Planning applications

Where the real challenges are

Jana Koehler
Lucerne University of Applied Sciences and Arts

jana.koehler@hslu.ch
Agenda

- A planning system used by millions of people every day …

- How easy is it to apply a state-of-the-art planner?
Conventional Elevator Control

1. Outside the cabin:
   One or two buttons to call elevator

2. Inside the cabin:
   One button per floor
Alternative: Destination Control

1. passenger enters destination floor
2. terminal indicates best elevator
3. passenger walks to elevator
4. destination indicator in door frame; no buttons inside cabin
Conventional vs. Destination Control

- Press twice
- Jump on the first elevator that stops
- Conglomerate of passengers

- Press once
- Walk to designated elevator
- Separation of passengers by destination
Main Driver 1: Mixed Usage of Buildings

94-93 observation
90-61 hotel
79-56 office
55-49 hotel
55-6 office
3 shops
2 hotel lobby
1 office lobby
-1 to -3 parking

Shanghai World Financial Center
Illustration: Mori Building Co. Ltd., Japan
Main Driver 2: Increase Customer Value

- Less space
- Less energy costs
- Higher performance
  - Less waiting time
  - Faster traveling
  - More direct travels

- Diversification of products
  - New services
  - Customization
New User Interfaces
- Individual **space requirements**
- Desired **travel direction**
- **VIP service** depending on status **and** traffic situation
- **Access restrictions** to zones in building
- **Separation** of passenger groups
- **Multi-deck** elevators
Schindler’s First Destination Control Algorithm

- Each elevator submits an offer
  - Serve new passenger as early as possible
  - Rule-based allocation scheme
- Terminal selects “less-disturbed” car
- Impossible to add new services
The Problem

- Simple rule-based allocations fail
  - Transportation performance decreases heavily
  - Rule set becomes complicated and incomprehensible
  - State space explodes, impossible to enumerate it explicitly

- New solution should be configuration free
  - Varying hardware configurations and frequently changing customer needs
  - Develop modular software architecture
  - Do not program control in advance, but compute it online
«Aktions» of Elevators

- Stop at floor
- Open door
- Close door
- Move up/down (2 – 10 meters/s)

- 0 - 3 - 5 - 7 - 4 - 9 ...

- $10^{10}$ - $10^{12}$ states
- Find optimal sequence in <100 ms
  - Minimal waiting and traveling times
  - Guarantee additional constraints
How does it Work?

Technology 1: Run an auction
Technology 2: Search for an optimal sequence of stops

Ask car planners for offers and compare

Select best car and request confirmation
A Behavioral Model of Passengers

- **Waiting** passengers enter as soon as the elevator reaches their entry floor

- **Boarded** passengers leave as soon as the elevator reaches their destination floor
  - Behavior of passengers cannot be planned
    - Non-selective boarding!
    - Boarding and leaving of passengers as side-effects of elevator behavior

- Algorithm enumerates possible actions of the elevator and determines their impact on passengers according to the behavioral model
The Planning Offline Problem

**initial state:**
set of destination calls with status information
“<31,5,Waiting>”, “<15,2,Boarded>”

position of car

**goal:**
carry all passengers to their destination

**actions:**
stop at floors, open/close doors, move up/down

评定：optimal sequence of stops

NP-hard, TSP-variant, feedback vertex set, point-to-point pairwise connection
The Search Algorithm

- Systematic, depth-first search
  - Branch-and-bound
  - Optimization criteria encoded in heuristic function
  - Forward checking to propagate constraints over non-expanded states
  - Domain-specific state space encoding (“tuned” data structures)

- 200,000 states per second can be expanded when all constraints need to be checked

- State size: $10^{10} - 10^{12}$ states
  - Practice: 1000 - 2000 states explored until optimum found
  - Chess: $10^{40}$ possible positions, Go: $200^{300}$
The Online Problem

- Planner solves a static traffic problem given at a certain moment in time

- Planning problem changes frequently
  - New passengers call
  - Passengers ‘misbehave’ (block doors, don’t register call)
  - Hardware failures can occur

Each new call needs to be allocated to the ‘best’ car
Plan execution needs to respond to external or planned changes
Graceful degradation in case of technical failures
Distributed Architecture (Multi-Agent System / SOA)

- communication via asynchronous messaging with publish/subscribe
- support of adhoc networking
The Testing Environment
50 % Reduction of Waiting Times during Up-Peak
Flexible Response to High Traffic Volumes
From Collecting Passengers to Shuttle Service

Breakdown

- Number of stops begins to decrease
- Travel times decrease, but waiting times increase
50 % Increased Capacity during an Up-Peak Pattern

Breakdown of conv. Controller
Simulation Environment
Friday, April 28, 2000

25197 calls
Peak: 1 call/s

Avg. waiting time = 88.75
Avg. estimation time = 144.93

new approach
Avg. waiting time = 52.06
- 58%
Avg. destination time = 87.2
- 60%
Heterogeneous Multi Decker Group
Effectiveness of the Heuristic Function

CPU Time: 3.48 s
#Solutions: 77
#Nodes: 724,046

65,332 nodes out of 105,617 nodes at depth 16 (61.86 %)
Forward Checking - Travel Direction + Space

231,732 nodes out of 336,937 nodes at depth 14 (68.78%)
Branching Factor at the Root

Initial Stoplist Width

% of Plans

Number of Stops

- 90P/5min
- 180P/5min
- 270P/5min
Length of Plans by Traffic Intensity

Plan Length (%)

% of Plans

Number of Stops

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

90°/5 min
180°/5 min
270°/5 min
Execution of Plans

Executed Stops (%)

% of Plans

Number of executed Stops

0 1 2 3 4 5 6

0 5 10 15 20 25 30 35 40 45 50

90P/5min
180P/5min
270P/5min
Communication in 2000

SchindlerID is an enhanced user interface paired with a control overlay, which provides access control as a core service. The basic utilization of SchindlerID follows a generic scheme:

1. The passenger is identified by his identification medium or personal PIN code at a SchindlerID terminal in front of the elevator.
2. The planning algorithm checks the passenger's access rights and assigns a car (A, B, C, ...) to the passenger. In doing so, the algorithm takes account of the following criteria:
   - overall traffic capacity
   - individual traveling time
   - security restrictions
   - attributes such as space required, handicapped passengers, preferences, privileges (VIP), or grouping
3. The passenger walks directly to the assigned car and will be transported to his destination without any need for further interaction.

Intended as a platform, SchindlerID allows the integration of a variety of methods and services. For example,
Increase from 1999 to 2000

Total 1482 lifts sold by end of 2000 (increase by 110%)

- EU 373 (237)
- AP 578 (240)
- AM 183 (125)
- IMEA 118 (14)
- S-AM 188 (62)
- AF 25 (13)
- ANZ 17 (17)
- Total 1482 lifts sold by end of 2000 (increase by 110%)
In Switzerland …

Basel

Zurich

Luzern
Other Countries

- groups of 3-8 elevators
- high populations
- new security standards
- “traffic peaks”

Metropolitan Tower
Ho Chi Minh City

Rockefeller Center
New York

Eurotheum Frankfurt

Coeur Defense Paris

Millennium Tower Vienna
Lessons Learned

- Getting the initial 1-2 actions right would be sufficient
- **Sense – Plan – Execute**
  - RESPOND IMMEDIATELY
  - Problem size is BOUNDED
- Each domain needs its own heuristic function
  - BUT likely also its own state representation

- Open system boundaries - need to integrate flexibly
- **Embedding** the AI component is critical to success
  - 6 months developing time for search algorithm
  - 1.5 years for the surrounding controller
10 Years Later …

How easy is it to embedd a state-of-the-art planner?
Business Prozess Management

Business Intelligence

Monitoring & Feedback

Modelling & Analysis

Design & Execution

Model-Driven Software Development
An Internet Order
Choreography Model of Partners

Customer

WebShop

Distributor

Delivery

Financial Service Provider

Producer

BPMN – Business Process Modell and Notation 2.0
Possible Process of the Webshops
Rules can dynamically orchestrate combinations of process components – the process becomes goal-directed
An Experiment

- Domain .pddl
- Problem .pddl
- Parsing SAS+ Translation (Python)
  - arguments
  - allgroups
- Search (C++)
  - output .sas
  - pipe
- Heuristic Function (C++)
  - pipe
  - output
  - must exist (not documented)
- sas_plan.1

File Transfer as the integration pattern
Code mostly platform-independent
Observations

- Runtimes on IPC8-SeqSat Problems
  - Approx. 700 ms, but 52 – 60 % of time in first two modules
  - Cybersecurity: 8000 ms (95%) preprocessing vs. 400 ms search

- No API, not so easy to configure

- Need more modular architecture
  - Well-defined interfaces (eliminate file transfer, define API)
  - Clean separation of interfaces from implementation
  - Modern input/output data representations, e.g. XML would eliminate hand-written parsers
Would you Receive Academic Merit?

- Approx. 120 citations for 3 elevator publications
  - Miconic domain (Bacchus, 2001)

IBM DeveloperWorks Article on Process Anti-patterns

Compare to only 24 citations

Essential for WebSphere Process Runtime
BPM paper (LNCS) not in Harzing PoP
Extended DKE paper 34 citations
Simpler ICSOC paper 114 citations
Conclusions

- Doing real applications is fun
- Not much overlap with academic value system
- Technology transfer mostly focused on software engineering
- Does not necessarily make you rich