Vorteile der Constraint-Programmierung

- die deklarative Modellierung von Problemen mithilfe passender Constraintsysteme, die schneller zu robuster, flexibler und wartbarer Software führt,
- die Möglichkeit zur korrekten Darstellung unvollständiger, spärlicher als auch vollständiger Information durch Constraints,
- die automatische Propagierung der Effekte, wenn neue Information (in Form von Constraints) bekannt wird,
- die gute Kombinierbarkeit von Constraintlösen mit Such- und Optimierungsverfahren.

Anmerkung

Seit Anfang der neunziger Jahre wird constraintbasierte Programmierung mit Erfolg von mehreren Firmen (Ilog mit IlogSolver und IlogSchedule, Cosytec mit CHIP 4, Siemens Nixdorf mit IFProlog, Prologia mit Prolog IV) weltweit kommerziell eingesetzt. Die Zahl der kommerziellen Anwendungen wurde 1996 auf 300 geschätzt, der Umsatz mit Constrainttechnologie auf etwa 100 Millionen Dollar, mit steigender Tendenz. Die erwähnten Firmen haben jeweils mehrere hundert Kunden für ihre constraintbasierten Produkte gefunden. Während Frankreich, England, USA und Asien stark wachsende Märkte für constraintbasierte Entscheidungshilfesysteme sind, ist der Markt in Deutschland erst im Entstehen.

Marktüberblick nach Branchen

• Transportwesen (British Airways, Lufthansa, Delta Airlines, SNCF)
• Automobil- und Flugzeugindustrie (Ford, Renault, Peugeot Citroën, Daimler-Benz, Michelin, Lockheed Martin, Airbus)
• Elektronik (IBM, Alcatel, Philips, Hewlett-Packard, Whirlpool, Bosch, Motorola, Thomson, Siemens, Nokia)
• Telekommunikation (France Telecom, AT & T)
• Finanzwesen (Banken und Versicherungen wie Citicorp, National Westminster Bank)
• Energieversorgung (Shell)
• Nahrungsmittelindustrie (Uncle Ben's)
• Handel (El Corte Ingles)
• Chemie (Rhone Poulenc)
• Militär (Dassault)

Keine Anmerkung

Anwendungsgebiete

- Modellierung,
- (ausführbare) Spezifikation,
- Design,
- Synthese,
- Simulation,
- Verifikation,
- Fehlerdiagnose

von elektronischen, elektrischen, mechanischen und Software-Komponenten sowie industriellen Abläufen.

Keine Anmerkung

Anwendungsgebiete

- Produktions-,
- Personal-,
- Finanz-,
- Verkehrs-,
- Netzwerk-
- Ressourcen-
- -planung,
- -logistik,
- -management,

- Transport- und Plazierungsoptimierung,
- Design,
- Konfiguration,
- Layoutgenerierung.

Keine Anmerkung

Marktüberblick

- Das System DAYSY von Cosytec adaptiert für die Lufthansa den Einsatz von Personal nach Störungen im Flugbetrieb (Verspätungen, Erkrankung,...).
- Nokia Mobile Phones, zweitgrößter Mobiltelefonhersteller der Welt, verwendet IFProlog zur automatischen Konfiguration von Software für Mobiltelefone.
- Siemens betriebsintern mit IFProlog entwickelte "Circuit Verification Environment" (CVE2) zum Design und zur Verifizierung von Hardware (VLSI Chips).
- Ilog für SNCF "Sagitaire" entwickelt, das im Bereich des Bahnhofes Paris Nord Routing für über 1700 Züge täglich plant.

Anmerkung

Aus den oben genannten Gebieten seien hier einige konkrete kommerzielle Anwendungen kurz beschrieben:

Marktüberblick

- ICL mit DecisionPower für Hongkong International Terminals, einem der größten Containerhafen der Welt, zur optimalen Plazierung von Containern in Lagerhallen zwischen Ankunft und Abfertigung entwickelt.
- Renault "Short Term Production Planning" ein, zur optimalen Planung von Zulieferung und Fertigung von Varianten eines Autotyps innerhalb eines Fertigungsabschnittes.
- Daussault's "Made", wo und wie komplexe Flugzeugteile aus einem Blech herausgeschnitten werden sollen, sodaß möglichst wenig Abfall und Zeitverlust entsteht.

Keine Anmerkung

Planung von Drahtlosen Kommunikationssystemen

Das Werkzeug POPULAR simuliert die Funkwellenausbreitung in Gebäuden und optimiert die Anzahl und die Plazierung von Sendeanlagen für lokale, kabellose Kommunikationsnetze.

Die Funkwellenausbreitung im Gebäude muß berücksichtigt werden.

Das Constraintlogikprogramm für POPULAR ist nur 4000 Zeilen lang, inklusive der Grafik für die Benutzerschnittstelle, die etwa die Hälfte des Programms ausmacht.

Anmerkung

In Deutschland hat bereits mehr als jedes dritte verkaufte Telefon kein Kabel mehr. Allein in Westeuropa sollen 1999 14 Millionen schnurlose Telefone nach dem DECT Standard verkauft werden, etwa ein Drittel davon in Deutschland.

Kabellos kann das interne Telefonnetz einer Firma jederzeit um Teilnehmer erweitert werden, und dies ohne aufwendige Montagearbeiten. Zudem bringt die digitale Datenübertragung mit DECT eine verbesserte Übertragungsqualität bei mehr Abhörschutz.

Modellierung

Die Funkausbreitung wird hauptsächlich durch folgende Faktoren beeinträchtigt:

- Dämpfung des Signals in Abhängigkeit von der Entfernung,
- Abschattung und Beugung durch Hindernisse, und
- Mehrwegeausbreitung durch Reflexion.

Anmerkung

Modellierung

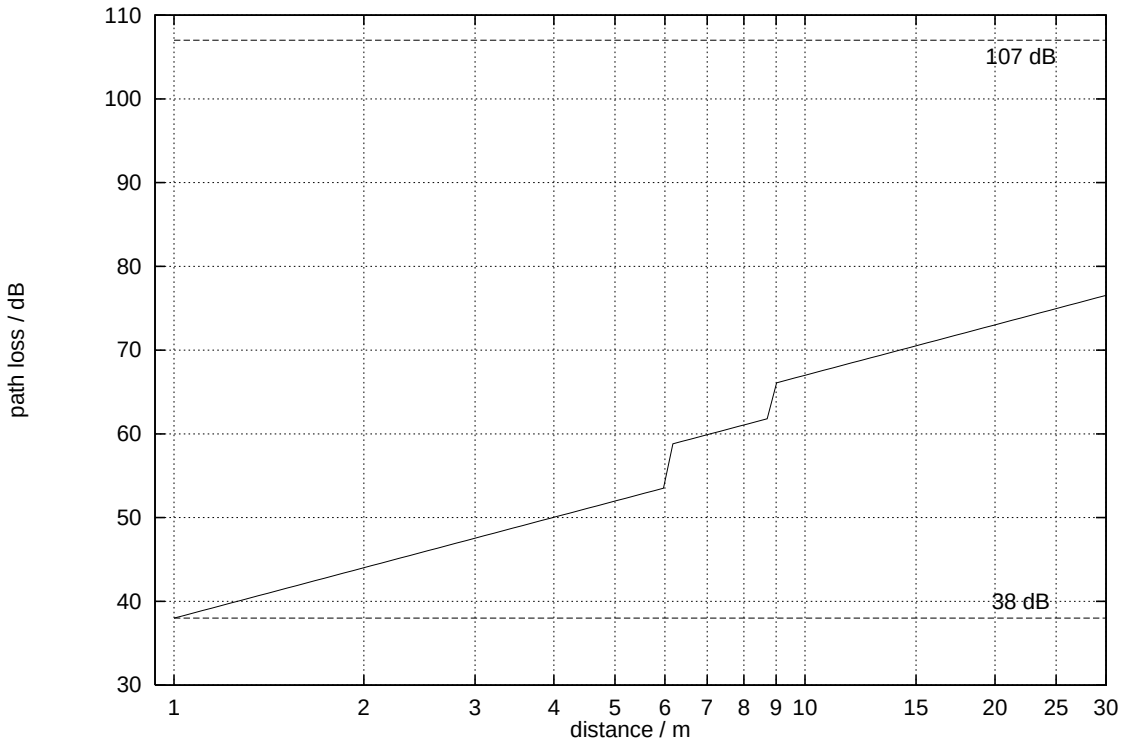


Abbildung 1: Typischer Pfadverlust mit Zusatzdämpfung durch Wände bei $d = 6m$ und $d = 9m$ (logarithmische Darstellung)

Anmerkung

Die Einfügungsdämpfung gibt dabei den Wert wieder, der als zusätzlicher Verlust beim Durchdringen einer bestimmten Wand bzw. Decke entsteht, der Wert ist je nach Baumaterial stark unterschiedlich. Die folgende Abbildung soll das Modell verdeutlichen. Die Darstellung beginnt bei einer Entfernung $d = 1m$ und dem damit verbundenen unteren Grenzwert von 38 dB. Die Sprungstellen in der Funktion entsprechen zwei Wänden im Ausbreitungspfad.

Simulation der Funkausbreitung

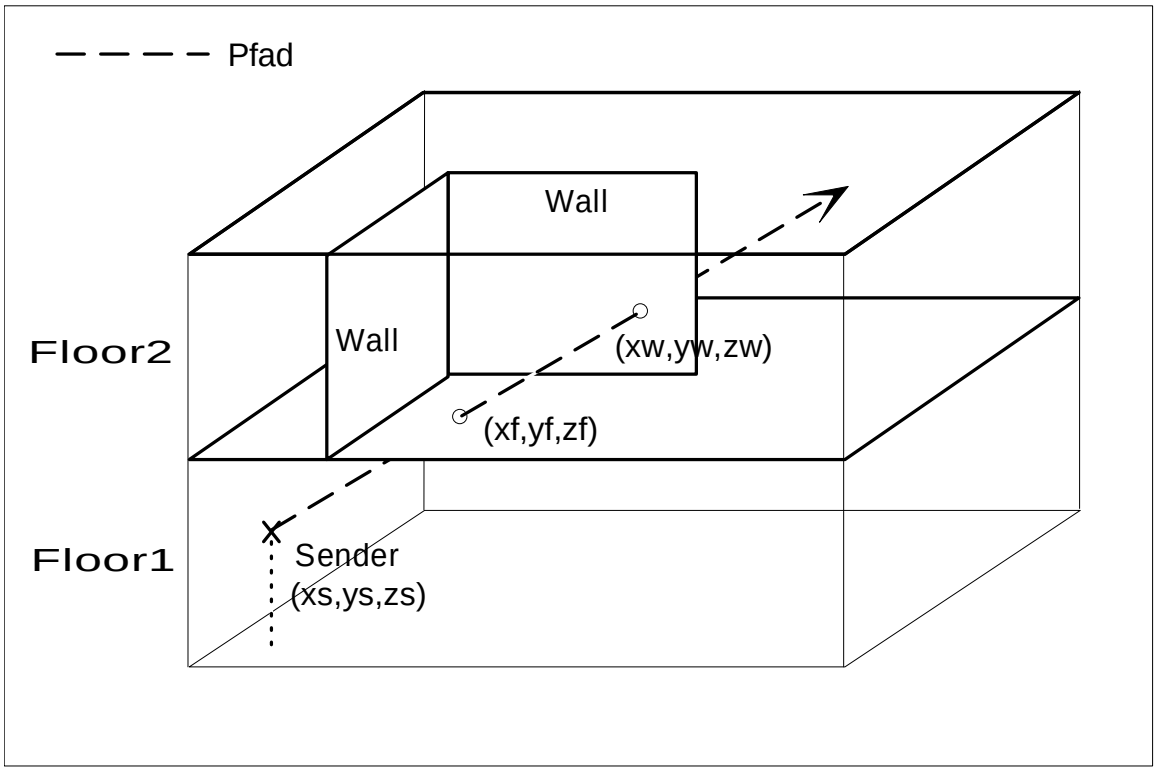
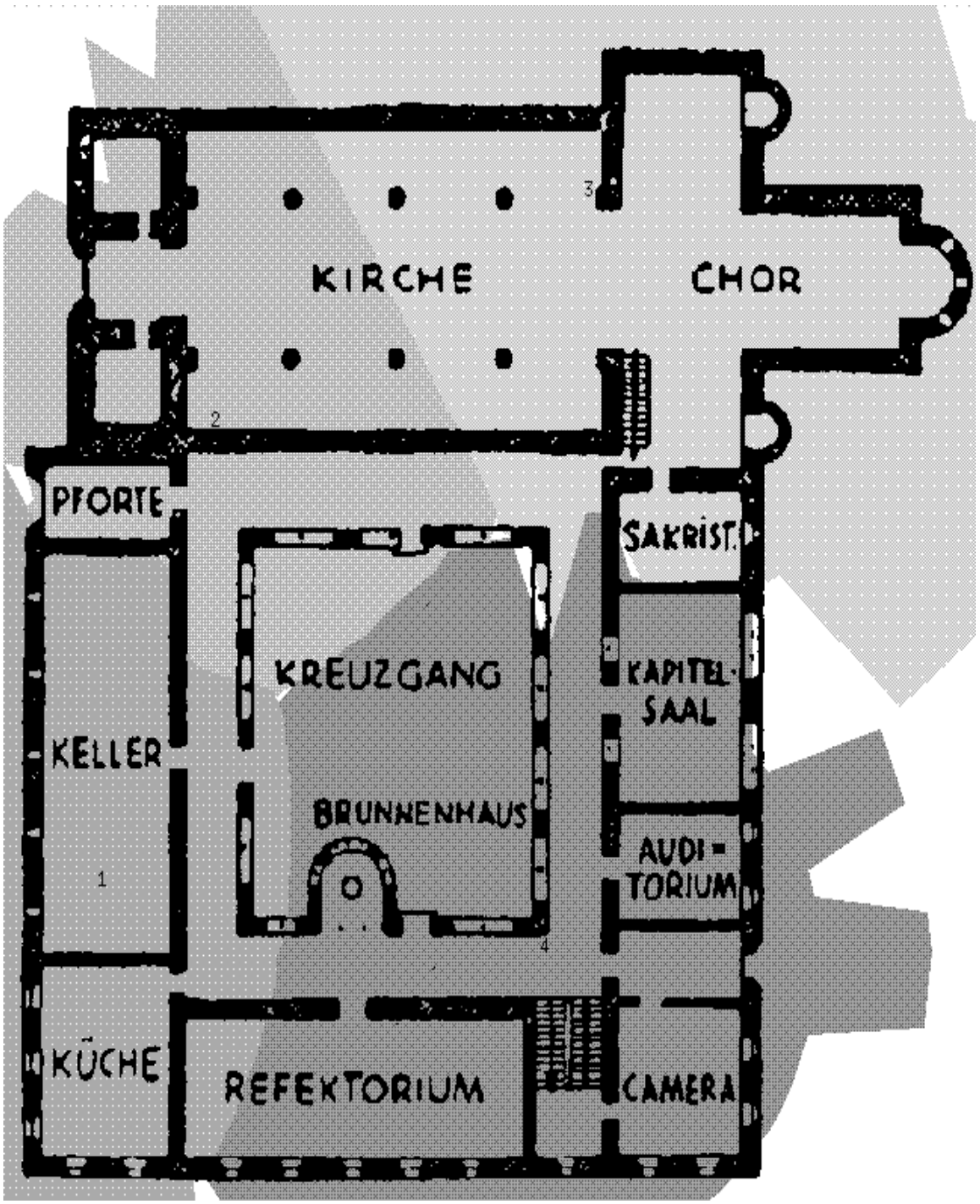


Abbildung 2: Pfad vom Sender durch ein Gebäude mit zwei Hindernissen

Anmerkung

Dazu wird eine fiktive Funkwelle in in einer ausreichenden Anzahl von Richtungen (mehr als hundert) untersucht. Der Ausbreitungspfad wird durch das gesamte Gebäude hindurch verfolgt.

Entlang des Pfades werden alle Schnittpunkte mit Wänden und Decken festgestellt und die zugehörigen Einfügungsdämpfungen registriert. Dabei werden die Antennendämpfung in Pfadrichtung, die entfernungsabhängige Dämpfung und die registrierten Einfügungsdämpfungen längs des Pfades bis zum zulässigen Grenzwert des Pfadverlustes aufaddiert. Danach werden durch binäre Suche entlang des Pfades ausreichend genau die Koordinaten des zugehörigen Grenzwertpunktes ermittelt. Die Hülle der Grenzpunkte eines Testpunktes beschreibt dann die Funkzelle. Eine Funkzelle umfaßt normalerweise mehrere Räume in mehreren Stockwerken.


KIRCHE
CHOR
PFORTE
SAKRIST.
KREUZGANG
KAPITEL-
SAAL
KELLER
BRUNNENHAUS
AUDI-
TORIUM
KÜCHE
REFEKTORIUM
CAMERA
1
2
3
4

Anmerkung

Das Beispiel zeigt das Ergebnis der Versorgung eines mittelalterlichen Klosters mit vier Sendern. Ein Abdeckungsgebiet ist jeweils dem stärksten Sender zugeordnet.

Constraintbasierte Optimierung

```
% inside(Sender, LeftLowerCorner - RightUpperCorner)

not_empty @ inside(S,A#B-C#D) ==> A<C,B<D.

intersect @ inside(S,A1#B1-C1#D1),inside(S,A2#B2-C2#D2) <=>
   A is max(A1,A2), B is max(B1,B2),
   C is min(C1,C2), D is min(D1,D2),
   inside(S,A#B-C#D).
```

Anmerkung

In einer ersten Implementierung des Constraintlösers haben wir uns auf ein Stockwerk beschränkt und jede Funkzelle durch ein Rechteck approximiert. Die 2-D Koordinaten für einen Punkt haben die Form `X#Y`. Rechtecke sind orthogonal zum Koordinatensystem und jeweils durch den linke unteren und rechten oberen Eckpunkt definiert. Für jede Funkzelle wird das Constraint `inside(Sender, Rectangle)` einfgeführt, wobei `Sender` geometrisch gesehen ein Punkt ist, der im Rechteck `Rectangle` liegen muß. Für einen vollständigen Constraintlöser sind nur zwei CHR-Regeln notwendig:

Constraintbasierte Optimierung

```
equate_senders([]).
equate_senders([S|L]) :-
        equate_sender(S,L),
        equate_senders(L).

    equate_sender(S,[]).
    equate_sender(S,[S1|L]) :- S=S1.
    equate_sender(S,[S1|L]) :-
        equate_sender(S,L).
```

Anmerkung

Um eine Lösung zu berechnen, versucht man möglichst viele Sender einander gleichzusetzen, nachdem man alle Constraints eingeführt hat. Für die Enumeration reicht im einfachsten Fall das folgende rekursive Prädikat, das über der Liste aller Sender operiert:

Constraintbasierte Optimierung

```
% inside(Sender, GeomList)

intersect @ inside(S, L1), inside(S, L2) <=>
    intersect_geoms(L1, L2, L3),
    L3 = [Geom|Geoms],  % nicht leer
    inside(S, L3).

 intersect_geoms(L1, L2, L3) <=>
    setof(Geom, intersect_geom(L1,L2,Geom), L3).

  intersect_geom(L1, L2, (A#B-C#D)) <=>
    member((A1#B1-C1#D1), L1),
    member((A2#B2-C2#D2), L2),
    A is max(A1,A2), B is max(B1,B2),
    C is min(C1,C2), D is min(D1,D2),
    A<C, B<D.            % nicht leer

   member(X,[X|L]).
   member(X,[Y|L]) :- member(X,L).
```

Anmerkung

In nur zehn Minuten konnten wir diesen Constraintlöser so erweitern, daß er mit Vereinigungen von Rechtecken funktioniert. Damit können die Funkzellen mit beliebiger Genauigkeit approximiert werden. Die Vereinigung von Rechtecken enstpricht dem disjunktiven Constraint `inside(S,R1) ; inside(S,R2) ; ... ; inside(S,Rn)`. Es wird kompakter als `inside(S,[R1,R2,...,Rn])` geschrieben. `geom` steht für "geometrisches Objekt".