

Principles of AI Planning

12. Planning as search: potential heuristics

Albert-Ludwigs-Universität Freiburg



UNI
FREIBURG

Bernhard Nebel and Robert Mattmüller

December 18th, 2018



Motivation

Potential
Heuristics

Summary

Motivation

Previous chapters:

“Procedural” method for obtaining a heuristic

Solve an easier version of the problem.

We have studied two common simplification methods:
relaxation and **abstraction**.

This chapter:

“Declarative” method for obtaining a heuristic

- **Declaratively** describe the information we want the heuristic estimator to exploit.
- Let a computer find a heuristic that fits the declarative description.

Motivation

Potential
Heuristics

Summary

Example (potential heuristic in chess)

Evaluation function for chess position s
(from White's perspective; the higher, the better):

$$h(s) = 9 \cdot (\text{♔} - \text{♚}) + 5 \cdot (\text{♖} - \text{♜}) + \\ 3 \cdot (\text{♘} - \text{♙}) + 3 \cdot (\text{♞} - \text{♟}) + 1 \cdot (\text{♙} - \text{♟})$$

where $\text{♔}, \text{♚}, \text{♖}, \text{♜}, \dots$ is the number of white and black queens, rooks, etc. still on the board.

Question: Can we derive a similar heuristic for planning?

Answer: Yes! (Even declaratively!)



Potential Heuristics

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Potential heuristics: idea

Heuristic design as an optimization problem:

- Define simple numerical **state features** f_1, \dots, f_n .
- Consider heuristics that are **linear combinations** of features:

$$h(s) = w_1 f_1(s) + \dots + w_n f_n(s)$$

with weights (**potentials**) $w_i \in \mathbb{R}$.

- Find potentials for which h is admissible and well-informed.

Motivation:

- **declarative approach** to heuristic design
- heuristic **very fast to compute** if features are

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Definition (feature)

A (state) **feature** for a planning task is a numerical function defined on the states of the task: $f : S \rightarrow \mathbb{R}$.

Atomic features test if some atom is true in a state.

Definition (atomic feature)

Let $v = d$ be an atom of an FDR planning task.
Then the **atomic feature** $f_{v=d}$ is defined as:

$$f_{v=d}(s) = \begin{cases} 1 & \text{if } s \models v = d \\ 0 & \text{otherwise} \end{cases}$$

\rightsquigarrow atomic features \approx facts

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Definition (potential heuristic)

A **potential heuristic** for a set of features $\mathcal{F} = \{f_1, \dots, f_n\}$ is a heuristic function h defined as a **linear combination** of the features:

$$h(s) = w_1 f_1(s) + \dots + w_n f_n(s)$$

with weights (**potentials**) $w_i \in \mathbb{R}$.

- We only consider **atomic** potential heuristics, which are based on the set of all atomic features.
- **Example** for a task with state variables v_1 and v_2 and $\mathcal{D}_{v_1} = \mathcal{D}_{v_2} = \{d_1, d_2, d_3\}$:

$$h(s) = 3f_{v_1=d_1} + 1/2f_{v_1=d_2} - 2f_{v_1=d_3} + 5/2f_{v_2=d_1}$$

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

How to set the weights?



Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

We want to find **good** atomic potential heuristics:

- admissible
- consistent
- well-informed

Question: How to achieve this?

Answer: Linear programming.

Goal: solve a system of linear inequalities over n real-valued variables while optimizing some linear objective function.

Example (Production domain)

Two sorts of items with time requirements and profit per item.

	Cutting	Assembly	Postproc.	Profit per item
(x) sort 1	25	60	68	30
(y) sort 2	75	60	34	40
per day	≤ 450	≤ 480	≤ 476	maximize!

Aim: Find numbers of pieces x of sort 1 and y of sort 2 produced per day such that resource constraints are met and objective function is maximized.

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., formalization)

$$\text{maximize } z = 30x + 40y \text{ subject to:} \quad (1)$$

$$x \geq 0, y \geq 0 \quad (2)$$

$$25x + 75y \leq 450 \quad (3)$$

$$60x + 60y \leq 480 \quad (4)$$

$$68x + 34y \leq 476 \quad (5)$$

- Line (1): Objective function
- Inequalities (2)–(5): Admissible solutions

Motivation

Potential
Heuristics

General Idea

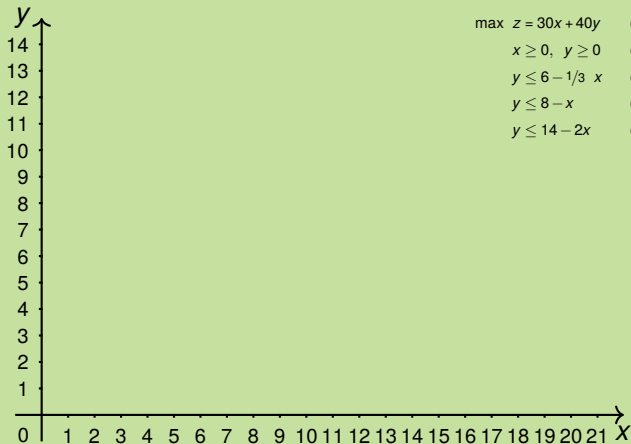
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

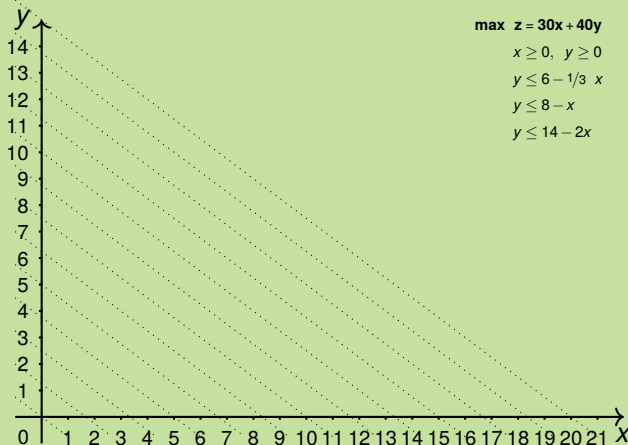
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

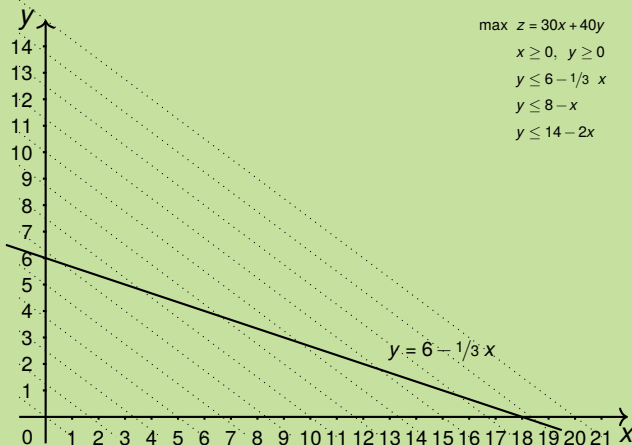
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



$$\max z = 30x + 40y \quad (1)$$

$$x \geq 0, y \geq 0 \quad (2)$$

$$y \leq 6 - \frac{1}{3}x \quad (3)$$

$$y \leq 8 - x \quad (4)$$

$$y \leq 14 - 2x \quad (5)$$

Motivation

Potential
Heuristics

General Idea

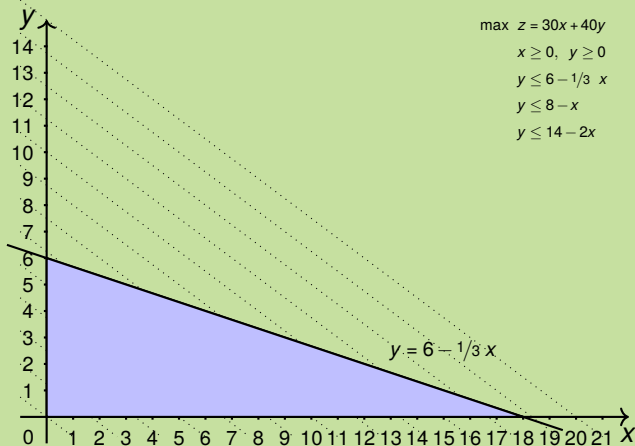
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

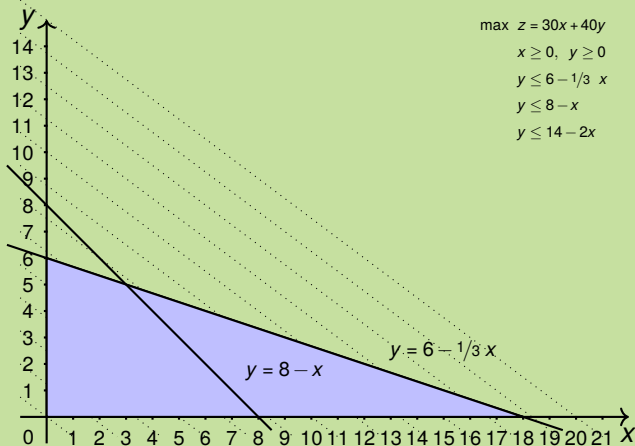
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

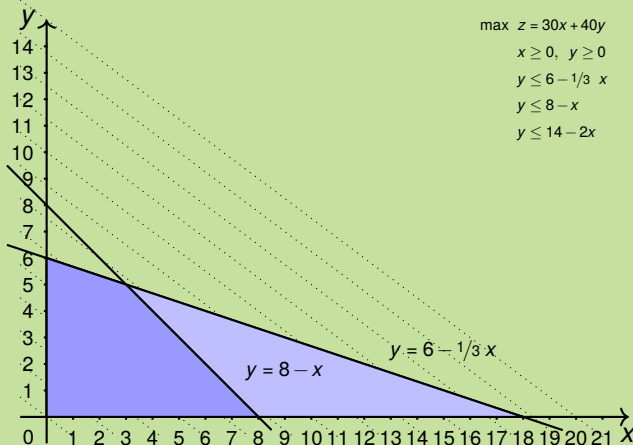
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

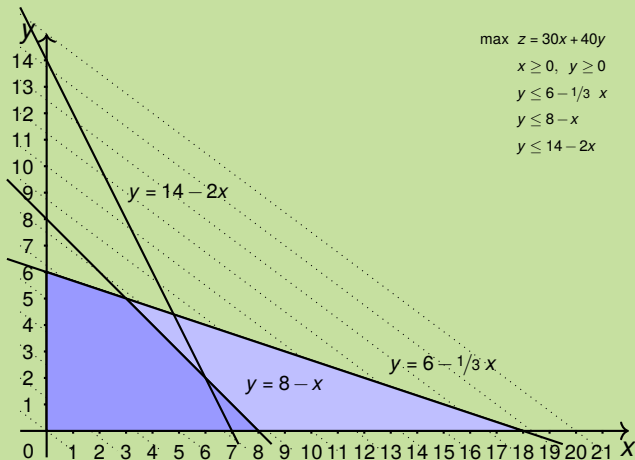
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

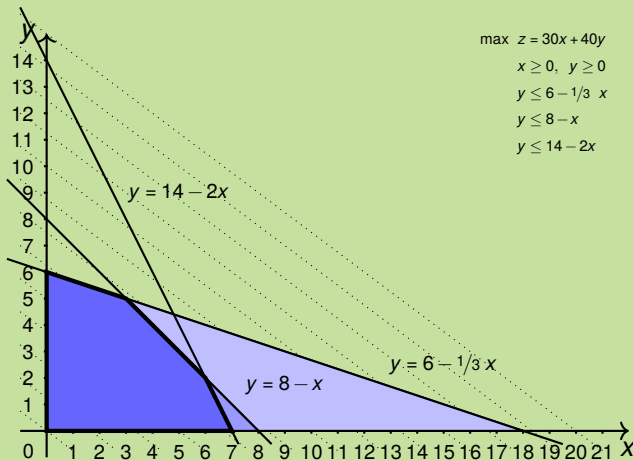
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

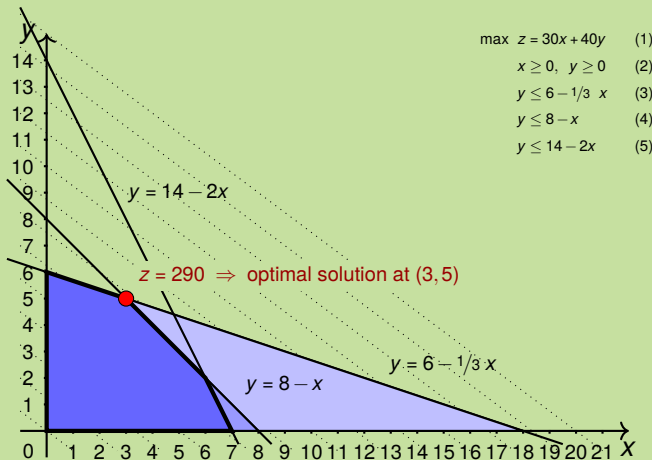
Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Example (ctd., visualization)



Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Definition (Linear program)

A **linear program** (LP) over variables x_1, \dots, x_n consists of

- **m linear constraints** of the form

$$\sum_{i=1}^n a_{ji} x_i \leq b_j$$

with $a_{ji} \in \mathbb{R}$ for all $j = 1, \dots, m$ and $i = 1, \dots, n$, and

- a **linear objective function** to be maximized ($x_i \geq 0$):

$$\sum_{i=1}^n c_i x_i$$

with $c_i \in \mathbb{R}$ for all $i = 1, \dots, n$.

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Solution of an LP:

assignment of values to the x_i **satisfying the constraints** and **maximizing the objective function**.

Solution algorithms:

- Usually: **simplex algorithm** (worst-case exponential).
- There are also polynomial-time algorithms.

Standard description of LP-based derivation of potentials assumes **transition normal form**.

Assumption (for the rest of the chapter): only SAS⁺ tasks.

Notation: variables occurring in conditions and effects.

Definition ($vars(\varphi)$, $vars(e)$)

For a logical formula φ over finite-domain variables \mathcal{V} , $vars(\varphi)$ denotes the set of finite-domain variables occurring in φ .

For an effect e over finite-domain variables \mathcal{V} , $vars(e)$ denotes the set of finite-domain variables occurring in e .

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Definition (transition normal form)

An SAS⁺ planning task $\Pi = \langle \mathcal{V}, I, O, \gamma \rangle$ is in **transition normal form (TNF)** if

- for all $o \in O$, $\text{vars}(\text{pre}(o)) = \text{vars}(\text{eff}(o))$, and
- $\text{vars}(\gamma) = \mathcal{V}$.

In words, an **operator** in TNF must mention the same variables in the precondition and effect, and a **goal** in TNF must mention all variables (= specify exactly one goal state).

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

There are two ways in which an operator o can violate TNF:

- There exists a variable $v \in \text{vars}(\text{pre}(o)) \setminus \text{vars}(\text{eff}(o))$.
- There exists a variable $v \in \text{vars}(\text{eff}(o)) \setminus \text{vars}(\text{pre}(o))$.

The **first case** is easy to address: if $v = d$ is a precondition with no effect on v , just add the effect $v := d$.

The **second case** is more difficult: if we have the effect $v := d$ but no precondition on v , how can we add a precondition on v without changing the meaning of the operator (and without introducing exponentially many new operators)?

- 1 For every variable v , add a new **auxiliary value** u to its domain.
- 2 For every variable v and value $d \in \mathcal{D}_v \setminus \{u\}$, add a new operator to change the value of v from d to u at no cost: $\langle v = d, v := u \rangle$.
- 3 For all operators o and all variables $v \in \text{vars}(\text{eff}(o)) \setminus \text{vars}(\text{pre}(o))$, add the precondition $v = u$ to $\text{pre}(o)$.

Properties:

- Transformation can be computed in linear time.
- Due to the auxiliary values, there are new states and transitions in the induced transition system, but all **path costs** between **original states** remain the same.

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary



- The auxiliary value idea can also be used to convert the goal γ to TNF.
- For every variable $v \notin \text{vars}(\gamma)$, add the condition $v = u$ to γ .

With these ideas, every SAS⁺ planning task can be converted into transition normal form in linear time.

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary

Assume that $\Pi = \langle \mathcal{V}, I, O, \gamma \rangle$ is in TNF.

Definition (producers and consumers)

Fact $v = d$ is **produced** by operator $o \in O$
if $v = d$ is an **effect** of o , but **not a precondition** of o .

Fact $v = d$ is **consumed** by operator $o \in O$
if $v = d$ is a **precondition** of o , but **not an effect** of o .

Constraints on potentials **characterize** (= are necessary and sufficient for) admissible and consistent atomic potential heuristics:

Goal-awareness constraint

$$\sum_{\text{goal atoms } a} w_a = 0$$

Consistency constraints (for all operators $o \in O$)

$$\sum_{a \text{ consumed by } o} w_a - \sum_{a \text{ produced by } o} w_a \leq \text{cost}(o)$$

Remarks:

- all linear constraints \rightsquigarrow LP
- goal-aware and consistent \rightsquigarrow admissible and consistent

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary



How to find a **well-informed** potential heuristic?

↪ encode **quality metric** in the **objective function** and use LP solver to find a heuristic maximizing it

Examples:

- maximize **heuristic value of a given state** (e.g., initial state)
- maximize average heuristic value of **all states** (including unreachable ones)
- maximize average heuristic value of some **sample states**

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary



- Further constraints can be added to the LP to obtain stronger heuristics.
- The hard work is done by the LP solver.

Motivation

Potential
Heuristics

General Idea

Digression I: Linear
Programming

Digression II:
Transition Normal
Form

Definition and
Properties

Summary



Motivation

Potential
Heuristics

Summary

Summary



- **Declarative** method for obtaining a heuristic
- **Potential heuristics** are **linear combinations** of **features**.
- **Needed: features** and **weights (potentials)**
- **Features:** facts (for us; can be generalized)
- **Potentials:** computed by solving an LP, given constraints that encode goal-awareness and consistency, and an objective function to maximize heuristic value.
- **Necessary prerequisite:** without loss of generality, task is in transition normal form (same variables in preconditions and effects, all variables mentioned in the goal).

Motivation

Potential
Heuristics

Summary



Motivation

Potential
Heuristics

Summary

Slides heavily based on those by Gabriele Röger and Thomas Keller (Uni Basel).