



Vincoli e preferenze come formalismo
unificante per l'analisi di sistemi informatici e
la soluzione di problemi reali

Progetto MIUR PRIN 2005



Obbiettivi

- Studiare estensioni di formalismi e tecniche esistenti per CP
- Applicazioni innovative
 - a sistemi informatici (es. sicurezza, analisi di programmi)
 - a problemi reali (es. scheduling, bioinformatica)
- Vincoli come formalismo unificante
- Applicazioni come spunto per le estensioni di linguaggi e tecniche CP



Cosa faremo

- Studieremo come
 - estendere i formalismi per poter modellare preferenze positive e negative, condizionate e non, qualitative e quantitative
 - modellare e risolvere situazioni con piu' agenti, e studiarne le proprieta' fondamentali
 - sviluppare tecniche per situazioni con piu' criteri di ottimizzazione
 - gestire vincoli su intervalli e su insiemi



Campi applicativi principali

- Problemi di scheduling
- Verifica di proprieta' di sistemi concorrenti a stati infiniti
- Analisi di protocolli di sicurezza multilivello
- Bioinformatica



Unita' di ricerca

- Padova (Francesca Rossi)
 - Vincoli soft, preferenze, multi-agente, multi-obiettivo, scheduling
- Udine (Agostino Dovier)
 - Linguaggi di programmazione logica con vincoli su insiemi, sistemi ibridi, bioinformatica
- Genova (Giorgio Delzanno)
 - Analisi di sistemi tramite tecniche CP
- Chieti-Pescara (Stefano Bistarelli)
 - Linguaggi logici e concorrenti con vincoli, applicazione di CP alla sicurezza



Task 1: linee di lavoro teoriche

- Estensione di formalismi CP per modellare vari tipi di preferenze, sistemi CP multi-agente
 - Padova, Chieti-Pescara
- Sviluppo di tecniche CP/OR per gestire situazioni con piu' criteri di ottimizzazione
 - Padova, Udine
- Gestione di vincoli su intervalli e su (multi)insiemi
 - Udine, Genova



Task 2: linee di lavoro applicative

- Scheduling
 - Padova
- Verifica di proprieta' di sistemi concorrenti a stati infiniti
 - Genova, Udine, Pescara
- Analisi di protocolli di sicurezza multilivello
 - Pescara, Genova
- Bioinformatica
 - Udine



Fasi (per entrambi i task)

- *Mesi 1-6*: studio dello stato dell'arte
- *Mesi 7-12*: sviluppo di nuovi formalismi e/o tecniche
- *Mesi 13-18*: studio delle proprietà dei nuovi formalismi/tecniche
- *Mesi 19-24*: per risultati teorici, nuove proprietà, attività implementative, e validazione sperimentale



Metodo di lavoro

- Individuale nelle sedi, ma anche in collaborazione
- Mailing list del progetto (Padova)
- Sito web con risultati aggiornati (Pescara?)
- Tre workshop: inizio (Padova), dopo 1 anno, fine



Unita' di Padova

- Preferenze
- Incertezza e preferenze
- Sistemi multi-agente
- Ottimizzazione multi-criterio
- Applicazioni a problemi di scheduling



Preferenze e incertezza (Rossi, Venable, Pini)

- Generalizzano i vincoli, dando una valutazione alle soluzioni
- Problemi di ottimizzazione
- Molti tipi di preferenze:
 - Qualitative (preferisco A a B)
 - Quantitative (A mi piace 10, B mi piace 20)
 - Positive (mi piace piu' A di B)
 - Negative (non mi piace A, e ancora meno B)
- Formalismi esistenti: vincoli soft, vincoli valued, CP-nets
- Incertezza: variabili non-controllabili
- Goal: formalismo unificante per i vari tipi di preferenza, che permetta di gestire anche l'incertezza



Ottimizzazione multi-obiettivo (Gavanelli, Univ. Ferrara) e aggregazione di preferenze multi-agente (Rossi, Venable, Pini)

- Molte tecniche in CP, AI, OR
- Goal: definire nuove tecniche ibride che sfruttino il meglio delle varie linee AI, CP, OR, per generare efficientemente la frontiera Pareto ottima
- Legata all'aggregazione di preferenze multi-agente
 - agente = obiettivo
- Proprietà dell'aggregazione multi-agente
 - Possibilità (di essere fair, o manipolabile, ...)
 - Proprietà computazionali: quanto è facile o difficile calcolare il risultato, o avere certe proprietà



Problemi di scheduling

(Cesta, Oddi, Policella, CNR Roma)

- Assegnamento di tempi di inizio e di fine per un insieme di attività soggette a certi vincoli
 - Esempio: vincoli temporali o disponibilità di risorse
- Soluzioni robuste che possano
 - resistere ad eventi imprevisti
 - Far generare nuove soluzioni rapidamente
 - Far preservare la qualità delle soluzioni
- Diversi livelli di preferenza per ogni attività