ONERA

Search Space Splitting in order to compute admissible heuristics in Planning

Yacine Zemali

Ph.D. student at ONERA / SUPAERO

Advisors : P. Fabiani and

M. Ghallab (LAAS-CNRS)

Talk by Stefan Edelkamp (Universität Dortmund)

Centre d'Études et de Recherches de
Toulouse

ONERA

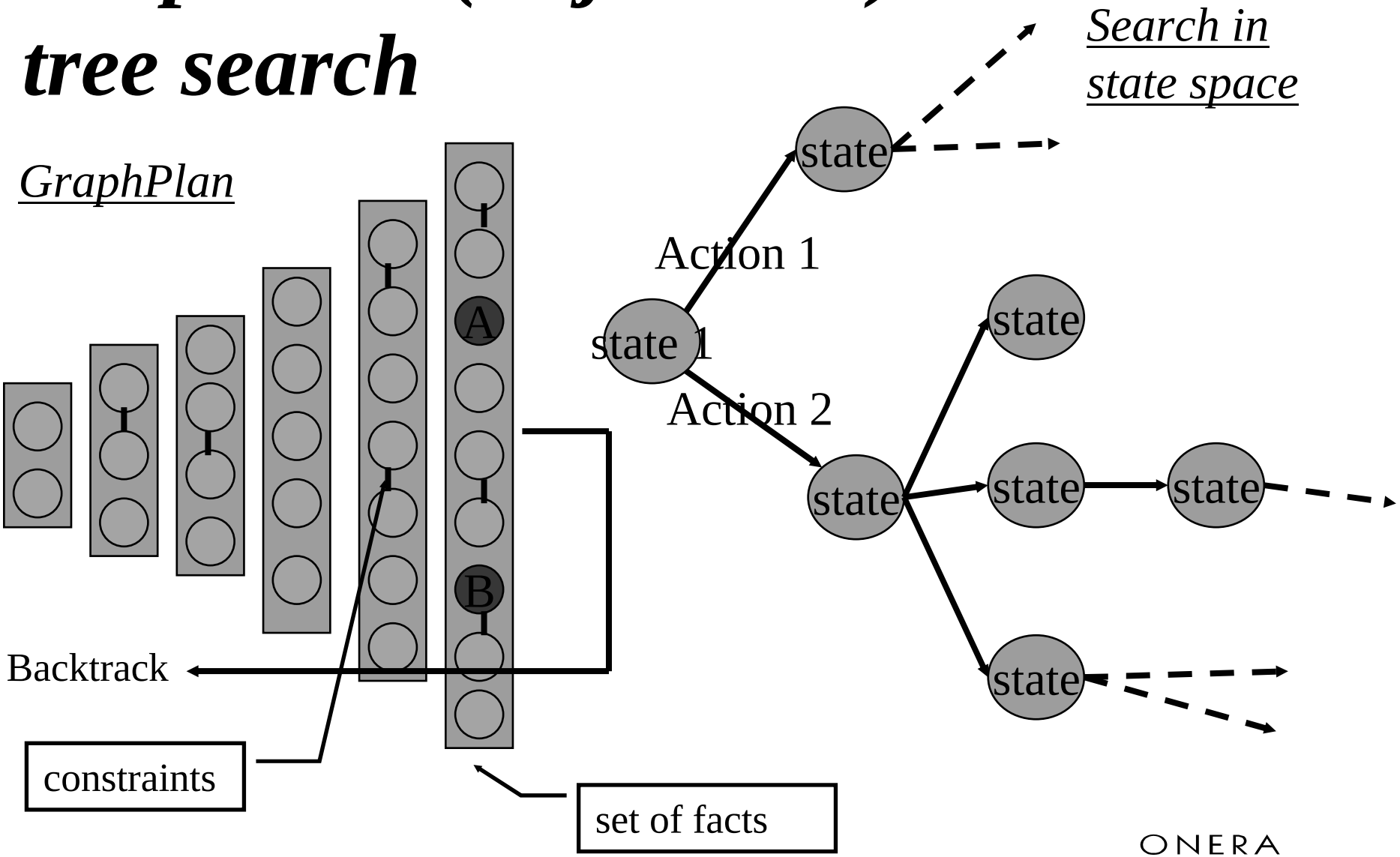
Introduction motivations

Centre d'Études et de Recherches de Toulouse

Introduction / definitions

- **Given: an initial and a final state, a set of actions that can modify the current state**
- **Goal: to find a sequence of actions that leads to the goals**
- **Problem: to explore every possible combination is not possible**
- **Classical AI planning: deals with symbolic data independent from the described domain**

GraphPlan (disjunctive) Vs tree search



ONERA

Heuristic planning

- **Map the planning problem to a local search problem**
- **General case:**
  - nodes → states (described by properties or facts)
  - edges → actions (modifying properties from one state to another)
- **Guided search with heuristics (e.g. estimate of plan length)**

Definitions

- **Positive interaction:**
  - 2 actions have a positive interaction if they produce one same fact.
- **Negative interaction:**
  - 2 actions have a negative interaction if one of them deletes a fact that the other produces

Mini state of the art

- **HSP / HSPr [Geffner]: solve a relaxed problem by assuming that all actions are independent**
- **FF [Hoffmann]: same idea, but takes into account some of the positive interactions**
- **AltAlt[Kambhampati]: some positive and negative interactions**

More interactions

Motivations

- **The more we take interactions into account, the more the heuristic is informative**
- **FF takes into account a large part of positive interactions. Based on a close framework, we want to add the possibility of taking some negative interactions into account. The main difficulty is to choose the interactions to be taken into account: it should be the ones that add accuracy, but that are not too much time consuming**

ONERA

Our planning system

Centre d'Études et de Recherches de
Toulouse

Our approach

- **Positive interactions are taken into account**
- **Negative interactions are taken into account in a controlled way**
- **The control strategy can be adapted to the problem (automatically or by hand)**

Why ?

- **The more we can take interactions into account, the more informative is the heuristic**
- **It is not possible to take all the interactions into account (NP-hard)**

How ?

- **The heuristic is derived from a disjunctive planning graph in which some facts are grouped into sets thanks to the state space splitting**

Search scheme

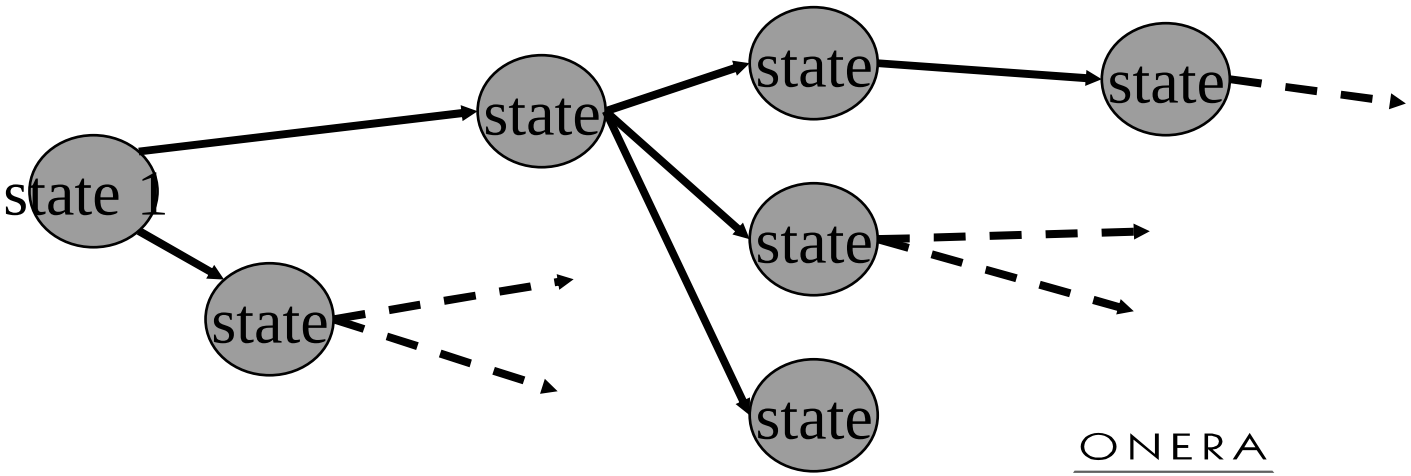
- ***A* forward*** **or** ***backward*** **(to avoid** ***h*** **recomputations) for an optimal solution**
- ***Hill Climbing*** **for a quick solution**

State space splitting: our definition

- **While the disjunctive planning graph is build, some facts a grouped together into sets**
- **The facts are grouped together according to a splitting strategy**
- **The splitted planning graph is a leveled disjunctive planning graph that contains a tree structure**

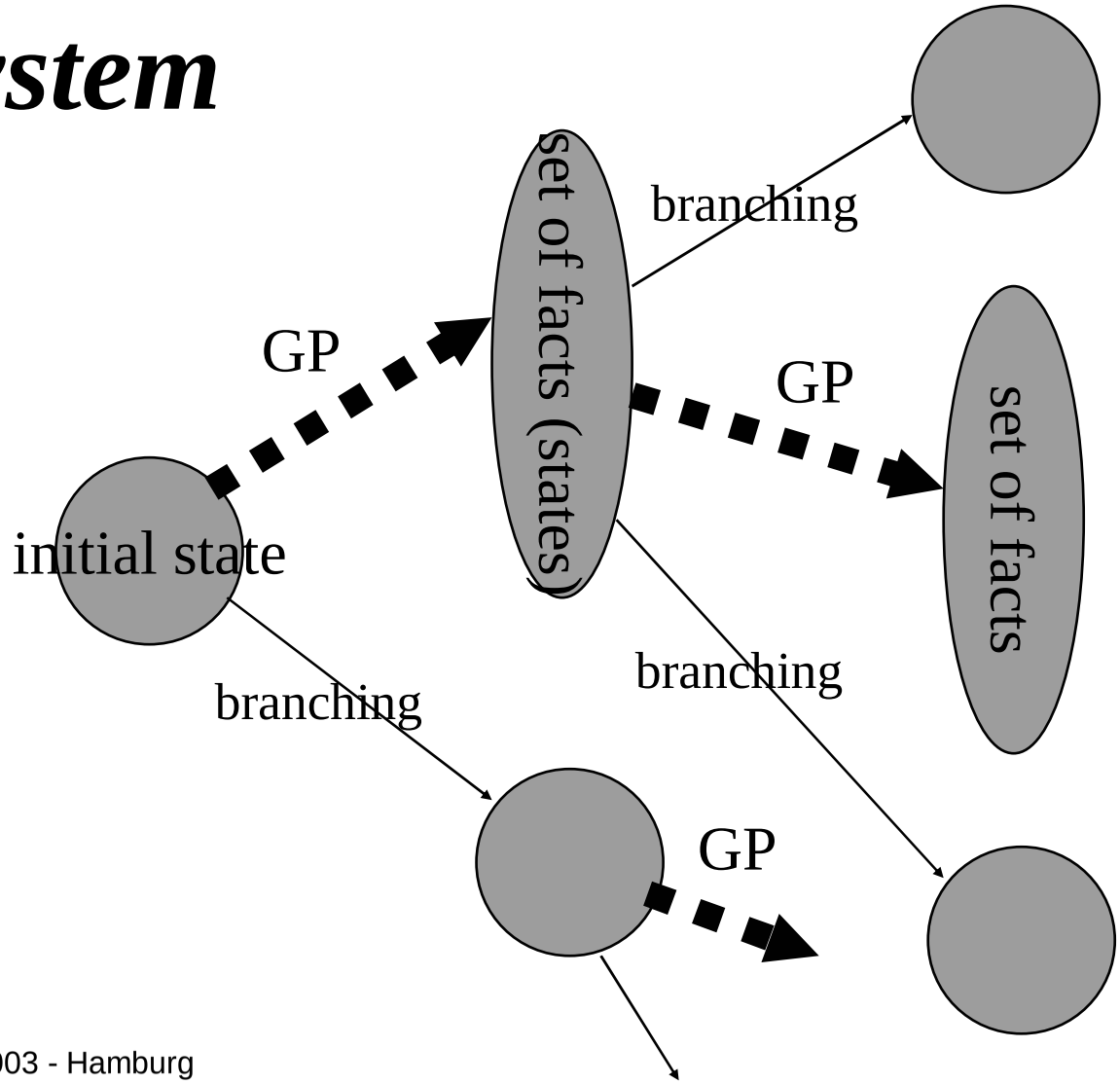
State space splitting: intuition

- **GraphPlan ⇔ no splitting because all facts are grouped together (and thus all states described by those facts) in a same level**
- **Forward state search (FSS) ⇔ full splitting**



ONERA

Search state splitting in our system



ONERA

The heuristic

- **Actions that create lots of incoherencies are the ones which create new branches. Those are the most “interactive” actions that are determined off-line**
- **The number of level of the splitted planning graph is an admissible heuristic**

ONERA

Conclusion

Centre d'Études et de Recherches de
Toulouse

Conclusion

- **Several splitting strategies can be imagined:**
  - We can select some actions off-line and label them as critical actions that should be applied in their own branch. For instance, *harmful* actions [J. Hoffmann] (non invertible actions that destroy a fact which is necessary to achieve the goal). When relaxing such an action the search algorithm can fall into a *dead-end*. That is one good reason not to relax such an action while computing the heuristic.
- **Storage of intermediate results**
  - Intermediate results are stored into decision trees. Those data structure could be reduced by using MTbdd (multi terminal binary decision diagram) algorithms

ONERA

Questions ...

zemali@onera.fr

Centre d'Études et de Recherches de
Toulouse