Aalto University
School of Science

Introducing Real Variables and Integer Objective Functions to Answer Set Programming

Guohua Liu, Tomi Janhunen, and Ilkka Niemelä

Helsinki Institute for Information Technology HIIT
Department of Information and Computer Science
Aalto University

INAP, Kiel, Germany, September 11, 2013

Background

- Answer set programming (ASP) features a *rule-based* syntax subject to *answer-set semantics*.

$$
\begin{tikzcd}
\boxed{\text{Problem}} \arrow[r, "\textit{Solve}"] \arrow[d, "\textit{Formalize}"'] & \boxed{\text{Solution(s)}} \\
\boxed{\text{Set of rules}} \arrow[r, "\textit{Ground \& Search}"] & \boxed{\text{Answer set(s)}} \arrow[u, "\textit{Extract}"']
\end{tikzcd}
$$

Background

- Answer set programming (ASP) features a *rule-based* syntax subject to *answer-set semantics*.

$$\begin{tikzcd} \boxed{\text{Problem}} \arrow[rr, "\textit{Solve}"] \arrow[d, "\textit{Formalize}"'] & & \boxed{\text{Solution(s)}} \\ \boxed{\text{Set of rules}} \arrow[rr, "\textit{Ground \& Search}"] & & \boxed{\text{Answer set(s)}} \arrow[u, "\textit{Extract}"'] \end{tikzcd}$$

- One viable way to implement ASP is to *translate* programs into SAT and its extensions in the SMT framework:
  - — Pure SAT [Janhunen, ECAI, 2004]
  - — Difference logic [Niemelä, AMAI, 2008]
  - — Bit-vector logic [Nguyen et al., INAP, 2011]
  - — Mixed integer programming [Liu et al., KR, 2012]

ASP and Linear Constraints

- It is also possible to enrich the language of ASP using linear constraints as *additional primitives*:
  - Integrating ASP and CLP [Mellarkod et al., AMAI, 2008]
  - Constraint ASP [Gebser et al., ICLP, 2009]
  - ASP(LC) programs [Liu et al., KR, 2012]

$$e(I) - s(I) \geq D \leftarrow \mathrm{task}(I, E, D).$$

ASP and Linear Constraints

- It is also possible to enrich the language of ASP using linear constraints as *additional primitives*:
  - Integrating ASP and CLP [Mellarkod et al., AMAI, 2008]
  - Constraint ASP [Gebser et al., ICLP, 2009]
  - ASP(LC) programs [Liu et al., KR, 2012]

$$e(I) - s(I) \geq D \leftarrow \mathrm{task}(I, E, D).$$

- The answer sets of an ASP(LC) program can be computed in three steps:
  1. All rules are translated into linear constraints.
  2. The resulting linear program is solved using a MIP solver.
  3. Answer sets are recovered from the solutions found (if any).

ASP and Linear Constraints

- It is also possible to enrich the language of ASP using linear constraints as *additional primitives*:
  - Integrating ASP and CLP [Mellarkod et al., AMAI, 2008]
  - Constraint ASP [Gebser et al., ICLP, 2009]
  - ASP(LC) programs [Liu et al., KR, 2012]

$$e(I) - s(I) \geq D \leftarrow \mathrm{task}(I, E, D).$$

- The answer sets of an ASP(LC) program can be computed in three steps:
  1. All rules are translated into linear constraints.
  2. The resulting linear program is solved using a MIP solver.
  3. Answer sets are recovered from the solutions found (if any).
- A proof of concept implementation is available under `http://research.ics.aalto.fi/software/asp/mingo/`

Objectives

- The goal of this paper is to study how real variables could be incorporated into ASP(LC) programs.
- The introduction of real-valued variables is non-trivial:
  1. *Strict constraints* over reals are not fully supported by contemporary MIP solvers.
  2. In the strict setting, the existence of *optimal solutions* is not guaranteed even if all variables are bounded.
  3. Numerical instability may result due to coefficients used.

Objectives

- The goal of this paper is to study how real variables could be incorporated into ASP(LC) programs.
- The introduction of real-valued variables is non-trivial:
  1. *Strict constraints* over reals are not fully supported by contemporary MIP solvers.
  2. In the strict setting, the existence of *optimal solutions* is not guaranteed even if all variables are bounded.
  3. Numerical instability may result due to coefficients used.

Example

The *completion* of $\{a \leftarrow x \leq 1\}$ is effectively $a \leftrightarrow x \leq 1$:

- Then $\neg a$ implies $x > 1$.
- What if $f(x) = x$ is additionally minimized?

Outline

1. Preliminaries
2. Extension with Real Variables
3. Extension with Objective Functions
4. Comparison of ASP(LC) and MIP
5. Experiments
6. Conclusions and Future Work

1. PRELIMINARIES

- A *linear constraint* is an expression of the form

  $$\sum_{i=1}^{n} u_i x_i \sim k$$

  where
  - the u_i's and k are real numbers,
  - the x_i's are variables ranging over real numbers, and
  - the operator $\sim$ is one of $\leq$, $\geq$, $<$, and $>$.

1. PRELIMINARIES

- A *linear constraint* is an expression of the form

  $$\sum_{i=1}^{n} u_i x_i \sim k$$

  where
  - the u_i's and k are real numbers,
  - the x_i's are variables ranging over real numbers, and
  - the operator $\sim$ is one of $\leq$, $\geq$, $<$, and $>$.

- A *mixed integer program* (a *MIP program*) has the form

  $$\begin{array}{rl} \texttt{minimize/maximize} & \sum_{i=1}^{n} u_i x_i \\ \texttt{subject to} & C_1, \ldots, C_m. \end{array}$$

  where the C_i's are linear constraints.

ASP(LC) Programs

- An *ASP(LC) program* has extended normal rules of form
$$a \leftarrow b_1, \ldots, b_n, \text{not } c_1, \ldots, \text{not } c_m, t_1, \ldots, t_l$$
where each *theory atom* t_i is a linear constraint.

ASP(LC) Programs

- An *ASP(LC) program* has extended normal rules of form

$$a \leftarrow b_1, \ldots, b_n, \text{not } c_1, \ldots, \text{not } c_m, t_1, \ldots, t_l$$

  where each *theory atom* t_i is a linear constraint.

- The *reduct* of an ASP(LC) program P with respect to an *interpretation* $\langle M, T\rangle$ such that $T \cup \bar{T}$ is satisfiable:

$$P^{\langle M,T\rangle} = \{\mathrm{H}(r) \leftarrow \mathrm{B}^+(r) \mid r \in P, \\ \mathrm{H}(r) \neq \emptyset,\ \mathrm{B}^-(r) \cap M = \emptyset,\ \text{and } \mathrm{B}^t(r) \subseteq T\}.$$

ASP(LC) Programs

- An *ASP(LC) program* has extended normal rules of form

$$a \leftarrow b_1, \ldots, b_n, \text{not } c_1, \ldots, \text{not } c_m, t_1, \ldots, t_l$$

where each *theory atom* t_i is a linear constraint.

- The *reduct* of an ASP(LC) program P with respect to an *interpretation* $\langle M, T\rangle$ such that $T \cup \bar{T}$ is satisfiable:

$$P^{\langle M,T\rangle} = \{\mathrm{H}(r) \leftarrow \mathrm{B}^+(r) \mid r \in P, \\ \mathrm{H}(r) \neq \emptyset,\ \mathrm{B}^-(r) \cap M = \emptyset,\ \text{and } \mathrm{B}^t(r) \subseteq T\}.$$

- Given an ASP(LC) program P, an *answer set* $\langle M, T\rangle$ satisfies P such that M is a $\subseteq$-minimal model of $P^{\langle M,T\rangle}$.

Example

Consider the following ASP(LC) program P:

$$a \leftarrow x - y \leq 2. \qquad b \leftarrow x - y \geq 5. \qquad \leftarrow x - y \geq 0.$$

Example

Consider the following ASP(LC) program P:

$$a \leftarrow x - y \leq 2. \qquad b \leftarrow x - y \geq 5. \qquad \leftarrow x - y \geq 0.$$

1. $I_1 = \langle \{a\}, \{x - y \leq 2\} \rangle \in AS(P)$ since
   - $\{(x - y \leq 2), \neg(x - y \geq 5), \neg(x - y \geq 0)\}$ is satisfiable,
   - $I_1 \models P$, and
   - $\{a\}$ is the minimal model of $P^{I_1} = \{a \leftarrow .\}$.

Example

Consider the following ASP(LC) program P:

$$a \leftarrow x - y \leq 2. \qquad b \leftarrow x - y \geq 5. \qquad \leftarrow x - y \geq 0.$$

1. $I_1 = \langle \{a\}, \{x - y \leq 2\} \rangle \in AS(P)$ since
   - $\{(x - y \leq 2), \neg(x - y \geq 5), \neg(x - y \geq 0)\}$ is satisfiable,
   - $I_1 \models P$, and
   - $\{a\}$ is the minimal model of $P^{I_1} = \{a \leftarrow .\}$.

2. $I_2 = \langle \{b\}, \{x - y \geq 5\} \rangle \notin AS(P)$ since

$$\{(x - y \geq 5), \neg(x - y \leq 2), \neg(x - y \geq 0)\} \models \bot.$$

Example

Consider the following ASP(LC) program P:

$$a \leftarrow x - y \leq 2. \qquad b \leftarrow x - y \geq 5. \qquad \leftarrow x - y \geq 0.$$

1. $I_1 = \langle \{a\}, \{x - y \leq 2\} \rangle \in AS(P)$ since
   - $\{(x - y \leq 2), \neg(x - y \geq 5), \neg(x - y \geq 0)\}$ is satisfiable,
   - $I_1 \models P$, and
   - $\{a\}$ is the minimal model of $P^{I_1} = \{a \leftarrow .\}$.
2. $I_2 = \langle \{b\}, \{x - y \geq 5\} \rangle \notin AS(P)$ since

   $$\{(x - y \geq 5), \neg(x - y \leq 2), \neg(x - y \geq 0)\} \models \bot.$$
3. $I_3 = \langle \emptyset, \{x - y \geq 0\} \rangle \notin AS(P)$ since $I_3 \not\models P$.

MIP Translation

- For simplicity, let us consider simple rules of form $a \leftarrow t$.
- Consider a *definition* of the form $a \leftarrow t_1, \ldots, a \leftarrow t_n$.

MIP Translation

- For simplicity, let us consider simple rules of form $a \leftarrow t$.
- Consider a *definition* of the form $a \leftarrow t_1, \ldots, a \leftarrow t_n$.
- The MIP translation of this definition consists of:
  1. For each $i \in \{1, \ldots, n\}$, *indicator constraints*

     $$d_i = 1 \rightarrow t_i \qquad d_i = 0 \rightarrow \neg t_i$$

     where $d_1, \ldots, d_n$ act as *names* for $t_1, \ldots, t_n$.

MIP Translation

- For simplicity, let us consider simple rules of form $a \leftarrow t$.
- Consider a *definition* of the form $a \leftarrow t_1, \ldots, a \leftarrow t_n$.
- The MIP translation of this definition consists of:
  1. For each $i \in \{1, \ldots, n\}$, *indicator constraints*
     $$d_i = 1 \rightarrow t_i \qquad d_i = 0 \rightarrow \neg t_i$$
     where $d_1, \ldots, d_n$ act as *names* for $t_1, \ldots, t_n$.
  2. To satisfy the definition, linear constraints
     $$a - d_1 \geq 0, \quad \ldots, \quad a - d_k \geq 0.$$

MIP Translation

- For simplicity, let us consider simple rules of form $a \leftarrow t$.
- Consider a *definition* of the form $a \leftarrow t_1, \ldots, a \leftarrow t_n$.
- The MIP translation of this definition consists of:
  1. For each $i \in \{1, \ldots, n\}$, *indicator constraints*
     $$d_i = 1 \rightarrow t_i \qquad d_i = 0 \rightarrow \neg t_i$$
     where $d_1, \ldots, d_n$ act as *names* for $t_1, \ldots, t_n$.
  2. To satisfy the definition, linear constraints
     $$a - d_1 \geq 0, \quad \ldots, \quad a - d_k \geq 0.$$
  3. For the *completion* of the definition, linear constraint
     $$d_1 + \ldots + d_k - a \geq 0.$$

Correspondence

Let ν be an assignment for the *MIP translation* $\tau(P)$ of P.

- The ν-*induced interpretation* of P is $I_P^\nu = \langle M, T \rangle$ where

$$\begin{aligned} M &= \{a \mid a \in \mathcal{A}(P),\ \nu(a) = 1\} \text{ and} \\ T &= \{t \mid t \in \mathcal{T}(P),\ \nu \models t\}. \end{aligned}$$

- The correspondence theorem from [Liu et al., KR, 2012]:
  1. If ν is a solution of $\tau(P)$, then $I_P^\nu \in AS(P)$.
  2. If $I \in AS(P)$, there is a solution ν of $\tau(P)$ such that $I = I_P^\nu$.

Correspondence

Let ν be an assignment for the *MIP translation* $\tau(P)$ of P.

- The ν-*induced interpretation* of P is $I_P^\nu = \langle M, T\rangle$ where

$$\begin{aligned} M &= \{a \mid a \in \mathcal{A}(P),\ \nu(a) = 1\} \text{ and} \\ T &= \{t \mid t \in \mathcal{T}(P),\ \nu \models t\}. \end{aligned}$$

- The correspondence theorem from [Liu et al., KR, 2012]:
  1. If ν is a solution of $\tau(P)$, then $I_P^\nu \in AS(P)$.
  2. If $I \in AS(P)$, there is a solution ν of $\tau(P)$ such that $I = I_P^\nu$.

Example

$$\begin{array}{lll} d_1 = 1 \rightarrow x - y \leq 2, & d_1 = 0 \rightarrow x - y > 2, & a - d_1 = 0, \\ d_2 = 1 \rightarrow x - y \geq 5, & d_2 = 0 \rightarrow x - y < 5, & b - d_2 = 0, \\ d_3 = 1 \rightarrow x - y \geq 0, & d_3 = 0 \rightarrow x - y < 0, & d_3 = 0. \end{array}$$

Correspondence

Let ν be an assignment for the *MIP translation* $\tau(P)$ of P.

- The ν-*induced interpretation* of P is $I_P^\nu = \langle M, T\rangle$ where

$$\begin{aligned} M &= \{a \mid a \in \mathcal{A}(P),\ \nu(a) = 1\} \text{ and} \\ T &= \{t \mid t \in \mathcal{T}(P),\ \nu \models t\}. \end{aligned}$$

- The correspondence theorem from [Liu et al., KR, 2012]:
  1. If ν is a solution of $\tau(P)$, then $I_P^\nu \in AS(P)$.
  2. If $I \in AS(P)$, there is a solution ν of $\tau(P)$ such that $I = I_P^\nu$.

Example

$$\begin{array}{lll} d_1 = 1 \rightarrow x - y \leq 2, & d_1 = 0 \rightarrow x - y > 2, & a - d_1 = 0, \\ d_2 = 1 \rightarrow x - y \geq 5, & d_2 = 0 \rightarrow x - y < 5, & b - d_2 = 0, \\ d_3 = 1 \rightarrow x - y \geq 0, & d_3 = 0 \rightarrow x - y < 0, & d_3 = 0. \end{array}$$

$\nu(a) = 1$, $\nu(b) = 0$, $\nu \models x - y \leq 2 \implies \langle\{a\}, \{x - y \leq 2\}\rangle \in AS(P)$.

2. EXTENSION WITH REAL VARIABLES

- The MIP translation of ASP(LC) programs brings about *strict constraints* involving integer and/or real variables.

2. EXTENSION WITH REAL VARIABLES

- The MIP translation of ASP(LC) programs brings about *strict constraints* involving integer and/or real variables.
- It is possible to *isolate* strict relationships: for
  1. a set Γ of non-strict constraints,
  2. a set $S = \{x_1 > 0, \ldots, x_n > 0\}$ of strict constraints, and
  3. a new variable δ:

  $\Gamma \cup S$ is satisfiable iff for any bound $b > 0$,

  $$\Gamma \cup S_\delta \cup \{0 < \delta \leq b\}$$

  with $S_\delta = \{x_1 \geq \delta, \ldots, x_n \geq \delta\}$ is satisfiable.

2. EXTENSION WITH REAL VARIABLES

- The MIP translation of ASP(LC) programs brings about *strict constraints* involving integer and/or real variables.
- It is possible to *isolate* strict relationships: for
  1. a set Γ of non-strict constraints,
  2. a set $S = \{x_1 > 0, \ldots, x_n > 0\}$ of strict constraints, and
  3. a new variable δ:

  $\Gamma \cup S$ is satisfiable iff for any bound $b > 0$,

  $$\Gamma \cup S_\delta \cup \{0 < \delta \leq b\}$$

  with $S_\delta = \{x_1 \geq \delta, \ldots, x_n \geq \delta\}$ is satisfiable.
- The result is a generalization of a lemma by Dutertre and de Moura [CAV, 2006] based on rationals (no bound b).

Non-Strict Translation of Programs

- The set S can be replaced by *strict indicator constraints*.
- The idea is to establish $\delta > 0$ indirectly by *maximizing* δ subject to $\Gamma \cup S_\delta \cup \{0 \leq \delta \leq b\}$.
- All strict relationships occurring in the MIP translation $\tau(P)$ can be isolated in indicator constraints.

Non-Strict Translation of Programs

- The set S can be replaced by *strict indicator constraints*.
- The idea is to establish $\delta > 0$ indirectly by *maximizing* δ subject to $\Gamma \cup S_\delta \cup \{0 \leq \delta \leq b\}$.
- All strict relationships occurring in the MIP translation $\tau(P)$ can be isolated in indicator constraints.

Theorem

Let P be an ASP(LC) program that may involve real variables, δ a new variable, and $b > 0$ a bound.

1. *If ν is a solution of $\tau(P)^b_\delta$ such that $\nu(\delta) > 0$, then $I^\nu_P \in AS(P)$.*
2. *If $I \in AS(P)$, then there is a solution ν of $\tau(P)^b_\delta$ such that $I = I^\nu_P$ and $\nu(\delta) > 0$.*

3. EXTENSION WITH OBJECTIVE FUNCTIONS

- Typical ASP languages support objective functions of form

$$\texttt{\#minimize/maximize}\ [a_1 = w_{a_1}, \ldots, a_m = w_{a_m},\\ \text{not } b_1 = w_{b_1}, \ldots, \text{not } b_n = w_{b_n}].$$

  where a_i's and b_i's are *Booleans* with *integer weights*.

3. EXTENSION WITH OBJECTIVE FUNCTIONS

- Typical ASP languages support objective functions of form

$$\texttt{\#minimize/maximize}\ [a_1 = w_{a_1}, \ldots, a_m = w_{a_m},$$
$$\text{not } b_1 = w_{b_1}, \ldots, \text{not } b_n = w_{b_n}].$$

  where a_i's and b_i's are *Booleans* with *integer weights*.
- The MIP translation $\tau(P)$ can be conjoined with any integer objective function if P is free from optimization statements.

3. EXTENSION WITH OBJECTIVE FUNCTIONS

- Typical ASP languages support objective functions of form

$$\texttt{\#minimize/maximize}\ [a_1 = w_{a_1}, \ldots, a_m = w_{a_m},\ \text{not}\ b_1 = w_{b_1}, \ldots, \text{not}\ b_n = w_{b_n}].$$

  where a_i's and b_i's are *Booleans* with *integer weights*.
- The MIP translation $\tau(P)$ can be conjoined with any integer objective function if P is free from optimization statements.

Example

Consider an integer variable x in the following setting:

$\texttt{minimize}\ x. \quad x \geq 1. \quad x \leq n.$

Identifying Optimal Answer Sets

Let P be an ASP(LC) program with an objective function f.

- An answer set $\langle M, T \rangle \in AS(P)$ is *optimal* iff there is a solution of $T \cup \bar{T}$ that gives the optimal value to f among the set of valuations

$$\{\nu \mid \nu \models T \cup \bar{T} \text{ for some } \langle M, T \rangle \in AS(P)\}.$$

Identifying Optimal Answer Sets

Let P be an ASP(LC) program with an objective function f.

- An answer set $\langle M, T\rangle \in AS(P)$ is *optimal* iff there is a solution of $T \cup \bar{T}$ that gives the optimal value to f among the set of valuations
$$\{\nu \mid \nu \models T \cup \bar{T} \text{ for some } \langle M, T\rangle \in AS(P)\}.$$
- The following result can be established for ASP(LC) programs involving *integer variables* only.

Theorem

An answer set $\langle M, T\rangle \in AS(P)$ is optimal iff there is a solution $\nu \models \tau(P)$ so that $I_P^\nu = \langle M, T\rangle$ and ν gives the optimal value to f.

4. COMPARISON OF ASP(LC) AND MIP

- The aim of this part is to study modeling capabilities of ASP(LC) and pure MIP languages using two problems:
  1. Hamiltonian Routing Problem (HRP)
  2. Job Shop Problem (JSP)

4. COMPARISON OF ASP(LC) AND MIP

- The aim of this part is to study modeling capabilities of ASP(LC) and pure MIP languages using two problems:
  1. Hamiltonian Routing Problem (HRP)
  2. Job Shop Problem (JSP)

- The ASP(LC) language can express non-trivial logical relationships leading to more intuitive/compact encodings.

4. COMPARISON OF ASP(LC) AND MIP

- The aim of this part is to study modeling capabilities of ASP(LC) and pure MIP languages using two problems:
  1. Hamiltonian Routing Problem (HRP)
  2. Job Shop Problem (JSP)
- The ASP(LC) language can express non-trivial logical relationships leading to more intuitive/compact encodings.
- Further observations on the relationship:
  1. It is difficult to write/maintain $\tau(P)$ directly as part of a MIP program.
  2. Any MIP program can be viewed as an ASP(LC) program.

An ASP(LC) Encoding of HRP

$$\mathrm{hc}(X, Y) \leftarrow \mathrm{e}(X, Y, D), \text{ not } \mathrm{nhc}(X, Y).$$
$$\mathrm{nhc}(X, Y) \leftarrow \mathrm{e}(X, Y, D_1), \mathrm{e}(X, Z, D_2), \mathrm{hc}(X, Z), \ Y \neq Z.$$
$$\mathrm{nhc}(X, Y) \leftarrow \mathrm{e}(X, Y, D_1), \mathrm{e}(Z, Y, D_2), \mathrm{hc}(Z, Y), \ X \neq Z.$$

An ASP(LC) Encoding of HRP

$$\mathrm{hc}(X, Y) \leftarrow \mathrm{e}(X, Y, D), \text{ not } \mathrm{nhc}(X, Y).$$
$$\mathrm{nhc}(X, Y) \leftarrow \mathrm{e}(X, Y, D_1), \mathrm{e}(X, Z, D_2), \mathrm{hc}(X, Z), \; Y \neq Z.$$
$$\mathrm{nhc}(X, Y) \leftarrow \mathrm{e}(X, Y, D_1), \mathrm{e}(Z, Y, D_2), \mathrm{hc}(Z, Y), \; X \neq Z.$$

$$\mathrm{initial}(1).$$
$$\mathrm{reach}(X) \leftarrow \mathrm{reach}(Y), \mathrm{hc}(Y, X), \text{ not initial}(Y), \mathrm{e}(Y, X, D).$$
$$\mathrm{reach}(X) \leftarrow \mathrm{hc}(Y, X), \mathrm{initial}(Y), \mathrm{e}(Y, X, D).$$
$$\leftarrow \mathrm{v}(X), \text{ not reach}(X).$$

An ASP(LC) Encoding of HRP

$$\mathrm{hc}(X, Y) \leftarrow \mathrm{e}(X, Y, D), \text{ not } \mathrm{nhc}(X, Y).$$
$$\mathrm{nhc}(X, Y) \leftarrow \mathrm{e}(X, Y, D_1), \mathrm{e}(X, Z, D_2), \mathrm{hc}(X, Z), \ Y \neq Z.$$
$$\mathrm{nhc}(X, Y) \leftarrow \mathrm{e}(X, Y, D_1), \mathrm{e}(Z, Y, D_2), \mathrm{hc}(Z, Y), \ X \neq Z.$$

$$\mathrm{initial}(1).$$
$$\mathrm{reach}(X) \leftarrow \mathrm{reach}(Y), \mathrm{hc}(Y, X), \text{ not initial}(Y), \mathrm{e}(Y, X, D).$$
$$\mathrm{reach}(X) \leftarrow \mathrm{hc}(Y, X), \mathrm{initial}(Y), \mathrm{e}(Y, X, D).$$
$$\leftarrow \mathrm{v}(X), \text{ not reach}(X).$$

$$t(1) = 0.$$
$$t(X) - t(Y) = D \leftarrow \mathrm{hc}(Y, X), \mathrm{e}(Y, X, D), \ X \neq 1.$$
$$t(X) \leq T \leftarrow \mathrm{critical}(X, T).$$

A MIP Encoding of HRP

$$\begin{array}{ll}
\sum_{(i,j,d)\in E} x_{ij} = 1 & i \in V \\
\sum_{(j,i,d)\in E} x_{ji} = 1 & i \in V \\
1 \leq p_i \leq n & i \in V
\end{array}$$

A MIP Encoding of HRP

$$
\begin{array}{ll}
\sum_{(i,j,d)\in E} x_{ij} = 1 & i \in V \\
\sum_{(j,i,d)\in E} x_{ji} = 1 & i \in V \\
1 \leq p_i \leq n & i \in V \\
 & \\
p_i \neq p_j & i \in V,\ j \in V,\ i \neq j \\
p_j \neq p_i + 1 & (i,j,d) \notin E,\ i \neq j \\
(p_i = n) \rightarrow (p_j \geq 2) & (i,j,d) \notin E,\ i \neq j
\end{array}
$$

A MIP Encoding of HRP

$$
\begin{array}{ll}
\sum_{(i,j,d)\in E} x_{ij} = 1 & i \in V \\
\sum_{(j,i,d)\in E} x_{ji} = 1 & i \in V \\
1 \le p_i \le n & i \in V \\
\\
p_i \neq p_j & i \in V,\ j \in V,\ i \neq j \\
p_j \neq p_i + 1 & (i,j,d) \notin E,\ i \neq j \\
(p_i = n) \rightarrow (p_j \geq 2) & (i,j,d) \notin E,\ i \neq j \\
\\
r_1 = 0 & \\
x_{ij} = 1 \rightarrow r_j - r_i = d & (i,j,d) \in E \\
r_i \le t & (i,t) \in CV
\end{array}
$$

5. EXPERIMENTS

- We modified the MINGO system to enable real-valued variables to the extent possible.
- We used several benchmark problems that involve *reachability* conditions and *disjunctive* information:
  1. HRP and JSP
  2. Newpaper, Routing Max, and Routing Min Problems [Liu et al., KR, 2012]
  3. Disjunctive Scheduling Problem [2nd ASP-COMP, 2009]

5. EXPERIMENTS

- We modified the MINGO system to enable real-valued variables to the extent possible.
- We used several benchmark problems that involve *reachability* conditions and *disjunctive* information:
  1. HRP and JSP
  2. Newpaper, Routing Max, and Routing Min Problems [Liu et al., KR, 2012]
  3. Disjunctive Scheduling Problem [2nd ASP-COMP, 2009]
- Memory and time were limited to 4GB and 600s.

5. EXPERIMENTS

- We modified the MINGO system to enable real-valued variables to the extent possible.
- We used several benchmark problems that involve *reachability* conditions and *disjunctive* information:
  1. HRP and JSP
  2. Newpaper, Routing Max, and Routing Min Problems [Liu et al., KR, 2012]
  3. Disjunctive Scheduling Problem [2nd ASP-COMP, 2009]
- Memory and time were limited to 4GB and 600s.
- The *bound* $b = 10^{-6}$ was determined experimentally.
- Due to maximizing δ subject to $0 \leq \delta \leq b$, real variables could be incorporated into decision problems only.

Results: Hamiltonian Routing

Density	Decision		Optimization	
	MINGO^r	CPLEX	MINGO^r	CPLEX
10	0.03	0.01	0.07	0.01
20	0.05	0.01	0.12	0.01
30	0.92	NA	50.81	NA
40	41.62	NA	NA	NA
50	13.94	NA	NA	NA
60	64.91	NA	NA	NA
70	35.78	NA	NA	NA
80	8.02	95.40	NA	NA
90	181.33	24.74	NA	NA
100	146.18	13.88	NA	NA

Results: Job Shop Scheduling

Tasks	Decision		Optimization	
	MINGO^r	CPLEX	MINGO^r	CPLEX
10	0.42	0.14	0.35	0.08
20	4.04	0.18	1.56	0.14
30	6.78	0.40	4.69	0.49
40	13.74	0.72	12.18	1.62
50	27.37	1.36	16.15	1.16
60	45.44	1.72	30.82	2.01
70	51.56	1.57	47.85	1.80
80	88.72	2.34	68.99	2.83
90	114.32	2.97	79.28	6.43
100	192.09	4.19	112.09	8.05

6. CONCLUSIONS AND FUTURE WORK

- In this research, we alternatively extend ASP by
  - linear constraints involving real variables, or
  - integer objective functions.
- The proposed extensions of ASP facilitate modeling.
- For some benchmarks, also computational advantage can be obtained in the ASP(LC) approach.
- The ASP and MIP paradigms can benefit from integration.

6. CONCLUSIONS AND FUTURE WORK

- In this research, we alternatively extend ASP by
  - linear constraints involving real variables, or
  - integer objective functions.
- The proposed extensions of ASP facilitate modeling.
- For some benchmarks, also computational advantage can be obtained in the ASP(LC) approach.
- The ASP and MIP paradigms can benefit from integration.

Future work:

- There are ways to improve the current translation and its computational performance:
  - reducing the number of extra variables needed, and
  - stopping optimization early (as soon as $\delta > 0$).
- Our prototype lacks a user-friendly front-end parser (special predicates are used to express linear constraints).