Outline of Talk

• Preliminaries
• The WCN Algorithm
• Variants
  – Interactive algorithm
  – Approximation algorithm
• Enhancements
  – Avoiding weakly dominated solutions
  – Improving efficiency
• Examples and Applications
  – Parametric Programming
  – Network Routing
• Computational Results
Biobjective Mixed-integer Programs

A biobjective or bicriterion mixed-integer program (BMIP) is an optimization problem of the form

\[
\begin{align*}
\text{vmax} & \quad f(x) \\
\text{subject to} & \quad x \in X,
\end{align*}
\]

where

- \( f : \mathbb{R}^n \to \mathbb{R}^2 \) is the (bicriteria) objective function, and
- \( X \subset \mathbb{Z}^p \times \mathbb{R}^{n-p} \) is the feasible region, usually defined to be

\[
\{ x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \mid g_i(x) \leq 0, i = 1, \ldots, m \}
\]

for functions \( g_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m \).

The \( \text{vmax} \) operator indicates that the goal is to generate the set of efficient solutions (defined next).
Some Definitions

• We define the set of outcomes to be $Y = f(X) \subset \mathbb{R}^2$.

• In outcome space, BMIP can be restated as

$$\begin{align*}
v_{\text{max}} & \quad y \\
\text{subject to} & \quad y \in f(X),
\end{align*}$$

• For convenience, we will work primarily in outcome space.

• $x^1 \in X$ dominates $x^2 \in X$ if $f_i(x_1) \leq f_i(x_2)$ for $i = 1, 2$ and at least one inequality is strict.

• If both inequalities are strict the dominance is strong (otherwise weak).

• Any $x \in X$ not dominated by another member of $X$ is said to be efficient.

• If $x \in X$ is efficient, then $y = f(x)$ is a Pareto outcome.

• Our goal is to generate the set of all Pareto outcomes.
More Definitions

• We will denote the set of efficient solutions by $X_E$.

• The set of Pareto outcomes is then $Y_E = f(X_E)$.

• We assume that $|Y_E|$ is finite.

• If $x \in X_E$ strongly dominates all members of $X \setminus X_E$, then $x$ is said to be strongly efficient.

• Likewise, if $x \in X_E$ is strongly efficient, then $y = f(x)$ is strongly Pareto.

• If all members of $Y_E$ are strongly Pareto, then $Y_E$ is said to be uniformly dominant.

• The assumption of uniform dominance simplifies computation substantially, but is not satisfied in most practical settings.
Illustrating Pareto Outcomes
A number of algorithms for generating Pareto outcomes have been proposed.

These can be categorized in several ways:

- By output: complete enumeration, partial enumeration, or heuristic enumeration of $Y_E$.
- By user interaction: Interactive or non-interactive.
- By methodology: branch and bound, dynamic programming, implicit enumeration, weighted sums, weighted norms, probing.

We present an algorithm

- that can either partially or completely enumerate the Pareto set,
- has both interactive and non-interactive variants,
- is based on a modified branch and bound algorithm.
Probing Algorithms

- We will focus on *probing algorithms* that *scalarize* the objective, i.e., replace it with a single criterion.
- Such algorithms reduce solution of a BMIP to a series of MIPs.
- The main factor in the running time is then the number of probes.
- The most obvious scalarization is the *weighted sum objective*.
- We replace the original objective with

\[
\max_{y \in f(X)} \beta y_1 + (1 - \beta) y_2
\]

to obtain a parameterized family of MIPs.
Supported Outcomes

- Optimal solutions to weighted sum MIPs are extreme points of $\text{conv}(Y_E)$.
- Such outcomes are called *supported outcomes*.
- The set of all supported outcomes can easily be generated by solving a sequence of MIPs.
- Every supported outcome is Pareto, but the converse is not true.
- This makes it difficult as a tool to generate all Pareto outcomes.
- Chalmet (1986) suggested restricting the subproblems so that each Pareto outcome is supported on some subregion.
- Using this technique, it is possible to generate all Pareto outcomes.
The Weighted Chebyshev Norm

- Another option is to replace the weighted sum objective with a *weighted Chebyshev norm* (WCN) objective.
- The *Chebyshev norm* ($l_\infty$ norm) in $\mathbb{R}^2$ is defined by $\|y\|_\infty = \max\{|y_1|, |y_2|\}$.
- The *weighted Chebyshev norm* with weight $0 \leq \beta \leq 1$ is defined by $\|y\|_\infty = \max\{\beta|y_1|, (1-\beta)|y_2|\}$.
- The *ideal point* $y^*$ is $(y_1^*, y_2^*)$ where $y_i^* = \max_{x \in X} (f(x))_i$.
- Methods based on the WCN select outcomes with minimum WCN distance from the ideal point by solving

$$\min_{y \in f(X)} \{\|y^* - y\|_\infty^\beta\}. \tag{1}$$

- Bowman (1976) showed that every Pareto outcome is a solution to (1) for some $0 \leq \beta \leq 1$.
- The converse only holds if $Y_E$ is uniformly dominant.
Illustrating the WCN

level line for $\beta = 0.57$

level line for $\beta = 0.29$

ideal point
Ordering the Pareto Outcomes

- **Eswaran** (1989) suggested ordering the Pareto outcomes so that
  - $Y_E = \{y_p | 1 \leq p \leq N\}$, and
  - if $p < q$, then $y^p_1 < y^q_1$ (and hence $y^p_2 > y^q_2$).

- For any Pareto outcome $y_p$, if we define
  \[
  \beta_p = (y^*_2 - y^p_2)/(y^*_1 - y^p_1 + y^*_2 - y^p_2),
  \]
  then $y^p$ is the unique optimal outcome for (1) with $\beta = \beta_p$.

- For any pair of Pareto outcomes $y^p$ and $y^q$ with $p < q$, if we define
  \[
  \beta_{pq} = (y^*_2 - y^q_2)/(y^*_1 - y^p_1 + y^*_2 - y^q_2), \tag{2}
  \]
  then $y^p$ and $y^q$ are both optimal outcomes for (1) with $\beta = \beta_{pq}$.

- This provides us with a notion of *adjacency* and *breakpoints*. 
Breakpoints Between Pareto Outcomes with the WCN
Algorithms Based on the WCN

- **Solanki** (1991) proposed an algorithm to generate an approximation to the Pareto set using the WCN.
  - The algorithm probes between pairs of known outcomes for new outcomes by restricting the domain ala Chalmet.
  - The search is controlled by an “error measure,” which can be set to zero to get complete enumeration.
  - The number of probes is asymptotically optimal, but the algorithm does not produce breakpoints (directly).

- **Eswaran** (1989) proposed an algorithm based on binary search over the values of $\beta$.
  - In the worst case, the number of probes is
    \[ |Y_E|(1 - \log(\xi(|Y_E| - 1))), \]
  - where $\xi$ is a chosen error parameter.
  - The algorithm produces only approximate breakpoint information.
The WCN Algorithm

Let $P(\beta)$ be the parameterized subproblem defined by (1) for a given weight $\beta$. The WCN algorithm is then:

**Initialization** Solve $P(1)$ and $P(0)$ to identify optimal outcomes $y^1$ and $y^N$, respectively, and the ideal point $y^* = (y^1_1, y^N_2)$. Set $I = \{(y^1, y^N)\}$.

**Iteration** While $I \neq \emptyset$ do:

1. Remove any $(y^p, y^q)$ from $I$.
2. Compute $\beta_{pq}$ as in (2) and solve $P(\beta_{pq})$. If the outcome is $y^p$ or $y^q$, then $y^p$ and $y^q$ are adjacent in the list $(y^1, y^2, \ldots, y^N)$.
3. Otherwise, a new outcome $y^r$ is generated. Add $(y^p, y^r)$ and $(y^r, y^q)$ to $I$.

This reduces solution of the original BMIP to solution of a sequence of $2N - 1$ MIPs, but still requires the assumption of uniform dominance.
**Solving** $P(\beta)$

- Problem (1) is equivalent to

  $\begin{align*}
  \text{minimize} & \quad z \\
  \text{subject to} & \quad z \geq \beta(y_1^* - y_1), \\
  & \quad z \geq (1 - \beta)(y_2^* - y_2), \text{ and} \\
  & \quad y \in f(X).
  \end{align*}$

- This is a MIP, which can be solved by standard methods.

- This reformulation can still produce weakly dominated outcomes.
Relaxing the Uniform Dominance Requirement

- Dealing with weakly dominated outcomes is the most challenging aspect of these methods.

- We need a method of preventing $P(\beta)$ from producing weakly dominated outcomes.

- Weakly dominated outcomes are the same WCN distance from the ideal point as the outcomes they are dominated by.

- However, they are farther from the ideal point as measured by the $l_p$ norm for $p < \infty$.

- One solution is to replace the WCN with the augmented Chebyshev norm (ACN), defined by

\[
\| (y_1, y_2) \|_{\infty}^{\beta, \rho} = \max\{ \beta |y_1|, (1 - \beta) |y_2| \} + \rho (|y_1| + |y_2|),
\]

where $\rho$ is a small positive number.
Illustrating the ACN
Solving $P(\beta)$ with the ACN

- The problem of determining the outcome closest to the ideal point under this metric is

$$\begin{align*}
\min & \quad z + \rho(|y^*_1 - y_1| + |y^*_2 - y_2|) \\
\text{subject to} & \quad z \geq \beta(y^*_1 - y_1) \\
& \quad z \geq (1 - \beta)(y^*_2 - y_2) \\
& \quad y \in f(X).
\end{align*}$$

(4)

- Because $y^*_k - y_k \geq 0$ for all $y \in f(X)$, the objective function can be rewritten as

$$\min z - \rho(y_1 + y_2).$$

- For fixed $\rho > 0$ small enough:
  - all optimal outcomes for problem (4) are Pareto (in particular, they are not weakly dominated), and
  - for a given Pareto outcome $y$ for problem (4), there exists $0 \leq \hat{\beta} \leq 1$ such that $y$ is the unique outcome to problem (4) with $\beta = \hat{\beta}$.

- In practice, choosing a proper value for $\rho$ can be problematic.
Combinatorial Methods for Eliminating Weakly Dominated Solutions

• In the case of *biobjective linear integer programs* (BLIPs), we can employ combinatorial methods.

• Such a strategy involves implicitly enumerating alternative optimal solutions to $P(\beta)$.

• Weakly dominated outcomes are eliminated with cutting planes during the branch and bound procedure.

• Instead of pruning subproblems that yield feasible outcomes, we continue to search for alternative optima that dominate the current incumbent.

• To do so, we determine which of the two constraints

\[
\begin{align*}
  z &\geq \beta(y_1^* - y_1) \\
  z &\geq (1 - \beta)(y_2^* - y_2)
\end{align*}
\]

from problem (1) is binding at $\hat{y}$. 
Combinatorial Methods for Eliminating Weakly Dominated Solutions (cont’d)

- Let $\epsilon_1$ and $\epsilon_2$ be such that if $y_r$ is a new outcome between $y^p$ and $y^q$, then $y_i^r \geq \min\{y_i^p, y_i^q\} + \epsilon_i$, for $i = 1, 2$.

- If only the first constraint is binding, then the cut

  $$y_1 \geq \hat{y}_1 + \epsilon_1$$

  is valid for any outcome that dominates $\hat{y}$.

- If only the second constraint is binding, then the cut

  $$y_2 \geq \hat{y}_2 + \epsilon_2$$

  is valid for any outcome that dominates $\hat{y}$.

- If both constraints are binding, either cut can be imposed.
Hybrid Methods

• In practice, the ACN method is fast, but choosing the proper value of $\rho$ is problematic.

• Combinatorial methods are less susceptible to numerical difficulties, but are slower.

• Combining the two methods improves running times and reduces dependence on the magnitude of $\rho$. 
Other Enhancements to the Algorithm

• In Step 2, any new outcome $y^r$ will have $y^r_1 > y^p_1$ and $y^r_2 > y^q_2$.

• If no such outcome exists, then the subproblem solver must still re-prove the optimality of $y^p$ or $y^q$.

• Then it must be the case that

$$\|y^* - y^r\|_{\infty}^{\beta_{pq}} + \min\{\beta_{pq}\epsilon_1, (1 - \beta_{pq})\epsilon_2\} \leq \|y^* - y^p\|_{\infty}^{\beta_{pq}} = \|y^* - y^q\|_{\infty}^{\beta_{pq}}$$

• Hence, we can impose an a priori upper bound of

$$\|y^* - y^p\|_{\infty}^{\beta_{pq}} - \min\{\beta_{pq}\epsilon_1, (1 - \beta_{pq})\epsilon_2\}$$

when solving the subproblem $P(\beta_{pq})$.

• With this upper bound, each subproblem will either be infeasible or produce a new outcome.
Using Warm Starting

• We have been developing methodology for *warm starting* branch and bound computations.

• Because the WCN algorithm involves solving a sequence of slightly modified MILPs, warm starting can be used.

• **Three approaches**
  - Warm start from the result of the previous iteration.
  - Solve a “base” problem first and warm each subsequent problem from there.
  - Warm start from the “closest” previously solved subproblem.

• In addition, we can optionally save the global cut pool from iteration to iteration.
Approximating the Pareto Set

• If the number of Pareto outcomes is large, it may not be desirable to generate the entire set.

• If only part of the set is generated, it is important that the subset be well-distributed among the entire set.

• Any probing algorithm can generate an approximation to the Pareto set by terminating early.
  – In such case, the key is to avoid failed probes whenever possible.
  – The order in which the intervals are explored affects both the distribution of solutions and the number of failed probes.
  – Empirically, FIFO selection schemes tend to distribute the points well and also minimize the number of failed probes.

• Another approach is to generate the set of supported solutions.

• This can be an extremely bad approximation in some cases.
Interactive Algorithms

- Interactive algorithms offer another method of avoiding enumeration of the entire set.

- In an interactive algorithm, the user guides the solution process by providing real-time feedback.

- This feedback provides information of the user’s unknown utility function.

- A simple feedback mechanism for the WCN algorithm is to allow the user to select the next interval to be explored.

- In this way, the user is able to zero in on the portion of the tradeoff curve that is most attractive.

- There are a number of mechanisms for providing estimated tradeoff information to the user as the algorithm progresses.
Implementation: A Brief Overview of SYMPHONY

- **SYMPHONY** is an open-source software package for solving and analyzing mixed-integer linear programs (MILPs).

- **SYMPHONY** can be used in three distinct modes.
  - **Black box solver**: Solve generic MILPs (command line or shell).
  - **Callable library**: Call SYMPHONY from a C/C++ code.
  - **Framework**: Develop a customized black box solver or callable library.

- Makes extensive use of the Computational Infrastructure for Operations Research (COIN-OR) libraries ([www.coin-or.org](http://www.coin-or.org)).

- Complete documentation, code samples, data sets, and application plug-ins are available ([www.BranchAndCut.org](http://www.BranchAndCut.org)).

- Advanced features
  - Warm starting
  - Bicriteria solve
  - Sensitivity analysis
  - Parallel execution mode
Example: Bicriteria ILP

• Consider the following bicriteria ILP:

\[
\begin{align*}
\text{vmax} & \quad [8x_1, x_2] \\
\text{s.t.} & \quad 7x_1 + x_2 \leq 56 \\
& \quad 28x_1 + 9x_2 \leq 252 \\
& \quad 3x_1 + 7x_2 \leq 105 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]

• The following code solves this model using SYMPHONY.

```c
int main(int argc, char **argv)
{
    OsiSymSolverInterface si;
    si.parseCommandLine(argc, argv);
    si.setObj2Coeff(1, 1);
    si.loadProblem();
    si.multiCriteriaBranchAndBound();
}
```
Example: Pareto Outcomes for Example

Non-dominated Solutions
Example: Sensitivity Analysis

- By examining the supported solutions and break points, we can easily determine $p(\theta)$, the optimal solution to the ILP with objective $8x_1 + \theta x_2$.

<table>
<thead>
<tr>
<th>$\theta$ range</th>
<th>$p(\theta)$</th>
<th>$x_1^*$</th>
<th>$x_2^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(-\infty, 1.333)$</td>
<td>64</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>$(1.333, 2.667)$</td>
<td>$56 + 6\theta$</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>$(2.667, 8.000)$</td>
<td>$40 + 12\theta$</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>$(8.000, 16.000)$</td>
<td>$32 + 13\theta$</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>$(16.000, \infty)$</td>
<td>$15\theta$</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>
Example: Price Function

Price Function

Range

Z

0.00
25.00
50.00
75.00
100.00
125.00
150.00
175.00
200.00
225.00
250.00
275.00
300.00
325.00
Application: Capacitated Network Routing Problems

- Using SYMPHONY, we developed a custom solver for a class of capacitated network routing problems (CNRPs).

- A single commodity is supplied to a set of customers from a single supply point.

- We must design the network and route the demand, obeying capacity and other side constraints.

- We wish to consider both
  - the cost of construction (the sum of lengths of all links), and
  - the latency of the resulting network (the sum of length multiplied by demand carried for all links).

- These are competing objectives, so we can analyze the tradeoff by using the SYMPHONY multicriteria solver.
Application: Efficient Solutions for a Small CNRP

(a)  (b)  (c)  (d)
Application: Pareto Outcomes for a Small CNRP
Application: Pareto Outcomes for a Larger CNRP
Computational Results: Comparing WCN with Bisection Search

### Knapsack

<table>
<thead>
<tr>
<th>Size</th>
<th>Iterations</th>
<th>WCN</th>
<th>Δ from WCN</th>
<th>Outcomes Found</th>
<th>WCN</th>
<th>Δ from WCN</th>
<th>Max Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>10⁻¹</td>
<td>10⁻²</td>
<td>10⁻³</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>278</td>
<td>12</td>
<td>300</td>
<td>679</td>
<td>149</td>
<td>−17</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>364</td>
<td>−1</td>
<td>390</td>
<td>896</td>
<td>192</td>
<td>−22</td>
<td>−2</td>
</tr>
<tr>
<td>30</td>
<td>324</td>
<td>−43</td>
<td>246</td>
<td>712</td>
<td>167</td>
<td>−25</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>490</td>
<td>−108</td>
<td>235</td>
<td>898</td>
<td>250</td>
<td>−55</td>
<td>−11</td>
</tr>
<tr>
<td>50</td>
<td>686</td>
<td>−138</td>
<td>235</td>
<td>1123</td>
<td>348</td>
<td>−69</td>
<td>−9</td>
</tr>
<tr>
<td>Totals</td>
<td>2142</td>
<td>−278</td>
<td>1406</td>
<td>4308</td>
<td>1106</td>
<td>−188</td>
<td>−22</td>
</tr>
</tbody>
</table>

### CNRP

<table>
<thead>
<tr>
<th>Name</th>
<th>Iterations</th>
<th>WCN</th>
<th>Δ from WCN</th>
<th>Outcomes Found</th>
<th>WCN</th>
<th>Δ from WCN</th>
<th>Max Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>10⁻¹</td>
<td>10⁻²</td>
<td>10⁻³</td>
<td>0</td>
</tr>
<tr>
<td>att48</td>
<td>147</td>
<td>−35</td>
<td>−9</td>
<td>104</td>
<td>74</td>
<td>−18</td>
<td>−15</td>
</tr>
<tr>
<td>Totals</td>
<td>2381</td>
<td>−264</td>
<td>724</td>
<td>3794</td>
<td>1207</td>
<td>−135</td>
<td>−13</td>
</tr>
</tbody>
</table>
### Computational Results: Comparing WCN with ACN

#### Knapsack

<table>
<thead>
<tr>
<th>Size</th>
<th>WCN</th>
<th>( \Delta ) from WCN</th>
<th>Max Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>278</td>
<td>(-4)</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>364</td>
<td>(-6)</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>324</td>
<td>(-6)</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>490</td>
<td>(-24)</td>
<td>1</td>
</tr>
<tr>
<td>50</td>
<td>686</td>
<td>(-28)</td>
<td>3</td>
</tr>
<tr>
<td>Totals</td>
<td>2142</td>
<td>(-70)</td>
<td>3</td>
</tr>
</tbody>
</table>

#### CNRP

<table>
<thead>
<tr>
<th>Name</th>
<th>WCN</th>
<th>( \Delta ) from WCN</th>
<th>Max Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td>att48</td>
<td>147</td>
<td>(-140)</td>
<td>44</td>
</tr>
<tr>
<td>Totals</td>
<td>2381</td>
<td>(-2056)</td>
<td>18</td>
</tr>
</tbody>
</table>
# Computational Results: Comparing WCN with Hybrid ACN

## Knapsack

<table>
<thead>
<tr>
<th>Size</th>
<th>WCN</th>
<th>$\Delta$ from WCN</th>
<th>$\Delta$ from WCN</th>
<th>Max Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$10^{-2}$</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>10</td>
<td>278</td>
<td>-4 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>364</td>
<td>-6 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>324</td>
<td>-6 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>490</td>
<td>-24 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>686</td>
<td>-28 -4 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td>2142</td>
<td>-68 -4 0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

## CNRP

<table>
<thead>
<tr>
<th>Name</th>
<th>WCN</th>
<th>$\Delta$ from WCN</th>
<th>$\Delta$ from WCN</th>
<th>Max Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>att48</td>
<td>147</td>
<td>-106 -62 -6</td>
<td>74 -53 -31 -3</td>
<td>17 8 2</td>
</tr>
<tr>
<td>Totals</td>
<td>2381</td>
<td>-1012 -44 -2</td>
<td>1207 -612 -22 -1</td>
<td>5 1 1</td>
</tr>
</tbody>
</table>
## Computational Results: Comparing WCN with ACN and Hybrid ACN (CPU Time)

### Knapsack

<table>
<thead>
<tr>
<th>Size</th>
<th>WCN</th>
<th>$\Delta$ from WCN</th>
<th>CPU Time (ACN)</th>
<th>WCN</th>
<th>$\Delta$ from WCN</th>
<th>CPU Time (Hybrid)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>$10^{-2}$</td>
<td>$10^{-3}$</td>
<td>0</td>
<td>$10^{-2}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>10</td>
<td>13.18</td>
<td>0.06</td>
<td>−0.23</td>
<td>13.18</td>
<td>0.34</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>−1.33</td>
<td>−0.41</td>
<td>17.46</td>
<td>−1.17</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>−1.28</td>
<td>−0.43</td>
<td>24.93</td>
<td>−1.02</td>
<td>−0.11</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>−5.69</td>
<td>−1.70</td>
<td>24.93</td>
<td>−1.02</td>
<td>−0.11</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>−27.18</td>
<td>−3.78</td>
<td>65.88</td>
<td>−4.89</td>
<td>−1.09</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>−35.42</td>
<td>−6.55</td>
<td>139.42</td>
<td>−13.04</td>
<td>−3.37</td>
</tr>
<tr>
<td>Totals</td>
<td>260.87</td>
<td>−35.42</td>
<td>−6.55</td>
<td>260.87</td>
<td>−19.78</td>
<td>−4.42</td>
</tr>
</tbody>
</table>

### CNRP

<table>
<thead>
<tr>
<th>Name</th>
<th>WCN</th>
<th>$\Delta$ from WCN</th>
<th>CPU Time (ACN)</th>
<th>WCN</th>
<th>$\Delta$ from WCN</th>
<th>CPU Time (Hybrid)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>$10^{-2}$</td>
<td>$10^{-3}$</td>
<td>0</td>
<td>$10^{-2}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>att48</td>
<td>83.67</td>
<td>−80.14</td>
<td>−59.83</td>
<td>83.67</td>
<td>−59.34</td>
<td>−30.19</td>
</tr>
<tr>
<td></td>
<td>8122.36</td>
<td>−7728.51</td>
<td>−5244.54</td>
<td>8122.36</td>
<td>−5481.53</td>
<td>−1531.35</td>
</tr>
<tr>
<td>Totals</td>
<td>8122.36</td>
<td>−7728.51</td>
<td>−5244.54</td>
<td>8122.36</td>
<td>−5481.53</td>
<td>−1531.35</td>
</tr>
</tbody>
</table>
Computational Results: Using Warm Starting to Solve CNRP Instances

These are results using SYMPHONY to solve CNRP instances with two different warm starting strategies.
Conclusion

• Generating the complete set of Pareto outcomes is a challenging computational problem.

• We presented a new algorithm for solving bicriteria mixed-integer programs.

• The algorithm is
  – asymptotically optimal,
  – generates exact breakpoints,
  – has good numerical properties, and
  – can exploits modern solution techniques.

• We have shown how this algorithm is implemented in the SYMPHONY MILP solver framework.

• Future work
  – Improvements to warm starting procedures
  – Parallelization
  – More than two objective
Shameless Plug

- The software discussed in this talk is available for free download from the **Computational Infrastructure for Operations Research** Web site

  [www.coin-or.org](http://www.coin-or.org)

- **The COIN-OR Project**
  - An **initiative** promoting the development and use of interoperable, open-source software for operations research.
  - A **consortium** of researchers in both industry and academia dedicated to improving the state of computational research in OR.
  - A non-profit educational foundation known as the COIN-OR Foundation.

- **The COIN-OR Repository**
  - A **library** of interoperable software tools for building optimization codes, as well as some stand-alone packages.
  - A **venue for peer review** of OR software tools.
  - A **development platform** for open source projects, including a CVS repository.
More Information

• SYMPHONY
  – Prepackaged releases can be obtained from www.BranchAndCut.org.
  – Up-to-date source is available from www.coin-or.org.
  – Available Solvers
    - Generic MILP
    - Traveling Salesman Problem
    - Vehicle Routing Problem
    - Mixed Postman Problem
    - Bicriteria Knapsack Solver
    - Set Partitioning Problem
    - Matching Problem
    - Network Routing

• For references and further details, see An Improved Algorithm for Biobjective Integer Programming, to appear in Annals of OR, available from
  www.lehigh.edu/~tkr2

• Overviews of multiobjective integer programming
  – Climaco (1997)
  – Ehrgott and Wiecek (2005)