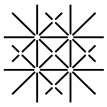

University of Basel

# Mutex Based Potential Heuristics

Bachelor thesis

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Artificial Intelligence
https://ai.dmi.unibas.ch

Examiner: Prof. Dr. Malte Helmert
Supervisor: Salomé Eriksson

Salome Müller
salo.mueller@unibas.ch
2017-063-058

06. 11. 2020

# Table of Contents

# 1

# Strengthening Potential Heuristics

Fišer et al. propose a method to improve potential heuristics with mutexes and disambiguations. This chapter contains the changes which are required to do so, regarding the transformation of a planning task into TNF and the adaption of the optimization functions. It shows how the equations which were later implemented (Chapter **??**) are derived.

## 1.1 Potential Heuristics

When Pommerening et al. first introduced potential heuristics, they showed that two inequalities are sufficient to proof admissibility.

**Theorem 1.** *Let* $\Pi = \langle \mathcal{V}, \mathcal{O}, I, G \rangle$ *denote a planning task,* $\mathtt{P}$ *a potential function, and for every operator* $o \in \mathcal{O}$*, let* $\text{pre}^*(o) = \{\langle V, \text{pre}(o)[V]\rangle | V \in \text{vars}(\text{pre}(o)) \cap \text{vars}(\text{eff}(o))\}$ *and* $\text{vars}^*(o) = \text{vars}(\text{eff}(o)) \setminus \text{vars}(\text{pre}(o))$*. If*

$$\sum_{f \in G} \mathtt{P}(f) + \sum_{V \in \mathcal{V} \setminus \text{vars}(G)} \max_{f \in \mathcal{F}_V} \mathtt{P}(f) \leq 0 \tag{1.1}$$

*and for every operator* $o \in \mathcal{O}$ *it holds that*

$$sum_{f \in \text{pre}^*(o)} \mathtt{P}(f) + \sum_{V \in \text{vars}^*(o)} \max_{f \in \mathcal{F}_V} \mathtt{P}(f) - \sum_{f \in \text{eff}(o)} \mathtt{P}(f) \leq c(o) \tag{1.2}$$

*then the potential heuristic for* $\mathtt{P}$ *is admissible.*

Eq. (1.1) of the theorem by Fišer et al. assures goal-awareness of the potential heuristic. As all variables are assigned in the goal state, the potential of one fact per variable has to be summed up. For the variables $v \in \text{vars}(G)$ we can simply use the potentials of their respective facts. Meanwhile we assume the worst case for the other variables, by using the maximal potential over their facts, as we do not know what fact they are assigned.
Eq. (1.2) assures consistency. Recall the general heuristics consistency equation $h(s) \leq h(o[\![s]\!]) + \text{c}(o)$. It can be rewritten as $h(s) - h(o[\![s]\!]) \leq \text{c}(o)$. As the facts which do not occur in the effect are the same in both $s$ and $o[\![s]\!]$ we can leave them aside. For $s$ we know what facts of the variables of the preconditions are assigned and sum the potentials of those which are also in the effect. For the variables which are in the effect but not in the precondition

we proceed similarly to (1.1), as we do not know their values. The potentials of the facts in the effect can be used without modification for $o[\![s]\!]$.
The advantage of these equations is that they are not state-dependent, even though they do not tell us explicitly what the potentials should be. However, they can be used as the constraints for a linear program, the solution of which is a potential function that forms an admissible potential heuristic. More about this in Section 1.2.

### 1.1.1 Generalization with Mutexes

Mutexes can be used to reduce the domain of variables, which are not yet assigned in a partial state $p$. This property is very helpful, as it minimizes the amount of facts which are candidates for the not assigned variables in Equations (1.1) and (1.2) of Theorem 1.

**Algorithm 1** Multi-fact fixpoint disambiguation.

**Input:** A planning task $\Pi$ with variables $\mathcal{V}$ and facts $F$, a partial state $p$, and a mutex-set $\mathcal{M}$.
**Output:** A set of disambiguations $\mathcal{D}$ of all variables $\mathcal{V}$ for $p$.

```
1:  D_v ← F_V for every V ∈ V
2:  A ← M_p
3:  change ← True
4:  while change do
5:      change ← False
6:      for all V ∈ V do
7:          if D_V \ A ≠ D_V then
8:              D_V ← D_V \ A
9:              A ← A ∪ ⋂_{f∈D_V} M_{p∪{f}}
10:             change ← True
11:         end if
12:     end for
13: end while
14: D ← {D_V | V ∈ V}
```

At the beginning, the set $D_V$ contains all possible values for the variable $V \in \mathcal{V}$, while $A$ contains all facts which are a mutex with any fact in $p$. In each iteration of the while-loop, all $f = \langle v, V \rangle$ which are in $A$ and in $D_V$ are removed from the corresponding $D_V$. On line 9, $A$ is extended with all facts that form a mutex with all facts remaining in $D_V$, i.e., which are a mutex with $p \cup \{f\}$ for all $f \in D_V$.
In conclusion, after applying mutli-fact fixpoint disambiguation $p$ can be extended with any fact in $\mathcal{D}$ without reaching a dead-end state. If any $D_V \in \mathcal{D}$ is empty, then $p$ is already a dead-end itself.
This algorithm is used for several applications during the remainder of this chapter. In this section, it can be used to generalize Theorem 1 by the following theorem.

**Theorem 2.** *Let $\Pi = \langle \mathcal{V}, \mathcal{O}, I, G \rangle$ denote a planning task with facts $\mathcal{F}$, and let $\mathtt{P}$ denote a potential function, and*

*(i) for every variable $V \in \mathcal{V}$, let $G_V \subseteq \mathcal{F}_V$ denote a disambiguation of $V$ for $G$ s.t. $|G_V| \geq 1$, and*

*(ii) for every operator $o \in \mathcal{O}$ and every variable $V \in \text{vars}(\text{eff}(o))$, let $E^o_V \subseteq \mathcal{F}_V$ denote a disambiguation of $V$ for $\text{pre}(o)$ s.t. $|E^o_V| \geq 1$.*

*If*

$$\sum_{V \in \mathcal{V}} \max_{f \in G_V} \mathtt{P}(f) \leq 0 \tag{1.3}$$

*and for every operator $o \in \mathcal{O}$ it holds that*

$$\sum_{V \in \text{vars}(\text{eff}(o))} \max_{f \in E^o_V} \mathtt{P}(f) - \sum_{f \in \text{eff}(o)} \mathtt{P}(f) \leq \text{c}(o) \tag{1.4}$$

*then the potential heuristic* P *is admissible.*

Fišer et al. prove the theorem by showing that Equations (1.3) and (1.4) are generalizations of Equations (1.1) and (1.2), respectively.
Both $G_V$ and $E^o_V$ are equal to $D_V \in \mathcal{D}$ which can be derived with Algorithm 1 of the goal state or $pre(o)$, respectively. If $G_V$ is empty for any of the variables, then the problem is unsolvable, as the goal contains a mutex and is therefore not a reachable state. $o$ is not applicable in any (partial) state, if $E^o_V$ is empty for any $V \in \text{vars}(\text{eff}(o))$.
To show the reader why this property is useful in practice, we first introduce the Transition Normal Form.

## 1.2 Transition Normal Form

Planning tasks can be in Transition Normal Form (TNF). A planning task in TNF has a fully defined goal ($\text{vars}(G) = \mathcal{V}$) and all variables of the effect are also in the precondition for each operator $o \in \mathcal{O}$ ($\text{vars}(\text{pre}(o)) = \text{vars}(\text{eff}(o))$). Any planing task $\Pi = \langle \mathcal{V}, \mathcal{O}, I, G \rangle$ can be transformed into TNF with the following rules cited from Fišer et al.:

- Add a fresh value $U$ to the domain of every variable.
- For every variable $V \in \mathcal{V}$ and every fact $f \in \mathcal{F}_V$, $f \neq \langle V, U \rangle$, add a new *forgetting* operator $o_f$ with $\text{pre}(o_f) = \{f\}$ and $\text{eff}(o_f) = \{\langle V, U \rangle\}$ and the cost $\text{c}(o_f) = 0$.
- For every operator $o \in \mathcal{O}$ and every variable $V \in \mathcal{V}$:
  - If $V \in \text{vars}(\text{pre}(o))$ and $V \notin \text{vars}(\text{eff}(o))$, then add $\langle V, \text{pre}(o)[V] \rangle$ to $\text{eff}(o)$.
  - If $V \in \text{vars}(\text{eff}(o))$ and $V \notin \text{vars}(\text{pre}(o))$, then add $\langle V, U \rangle$ to $\text{pre}(o)$.
- For every $V \in \mathcal{V} \setminus \text{vars}(G)$ add $\langle V, U \rangle$ to $G$.

Each Variable $V \in \mathcal{V}$ gets a new value $U$ in its domain, which can be seen as a sort of placeholder. The fact $\langle V, U \rangle$ can be assigned with cost 0, as the forgetting operator $o_f$ which assigns it has no cost, regardless of the current state and especially the current assignment of $V$. The next point is to assure that for each operator the variables which are in the precondition are also in the effect.
If $V$ is present in the preconditions of an operator $o \in \mathcal{O}$ but not in the effect, then we can simply add the variable and the value it is already assigned to the effect. This is a formal

change, but does not change the effect of the operator at all, as it would not have changed this fact anyway.
The case of an operator $o \in \mathcal{O}$, where $V$ is in the effect but not in the precondition, is a little more complicated. Here, the precondition is changed such that it contains also the fact $\langle V, U \rangle$. If $o$ was applicable in $s$ before, then, after transforming the plan into TNF, the corresponding $o_f$ needs to be applied beforehand in order to forget the value of $V$. This change of the variable is insignificant, as the value then gets changed by applying the operator anyways.
Last, all variables which were not included in the partial state $G$ need to be added into it. If they are assigned the fresh value $U$, then the goal state can be reached from every state which expanded it before. Without creating more cost, the values of all unimportant variables are changed to the fresh value. The compilation into TNF can produce a plan twice the size of the original task in worst case [3].

### 1.2.1 Generalization with Mutexes

Similar to Section 1.1.1, these rules can be generalized with disambiguations. Therefore, to replace $U$ we introduce the fresh values $U_{G_V}$ and $U_{E_V^o}$. Instead of adding forgetting operators from every fact in every $V \in \mathcal{V}$, we use the disambiguation sets $G_V$ and $E_V^o$.

- Add fresh value $U_{G_V}$ to the domain of every $V \in \mathcal{V}$.
- For every variable $V \in \mathcal{V}$ and every fact $f \in G_V$, $f \neq \langle V, U_{G_V} \rangle$, add new *forgetting* operators $o_{f_G}$ with $\text{pre}(o_{f_G}) = \{f\}$ and $\text{eff}(o_{f_G}) = \{\langle V, U_{G_V} \rangle\}$ and the cost $\text{c}(o_{f_G}) = 0$.
- For every $V \in \mathcal{V} \setminus \text{vars}(G)$ add $\langle V, U_{G_V} \rangle$ to $G$.
- For every operator $o \in \mathcal{O}$ add fresh value $U_{E_V^o}$ to the domain of every $V \in \mathcal{V}$:
  - If $V \in \text{vars}(\text{pre}(o))$ and $V \notin \text{vars}(\text{eff}(o))$, then add $\langle V, \text{pre}(o)[V] \rangle$ to $\text{eff}(o)$.
  - If $V \in \text{vars}(\text{eff}(o))$ and $V \notin \text{vars}(\text{pre}(o))$, then add $\langle V, U_{E_V^o} \rangle$ to $\text{pre}(o)$.
- For every variable $V \in \mathcal{V}$ and every fact $f \in E_V^o$, $f \neq \langle V, U_{E_V^o} \rangle$, add new *forgetting* operators $o_{f_E}$ with $\text{pre}(o_{f_E}) = \{f\}$ and $\text{eff}(o_{f_E}) = \{\langle V, U_{E_V^o} \rangle\}$ and the cost $\text{c}(o_{f_E}) = 0$.

For the goal state, forgetting operators are only created for the facts in $F_V$ which are not a mutex with any $f \in G$ for every $V \notin \text{vars}(G)$. Similarly, facts in $F_V$ which are a mutex with any $f \in \text{pre}(o)$ are not taken into account for all $o \in \mathcal{O}$ and every $V \in \text{vars}(\text{eff}(o))$. This creates at most $|\mathcal{O}| * |\mathcal{V}|$ $U_{E_V^o}$ and $|\mathcal{V}|$ $U_{G_V}$, and the same amount of forgetting operators in the worst case. In practice, Fišer et al. show that the transformation with disambiguations is never bigger than without disambiguations.

## 1.3 Linear Program

The formulas and rules which were defined in the previous two sections can now be used to form a Linear Program (LP).

An LP consists of LP-variables which are constrained by multiple inequalities (constraints) and which are part of an optimization function. An LP-solver then assigns each LP-variable a value, such that all constraints are satisfied and the optimization function is optimized. We will look at different optimization functions in the next section (1.4).

**Definition 3.** *Let $f$ be a solution to the following LP:*
*Maximize* opt *subject to* $\sum_{V\in\mathcal{V}} \mathtt{P}_{\langle V,s[V]\rangle} \leq 0$ *and* $\sum_{V\in\text{vars}(\text{eff}(o))}(\mathtt{P}_{\langle V,\text{pre}(o)[V]\rangle} - \mathtt{P}_{\langle V,\text{eff}(o)[V]\rangle}) \leq \text{c}(o)$ *for all* $o \in \mathcal{O}$*, where the objective function* opt *can be chosen arbitrarily.*
*Then the function* $\text{pot}_{\text{opt}}(\langle V, v\rangle) = f(\mathtt{P}_{\langle V,v\rangle})$ *is the* potential function optimized for opt *and* $h^{\text{pot}}_{\text{opt}}$ *is the* potential heuristic optimized for opt.

In order to find a potential heuristic, the LP-variables are the potentials of the facts, $P(f)$. Further we define the constraints as Equation (1.3) for the goal state and Equations (1.4) for every operator. Since these two formulas assure admissibility, any solution of the LP builds an admissible $h^{\mathtt{P}}$. We introduce the the LP-variables $X_f = \mathtt{P}(f)$ for every $f \in \mathcal{F}$ and $M_{G_V}$ and $M_{E^o_V}$ corresponding to $U_{G_V}$ and $U_{E^o_V}$, respectively, with the constraint that $X_f \leq M_{G_V}$ and $X_f \leq M_{E^o_V}$ for every $f \in \mathcal{F}$. This gives the constraints

$$\sum_{f\in G} X_f + \sum_{V\in\mathcal{V}\setminus\text{vars}(G)} M_{G_V} \leq 0 \tag{1.5}$$

and

$$\sum_{f\in\text{pre}^*(o)} X_f + \sum_{V\in\text{vars}^*(o)} M_{E^o_V} - \sum_{f\in\text{eff}(o)} X_f \leq c(o) \tag{1.6}$$

which are coherent to the formulas in Definition 3 (Pommerening et al.). $M_{G_V}$ and $M_{E^o_V}$ correspond to $U_{G_V}$ and $U_{E^o_V}$, respectively, since these are the facts which are assigned if a variable is not defined in the goal state or in a precondition of an operator.
The solution of the LP might differ vastly depending on the used optimization function. We will look at multiple different possibilities for optimization functions.

## 1.4 Optimization Functions

An optimization function opt is a linear combination of the LP-variables. In our case, the goal is to maximize it in order to maximize the average $h^{\mathtt{P}}$-value.
In the first proposal for potential heuristics from Pommerening et al., the optimization for the initial state was used,

$$\text{opt}_I = \sum_{f\in I} \mathtt{P}(f). \tag{1.7}$$

The drawback of this function is that facts which do not appear in the initial state are not taken into account. One solution to this would be to recalculate the potentials for each single state, but this is computationally too expensive.
Alternatively, we could optimize the potentials for all reachable states at once, with the all-states-potentials optimization function:

$$\text{opt}_{\mathcal{R}} = \frac{1}{|\mathcal{R}|} \sum_{s\in\mathcal{R}} \sum_{f\in s} \mathtt{P}(f). \tag{1.8}$$

It calculates the weighted sum of all facts, i.e., it multiplies the potential of a fact with the amount of reachable states containing this fact and normalizes it with the total amount of reachable states. The potentials generated with this optimization function would result in the heuristic with the maximal average heuristic value over all reachable states. Unfortunately, calculating this is, again, computationally expensive or even infeasible, if the planning task and therefore the size of $\mathcal{R}$ are big. To avoid this, we could sample some states $S \subseteq \mathcal{R}$, and calculate Equation (1.8) over these states, instead of $\mathcal{R}$:

$$\text{opt}_{\hat{\mathcal{S}}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \sum_{f \in s} \text{P}(f). \tag{1.9}$$

The optimization function could also assume uniform distribution and give all facts the same weight. Instead of going over all facts of all states, we would sum over the domains of all variables:

$$\text{opt}_{\mathcal{S}} = \sum_{\langle V, v \rangle \in \mathcal{F}} \frac{1}{|\text{dom}(V)|} \text{P}(\langle V, v \rangle). \tag{1.10}$$

These two approaches for the all-states-potential optimization function can be strengthened with mutexes and disambiguations, which we will describe in the next sections.

### 1.4.1 Strengthening All State Potentials

To estimate the amount of reachable states containing $f = \langle V, v \rangle$ we will calculate the upper bound of these states, and try to lower it. The product of the domains of all variables except $V$, since $V$ is already assigned, is the total amount of (non-reachable) states which contain $f$. With Algorithm 1 from Section 1.1.1 we can remove all facts from all domains which are mutex with $f$. These facts could never be assigned in any reachable state containing $f$. Therefore, the product over all domains in the resulting disambiguation set is again an upper bound of appearances of $f$.

This holds not only for a single fact, but can be applied to estimate the appearances of any partial state $p$. As the product of all domains in the disambiguation set is taken, the value is zero, if $p$ is a mutex itself.

If we define $\mathcal{M}$ as the superset of all mutexes and $\mathcal{M}_p$ as the set of all facts which form a mutex with patial state $p$, then an upper bound for the amount of reachable states of size $k$ containing $f$ is

$$\mathcal{C}_f^k(\mathcal{M}) = \sum_{p \in \mathcal{P}_k^{\{f\}}} \prod_{V \in \mathcal{V}} |F_V \setminus \mathcal{M}_p|. \tag{1.11}$$

The Equation has the constraint to only extend states to size $k$, as it would get computationally too expensive to compute all reachable states which hold $f$.

We will now use Equation (1.11) to calculate the weights for the potential of each $f \in \mathcal{F}$ and form the following optimization function:

$$\text{opt}_{\mathcal{M}}^k = \sum_{f = \langle V, v \rangle \in \mathcal{F}} \frac{\mathcal{C}_f^k(\mathcal{M})}{\sum_{f' \in \mathcal{F}_V} \mathcal{C}_{f'}^k(\mathcal{M})} \text{P}(f). \tag{1.12}$$

The result is a modification of $\text{opt}_{\hat{S}}$ (Eq. (1.9)) that des not assume uniformly distributed facts. Each fact is weighted according to its appearance in all reachable states of size $k$ in relation to the other facts of the corresponding variable. The sum of the weights for all facts of one variable is 1. We can use this not only as a single heuristic, but also in ensemble heuristics as we will show in the next section.

### 1.4.2 Strengthening Conditioned Ensemble Potentials

If we divide $\mathcal{R}$ into multiple subsets $S_i \subset \mathcal{R}$, with $i = 1, \ldots, n$ and $S_1 \cup \cdots \cup S_n = \mathcal{R}$, the solution of the LP with optimization function $\text{opt}_{\hat{S}_i}$ gives better heuristic values for the states in $S_i$ than $\text{opt}_{\mathcal{R}}$. If we calculate the potentials for the optimization of all $S_i$, the resulting heuristics can be used as ensemble heuristics. This gives a result which lies somewhere between the all-states-potential heuristic and calculating the potentials for each individual state.

We will choose $S_i$ as the states extending one particular partial state $t$, and use the potentials generated with an adjusted $\text{opt}^k_{\mathcal{M}}$ as one of multiple ensemble heuristics. For this we adapt Equation (1.11), such that it counts how many reachable states of size $|t| + k$ extend $t$:

$$\mathcal{K}^k_f(\mathcal{M}, t) = \sum_{p \in \mathcal{P}^{t \cup \{f\}}_{|t|+k}} \prod_{V \in \mathcal{V}} |\mathcal{F} \setminus \mathcal{M}_p|. \tag{1.13}$$

The corresponding optimization function uses the weights calculated with $\mathcal{K}^k_f(\mathcal{M}, t)$,

$$\text{opt}^{t,k}_{\mathcal{M}} = \sum_{f=\langle V,v\rangle \in \mathcal{F}} \frac{\mathcal{K}^k_f(\mathcal{M}, t)}{\sum_{f' \in \mathcal{F}_V} \mathcal{K}^k_f(\mathcal{M}, t)} \mathtt{P}(f). \tag{1.14}$$

For this thesis, the partial states $t$ were sampled uniformly at random, as did Fišer et al.. Future research could be to investigate other ways to choose them.

### 1.4.3 Adding Constraint on Initial State

Both $\text{opt}^k_{\mathcal{M}}$ and $\text{opt}^{t,k}_{\mathcal{M}}$ optimize the potentials in regard of the entire reachable state space. However, the importance of states may vary strongly, depending on where we start and what path is taken from there towards the goal. For example, a planning task with multiple goal states may have smaller heuristic values, even though the constraints enforce goal-awareness. This is why we now look at a third approach which gives the initial state more weight and adds the constraint

$$\sum_{f \in I} \mathtt{P}(f) = h^{\mathtt{P}}_I(I) \tag{1.15}$$

to the linear program, where $h^{\mathtt{P}}_I$ denotes the potential heuristic optimized for the initial state (Eq. (1.7)). By calculating the potentials for this heuristic first and then taking it into account for the actual potentials, we force the solver to find potentials which guarantee high heuristic values for the initial state. This approach can be combined with all previously discussed optimization functions.

# Bibliography

[1] Daniel Fišer, Rostislav Horčík, and Antonín Komenda. Strengthening potential heuristics with mutexes and disambiguations. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 124–133, 2020.

[2] Florian Pommerening, Malte Helmert, Gabriele Röger, and Jendrik Seipp. From non-negative to general operator cost partitioning. 2015.

[3] Jendrik Seipp, Florian Pommerening, and Malte Helmert. New optimization functions for potential heuristics. 2015.

# Declaration on Scientific Integrity
# Erklärung zur wissenschaftlichen Redlichkeit

includes Declaration on Plagiarism and Fraud
beinhaltet Erklärung zu Plagiat und Betrug

**Author — Autor**
Salome Müller

**Matriculation number — Matrikelnummer**
2017-063-058

**Title of work — Titel der Arbeit**
Mutex Based Potential Heuristics

**Type of work — Typ der Arbeit**
Bachelor thesis

**Declaration — Erklärung**
I hereby declare that this submission is my own work and that I have fully acknowledged the assistance received in completing this work and that it contains no material that has not been formally acknowledged. I have mentioned all source materials used and have cited these in accordance with recognised scientific rules.

Hiermit erkläre ich, dass mir bei der Abfassung dieser Arbeit nur die darin angegebene Hilfe zuteil wurde und dass ich sie nur mit den in der Arbeit angegebenen Hilfsmitteln verfasst habe. Ich habe sämtliche verwendeten Quellen erwähnt und gemäss anerkannten wissenschaftlichen Regeln zitiert.

Basel, 06. 11. 2020

---

**Signature — Unterschrift**