Washington State University
Department of Mathematics

Cascade Knapsack Problems

Bala Krishnamoorthy
Washington State University

joint work with

Gábor Pataki, UNC Chapel Hill

MIP 2008

August 04, 2008

Hard IP Instances

Hard IP Instances

- worst case behavior of IP algorithms

Hard IP Instances

- worst case behavior of IP algorithms
- feasibility problems;

Hard IP Instances

- worst case behavior of IP algorithms
- feasibility problems; integer infeasible

Hard IP Instances

- worst case behavior of IP algorithms
- feasibility problems; integer infeasible
- hard for branch-and-bound (B&B), cutting planes

Hard IP Instances

- worst case behavior of IP algorithms
- feasibility problems; integer infeasible
- hard for branch-and-bound (B&B), cutting planes
- prove bounds on running time, # B&B nodes

Hard IP Instances

- worst case behavior of IP algorithms
- feasibility problems; integer infeasible
- hard for branch-and-bound (B&B), cutting planes
- prove bounds on running time, # B&B nodes
- gain computational insights

Hard IP Instances

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint
  - provably hard for branch-and-bound,

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint
  - provably hard for branch-and-bound, cutting planes

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint
  - provably hard for branch-and-bound, cutting planes
  - can analyze mathematically

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint
  - provably hard for branch-and-bound, cutting planes
  - can analyze mathematically
- marketshare problems

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint
  - provably hard for branch-and-bound, cutting planes
  - can analyze mathematically
- marketshare problems
  - binary IPs with a few dense constraints

Hard IP Instances

- hard knapsack problems: $\{\beta_1 \leq \boldsymbol{ax} \leq \beta_2 \,|\, \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{x} \in \mathbb{Z}^n\}$
  - "simple" – one constraint
  - provably hard for branch-and-bound, cutting planes
  - can analyze mathematically
- marketshare problems
  - binary IPs with a few dense constraints
  - computationally hard

Hard Knapsack Problems

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j \,/2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j \,/2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j \,/2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j \,/2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j / 2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$

  ▶ (ordinary) B&B takes at least $2^{(n-1)/2}$ nodes

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j / 2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$
  - ▶ (ordinary) B&B takes at least $2^{(n-1)/2}$ nodes
  - ▶ preprocessing, or single knapsack cover inequalities kill them

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j / 2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$

  ▶ (ordinary) B&B takes at least $2^{(n-1)/2}$ nodes

  ▶ preprocessing, or single knapsack cover inequalities kill them

- $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j / 2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$
  - ▶ (ordinary) B&B takes at least $2^{(n-1)/2}$ nodes
  - ▶ preprocessing, or single knapsack cover inequalities kill them
- $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Chvátal (80): $a_j = U[1, 10^{n/2}]$; Hunsaker and Tovey (04)

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j / 2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$
  - ▶ (ordinary) B&B takes at least $2^{(n-1)/2}$ nodes
  - ▶ preprocessing, or single knapsack cover inequalities kill them
- $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0, 1\}$
  - Chvátal (80): $a_j = U[1, 10^{n/2}]$; Hunsaker and Tovey (04)
  - Gu, Nemhauser, Savelsberg (98,99): $a_j \approx 2^{n/20}$

Hard Knapsack Problems

- with $\beta_1 = \beta_2 = \beta = \left\lfloor \sum_j a_j /2 \right\rfloor$, $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0,1\}$
  - Jeroslow (74): $a_j = 2$, n is odd $(2x_1 + \cdots + 2x_n = n)$
  - Avis (Chvátal, 80): $a_j = n(n+1) + j$
  - Todd (Chvátal, 80): $a_j = 2^{n+\ell+1} + 2^{\ell+j} + 1$ for $\ell = \lfloor \log 2n \rfloor$

  ▶ (ordinary) B&B takes at least $2^{(n-1)/2}$ nodes

  ▶ preprocessing, or single knapsack cover inequalities kill them

- $\boldsymbol{u} = \boldsymbol{e}$, i.e., $x_j \in \{0,1\}$
  - Chvátal (80): $a_j = U[1, 10^{n/2}]$; Hunsaker and Tovey (04)
  - Gu, Nemhauser, Savelsberg (98,99): $a_j \approx 2^{n/20}$

  ▶ B&C using lifted cover inequalities takes at least $2^{n/30}$ nodes

More Hard Knapsacks

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97)

More Hard Knapsacks

- $\beta_2 \approx \text{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\; u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04)

More Hard Knapsacks

- $\beta_2 \approx \text{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04) ▶ Aardal et al. (00) reformulation

More Hard Knapsacks

- $\beta_2 \approx \text{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04) ▶ Aardal et al. (00) reformulation
    ▶ equality version of Cornujols et al. knapsack

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04) ▶ Aardal et al. (00) reformulation
    - ▶ equality version of Cornujols et al. knapsack
    - ▶ $\mathrm{Frob}(\boldsymbol{a})$ is the *largest* rhs, hence gives "hardest" instance

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04) ▶ Aardal et al. (00) reformulation
    - ▶ equality version of Cornujols et al. knapsack
    - ▶ $\mathrm{Frob}(\boldsymbol{a})$ is the *largest* rhs, hence gives "hardest" instance
    - ▶ for $\boldsymbol{a} = \boldsymbol{p}M + \boldsymbol{r}$, lower bound for $\mathrm{Frob}(\boldsymbol{a})$ quadratic in M

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04) ▶ Aardal et al. (00) reformulation
    - ▶ equality version of Cornujols et al. knapsack
    - ▶ $\mathrm{Frob}(\boldsymbol{a})$ is the *largest* rhs, hence gives "hardest" instance
    - ▶ for $\boldsymbol{a} = \boldsymbol{p}M + \boldsymbol{r}$, lower bound for $\mathrm{Frob}(\boldsymbol{a})$ quadratic in M
    - ▶ *large* rhs implies hard for B&B

More Hard Knapsacks

- $\beta_2 \approx \mathrm{Frob}(\boldsymbol{a}),\ u_j = +\infty$, i.e., x_j are **unbounded**
  - Cornuejols et al. (97) ▶ generating sets
  - Aardal and Lenstra (04) ▶ Aardal et al. (00) reformulation
    - ▶ equality version of Cornujols et al. knapsack
    - ▶ $\mathrm{Frob}(\boldsymbol{a})$ is the *largest* rhs, hence gives "hardest" instance
    - ▶ for $\boldsymbol{a} = \boldsymbol{p}M + \boldsymbol{r}$, lower bound for $\mathrm{Frob}(\boldsymbol{a})$ quadratic in M
    - ▶ *large* rhs implies hard for B&B

- We study a very general class of knapsacks

$t + 1$-level Decomposable Knapsack Problem

$t + 1$-level Decomposable Knapsack Problem

- $\{\beta_1 \leq \boldsymbol{a}\boldsymbol{x} \leq \beta_2 \,|\, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$ with

$t + 1$-level Decomposable Knapsack Problem

- $\{\beta_1 \le \boldsymbol{a}\boldsymbol{x} \le \beta_2 \,|\, \mathbf{0} \le \boldsymbol{x} \le \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$ with

$$\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r};\ \boldsymbol{p}_i \in \mathbb{Z}^n_{>0}, M_i \in \mathbb{Z}_{>0};\ M_i > M_{i+1}$$

$t+1$-level Decomposable Knapsack Problem

- $\{\beta_1 \leq \boldsymbol{a}\boldsymbol{x} \leq \beta_2 \,|\, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$ with

$$\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r};\ \boldsymbol{p}_i \in \mathbb{Z}^n_{>0}, M_i \in \mathbb{Z}_{>0};\ M_i > M_{i+1}$$

- denoted as $t+1$-DKP

$t+1$-level Decomposable Knapsack Problem

- $\{\beta_1 \leq \boldsymbol{a}\boldsymbol{x} \leq \beta_2 \,|\, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$ with

  $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r};\ \boldsymbol{p}_i \in \mathbb{Z}^n_{>0}, M_i \in \mathbb{Z}_{>0};\ M_i > M_{i+1}$

- denoted as $t+1$-DKP
- for $t = 1$, we write $\boldsymbol{p}_1 = \boldsymbol{p},\ M_1 = M$, and call it simply DKP

$t+1$-level Decomposable Knapsack Problem

- $\{\beta_1 \leq \boldsymbol{a}\boldsymbol{x} \leq \beta_2 \,|\, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u},\, \boldsymbol{x} \in \mathbb{Z}^n\}$ with

  $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r};\ \boldsymbol{p}_i \in \mathbb{Z}^n_{>0}, M_i \in \mathbb{Z}_{>0};\ M_i > M_{i+1}$

- denoted as $t+1$-DKP
- for $t=1$, we write $\boldsymbol{p}_1 = \boldsymbol{p},\ M_1 = M$, and call it simply DKP
- Krishnamoorthy and Pataki (06) - Column basis reduction and decomposable knapsack problems (preprint available in Optimization Online)

Special Cases of $t+1$-DKPs for $t=1$

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p}=\boldsymbol{e}$, $M=2, \boldsymbol{r}=\boldsymbol{0}, \boldsymbol{u}=\boldsymbol{e}$ gives Jeroslow knapsack $2x_1+\cdots+2x_n=n$

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p}=\boldsymbol{e}$, $M=2, \boldsymbol{r}=\boldsymbol{0}, \boldsymbol{u}=\boldsymbol{e}$ gives Jeroslow knapsack
  $2x_1+\cdots+2x_n=n$

- other special cases: with $\boldsymbol{u}=\boldsymbol{e}$
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=n(n+1), \boldsymbol{r}=(1,\ldots,n)$: Avis knapsack

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p} = \boldsymbol{e}$, $M = 2, \boldsymbol{r} = \boldsymbol{0}, \boldsymbol{u} = \boldsymbol{e}$ gives Jeroslow knapsack $2x_1 + \cdots + 2x_n = n$
- other special cases: with $\boldsymbol{u} = \boldsymbol{e}$
  - $\boldsymbol{p} = \boldsymbol{e}$, $M = n(n+1), \boldsymbol{r} = (1, \ldots, n)$: Avis knapsack
  - $\boldsymbol{p} = \boldsymbol{e}$, $M = 2^{n+\ell+1}, r_j = 2^{\ell+j} + 1$: Todd knapsack

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p}=\boldsymbol{e}$, $M=2, \boldsymbol{r}=\mathbf{0}, \boldsymbol{u}=\boldsymbol{e}$ gives Jeroslow knapsack
  $2x_1+\cdots+2x_n=n$

- other special cases: with $\boldsymbol{u}=\boldsymbol{e}$
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=n(n+1), \boldsymbol{r}=(1,\ldots,n)$: Avis knapsack
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=2^{n+\ell+1}, r_j=2^{\ell+j}+1$: Todd knapsack
  - modification of above (Todd) settings: Gu et al. knapsacks

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p}=\boldsymbol{e}$, $M=2, \boldsymbol{r}=\mathbf{0}, \boldsymbol{u}=\boldsymbol{e}$ gives Jeroslow knapsack
  $2x_1+\cdots+2x_n=n$

- other special cases: with $\boldsymbol{u}=\boldsymbol{e}$
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=n(n+1), \boldsymbol{r}=(1,\ldots,n)$: Avis knapsack
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=2^{n+\ell+1}, r_j=2^{\ell+j}+1$: Todd knapsack
  - modification of above (Todd) settings: Gu et al. knapsacks

- with $\boldsymbol{u}=+\infty$

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p}=\boldsymbol{e}$, $M=2, \boldsymbol{r}=\mathbf{0}, \boldsymbol{u}=\boldsymbol{e}$ gives Jeroslow knapsack
  $2x_1+\cdots+2x_n=n$

- other special cases: with $\boldsymbol{u}=\boldsymbol{e}$
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=n(n+1), \boldsymbol{r}=(1,\ldots,n)$: Avis knapsack
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=2^{n+\ell+1}, r_j=2^{\ell+j}+1$: Todd knapsack
  - modification of above (Todd) settings: Gu et al. knapsacks

- with $\boldsymbol{u}=+\infty$
  - $\boldsymbol{p}>0$: Cornuejols et al. knapsacks

Special Cases of $t+1$-DKPs for $t=1$

- $\boldsymbol{p}=\boldsymbol{e}$, $M=2, \boldsymbol{r}=\mathbf{0}, \boldsymbol{u}=\boldsymbol{e}$ gives Jeroslow knapsack
  $2x_1+\cdots+2x_n=n$

- other special cases: with $\boldsymbol{u}=\boldsymbol{e}$
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=n(n+1), \boldsymbol{r}=(1,\ldots,n)$: Avis knapsack
  - $\boldsymbol{p}=\boldsymbol{e}$, $M=2^{n+\ell+1}, r_j=2^{\ell+j}+1$: Todd knapsack
  - modification of above (Todd) settings: Gu et al. knapsacks

- with $\boldsymbol{u}=+\infty$
  - $\boldsymbol{p}>0$: Cornuejols et al. knapsacks
  - same as above, but equality: Aardal & Lenstra knapsack

DKPs: Properties

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \le k \vee \boldsymbol{px} \ge k+1$, for some integer k
- *easy* if branching on hyperplane $\boldsymbol{px}$

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k
- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k

- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B

  Theorem: If $u_j = +\infty$, then B&B takes at least

$$\binom{\lfloor k / \|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k

- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B
  Theorem: If $u_j = +\infty$, then B&B takes at least

$$\binom{\lfloor k / \|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$

  - ▶ easiness for hyperplane branching $\Rightarrow$ hardness for ordinary B&B

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k

- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B
  Theorem: If $u_j = +\infty$, then B&B takes at least
  $$\binom{\lfloor k / \|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$
  - ▶ easiness for hyperplane branching ⇒ hardness for ordinary B&B

- Krishnamoorthy (07): generic lower bound for the # B&B nodes for infeasible integer knapsacks;

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k
- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B
  Theorem: If $u_j = +\infty$, then B&B takes at least
  $$\binom{\lfloor k / \|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$
  - easiness for hyperplane branching $\Rightarrow$ hardness for ordinary B&B
- Krishnamoorthy (07): generic lower bound for the # B&B nodes for infeasible integer knapsacks; M^{n-1} for DKPs

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \leq k \vee \boldsymbol{px} \geq k+1$, for some integer k

- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B
Theorem: If $u_j = +\infty$, then B&B takes at least
$$\binom{\lfloor k/\,\|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$
  - ▶ easiness for hyperplane branching $\Rightarrow$ hardness for ordinary B&B

- Krishnamoorthy (07): generic lower bound for the # B&B nodes for infeasible integer knapsacks; M^{n-1} for DKPs

- Recipe for generating DKPs (for $t = 1$):

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{p}\boldsymbol{x} \le k \vee \boldsymbol{p}\boldsymbol{x} \ge k+1$, for some integer k

- *easy* if branching on hyperplane $\boldsymbol{p}\boldsymbol{x}$ but hard for B&B
  Theorem: If $u_j = +\infty$, then B&B takes at least
  $$\binom{\lfloor k/\, \|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$
  - easiness for hyperplane branching $\Rightarrow$ hardness for ordinary B&B

- Krishnamoorthy (07): generic lower bound for the # B&B nodes for infeasible integer knapsacks; M^{n-1} for DKPs

- Recipe for generating DKPs (for $t = 1$): INPUT: $\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{u}$;

DKPs: Properties

- infeasibility proven by split disjunction $\boldsymbol{px} \le k \vee \boldsymbol{px} \ge k+1$, for some integer k

- *easy* if branching on hyperplane $\boldsymbol{px}$ but hard for B&B
  Theorem: If $u_j = +\infty$, then B&B takes at least
  $$\binom{\lfloor k/\, \|\boldsymbol{p}\|_\infty \rfloor + n - 1}{n-1} \text{ nodes}$$
  - easiness for hyperplane branching $\Rightarrow$ hardness for ordinary B&B

- Krishnamoorthy (07): generic lower bound for the # B&B nodes for infeasible integer knapsacks; M^{n-1} for DKPs

- Recipe for generating DKPs (for $t = 1$): INPUT: $\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{u}$; OUTPUT: M, β_1, β_2 s.t. infeasibility of DKP is proven by branching on $\boldsymbol{px}$

Rangespace Reformulation (RSRef)

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\}$$

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \rightarrow \{\boldsymbol{b}' \leq (AU)\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \ \rightarrow \ \{\boldsymbol{b}' \leq (AU)\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

U is unimodular, found by basis reduction

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \ \rightarrow\ \{\boldsymbol{b}' \leq (AU)\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

U is unimodular, found by basis reduction

– simplifies and generalizes the Aardal et al. reformulation

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \ \rightarrow\ \{\boldsymbol{b}' \leq (AU)\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

  U is unimodular, found by basis reduction

  – simplifies and generalizes the Aardal et al. reformulation
  – dimension remains the same

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \le A\boldsymbol{x} \le \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \ \rightarrow\ \{\boldsymbol{b}' \le (AU)\boldsymbol{y} \le \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

  U is unimodular, found by basis reduction

  – simplifies and generalizes the Aardal et al. reformulation
  – dimension remains the same

- DKPs become *easy* after RSRef is applied

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \ \rightarrow\ \{\boldsymbol{b}' \leq (AU)\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

  U is unimodular, found by basis reduction

  – simplifies and generalizes the Aardal et al. reformulation
  – dimension remains the same

- DKPs become *easy* after RSRef is applied

- branching on $\boldsymbol{px}$ $\iff$ branching on "last few" y_j's

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \le A\boldsymbol{x} \le \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \ \rightarrow\ \{\boldsymbol{b}' \le (AU)\boldsymbol{y} \le \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

  U is unimodular, found by basis reduction

  – simplifies and generalizes the Aardal et al. reformulation
  – dimension remains the same

- DKPs become *easy* after RSRef is applied

- branching on $\boldsymbol{px}$ $\iff$ branching on "last few" y_j's
  – e.g., $n = 50$, $x_j \in \{0,1\}$, $p_j \in [1,10]$, $r_j \in [-10,10]$, $M = 10^4$: CPLEX 9.0 takes $\ge$ 6.7 million B&B nodes

Rangespace Reformulation (RSRef)

- reformulation of general IPs

$$\{\boldsymbol{b}' \leq A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\} \;\rightarrow\; \{\boldsymbol{b}' \leq (AU)\boldsymbol{y} \leq \boldsymbol{b},\ \boldsymbol{y} \in \mathbb{Z}^n\}$$

  U is unimodular, found by basis reduction

  – simplifies and generalizes the Aardal et al. reformulation
  – dimension remains the same

- DKPs become *easy* after RSRef is applied

- branching on $\boldsymbol{px}$ $\iff$ branching on "last few" y_j's
  – e.g., $n = 50$, $x_j \in \{0, 1\}$, $p_j \in [1, 10]$, $r_j \in [-10, 10]$, $M = 10^4$: CPLEX 9.0 takes $\geq$ 6.7 million B&B nodes
  – RSRef solves in root node

DKP example in 2D

Let $\boldsymbol{p} = (1,\, 1),\;\; M = 20,\;\; \boldsymbol{r} = (1,\, -1),\;\; \boldsymbol{u} = (6,\, 6)$

DKP example in 2D

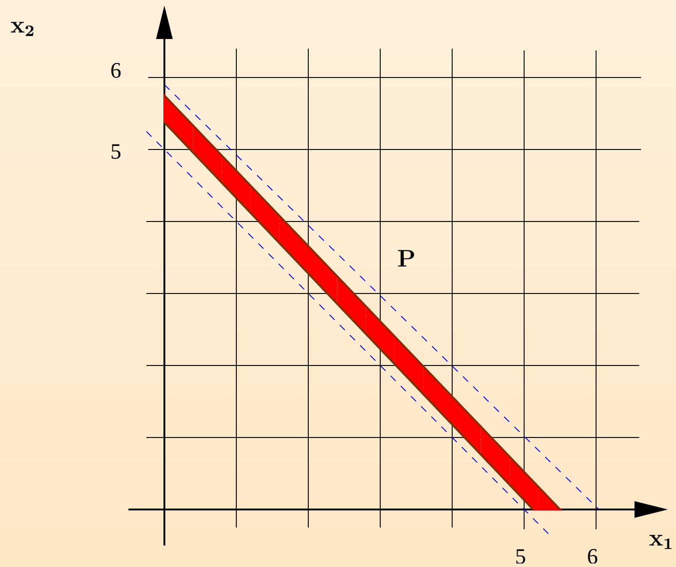
Let $\boldsymbol{p} = (1,\, 1),\ M = 20,\ \boldsymbol{r} = (1,\, -1),\ \boldsymbol{u} = (6,\, 6)$

$$\begin{array}{rcl} 106 & \leq & 21x_1 + 19x_2 & \leq & 113 \\ 0 & \leq & x_1, x_2 & \leq & 6 \\ & & x_1, x_2 \in \mathbb{Z} & & \end{array}$$

DKP example in 2D

Let $\boldsymbol{p} = (1,\, 1),\ M = 20,\ \boldsymbol{r} = (1,\, -1),\ \boldsymbol{u} = (6,\, 6)$

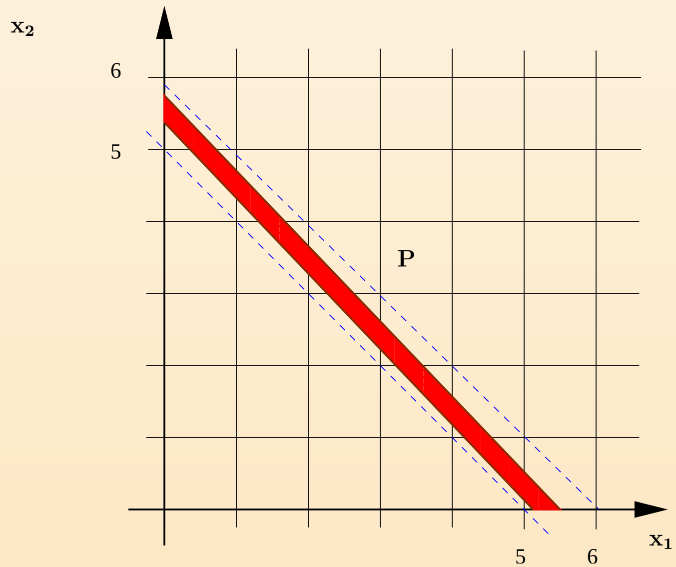
$$\begin{array}{rcl} 106 \leq & 21x_1 + 19x_2 & \leq 113 \\ 0 \leq & x_1, x_2 & \leq 6 \\ & x_1, x_2 \in \mathbb{Z} & \end{array}$$



DKP example in 2D

Let $\boldsymbol{p} = (1,\ 1),\ M = 20,\ \boldsymbol{r} = (1,\ -1),\ \boldsymbol{u} = (6,\ 6)$

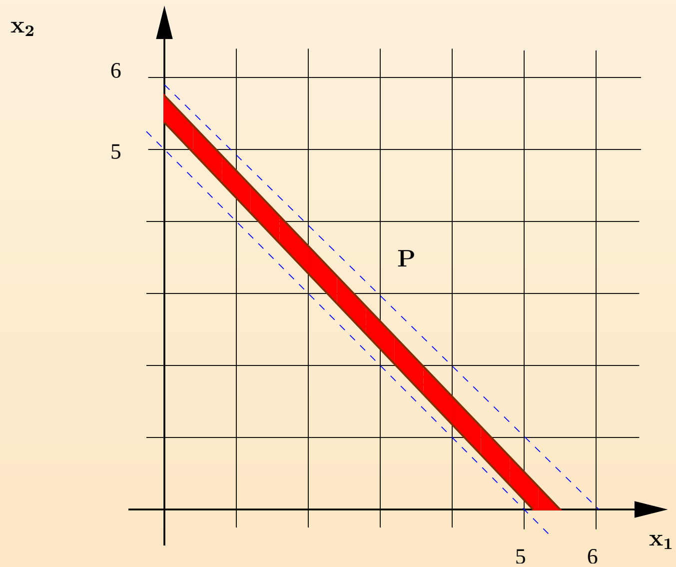
$$\begin{array}{rcl} 106 \leq & 21x_1 + 19x_2 & \leq 113 \\ 0 \leq & x_1, x_2 & \leq 6 \\ & x_1, x_2 \in \mathbb{Z} & \end{array} \quad \rightarrow \quad \begin{array}{rcl} 106 \leq & -2y_1 + 7y_2 & \leq 113 \\ 0 \leq & -y_1 - 6y_2 & \leq 6 \\ 0 \leq & y_1 + 7y_2 & \leq 6 \\ & y_1, y_2 \in \mathbb{Z} & \end{array}$$

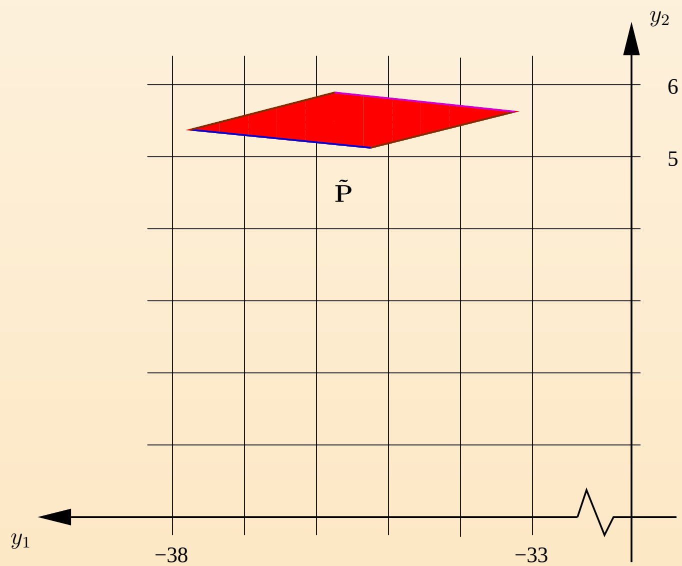


DKP example in 2D

Let $\boldsymbol{p} = (1,\,1),\ M = 20,\ \boldsymbol{r} = (1,\,-1),\ \boldsymbol{u} = (6,\,6)$

$$\begin{array}{rcl} 106 \leq & 21x_1 + 19x_2 & \leq 113 \\ 0 \leq & x_1, x_2 & \leq 6 \\ & x_1, x_2 \in \mathbb{Z} & \end{array} \quad \longrightarrow \quad \begin{array}{rcl} 106 \leq & -2y_1 + 7y_2 & \leq 113 \\ 0 \leq & -y_1 - 6y_2 & \leq 6 \\ 0 \leq & y_1 + 7y_2 & \leq 6 \\ & y_1, y_2 \in \mathbb{Z} & \end{array}$$





DKPs get harder as t grows

DKPs get harder as t grows

Two infeasible knapsack problems: Can you tell which one is harder?

$$\begin{aligned}1473x_1 + 1524x_2 + 1569x_3 + 1570x_4 + 1575x_5 + 1624x_6 + 1625x_7\\ +2160x_8 + 2206x_9 + 2207x_{10} + 2211x_{11} + 2211x_{12} + 2257x_{13}\\ +2260x_{14} + 2305x_{15} + 2843x_{16} + 2943x_{17} + 2947x_{18} + 2991x_{19}\\ +2993x_{20} + 2997x_{21} + 3528x_{22} + 3577x_{23} + 3631x_{24} + 3677x_{25}\\ = 28980,\ x_i \in \{0,1\}\end{aligned}$$

$$\begin{aligned}1314x_1 + 1315x_2 + 1317x_3 + 1318x_4 + 1971x_5 + 1972x_6 + 1973x_7\\ +1976x_8 + 1977x_9 + 1977x_{10} + 2629x_{11} + 2630x_{12} + 2631x_{13}\\ +2631x_{14} + 2633x_{15} + 2634x_{16} + 2635x_{17} + 2635x_{18} + 3287x_{19}\\ +3287x_{20} + 3287x_{21} + 3289x_{22} + 3292x_{23} + 3293x_{24} + 3293x_{25}\\ = 28981,\ x_i \in \{0,1\}\end{aligned}$$

Two hard knapsacks

Two hard knapsacks

using CPLEX 9.0 to prove infeasibility

Two hard knapsacks

using CPLEX 9.0 to prove infeasibility

- second knapsack has $t = 1$, and takes $\approx 22,000$ nodes

Two hard knapsacks

using CPLEX 9.0 to prove infeasibility

- second knapsack has $t = 1$, and takes $\approx 22,000$ nodes
- first knapsack has $t = 2$, and takes ≈ 3.6 million nodes

Questions

Questions

- Can we create and **analyze** classes of $t+1$-DKPs for $t \geq 2$?

Questions

- Can we create and **analyze** classes of $t + 1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?

Questions

- Can we create and **analyze** classes of $t+1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?
- "thin" directions and integer width?

Questions

- Can we create and **analyze** classes of $t+1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?
- "thin" directions and integer width?
- width and integer width:

Questions

- Can we create and **analyze** classes of $t+1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?
- "thin" directions and integer width?
- width and integer width: given polyhedron $\mathcal{K}$, direction $\boldsymbol{p}$

Questions

- Can we create and **analyze** classes of $t + 1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?
- "thin" directions and integer width?
- width and integer width: given polyhedron $\mathcal{K}$, direction $\boldsymbol{p}$

$$\text{width}(\boldsymbol{p}, \mathcal{K}) \;=\; \max\{\boldsymbol{p}\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{K}\} \,-\, \min\{\boldsymbol{p}\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{K}\}$$

Questions

- Can we create and **analyze** classes of $t+1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?
- "thin" directions and integer width?
- width and integer width: given polyhedron $\mathcal{K}$, direction $\boldsymbol{p}$

$$\begin{aligned}\text{width}(\boldsymbol{p},\mathcal{K}) &= \max\{\boldsymbol{p}\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{K}\} - \min\{\boldsymbol{p}\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{K}\} \\ \text{iwidth}(\boldsymbol{p},\mathcal{K}) &= \lfloor \max\{\boldsymbol{p}\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{K}\} \rfloor - \lceil \min\{\boldsymbol{p}\boldsymbol{x} \,|\, \boldsymbol{x} \in \mathcal{K}\} \rceil + 1\end{aligned}$$

Questions

- Can we create and **analyze** classes of $t+1$-DKPs for $t \geq 2$?
- Do they have more interesting structure than when $t = 1$?
- "thin" directions and integer width?
- width and integer width: given polyhedron $\mathcal{K}$, direction $\boldsymbol{p}$

$$\begin{aligned} \text{width}(\boldsymbol{p}, \mathcal{K}) &= \max\{\boldsymbol{px} \,|\, \boldsymbol{x} \in \mathcal{K}\} - \min\{\boldsymbol{px} \,|\, \boldsymbol{x} \in \mathcal{K}\} \\ \text{iwidth}(\boldsymbol{p}, \mathcal{K}) &= \lfloor \max\{\boldsymbol{px} \,|\, \boldsymbol{x} \in \mathcal{K}\} \rfloor - \lceil \min\{\boldsymbol{px} \,|\, \boldsymbol{x} \in \mathcal{K}\} \rceil + 1 \end{aligned}$$

$\text{iwidth}(\boldsymbol{p})$: # branches created by branching on the hyperplane $\boldsymbol{px}$

Cascade Knapsack Problem (CKP)

Cascade Knapsack Problem (CKP)

- instance of 3-DKP $(t = 2,\ \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r})$

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2,\ \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2,\ \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that
  - it is integer infeasible by choice of β_1, β_2;

Cascade Knapsack Problem (CKP)

- instance of 3-DKP $(t = 2,\ \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r})$ with $\boldsymbol{u} = \boldsymbol{e}$ $(x_j \in \{0,1\})$ such that
  - it is integer infeasible by choice of β_1, β_2;
  - $\mathrm{width}(\boldsymbol{e}_j, \mathsf{CKP}) = 1 - 0,\ \ \mathrm{iwidth}(\boldsymbol{e}_j, \mathsf{CKP}) = 2$ for all j;

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2,\ \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that
  - it is integer infeasible by choice of β_1, β_2;
  - $\mathrm{width}(\boldsymbol{e}_j, \mathsf{CKP}) = 1 - 0,\ \ \mathrm{iwidth}(\boldsymbol{e}_j, \mathsf{CKP}) = 2$ for all j;
  - $1 < \mathrm{width}(\boldsymbol{p}_1, \mathsf{CKP}) < 2$ and $\mathrm{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}) = 1$,

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2,\ \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that
  - it is integer infeasible by choice of β_1, β_2;
  - $\text{width}(\boldsymbol{e}_j, \text{CKP}) = 1 - 0,\ \ \text{iwidth}(\boldsymbol{e}_j, \text{CKP}) = 2$ for all j;
  - $1 < \text{width}(\boldsymbol{p}_1, \text{CKP}) < 2$ and $\text{iwidth}(\boldsymbol{p}_1, \text{CKP}) = 1$, so branching on $\boldsymbol{p}_1\boldsymbol{x}$ amounts to just adding $\boldsymbol{p}_1\boldsymbol{x} = k_1$ for some integer k_1;

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2$, $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that
  - it is integer infeasible by choice of β_1, β_2;
  - $\mathrm{width}(\boldsymbol{e}_j, \text{CKP}) = 1 - 0$, $\mathrm{iwidth}(\boldsymbol{e}_j, \text{CKP}) = 2$ for all j;
  - $1 < \mathrm{width}(\boldsymbol{p}_1, \text{CKP}) < 2$ and $\mathrm{iwidth}(\boldsymbol{p}_1, \text{CKP}) = 1$, so branching on $\boldsymbol{p}_1\boldsymbol{x}$ amounts to just adding $\boldsymbol{p}_1\boldsymbol{x} = k_1$ for some integer k_1;
  - $\mathrm{width}(\boldsymbol{p}_2, \text{CKP} \wedge \boldsymbol{p}_1\boldsymbol{x} = k_1) < 1$ and $\mathrm{iwidth}(\boldsymbol{p}_2, \text{CKP} \wedge \boldsymbol{p}_1\boldsymbol{x} = k_1) = 0$.

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2$, $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that
  - it is integer infeasible by choice of β_1, β_2;
  - $\mathrm{width}(\boldsymbol{e}_j, \mathsf{CKP}) = 1 - 0$, $\mathrm{iwidth}(\boldsymbol{e}_j, \mathsf{CKP}) = 2$ for all j;
  - $1 < \mathrm{width}(\boldsymbol{p}_1, \mathsf{CKP}) < 2$ and $\mathrm{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}) = 1$, so branching on $\boldsymbol{p}_1\boldsymbol{x}$ amounts to just adding $\boldsymbol{p}_1\boldsymbol{x} = k_1$ for some integer k_1;
  - $\mathrm{width}(\boldsymbol{p}_2, \mathsf{CKP} \wedge \boldsymbol{p}_1\boldsymbol{x} = k_1) < 1$ and $\mathrm{iwidth}(\boldsymbol{p}_2, \mathsf{CKP} \wedge \boldsymbol{p}_1\boldsymbol{x} = k_1) = 0$.
- branching on $\boldsymbol{p}_1\boldsymbol{x}$ and then on $\boldsymbol{p}_2\boldsymbol{x}$ kills the problem

Cascade Knapsack Problem (CKP)

- instance of 3-DKP ($t = 2$, $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$) with $\boldsymbol{u} = \boldsymbol{e}$ ($x_j \in \{0, 1\}$) such that
  - it is integer infeasible by choice of β_1, β_2;
  - $\text{width}(\boldsymbol{e}_j, \text{CKP}) = 1 - 0$, $\text{iwidth}(\boldsymbol{e}_j, \text{CKP}) = 2$ for all j;
  - $1 < \text{width}(\boldsymbol{p}_1, \text{CKP}) < 2$ and $\text{iwidth}(\boldsymbol{p}_1, \text{CKP}) = 1$, so branching on $\boldsymbol{p}_1 \boldsymbol{x}$ amounts to just adding $\boldsymbol{p}_1 \boldsymbol{x} = k_1$ for some integer k_1;
  - $\text{width}(\boldsymbol{p}_2, \text{CKP} \wedge \boldsymbol{p}_1 \boldsymbol{x} = k_1) < 1$ and $\text{iwidth}(\boldsymbol{p}_2, \text{CKP} \wedge \boldsymbol{p}_1 \boldsymbol{x} = k_1) = 0$.

- branching on $\boldsymbol{p}_1 \boldsymbol{x}$ and then on $\boldsymbol{p}_2 \boldsymbol{x}$ kills the problem

- effect of branching on $\boldsymbol{p}_1 \boldsymbol{x}$ *cascades* to the next level $\boldsymbol{p}_2 \boldsymbol{x}$

Example 1: CKP_1

Example 1: CKP_1

$$\begin{aligned}4196 \;\le\; & 340\,x_1 + 452\,x_2 + 695\,x_3 + 926\,x_4 + 1050\,x_5 \\ & + 1089\,x_6 + 1190\,x_7 + 1296\,x_8 + 1342\,x_9 \;\le\; 4197 \\ & x_j \in \{0, 1\}\end{aligned}$$

Example 1: CKP_1

$$\begin{aligned} 4196 \;\leq\; & 340\,x_1 + 452\,x_2 + 695\,x_3 + 926\,x_4 + 1050\,x_5 \\ & + 1089\,x_6 + 1190\,x_7 + 1296\,x_8 + 1342\,x_9 \;\leq\; 4197 \\ & x_j \in \{0, 1\} \end{aligned}$$

- CPLEX 11.0 takes 64 B&B nodes

Example 1: CKP_1

$$
\begin{aligned}
4196 \;\le\; & 340\,x_1 + 452\,x_2 + 695\,x_3 + 926\,x_4 + 1050\,x_5 \\
& + 1089\,x_6 + 1190\,x_7 + 1296\,x_8 + 1342\,x_9 \;\le\; 4197 \\
& x_j \in \{0, 1\}
\end{aligned}
$$

- CPLEX 11.0 takes 64 B&B nodes
- $\mathrm{width}(\boldsymbol{e}_j, \mathsf{CKP}_1) = 1,\ \ \mathrm{iwidth}(\boldsymbol{e}_j, \mathsf{CKP}_1) = 2$ for all j

Example 1: CKP_1

$$\begin{aligned} 4196 \;\le\; & 340\,x_1 + 452\,x_2 + 695\,x_3 + 926\,x_4 + 1050\,x_5 \\ & + 1089\,x_6 + 1190\,x_7 + 1296\,x_8 + 1342\,x_9 \;\le\; 4197 \\ & x_j \in \{0,1\} \end{aligned}$$

- CPLEX 11.0 takes 64 B&B nodes
- $\mathrm{width}(\boldsymbol{e}_j, \mathsf{CKP}_1) = 1, \;\; \mathrm{iwidth}(\boldsymbol{e}_j, \mathsf{CKP}_1) = 2$ for all j
- $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$, with $M_1 = 127, \;\; M_2 = 12,$

Example 1: CKP_1

$$
\begin{aligned}
4196 \;\leq\; & 340\,x_1 + 452\,x_2 + 695\,x_3 + 926\,x_4 + 1050\,x_5 \\
& + 1089\,x_6 + 1190\,x_7 + 1296\,x_8 + 1342\,x_9 \;\leq\; 4197 \\
& x_j \in \{0,1\}
\end{aligned}
$$

- CPLEX 11.0 takes 64 B&B nodes
- $\mathrm{width}(\boldsymbol{e}_j, \mathsf{CKP}_1) = 1, \;\; \mathrm{iwidth}(\boldsymbol{e}_j, \mathsf{CKP}_1) = 2$ for all j
- $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \boldsymbol{p}_2 M_2 + \boldsymbol{r}$, with $M_1 = 127, \; M_2 = 12$,

$$
\begin{aligned}
\boldsymbol{p}_1 &= (2,\; 3, 5, 7,\; 8, 8,\; 9, 10, 10), \\
\boldsymbol{p}_2 &= (7,\; 6, 5, 3,\; 3, 6,\; 4,\; 2,\; 6), \quad \text{and} \\
\boldsymbol{r} &= (2, -1, 0, 1, -2, 1, -1, 2,\; 0)
\end{aligned}
$$

Example 1: CKP_1 – Properties

Example 1: $\mathbf{CKP}_1$ – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$

Example 1: $\mathbf{CKP}_1$ – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$
  $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865,$

Example 1: $\mathbf{CKP}_1$ – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$
  $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \text{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$
  $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \text{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\,\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\,\mathsf{CKP}_1\} = 30.102;$
  $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \text{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;
  - CPLEX 11.0 takes 37 B&B nodes for $\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31$

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$
  $\mathrm{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \mathrm{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;
  - CPLEX 11.0 takes 37 B&B nodes for $\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31$
  - $\max\{\boldsymbol{p}_2\boldsymbol{x}\,|\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.989,$
    $\min\{\boldsymbol{p}_2\boldsymbol{x}\,|\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.083;$

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$
  $\mathrm{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \mathrm{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;
  - CPLEX 11.0 takes 37 B&B nodes for $\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31$
  - $\max\{\boldsymbol{p}_2\boldsymbol{x}\,|\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.989,$
    $\min\{\boldsymbol{p}_2\boldsymbol{x}\,|\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.083;$
    $\mathrm{width}(\boldsymbol{p}_2, \mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31) = 0.906$, $\mathrm{iwidth} = 0$

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_1\} = 30.102;$
  $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \text{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;
  - CPLEX 11.0 takes 37 B&B nodes for $\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31$
  - $\max\{\boldsymbol{p}_2\boldsymbol{x}\,|\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.989,$
    $\min\{\boldsymbol{p}_2\boldsymbol{x}\,|\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.083;$
    $\text{width}(\boldsymbol{p}_2, \mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31) = 0.906$, $\text{iwidth} = 0$

- comparable DKP: $\boldsymbol{a} = \boldsymbol{p}_1 M + \boldsymbol{r}$, with $M = 136$; for $x_j \in \{0, 1\}$

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x} \,|\text{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x} \,|\text{CKP}_1\} = 30.102;$
  $\text{width}(\boldsymbol{p}_1, \text{CKP}_1) = 1.865, \quad \text{iwidth}(\boldsymbol{p}_1, \text{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;
  - CPLEX 11.0 takes 37 B&B nodes for $\text{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31$
  - $\max\{\boldsymbol{p}_2\boldsymbol{x} \,|\text{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.989,$
    $\min\{\boldsymbol{p}_2\boldsymbol{x} \,|\text{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.083;$
    $\text{width}(\boldsymbol{p}_2, \text{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31) = 0.906$, $\text{iwidth} = 0$

- comparable DKP: $\boldsymbol{a} = \boldsymbol{p}_1 M + \boldsymbol{r}$, with $M = 136$; for $x_j \in \{0, 1\}$

$$\begin{aligned} 4223 \;\le\; & 274\,x_1 + 407\,x_2 + 680\,x_3 + 953\,x_4 + 1086\,x_5 + \\ & 1089\,x_6 + 1223\,x_7 + 1362\,x_8 + 1360\,x_9 \;\le\; 4224 \end{aligned}$$

Example 1: CKP_1 – Properties

- $\max\{\boldsymbol{p}_1\boldsymbol{x} \,|\, \mathsf{CKP}_1\} = 31.967, \quad \min\{\boldsymbol{p}_1\boldsymbol{x} \,|\, \mathsf{CKP}_1\} = 30.102;$ $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1.865, \quad \text{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 1;$

- $\boldsymbol{p}_1\boldsymbol{x} = 31$ is the only branch;
  - CPLEX 11.0 takes 37 B&B nodes for $\mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31$
  - $\max\{\boldsymbol{p}_2\boldsymbol{x} \,|\, \mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.989,$
    $\min\{\boldsymbol{p}_2\boldsymbol{x} \,|\, \mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31\} = 21.083;$
    $\text{width}(\boldsymbol{p}_2, \mathsf{CKP}_1 \wedge \boldsymbol{p}_1\boldsymbol{x} = 31) = 0.906$, $\text{iwidth} = 0$

- comparable DKP: $\boldsymbol{a} = \boldsymbol{p}_1 M + \boldsymbol{r}$, with $M = 136$; for $x_j \in \{0, 1\}$

$$
\begin{aligned}
4223 \;\le\; & 274\,x_1 + 407\,x_2 + 680\,x_3 + 953\,x_4 + 1086\,x_5 + \\
& \quad 1089\,x_6 + 1223\,x_7 + 1362\,x_8 + 1360\,x_9 \;\le\; 4224
\end{aligned}
$$

CPLEX 11.0 takes 44 B&B nodes

width v/s integer width

width v/s integer width

- for CKP_1, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$

width v/s integer width

- for CKP_1, $\text{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\text{width}(\boldsymbol{e}_j) = 1$
- but $\text{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\text{iwidth}(\boldsymbol{e}_j) = 2$

width v/s integer width

- for CKP_1, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$
- but $\mathrm{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\mathrm{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth

width v/s integer width

- for CKP$_1$, $\text{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\text{width}(\boldsymbol{e}_j) = 1$
- but $\text{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\text{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth

width v/s integer width

- for CKP_1, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$
- but $\mathrm{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\mathrm{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth
  - *non-trivial* to identify $\boldsymbol{p}_1$

width v/s integer width

- for CKP_1, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$
- but $\mathrm{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\mathrm{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth
  - *non-trivial* to identify $\boldsymbol{p}_1$ (RSRef)

width v/s integer width

- for CKP$_1$, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$
- but $\mathrm{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\mathrm{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth
  - *non-trivial* to identify $\boldsymbol{p}_1$ (RSRef)
- Cook and Kannan (personal communication)

width v/s integer width

- for CKP_1, $\text{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\text{width}(\boldsymbol{e}_j) = 1$
- but $\text{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\text{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth
  - *non-trivial* to identify $\boldsymbol{p}_1$ (RSRef)
- Cook and Kannan (personal communication) studied cases when $\text{width} = 1.9$ (say) and $\text{iwidth} = 1$

width v/s integer width

- for CKP_1, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$
- but $\mathrm{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\mathrm{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth
  - *non-trivial* to identify $\boldsymbol{p}_1$ (RSRef)
- Cook and Kannan (personal communication) studied cases when $\mathrm{width} = 1.9$ (say) and $\mathrm{iwidth} = 1$
- We create variation of CKP with $\mathrm{width}(\boldsymbol{p}_1) > 1$ and $\mathrm{iwidth}(\boldsymbol{p}_1) = \mathbf{2}$;

width v/s integer width

- for CKP$_1$, $\mathrm{width}(\boldsymbol{p}_1) = 1.865$, bigger than $\mathrm{width}(\boldsymbol{e}_j) = 1$
- but $\mathrm{iwidth}(\boldsymbol{p}_1) = 1$, which is smaller than $\mathrm{iwidth}(\boldsymbol{e}_j) = 2$
  - width is not a good *predictor* of iwidth
  - preferable to branch on $\boldsymbol{p}_1$ according to iwidth
  - *non-trivial* to identify $\boldsymbol{p}_1$ (RSRef)
- Cook and Kannan (personal communication) studied cases when $\mathrm{width} = 1.9$ (say) and $\mathrm{iwidth} = 1$
- We create variation of CKP with $\mathrm{width}(\boldsymbol{p}_1) > 1$ and $\mathrm{iwidth}(\boldsymbol{p}_1) = \mathbf{2}$; for both branches of $\boldsymbol{p}_1\boldsymbol{x}$, branching on $\boldsymbol{p}_2\boldsymbol{x}$ proves infeasibility

Example 2: CKP_2

Example 2: CKP_2

$$\begin{aligned} 4399 \leq \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \leq 4400 \\ & x_j \in \{0, 1\} \end{aligned}$$

Example 2: CKP_2

$$\begin{aligned} 4399 \le \quad & 344\, x_1 + 458\, x_2 + 705\, x_3 + 940\, x_4 + 1066\, x_5 \\ & + 1105\, x_6 + 1208\, x_7 + 1316\, x_8 + 1362\, x_9 \quad \le 4400 \\ & x_j \in \{0, 1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129,\ M_2 = 12$

Example 2: CKP_2

$$\begin{aligned} 4399 \leq \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \leq 4400 \\ & x_j \in \{0,1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129,\ M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes

Example 2: CKP_2

$$\begin{aligned} 4399 \le \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \le 4400 \\ & x_j \;\in\; \{0,1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129,\; M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\text{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\text{CKP}_2\} = 31.165;$

Example 2: CKP_2

$$\begin{aligned} 4399 \le \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \le 4400 \\ & x_j \;\in\; \{0,1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129, \; M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathrm{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathrm{CKP}_2\} = 31.165;$ $\mathrm{width}(\boldsymbol{p}_1, \mathrm{CKP}_2) = 1.867,$

Example 2: CKP_2

$$
\begin{aligned}
4399 \le \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\
& \quad + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \le 4400 \\
& x_j \ \in \ \{0,1\}
\end{aligned}
$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129,\ M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x} \,|\, \text{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x} \,|\, \text{CKP}_2\} = 31.165;$
    $\text{width}(\boldsymbol{p}_1, \text{CKP}_2) = 1.867, \quad \text{iwidth}(\boldsymbol{p}_1, \text{CKP}_1) = 2;$

Example 2: CKP_2

$$\begin{aligned} 4399 \le \;& 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \le 4400 \\ & x_j \;\in\; \{0,1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129,\; M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\text{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\text{CKP}_2\} = 31.165;$ $\text{width}(\boldsymbol{p}_1, \text{CKP}_2) = 1.867, \quad \text{iwidth}(\boldsymbol{p}_1, \text{CKP}_1) = 2;$
  - $\text{width}(\boldsymbol{p}_2, \text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32) = 0.895,\; \text{iwidth} = 0$

Example 2: CKP$_2$

$$\begin{aligned} 4399 \le \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \le 4400 \\ & x_j \ \in \ \{0, 1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP$_1$, but $M_1 = 129,\ M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x} \,|\, \text{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x} \,|\, \text{CKP}_2\} = 31.165;$
    $\text{width}(\boldsymbol{p}_1, \text{CKP}_2) = 1.867, \quad \text{iwidth}(\boldsymbol{p}_1, \text{CKP}_1) = 2;$
  - $\text{width}(\boldsymbol{p}_2, \text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32) = 0.895$, $\text{iwidth} = 0$
    CPLEX 11.0 takes 35 B&B nodes for CKP$_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32$

Example 2: CKP_2

$$\begin{aligned} 4399 \leq \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \leq 4400 \\ & x_j \;\in\; \{0,1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129,\ M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\mathsf{CKP}_2\} = 31.165;$
    $\text{width}(\boldsymbol{p}_1, \mathsf{CKP}_2) = 1.867, \quad \text{iwidth}(\boldsymbol{p}_1, \mathsf{CKP}_1) = 2;$
  - $\text{width}(\boldsymbol{p}_2, \mathsf{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32) = 0.895$, $\text{iwidth} = 0$
    CPLEX 11.0 takes 35 B&B nodes for $\mathsf{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32$
  - $\text{width}(\boldsymbol{p}_2, \mathsf{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 33) = 0.158$, $\text{iwidth} = 0$

Example 2: CKP_2

$$\begin{aligned} 4399 \leq \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\ & + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \leq 4400 \\ & x_j \ \in \ \{0,1\} \end{aligned}$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129, \ M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x}\,|\text{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x}\,|\text{CKP}_2\} = 31.165;$
    $\text{width}(\boldsymbol{p}_1, \text{CKP}_2) = 1.867, \quad \text{iwidth}(\boldsymbol{p}_1, \text{CKP}_1) = 2;$
  - $\text{width}(\boldsymbol{p}_2, \text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32) = 0.895$, $\text{iwidth} = 0$
    CPLEX 11.0 takes 35 B&B nodes for $\text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32$
  - $\text{width}(\boldsymbol{p}_2, \text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 33) = 0.158$, $\text{iwidth} = 0$
    CPLEX 11.0 solves $\text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32$ at the root node

Example 2: CKP_2

$$
\begin{aligned}
4399 \leq \quad & 344\,x_1 + 458\,x_2 + 705\,x_3 + 940\,x_4 + 1066\,x_5 \\
& + 1105\,x_6 + 1208\,x_7 + 1316\,x_8 + 1362\,x_9 \quad \leq 4400 \\
& x_j \; \in \; \{0, 1\}
\end{aligned}
$$

- same $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{r}$ as in CKP_1, but $M_1 = 129, \; M_2 = 12$
  - CPLEX 11.0 takes 95 B&B nodes
  - $\max\{\boldsymbol{p}_1\boldsymbol{x} \,|\text{CKP}_2\} = 33.032, \quad \min\{\boldsymbol{p}_1\boldsymbol{x} \,|\text{CKP}_2\} = 31.165;$
    $\text{width}(\boldsymbol{p}_1, \text{CKP}_2) = 1.867, \quad \text{iwidth}(\boldsymbol{p}_1, \text{CKP}_1) = 2;$
  - $\text{width}(\boldsymbol{p}_2, \text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32) = 0.895$, $\text{iwidth} = 0$
    CPLEX 11.0 takes 35 B&B nodes for $\text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32$
  - $\text{width}(\boldsymbol{p}_2, \text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 33) = 0.158$, $\text{iwidth} = 0$
    CPLEX 11.0 solves $\text{CKP}_2 \wedge \boldsymbol{p}_1\boldsymbol{x} = 32$ at the root node
- $\boldsymbol{p}_1$ is *not* preferable to $\boldsymbol{e}_j$ for branching, based on iwidth alone

CKP Generalizations

CKP Generalizations

- we can generalize CKPs:

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \mathsf{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \mathsf{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3$; $\boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$
- denoted as $t+1$-CKPs;

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$

- denoted as $t+1$-CKPs; Recipes to generate $t+1$-CKPs

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$

- denoted as $t+1$-CKPs; Recipes to generate $t+1$-CKPs

- computationally hard

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$
- denoted as $t+1$-CKPs; Recipes to generate $t+1$-CKPs
- computationally hard
  - with $x_j \in \{0, 1\}$, we get *small* a_j's, but CPLEX still struggles

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \mathrm{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\mathrm{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\mathrm{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$

- denoted as $t+1$-CKPs; Recipes to generate $t+1$-CKPs

- computationally hard
  - with $x_j \in \{0, 1\}$, we get *small* a_j's, but CPLEX still struggles
  - e.g., 4-CKP with $n = 30$, $a_{\max} \leq 9000$, CPLEX 9.0 takes ≈ 57 million B&B nodes

CKP Generalizations

- we can generalize CKPs:
  - to higher t's ($t \geq 3; \boldsymbol{a} = \boldsymbol{p}_1 M_1 + \cdots + \boldsymbol{p}_t M_t + \boldsymbol{r}$)
    $1 < \text{width}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) < 2$
    $\text{iwidth}(\boldsymbol{p}_i \,|\, \text{CKP} \wedge \boldsymbol{p}_j \boldsymbol{x} = k_j,\ j = 1, \ldots, i-1) = 1$ or 2
    for $i = 1, \ldots, t-1$, and then $\text{iwidth}(\boldsymbol{p}_t) = 0$
  - $\boldsymbol{u}$ can be more general than $\boldsymbol{e}$

- denoted as $t+1$-CKPs; Recipes to generate $t+1$-CKPs

- computationally hard
  - with $x_j \in \{0, 1\}$, we get *small* a_j's, but CPLEX still struggles
  - e.g., 4-CKP with $n = 30$, $a_{\max} \leq 9000$, CPLEX 9.0 takes ≈ 57 million B&B nodes
  - dynamic programming could be effective (time $= O(n\beta_1)$)?

Computation: 4-CKPs, $n = 30$, $\boldsymbol{u} = \boldsymbol{e}$

Computation: 4-CKPs, $n = 30$, $\boldsymbol{u} = \boldsymbol{e}$

	CKP widths			CKP		CKP$_{-\boldsymbol{p}_1}$		CKP$_{-\boldsymbol{p}_1\boldsymbol{p}_2}$		DKP		RS
#	w_1	w_{21}	w_{312}	BB	TM	BB	TM	BB	TM	BB	TM	BB
1	1.55	1.42	0.92	58,057,939	**u**	2,448,625	126.0	205,814	13.3	11756	0.4	3
2	1.47	1.44	0.90	56,937,604	3484	740,556	41.0	66189	4.6	8708	0.3	1
3	1.57	1.50	0.94	46,187,956	3027	2,005,687	99.4	249,232	14.1	9537	0.3	5
4	1.50	1.53	0.89	55,782,856	**u**	477,707	25.2	252,505	13.7	6496	0.3	4
5	1.49	1.48	0.94	56,313,840	**u**	1,421,719	69.0	334,046	19.0	5527	0.2	3
6	1.50	1.55	0.90	55,597,050	**u**	1,319,626	73.0	257,922	15.0	10520	0.4	15
7	1.50	1.59	0.91	60,453,028	**u**	1,595,424	78.6	151,812	9.1	7336	0.3	6
8	1.57	1.52	0.95	64,409,733	**u**	5,324,924	278.3	310,768	19.2	10360	0.4	6
9	1.50	1.48	0.96	55,491,175	**u**	3,366,436	167.2	312,653	18.0	10061	0.4	5
10	1.49	1.53	0.92	60,307,524	**u**	3,107,323	158.2	443,789	25.6	8227	0.3	68

BB: # B&B nodes, TM: CPU time (sec), **u**: unsolved in **1 hour** time limit, typical instance: $a_{\min} \approx 4000$, $a_{\max} \approx 9000$, β_1, $\beta_2 \approx 65000$; RS: RSRef

Used CPLEX 9.0; instances available at www.wsu.edu/∼kbala

Computation: 4-CKPs, $n = 30$, $\boldsymbol{u} = \boldsymbol{e}$

	CKP widths			CKP		CKP_$\boldsymbol{p}_1$		CKP_$\boldsymbol{p}_1\boldsymbol{p}_2$		DKP		RS
#	w_1	w_{21}	w_{312}	BB	TM	BB	TM	BB	TM	BB	TM	BB
1	1.55	1.42	0.92	58,057,939	**u**	2,448,625	126.0	205,814	13.3	11756	0.4	3
2	1.47	1.44	0.90	56,937,604	3484	740,556	41.0	66189	4.6	8708	0.3	1
3	1.57	1.50	0.94	46,187,956	3027	2,005,687	99.4	249,232	14.1	9537	0.3	5
4	1.50	1.53	0.89	55,782,856	**u**	477,707	25.2	252,505	13.7	6496	0.3	4
5	1.49	1.48	0.94	56,313,840	**u**	1,421,719	69.0	334,046	19.0	5527	0.2	3
6	1.50	1.55	0.90	55,597,050	**u**	1,319,626	73.0	257,922	15.0	10520	0.4	15
7	1.50	1.59	0.91	60,453,028	**u**	1,595,424	78.6	151,812	9.1	7336	0.3	6
8	1.57	1.52	0.95	64,409,733	**u**	5,324,924	278.3	310,768	19.2	10360	0.4	6
9	1.50	1.48	0.96	55,491,175	**u**	3,366,436	167.2	312,653	18.0	10061	0.4	5
10	1.49	1.53	0.92	60,307,524	**u**	3,107,323	158.2	443,789	25.6	8227	0.3	68

BB: # B&B nodes, TM: CPU time (sec), **u**: unsolved in **1 hour** time limit, typical instance: $a_{\min} \approx 4000$, $a_{\max} \approx 9000$, $\beta_1, \beta_2 \approx 65000$; RS: RSRef

Used CPLEX 9.0; instances available at www.wsu.edu/∼kbala

Summary

Summary

- CKPs are classes of $t+1$-level decomposable knapsacks

Summary

- CKPs are classes of $t+1$-level decomposable knapsacks
  - which are hard for ordinary B&B

Summary

- CKPs are classes of $t+1$-level decomposable knapsacks
  - which are hard for ordinary B&B
  - have a sequence of "good" branching directions $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_t$

Summary

- CKPs are classes of $t+1$-level decomposable knapsacks
  - which are hard for ordinary B&B
  - have a sequence of “good” branching directions $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_t$
  - $\mathrm{iwidth}(\boldsymbol{p}_i) = 1$ or 2 in the branching sequence for $i < t$

Summary

- CKPs are classes of $t+1$-level decomposable knapsacks
  - which are hard for ordinary B&B
  - have a sequence of "good" branching directions $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_t$
  - $\mathrm{iwidth}(\boldsymbol{p}_i) = 1$ or 2 in the branching sequence for $i < t$
- when M_i's are big enough, RSRef solves in at most t or 2^t nodes, respectively

Summary

- CKPs are classes of $t+1$-level decomposable knapsacks
  - which are hard for ordinary B&B
  - have a sequence of "good" branching directions $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_t$
  - $\mathrm{iwidth}(\boldsymbol{p}_i) = 1$ or 2 in the branching sequence for $i < t$
- when M_i's are big enough, RSRef solves in at most t or 2^t nodes, respectively
- **both** width and iwidth can be **poor** indicators of "good" branching directions

Slides