.
Discrete relaxation dual

Enumerate relaxations with induced width of 1

Bound = 90

75

95
Combine Lagrangean & discrete relaxation duals

\[
\begin{align*}
\text{x}_1 & \quad \text{x}_2 & \quad \text{x}_3 & \quad \text{x}_4 & \quad \text{x}_5 \\
\text{95} & \quad \text{81} & \quad \text{81} & \quad \text{98.3} \\
\text{(was 90)} & \quad \text{(was 75)} & \quad \text{(was 75)} & \quad \text{(was 95)}
\end{align*}
\]
Summary of relaxations

Value of continuous relaxation

Value of discrete relaxation

Value of discrete relaxation with Lagrangean dual

Value of discrete and Lagrangean dual

Optimal value

$43\frac{1}{3} < 90 < 95 < 96.9 < 98.3 < 105$
Outline

• A motivating example
  Constraint satisfaction approach
  Integer programming approach
  Combined approach

• The search/inference duality
  Complementary solution methods
  A formal duality

• The strengthening/relaxation duality
  Complementary solution methods
  Relaxations for global constraints
  Formal relaxation duality

• Relaxation duality
  An integer programming example
  Continuous relaxation
  Discrete relaxation
  \textit{dependency graph, nonserial dynamic programming}
  Relaxation duality
  Lagrangean & surrogate duals
  Discrete relax + Lagrangean dual
  Discrete relaxation dual
  Discrete + Lagrangean duals
  Summary of relaxations

• Research agenda
Research Agenda

• Identify cutting planes that propagate well.
• Learn how to choose constraints that have a useful continuous relaxation.
• Find continuous relaxations for global constraints not in inequality form (e.g., element, piecewise linear costs).
• Implement variable index sets as well as variable indices.
• Use relaxation duals to discover new relaxations common constraints in constraint satisfaction languages.
Research Agenda

• Identify inference techniques (other than cutting planes) that obtain relaxations that are easy to solve.

• Develop inference-based sensitivity analysis for problem classes.

• Investigate the possibility of using nogoods in branch-and-bound search along with cutting planes and domain reduction.

• Use generalized Benders decomposition to obtain useful nogoods.
Research Agenda

• Experiment with new ways to combine strengthening and relaxation.

• Solve a wide variety of problems with a view to how the search/inference and strengthening/relaxation dualities may be exploited.

• Build a solution technology that unifies and goes beyond classical optimization and constraint satisfaction methods.