Configuration
Workshop
2015
Vienna

FLEXDIAG: AnyTime Diagnosis for Reconfiguration

A. Felfernig, R. Walter, and S. Reiterer

Contact: alexander.felfernig@ist.tugraz.at

Agenda

- Related Work & Goal
- Definitions
- Application Scenarios
- Direct Diagnosis with FastDiag
- Extension of FastDiag: FlexDiag
- Evaluation: FastDiag vs. FlexDiag
- Ongoing & Future Work
- Conclusions

Related Work & Goal

- Conflict-directed diagnosis with hitting sets [Reiter 1987]
- Diagnosis of configuration knowledge bases based on test cases [Felfernig et al. 2004]
- Direct diagnosis (FastDiag) [Felfernig et al. 2012]
- Determination of preferred diagnoses [Felfernig et al. 2009]
- Overview of further diagnosis approaches, e.g., [Fijany and Vatan 2004]

Direct diagnosis approaches (e.g., FastDiag) outperform conflict directed approaches without loss of predictive performance.

Our goal: further improve the efficiency of direct diagnosis by giving up minimality.

Configuration Task: Definition

Definition 1 (Configuration Task and Configuration). A configuration task can be defined as a CSP (V, D, C) where $\boxed{V} = \{v_1, v_2, ..., v_n\}$ is a set of variables, $\boxed{D} = \cup dom(v_i)$ represents domain definitions, and $\boxed{C} = \{c_1, c_2, ..., c_m\}$ is a set of constraints. Additionally, user requirements are represented by a set of constraints $\boxed{R} = \{r_1, r_2, ..., r_k\}$. A configuration (solution) for a configuration task is a set of assignments (constraints) $\boxed{S} = \{s_1 : v_1 = a_1, s_2 : v_2 = a_2, ..., s_n : v_n = a_n)\}$ where $a_i \in dom(v_i)$ which is consistent with $C \cup R$.

R: requirements
S: solution

Diagnosis: Definition

Definition 2 (Diagnosis). A $\boxed{\text{diagnosis } \Delta}$ (correction subset) is a subset of $S = \{s_1 : v_1 = a_1, s_2 : v_2 = a_2, ..., s_n : v_n = a_n\}$ such that $\boxed{S - \Delta \cup C \cup R_\rho \text{ is consistent}}$. $\boxed{\Delta \text{ is minimal}}$ if there does not exist a diagnosis Δ' with $\Delta' \subset \Delta$.

R_ρ: reconfiguration requirements

Reconfiguration: Definition

Definition 3 (Reconfiguration Task and Reconfiguration). A reconfiguration task can be defined as a CSP (V, D, C, S, R_ρ) where $\boxed{V}$ is a set of variables, $\boxed{D}$ represents variable domain definitions, $\boxed{C}$ is a set of constraints, $\boxed{S}$ represents an existing configuration, and $\boxed{R_\rho} = \{r'_1, r'_2, ..., r'_k\}$ (R_ρ consistent with C) represents a set of reconfiguration requirements. A reconfiguration is a variable assignment $\boxed{S_\Delta} = \{s_1 : v_1 = a'_1, s_2 : v_2 = a'_2, ..., s_l : v_l = a'_l\}$ where $s_i \in \Delta$, $a'_i \neq a_i$, and $\boxed{S - \Delta \cup S_\Delta \cup C \cup R_\rho}$ is consistent.

Algorithm for determining reconfigurations used in this paper: FlexDiag.

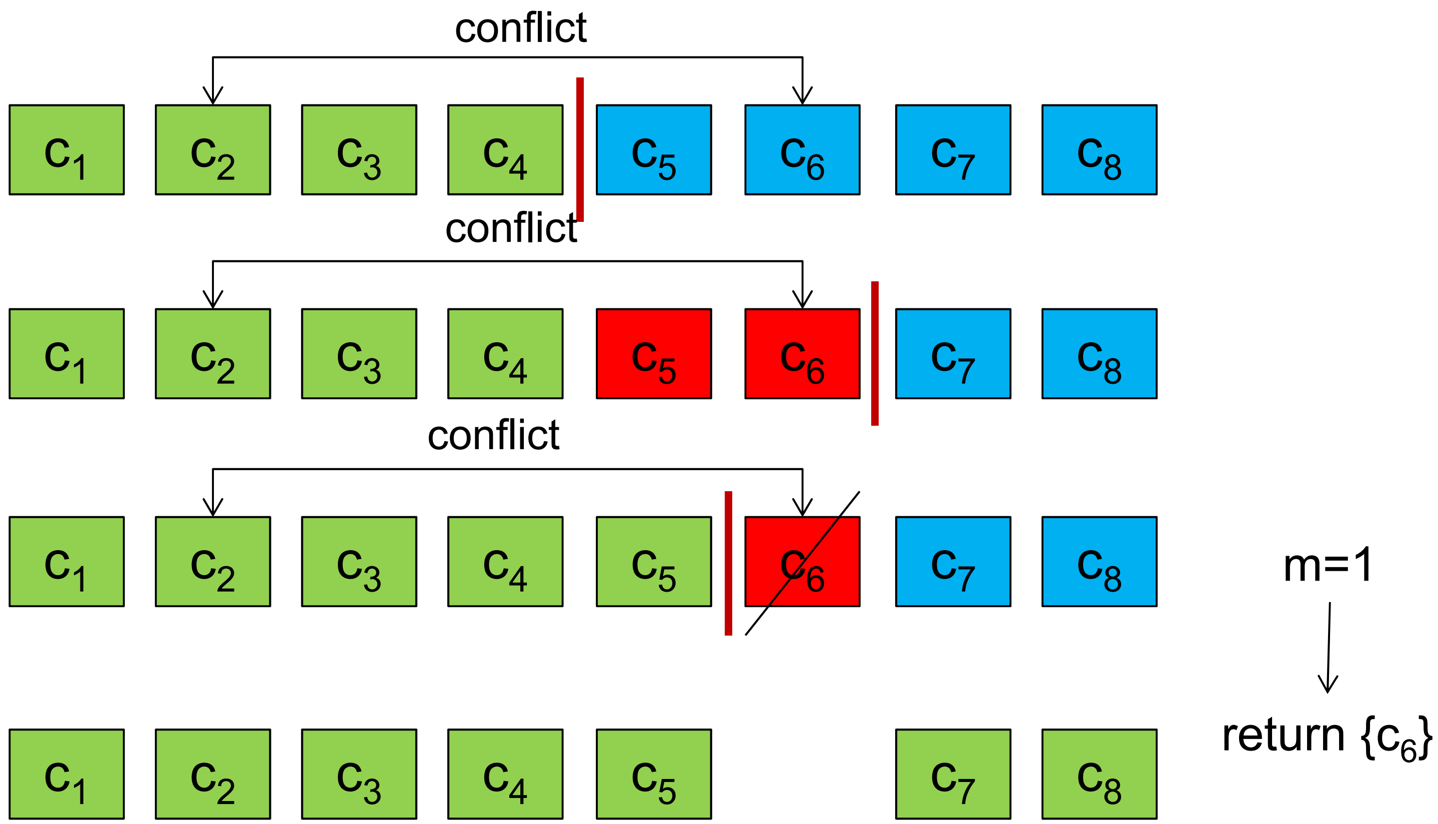
Scenario: Reconfiguration in Resource Allocation

- $V = \{fuel, paper, fireworks, pc, games, oil, roof, pipes\}$
- $dom(fuel) = dom(paper) = dom(fireworks) = dom(pc) = dom(games) = dom(oil) = dom(roof) = dom(pipes) = \{1, 2, 3\}$
- $C = \{c_1 : fireworks \neq fuel, c_2 : fireworks \neq paper, c_3 : fireworks \neq oil, c_4 : pipes = roof, c_5 : paper \neq fuel\}$
- $S = \{s_1 : pc = 3, s_2 : games = 1, s_3 : paper = 2, s_4 : fuel = 3, s_5 : fireworks = 1, s_6 : oil = 2, s_7 : roof = 1, s_8 : pipes = 1\}$
- $R_\rho = \{r'_1 : pc = games, r'_2 : paper = 3\}$

Scenario: Rescheduling

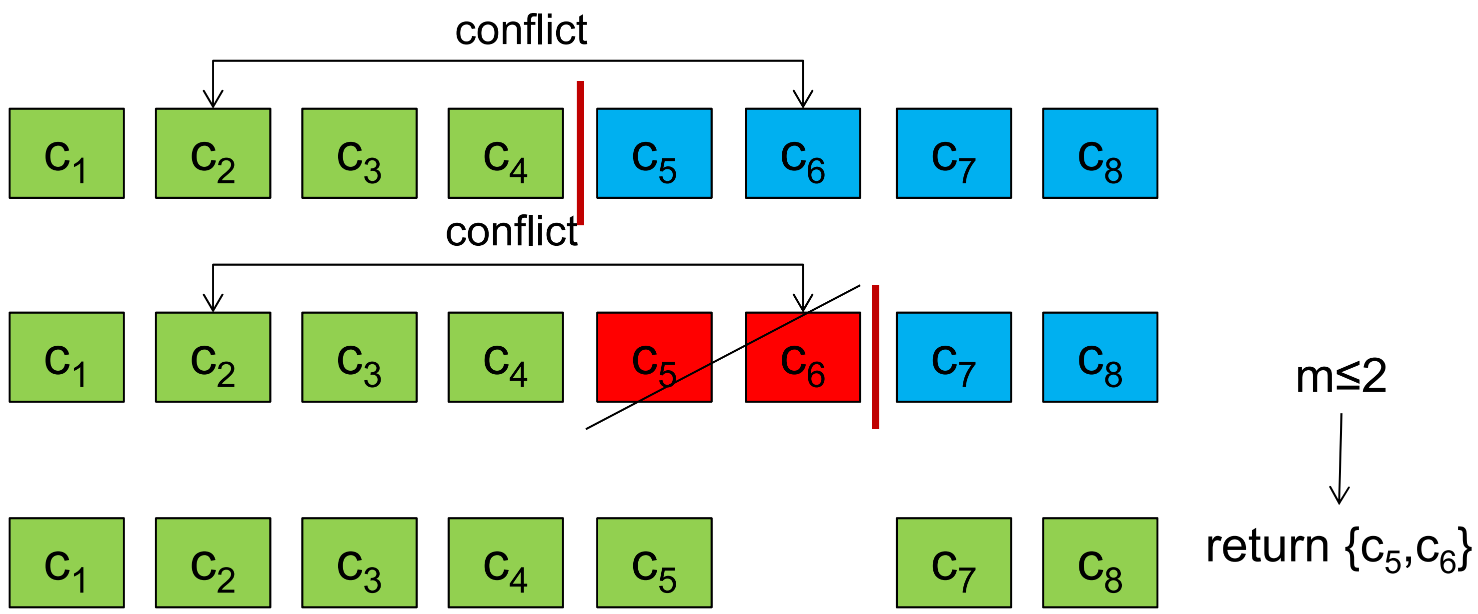
- $V = \{o_1m_1, o_1m_2, o_1m_3, o_2m_1, o_2m_2, o_2m_3, o_3m_1, o_3m_2, o_3m_3\}$
- $dom(o_1m_1) = dom(o_2m_1) = dom(o_3m_1) = \{1,2,3\}$. $dom(o_1m_2) = dom(o_2m_2) = dom(o_3m_2) = \{2,3,4\}$. $dom(o_1m_3) = dom(o_2m_3) = dom(o_3m_3) = \{3,4,5\}$.
- $C = \{c_1 : o_1m_1 < o_1m_2, c_2 : o_1m_2 < o_1m_3, c_3 : o_2m_1 < o_2m_2, c_4 : o_2m_2 < o_2m_3, c_5 : o_3m_1 < o_3m_2, c_6 : o_3m_2 < o_3m_3, c_7 : o_1m_1 \neq o_2m_1, c_8 : o_1m_1 \neq o_3m_1, c_9 : o_2m_1 \neq o_3m_1, c_{10} : o_1m_2 \neq o_2m_2, c_{11} : o_1m_2 \neq o_3m_2, c_{12} : o_2m_2 \neq o_3m_2, c_{13} : o_1m_3 \neq o_2m_3, c_{14} : o_1m_3 \neq o_3m_3, c_{15} : o_2m_3 \neq o_3m_3\}$
- $S = \{s_1 : o_1m_1 = 1, s_2 : o_1m_2 = 2, s_3 : o_1m_3 = 3, s_4 : o_2m_1 = 2, s_5 : o_2m_2 = 3, s_6 : o_2m_3 = 4, s_7 : o_3m_1 = 3, s_8 : o_3m_2 = 4, s_9 : o_3m_3 = 5\}$
- $R_\rho = \{r'_1 : o_3m_3 < 5\}$

Direct Diagnosis with FastDiag



Parameter m: determines the granularity of diagnosis elements.
FastDiag: m=1 (only minimal diagnoses are determined).

Extension of FastDiag: FlexDiag



In FlexDiag m can be ≥ 1: the higher the value of m, the higher the probability that a diagnosis includes irrelevant elements. If a diagnosis includes irrelevant elements, it is not minimal (theoretical analysis given in the paper).

Evaluation: FastDiag vs. FlexDiag

avg. runtime (ms)				
m=1	m=2	m=4	m=6	m=10
772	647	561	429	350
359	273	203	180	85
733	593	343	304	195
2176	1864	1451	1419	819

$$minimality(\Delta) = \frac{|\Delta_{min}|}{|\Delta|}$$

$$accuracy(\Delta) = \frac{|\Delta \cap \Delta_{min}|}{|\Delta_{min}|}$$

quality of approximation of the FastDiag result.

minimality (accuracy)				
m=1	m=2	m=4	m=6	m=10
1.0(1.0)	0.56(1.0)	0.45(1.0)	0.21(1.0)	0.15(1.0)
1.0(1.0)	0.55(0.91)	0.31(0.91)	0.30(0.83)	0.41(0.58)
1.0(1.0)	0.64(0.91)	0.83(0.55)	0.83(0.55)	1.07(0.30)
1.0(1.0)	0.66(0.89)	0.43(0.82)	0.41(0.67)	0.33(0.65)

Four example knowledge bases (feature models) used for the evaluation: www.splot-research.org.

Ongoing & Future Work

- Ongoing work: evaluation of FlexDiag with industrial configuration benchmark
- Further evaluation metrics to better estimate the quality of diagnoses
- Evaluation of FlexDiag in the context of knowledge engineering scenarios

Conclusions

- Efficiency of diagnosis algorithms crucial in different scenarios, e.g., scheduling and resource allocation
- FlexDiag is an extension of an existing direct diagnosis algorithm (FastDiag)
- FlexDiag allows performance improvements by accepting a potential loss of diagnosis minimality
- Anytime aspect: control of the upper limits of the number of consistency checks

Thank You!

References

- R. Reiter, ‘A theory of diagnosis from first principles’, Artificial Intelligence, 32(1), 57–95, (1987).
- A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner, ‘Consistency-based diagnosis of configuration knowledge bases’, Artificial Intelligence, 152(2), 213–234, (2004).
- A. Felfernig, M. Schubert, and C. Zehentner, ‘An efficient diagnosis algorithm for inconsistent constraint sets’, Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), 26(1), 53–62, (2012).
- A. Felfernig, M. Schubert, G. Friedrich, M. Mandl, M. Mairitsch, and E. Teppan, ‘Plausible repairs for inconsistent requirements’, in 21st International Joint Conference on Artificial Intelligence (IJCAI’09), pp. 791–796, Pasadena, CA, USA, (2009).
- A. Fijany and F. Vatan, ‘New approaches for efficient solution of hitting set problem’, in Winter International Symposium on Information and Communication Technologies, pp. 1–6, Cancun, Mexico, (2004). Trinity College Dublin.